

Particle Graph Autoencoders

Author(s): Steven Tsan, Javier Duarte

University of California San Diego, La Jolla, CA 92130, USA

Jean-Roch Vlimant

California Institute of Technology, Pasadena, CA 91125, USA

Maurizio Pierini

European Organization for Nuclear Research (CERN), CH-1211, Geneva 23, Switzerland

0.1 Method

We propose particle graph autoencoders (PGAEs) based on graph neural networks [1] for unsupervised detection of new physics in multijet final states at the LHC. By embedding particle jet showers as a graph, GNNs are able to exploit particle-particle relationships to efficiently encode and reconstruct particle-level information within jets. We posit that this can improve the capacity of autoencoders to learn a compressed representation of a jet and consequently help identify anomalous beyond-the-standard-model (BSM) multijet signal events from LHC data.

In our PGAE model, we represent each input jet as a graph in which each particle of the jet is a node, and each node has an edge connecting it to every other particle in the jet (i.e. a fully-connected particle graph). When encoding and decoding, the graph structure of the data remains the same, but the nodes' features, initially the particle's four-momentum (E, p_x, p_y, p_z) , have their dimensionality reduced during the encoding phase. We note the model can be expanded to consider additional particle-level information, such as particle type, electromagnetic charge, and pileup probability weight [2]. For the encoder and decoder, we use the edge convolution layer from Ref. [3], which performs message passing along the edges and aggregation of messages at the nodes of the graphs. A schematic of this is shown in Fig. 1.

The PGAE model is constructed using the PyTorch Geometric library [4]. In this model, the input node features are first processed by a batch normalization layer [5]. The encoder is an edge convolution layer [3], built from a fully connected neural network ϕ_e with layers of sizes $(32, 32, 2)$ and rectified linear activation unit (ReLU) activation functions [6]. The network takes in an 8-dimensional input, which is given by $(\mathbf{p}_i, \mathbf{p}_j - \mathbf{p}_i)$, where \mathbf{p}_i (\mathbf{p}_j) is the four-momentum for particle i (j) and $i \neq j$. The final layer produces a two-dimensional message vector from each pair of distinct particles. These two-dimensional message vectors are aggregated (using a mean function) for each receiving particle

$$\mathbf{h}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \phi_e(\mathbf{p}_i, \mathbf{p}_j - \mathbf{p}_i), \quad (0.1)$$

where $\mathcal{N}(i)$ is the neighborhood of particles connected to the i th particle, which corresponds to all other particles in this case. This summed message \vec{h}_i is the bottleneck or encoded representation for the i th particle. The decoder is also an edge convolution layer, containing

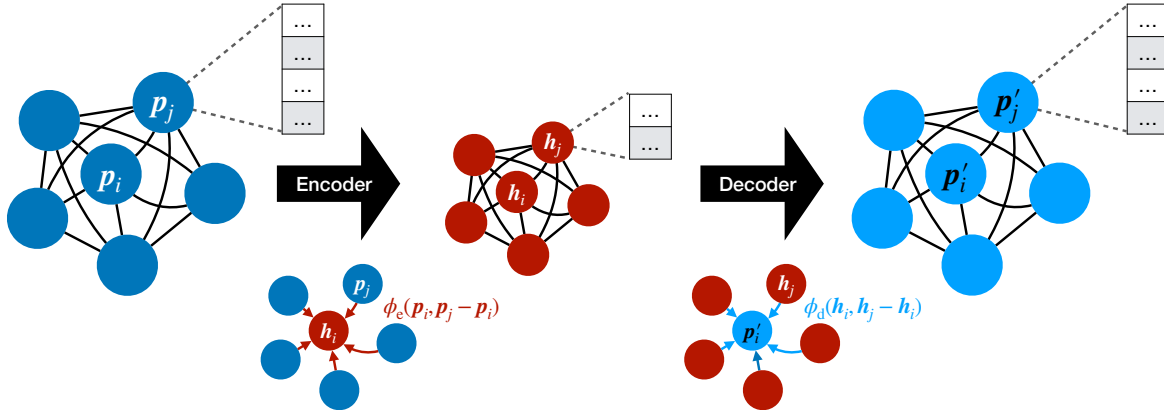


Figure 1: Schematic of the particle graph autoencoder model proposed. Each input jet is represented as a graph in which each particle of the jet is a node, and each node has an edge connecting it to every other particle in the jet. After an edge convolution layer [3], each particle is encoded in a reduced two-dimensional latent space, before another edge convolution layer reconstructs each particle’s four-momentum (E, p_x, p_y, p_z) .

a network ϕ_d with layers of sizes $(32, 32, 4)$ and ReLU activation functions, except for the final layer. The input is a 4-dimensional vector representing $(\mathbf{h}_i, \mathbf{h}_j - \mathbf{h}_i)$ and the output is intended to reconstruct each particle’s momentum. We note that the architecture itself is insensitive to the ordering and number of input particles. PyTorch Geometric supports variable-size input graphs so there is no need for zero-padding.

