Graph Generative Models for Fast Detector Simulations in Particle Physics

Ali Hariri
American University of Beirut
aah71@mail.aub.edu

Darya Dyachkova
Minerva Schools at KGI
darya.dyachkova@cern.ch

Sergei Gleyzer
University of Alabama
sgleyzer@ua.edu

Mariette Awad
American University of Beirut
ma162@aub.edu.lb

Daria Morozova
Pangea Formazione
daria.morozova@pangeaformazione.it

Abstract

Accurate and fast simulation of particle physics processes is crucial for high-energy physics community. Simulating the particle showers and interactions in the detector is both time consuming and computationally expensive. Classical fast simulation approaches based on non-parametric approaches can improve the speed of the full simulation but suffer from lower levels of fidelity. For this reason, alternative methods based on machine learning can provide faster solutions, while maintaining a high level of fidelity. The main goal of a fast simulator is to map the events from the generation level directly to the reconstruction level. We introduce a graph neural network-based autoencoder model that provides effective reconstruction of calorimeter deposits using the earth mover distance metric.

1 Introduction

High-energy collisions taking place at the Large Hadron Collider (LHC) are very complex in nature. To reconstruct collision events, advanced probabilistic models have been developed describing them as follows:

$$\begin{aligned} p(\text{r-particles}|\theta) = \int R(\text{r-particles}|\text{particles}) H(\text{particles}|\text{partons}) \\ \times P(\text{partons}|\theta)\ d\text{particles}\ d\text{partons} \end{aligned} \tag{1}$$

where P represents the probability density of observing a set of reconstruction particles given a point in the parameter space, H refers to the hadronization process where mapping from the parton to the particle level occurs and R(particles) is the detector response [1].The latter has so far been approximated through complex software toolkits performing full simulation of the passage of particles through matter such as Geant [2]. Such simulations are highly accurate yet prohibitively complex and require extensive computing resources and therefore, fast simulation techniques have been introduced as a computationally efficient and faster alternative. Typically, fast simulations are performed with parametric methods for detector response, while providing user interfaces to specify detector properties and calorimeter segmentation [3].

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

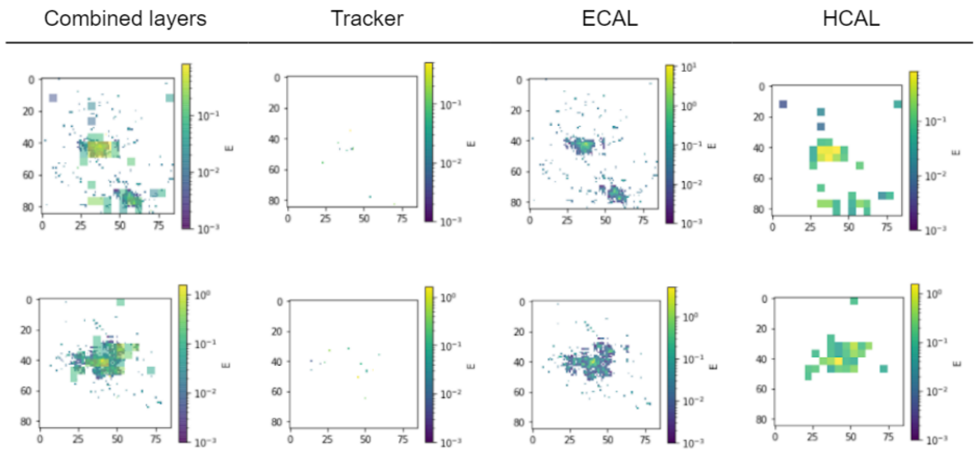

Figure 1: The comparison between the energy deposits in the Tracker, ECAL, and HCAL calorimeter layers. The leftmost images show the combined data from all the three layers

Recent advances in the field of deep learning and computer vision have produced notable applications in particle physics [4]. The potential of generative models for simulating collision events is a very active area of research. In [5], a combination of a Variational Autoencoder (VAE) and a Generative Adversarial Network (GAN) is used to simulate electromagnetic showers in calorimeters. Other studies focus on GANs for QCD Dijet events [6] and hadronic jets [7]. In our study, we make use of VAEs [8] and geometric deep learning [9] to learn a compressed representation of the data to be used for reconstruction of high-energy physics events. As detector data in non-Euclidean by nature, we use geometric deep learning techniques including spatial graph convolutional layers [10] to learn the properties of the graph-like jets and spectral clustering layers to compress these graphs into smaller, more representative nodes.

