PCN: A deep learning approach to jet tagging utilizing novel graph construction methods and Chebyshev graph convolutions

Yash Semlani
University of North Carolina
Chapel Hill, NC 27599
yvsemlani@unc.edu

Mihir Relan
Johns Hopkins University
Baltimore, MD 21218
mrelan1@jh.edu

Krithik Ramesh
Massachusetts Institute of Technology
Broad Institute of MIT and Harvard
Cambridge, MA 02139
krithik@mit.edu

Abstract

Jet tagging is a classification problem in high-energy physics experiments that aims to identify the collimated sprays of subatomic particles, jets, from particle collisions and 'tag' them to their emitter particle. Advances in jet tagging present opportunities for searches of new physics beyond the Standard Model. Current approaches use deep learning to uncover hidden patterns in complex collision data. However, the representation of jets as inputs to a deep learning model have been varied, and often, informative features are withheld from models. In this study, we propose a graph-based representation of a jet that encodes the most information possible. To learn best from this representation, we design Particle Chebyshev Network (PCN), a graph neural network (GNN) using Chebyshev graph convolutions (ChebConv). ChebConv has been demonstrated as an effective alternative to classical graph convolutions in GNNs and has yet to be explored in jet tagging. PCN achieves a substantial improvement in accuracy over existing taggers and opens the door to future studies into graph-based representations of jets and ChebConv layers in high-energy physics experiments.

1 Introduction

The Large Hadron Collider (LHC) at CERN is a high-energy particle accelerator that collides particles to detect novel physics findings, such as the Higgs boson discovery [1, 2]. As proton-proton collisions continue in higher luminosity phases, more data will be available for research beyond the Standard Model. Machine learning is crucial for analyzing collider physics data, with jet tagging—a process to identify the elementary particle initiating a jet—being a primary application. This is essential for studying particle properties and interactions.

A key challenge in jet tagging is developing an expressive jet representation that captures complex inter-particle relationships. Representing jets as particle clouds [3–19], ordered sequences, and images [20–38] have been explored, with particle clouds and graph-based representations showing the most success. These have been with graph neural networks (GNNs) and transformer architectures, which outperform non-geometric methods like CNNs, RNNs, and MLPs. Particle clouds preserve permutation, rotational, and translational invariance but lack explicit modeling of inter-particle

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

relations. In contrast, graph-based representations incorporate relational features through nodes and edges, with graph convolutions synthesizing information across spatial scales. A notable advance in GNNs is the Chebyshev graph convolution (ChebConv) [39], which primarily prevents the explosion of the graph laplacian when raised to the ith power in a given convolutional filter. The usage of ChebConv in graph convolutional networks was shown to be effective in different tasks [39–41], and its usage in jet tagging has yet to be explored. Our main contributions are a graph-based jet representation with comprehensive particle-level features and the use of Chebyshev graph convolutions to enhance model performance.

2 Methodology

We extend existing methodologies, notably Chebyshev graph convolutions and other prominent graph neural networks in building PCN.

2.1 Graph Construction

A jet j is a collimated spray of subatomic particles from a collision, defined by its constituent particles $j_i = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where n is the number of particles in the jet. Each particle $\mathbf{x}_i$ within a jet is represented as a vector of features $\mathbf{x}_i = \{f_1, \ldots, f_{16}\}$, where f is one of the 16 features. We represent each jet as a graph $G = (V, E)$, where V is the set of vertices representing particles and E is the set of edges defined by a k-Nearest Neighbors (kNN) algorithm. The Euclidean distance between any two particles $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed as:

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{k=1}^{16} (f_{ik} - f_{jk})^2}$$

The set of k-nearest neighbors for each particle $\mathbf{x}_i$ is determined by selecting the k particles with the smallest distances d_{ij}, forming the set:

$$\mathcal{N}_k(\mathbf{x}_i) = \{\mathbf{x}_j \in j \mid j \neq i \text{ and } d_{ij} \text{ is among the smallest k distances}\}$$

