Graph Variational neural network for fast end to end detector simulation

Dinesh PR

October 31, 2023

towards partial fulfillment of course requirements for PH 561 (Fall 2023) at the University of Alabama

1 Introduction and Background

In the current work, we build and study a generative model based on graph variational autoencoders to accelerate simulations of boosted top quark production in the Large Hadron Collider (LHC)[30]. While it trains on dataset obtained from existing Monte Carlo-based generation and detector simulation models, we aim to accelerate re-sampling the same without compromising on the accuracy and fidelity. A successful implementation of the model can be used to program customized FPGA-based triggers that can avoid pile-up of unnecessary background hits during collision experiments at the LHC.

1.1 CMS detector and collision "images"

The CMS detector[9] in the LHC is arranged in cylindrical sections of different types of subdetectors (see fig 1), with the central axis along the line of collision between proton beams. This includes both barrel and endcap (see figure) sections of each subdetector. We look at 3 particular subdetectors as relevant to this study- the inner tracking system (Tracker), the electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL)[3].

Tracker is the innermost series of finely segmented silicon wafers that (non-destructively) detect positional tracks left by particles formed from a collision beam, deciphering practically nothing about their energies. The "pixels" in the Tracker are labelled using (z, ϕ) and (ρ, ϕ) for the barrel and endcap sections respectively- where z denotes length along the axis, ρ denotes the perpendicular radius and ϕ denotes the azimuthal angle.

Surrounding the Tracker is the ECAL subdetector which captures the incoming photons and electrons, and measures their energies using scintillating lead tungstate crystals. In the barrel section, it is segmented by pseudorapidity ($i\eta_{EB}$) and azimuthal angle ($i\phi_{EB}$) and spans within $|\eta| < 1.479$ making it a

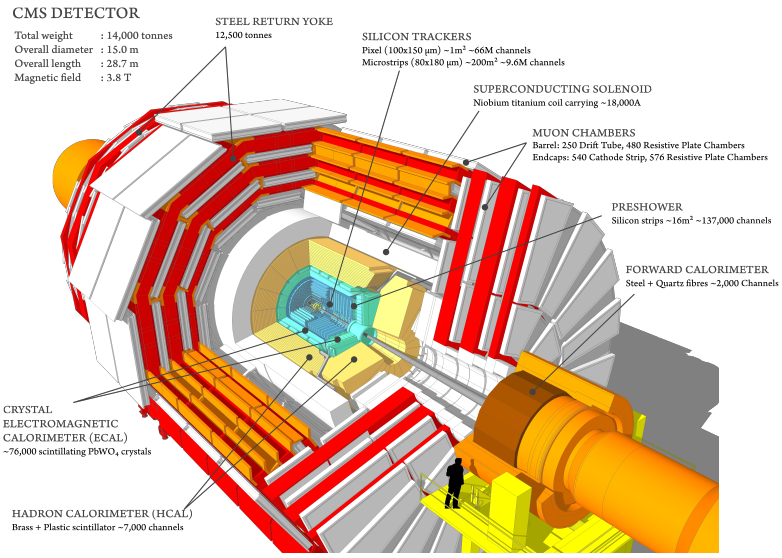

Figure 1: Schematics of the CMS detector at LHC, CERN. Source: `https://cms.cern/detector`

170×360 grid of "pixels". The endcap is labelled using (iX, iY) as a Cartesian grid of pixels arranged in a circle. Psudorapidity is an alternative to the polar angle (θ) frequently used in spherical coordinates and they're related by:

$$\eta = -\ln\left[\tan\frac{\theta}{2}\right]$$

The HCAL subdetector encloses the ECAL and measures the energies of hadronic particles (mostly charged pions and kaons) using scintillating brass towers. The barrel section encloses the pseudorapidity range $|\eta| < 3$. The pixels in the barrel section of HCAL produce an image that's more coarsely segmented with roughly 1 HCAL pixel per 5 ECAL pixels along both the $i\phi$ and $i\eta$ directions.

1.2 Event generation for LHC

Quantum field theory[32, 31, 28] is the mathematical framework based on which theoretical as well as numerical computations of physical quantities observed in high energy experiments are performed. In particular, the values of Green's functions or S matrices determine the various scattering *cross sections* for the final states of particles observed in different detectors. The simulation pipeline can be divided into several steps[6].

