Snowmass2021 - Letter of Interest

Graph Data Structures and Graph Neural Networks for High Energy Physics

Thematic Areas: (check all that apply □/■)
□ (CompF1) Experimental Algorithm Parallelization
□ (CompF2) Theoretical Calculations and Simulation
■ (CompF3) Machine Learning
□ (CompF4) Storage and processing resource access (Facility and Infrastructure RD)
□ (CompF5) End user analysis
□ (CompF6) Quantum computing
□ (CompF7) Reinterpretation and long-term preservation of data and code

Contact Information:
Xiangyang Ju (Berkeley Lab) [xju@lbl.gov]

Additional Authors:
Mark Neubauer (University of Illinois at Urbana-Champaign) [msn@illinois.edu],
Paolo Calafiura (Berkeley Lab) [pcalafiura@lbl.gov],
Lindsey Gray (FermiLab) [lagray@fnal.gov],
Adam Aurisano (University of Cincinnati) [adam.aurisano@uc.edu],
Thiago R. F. P. Tomei (SPRACE-Unesp) [Thiago.Tomei@unesp.br],
Jean-Roch Vlimant (California Institute of Technology) [jvlimant@caltech.edu],
Giuseppe Cerati (FermiLab) [cerati@fnal.gov],
Jeremy Hewes (University of Cincinnati) [jhewes15@fnal.gov],
Kazuhiro Terao (SLAC National Accelerator Laboratory) [kterao@slac.stanford.edu],
Haichen Wang (Berkeley Lab and UC Berkeley) [haichenwang@lbl.gov],
Savannah Thais (Princeton University) [sthais@princeton.edu],
Daniel Murnane (Berkeley Lab) [dtmurnane@lbl.gov]

Abstract: This letter of intent will describe how graph data structures can be used to represent global and local relationships between physics objects, including the ones deriving from complex detector geometries, and briefly review current applications of geometric deep learning [1] to HEP event reconstruction/classification. Some promising research directions and their potential impact for HEP online and offline computing will be discussed.

Keywords: snowmass2021, computational frontier, particle physics, reconstruction, tracking, geometric learning, graph neural networks

1 Introduction

As HEP experiments look for rarer and rarer phenomena, and detectors become more and more segmented to cope with increased interaction rates, HEP datasets continue to grow in size and complexity. For example, at the HL-LHC experiments will record about ten times more events per second, and each event will capture the products of about five times more collisions. Unless new pattern recognition algorithms are developed, computational complexity, in particular of pattern recognition algorithms, may limit the experiments' physics reach.

Different data representations are used in machine learning applications to represent collision events, including physics-inspired feature vectors [2, 3], event images [4–6] and point clouds [7–9]. However, they have some inherent disadvantages in handling collision data. The physics-inspired vectors implicitly impose an ordering on the data presented to machine learning models and the ordering may reduce the learning performance. The event images face some limitations with irregular detector geometry or sparsity of collision data. Both physics-inspired vectors and event images could not naturally handle dynamic event sizes. The point clouds, which treat each collision event as a set of un-ordered physics objects, neglect the local relationships of the physics objects in the data.

We are proposing the graph data structure for representing collision events and the Geometric Deep Learning [1] for learning the properties of collision events. Our work in using graph neural network in High Energy Physics can be found in Ref [10–12].

2 Graph data structures

A graph contains a set of nodes, a set of edges with each connecting a pair of nodes, and a set of node-, edge- and graph-level attributes, collectively called graph attributes. Graph representation is flexible and dynamic because a graph can have arbitrary number of nodes and arbitrary connections between nodes, providing unlimited expressive power. A graph can be dense or sparse, depending on the relative number of edges comparing to the number of nodes. A graph can be an un-directed graph, where edges link two nodes symmetrically, or a directed graph, where edges link two nodes asymmetrically. The direction in which an edge points one node to another could represent the direction of energy flow or event thrust.

