Particle Transformer for Jet Tagging

Huilin Qu 1  Congqiao Li 2  Sitian Qian 2

Abstract

Jet tagging is a critical yet challenging classification task in particle physics. While deep learning has transformed jet tagging and significantly improved performance, the lack of a large-scale public dataset impedes further enhancement. In this work, we present JETCLASS, a new comprehensive dataset for jet tagging. The JETCLASS dataset consists of 100 M jets, about two orders of magnitude larger than existing public datasets. A total of 10 types of jets are simulated, including several types unexplored for tagging so far. Based on the large dataset, we propose a new Transformer-based architecture for jet tagging, called Particle Transformer (ParT). By incorporating pairwise particle interactions in the attention mechanism, ParT achieves higher tagging performance than a plain Transformer and surpasses the previous state-of-the-art, ParticleNet, by a large margin. The pre-trained ParT models, once fine-tuned, also substantially enhance the performance on two widely adopted jet tagging benchmarks. The dataset, code and models are publicly available at https://github.com/jet-universe/particle_transformer.

1. Introduction

Machine learning has revolutionized how large-scale data samples are analyzed in particle physics and greatly increased the discovery potential for new fundamental laws of nature (Radovic et al., 2018). Specifically, deep learning has transformed how jet tagging, a critical classification task at high-energy particle colliders such as the CERN LHC, is performed, leading to a drastic improvement in its performance (Kogler et al., 2019; Larkoski et al., 2020).

At the CERN LHC, two beams of protons are accelerated to nearly the speed of light and made to collide at a frequency of 40 million times per second (40 MHz). Such high-energy collisions can create new unstable particles, which then decay and produce sprays of outgoing particles. Complex detector systems, such as the general-purpose ATLAS (ATLAS Collaboration, 2008) and CMS (CMS Collaboration, 2008) detectors with \(O(100 \text{ M})\) individual sensors of various types, are used to measure the positions, trajectories, energies, and momenta of the outgoing particles. From these measurements, an event is reconstructed for each collision. The primary goal in the analysis of the collision data is to identify events involving novel physics processes, an example of which is the discovery of the Higgs boson (ATLAS Collaboration, 2012; CMS Collaboration, 2012).

A crucial step in the data analysis process is jet tagging. A jet refers to a collimated spray of outgoing particles. Jet tagging is the process of identifying the type of particle that initiates a jet. It is essentially a classification task that aims to distinguish jets arising from particles of interest, such as the Higgs boson or the top quark, from other less interesting types of jets. Jet tagging is a challenging task because the particle initiating a jet can radiate, and the...
radiated particles further produce more particles, leading to a cascade of $O(10)$ to $O(100)$ particles at the end. The radiation also smears the characteristics of the initial particle and makes the identification very difficult.

Traditional approaches for jet tagging rely on hand-crafted features motivated by the principles of quantum chromodynamics (QCD), the theory governing the evolution of particles inside a jet. The rise of deep learning has led to a plethora of new approaches (Larkoski et al., 2020). The prevailing ones represent a jet as a particle cloud, i.e., an unordered and variable-sized set of the outgoing particles, as illustrated in Figure 1. Based on the particle cloud representation, ParticleNet (Qu & Gouskos, 2020) adapts the Dynamic Graph CNN architecture (Wang et al., 2019) and achieves substantial performance improvement on two representative jet tagging benchmarks. Since then, several new models (e.g., Mikuni & Canelli (2020; 2021); Shimmin (2021)) have been proposed, but no significant performance improvement has been reported so far. We deem the lack of a sufficiently large public dataset an impeding factor.

In this work, we advocate for JetCLASS, a new large and comprehensive dataset to advance deep learning for jet tagging. The JetCLASS dataset (Qu et al., 2022) consists of 100 M jets for training, about two orders of magnitude larger than existing public datasets. It also includes more types of jets, several of which have not been explored for tagging yet but are promising for future applications at the LHC.

Based on this dataset, we propose Particle Transformer (ParT), a new Transformer-based architecture for jet tagging. We demonstrate that Transformer architectures, together with a large dataset, can reach powerful performance on jet tagging. We introduce a small modification to the attention mechanism by incorporating a new term characterizing pairwise particle interactions. The resulting ParT achieves significantly higher performance than a plain Transformer and surpasses the previous state-of-the-art, ParticleNet, by a large margin. We also apply the pre-trained ParT models to two widely adopted jet tagging benchmarks with fine-tuning and observe a substantial gain on these tasks.

2. The JetCLASS Dataset

We provide an overview of the new JetCLASS dataset in this section. The dataset includes a total of 10 types of jets. Representative jets of each type are visualized as particle clouds in Figure 2. The jets in this dataset generally fall into two categories. The background jets are initiated by light quarks or gluons ($q/g$) and are ubiquitously produced at the LHC. The signal jets are those arising either from the top quarks ($t$), or from the $W$, $Z$ or Higgs ($H$) bosons. For top quarks and Higgs bosons, we further consider their different decay modes as separate types, because the resulting jets have rather distinct characteristics and are often tagged individually. The use of jet tagging typically involves selecting one (or a few) specific type of signal jets with high confidence, and rejecting background jets as much as possible, since the background jets usually appear orders of magnitude more frequently than the targeted signal jets. Note that for several types of signal jets in this dataset, such as $H \rightarrow 4q$, $t \rightarrow \ell\nuqq'$, and $t \rightarrow b\ell\nu$, no dedicated methods have been developed so far to tag them. However, as we will demonstrate in Section 5.1, these types of jets can also be cleanly tagged with deep learning approaches, opening
Particle Transformer for Jet Tagging

up new possible territories for jet tagging at the LHC.