2 Data

2.1 Definition

In this work, we make use of the CMS Open Data release [11] - publicly accessible data from the LHC experiments. We consider the boosted top quark jets produced using Pythia 6, a program for generating particle collisions events. The data was transformed into image-based form, specifying the location and values of energy deposits in the calorimeter by following the prescription in [12]. 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. We aim to reconstruct jets that deposit their energy in the calorimeters, initially focusing only on the ECAL subdetector hits. The non-zero hits within this 125x125 array correspond to the hit energy of the corresponding particle shower deposited at that specific grid cell.

2.2 Pre-processing

For the data loading step, we utilize the DataLoader objects from PyTorch library. The split of the dataset follows random shuffling when loading the data from each file, in addition to each file containing randomly selected data. We then create three separate objects - the training, testing, and validation loader respectively.

We pre-process the data by selecting the non-zero hit locations within the array, providing their respective x and y locations as per the calorimeter segmentation. Afterwards, we concatenate the x,y locations with their corresponding hit energy at that location. At this stage, each sample has the shape Nx3 where N is the number of non-zero particle hits within the detector for one specific sample jet, with each sample containing 3 features: the x,y locations and their hit energies, respectively. In the next section we show that N is also the number of nodes within one graph representing a jet.

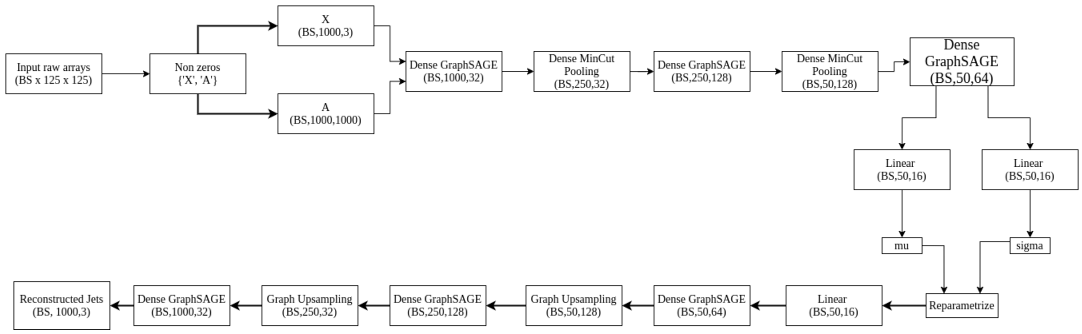

Figure 2: Model architecture of the Graph Variational Autoencoder showing GraphSAGE layers and pooling blocks

3 Model architectures

In this study, we make use of geometric deep learning methods whereas convolution operations are translated to non-Euclidean structures [13], in contrast to previous studies where the entire grid-like structure of a given detector is used as an input to fully-connected or convolutional layers for classification or regression purposes. Therefore, we consider particle hits within a detector to be interconnected nodes in a graph. In contrast to molecular chemistry, where the graph topology is constrained by the molecule shape [14], jets in particle collisions are not characterized by such pre-defined topology. We proceed by connecting each node to its k-nearest neighbours based on 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. To learn the properties of these jets as graphs in addition to a compressed representation to be used in an encoder-decoder architecture, we develop a Graph VAE architecture whose encoder embeds the node features into latent space dimensions through Dense GraphSAGE layers [15], then compresses them into smaller dimensions using dense mincut graph pooling operations inspired by [16] where spectral clustering of the graph nodes is performed. In the next stage, a decoder performs decoding of the latent space compressed nodes to obtain upsampled feature matrix X and adjacency matrix A, respectively as follows:

$$X^{rec} = SX^{Pooled}; A^{rec} = SA^{Pooled}S^T \tag{2}$$

where S is a learned cluster assignment matrix similar to the one defined in [16]. A pictorial representation of our model is given in Figure 2.

To proceed with training, we choose k=4 as the number of nearest neighbours to be connected to each node. In addition, we use the Adam optimizer with a learning rate of 0.001. Finally, our loss function includes the MSE loss between node features on the one hand, and the Kullback-Leibler divergence between the latent space and the real P(z|x) distribution. Prior to training, we split our dataset into 70% training, 20% validation, 10% testing.

4 Metrics and Results

First introduced in 2019, the Earth Mover Distance (EMD) metric describes the space of two collider events [17]. It represents the minimum "work" needed to be applied by the movements of energy f_{ij} from particle i in one event to particle j in the other so that event $\mathcal{E}$ is rearranged into event $\mathcal{E}'$. We therefore use the EMD metric to assess the quality of reconstructed jets produced by our fast simulation.

