Neural Message Passing for Jet Physics

Isaac Henrion, Johann Brehmer, Joan Bruna, Kyunghun Cho, Kyle Cranmer
Center for Data Science
New York University
New York, NY 10012
{henrion*, johann.brehmer, bruna, kyunghyun, kyle.cranmer*}@nyu.edu

Gilles Louppe
Department of Computer Science
University of Liège
Belgium
g.louppe@ulg.ac.be

Gaspar Rochette
Department of Computer Science
École Normale Supérieure
Paris, France
gaspar.rochette@ens.fr

Abstract

Supervised learning has incredible potential for particle physics, and one application that has received a great deal of attention involves collimated sprays of particles called jets. Recent progress for jet physics has leveraged machine learning techniques based on computer vision and natural language processing. In this work, we consider message passing on a graph where the nodes are the particles in a jet. We design variants of a message-passing neural network (MPNN); (1) with a learnable adjacency matrix, (2) with a learnable symmetric adjacency matrix, and (3) with a set2set aggregated hidden state and MPNN with an identity adjacency matrix. We compare these against the previously proposed recursive neural network with a fixed tree structure and show that the MPNN with a learnable adjacency matrix and two message-passing iterations outperforms all the others.

1 Introduction

Several physics goals for the Large Hadron Collider (LHC) are inextricably linked to the treatment of collimated sprays of energetic hadrons referred to as 'jets'. There are a number of tasks encountered in jet physics including classification and regression associated to the progenitor particle(s) giving rise to the jet. For instance, a jet may result from a quark, gluon, W-boson, top-quark, or Higgs boson. Several Beyond the Standard Model (BSM) theories involve new particles and interactions that predict specific jet signatures, but testing these theories is challenging because jets from more mundane processes occur *much* more frequently. Often sensitivity to these BSM theories requires classifiers with true positive rates of $\mathcal{O}(1)$ and false positive rates of $\mathcal{O}(10^{-2})$. There has been an enormous amount of effort from both the theoretical and experimental communities to develop techniques for jet physics [1].

Recent progress in applying machine learning techniques for jet physics has been built upon an analogy between calorimeters and images [2–9]. These methods take a variable-length set of 4-momenta and project them into a fixed grid of $\eta - \phi$ towers or 'pixels' to produce a 'jet image'.

More recently, recursive neural networks have been applied to this classification problem based on an analogy between QCD and natural languages [10]. Much like a sentence is composed of words following a syntactic structure organized as a parse tree, a jet is also composed of particles following

*Corresponding authors

Workshop on Deep Learning for Physical Sciences (DLPS 2017), NIPS 2017, Long Beach, CA, USA.

a structure dictated by quantum chromodynamics (QCD) and organized via the clustering history of a sequential recombination jet algorithm. This work showed that the projection of particles' momenta into images loses information, which impacts classification performance. The recursive networks were able to avoid this pre-processing and provide superior performance.

In this work we represent the jet as a graph and consider Message Passing Neural Networks (MPNN) [11] on the same benchmark data and binary classification task from [7, 10].

1.1 Jets as a graph

In the graph picture, nodes of the graph correspond to the particles and the data on the nodes are the features calculated from the 4-momenta of those particles: p, η, θ, ϕ, E, and p_T. The graph picture is natural in jet physics and was implicit in previous work with the recursive network operating over a binary tree. The binary tree was created from a sequential recombination jet algorithm [12, 13] that recursively combines the pair i, j that minimize

$$d_{ij}^{\alpha} = \min(p_{ti}^{2\alpha}, p_{tj}^{2\alpha}) \frac{\Delta R_{ij}^2}{R^2} \ . \tag{1}$$

The d_{ij}^{α} provides a family of adjacency matrices motivated by physics considerations; however, we will consider various approaches to adjacency matrix A_{ij}, including directed graphs.

The authors of Ref. [10] pointed out that it is compelling to think of generalizations of their technique in which the optimization would include the binary tree used for the embedding as a learnable component instead of considering it fixed a priori. An immediate challenge of this approach is that a discontinuous change in the binary tree (e.g., from varying α or R) makes the loss non-differentiable. However, the graph defined by d_{ij}^{α} evolves continuously with R and α. The authors of Ref. [10] pointed out that this makes graph-convolutional networks a natural approach [14–19]. In future work we plan to compare QCD-motivated adjacency matrix A_{ij} based on d_{ij}^{α}; however, in this work we attempt to learn the adjacency matrix directly.

