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

Citation: Semlani, Y., Relan, M. & Ramesh, K. PCN: a deep learning approach to jet tagging utilizing novel graph construction methods and Chebyshev graph convolutions. J. High Energ. Phys. 2024, 247 (2024).

Published Version: 10.1007/jhep07(2024)247

Publisher: Springer Science and Business Media LLC

Permanent Link: https://hdl.handle.net/1721.1/155807

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Published for SISSA by Springer

Received: *September 19, 2023*
Revised: *February 11, 2024*
Accepted: *July 1, 2024*
Published: *July 26, 2024*

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

Yash Semlani,[a] **Mihir Relan**[b] **and Krithik Ramesh**[c]

[a]*University of North Carolina at Chapel Hill, Chapel Hill, NC, U.S.A.*
[b]*Johns Hopkins University, 3400 N. Charles St, Baltimore, MD, U.S.A.*
[c]*Massachusetts Institute of Technology, 32 Vassar St, Cambridge, Massachusetts, U.S.A.*

E-mail: yvsemlani@unc.edu, mrelan1@jh.edu, 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. Code is available at https://github.com/YVSemlani/PCN-Jet-Tagging

Keywords: Jets and Jet Substructure, Top Quark

ArXiv ePrint: 2309.08630

https://doi.org/10.1007/JHEP07(2024)247

Contents

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]. Proton-proton collisions will continue to advance research beyond the Standard Model, particularly in higher luminosity phases, which will provide more data to be gathered for observation. Machine learning is a pragmatic approach to analyzing the data from collider physics. One of its primary applications is jet tagging. Jet tagging is the process of identifying the elementary particle responsible for initiating a jet, which is the fragmentation and hadronization of partons. Ongoing research in jet tagging is crucial to studying the fundamental properties and interactions of elementary particles and the search for physics beyond the Standard Model. Jets are difficult to classify because machine learning models must effectively separate background jets from signal jets, requiring a generalized understanding of Quantum Chromodynamics (QCD).

A prevailing challenge in the field is developing an expressive representation of jets that captures complex relational information between jets. A variety of strategies for representing jets have been explored, including particle clouds [3–19], ordered sequences, and images [20–38]. Notably, particle cloud and graph representations have seen widespread success in conjunction with transformer architectures and graph neural networks (GNNs), consistently surpassing the performance of non-geometric methods like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and multilayer perceptrons MLPs). This advantage underscores the importance of a representation that aligns with the fundamental nature of jets. Particle clouds, in particular, are well-suited due to their preservation of permutation, rotational, and translational invariance properties. However, a limitation of particle clouds is their lack of explicit modeling of inter-particle relations within the jet structure, which contributes significantly to the fragmentation processes that take place during jet formation.

JHEP07(2024)247

The choice of jet representation significantly influences the capabilities of machine learning models in this domain. The state-of-the-art employs particle clouds often augmented with particle interaction information. Graph-based representations present a compelling alternative, with nodes representing particles and edges encoding relational features. It is worth noting that though graphs can theoretically encompass the same information as particle clouds, current graph-based approaches often fall short.

Differences in model architecture can present significant opportunities for increasing model performance when combined with a suitable representation. Qu. et al. [3] utilizes a transformer architecture that incorporates relational information into prediction through the particle attention layer. However, transformer models lack methods for analyzing these relational features separately. In contrast, graph-based architectures utilize edge convolutions and graph convolutions, which are heavily influenced by the relational structure of the graph. Our main contribution is the development of a graph-based representation of jets that incorporates comprehensive particle-level features and the usage of Chebyshev graph convolutions to synthesize information across disparate spatial scales.

2 Related works

Deep learning using GNNs is a natural choice for jet tagging as jets are most naturally represented using graph-based representations, such as particle clouds. This makes GNNs particularly well-suited for capturing the relational information within jets and extracting features for accurate tagging. Studies using GNNs for jet tagging have explored different graph construction methods for jet representation and model architectures for optimal performance.

The usage of a particle cloud jet representation was first proposed by Qu & Gouskos [5], who built upon the notion of graph construction based on point-cloud structures [39] and treat a jet as an invariant set of particles plotted in the η-ϕ space. Gong et al. [6] advanced these particle cloud representations by making the edges of their graph represent particle interactions in the Minkowski space. Qu et al. [3] construct their particle clouds in the η-ϕ space with each node representing a particle and edges representing particle interactions. Another graph-based representation by Dreyer et al. [7] introduced using the Lund tree of a jet, which encodes a jet's clustering history and substructure. In these graph construction methods, kinematic, identification, and trajectory properties are provided as node features. Dreyer et al. [7] provide 5 features per node, Gong et al. [6] provide 12 features per node, and Qu et al. [3] provide 16 features per node. Models with more information encoded performed experimentally better, with Qu et al. [3] demonstrating the best performance, followed by Gong et al. [6] and Dreyer et al. [7] respectively.