The adjacency matrix A for the graph is then constructed as:

$$A_{ij} = \begin{cases} 1, & \text{if } x_j \in \mathcal{N}_k(x_i) \text{ or } x_i \in \mathcal{N}_k(x_j), \\ 0, & \text{otherwise.} \end{cases}$$

The parameter k balances local and global feature capture: lower k values result in sparser graphs, emphasizing localized interactions but potentially missing broader relationships. Higher k values create denser graphs but may introduce noise from weaker particle correlations. We use the elbow method to optimize k, evaluating the performance of downstream tasks (e.g., classification accuracy or clustering quality) against varying k values. Based on this analysis, we select $k = 3$ as a trade-off between sparsity and representational power. The Scipy `cKDTree` algorithm is used to efficiently identify the nearest neighbors and form graphs as adjacency matrices [38]. The split dimension d^* used in this algorithm is chosen as:

$$d^* = \arg\max_d \left(\max(x_{id}) - \min(x_{id})\right),$$

where x_{id} refers to the d-th feature of particle x_i within the current node of the tree. The split value is then set to the median of the d^*-th feature across particles in the node:

$$\text{Split value} = \text{median}(\{x_{id^*} \mid x_i \in \text{node}\}).$$

Our approach differs from others by providing more features per node, which empirically enhances the models' ability to learn expressive interactions between features.

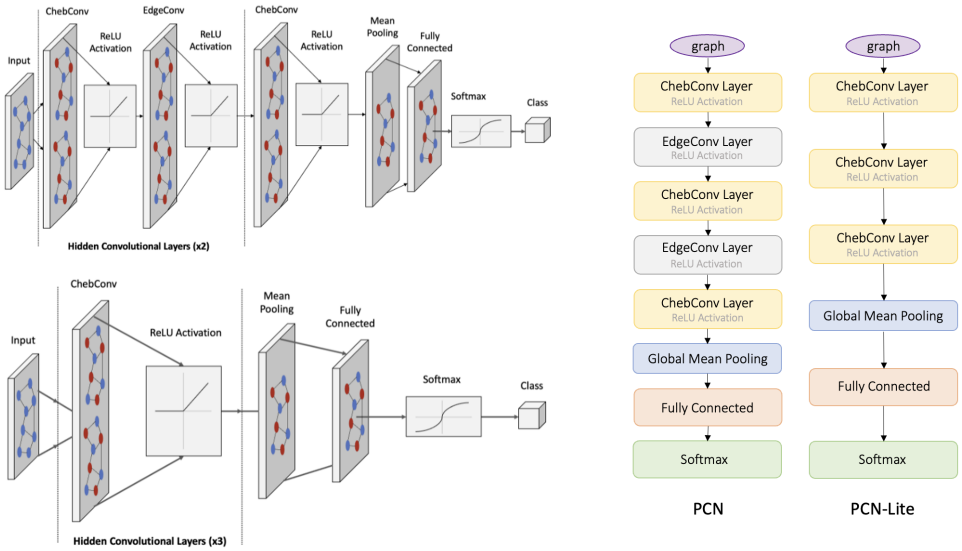

Figure 1: (*left*): graph neural network classification pathway of input graphs to PCN and PCN-Lite. (*right*): the network architectures of PCN and PCN-Lite.

2.2 Model Architectures

Our models, Particle Chebyshev Network (PCN) and the streamlined PCN-Lite, leverage Chebyshev Convolutional layers to process the constructed graphs. Figure 1 illustrates the model architectures.

Chebyshev Graph Convolutions We first process the graph-based representation of a jet, $\boldsymbol{G} \in \mathbb{R}^{N\times d}$, where N is the number of particles and d is the dimensionality of node features. In each layer ℓ, the graph $\boldsymbol{G}$ undergoes Chebyshev graph convolutions, applying learnable filters across nodes based on their local neighborhood.

