
Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors

Xiangyang Ju, Steven Farrell, Paolo Calafiura, Daniel Murnane, Prabhat
Lawrence Berkeley National Laboratory
Berkeley, CA
xju@lbl.gov

**Lindsey Gray, Thomas Klijnsma, Kevin Pedro, Giuseppe Cerati,
Jim Kowalkowski, Gabriel Perdue, Panagiotis Spentzouris, Nhan Tran**
Fermi National Accelerator Laboratory
Batavia, IL

Jean-Roch Vlimant, Alexander Zlokapa, Joosep Pata, Maria Spiropulu
California Institute of Technology
Pasadena, CA

Sitong An
CERN, Geneva, Switzerland &
Carnegie Mellon University, Pittsburgh, PA

Adam Aurisano, Jeremy Hewes
University of Cincinnati
Cincinnati, OH

Aristeidis Tsaris
Oak Ridge National Laboratory
Oak Ridge, TN

Kasuhiko Terao, Tracy Usher
SLAC National Accelerator Laboratory
Menlo Park, CA

Abstract

Pattern recognition problems in high energy physics are notably different from traditional machine learning applications in computer vision. Reconstruction algorithms identify and measure the kinematic properties of particles produced in high energy collisions and recorded with complex detector systems. Two critical applications are the reconstruction of charged particle trajectories in tracking detectors and the reconstruction of particle showers in calorimeters. These two problems have unique challenges and characteristics, but both have high dimensionality, high degree of sparsity, and complex geometric layouts. Graph Neural Networks (GNNs) are a relatively new class of deep learning architectures which can deal with such data effectively, allowing scientists to incorporate domain knowledge in a graph structure and learn powerful representations leveraging that structure to identify patterns of interest. In this work we demonstrate the applicability of GNNs to these two diverse particle reconstruction problems.

1 Introduction

The reconstruction of particle collision events in high energy physics experiments such as those at the Large Hadron Collider [9] involves challenging pattern recognition tasks. Particle detectors such as ATLAS [1] and CMS [6] are 40m long, 25m diameter instruments with complex geometry and sparse

high dimensional data. Specialized detector sub-systems and algorithms are used to reconstruct the different types and properties of particles produced in collisions. For example, charged particle trajectories are reconstructed from spacepoint measurements (“hits”) in tracking detectors, and particle showers are reconstructed from clusters in calorimeters. The upgraded High-Luminosity LHC [2], expected to begin operation in 2026, will deliver increased collision data rates and volumes to the experiments, presenting challenges for current reconstruction solutions.

The traditional approach to particle track reconstruction utilizes combinatorial search algorithms guided by a Kalman Filter. These algorithms are highly tuned for physics performance in today’s LHC conditions, but are inherently sequential and scale poorly to the expected HL-LHC conditions with $\mathcal{O}(10^4)$ particles and $\mathcal{O}(10^5)$ hits in each event. The expected challenges of deploying the traditional tracking solutions to HL-LHC data motivated the formation of the HEP.TrkX project to investigate potential new solutions with modern deep learning techniques [11, 10].

In this paper we present our work to apply Graph Neural Networks (GNNs) to the particle track and shower reconstruction problems. GNNs were first introduced in [16] and have been applied to a growing variety of problems including social networks, knowledge graphs, recommender systems, and 3D shape analysis [19, 5]. They were first studied for particle tracking applications in [10] and were also studied for the problem of particle and event classification in [3, 13, 15, 7].

2 Methodology

For both tracking and calorimeter cluster problems, we define a graph representation of the data using individual detector measurements as nodes and then constructing edges between nodes with heuristics based on domain knowledge. The GNN models used are based on the Interaction Networks architecture [4]. The primary task of the GNN is to associate detector elements together by classifying the edges of the graph.

2.1 Tracking

For track finding, we consider only tracks and hits in the barrel region of the detector. The graph is constructed so that the nodes are the hits recorded by the detector and the edges are connections of the hits between adjacent detector layers that pass a pre-defined filter that is tuned to be efficient for tracks resulting from high transverse momentum particles. In the input graphs, node features are the three cylindrical coordinates (r, ϕ, z) and edge features are the difference of the coordinates $(\Delta\eta, \Delta\phi)$. The edge labels are 1 if two hits come from the same track, and 0 otherwise.