There are many ways in constructing graphs to represent collision events. Depending on the task in question, one can treat detector measurements as graph nodes and associate the measured values as node attributes. Edges can be pair-wise connections between the nodes, therefore, the number of edges will be at the level of $\mathrm{O}(N^2)$ where N is the number of nodes. Such graph is called fully-connected graph. The fully-connected graph captures all possible relations between the nodes, which may not needed for some tasks. Instead one can use clustering methods or metric learning methods to only connect the nodes that reside in a neighbourhood so as to reduce the number of edges. Metric learning is a method that transforms input node features into an embedding phase spaces where similar

nodes are close and away from dissimilar nodes. Using metric learning method to construct graphs is explored in Ref. [12].

3 Geometric Deep Learning

Geometric deep learning (GDL) generalizes convolutional and recurrent neural networks to datasets with arbitrary geometry and sparsity. GDL architectures learn local and global relations themselves, whether implicitly by spectrally or spatially convolving over combinations of nodes, or explicitly by attaching features to nodes and edges. Combining GDL with the encoding/decoding of node and edge features allows to learn non-trivial geometric relationships like those between all hits belonging to the same track.

4 Brief review of GNN applications in HEP

The application of GNN in classification problems can be grouped into three categories: graph-, node- and edge level classifications. An extensive review is done in Ref. [13]. In the following, some exemplary applications in each category are presented.

The objective of graph-level classification is to separate signal events from background events by training GNN to learning the graph-level labeling. Ref. [14] represented the events recorded by the IceCube detector as graphs and used the convolution graph neural network to classify events into positives (neutrino events) or negatives (background). Similarly, Ref [15] uses message passing neural network to search for events from the stop productions at the LHC.

The objective of node-level classification is to separate different node types, where each node represents a reconstructed particle. Ref [16] uses the interaction networks [17] to identify jets from all-hadronic decays of high-momentum heavy particles and distinguish them from ordinary jets originating from the hadronization of quarks and gluons. Ref [18] trained a Gated Graph Neural Network [19] to retain particles coming from high-transverse-momentum collisions, while rejecting those coming from additional inelastic proton-proton interactions (pile-up) at the LHC.

The objective of edge-level classification is to classify the particles coming from the same origin. In Ref [11], the interaction network was used in the tracking reconstruction to identify the edges that connect two space points coming from the same track, and in the calorimeter clustering problem to identify the edges that connect two energy clusters coming from the same particle decays.

5 Future research opportunities

This section describes some promising research opportunities for using GNN in HEP from the aspects of computing and physics.

Accelerating GNN pipelines A full pipeline for tracking [12] is achieved by first learning non-linear embeddings for hits such that hits from the same track are close to each other in the embedding spaces, and then constructing a graph from the embeddings for a

message passing graph neural network to classify edges between hits. Given a classified graph, simple clustering techniques such as DBSCAN are used to label hits belonging to contiguous collections of edges. Treating these labels as track numbers, this pipeline proves to be competitive on the tracking ML challenge [20]. Each of these stages can run in parallel within an event, and the computational complexity is less than quadratic with respect to the number of collisions per second, due to both the non-linear metric construction and the nature of sparse message passing. The pipeline can potentially extended by treating the graph construction and GNN as modular, which could be plugged into a regression network to derive track parameters and other global-level observable.

We recognize it is important to explore different computing hardware to accelerate the training and inference of GNN architectures. Current computing hardware, such as GPU, is optimized to process data structures as 1D or 2D dense arrays. Graphs are sparse and dynamical data structures that are difficult to handle efficiently in memory and to distribute for parallel processing. Finding a hardware accelerator that is optimized for sparse and irregular tensor computations could benefit the offline training and inference and more importantly potentially enable the application of GNNs for real-time data processing. Preliminary studies are carried out in Ref. [21].

Innovative GNN architectures Ref [12] founds significant gains in GNN performance when doublet graphs are contracted into triplet graphs. Useful information contained in two-, three- or four-hit combinations is often used in traditional tracking techniques, and we can pass this information to a GNN by constructing a "triplet graph" - wherein nodes represent pairs of hits, and edges represent triplet combinations. An edge classifier then predicts likely triplet connections, and these can be used as highly pure sets of seeds for conventional tracking techniques. Currently, the conversion of "doublet graph" to "triplet graph" is done statically, prior to training, and the resulting edges are classified in a conventional way. However, this conversion represents a generalisation from graph to "hypergraph" - where each edge can generally connect any combination of nodes. Hypergraph edge classification GNNs have recently been explored [22, 23], and we see the field heading towards a much more dynamic approach to edge construction and conversion [24–26]. For example, allowing the graph to change during training allows the GNN to learn when edges should be pruned, added or converted to hyperedges between combinations of nodes. In the inference stage, this could lead to memory savings and improved scaling with luminosity. Currently, this is limited by the memory available on GPUs while training, but near-term GPU generations would allow for dynamic (hyper)graph training of the scale of HL-LHC tracking data.