Consider the 16-dimensional particle cloud consisting of N particles, which is then converted into a graph using the method explained in Section 2.1. A general graph convolutional operator is defined as $x' = p_w(L)x$, where x is the feature vector of the graph and $p_w(L)$ is a polynomial of form:

$$p_w(L) = \sum_{i=0}^{d} w_i L^i \tag{1}$$

where w is a learnable parameter representing the weight of the polynomial term. The weights are learned during training to adapt the operator to the graph data. For any given feature vector x, it is convolved with its neighboring nodes such that the fully expanded operator is:

$$x'_v = \sum_{i=0}^{d} w_i \sum_{u\in N(v)} L^i_{u,v} x_u \tag{2}$$

where $N(v)$ is the neighboring nodes of the convolved x_v. The Chebyshev convolutional operator differs from this by passing the normalized graph laplacian through a Chebyshev polynomial to create filters of the form:

$$p_w(L) = \sum_{i=0}^{d} w_i T_i(L_{norm}) \tag{3}$$

T_i is the i^{th} Chebyshev polynomial computed with initial conditions $T_0(x) = 1, T_1(x) = x$ and definition:

$$T_i(x) = 2xT_{i-1} - T_{i-2}x. \tag{4}$$

L_{norm} is the normalized graph Laplacian defined as:

$$L_{norm} = \frac{2L}{\lambda_{max}} - I \tag{5}$$

where λ_{max} is the largest eigenvalue of L and I is the identity matrix. The Chebyshev polynomial offers computational efficiency by approximating certain graph operations without fully diagonalizing the Laplacian, which becomes infeasible at large scales of data.

The local neighborhood $N(v)$ for convolution is determined by the K polynomial, $K_k(x)$, defined as the difference between the k^{th} and $(k-2)^{th}$ Chebyshev polynomials. ChebConv outperforms classical graphical convolutions by preserving local graph structures and mitigating oversmoothing.

Edge Convolutions Edge convolutional operators follow Chebyshev convolutions in PCN to capture global jet structures. At its core, EdgeConv utilizes two sets of independently learnable parameters to encode global shape information and local relational information between nodes resulting in its fully expanded form:

$$x_i' = max_{j \in N(i)} \Theta \cdot (x_i - x_j) + \Phi \cdot x_i \tag{6}$$

where Φ and Θ are independently learnable sets of weights.

3 Dataset

We use the JETCLASS dataset with 100M jets for training, 5M for validation, and 20M for testing, featuring ten evenly distributed jet types. A 1M jet sample suffices for training, with no performance gain from larger samples. We split these 1M jets into 800k/100k/100k for training, validation, and initial testing. Final metrics come from the full 20M jet test set. Training employs the AdamW optimizer [4] with a 1e-3 learning rate for up to 500 epochs, with early stopping if validation loss improvement is less than 0.0001 over 10 epochs.

4 Results

We trained and tested three related models: PCN-Edge (five layers of EdgeConv with no ChebConv layers), PCN-Inverse (three layers of EdgeConv interleaved with two layers of ChebConv), and PCN-Cheb (five layers of ChebConv with no EdgeConv layers). This analysis supports the motivation for interleaving EdgeConv layers between ChebConv layers, as summarized in Table 1.

Table 1: Accuracies of related PCN models on the 9 signal classes in the JETCLASS dataset.