**Simulation setup.** Jets in this dataset are simulated with standard Monte Carlo event generators used by LHC experiments. The production and decay of the top quarks and the $W$, $Z$ and Higgs bosons are generated with MAdGraph5_@NLO (Alwall et al., 2014). We use PYTHIA (Sjöstrand et al., 2015) to evolve the produced particles, i.e., performing parton showering and hadronization, and produce the final outgoing particles\(^1\). To be close to realistic jets reconstructed at the ATLAS or CMS experiment, detector effects are simulated with DELPHES (de Favereau et al., 2014) using the CMS detector configuration provided in DELPHES. In addition, the impact parameters of electrically charged particles are smeared to match the resolution of the CMS tracking detector (CMS Collaboration, 2014). Jets are clustered from DELPHES E-Flow objects with the anti-$k_T$ algorithm (Cacciari et al., 2008; 2012) using a distance parameter $R = 0.8$. Only jets with transverse momentum in 500–1000 GeV and pseudorapidity $|\eta| < 2$ are considered. For signal jets, only the “high-quality” ones that fully contain the decay products of initial particles are included\(^2\).

**Input features.** The dataset provides all constituent particles of each jet as inputs for jet tagging. Note that the number of particles varies from jet to jet, typically between 10 and 100, with an average of 30–50 depending on the jet type. For each particle of a jet, three categories of features are provided:

- **Kinematics.** This includes the energy and momentum, described by the 4-vector $(E, p_x, p_y, p_z)$ in units of GeV, which are the most fundamental quantities measured by a particle detector. All other kinematic variables can be computed from the 4-vectors.

- **Particle identification.** This includes the electric charge, with values of $\pm 1$ (positively/negatively charged particles) and 0 (neutral particles), and the particle identity determined by the detector systems. For the latter, a 5-class one-hot encoding is used to be consistent with current LHC experiments: charged hadron ($\pm 211$, $\pm 321$, $\pm 2211$), neutral hadron (0), electron (±11), muon (±13), and photon (22). The particle identification information is especially important for tagging jets involving a charged lepton, e.g., $H \rightarrow \ell qq'$ and $t \rightarrow bl\ell$, as leptons can be almost unambiguously identified at the LHC.

- **Trajectory displacement.** This includes the measured values and errors of the transverse and longitudinal impact parameters of the particle trajectories in units of mm, in total 4 variables. These measurements are only available for electrically charged particles, and a value of 0 is used for neutral particles. The trajectory displacement information is critical for tagging jets involving a bottom ($b$) or charm ($c$) quark (CMS Collaboration, 2020b), such as $H \rightarrow bb$, $H \rightarrow cc$, $t \rightarrow bq\ell$, etc., but is missing from most of the existing datasets.

**Training, validation and test sets.** The training set consists of 100 M jets in total, equally distributed in the 10 classes. An additional set of 500 k jets per class (in total 5 M) is intended for model validation. For the evaluation of performance, a separate test set with 2 M jets in each class (in total 20 M) is provided.

**Evaluation metrics.** To thoroughly evaluate the performance of deep learning models on this dataset, we advocate for a series of metrics. Since jet tagging on this dataset is naturally framed as a multi-class classification task, two common metrics, i.e., the accuracy and the area under the ROC curve (AUC)\(^3\) are adopted to quantify the overall performance. In addition, we propose the background rejection (i.e., the inverse of the false positive rate) at a certain signal efficiency (i.e., the true positive rate, TPR) of $X\%$, i.e.,

$$\text{Rej}_{X\%} \equiv 1/\text{FPR at TPR} = X\%,$$

for each type of signal jets. By default, the $q/g$ jets are considered as the background, as is the case in most LHC data analyses, and each of the other 9 types of jets can be considered as the signal. The signal efficiency (TPR) for each signal type is chosen to be representative of actual usages at the LHC experiments and is typically 50%. It is increased to 99% (99.5%) for $H \rightarrow \ell qq'$ ($t \rightarrow bl\ell$), as these types of jets have more distinct characteristics and can be more easily separated from $q/g$ jets. Since the definition of the Rej\(_X\) metric involves only two classes, i.e., the signal class under consideration ($S$) and the background class ($B$), the TPR and FPR are evaluated using a two-class score,

$$\text{score}_{SvsB} = \frac{\text{score}(S)}{\text{score}(S) + \text{score}(B)},$$

where score($S$) and score($B$) are the softmax outputs for class $S$ and $B$, respectively, to achieve optimal performance for $S$ vs $B$ separation. This is aligned with the convention adopted by the CMS experiment (CMS Collaboration, 2020b). Note that the background rejection metric, although rarely used in vision or language tasks, is actually a standard metric for jet tagging because it is directly related to the discovery potential at the LHC experiments. A factor

\(^1\)We include multiple parton interactions but omit pileup interactions in the simulation.