The GNN architecture has three components: an encoder which transforms input node and edge features into their latent representations, a graph module which performs message passing to update latent features, and an output module which computes edge classification scores. A diagram of the architecture is shown in figure 1. The encoder uses two fully-connected 2-layer networks for transforming node and edge features, respectively. The initial latent features of the nodes and edges are collectively named H_0 . The graph module is applied recursively to the latent features. At each iteration i the initial features H_0 are concatenated onto the current features H_i . This shortcut connection was empirically found to improve model performance. The graph module also uses two fully-connected 2-layer networks, one which computes updated edge features and one which computes updated node features using aggregated incoming edge features. After N iterations of the graph module, the output module takes the last latent features H_N and uses a 2-layer fully-connected network to produce classification scores for every edge. All fully-connected layers use a hidden size of 128 and ReLU activation functions, except the final layer of the output module which uses sigmoid activation. We found that using $N = 8$ graph iterations gave the best model performance.

2.2 Calorimeter clustering

Similar methodologies to those in tracking can be employed to identify energy deposits that should be clustered together to form physically meaningful objects. In fact, with some minor modifications the same variety of edge classification networks used in the tracking problems described above can be immediately applied to the problem of calorimetry. If instead of requiring the final output graph to be a collection of tracks, we allow the output graph to be a mesh on a point cloud and label those edges, an energy cluster can be identified. Moreover, instead of simply being ‘true’ or ‘false’ edges

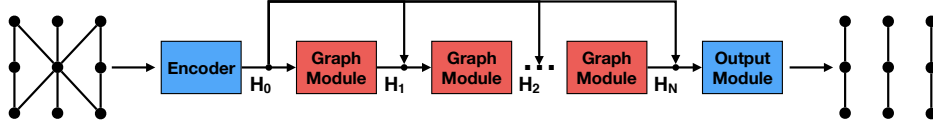


Figure 1: The Graph Neural Network architecture used for tracking.

the particle type of the edge can also be encoded and inferred. This can be achieved with a graph neural network using architectures similar to those demonstrated for tracking as well as networks where the graph is determined dynamically [14]. Here we will focus on the static graph networks and demonstrate results for future calorimeters in particle physics experiments [8].

In particular, we have studied the application of message passing networks to the task of calorimeter clustering, yielding initial promising results. The calorimeter clustering problem is very similar to the tracking problem except that there may be more than two true edges connected to an input node. We cast the task of calorimeter clustering as an operation on an initial static graph generated with a simple algorithm like k-Nearest-Neighbours (kNN), passing messages to generate features for classifying those edges as true or false. Here we are using kNN as stand-in for a lightweight reconstruction algorithm as a first pass to generate a graph on the data. The parameter k was chosen such that there was at least one true edge between all hits in the same truth-level cluster after applying the algorithm. Smaller k results in lower clustering efficiency, depending on the use of noise suppression k can be in the range of 8-24. In particular, these networks use the 'EdgeConv' operator defined in [18], and it was found that concatenating the intermediate hidden states in the output stage improved the rate of model convergence by about a factor of two compared to using no such shortcuts. A diagram of the GNN architecture used for calorimeter clustering is shown in figure 2.

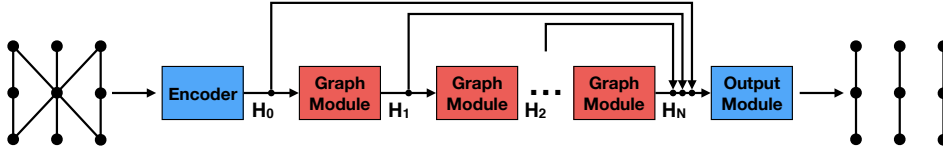


Figure 2: The Graph Neural Network architecture used for calorimeter clustering.