Model architectures yield very different performance on classification accuracy in jet tagging. ParticleNet by Qu & Gouskos [5] utilizes Edge Convolution (EdgeConv) blocks to classify using the particle cloud representation. LundNet by Dreyer et al. [7] follows a similar model architecture using EdgeConv blocks. LorentzNet by Gong et al. [6] utilizes Lorentz Equivariance Group Blocks (LGEB) consisting of multiple multilayer perceptrons (MLPs) and encoder-decoder layers. Particle Transformer (ParT) by Qu et al. [3] utilizes attention blocks consisting of multi-head attention modules. These mentioned model architectures used techniques that catered to the particle cloud representation and their respective graph

JHEP07(2024)247

construction methods. For a graph-based representation, however, graph convolutional layers to extract relational information would be useful to complement EdgeConv blocks. A notable advance in GNNs is the Chebyshev graph convolution (ChebConv) [40], which primarily prevents the explosion of the graph laplacian when raised to the i^{th} power in a given convolutional filter. The usage of ChebConv in graph convolutional networks was shown to be effective in different tasks [40–42], and its usage in jet tagging has yet to be explored.

3 Methodology

Our jet tagging approach integrates salient features from existing methodologies, most notably ParT, Chebyshev graph convolutions, and other prominent graph neural networks. The subsequent sections will describe the core components of our study.

3.1 Graph construction

A jet j is a collimated spray of subatomic particles from a collision. We define j by its constituent particles $j_i = \{x_1, \ldots, x_n\}$, where n is the number of particles in the jet. Note that the number of particles in each jet may be different. Each particle x within a jet j can be represented as a vector of features $x_i = \{f_1, \ldots, f_{16}\}$, where f is one of the 16 features provided in the dataset.

We represent each jet as a graph denoted as $G = (V, E)$, where V is the set of vertices and E is the set of edges. We define the vertices V as $\{1, \ldots, n\}$, where each particle within the jet is considered to be a vertex. We define the edges as $E = \{(u, v)|u, v \in V\}$, where u and v are vertices defined by a k-Nearest Neighbors (kNN) algorithm.

It is imperative to determine the appropriate value for k (number of nearest neighbors). Selections of k that are excessively low result in denser graphs, fostering increased interconnectivity among particles and capturing more localized interactions but introducing noise. Conversely, excessively high values of k yield sparser graphs with fewer interconnections between particles, potentially capturing more global features but overlooking crucial local interactions. Moreover, higher values for k result in the exclusion of smaller jets, given the constraint that k must be less than the number of particles in a jet. Notably, when k is set within range of 7-10 (the ideal value suggested by the elbow method), a proportion of the *H→lvqq'* jets is omitted from the training dataset. Preserving the representation of these smaller jets ensures the model's proficiency in discerning subtle interactions within collisions. Utilizing the elbow method while seeking to prevent the omission of jets, we find the optimal value of k as $k = 3$.

Our jet, j, is given as input to the Scipy cKDTree which creates a binary tree for quick lookup of nearest neighbors using a dimensional heuristic further explained by Maneewongvatana et al. [43]. Each node's three nearest neighbors are then retrieved and linked together with an edge. Finally, the completed graph is outputted as an adjacency matrix for use in training. Figure 1 visualizes the graph-based representation versus a standard particle cloud.

Our pre-processing method differs most significantly from other graph-based representations in the features provided at each node. LundNet [7], a highly accurate graph-based representation, provides fewer features per node than our pre-processing method, meaning we encode more information overall. The features provided to the model by the authors of

JHEP07(2024)247

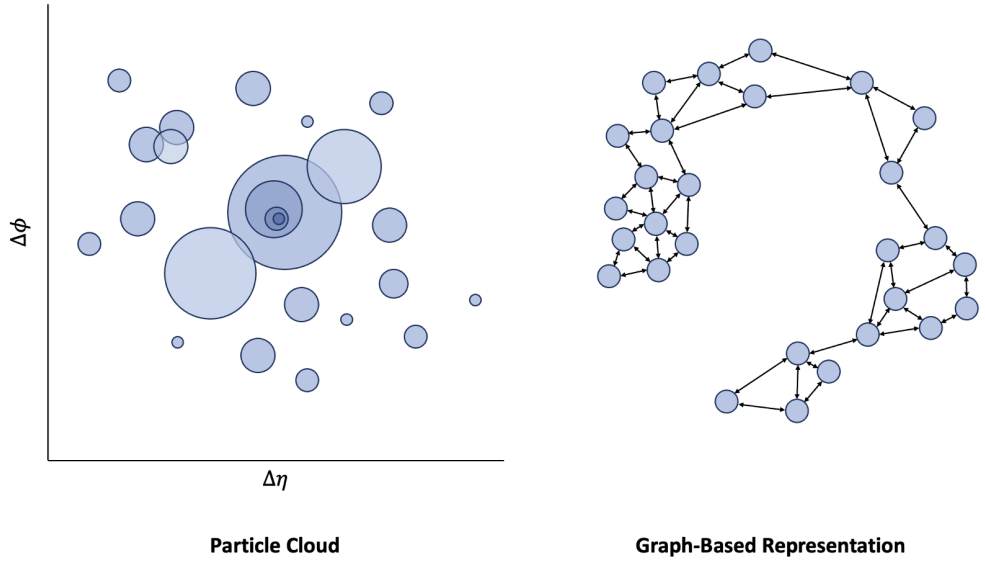

Figure 1. *Left*: particle cloud representation of a jet used by state-of-the-art ParT [3]. *Right*: graph-based representation of a jet used by PCN and PCN-Lite.