Beyond Particle Tracking Imaging calorimeter can also benefit from the use of graph neural networks. The LArTPC's millimetre-scale spatial resolution provides the opportunity for precise particle reconstruction and identification at DUNE, and the CMS High-Granularity Calorimeter provides an opportunity for unprecedented shower separation in HL-LHC. However, realising these opportunities using traditional reconstruction methods has proven to be challenging. Recently we demonstrated a hierarchical clustering techniques using GNNs applied to clustering of electromagnetic shower fragments, and then

particles into a neutrino interaction [27]. In this demonstration, the most granular information used was a detector hit with sparse convolutional layers as a feature extractor per fragment. Moving forward, we will study hierarchical event reconstruction using only GNNs, which builds up from low-level objects (hits) to high-level objects (interactions) by simplifying the clustering task using an embedding to predict clusters or dynamically pooling an input graph through intermediate stages.

We propose network architectures in which embedding or graph pooling is applied to a graph where each node represents a detector hit, in order to dynamically cluster the hits into a reduced graph where each node represents a particle cluster. Further graph reductions can then be applied – for instance, clustering together short $e^{\pm}$ tracks into an electromagnetic shower – to produce a graph in which each node represents a particle. By retaining the node features of individual hits inside each cluster, one can then perform convolutions to identify particle types for each node, and classify graph edges to reconstruct particle flow.

Jet clustering is traditionally an unsupervised learning task because there is no unique way to associate hadronic final states with the quark and gluon degrees of freedom that generated them. However, for uncolored particles like W, Z and Higgs bosons, it is possible to approximately (though not exactly) associate final state hadrons to their ancestor. By labelling simulated final state hadrons as descending from an uncolored particle, it is possible to train a supervised learning method to create boson jets. GNN are well-suited for this purpose as they can act on unordered sets and naturally create strong connections between particles with the same label. Such study opens a new exploration in jet physics to use machine learning to optimize the construction of jets. Preliminary study is carried out in Ref. [28].

Generative models for event generation In recent years, much progress has been made in developing generative models for graphs, among which the GraphRNN [29], Graph-Normalizing-Flows [30] and the Deep Generative Models for Graphs [31] are the most promising ones. They capture the structure and attributes of the given graphs and use it to generate high fidelity graphs. These models generate a graph sequentially starting from a root node, not unlike how an event generator simulates an event generation. In the future, it could be interesting to apply graph generative models to generate graphs that represent a particular physics process.

Data-sets, Data Challenges and Community-building We also note that the public data from the trackML Kaggle challenge [20] has facilitated collaboration on GNN-based tracking development from a large number of scientists and projects. Going forward, it will be critical for our community develop, publish and maintain realistic but detector-agnostic challenge data sets to foster innovation and multi-disciplinary development in important but challenging technical areas such as particle tracking.

References

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, et al., *Geometric deep learning: Going beyond euclidean data*, *IEEE Signal Processing Magazine* **34** (Jul, 2017) 18–42, [arXiv:1611.08097].

[2] T. Q. Nguyen, D. Weitekamp, D. Anderson, R. Castello, et al., *Topology classification with deep learning to improve real-time event selection at the LHC*, *Comput. Softw. Big Sci.* **3** (2019), no. 1 12, [arXiv:1807.00083].

[3] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, et al., *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev.* **D94** (2016), no. 11 112002, [arXiv:1607.08633].

[4] J. Lin, M. Freytsis, I. Moult, and B. Nachman, *Boosting $H \to b\bar{b}$ with Machine Learning*, *JHEP* **10** (2018) 101, [arXiv:1807.10768].

[5] **ATLAS Collaboration** Collaboration, *Convolutional Neural Networks with Event Images for Pileup Mitigation with the ATLAS Detector*, Tech. Rep. ATL-PHYS-PUB-2019-028, CERN, Geneva, Jul, 2019.