3 Results

The tracking results are based on the TrackML challenge data [17] generated by the ACTS framework [12]. This dataset simulates the very dense environment in the HL-LHC with 200 interactions per bunch crossing on average.

The GNN is trained on an NVIDIA V100 GPU for about 2 epochs in about two hours, resulting in the performance showed in figure 3. With a threshold of 0.5 on the GNN output, the edge efficiency, defined as the ratio of the number of true edges passing the threshold over the number of total true edges, reaches 95.9%, and the purity, defined as the ratio of the number of true edges passing the threshold over the number of total edges passing the threshold, is 95.7%. Guided by the GNN outputs, a simple algorithm is used to reconstruct track candidates. The algorithm makes iterative visits to all hits from inside to outside and reconstructs a best track candidate for the hit in question. Each hit is used only by one track so no ambiguity resolving is needed. This step is called ‘‘Connecting The Dots’’ (CTD). Using the GNN and CTD together reconstructs about 95% of true tracks that can be reconstructed in the graph across the transverse momentum range from 100 MeV to 5 GeV beyond which lacks statistics.

Ongoing work in reconstructing tracks with GNNs includes extending the method to whole detector data and improving the performance of the CTD post-processing algorithm to recover lost efficiency.

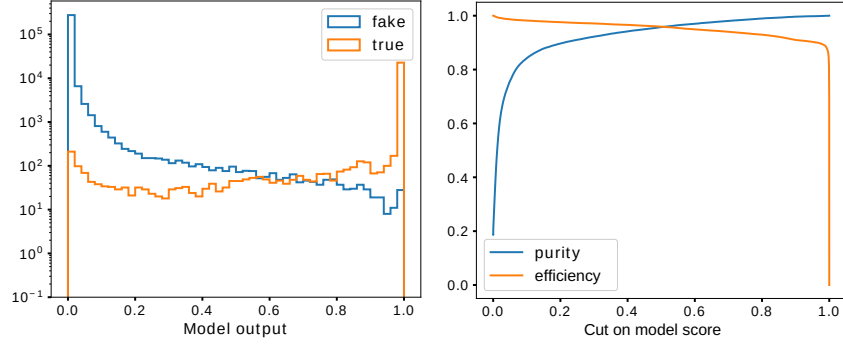


Figure 3: Training results of the Graph Neural Network. Left: The distribution of the edge scores (aka model output) predicted by GNN for true edges that connect the nodes that come from the same track, and for fake edges that do not. Right: The edge purity and efficiency as a function of different cuts on the model score. The definition of efficiency and purity can be found in the text.

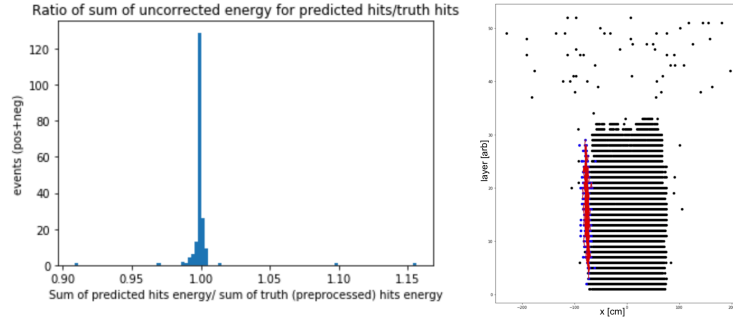


Figure 4: Left: The ratio, per event, for photons of total collected calorimeter energy deposits connected by predicted edges to the energy collected by the associations from ground truth. Right: The event display of a single photon showing the predicted edges (red) and underlying ground-truth nodes (blue) in addition to the energy deposits from noise (black).