The model is trained on the QCD background dataset with two different loss functions. The first is the mean squared error (MSE) between the input and output particles. This choice of loss function violates the permutation invariance of the algorithm because the particles must be reconstructed in the same order as they are input to achieve a small value of the loss function. For this reason, we also investigate a second, alternative loss function, the Chamfer distance loss, whose value does not depend on either the order of the input particles or the reconstructed particles [7–9]. Given two input sets of particles \mathcal{M} and \mathcal{N} , expressed in terms of the momentum vectors \mathbf{p}_i and \mathbf{p}_j (with $i \in \mathcal{M}$ and $j \in \mathcal{N}$), the loss function is defined as

$$D^{\text{NN}}(\mathcal{M}, \mathcal{N}) = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \min_{j \in \mathcal{N}} (\|\mathbf{p}_i - \mathbf{p}_j\|)^2 + \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} \min_{i \in \mathcal{M}} (\|\mathbf{p}_i - \mathbf{p}_j\|)^2, \quad (0.2)$$

where $\|\mathbf{p}_i - \mathbf{p}_j\|$ is the Euclidean distance.

0.2 Results on LHC Olympics

First, we studied our algorithm on the R&D dataset. As the truth information is provided, we can create a receiver operating characteristic (ROC) curve to determine the effectiveness of the PGAE to identify a signal ($W' \rightarrow XY$, $X \rightarrow qq$, and $Y \rightarrow qq$ with $m_{W'} = 3.5$ TeV, $m_X = 500$ GeV, and $m_Y = 100$ GeV) that it did not observe during training. The ROC

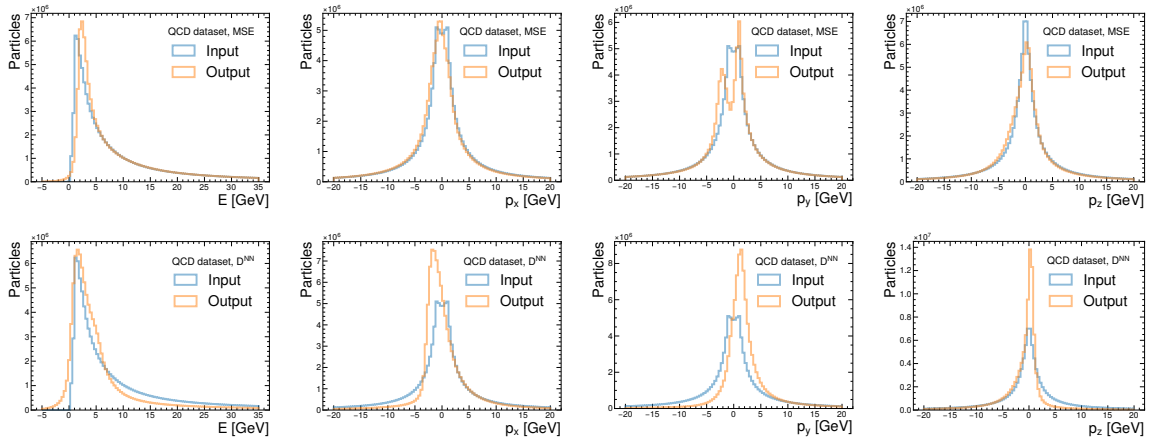


Figure 2: Comparison of input and reconstructed features E (far left), p_x (center left), p_y (center right), and p_z (far right) for the models trained with MSE (top) and Chamfer (bottom) loss functions on the QCD testing dataset.