We obtain our results after training on a Tesla P100 GPU. In Figure 3 we show the reconstruction result for several simulated jets from our GVAE model. The plots show that our model's decoder is able to accurately reconstruct the jets from compressed latent vectors, both in terms of locations and energy values. In addition, Figure 4 shows the EMD values corresponding for 4800 reconstructed jets. Relatively low values of the EMD imply a high-level of similarity between GVAE reconstructed and fully-simulated jets. The GVAE model spends a total of 0.1235 seconds on a batch of 64 jets, which is orders of magnitude smaller than running full simulation. We therefore conclude that our approach

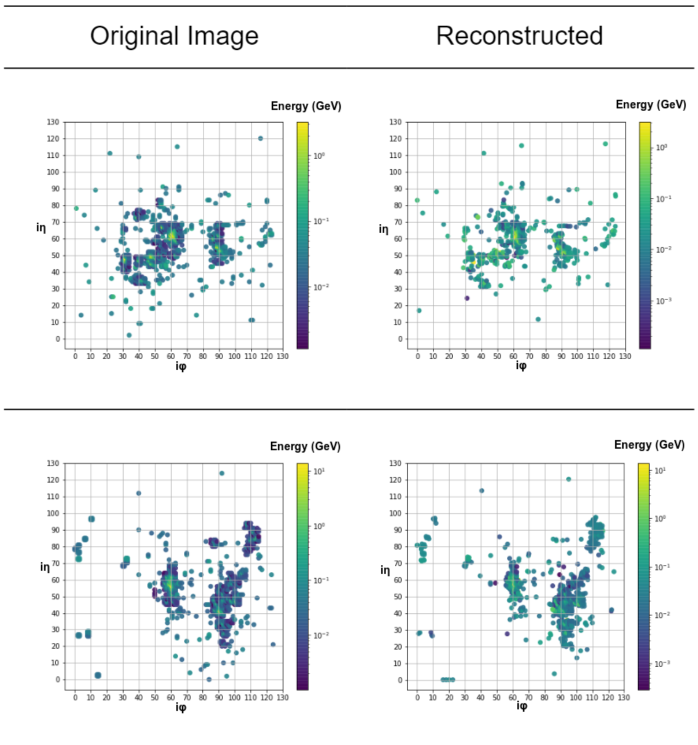

Figure 3: Real simulated jet (left) compared to the Reconstructed GVAE jet (right) in the detector

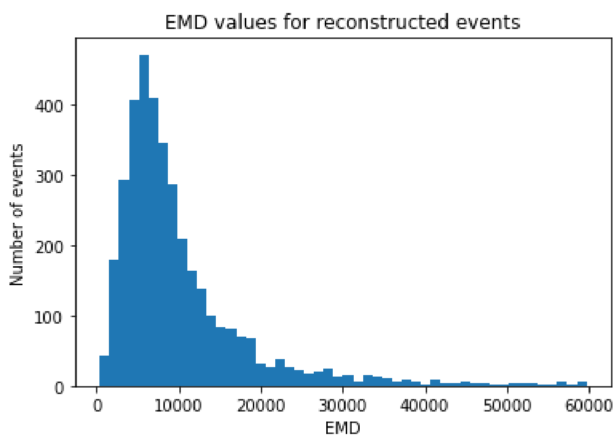

Figure 4: Earth Mover Distance

is successful in reproducing particle physics jet data at high-levels of fidelity and with acceptably low inference times.

For benchmarking, we compare the execution time to the full Monte Carlo simulations used for approximating the probability density function from eq.1. Such simulation takes 45 seconds for the event batch of the same size. In contrast to this results, the graph method takes around 0.1 second for the inference, which is over 400% speedup (Figure 5).

In this proof-of-concept work we shed light on the potential of graph-based architectures for representing particle jets resulting from high-energy collisions. 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 the training of the model. Through spatial convolution, the model is