LundNet are also high-level features that can be derived from kinematics, which can hinder the creation of new, non-theory-based features by a model. Investigations of LorentzNet [6] and ParticleNet [5] follow very similar pre-processing methods to ours. In relation to these studies, our major difference is in the model architecture we utilize for classification.

3.2 Model architectures

Our models, Particle Chebyshev Network (PCN) and the streamlined PCN-Lite, leverage Chebyshev Convolutional layers (ChebConv) to process the constructed graphs (see section 3.1). PCN incorporates both ChebConv and Edge Convolutional (EdgeConv) layers for enhanced feature extraction while PCN-lite uses only EdgeConv. Figure 2 illustrates the model architectures.

3.2.1 Chebyshev graph convolutions for localized interaction analysis

The first stage of PCN involves processing the graph-based representation of a particle jet, expressed as $\boldsymbol{G} \in \mathbb{R}^{N\times d}$, where N represents the number of particles (nodes) in the jet and d is the dimensionality of node features. In each layer ℓ, the graph $\boldsymbol{G}$ undergoes transformation through Chebyshev graph convolutions, focusing on local particle interactions. The convolution applies a series of learnable filters across the graph's nodes, considering their local neighborhood, defined by the graph structure. This process, crucial for discerning subtle particle dynamics, enables the model to capture local dependencies intrinsic to jet formation. Concurrently, these layers employ Chebyshev polynomials to adaptively filter node features, ensuring the capture of relevant local information within the jet.

The Chebyshev convolutional operation can be formulated as:

$$\boldsymbol{G'}_{\text{Cheb}} = \text{ChebConv}\left(\boldsymbol{G}, \boldsymbol{W}^{\ell}_{\text{Cheb}}\right) \tag{3.1}$$

where $\boldsymbol{W}^{\ell}_{\text{Cheb}}$ represents the learnable weights in the Chebyshev convolution at layer ℓ.

JHEP07(2024)247

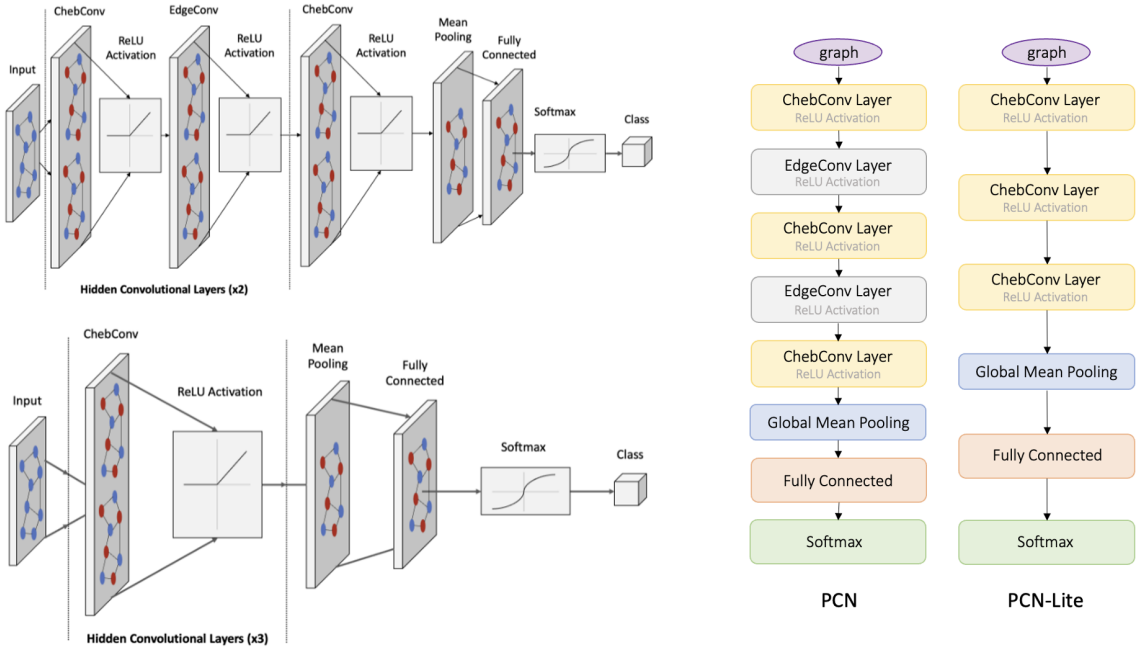

Figure 2. *Left*: graph neural network classification pathway of input graphs to PCN and PCN-Lite. *Right*: the network architectures of PCN and PCN-Lite.