2 Message Passing Neural Networks

This section describes a family of neural architectures defined over input sets exhibiting some geometric structure. We consider a dataset $\{\mathbf{x}_e, y_e\}_{e \leq N}$ consisting of observations $\mathbf{x}_e = \{x_i\}_{i < N_e}$, $x_i \in \mathbb{R}^S$, and labels $y_e \in \mathcal{Y}$. Each observation thus consists of a set measurements (that we assume Euclidean for simplicity), possibly of varying size. We are interested in neural network models that operate on such input sets, and with the ability to leverage the geometric structure determined by the measurements. If the measurements x_i are related with a known similarity structure $K(x_i, x_j)$, the appropriate data structure to represent the input is a (input-varying) graph $G_e = (V_e, E_e)$, where nodes are associated with measurements and weighted edges with their similarity.

Graph neural networks (GNN), introduced in [20, 21] and further simplified in [16, 22, 23], based on local operators of a graph $G = (V, E)$ offer a powerful balance between expressivity and sample complexity; see [24] for a recent survey on models and applications of deep learning on graphs. In its simplest incarnation, given an input signal $\mathbf{x} \in \mathbb{R}^{V \times S}$ on the vertices of a weighted graph G, we consider a family $\mathcal{A}$ of graph-intrinsic linear operators that act locally on this signal. The simplest example is to consider $\mathcal{A} = \{\mathbf{1}, A\}$, where A is the graph adjacency matrix. A GNN layer receives as input a signal $\mathbf{h}^{(t)} \in \mathbb{R}^{V \times d_t}$ and produces $\mathbf{h}^{(t+1)} \in \mathbb{R}^{V \times d_{t+1}}$ as

$$\mathbf{h}^{(t+1)} = \mathrm{Gc}(\mathbf{h}^{(t)}) = \rho\left(\sum_{q=1}^{|\mathcal{A}|} A_q \mathbf{h}^{(t)} \theta_q^{(t)}\right) , \tag{GNN NodeUpdate}$$

where $\theta_q^{(t)} \in \mathbb{R}^{d_t \times d_{t+1}}$, are trainable parameters, d_t is the number of feature maps at layer t, and $\rho(\cdot)$ is a point-wise non-linearity. One can easily verify that (GNN NodeUpdate) defines a model that is permutation-equivariant and that effectively shares the parameters across all the node locations, yielding gains in sample complexity similar to convolutional neural networks in computer vision.

Authors have explored several modeling variants from this basic formulation, by replacing the point-wise nonlinearity with gating operations [16, 22], or by generalizing the generator family to Laplacian polynomials [25, 19, 14], or including powers of A to encode multiple-hop neighborhoods of each

node [26]. Cascaded operations in the form (GNN NODEUPDATE) are able to approximate a wide range of graph inference tasks. Inspired by message-passing algorithms, [27–29] generalized GNNs to also learn edge features $A_{i,j}^{(t)}$ from the current node hidden representation, leading to the so-called *Message Passing Neural Network* (MPNN) model. Adjacency learning is particularly important in applications where the input set is believed to have some geometric structure, but the metric is not known a priori. In the case of jet physics, the d_{ij}^{α} provide a well-motivated family of adjacency matrices, but in what follows we explore learning the adjacency directly. In particular, we consider the family of adjacency matrices given by

$$A_{i,j}^{(t)} = \text{softmax}_{\text{row}} \varphi(h_i^{(t)}, h_j^{(t)}) \,, \qquad \text{(ADJACENCYMATRIX)}$$

where φ is a symmetric function parametrized with, e.g., a neural network, and the softmax over rows turns the kernel into a transition matrix. In this work, we choose for simplicity $\varphi(h, h') = v^\top (h + h') + b$.

In our experiments, we call this model MPNN-directed. We also experiment with undirected graph models by symmetrizing A via the transformation $S(A) = \frac{1}{2}(A + A^\top)$, and use the identity matrix $A = \mathbf{1}$ as a baseline. The resulting update rules for node features is obtained from (GNN NODEUPDATE) by using the edge feature kernel $A^{(t)}$ and modifying the node-wise nonlinearity with a more powerful gated recurrent unit (GRU, [30]):

$$m_i^{(t)} = \tanh\left(\sum_j A_{i,j}^{(t)} h_j^{(t)}\right), \qquad \text{(MESSAGE)}$$

$$h_i^{(t+1)} = \text{GRU}(h_i^{(t)}, [m_i^{(t)}, x_i]) \,, \qquad \text{(MPNN VERTEXUPDATE)}$$

where $[*, *]$ denotes concatenation of vectors. In this model, the hidden dimension d_t is kept constant across timesteps at 40. We also considered a `set` variant in which the current mean hidden state across all nodes $\hat{h}$ is passed as additional input to the GRU. These updates induce an order-invariance amongst the hidden states, as in the set2set model [31]. Finally, the collection of hidden states is collapsed to a single hidden state representing the whole graph, using another neural network (READOUT). A simple particular case of MPNN considered in [27] (RelNet) consists in only one message passing iteration that aggregates the adjacency kernel to directly predict the output label:

$$p(\hat{y} \mid \mathbf{x}) = f_\theta\left(\sum_{i,j} A_{i,j}\right) \,. \qquad \text{(RELNET)}$$