In the context of calorimetry, we have achieved results separately for muon, photon, and pion energy deposits in the CMS High-Granularity Calorimeter (HGCAL). Each variety of particle deposits energy in the calorimeter in a qualitatively different way, with different expected fluctuations in their energy deposition patterns. Pion showers in particular are the most difficult since they exhibit large variability in their shower transverse profile as a function of the shower depth within the calorimeter. In each case, with examples for photon in figure 4 and pion in figure 5, we have observed excellent performance for correctly associating energy together using the predictions of these networks. For muons we found 99% efficiency with 90% purity, photons we are able to attain 99% efficiency and purity, and for pions we are able to attain better than 90% purity and efficiency. The purity of muons is driven by the large amount of noise hits and edges present in the training sample. All of these measurements are made in dedicated single particle samples for each type of particle, the next step of these tests are to move to variable multi-particle final states such as the decays of τ leptons and then multi-particle jets created in LHC physics events.

This indicates great potential for discovering GNN architectures which can scale to very large number of edges and that can handle multiple high energy particle physics reconstruction tasks. This, in turn, would allow computing centers for high energy physics to focus on certain types of acceleration and better determine where to spend resources and effort in order to become as efficient as possible.

Ongoing work for GNN applications in calorimetry includes studies on how to reconstruct multiple particle types simultaneously using new network architectures which can assign categories to edges. In addition, we are exploring how to better deal with overlapping showers and fractional assignment of calorimeter hit energy into cluster, both of which will be necessary to achieve the best performance

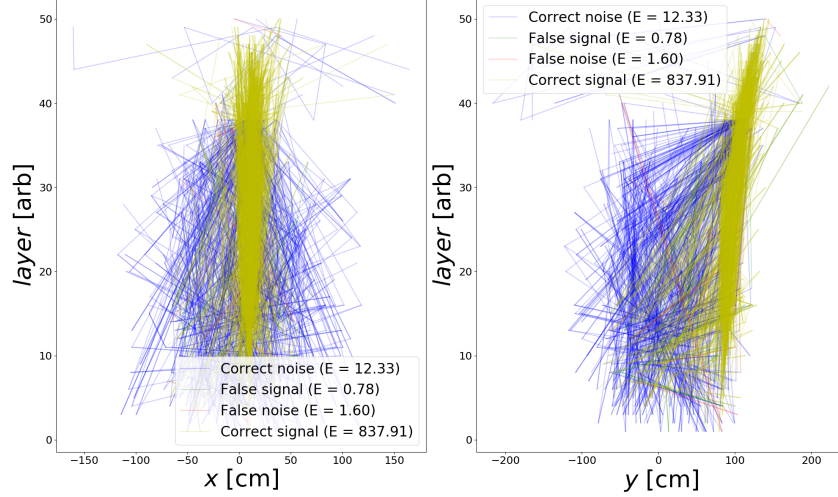


Figure 5: x and y projections of edge classification in a pion shower within the CMS HGCal. The input graph is derived using kNN. The vertical axis in each case is the calorimeter layer number. A score cut of 0.5 is used to identify true edges in this case and edges are labelled according to being true positives (yellow), true negatives (blue), false positives (green), and false negatives (red).

for the HGCal. Finally, explorations into deploying these networks for Liquid Argon Time Projection Chambers are in their initial stages.

4 Conclusion

We have demonstrated that Graph Neural Networks on Point Clouds are suitable for both tracking and calorimetry in high energy physics, having promising physics performance and good scalability. For the track finding problem, the GNNs combined with a simple connecting-the-dot algorithm results in a relative efficiency of over 95% for all particles. Ongoing work is recovering the inefficiency introduced by each selection. For the calorimeter clustering problem, we have found that very similar graph network architectures yield promising solutions. In the individual clustering problems used for testing so far we have found excellent energy collection efficiency, as well as efficiencies and purities better than 90% even in the most difficult scenarios. The next step will be to connect the dots as in the tracking algorithms and derive useful physics quantities from the collections of connected calorimeter energy deposits.

Acknowledgments

We are grateful to Javier Duarte, Phillip Harris, and Jim Hirschauer for the useful discussions. This research was supported in part by the Office of Science, Office of High Energy Physics, of the US Department of Energy under Contracts No. DE-AC02-05CH11231 and No. DE-AC02-07CH11359, FNAL LDRD 2019.017.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Part of this work was conducted at "iBanks", the AI GPU cluster at Caltech. We acknowledge NVIDIA, SuperMicro and the Kavli Foundation for their support of "iBanks".