Hard scattering The *hard scattering* component probes the smallest length scales around the collision center. The cross sections are usually determined

using ab inito (first principle), perturbative QFT calculations at the highest order of accuracy feasible in terms of the coupling constants. Terms of the perturbation series can be represented visually using Feynman diagrams[15] and their values are often plagued by problems with IR divergences in loop integrals which often cancel analytically with tree-level counterparts. If analytic solutions are not available, Monte Carlo-based simulations[19, 24, 2] are used for which there are ever-increasing quests for improvements in efficiency and numerical stability.

Secondary interactions and decays Since detectors are present at much larger distances from the center, there can be additional collisions and decays beyond the hard scattering which need to simulated accurately. Factorization theorems[11, 10] give rise to correlations between the momenta of partonic (quarks/gluons) currents of different flavours, which are encoded semi-classically as integrals of parton distribution functions (PDF). Their time-evolution is modelled using Monte Carlo-based algorithms like parton showers [25]and dipole showers[20].

Numerical libraries and open simulated datasets Many generic software libraries that simulate parts of the event generation chain we briefly summarized are available. Most popular general-purpose packages available for the same include SHERPA[16], HERWIG[12] and PYTHIA[29]. They're run in combination with software that simulate the behavior of particular detectors (in say, CMS) like GEANT[1]. While the Monte Carlo-based generators are highly accurate, they often scale poorly with the collision energy, luminosity and detector count. Faster detector simulation libraries like DELPHES[14] tend to make approximations that trade off accuracy for speed. Full simulation of collider experiments using theory-driven collision event generators and experimentally optimized detector simulators is referred to as end-to-end event generation.

1.3 Generative ML and Graph VAE

Generative networks[26] are machine learning tools used to sample from a learned data distribution from the training set. While Monte Carlo generators asymptotically (in the limit of infinite samples) converge to the actual distribution for averaged quantities, they can suffer from large autocorrelations between successive samples and poor accuracies for non-averaged quantities for practically finite number of samples[8]. On the other hand, most generative ML models sample with no autocorrelations but their asymptotic closeness to the target distribution depends on the model architecture, training parameters, loss function and the quality of training data. Most popular classes of generative networks include variational autoencoders[21] (VAE), generative adversarial networks[13] (GAN) and normalizing flows (NF)[27]. VAEs and GANs generally offer highly expressive models and have already been used in many high energy physics applications including end to end event generation[5]. NFs are known to be less

expressive but allow us to explicitly compute likelihoods of different samples and can be inverted for reverse simulations, which have been exploited in many HEP applications.

Variational autoencoders VAEs use neural networks to *encode* or compress high dimensional input data to lower dimensional latent space which is also the parameter space for an easily sampled probability distribution (like Gaussian) and these samples are then *decoded* by other neural networks back to the original input space. They are typically trained to minimize a distance between the input and the (decoded) output data, like mean squared error (MSE), along with a prior likelihood term into an evidence lower bound (ELBO) estimator. In our work, we use graph-based neural networks for encoders and decoders in a VAE, an idea which has already been explored[18].

Graph convolution and pooling operations In the geometric deep learning framework[4], we tend to use operations that preserve the structure and symmetries of the input data. In the case of graph data, we'd like to preserve the neighborhood of each node and require that the output be invariant to permutations within the neighborhoods. Spectral convolutions[22] were initially developed as compatible operations on graph data but they tend to scale poorly with the number of nodes and edges. Message passing layers like GCN [23]and SAGE[17] were developed to compute convolutions in polynomial time on the size of input data. We convert the sparse detector image by taking the non-zero pixels as nodes and putting edges between k nearest neighbors (knn) of every node. We use SAGE convolution layers, which is ideal for aggregating information from the neighborhood of every node when the edge count ($\sim k$) is very small. The operation is mathematically described using

$$\boldsymbol{x}_i' = \boldsymbol{W}_1 \boldsymbol{x}_i + \boldsymbol{W}_2 . \mathrm{agg}_{j \in \mathcal{N}(i)} \boldsymbol{x}_j$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are weights of the layer (which are optimized during training) and 'agg' is an aggregating operation like mean or max.