In general graphs, the network depth is chosen to be of the order of the graph diameter, so that all nodes obtain information from the entire graph. In our context, however, since the graph is densely connected, the depth is interpreted simply as giving the model more expressive power. The resulting model is summarized in Algorithm 1. Note that each iteration of neural message passing has its own parametrized functions ADJACENCYMATRIX, MESSAGE and VERTEXUPDATE.

Algorithm 1 Message passing neural network

Require: $N \times S$ array of jet constituents $\mathbf{x}$
▷ N is the number of particles, S is their data dimension

```
h ← tanh(W_e x + b_e)                          ▷ Embed the jets
for t = 1, ..., T do                           ▷ Message passing
    A ← AdjacencyMatrix_t(h)
    m ← Message_t(A, h)
    h ← VertexUpdate_t(h, m, x)
end for
return Readout(h)
```

3 Results

We consider the same benchmark data and binary classification problem as in Ref. [10]. The first class, which we denote 'QCD jets', arises from a known mixture of quarks and gluons. The second

Table 1: Summary of classification performance for several approaches.

Network	Iterations	ROC AUC	$R_{\epsilon=50\%}$
RecNN-k_t (without gating) [10]	1	0.9185 ± 0.0006	68.3 ± 1.8
RecNN-k_t (with gating) [10]	1	0.9195 ± 0.0009	74.3 ± 2.4
RecNN-desc-p_T (without gating) [10]	1	0.9189 ± 0.0009	70.4 ± 3.6
RecNN-desc-p_T (with gating) [10]	1	0.9212 ± 0.0005	$\mathbf{83.3 \pm 3.1}$
RelNet	1	0.9161 ± 0.0029	67.69 ± 6.80
MPNN (directed)	1	0.9196 ± 0.0015	89.35 ± 3.54
MPNN (directed)	2	0.9223 ± 0.0008	$\mathbf{98.26 \pm 4.28}$
MPNN (directed)	3	0.9188 ± 0.0031	85.93 ± 8.50
MPNN (undirected)	1	0.9193 ± 0.0015	86.41 ± 3.80
MPNN (undirected)	2	0.8949 ± 0.1004	$\mathbf{97.27 \pm 5.02}$
MPNN (undirected)	3	0.9185 ± 0.0036	84.53 ± 8.64
MPNN (set, directed)	1	0.9189 ± 0.0017	88.23 ± 4.53
MPNN (set, directed)	2	0.9191 ± 0.0046	87.46 ± 14.14
MPNN (set, directed)	3	0.9176 ± 0.0049	$\mathbf{88.33 \pm 9.84}$
MPNN (set, undirected)	1	0.9196 ± 0.0014	85.65 ± 4.48
MPNN (set, undirected)	2	0.9220 ± 0.0007	$\mathbf{94.70 \pm 2.95}$
MPNN (set, undirected)	3	0.9158 ± 0.0054	75.94 ± 12.54
MPNN (id)	1	0.9169 ± 0.0013	74.75 ± 2.65
MPNN (id)	2	0.9162 ± 0.0020	74.41 ± 3.50
MPNN (id)	3	0.9158 ± 0.0029	74.51 ± 5.20

class, which we denote 'W jets', arises from W bosons decaying into two quarks leading a single "fat jet" with characteristic substructure. Specifically, we use particle-level input used in Ref. [10] and compare with the results using the best performing RNN based on a simple descending p_T ordering and the binary tree defiend by the k_t jet algorithm ($\alpha = 1$). We use background rejection (i.e., 1/FPR) at 50% signal efficiency, which we denote $R_{\epsilon=50\%}$, for early stopping. For each model architecture considered, we train models with different initialization and follow the same prescription as Ref. [10] to provide a robust estimate of the mean and standard deviation by excluding outliers. We note the standard error on the mean is roughly five times smaller than the standard deviation.