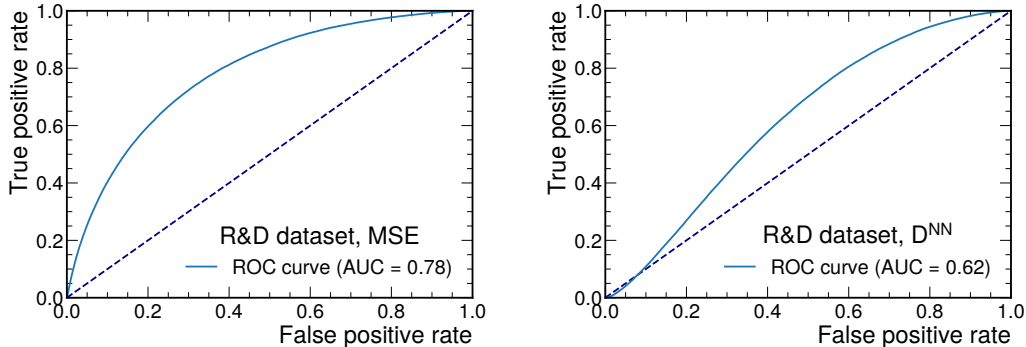


Figure 3: ROC curves for the PGAE trained with the MSE (left) and Chamfer loss (right).

curves for both the MSE and Chamfer loss functions are shown in Fig. 3. Although the MSE loss is not permutation invariant, we find it provides better discrimination for a new unseen signal.

To evaluate our model’s performance for anomaly detection, we perform a resonance search (or “bump hunt”) in the dijet invariant mass m_{jj} , computed from the two jets with highest p_T in the event. We perform this dijet search in black box (BB) 1, which contains a resonant dijet signal at $m_{jj} \sim 3.8$ TeV, and BB 2, which contains no signal. We require both of the jets to be “outliers,” which we define as jets with a reconstruction loss exceeding a threshold corresponding to the 90% quantile of the loss distribution for the leading two jets in the corresponding evaluation dataset. We note that because our algorithm is jet-focused, it is straightforward to generalize this search to multijet events.

For the background prediction in the signal-enriched outlier region, we perform a simpli-

fied analysis using the shape of the data in the background-enriched nonoutlier region. Specifically, we fit the ratio of the nonoutlier-to-outlier dijet mass distribution with a fourth-order polynomial to derive a transfer factor (TF). We take nonoutlier data distribution weighted by the TF as an estimate of the expected background in the outlier region. We do not consider systematic uncertainties associated to the TF although these could be taken into account in a more complete analysis in the future. The procedure is illustrated in Fig. 4 for BB 2.

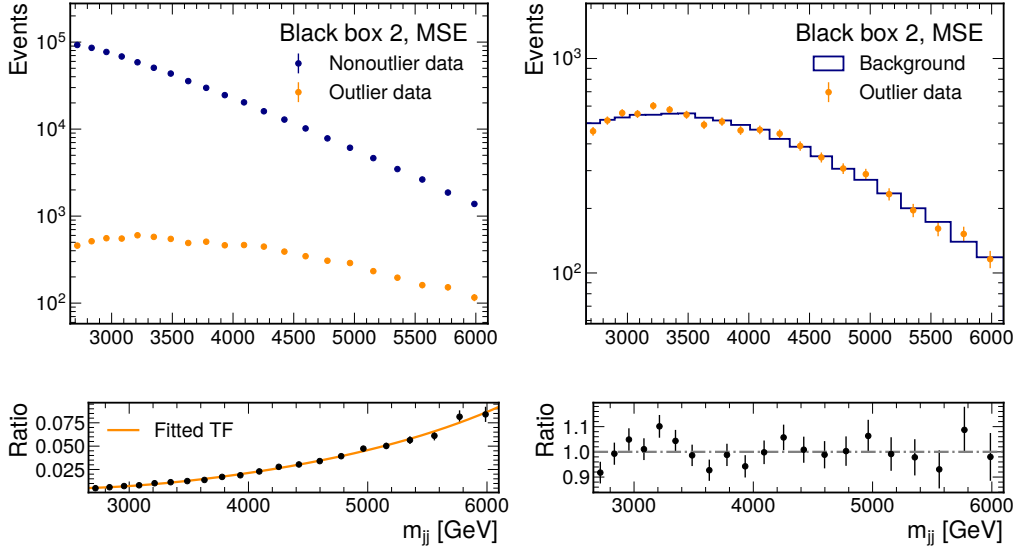


Figure 4: Illustration of the simplified background estimation procedure in BB 2 for the GAE trained with MSE loss. A comparison between the nonoutlier and outlier jet mass distribution is shown (upper left). The ratio of the two distributions is fit with a fourth-order polynomial to derive a transfer factor (lower left). The corresponding postfit prediction is also shown (upper right). The postfit ratio is randomly scattered around one as expected for BB 2, which contains no signal.