\(^2\)We require all the quarks ($q$) and charged leptons (electrons or muons, denoted $\ell$) from the decay of the top quark or the $W$, $Z$ or Higgs boson satisfy $\Delta R(\text{jet}, q/\ell) < 0.8$, where $\Delta R(a, b) \equiv \sqrt{(\eta_a - \eta_b)^2 + (\phi_a - \phi_b)^2}$, in which $\eta$ ($\phi$) is the pseudorapidity (azimuthal angle) of the momentum of the jet or the particle.

\(^3\)The AUC can be calculated using roc_auc_score in scikit-learn with average='macro' and multi_class='ovo'.
of two increase in background rejection can lead to about 40% increase in the discovery potential, which would otherwise require a dataset of twice the size, or in other words, doubling the running time of the LHC.

3. Related Work

Jet tagging with deep learning. Deep learning approaches have been proposed extensively to improve jet tagging. Previous models handle jets with different representations, e.g., images (de Oliveira et al., 2016), sequences (Guest et al., 2016), trees (Louppe et al., 2019), graphs (Henrion et al., 2017), with corresponding deep learning architectures such as 2D CNNs, recurrent or recursive networks, and graph neural networks. More recently, the particle cloud representation (Komiske et al., 2019b; Qu & Gouskos, 2020), analogous to point clouds, which treats a jet as a permutation-invariant set of particles as visualized in Figure 2, has been proposed. The Deep Sets (Zaheer et al., 2017) and Dynamic Graph CNN (Wang et al., 2019) architectures are adapted for jet tagging, resulting in the Energy Flow Network (Komiske et al., 2019b) and the state-of-the-art, ParticleNet (Qu & Gouskos, 2020), respectively. Since then, particle clouds have become the prevailing representation of jets and more architectures based on GAPNet (Chen et al., 2021; Mikuni & Canelli, 2020), the Point Cloud Transformer (Guo et al., 2021; Mikuni & Canelli, 2021) have been studied, but no significant performance improvement over ParticleNet has been reported. Lately, researches have been focused more on incorporating inductive biases from physics principles in the architecture design, such as the usage of the Lund jet plane (Dreyer et al., 2018; Dreyer & Qu, 2021; Dreyer et al., 2021; 2022), the Lorentz group symmetry (Bogatskiy et al., 2020; Gong et al., 2022), and the rotational symmetry (Shimmin, 2021; Dillon et al., 2021).

Deep-learning-based jet tagging algorithms have been widely adopted in real-world data analysis at the LHC. For example, the CMS Collaboration develops the DeepAK8 (CMS Collaboration, 2020b) algorithm to tag jets arising from the top quark or the Higgs, W, or Z boson, using a 1D CNN following the ResNet (He et al., 2016) architecture, and a significant increase in the discovery potential for new heavy particles has been achieved (CMS Collaboration, 2021; 2022a). Moreover, using ParticleNet, CMS achieves the first observation of Z boson decay to a pair of charm quarks at a hadron collider and obtains the most stringent constraint on $H \rightarrow c\bar{c}$ decay (CMS Collaboration, 2022c). ParticleNet is also used by CMS to probe the quartic interaction between the Higgs and vector bosons, indirectly confirming its existence for the first time (CMS Collaboration, 2022b). Clearly, advances in jet tagging play a vital role in accelerating our understanding of elementary particles, the fundamental building blocks of nature.

Jet tagging datasets. A number of datasets have been published so far to study jet tagging:

- **Top quark tagging dataset** (Kasieczka et al., 2019) proposed in Butter et al. (2019), consisting of 2 M jets in 2 types ($t \rightarrow bqq'$ and $q/g$) and providing only the kinematic information.
- **Quark-gluon tagging dataset** (Komiske et al., 2019a) proposed in Komiske et al. (2019b), consisting of 2 M jets in 2 types (quark and gluon), and providing both the kinematic and particle identification information.
- **Higgs boson tagging dataset** (Duarte, 2019; Chen et al., 2022), containing 3.9 M $H \rightarrow bb$ jets and 1.9 M $q/g$ jets, with all three categories of information.
- **JetNet dataset** (Kansal et al., 2021b) proposed in Kansal et al. (2021a), containing $\approx$500 k jets in 3 types: gluon, light quark, and top quark, and providing only the kinematic information.
- **A multiclass dataset** (Pierini et al., 2020) proposed in Moreno et al. (2020), with 880 k jets in 5 classes: light quark, gluon, $W$ boson, $Z$ boson and top quark and providing only the kinematic information.

Compared with existing datasets, the JETCLASS dataset is not only substantially larger in size, but also more inclusive in terms of the types of jets contained.

Transformers. Recent years have witnessed the enormous success of Transformer models. Starting from natural language processing and then spreading to computer vision, the original Transformer (Vaswani et al., 2017), as well as its variants, e.g., BERT (Devlin et al., 2019), ViT (Dosovitskiy et al., 2020), have refreshed the performance records in various tasks, demonstrating the power of Transformer as a universal architecture. Transformers, and the attention mechanism at its core, have proved to be powerful for fundamental scientific problems as well. For example, AlphaFold2 (Jumper et al., 2021), which reaches the state-of-the-art performance in protein structure prediction, employs the attention mechanism. In particular, adding a pair bias, derived from pairwise features, to the self attention helps improve the model explainability.