	Macro-Acc	$H \to b\bar{b}$ Acc	$H \to c\bar{c}$ Acc	$H \to gg$ Acc	$H \to 4q$ Acc	$H \to lvqq'$ Acc	$t \to bqq'$ Acc	$t \to blv$ Acc	$W \to qq'$ Acc	$Z \to q\bar{q}$ Acc
PCN-Edge	0.891	0.887	0.881	0.890	0.870	0.889	0.907	0.929	0.887	0.869
PCN-Inverse	0.873	0.864	0.873	0.876	0.853	0.876	0.881	0.892	0.880	0.857
PCN-Cheb	0.853	0.822	0.858	0.856	0.839	0.860	0.856	0.876	0.810	0.869
PCN	**0.942**	**0.951**	**0.929**	**0.923**	**0.929**	**0.981**	**0.961**	**0.987**	**0.920**	**0.900**
PCN-Lite	0.936	0.950	0.923	0.917	0.919	0.973	0.957	0.984	0.914	0.892

PCN and PCN-Lite significantly outperform related models, highlighting two key conclusions. First, the interleaved configuration of ChebConv and EdgeConv layers in PCN enhances performance by balancing local and global feature learning. Second, adding more layers does not improve accuracy and may hinder the model's ability to capture complementary features.

We additionally compare the performances of PCN and PCN-Lite with four baseline models trained on the JETCLASS dataset: PFN [6], P-CNN [42], ParticleNet [5], and ParT [3]. The Particle Flow Network (PFN) architecture is based on the Deep Sets framework proposed by Zaheer et al. [43]. The P-CNN architecture was used in the CMS experiment by the DeepAK8 algorithm [26]. ParticleNet

follows a dynamic graph convolutional neural network similar to PCN using only EdgeConv layers [5]. Particle Transformer (ParT) is the state-of-the-art tagger utilizing a transformer architecture [3]. Table 2 summarizes the comparison of accuracy and AUC with the baseline models.

Table 2: Comparison of model performances on the JETCLASS dataset.

	Macro-Accuracy	Macro-AUC
PFN	0.772	0.9714
P-CNN	0.809	0.9789
ParticleNet	0.844	0.9849
ParT	0.861	**0.9877**
PCN-Lite	0.936	0.9400
PCN	**0.942**	0.9500

We conclude that PCN significantly improves upon the state-of-the-art accuracy set by ParT by 8.1%. PCN-Lite also improves upon ParT and other taggers in accuracy by a notable amount. The improvement in accuracy indicates a reduced number of misclassifications by PCN and PCN-Lite. Both models report lower AUC's than ParT and other taggers, but when contextualized to specific classes, the high AUC in conjunction with improved accuracy leads to a substantial improvement in discriminative power. This is particularly true for the $H\to 4q$ jets and $Z\to q\bar{q}$ jets, in which PCN achieves near perfect AUC.

5 Discussion

In this work, we propose two effective graph neural networks, PCN and PCN-Lite, combining Chebyshev graph convolutions and edge convolutions, achieving state-of-the-art accuracy on the JETCLASS dataset. Future research could explore using ChebConv with other graph convolutional layers or attention mechanisms for jet tagging. For more comprehensive comparisons of network performance, fine-tuning for training and testing on the top quark tagging dataset [44] and quark-gluon tagging dataset [45] would be an interesting domain to explore.

References

[1] Georges Aad, Tatevik Abajyan, B Abbott, J Abdallah, S Abdel Khalek, Ahmed Ali Abdelalim, R Aben, B Abi, M Abolins, OS AbouZeid, et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1): 1–29, 2012.

[2] Serguei Chatrchyan, Vardan Khachatryan, Albert M Sirunyan, Armen Tumasyan, Wolfgang Adam, Ernest Aguilo, Thomas Bergauer, M Dragicevic, J Erö, C Fabjan, et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1): 30–61, 2012.

[3] Huilin Qu, Congqiao Li, and Sitian Qian. Particle transformer for jet tagging. In *International Conference on Machine Learning*, pages 18281–18292. PMLR, 2022.

[4] Vinicius Mikuni and Florencia Canelli. Point cloud transformers applied to collider physics. *Machine Learning: Science and Technology*, 2(3):035027, 2021.

[5] Huilin Qu and Loukas Gouskos. Jet tagging via particle clouds. *Physical Review D*, 101(5): 056019, 2020.