L.G., T.K., K.P., and N.T. are partially supported by Fermilab LDRD L2019.017: "Graph Neural Networks for Accelerating Calorimetry and Event Reconstruction".

S.A. is supported by the Marie Skłodowska-Curie Innovative Training Network Fellowship of the European Commission's Horizon 2020 Programme under contract number 765710 INSIGHTS.

References

- [1] G. Aad et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.
- [2] G. Apollinari et al. “High Luminosity Large Hadron Collider HL-LHC”. In: *CERN Yellow Report* 5 (2015), pp. 1–19. DOI: 10.5170/CERN-2015-005.1. arXiv: 1705.08830 [physics.acc-ph].
- [3] J. Arjona Martínez et al. “Pileup mitigation at the Large Hadron Collider with graph neural networks”. In: *Eur. Phys. J. Plus* 134.7 (2019), p. 333. DOI: 10.1140/epjp/i2019-12710-3. arXiv: 1810.07988 [hep-ph].
- [4] Peter W. Battaglia et al. “Interaction Networks for Learning about Objects, Relations and Physics”. In: *CoRR* abs/1612.00222 (2016). arXiv: 1612.00222.
- [5] Michael M. Bronstein et al. “Geometric deep learning: going beyond Euclidean data”. In: *CoRR* abs/1611.08097 (2016). arXiv: 1611.08097.
- [6] S. Chatrchyan et al. “The CMS experiment at the CERN LHC”. In: *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.
- [7] N. Choma et al. “Graph Neural Networks for IceCube Signal Classification”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2018, pp. 386–391. DOI: 10.1109/ICMLA.2018.00064.
- [8] CMS Collaboration. *The Phase-2 Upgrade of the CMS Endcap Calorimeter*. Tech. rep. CERN-LHCC-2017-023. CMS-TDR-019. Technical Design Report of the endcap calorimeter for the Phase-2 upgrade of the CMS experiment, in view of the HL-LHC run. Geneva: CERN, Nov. 2017. URL: <https://cds.cern.ch/record/2293646>.
- [9] Lyndon Evans and Philip Bryant. “LHC Machine”. In: *JINST* 3 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.
- [10] Steven Farrell et al. “Novel deep learning methods for track reconstruction”. In: *4th International Workshop Connecting The Dots 2018 (CTD2018) Seattle, Washington, USA, March 20-22, 2018*. 2018. arXiv: 1810.06111 [hep-ex].
- [11] Steven Farrell et al. “The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking”. In: *EPJ Web Conf.* 150 (2017), p. 00003. DOI: 10.1051/epjconf/201715000003.
- [12] C Gumpert et al. “ACTS: from ATLAS software towards a common track reconstruction software”. In: *Journal of Physics: Conference Series* 898 (Oct. 2017), p. 042011. DOI: 10.1088/1742-6596/898/4/042011.
- [13] Isaac Henrion et al. “Neural message passing for jet physics”. In: (2017). URL: https://dl4physicsciences.github.io/files/nips_dlps_2017_29.pdf.
- [14] Shah Rukh Qasim et al. “Learning representations of irregular particle-detector geometry with distance-weighted graph networks”. In: *Eur. Phys. J. C* 79.7 (2019), p. 608. DOI: 10.1140/epjc/s10052-019-7113-9. arXiv: 1902.07987 [physics.data-an].
- [15] Huilin Qu and Loukas Gouskos. “ParticleNet: Jet Tagging via Particle Clouds”. In: (2019). arXiv: 1902.08570 [hep-ph].
- [16] F. Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (Jan. 2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.
- [17] *TrackML Particle Tracking Challenge*. URL: <https://www.kaggle.com/c/trackml-particle-identification>.
- [18] Yue Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: (2018). arXiv: 1801.07829.
- [19] Jie Zhou et al. “Graph Neural Networks: A Review of Methods and Applications”. In: *abs/1612.00222* (2019). arXiv: 1812.08434.