To derive the observed significance with the simplified background prediction, we use the bump hunter (BH) algorithm [10], recently implemented in Python [11]. We choose the variable-width mass binning from the CMS dijet searches [12] in the range from 2659 GeV to 6099 GeV. We look for resonances in windows spanning two to five bins. With the MSE model in BB 1, we identify a possible resonance around 3.9 TeV with a local significance of 2.1σ , which is close to the region of the injected dijet resonance with $m'_Z = 3823$ GeV. In BB 2 using the same model, the most discernable bump lies around 3.3 TeV with a small local significance of 0.8σ , which agrees with the fact that BB 2 has no injected signal. For the model trained with the Chamfer loss, a 1.5σ excess is seen at 2.8 TeV in BB 1 and a -1.4σ excess at 5.1 TeV in BB 2. Neither is significant. As noted previously, the permutation invariant Chamfer loss performs worse at the unsupervised anomaly detection task. This may

be due to the minimization, which will often return a smaller loss value than MSE even for poorly reconstructed, anomalous jets. Fig. 5 shows the BH results for BBs 1 and 2 using the models trained with both losses.

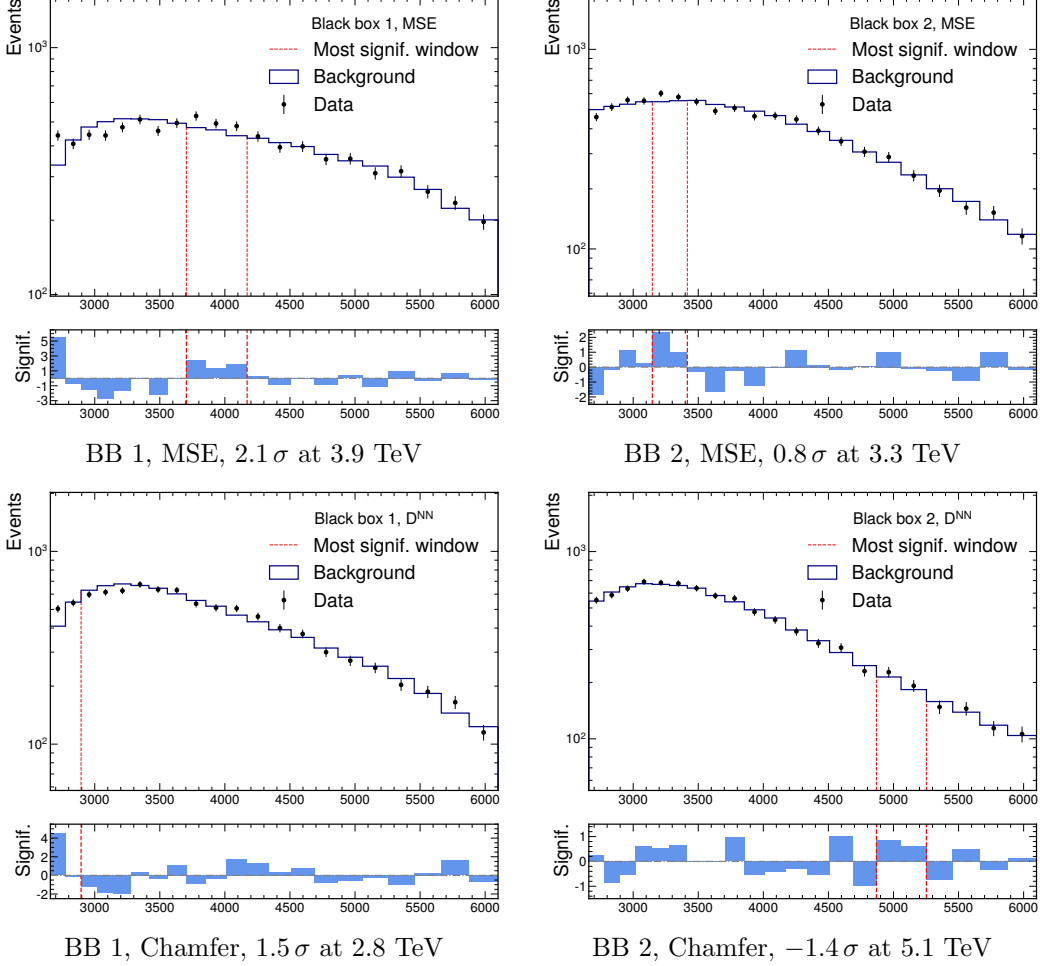


Figure 5: Bump hunt in the dijet invariant mass in BB 1 (left) and 2 (right) using MSE (top) and Chamfer (bottom) as the loss functions. Outlier jets have a reconstruction loss in the top 10% with respect to the corresponding BB. Outlier events are required to have both jets be outliers. BB 1 has an anomalous large-radius dijet signal $Z' \rightarrow XY \rightarrow (qq)(qq)$ injected at $m'_Z = 3823$ GeV (with $m_X = 732$ GeV and $m_Y = 378$ GeV), while BB 2 has no injected anomalies.