Consider the 16-dimensional particle cloud consisting of N particles, which is then converted into a graph using the method explained in section 3.1. 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:

$$p_w(L) = \sum_{i=0}^{d} w_i L^i \tag{3.2}$$

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:

$$x'_v = \sum_{i=0}^{d} w_i \sum_{u \in N(v)} L^i_{u,v} x_u \tag{3.3}$$

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:

$$p_w(L) = \sum_{i=0}^{d} w_i T_i(L_{\text{norm}}) \tag{3.4}$$

Therefore, the full Chebyshev convolutional operator is:

$$x' = \sum_{i=0}^{d} w_i T_i(L_{\text{norm}}) x \tag{3.5}$$

JHEP07(2024)247

and can be expanded as:

$$x'_v = \sum_{i=0}^{d} w_i \sum_{u \in N(v)} T_i(L_{\text{norm}})_{u,v} x. \tag{3.6}$$

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

$$T_i(x) = 2xT_{i-1} - T_{i-2}x. \tag{3.7}$$

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

$$L_{\text{norm}} = \frac{2L}{\lambda_{\max}} - I \tag{3.8}$$

where $\lambda_{\max}$ 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 $N(v)$ over which the convolution spans is decided by the K polynomial. The K polynomial, $K_k(x)$, is defined as the difference between the k^{th} Chebyshev polynomial, $T_k(x)$, and the $(k-2)^{th}$ Chebyshev polynomial, $T_{k-2}(x)$. By analyzing the skipping distance in graphs of $K_k(x)$ compared to $T_k(x)$, we gain valuable insights into the polynomial's approximation properties, which directly influence the performance of Chebyshev convolutions. Understanding how the number of intersections with the x-axis between consecutive zeros differs between $T_k(x)$ and $K_k(x)$ allows us to refine the techniques used for particle identification and classification in jet tagging applications. For our models, we set the k-value of the Chebyshev polynomial kernel to two, such that it is one less than the number of nearest neighbors per nearest neighbor node retrieval to capture substructure interactions.

This convolutional method is similar to image convolutions in utilizing $x(v)$ as analogous to a central pixel in image convolution and $N(v)$, the neighborhood of nodes around $x(v)$, as analogous to a patch in image convolution. We found that ChebConv yielded better performance compared to classical graphical convolutions. With classical graphical convolutions, a sum of neighbor node features is multiplied by a learnable weight matrix, which may lead to oversmoothing that loses fine details and nuances in the data. The repeated convolutions may also lead to vanishing gradients, which causes less informative training. However, ChebConv leverages Chebyshev polynomials, which when applied as filters, focus on a small neighborhood around each node that helps preserve the local structure of the graph and mitigate oversmoothing. This is particularly beneficial when dealing with jets that exhibit patterns or structures at various spatial resolutions. ChebConv also adapts well to graphs with different sizes as they operate on the graph structure rather than relying on a fixed-size filter, which becomes advantageous when dealing with jets of different numbers of constituent particles.

3.2.2 Edge convolutions for global jet structure analysis

The edge convolutional operators are introduced after the Chebyshev convolutional operators in PCN to capture global jet structures. This stage is pivotal for comprehending the overall

JHEP07(2024)247

jet formation and identifying patterns indicative of specific particle origins. EdgeConv excels in capturing the broader relational context within the jet, analyzing how localized interactions aggregate to form distinct jet features observable at larger scales. This global perspective is essential for distinguishing between different types of jets, particularly in complex collision events.

The EdgeConv operation integrates the Chebyshev-processed features:

$$\boldsymbol{G'}_{\text{Edge}} = \text{EdgeConv}\left(\boldsymbol{G'}_{\text{Cheb}}, \boldsymbol{W}^{\ell}_{\text{Edge}}\right) \tag{3.9}$$

where $\boldsymbol{W}^{\ell}_{\text{Edge}}$ denotes the learnable weights in the EdgeConv layer.

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:

$$x'_i = max_{j \in N(i)} \Theta \cdot (x_i - x_j) + \Phi \cdot x_i \tag{3.10}$$

where Φ and Θ are independently learnable sets of weights.

3.2.3 Synthesis of local and global features in jet tagging

The architectural decision to interleave EdgeConv layers between ChebConv layers is grounded in the rationale that this configuration facilitates the concurrent extraction of local features through ChebConv and relational information via EdgeConv at each processing stage. While a sequential arrangement of ChebConv layers followed by EdgeConv layers may yield a more hierarchical feature extraction, it defers the integration of global context until the EdgeConv layers. Given the inherently intricate structures exhibited by jets across various resolutions, an interleaved structure is favored for the comprehensive extraction of both local and global features.

Furthermore, the interleaved structure proves advantageous in accommodating different jet compositions, characterized by varying numbers of constituent particles. The adaptability of ChebConv to varying graph sizes is particularly beneficial in this context. The introduction of an EdgeConv layer between ChebConv layers enables the model to dynamically adjust the receptive field, allowing it to flexibly adapt to the diverse structures of jets. In contrast, employing consecutive ChebConv layers may result in the capturing of information solely from the immediate neighborhood of nodes, progressively extending to larger neighborhoods. This approach could potentially lead to the model inadequately capturing the local structures of jets with fewer particles and oversampling those with more particles. An analysis of different model configurations is presented in section 4.2.