4. Model Architecture

Together with the JETCLASS dataset, we propose the Particle Transformer (ParT) as a new baseline for jet tagging. An overview of the ParT architecture is presented in Figure 3(a). For a jet with $N$ particles, ParT makes use of two sets of inputs: the particle input includes a list of $C$ features for every particle and forms an array of a shape ($N, C$); the interaction input is a matrix of $C'$ features for every pair

\footnote{The term \textit{interaction} here refers to any feature involving a pair of particles, which may or may not be related to the physical forces between them.}
of particles, in a shape \((N, N, C')\). The particle and interaction inputs are each followed by an MLP to project them to a \(d\)- and \(d'\)-dimensional embedding, \(x^0 \in \mathbb{R}^{N \times d} \) and \(U \in \mathbb{R}^{N \times N \times d'}\), respectively. Unlike Transformers for NLP and vision, we do not add any ad-hoc positional encodings, as the particles in a jet are permutation invariant. The spatial information (i.e., the flying direction of each particle) is directly included in the particle inputs. We feed the particle embedding \(x^0\) into a stack of \(L\) particle attention blocks to produce new embeddings, \(x^1, \ldots, x^L\) via multi-head self attention. The interaction matrix \(U\) is used to augment the scaled dot-product attention by adding it as a bias to the pre-softmax attention weights. The same \(U\) is used for all the particle attention blocks. After that, the last particle embedding \(x^L\) is fed into two action blocks, and a global class token \(x'\)class\) is used to extract information for jet classification via attention to all the particles, following the CaiT approach (Touvron et al., 2021). The class token is passed to a single-layer MLP, followed by softmax, to produce the final classification scores.

**Remark.** ParT can also be viewed as a graph neural network on a fully-connected graph, in which each node corresponds to a particle, and the interactions are the edge features.

**Particle interaction features.** While the ParT architecture is designed to be able to process any kinds of pairwise interaction features, for this paper we only consider a specific scenario in which the interaction features are derived from the energy-momentum 4-vector, \(p = (E, p_x, p_y, p_z)\), of each particle. This is the most general case for jet tagging, as the particle 4-vectors are available in every jet tagging task. Specifically, for a pair of particles \(a, b\) with 4-vectors \(p_a, p_b\), we calculate the following 4 features:

\[
\begin{align*}
\Delta &= \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}, \\
k_T &= \min(\|p_{T,a} + p_{T,b}\|, \|p_{T,a} - p_{T,b}\|), \\
z &= \min(\|p_{T,a} + p_{T,b}\|, \|p_{T,a} - p_{T,b}\|), \\
m^2 &= (E_a + E_b)^2 - \|p_a + p_b\|^2, \\
\end{align*}
\]

where \(y_i\) is the rapidity, \(\phi_i\) is the azimuthal angle, \(p_{T,i} = (p_{T,i}^2, p_{T,i}^2)^{1/2}\) is the transverse momentum, and \(p_i = (p_{x,i}, p_{y,i}, p_{z,i})\) is the momentum 3-vector and \(\|\cdot\|\) is the norm, for \(i = a, b\). Since these variables typically have a long-tail distribution, we take the logarithm and use \((\ln \Delta, \ln k_T, \ln z, \ln m^2)\) as the interaction features for each particle pair. The choice of this set of features is motivated by Dreyer & Qu (2021).

**Particle attention block.** A key component of ParT is the particle attention block. As illustrated in Figure 3(b), the particle attention block consists of two stages. The first stage includes a multi-head attention (MHA) module with a LayerNorm (LN) layer both before and afterwards. The
second stage is a 2-layer MLP, with an LN before each linear layer and GELU nonlinearity in between. Residual connections are added after each stage. The overall block structure is based on NormFormer (Shleifer et al., 2021), however, we replace the standard MHA with P-MHA, an augmented version that can also exploit the pairwise particle interactions directly. The P-MHA is computed as

\[
P-MHA(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d_k} + U)V, \tag{4}
\]

where \(Q\) and \(V\) are linear projections of the particle embedding \(x^l\). Essentially, we add the interaction matrix \(U\) to the pre-softmax attention weights. This allows P-MHA to incorporate particle interaction features designed from physics principles and modify the dot-product attention weights, thus increasing the expressiveness of the attention mechanism.

**Class attention block.** As illustrated in Figure 3(c), the class attention block has a similar structure as the particle attention block. However, unlike in the particle attention block where we compute the self attention between particles, here we compute the attention between a global class token \(x_{\text{class}}\) and all the particles using the standard MHA. Specifically, the inputs to the MHA are

\[
Q = W_q x_{\text{class}} + b_q, \\
K = W_k z + b_k, \\
V = W_v z + b_v,
\]

where \(z = [x_{\text{class}}, x^L]\) is the concatenation of the class token and the particle embedding after the last particle attention block, \(x^L\).