Table 1 compares the results of various approaches using the same test data as Ref. [10]. The MPNN with a learned adjacency matrix and two iterations of message passing achieves the best performance in terms of both ROC AUC and $R_{\epsilon=50\%}$. The directed graph slightly outperforms the undirected graph, though not significantly. The learned adjacency matrix outperforms the *identity*, confirming the fact that pairwise particle interactions need to be taken into account. Our experiments indicate that adding message passing iterations does not monotonically increase the performance. We attribute this fact to the learning instability, evidenced by the increased variance, caused by the increased number of parameters, suggesting that better regularization techniques may be necessary in the future to stabilize learning and further improve the performance. We also notice that the set variants generally underperform. Although more in-depth analysis is necessary to make any firm conclusion, currently the aggregated hidden state seems to act more as noise than useful signal in the MPNN iteration.

4 Conclusions

With these initial results we conclude that the MPNNs are a powerful model for jet physics. Similar to recursive neural networks, they can operate on a variable number of particles and do not require any discretization into a fixed-length input or image-like pre-processing. In addition, the graph representation allows for information between all particles to be exchanged, where such communication is restricted to a tree structure in the recursive approach. We have observed that the model configuration influences the final result, thus care must be taken when designing the MPNN.

Our results motivate future work with a QCD-motivated adjacency matrix A_{ij} based on d_{ij}^{α}. Similarly, the graph picture enables a comparison of traditional jet clustering algorithms based on d_{ij}^{α} to data-driven clustering and community detection presented in [26]. Furthermore, the MPNN must be extended with a multi-scale coarsening scheme in order to avoid the expensive quadratic complexity of the presented fully-connected, single-scale version, in order to process a larger-scale input.

Acknowledgments

Cranmer and Louppe are supported through NSF ACI-1450310 and PHY-1505463. Bruna is supported through DOA W911NF-17-1-0438 and Intel NERSC-BDD. Cho thanks support by eBay, TenCent, Facebook, Google and NVIDIA, and is a CIFAR Azrieli Global Scholar. The authors are grateful for the support of the Moore-Sloan Data Science Environment at NYU.

References

[1] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning. 2017, 1709.04464.

[2] Josh Cogan, Michael Kagan, Emanuel Strauss, and Ariel Schwartzman. Jet-Images: Computer Vision Inspired Techniques for Jet Tagging. *JHEP*, 02:118, 2015, 1407.5675.

[3] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman. Jet-Images – Deep Learning Edition. 2015, 1511.05190.

[4] Leandro G. Almeida, Mihailo Backović, Mathieu Cliche, Seung J. Lee, and Maxim Perelstein. Playing Tag with ANN: Boosted Top Identification with Pattern Recognition. *JHEP*, 07:086, 2015, 1501.05968.

[5] Pierre Baldi, Kevin Bauer, Clara Eng, Peter Sadowski, and Daniel Whiteson. Jet Substructure Classification in High-Energy Physics with Deep Neural Networks. 2016, 1603.09349.

[6] Daniel Guest, Julian Collado, Pierre Baldi, Shih-Chieh Hsu, Gregor Urban, and Daniel Whiteson. Jet Flavor Classification in High-Energy Physics with Deep Neural Networks. *Phys. Rev.*, D94(11):112002, 2016, 1607.08633.

[7] James Barnard, Edmund Noel Dawe, Matthew J. Dolan, and Nina Rajcic. Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks. 2016, 1609.00607.

[8] Patrick T. Komiske, Eric M. Metodiev, and Matthew D. Schwartz. Deep learning in color: towards automated quark/gluon jet discrimination. *JHEP*, 01:110, 2017, 1612.01551.

[9] Gregor Kasieczka, Tilman Plehn, Michael Russell, and Torben Schell. Deep-learning Top Taggers or The End of QCD? 2017, 1701.08784.

[10] Gilles Louppe, Kyunghyun Cho, Cyril Becot, and Kyle Cranmer. QCD-Aware Recursive Neural Networks for Jet Physics. 2017, 1702.00748.

[11] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017, 1704.01212.

[12] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The Anti-k(t) jet clustering algorithm. *JHEP*, 04:063, 2008, 0802.1189.

[13] Gavin P. Salam. Towards Jetography. *Eur. Phys. J.*, C67:637–686, 2010, 0906.1833.

[14] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[15] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[16] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. *CoRR*, abs/1511.05493, 2015.

[17] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. *CoRR*, abs/1605.05273, 2016.

[18] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375, 2016.

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[20] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proc. IJCNN*, 2005.

[21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[22] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Neural Information Processing Systems*, 2015.

[23] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pages 2244–2252, 2016.

[24] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *CoRR*, abs/1611.08097, 2016.

[25] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016.

[26] Joan Bruna and Xiang Li. Community detection with graph neural networks. *arXiv preprint arXiv:1705.08415*, 2017.

[27] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.

[28] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[29] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[30] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[31] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.