2 Data and Pre-processing

In this work, we focus on the quark-gluon shower dataset for the CMS detector, available on the CERN open data portal[7] and was simulated using PYTHIA and GEANT. Due to lack of availability of Tracker hits in CERN open data. <WIP: There's a straightforward way to encode $(i\eta, i\phi)$ instead of (x, y) coordinates of pre-processed image as done in [18]- can potentially improve model performance>

3 Results

<WIP: Description of training hyperparameters like optimizer, learning rate schedules and regularization terms in loss function. Inference speed will be plotted against input (total graph nodes) and batch sizes. Comparisons of reconstructed image against training image will be plotted to illustrate fidelity of the model>

4 Conclusion

<WIP>

References

[1] Sea Agostinelli et al. "GEANT4, a simulation toolkit". In: *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303.

[2] Johan Alwall et al. "MadGraph 5: going beyond". In: *Journal of High Energy Physics* 2011.6 (2011), pp. 1–40.

[3] M Andrews et al. "End-to-end event classification of high-energy physics data". In: *Journal of Physics: Conference Series*. Vol. 1085. 4. IOP Publishing. 2018, p. 042022.

[4] Michael M Bronstein et al. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[5] Anja Butter et al. "Machine learning and LHC event generation". In: *SciPost Physics* 14.4 (2023), p. 079.

[6] JM Campbell et al. "Event generators for high-energy physics experiments". In: *arXiv preprint arXiv:2203.11110* (2022).

[7] *CERN Open data portal*. URL: `http://opendata.cern.ch`.

[8] Siddhartha Chib and Edward Greenberg. "Understanding the metropolis-hastings algorithm". In: *The american statistician* 49.4 (1995), pp. 327–335.

[9] CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Jinst* 3 (2008), S08004.

[10] John Collins. *Foundations of perturbative QCD*. Cambridge University Press, 2011.

[11] John C Collins, Davison E Soper, and George Sterman. "Factorization of hard processes in QCD". In: *Perturbative QCD*. World Scientific, 1989, pp. 1–91.

[12] Gennaro Corcella et al. "HERWIG 6: an event generator for hadron emission reactions with interfering gluons (including supersymmetric processes)". In: *Journal of High Energy Physics* 2001.01 (2001), p. 010.

[13] Antonia Creswell et al. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[14] J De Favereau et al. "DELPHES 3: a modular framework for fast simulation of a generic collider experiment". In: *Journal of High Energy Physics* 2014.2 (2014), pp. 1–26.

[15] Richard Phillips Feynman. *Selected papers of Richard Feynman: with commentary*. Vol. 27. World Scientific, 2000.

[16] Tanju Gleisberg et al. "Event generation with SHERPA 1.1". In: *Journal of High Energy Physics* 2009.02 (2009), p. 007.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[18] Ali Hariri, Darya Dyachkova, and Sergei Gleyzer. "Graph generative models for fast detector simulations in high energy physics". In: *arXiv preprint arXiv:2104.01725* (2021).

[19] Aggeliki Kanaki and Costas G Papadopoulos. "HELAC: a package to compute electroweak helicity amplitudes". In: *Computer physics communications* 132.3 (2000), pp. 306–315.

[20] Hamid Kharraziha and Leif Lönnblad. "The linked dipole chain Monte Carlo". In: *Journal of High Energy Physics* 1998.03 (1998), p. 006.

[21] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[22] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[23] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[24] Frank Krauss, Ralf Kuhn, and Gerhard Soff. "AMEGIC++ 1.0, a matrix element generator in C++". In: *Journal of High Energy Physics* 2002.02 (2002), p. 044.

[25] G Marchesini and Bryan R Webber. "Monte Carlo simulation of general hard processes with coherent QCD radiation". In: *Nuclear Physics B* 310.3-4 (1988), pp. 461–526.

[26] Achraf Oussidi and Azeddine Elhassouny. "Deep generative models: Survey". In: *2018 International conference on intelligent systems and computer vision (ISCV)*. IEEE. 2018, pp. 1–8.

[27] George Papamakarios et al. "Normalizing flows for probabilistic modeling and inference". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 2617–2680.

[28] Michael E Peskin. *An introduction to quantum field theory*. CRC press, 2018.

[29] Torbjörn Sjöstrand et al. "High-energy-physics event generation with PYTHIA 6.1". In: *Computer Physics Communications* 135.2 (2001), pp. 238–259.

[30] *The Large Hadron Collider at CERN*. URL: `https://home.web.cern.ch/science/accelerators/large-hadron-collider`.

[31] Steven Weinberg. *The quantum theory of fields*. Vol. 2. Cambridge university press, 1995.

[32] Anthony Zee. *Quantum field theory in a nutshell*. Vol. 7. Princeton university press, 2010.