**Implementation.** We implement the ParT model in PyTorch (Paszke et al., 2019). Specifically, the P-MHA is implemented using the PyTorch’s MultiheadAttention by providing the interaction matrix \(U\) as the attn_mask input. The baseline ParT model has a total of \(L = 8\) particle attention blocks and 2 class attention blocks. It uses a particle embedding of a dimension \(d = 128\), encoded from the input particle features using a 3-layer MLP with \((128, 512, 128)\) nodes each layer with GELU nonlinearity, and LN is used in between for normalization. The interaction input features are encoded using a 4-layer pointwise 1D convolution with \((64, 64, 64, 16)\) channels with GELU nonlinearity and batch normalization in between to yield a \(d' = 16\) dimensional interaction matrix. The P-MHA (MHA) in the particle (class) attention blocks all have 8 heads, with a query dimension \(d' = 16\) for each head, and an expansion factor of 4 for the MLP. We use a dropout of 0.1 for all particle attention blocks, and no dropout for the class attention block. The choice of hyperparameters provides a reasonable baseline but is not extensively optimized.

**5. Experiments**

We conduct experiments on the new JETCLASS dataset and show the results in Section 5.1. The pre-trained ParT models are also applied to two existing datasets with fine-tuning, and the performance is compared to previous state-of-the-arts in Section 5.2.

**5.1. Experiments on JETCLASS Dataset**

**Setup.** For experiments on the JETCLASS dataset, we use the full set of particle features, including kinematics, particle identification, and trajectory displacement, as inputs. The full list of 17 features for each particle is summarized in Table 2. In addition, the 4 interaction features introduced in Equation (3) are also used for the ParT model. The training is performed on the full training set of 100M jets. We employ the Lookahead optimizer (Zhang et al., 2019) with \(k = 6\) and \(\alpha = 0.5\) to minimize the cross-entropy loss, and the inner optimizer is RAdam (Liu et al., 2020) with \(\beta_1 = 0.95, \beta_2 = 0.999,\) and \(\epsilon = 10^{-5}\). A batch size of 512 and an initial learning rate (LR) of 0.001 are used. No weight decay is applied. We train for a total of 1M iterations, amounting to around 5 epochs over the full training set. The LR remains constant for the first 70% of the iterations, and then decays exponentially, at an interval of every 20k iterations, down to 1% of the initial value at the end of the training. Performance of the model is evaluated every 20k iterations on the validation set and a model checkpoint is saved. The checkpoint with the highest accuracy on the validation set is used to evaluate the final performance on the test set.

**Baselines.** We compare the performance of ParT with 3 baseline models: the PFN (Komiske et al., 2019b) architecture based on Deep Sets (Zaheer et al., 2017), the P-CNN architecture used by the DeepAK8 algorithm of the CMS experiment (CMS Collaboration, 2020b), and the state-of-the-art ParticleNet architecture (Qu & Gouskos, 2020) adapted from DGCNN (Wang et al., 2019). All the models are trained end-to-end on the JETCLASS dataset for the same number of effective epochs for a direct comparison. For ParticleNet, we directly use the existing PyTorch implementation. For PFN and P-CNN, we re-implement them in PyTorch and verify that the published results are reproduced. The optimizer and LR schedule remain the same as in the training of ParT. The (batch size, LR) combination is re-optimized and chosen to be \((512, 0.01)\) for ParticleNet and \((4096, 0.02)\) for PFN and P-CNN.

**Results.** Performance on the JETCLASS dataset is evaluated using the metrics described in Section 2, and the results are summarized in Table 1. The proposed ParT architecture achieves the best performance on every metric, and outperforms the existing state-of-the-art, ParticleNet, by a large margin. The overall accuracy is increased by 1.7% com-
Table 1. Jet tagging performance on the JETCLASS dataset. ParT is compared to PFN (Komiske et al., 2019b), P-CNN (CMS Collaboration, 2020b) and the state-of-the-art ParticleNet (Qu & Gouskos, 2020). For all the metrics, a higher value indicates better performance. The ParT architecture using plain MHAs instead of P-MHAs, labelled as ParT (plain), is also shown for comparison.

<table>
<thead>
<tr>
<th>Category</th>
<th>Variable</th>
<th>Definition</th>
<th>JETCLASS</th>
<th>Top</th>
<th>QGexp</th>
<th>QGfull</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kinematics</strong></td>
<td>Δη</td>
<td>difference in pseudorapidity η between the particle and the jet axis</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Δφ</td>
<td>difference in azimuthal angle φ between the particle and the jet axis</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>log PT</td>
<td>logarithm of the particle’s transverse momentum pT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>log E</td>
<td>logarithm of the particle’s energy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>log PT/PT</td>
<td>logarithm of the particle’s pT relative to the jet pT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>log E/E</td>
<td>logarithm of the particle’s energy relative to the jet energy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>ΔR</td>
<td>angular separation between the particle and the jet axis ((\sqrt{(Δη)^2 + (Δφ)^2}))</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Particle</strong></td>
<td>charge</td>
<td>electric charge of the particle</td>
<td>✓</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>identification</td>
<td>electron</td>
<td>if the particle is an electron (</td>
<td>pid</td>
<td>==11)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>muon</td>
<td>if the particle is a muon (</td>
<td>pid</td>
<td>==13)</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>photon</td>
<td>if the particle is a photon (</td>
<td>pid</td>
<td>==22)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>CH</td>
<td>if the particle is a charged hadron (</td>
<td>pid</td>
<td>==211 or 321 or 2212)</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>NH</td>
<td>if the particle is a neutral hadron (</td>
<td>pid</td>
<td>==130 or 2112 or 0)</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Trajectory</strong></td>
<td>tanh d₀</td>
<td>hyperbolic tangent of the transverse impact parameter value</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>displacement</td>
<td>tanh dₜ</td>
<td>hyperbolic tangent of the longitudinal impact parameter value</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>σd₀</td>
<td>error of the measured transverse impact parameter</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>σdₜ</td>
<td>error of the measured longitudinal impact parameter</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