[6] 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*, arXiv:1807.11916.

[7] P. T. Komiske, E. M. Metodiev, and J. Thaler, *Energy Flow Networks: Deep Sets for Particle Jets*, *JHEP* **01** (2019) 121, [arXiv:1810.05165].

[8] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, *Phys. Rev. D* **101** (2020), no. 5 056019, [arXiv:1902.08570].

[9] J. Shlomi, S. Ganguly, E. Gross, K. Cranmer, et al., *Secondary Vertex Finding in Jets with Neural Networks*, arXiv:2008.02831.

[10] S. Farrell et al., *Novel deep learning methods for track reconstruction*, in *4th International Workshop Connecting The Dots 2018*, 10, 2018. arXiv:1810.06111.

[11] X. Ju et al., *Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors*, in *33rd Annual Conference on Neural Information Processing Systems*, 3, 2020. arXiv:2003.11603.

[12] N. Choma, D. Murnane, X. Ju, P. Calafiura, et al., *Track Seeding and Labelling with Embedded-space Graph Neural Networks*, arXiv:2007.00149.

[13] J. Shlomi, P. Battaglia, and J.-R. Vlimant, *Graph Neural Networks in Particle Physics*, arXiv:2007.13681.

[14] N. Choma, F. Monti, L. Gerhardt, T. Palczewski, et al., *Graph neural networks for icecube signal classification*, arXiv:1809.06166.

[15] M. Abdughani, J. Ren, L. Wu, and J. M. Yang, *Probing stop pair production at the LHC with graph neural networks*, *JHEP* **08** (2019) 055, [arXiv:1807.09088].

[16] E. A. Moreno, O. Cerri, J. M. Duarte, H. B. Newman, et al., *JEDI-net: a jet identification algorithm based on interaction networks*, *Eur. Phys. J. C* **80** (2020), no. 1 58, [arXiv:1908.05318].

[17] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, et al., *Interaction Networks for Learning about Objects, Relations and Physics*, arXiv:1612.00222.

[18] J. Arjona Martínez, O. Cerri, M. Pierini, M. Spiropulu, et al., *Pileup mitigation at the Large Hadron Collider with graph neural networks*, *Eur. Phys. J. Plus* **134** (2019), no. 7 333, [arXiv:1810.07988].

[19] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, *Gated graph sequence neural networks*, arXiv:1511.05493.

[20] The TrackML Kaggle challenge, "trackml." https://www.kaggle.com/c/trackml-particle-identification, 2018.

[21] Y. Iiyama et al., *Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics*, arXiv:2008.03601.

[22] R. Zhang, Y. Zou, and J. Ma, *Hyper-sagnn: a self-attention based graph neural network for hypergraphs*, in *International Conference on Learning Representations*, 2020.

[23] Y. Feng, H. You, Z. Zhang, R. Ji, et al., *Hypergraph neural networks*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3558–3565, 2019.

[24] S. R. Qasim, J. Kieseler, Y. Iiyama, and M. Pierini, *Learning representations of irregular particle-detector geometry with distance-weighted graph networks*, *The European Physical Journal C* **79** (2019), no. 7 1–11.

[25] E. Ranjan, S. Sanyal, and P. P. Talukdar, *Asap: Adaptive structure aware pooling for learning hierarchical graph representations.*, in *AAAI*, pp. 5470–5477, 2020.

[26] F. Diehl, *Edge contraction pooling for graph neural networks*, *arXiv preprint arXiv:1905.10990* (2019).

[27] F. Drielsma, Q. Lin, P. C. de Soux, L. Dominé, et al., *Clustering of electromagnetic showers and particle interactions with graph neural networks in liquid argon time projection chambers data*, 2020.

[28] X. Ju and B. Nachman, *Supervised Jet Clustering with Graph Neural Networks for Lorentz Boosted Bosons*, arXiv:2008.06064.

[29] J. You, R. Ying, X. Ren, W. L. Hamilton, et al., *GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models*, arXiv:1802.08773.

[30] J. Liu, A. Kumar, J. Ba, J. Kiros, et al., *Graph Normalizing Flows*, arXiv:1905.13177.

[31] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, et al., *Learning Deep Generative Models of Graphs*, arXiv:1803.03324.