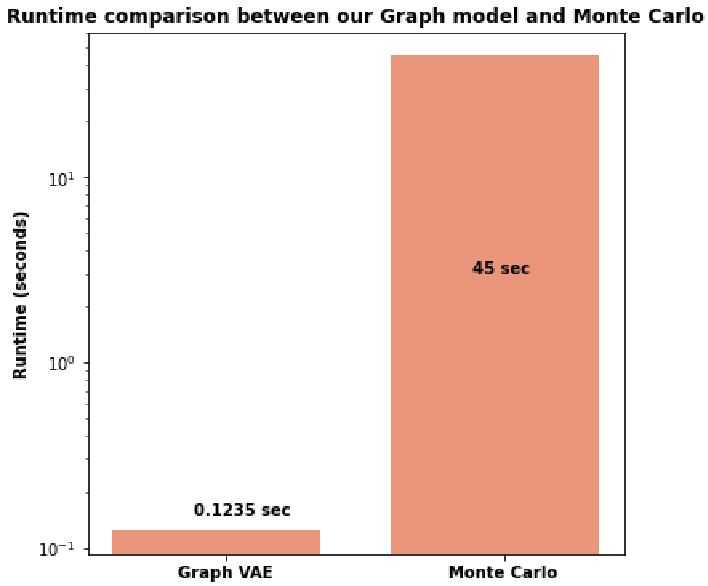

Figure 5: Runtime comparison between Graph VAE and traditional Monte Carlo methods for boosted top quark jet samples

able to learn the interactions between particle hits forming the topology. The latter is sequentially compressed by means of mincut pooling to preserve the most representative nodes in latent space. Finally, a trained decoder upsamples the compressed vectors to the original reconstruction, leading to a proof-of-concept simulator. We have demonstrated the initial application and will discuss additional extensions to the full detector in an upcoming more detailed publication.

5 Broader Impact

This work is an open-source project, and has a potential to impact many researchers who rely on particle simulations for physics studies. In terms of ethical and future societal consequences, the computational efficiency provided by the generative model presented in this work, allows to overcome computational and time constraints. In the absence of adequate computational resources, this type of simulation would be an advantage to the research, as it provides an opportunity for a speedup of obtaining the results. At the same time, due to a need for some computational requirements of our implementation, those with very limited access to computing resources may be at a disadvantage.

6 Acknowledgments

This paper and the research behind it would not have been possible without the support of the following people: Michael Andrews (Carnegie Mellon University), Emanuele Usai and Bjorn Burkle (Brown University), Christian Hundt and Giuseppe Fiameni (NVIDIA AI Technology center).
We thank them for their expertise and support. Ali Hariri was a participant in the Google Summer of Code 2020 Program.

References

[1] S. Gleyzer, Prosper Orlando R. D., S. H.B., Sekmen, and O.A. Zapata. Physics at tev colliders. new physics working group report. contribution 15. falcon: towards an ultra fast non-parametric detector simulator. *arXiv preprint arXiv:1203.1488*, 2016.

[2] R. Brun, R. Hagelberg, M. Hansroul, and J. Lassalle. Simulation program for particle physics experiments, geant : user guide and reference manual. *Report Number CERN-DD-78-2*, 1978.

[3] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, and M. Selvaggi. Delphes 3, a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2), 57., 2014.

[4] Albertsson et. al. Machine learning in high energy physics community white paper, 2019.

[5] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, and K. Krüger. Getting high: High fidelity simulation of high granularity calorimeters with high speed. *arXiv preprint arXiv:2005.05334*, 2020.

[6] R. Di Sipio, M. F. Giannelli, S. K Haghighat, and S. Palazzo. Dijetgan: a generative-adversarial network approach for the simulation of qcd dijet events at the lhc. *Journal of High Energy Physics, 2019(8), 110.*, 2019.

[7] P. Musella and F. Pandolfi. Fast and accurate simulation of particle detectors using generative adversarial networks. *Comput Softw Big Sci 2: 8*, 2018.

[8] D. Kingma and M. Welling. An introductionn to variational autoencoders. *Foundations and Trends in Machine Learning*, Vol. 12 (2019) No. 4:pp 307–392, 2019.

[9] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph Neural Networks: A Review of Methods and Applications. *arXiv e-prints*, page arXiv:1812.08434, December 2018.

[10] T. Danel, P. Spurek, J. Tabor, M. Smieja, L. Struski, A. Slowik, and L. Maziarka. Spatial graph convolutional networks. *arXiv e-prints, arXiv-1909*, 2019.

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

[12] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos. End-to-end physics event classification with the cms open data: Applying image-based deep learning on detector data to directly classify collision events at the lhc. *Computing and Software for Big Science*, 4(1):1–14, 2020.

[13] D. Guest, K. Cranmer, and D. Whiteson. Deep learning and its application to lhc physics. *Annual Review of Nuclear and Particle Science*, 68:161–181, 2018.

[14] J. Wengong, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.

[15] W. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Conference Notes from 31st Conference on Neural Information Processing Systems*, 2017.

[16] F. Bianchi, D. Grattarola, and C. Alippi. Spectral clustering with graph neural networks for graph pooling. *Proceedings of Machine Learning Research*, 2010.

[17] P. Komiske, E. Metodiev, and J. Thaler. The metric space of collider events. *Physical review letters*, 123(4), 041801, 2019.