[6] Shiqi Gong, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu. An efficient lorentz equivariant graph neural network for jet tagging. *Journal of High Energy Physics*, 2022(7):1–22, 2022.

[7] Frédéric A Dreyer and Huilin Qu. Jet tagging in the lund plane with graph networks. *Journal of High Energy Physics*, 2021(3):1–23, 2021.

[8] Vinicius Mikuni and Florencia Canelli. Abcnet: An attention-based method for particle tagging. *The European Physical Journal Plus*, 135:1–11, 2020.

[9] Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

[10] Alexander Bogatskiy, Timothy Hoffman, David W Miller, and Jan T Offermann. Pelican: Permutation equivariant and lorentz invariant or covariant aggregator network for particle physics. *arXiv preprint arXiv:2211.00454*, 2022.

[11] David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. *arXiv preprint arXiv:2305.11141*, 2023.

[12] Patrick T Komiske, Eric M Metodiev, and Jesse Thaler. Energy flow networks: deep sets for particle jets. *Journal of High Energy Physics*, 2019(1):1–46, 2019.

[13] Jun Guo, Jinmian Li, Tianjun Li, and Rao Zhang. Boosted higgs boson jet reconstruction via a graph neural network. *Physical Review D*, 103(11):116025, 2021.

[14] Chase Shimmin. Particle convolution for high energy physics. *arXiv preprint arXiv:2107.02908*, 2021.

[15] Eric A Moreno, Thong Q Nguyen, Jean-Roch Vlimant, Olmo Cerri, Harvey B Newman, Avikar Periwal, Maria Spiropulu, Javier M Duarte, and Maurizio Pierini. Interaction networks for the identification of boosted h→ bb decays. *Physical Review D*, 102(1):012010, 2020.

[16] Eric A Moreno, Olmo Cerri, Javier M Duarte, Harvey B Newman, Thong Q Nguyen, Avikar Periwal, Maurizio Pierini, Aidana Serikova, Maria Spiropulu, and Jean-Roch Vlimant. Jedi-net: a jet identification algorithm based on interaction networks. *The European Physical Journal C*, 80:1–15, 2020.

[17] Elias Bernreuther, Thorben Finke, Felix Kahlhoefer, Michael Krämer, and Alexander Mück. Casting a graph net to catch dark showers. *SciPost Physics*, 10(2):046, 2021.

[18] Matthew J Dolan and Ayodele Ore. Equivariant energy flow networks for jet tagging. *Physical Review D*, 103(7):074022, 2021.

[19] Partha Konar, Vishal S Ngairangbam, and Michael Spannowsky. Energy-weighted message passing: an infra-red and collinear safe graph neural network algorithm. *Journal of High Energy Physics*, 2022(2):1–30, 2022.

[20] Gregor Kasieczka, Tilman Plehn, Michael Russell, and Torben Schell. Deep-learning top taggers or the end of qcd? *Journal of High Energy Physics*, 2017(5):1–22, 2017.

[21] Sebastian Macaluso and David Shih. Pulling out all the tops with computer vision and deep learning. *Journal of High Energy Physics*, 2018(10):1–27, 2018.

[22] Frédéric A Dreyer, Gavin P Salam, and Grégory Soyez. The lund jet plane. *Journal of High Energy Physics*, 2018(12):1–42, 2018.

[23] Joshua Lin, Marat Freytsis, Ian Moult, and Benjamin Nachman. Boosting h→ bb¯ h\to b\overline {b} with machine learning. *Journal of High Energy Physics*, 2018(10):1–25, 2018.

[24] Yi-Lun Du, Kai Zhou, Jan Steinheimer, Long-Gang Pang, Anton Motornenko, Hong-Shi Zong, Xin-Nian Wang, and Horst Stöcker. Identifying the nature of the qcd transition in relativistic collision of heavy nuclei with deep learning. *The European Physical Journal C*, 80:1–17, 2020.