4 Results

The performance of PCN and PCN-Lite was evaluated on the JETCLASS dataset introduced by Qu et al. [3]. We investigated evaluations of accuracy, area under the ROC curve (AUC), and area under the precision-recall curve (AUPR) for each model.

JHEP07(2024)247

	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	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

Table 1. Accuracy of PCN and PCN-Lite on the 9 signal classes in the JETCLASS dataset.

	Macro-AUC	$H\to b\bar{b}$ AUC	$H\to c\bar{c}$ AUC	$H\to gg$ AUC	$H\to 4q$ AUC	$H\to lvqq'$ AUC	$t\to bqq'$ AUC	$t\to blv$ AUC	$W\to qq'$ AUC	$Z\to q\bar{q}$ AUC
PCN	0.95	0.95	0.94	0.90	1.00	0.98	0.97	0.92	0.95	0.99
PCN-Lite	0.94	0.94	0.93	0.88	1.00	0.98	0.97	0.90	0.94	0.99

Table 2. AUC of PCN and PCN-Lite on the JETCLASS dataset.

4.1 Dataset and experiment

We use the full set of features provided in the JETCLASS dataset and follow the graph construction method detailed in section 3.1. The generated graphs are given as inputs to our models. The JETCLASS dataset consists of 100M jets for training, 5M for validation, and 20M for testing. Ten different types of jets are provided, evenly distributed into the ten classes. Jet tagging on this dataset is a multi-class classification task, making studies on it more conducive for LHC tasks. Generation of the JETCLASS dataset was through Markov Chain Monte Carlo (MCMC) methods consistent with those used in CERN collaborations. More details about the production of particles and clustering of the jets can be found in ref. [3]. A sample size of 1M jets for the training process was determined to be sufficient as training with samples of between 2M and 10M provided no increase in performance. The 1M jet events were split 800k/100k/100k for training, validation, and initial testing. Final metrics are derived from the full testing set of 20M jets.

For implementation, we employ the AdamW optimizer [4] with a learning rate of 1e-3. Both models are given a maximum of 500 epochs to train over. A convergence threshold is set to 0.0001, and 10 epochs are given to a model for an improvement in validation loss greater than the convergence threshold. If the improvement is not greater, training is stopped.

PCN and PCN-Lite are evaluated on their accuracy and area under the ROC curve (AUC) to quantify overall performance. We also analyze the area under the precision-recall curves (AUPR) for each model. AUPR is a critical metric for jet tagging to assess how well a model can identify positive instances of a jet without accidentally classifying background instances as positive (false-positive). In addition, the background rejection metric is calculated for each class, as it is directly related to the discovery potential in high-energy physics experiments [3]. Background rejection is only calculated for PCN and follows the methodology described in [3]. Table 1 summarizes the accuracy of PCN and PCN-Lite.

PCN outperforms PCN-Lite in accuracy across all classes. A noticeable trend is the consistency of performance between both PCN and PCN-Lite. Both the PCN and PCN-Lite architectures were most accurate in classifying $t\to blv$ jets and $H\to lvqq'$ jets and were most inaccurate in classifying $Z\to q\bar{q}$ jets. Table 2 summarizes the AUC of PCN and PCN-Lite.

PCN overall outperforms PCN-Lite in AUC, but their performance is equal in some classes. The consistency of performance is present again. Both PCN and PCN-Lite achieve

JHEP07(2024)247

	Macro-AUPR	$H\to b\bar{b}$ AUPR	$H\to c\bar{c}$ AUPR	$H\to gg$ AUPR	$H\to 4q$ AUPR	$H\to lvqq'$ AUPR	$t\to bqq'$ AUPR	$t\to blv$ AUPR	$W\to qq'$ AUPR	$Z\to q\bar{q}$ AUPR
PCN	0.80	0.70	0.63	0.48	0.98	0.88	0.85	0.67	0.70	0.96
PCN-Lite	0.77	0.65	0.59	0.41	0.98	0.84	0.82	0.61	0.66	0.93

Table 3. AUPR of PCN and PCN-Lite on the JETCLASS dataset.

	$H\to b\bar{b}$ $Rej_{50\%}$	$H\to c\bar{c}$ $Rej_{50\%}$	$H\to gg$ $Rej_{50\%}$	$H\to 4q$ $Rej_{50\%}$	$H\to lvqq'$ $Rej_{99\%}$	$t\to bqq'$ $Rej_{50\%}$	$t\to blv$ $Rej_{99.5\%}$	$W\to qq'$ $Rej_{50\%}$	$Z\to q\bar{q}$ $Rej_{50\%}$
PCN	15.16	13.10	25.70	253.02	32.20	27.70	12.41	78.02	1353.02

Table 4. Background rejection of PCN on the JETCLASS dataset.

optimal AUC in classifying $H\to 4q$ jets and near-optimal AUC in classifying $Z\to q\bar{q}$ jets. This indicates the models are strongest in discriminating between instances of these jets compared to others. Both models had their worst AUC's in classifying $H\to gg$ jets.