0.3 Lessons Learned

Graph neural networks, like our proposed particle graph autoencoder, are promising methods for anomaly detection. However, further work is needed to define a permutation-invariant loss function for use with such architectures that is more performant for anomaly detection. In

addition, a more generic resonance search procedure, such a multidimensional fit in the trijet, dijet, trijet, and single-jet mass distributions possibly using methods like Gaussian process fitting [13], would be appropriate to use in combination with this algorithm. In our experience, the R&D dataset was extremely helpful in preparing our anomaly detection algorithms and gauging whether the algorithm we were developing was on the right track. In the future, more extensive R&D datasets, together with additional black boxes with different signals, may be useful. Finally, it may be productive to host a future competition on a well-known platform, such as Kaggle, to increase engagement with the broader machine learning community.

0.4 Code Availability

All code is publicly available at <https://github.com/stsan9/AnomalyDetection4Jets>.

Acknowledgments

We thank the University of California San Diego Triton Research and Experiential Learning Scholars (TRELS) program for supporting this research. We also thank Joosep Pata for helpful discussions. J. D. is supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187. M. P. is supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 772369). J.-R. V. is partially supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 772369) and by the U.S. DOE, Office of Science, Office of High Energy Physics under Award No. DE-SC0011925, DE-SC0019227, and DE-AC02-07CH11359. This work was supported in part by U.S. National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the University of California San Diego’s California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100 Gpbs networks.

References

- [1] J. Shlomi, P. Battaglia, and J.-R. Vlimant, *Graph Neural Networks in Particle Physics*, [arXiv:2007.13681](https://arxiv.org/abs/2007.13681). Submitted to *Mach. Learn.: Sci. Tech*.
- [2] D. Bertolini, P. Harris, M. Low, and N. Tran, *Pileup per particle identification*, *J. High Energy Phys.* **10** (2014) 059, [[arXiv:1407.6013](https://arxiv.org/abs/1407.6013)].
- [3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, *Dynamic graph CNN for learning on point clouds*, *ACM Trans. Graph.* **38** (2019) [[arXiv:1801.07829](https://arxiv.org/abs/1801.07829)].
- [4] M. Fey and J. E. Lenssen, *Fast graph representation learning with PyTorch Geometric*, in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. [arXiv:1903.02428](https://arxiv.org/abs/1903.02428).

- [5] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating deep network training by reducing internal covariate shift*, in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37, (Lille), p. 448, PMLR, 2015. [arXiv:1502.03167](#).
- [6] A. F. Agarap, *Deep learning using rectified linear units (ReLU)*, [arXiv:1803.08375](#).
- [7] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, *Parametric correspondence and Chamfer matching: Two new techniques for image matching*, in *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, (San Francisco, CA, USA), p. 659, Morgan Kaufmann Publishers Inc., 1977.
- [8] H. Fan, H. Su, and L. J. Guibas, *A point set generation network for 3D object reconstruction from a single image*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 2463, 6, 2017. [arXiv:1612.00603](#).
- [9] Y. Zhang, J. Hare, and A. Prügel-Bennett, *FSPool: Learning set representations with featurewise sort pooling*, in *8th International Conference on Learning Representations*, 2020. [arXiv:1906.02795](#).
- [10] G. Choudalakis, *On hypothesis testing, trials factor, hypertests and the BumpHunter*, in *PHYSTAT 2011*, 2011. [arXiv:1101.0390](#).
- [11] L. Vaslin, J. Donini, and H. Schreiner, “pyBumpHunter.” <https://github.com/lovaslin/pyBumpHunter>, 2020.
- [12] CMS Collaboration, A. M. Sirunyan et al., *Search for narrow and broad dijet resonances in proton-proton collisions at $\sqrt{s} = 13$ TeV and constraints on dark matter mediators and other new particles*, *JHEP* **08** (2018) 130, [[arXiv:1806.00843](#)].
- [13] M. Frate, K. Cranmer, S. Kalia, A. Vandenberg-Rodes, and D. Whiteson, *Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes*, [arXiv:1709.05681](#).