[25] Jakub Filipek, Shih-Chieh Hsu, John Kruper, Kirtimaan Mohan, and Benjamin Nachman. Identifying the quantum properties of hadronic resonances using machine learning. *arXiv preprint arXiv:2105.04582*, 2021.

[26] CMS collaboration et al. Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques. *Journal of Instrumentation*, 2020.

[27] James Barnard, Edmund Noel Dawe, Matthew J Dolan, and Nina Rajcic. Parton shower uncertainties in jet substructure analyses with deep neural networks. *Physical Review D*, 95(1): 014018, 2017.

[28] Patrick T Komiske, Eric M Metodiev, and Matthew D Schwartz. Deep learning in color: towards automated quark/gluon jet discrimination. *Journal of High Energy Physics*, 2017(1):1–23, 2017.

[29] Suyong Choi, Seung J Lee, and Maxim Perelstein. Infrared safety of a neural-net top tagging algorithm. *Journal of High Energy Physics*, 2019(2):1–14, 2019.

[30] Jinmian Li, Tianjun Li, and Fang-Zhou Xu. Reconstructing boosted higgs jets from event image segmentation. *Journal of High Energy Physics*, 2021(4):1–22, 2021.

[31] Katherine Fraser and Matthew D Schwartz. Jet charge and machine learning. *Journal of High Energy Physics*, 2018(10):1–18, 2018.

[32] 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. *Physical Review D*, 94(11):112002, 2016.

[33] Shannon Egan, Wojciech Fedorko, Alison Lister, Jannicke Pearkes, and Colin Gay. Long short-term memory (lstm) networks with jet constituents for boosted top tagging at the lhc. *arXiv preprint arXiv:1711.09059*, 2017.

[34] Emil Bols, Jan Kieseler, Mauro Verzetti, Markus Stoye, and Anna Stakia. Jet flavour classification using deepjet. *Journal of Instrumentation*, 15(12):P12012, 2020.

[35] Josh Cogan, Michael Kagan, Emanuel Strauss, and Ariel Schwartzman. Jet-images: computer vision inspired techniques for jet tagging. *Journal of High Energy Physics*, 2015(2):1–16, 2015.

[36] Anja Butter, Gregor Kasieczka, Tilman Plehn, and Michael Russell. Deep-learned top tagging with a lorentz layer. *SciPost Physics*, 5(3):028, 2018.

[37] Jannicke Pearkes, Wojciech Fedorko, Alison Lister, and Colin Gay. Jet constituents for deep neural network based top quark tagging. *arXiv preprint arXiv:1704.02124*, 2017.

[38] Gregor Kasieczka, Nicholas Kiefer, Tilman Plehn, and Jennifer Thompson. Quark-gluon tagging: Machine learning vs detector. *SciPost Physics*, 6(6):069, 2019.

[39] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. *arXiv preprint arXiv:2202.03580*, 2022.

[40] Lyuchao Liao, Zhiyuan Hu, Yuxin Zheng, Shuoben Bi, Fumin Zou, Huai Qiu, and Maolin Zhang. An improved dynamic chebyshev graph convolution network for traffic flow prediction with spatial-temporal attention. *Applied Intelligence*, 52(14):16104–16116, 2022.

[41] Osman Boyaci, M Rasoul Narimani, Katherine Davis, and Erchin Serpedin. Cyberattack detection in large-scale smart grids using chebyshev graph convolutional networks. In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, pages 217–221. IEEE, 2022.

[42] CMS collaboration et al. Boosted jet identification using particle candidates and deep neural networks. *Detector Performance Figures: CMS-DP-17-049*, 2017.

[43] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[44] Gregor Kasieczka, Tilman Plehn, Jennifer Thompson, and Michael Russel. Top quark tagging reference dataset. *Version v0 (2018_03_27). Mar*, 2019.

[45] P Komiske, E Metodiev, and J Thaler. Pythia8 quark and gluon jets for energy flow. *Zenodo. doi*, 10, 2019.