\(\text{a} \cdot |\text{pid}|=211) + \sum |\text{pid}|=321\cdot 0.5 + \sum |\text{pid}|=2212\cdot 0.2

\(\text{b} \cdot |\text{pid}|=130) + \sum |\text{pid}|=2112\cdot 0.2

pared to ParticleNet. Moreover, for the physics-oriented metric, the background rejection, ParT improves over ParticleNet by a factor of 3 for \(t \rightarrow bqq'\), a factor of 2 for \(H \rightarrow 4q\), and about 70% for \(H \rightarrow cc\). It is also clear that, the earlier PFN and P-CNN models lag substantially behind ParticleNet and ParT on this large dataset, amounting to up to an order of magnitude difference in background rejection. The large improvement of ParT is likely to lead to a significant jump in the discovery potential for related physics searches at the LHC.

Another observation is that there is a large variation in tagging performance between signals of different types. The best separation against the background \(q/g\) jets is achieved for \(t \rightarrow bν\) and \(H \rightarrow tqq'\) signals – with the powerful ParT model, these two can be selected almost perfectly, i.e., at an efficiency of more than 99% with nearly no contamination from background jets. This opens up new territory for jet tagging at the LHC, as these types of jets have not been exploited for tagging so far.

**Effectiveness of P-MHA.** To quantify the effectiveness of the P-MHA introduced in ParT, we carry out an ablation study by replacing the P-MHA with a standard MHA, the resulting architecture is then a plain Transformer and therefore denoted as ParT (plain). We train ParT (plain) with the same procedure as the full ParT and the performance is shown in Table 1. A drop of 1.2% in accuracy is observed compared to the full ParT and the background rejection is reduced by 20–30% for most signals. Note that, replacing P-MHA with plain MHA implies that the particle interaction input is discarded completely, but this does not lead to any reduction of information content, as the interaction features defined in Equation (3) are derived purely from the energy-momentum 4-vectors, which are already used as particle features via the 7 kinematic variables presented in Table 2. Therefore, the improvement of ParT over a plain Transformer indeed arise from an efficient exploitation of the particle kinematic information using the P-MHA.
Table 3. Impacts of the training dataset size. Entries in bold correspond to the training using the full 100 M training dataset.

<table>
<thead>
<tr>
<th></th>
<th>All classes</th>
<th>H → b+b</th>
<th>H → c+c</th>
<th>H → g+g</th>
<th>H → t+tq′</th>
<th>t → bq′</th>
<th>t → b+bτ</th>
<th>W → q+q′</th>
<th>Z → q+q′</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>AUC</td>
<td>Rej_{30%}</td>
<td>Rej_{50%}</td>
<td>Rej_{90%}</td>
<td>Rej_{30%}</td>
<td>Rej_{50%}</td>
<td>Rej_{90%}</td>
<td>Rej_{30%}</td>
</tr>
<tr>
<td>ParticleNet (2 M)</td>
<td>0.828</td>
<td>0.9825</td>
<td>5540</td>
<td>1681</td>
<td>90</td>
<td>662</td>
<td>1654</td>
<td>4049</td>
<td>4673</td>
</tr>
<tr>
<td>ParticleNet (10 M)</td>
<td>0.837</td>
<td>0.9837</td>
<td>5848</td>
<td>2070</td>
<td>96</td>
<td>770</td>
<td>2350</td>
<td>5495</td>
<td>6803</td>
</tr>
<tr>
<td>ParticleNet (100 M)</td>
<td>0.844</td>
<td>0.9849</td>
<td>7634</td>
<td>2475</td>
<td>104</td>
<td>954</td>
<td>3339</td>
<td>10526</td>
<td>11173</td>
</tr>
<tr>
<td>ParT (2 M)</td>
<td>0.836</td>
<td>0.9834</td>
<td>5587</td>
<td>1982</td>
<td>93</td>
<td>761</td>
<td>1609</td>
<td>6061</td>
<td>4474</td>
</tr>
<tr>
<td>ParT (10 M)</td>
<td>0.850</td>
<td>0.9860</td>
<td>8734</td>
<td>3040</td>
<td>110</td>
<td>1274</td>
<td>3257</td>
<td>12579</td>
<td>8969</td>
</tr>
<tr>
<td>ParT (100 M)</td>
<td>0.861</td>
<td>0.9877</td>
<td>10638</td>
<td>4149</td>
<td>123</td>
<td>1864</td>
<td>5479</td>
<td>32787</td>
<td>15873</td>
</tr>
</tbody>
</table>