The observed phenomenon in the vector boson jets, wherein $Z\to q\bar{q}$ exhibits lower accuracy than $W\to qq'$ but yields improved AUC, can be attributed to the inherent differences in the characteristics of the two decay channels. While accuracy focuses on the overall correctness of predictions, AUC considers the model's ability to discriminate between positive and negative instances across a range of decision thresholds. Notably, the model demonstrates a high level of confidence in the predictions it gets right for $W\to qq'$ jets, leading to a more concentrated distribution of high-confidence correct predictions. In contrast, the inherent complexity and intricacies of $Z\to q\bar{q}$ may introduce challenges in achieving consistently high confidence levels for correct predictions, thus impacting overall accuracy. Table 3 summarizes the AUPR of PCN and PCN-Lite.

The baselines for the AUPR of each class are equal to the fraction of positives [45], calculated as the number of positive instances over the total number of instances. In our setup, the test data is near-perfectly balanced, with each class having 2M positive instances. Thus, the baseline AUPR for comparison is 0.1. PCN consistently outperforms PCN-Lite in almost all classes. The performances are also consistent with their respective AUC performances. Both PCN and PCN-Lite achieve the best AUPR in classifying $H\to 4q$ jets and $Z\to q\bar{q}$ jets and the worst AUPR in classifying $H\to gg$ jets. Table 4 summarizes the background rejection of PCN .

The background rejection for each signal class (background is considered q/g jets) is relatively consistent with values reported in previous studies. Our discrepancy in values in relation to the previous state-of-the-art, ParT, in ref. [3] is due to not including the softmax probability function for calculation of the true positive rate (TPR) and false positive rate (FPR). When interpreting our background rejection, PCN approaches event selection by favoring the count of false positives rather than potentially discarding events containing novel or unexpected phenomena. This characteristic enables PCN to capture a greater number of events for further investigation and analysis.

To corroborate the motivations of interleaving EdgeConv layers between ChebConv layers, we trained and tested three related models:

JHEP07(2024)247

	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

Table 5. Accuracies of related PCN models on the 9 signal classes in the JETCLASS dataset.

1. PCN-Edge: five layers of EdgeConv with no ChebConv layers
2. PCN-Inverse: three layers of EdgeConv interleaved with two layers of ChebConv
3. PCN-Cheb: five layers of ChebConv with no EdgeConv layers

PCN-Edge investigates the model's ability to capture local graph structures and relationships without the additional spectral processing provided by ChebConv layers. We can think of this as extracting only the inter-particle relations and not the particle-level features. PCN-Inverse investigates whether emphasizing local edge-based convolutions over global spectral graph convolutions affects the model's ability to capture essential global features. PCN-Cheb investigates the effect of more ChebConv layers in comparison to PCN-Lite (only three ChebConv layers). Each model was trained on 1M jets with the same 800k/100k/100k for training, validation, and initial testing, and the same final testing set of 20M jets. Implementation follows the same procedure detailed in section 4.1. The results are summarized in the table below.

We observe that for all classes, PCN and PCN-Lite substantially outperform the related models. We can infer two main conclusions from this. First, the performance difference between PCN and the related models supports the interleaved configuration of ChebConv and EdgeConv layers. PCN-Inverse, by prioritizing EdgeConv layers over ChebConv layers, may have become more biased towards local structures, which led to a diminished capability to discern broader graph patterns. Second, the performance difference between PCN-Cheb, PCN-Edge and PCN-Lite suggests that adding more layers, either ChebConv or EdgeConv, does not lead to a more accurate model. In fact, the observed trend prompts the postulation that an excessive number of layers, be it ChebConv or EdgeConv, may constrain the model's capacity to learn the intricacies of the opposite feature type. This supports a balanced layer configuration (like PCN) for optimal performance.

4.2 Comparison to state-of-the-art models

We compare the performances of PCN and PCN-Lite with four baseline models trained on the JETCLASS dataset: PFN [12], P-CNN [46], ParticleNet [5], and ParT [3]. The Particle Flow Network (PFN) architecture is based on the Deep Sets framework proposed by Zaheer et al. [47]. 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-

JHEP07(2024)247

	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

Table 6. Comparison of model performances on the JETCLASS dataset.

	Accuracy	# params	Time (CPU) [ms]	Time (GPU) [ms]
PFN	0.772	82k	0.8	0.018
P-CNN	0.809	354k	1.6	0.020
ParticleNet	0.844	370k	23	0.92
PCN-Lite	0.936	148k	18.6	0.024
PCN	**0.942**	165k	19.4	0.035

Table 7. Comparison of number of parameters and inference time on the JETCLASS dataset.

art tagger utilizing a transformer architecture [3]. Table 6 summarizes the comparison of accuracy and AUC with the baseline models.

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.