Table 4. Number of trainable parameters and FLOPs.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th># params</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFN</td>
<td>0.772</td>
<td>86.1 k</td>
<td>4.62 M</td>
</tr>
<tr>
<td>P-CNN</td>
<td>0.809</td>
<td>354 k</td>
<td>15.5 M</td>
</tr>
<tr>
<td>ParticleNet</td>
<td>0.844</td>
<td>370 k</td>
<td>540 M</td>
</tr>
<tr>
<td>ParT</td>
<td>0.861</td>
<td>2.14 M</td>
<td>340 M</td>
</tr>
<tr>
<td>ParT (plain)</td>
<td>0.849</td>
<td>2.13 M</td>
<td>260 M</td>
</tr>
</tbody>
</table>

Impacts of the training dataset size. To evaluate the impacts of the training dataset size on the jet tagging performance, we perform additional trainings using only 2% and 10% of the JETCLASS dataset. For the former, the training is performed for only 100 k iterations, as it is already converged by then. For the latter, the training still lasts for 1 M iterations, although very little gain is observed compared to the training with only 100 k iterations. No overfitting is found in either case. The results are summarized in Table 3. For the ParticleNet model, a drop of 0.7% in accuracy is observed when the training dataset size is reduced to 10 M, and the drop in accuracy increases to 1.6% when only 2 M jets are used in the training. For the ParT model, the impact is even larger, the degradation in accuracy becomes 1.1% and 2.5% when the training dataset is reduced to 10% and 2%, respectively.

Model complexity. Table 4 compares the model complexity of ParT with the baselines. While the number of trainable parameters is increased by more than 5× compared to ParticleNet, the number of floating point operations (FLOPs) is actually 40% lower. We also observe that the FLOPs of ParT are 30% higher than ParT (plain), which mostly comes from the encoding of the pairwise features, because the computational cost there scales quadratically with the number of particles in a jet.

5.2. Fine-Tuning for Other Datasets

Top quark tagging dataset. The top quark tagging benchmark (Butter et al., 2019) provides a dataset of 2 M (1.2/0.4/0.4 M for train/validation/test) jets in two classes, t → bq′ (signal) and q/g (background). Only kinematic features, i.e., the energy-momentum 4-vectors, are provided. Therefore, we pre-train a ParT model on the JETCLASS dataset using only the kinematic features, and then fine-tune it on the top quark tagging dataset. The particle input features are the 7 kinematic features listed in Table 2, the same as used by ParticleNet. The JETCLASS pre-training follows the same setup as described in Section 5.1. For the fine-tuning, we replace the last MLP with a new randomly-initialized MLP with 2 output nodes, and then fine-tune all the weights on the top tagging dataset for 20 epochs. A smaller LR of 0.0001 is used for the pre-trained weights, while a larger LR of 0.005 is used to update the randomly-initialized weights of the MLP. The LR remains constant across the full training, with a weight decay of 0.01. We run a total of 9 experiments, starting from the same pre-trained model but different random initializations of the replaced MLP and report the performance of the model with median accuracy and the spread across the 9 trainings, following the procedure used by ParticleNet. For comparison, we also train ParT from scratch on this dataset for 20 epochs, using a start LR of 0.001, a schedule that decays the LR to 1% in the last 30% of the epochs, and a weight decay of 0.01. Both results are presented in Table 5. The pre-trained ParT achieves a significant improvement over the existing baselines, increasing Rej_{50%} by 70% compared to ParticleNet, and by 26% compared to the best-performing model on this dataset, LorentzNet. On the other hand, the ParT model trained from scratch only reaches similar performance as ParticleNet. We also investigate a similar pre-training and fine-tuning procedure using the ParticleNet model, but only a small improvement is observed compared to the training from scratch, due to the limited capacity of the ParticleNet model.

Quark-gluon tagging dataset. We also benchmark ParT on the quark-gluon tagging dataset (Komiske et al., 2019a) proposed in Komiske et al. (2019b), the target of which is to separate jets initiated by quarks (signal) from those by gluons (background). This dataset also consists of 2 M jets, with a recommended train/validation/test splitting of 1.6/0.2/0.2 M. It provides not only the kinematic features, but also particle identification information. We consider two scenarios in the usage of the particle identification information. In the “exp” scenario, we restrict the information to charged (and neural) hadrons of different types, which is the procedure adopted by ParticleNet, and also prescribed by
Table 5. Comparison between ParT and existing models on the top quark tagging dataset. ParT refers to the model trained from scratch on this dataset. ParticleNet-f.t. and ParT-f.t. denote the corresponding models pre-trained on JETCLASS and fine-tuned on this dataset. Results for other models are quoted from their published results: P-CNN and ParticleNet (Qu & Gouskos, 2020), PFN (Komiske et al., 2019b), JEDI-net (Moreno et al., 2020), PCT (Mikuni & Canelli, 2021), LGN (Bogatskiy et al., 2020), rPCN (Shimmin, 2021), and LorentzNet (Gong et al., 2022).