Table 7 demonstrates the model complexity of PCN and PCN-Lite in comparison to previously researched jet tagging models. We observe that the PCN achieves a notable increase in accuracy on the JETCLASS dataset with substantially fewer trainable parameters. This efficiency can be largely attributed to the use of Chebyshev graph convolutions, which enable PCN to capture information from neighboring nodes at different graph distances. Additionally, the spectral nature of Chebyshev polynomials allows them to approximate localized filters more effectively. However, we do note that the inference time for PCN and PCN-Lite is notably longer than that of other models. This can be explained by the increased computational complexity associated with spectral graph convolutions (ChebConv), which involves the computation of eigenvalues and eigenvectors that can lead to longer inference times compared to simpler convolutional methods.

5 Discussion

In this work, we propose two performant graph neural networks Particle Chebyshev Network (PCN) and PCN-Lite that utilize a combination of Chebyshev graph convolutions and edge convolutions. PCN achieves state-of-the-art accuracy in classification tasks on the JETCLASS

JHEP07(2024)247

dataset. Future areas of research could extend into the usage of ChebConv in conjunction with other graph convolutional layers for jet tagging applications. Further investigations extending beyond convolution graph operators, such at attention mechanisms could elicit different interactions and performance on JETCLASS. For more comprehensive comparisons of network performance, fine-tuning for training and testing on the top quark tagging dataset [48] and quark-gluon tagging dataset [49] would be an interesting domain to investigate further.

JHEP07(2024)247

References

[1] ATLAS collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett. B* **716** (2012) 1 [arXiv:1207.7214] [INSPIRE].

[2] CMS collaboration, *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, *Phys. Lett. B* **716** (2012) 30 [arXiv:1207.7235] [INSPIRE].

[3] H. Qu, C. Li and S. Qian, *Particle Transformer for Jet Tagging*, in *International Conference on Machine Learning*, PMLR (2022) pp. 18281–18292 [arXiv:2202.03772] [INSPIRE].

[4] V. Mikuni and F. Canelli, *Point cloud transformers applied to collider physics*, *Mach. Learn. Sci. Tech.* **2** (2021) 035027 [arXiv:2102.05073] [INSPIRE].

[5] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, *Phys. Rev. D* **101** (2020) 056019 [arXiv:1902.08570] [INSPIRE].

[6] S. Gong et al., *An efficient Lorentz equivariant graph neural network for jet tagging*, *JHEP* **07** (2022) 030 [arXiv:2201.08187] [INSPIRE].

[7] F.A. Dreyer and H. Qu, *Jet tagging in the Lund plane with graph networks*, *JHEP* **03** (2021) 052 [arXiv:2012.08526] [INSPIRE].

[8] V. Mikuni and F. Canelli, *ABCNet: An attention-based method for particle tagging*, *Eur. Phys. J. Plus* **135** (2020) 463 [arXiv:2001.05311] [INSPIRE].

[9] A. Bogatskiy et al., *Lorentz Group Equivariant Neural Network for Particle Physics*, in *International Conference on Machine Learning*, PMLR (2020) pp. 992–1002 [arXiv:2006.04780] [INSPIRE].

[10] A. Bogatskiy, T. Hoffman, D.W. Miller and J.T. Offermann, *PELICAN: Permutation Equivariant and Lorentz Invariant or Covariant Aggregator Network for Particle Physics*, arXiv:2211.00454 [INSPIRE].

[11] D. Ruhe, J. Brandstetter and P. Forré, *Clifford Group Equivariant Neural Networks*, arXiv:2305.11141 [INSPIRE].

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

[13] E.A. Moreno et al., *JEDI-net: a jet identification algorithm based on interaction networks*, *Eur. Phys. J. C* **80** (2020) 58 [arXiv:1908.05318] [INSPIRE].

[14] E.A. Moreno et al., *Interaction networks for the identification of boosted $H \to b\bar{b}$ decays*, *Phys. Rev. D* **102** (2020) 012010 [arXiv:1909.12285] [INSPIRE].

[15] E. Bernreuther et al., *Casting a graph net to catch dark showers*, *SciPost Phys.* **10** (2021) 046 [arXiv:2006.08639] [INSPIRE].

[16] J. Guo, J. Li, T. Li and R. Zhang, *Boosted Higgs boson jet reconstruction via a graph neural network*, *Phys. Rev. D* **103** (2021) 116025 [arXiv:2010.05464] [INSPIRE].

[17] M.J. Dolan and A. Ore, *Equivariant Energy Flow Networks for Jet Tagging*, *Phys. Rev. D* **103** (2021) 074022 [arXiv:2012.00964] [INSPIRE].

[18] P. Konar, V.S. Ngairangbam and M. Spannowsky, *Energy-weighted message passing: an infra-red and collinear safe graph neural network algorithm*, *JHEP* **02** (2022) 060 [arXiv:2109.14636] [INSPIRE].

[19] C. Shimmin, *Particle Convolution for High Energy Physics*, arXiv:2107.02908 [INSPIRE].

[20] G. Kasieczka, T. Plehn, M. Russell and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, *JHEP* **05** (2017) 006 [arXiv:1701.08784] [INSPIRE].

[21] S. Macaluso and D. Shih, *Pulling Out All the Tops with Computer Vision and Deep Learning*, *JHEP* **10** (2018) 121 [arXiv:1803.00107] [INSPIRE].

[22] F.A. Dreyer, G.P. Salam and G. Soyez, *The Lund Jet Plane*, *JHEP* **12** (2018) 064 [arXiv:1807.04758] [INSPIRE].

[23] 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] [INSPIRE].

[24] Y.-L. Du et al., *Identifying the nature of the QCD transition in relativistic collision of heavy nuclei with deep learning*, *Eur. Phys. J. C* **80** (2020) 516 [arXiv:1910.11530] [INSPIRE].

[25] J. Filipek et al., *Identifying the Quantum Properties of Hadronic Resonances using Machine Learning*, arXiv:2105.04582 [INSPIRE].

[26] CMS collaboration, *Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques*, 2020 *JINST* **15** P06005 [arXiv:2004.08262] [INSPIRE].

[27] L. de Oliveira et al., *Jet-images — deep learning edition*, *JHEP* **07** (2016) 069 [arXiv:1511.05190] [INSPIRE].

[28] J. Barnard, E.N. Dawe, M.J. Dolan and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks*, *Phys. Rev. D* **95** (2017) 014018 [arXiv:1609.00607] [INSPIRE].

[29] P.T. Komiske, E.M. Metodiev and M.D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, *JHEP* **01** (2017) 110 [arXiv:1612.01551] [INSPIRE].

[30] S. Choi, S.J. Lee and M. Perelstein, *Infrared Safety of a Neural-Net Top Tagging Algorithm*, *JHEP* **02** (2019) 132 [arXiv:1806.01263] [INSPIRE].

[31] J. Li, T. Li and F.-Z. Xu, *Reconstructing boosted Higgs jets from event image segmentation*, *JHEP* **04** (2021) 156 [arXiv:2008.13529] [INSPIRE].

[32] K. Fraser and M.D. Schwartz, *Jet Charge and Machine Learning*, *JHEP* **10** (2018) 093 [arXiv:1803.08066] [INSPIRE].

[33] D. Guest et al., *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev. D* **94** (2016) 112002 [arXiv:1607.08633] [INSPIRE].

[34] S. Egan et al., *Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC*, arXiv:1711.09059 [INSPIRE].

JHEP07(2024)247

[35] E. Bols et al., *Jet Flavour Classification Using DeepJet*, 2020 *JINST* **15** P12012 [arXiv:2008.10519] [INSPIRE].

[36] J. Pearkes, W. Fedorko, A. Lister and C. Gay, *Jet Constituents for Deep Neural Network Based Top Quark Tagging*, arXiv:1704.02124 [INSPIRE].

[37] A. Butter, G. Kasieczka, T. Plehn and M. Russell, *Deep-learned Top Tagging with a Lorentz Layer*, *SciPost Phys.* **5** (2018) 028 [arXiv:1707.08966] [INSPIRE].

[38] G. Kasieczka, N. Kiefer, T. Plehn and J.M. Thompson, *Quark-Gluon Tagging: Machine Learning vs Detector*, *SciPost Phys.* **6** (2019) 069 [arXiv:1812.09223] [INSPIRE].

[39] Y. Wang et al., *Dynamic Graph CNN for Learning on Point Clouds*, arXiv:1801.07829 [INSPIRE].

[40] M. He, Z. Wei and J.-R. Wen, *Convolutional Neural Networks on Graphs with Chebyshev Approximation, Revisited*, arXiv:2202.03580.

[41] L. Liao et al., *An improved dynamic Chebyshev graph convolution network for traffic flow prediction with spatial-temporal attention*, *Appl. Intell.* **52** (2022) 16104.

[42] O. Boyaci, M.R. Narimani, K. Davis and E. Serpedin, *Cyberattack Detection in Large-Scale Smart Grids using Chebyshev Graph Convolutional Networks*, arXiv:2112.13166.

[43] S. Maneewongvatana and D.M. Mount, *Analysis of approximate nearest neighbor searching with clustered point sets*, cs/9901013.

[44] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, in *International Conference on Learning Representations*, (2018), arXiv:1711.05101 [INSPIRE].

[45] T. Saito and M. Rehmsmeier, *The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets*, *PLoS One* **10** (2015) e0118432.

[46] CMS collaboration, *Boosted jet identification using particle candidates and deep neural networks*, CMS-DP-2017-049 (2017).

[47] M. Zaheer et al., *Deep Sets*, *Adv. Neural Inf. Process* **30** (2017) 7264 [arXiv:1703.06114] [INSPIRE].

[48] G. Kasieczka, T. Plehn and M. Russel, *Top quark tagging reference dataset*, DOI:10.5281/zenodo.2603256 (2019).

[49] P. Komiske, E. Metodiev and J. Thaler, *Pythia8 Quark and Gluon Jets for Energy Flow*, DOI:10.5281/zenodo.3164691 (2019).

JHEP07(2024)247