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>AUC</th>
<th>Rej$_{50%}$</th>
<th>Rej$_{70%}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>0.930</td>
<td>0.9803</td>
<td>201 ± 4</td>
<td>759 ± 24</td>
</tr>
<tr>
<td>PFN</td>
<td>—</td>
<td>0.9819</td>
<td>247 ± 3</td>
<td>888 ± 17</td>
</tr>
<tr>
<td>ParticleNet</td>
<td>0.940</td>
<td>0.9858</td>
<td>397 ± 7</td>
<td>1615 ± 93</td>
</tr>
<tr>
<td>JEDI-net (w/ $\sum O$)</td>
<td>0.930</td>
<td>0.9807</td>
<td>—</td>
<td>774.6</td>
</tr>
<tr>
<td>PCT</td>
<td>0.940</td>
<td>0.9855</td>
<td>392 ± 7</td>
<td>1533 ± 101</td>
</tr>
<tr>
<td>LGN</td>
<td>0.929</td>
<td>0.964</td>
<td>—</td>
<td>435 ± 95</td>
</tr>
<tr>
<td>rPCN</td>
<td>—</td>
<td>0.9845</td>
<td>364 ± 9</td>
<td>1642 ± 93</td>
</tr>
<tr>
<td>LorentzNet</td>
<td>0.942</td>
<td>0.9868</td>
<td>498 ± 18</td>
<td>2195 ± 173</td>
</tr>
<tr>
<td>ParT</td>
<td>0.940</td>
<td>0.9858</td>
<td>413 ± 16</td>
<td>1602 ± 81</td>
</tr>
<tr>
<td>ParticleNet-f.t.</td>
<td>0.942</td>
<td>0.9866</td>
<td>487 ± 9</td>
<td>1771 ± 80</td>
</tr>
<tr>
<td>ParT-f.t.</td>
<td>0.944</td>
<td>0.9877</td>
<td>691 ± 15</td>
<td>2766 ± 130</td>
</tr>
</tbody>
</table>

The JETCLASS dataset. In the “full” scenario, we consider all particle types and further distinguish electrically charged (and neutral) hadrons into more types, such as pions, kaons, and protons. We perform the pre-training on JETCLASS using only kinematic and particle identification inputs under the “exp” scenario. For the fine-tuning, we then carry out experiments in both scenarios. The construction of the input features is described in Table 2. The pre-training and fine-tuning setup is the same as in the top quark tagging benchmark, and the fine-tuning also lasts for 20 epochs. Results are summarized in Table 6. The pre-trained ParT achieves the best performance and improves existing baselines by a large margin in both scenarios.

6. Discussion and Conclusion

Large-scale datasets have always been a catalyst for new breakthroughs in deep learning. In this work, we present JETCLASS, a new large-scale open dataset to advance deep learning research in particle physics. The dataset consists of 100 M simulated jets, about two orders of magnitude larger than existing public jet datasets, and covers a broad spectrum of 10 classes of jets in total, including several novel types that have not been studied with deep learning so far. While we focus on investigating a classification task, i.e., jet tagging, with this dataset, we highlight that this dataset can serve as the basis for many important deep learning researches in particle physics, e.g., unsupervised or self-supervised training techniques for particle physics (e.g., Dillon et al. (2021)), generative models for high-fidelity fast simulation of particle collisions (e.g., Kansal et al. (2021a)), regression models to predict jet energy and momentum with higher precision (e.g., CMS Collaboration (2020a)), and more. We invite the community to explore and experiment with this dataset and extend the boundary of deep learning and particle physics even further.

With this large dataset, we introduce Particle Transformer (ParT), a new architecture that substantially improves jet tagging performance over previous state-of-the-art. We propose it as a new jet tagging baseline for future research to improve upon. The effectiveness of ParT arises mainly from the augmented self-attention, in which we incorporate physics-inspired pairwise interactions together with the machine-learned dot-product attention. This approach is likely to be effective for other tasks on similar datasets, such as point clouds or many-body systems, especially when prior knowledge is available to describe the interaction or the geometry. On the other hand, one limitation of using the full pairwise interaction matrix is the increase in computational time and memory consumption. Novel approaches for particle (point) embeddings and self-attentions that alleviate the computational cost (e.g., Zhou et al. (2021); Kitaev et al. (2020)) could be an interesting direction for future research.

Acknowledgements

We are grateful to Loukas Gouskos, Qiang Li, and Alexandre De Moor for many helpful discussions. The work of C. Li and S. Qian is supported by National Natural Science Foundation of China under Grants No. 12061141002.

References

Alwall, J., Frederix, R., Frixione, S., Hirschi, V., Maltoni, F., Mattelaer, O., Shao, H. S., Stelzer, T., Torrielli, P., and
Particle Transformer for Jet Tagging


CMS Collaboration. Search for resonances decaying to three W bosons in proton-proton collisions at √s = 13 TeV. 2022a. Accepted for publication in *Phys. Rev. Lett.*


Duarte, J. Sample with jet, track and secondary vertex properties for Hbb tagging ML studies HiggsToBBN-TupleHiggsToBBQCD_RunII_13TeV_MC, 2019. URL http://opendata.cern.ch/record/12102.


