PARTICLEnET: 
Jet Tagging via Particle Clouds

Based on work with Loukas Gouskos
arXiv:1902.08570

Code available at https://github.com/hqucms/ParticleNet

Huilin Qu
UC Santa Barbara
July 23, 2019
INTRODUCTION

- Jet is one of the most ubiquitous objects at the LHC
  - probably also the most diverse and therefore fascinating one...
- Jet tagging: powerful tool for many new physics searches and standard model measurements
  - heavy flavour tagging (bottom/charm)
  - heavy resonance tagging (top/W/Z/Higgs)
  - quark/gluon tagging
  - ...
- The rise of machine learning (ML) has brought lots of new progresses to jet tagging
  - many new approaches and techniques has been proposed in the past few years, leading to significant improvement in performance and deeper insights into jet physics

Still: How to better represent a jet for ML?
As images...

250 < p_T/GeV < 260 GeV, 65 < mass/GeV < 95

Pythia 8, W, W2, f = 13 TeV

Herwig (dipole) - PYTHIA

ATLAS Simulation Preliminary
Gluon Jets, Topcluster Constituents
anti k_t, R = 0.4, 150 < p_T/GeV < 200

QCD jets, averaged primary Lund plane
√s = 14 TeV, p_T > 2 TeV
Pythia8.230(Monash13)

red = transverse momenta of charged particles

green = the transverse momenta of neutral particles

blue = charged particle multiplicity
As Collections of Particles...

ParticleNet: Jet Tagging via Particle Clouds - July 23, 2019 - Huilin Qu (UCSB)

1607.08633

1711.09059

1711.02633

1607.08966, 1812.09223

\[ k_{\mu,i} \xrightarrow{\text{Cola}} k_{\mu,j} = k_{\mu,i} C_{ij} \]

with \( C = \begin{pmatrix} 1 & \ldots & 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 \\ 1 & \ldots & 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & \ldots & 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 \end{pmatrix} \)

\[ \tilde{k}_j \xrightarrow{\text{Loka}} k_j = \begin{pmatrix} m^2(k_j) \\ p_T(k_j) \\ \Delta R_{\text{jet}} \\ w_{ij}^{(E)} E(k_{ij}) \\ w_{ij}^{(d)} d_{ij}^{(d)} \\ E_T(k_j) E_T(k_m) \end{pmatrix} \]

And more...
As... Point Clouds?

Image from https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cffe
As... Point Clouds?

Point cloud

From Wikipedia, the free encyclopedia

A point cloud is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure a large number of points on the external surfaces of objects around them.
As... Point Clouds?

Point cloud

From Wikipedia, the free encyclopedia

A point cloud is a set of data points in space. Point clouds are generally produced by 3D scanners, which measure a large number of points on the external surfaces of objects around them.

Jet (Particle cloud)

From Wikipedia, the free encyclopedia

A jet (particle cloud) is a set of particles in space. Particle clouds are generally created by clustering a large number of particles measured by particle detectors, e.g., ATLAS and CMS.
**Point vs Particle Clouds**

- **Point cloud**
  - points are intrinsically *unordered*
  - primary information:
    - 3D coordinates in the xyz space

- **Particle cloud**
  - particles are intrinsically *unordered*
  - primary information:
    - 2D coordinates in the $\eta$-$\phi$ space
  - *but also many additional features!*
    - energy/momenta
    - charge/particle ID
    - track quality/impact parameters/etc.
Why Particle Cloud?

- Image-based approaches
  - natural idea for calorimeters
  - powerful performance using convolutional neural networks (CNNs)
  - projecting particles into pixels leads to high sparsity and loss of information
  - difficult to include non-additive features (e.g., particle ID, track impact parameters)

- Particle-based approaches
  - preserves full granularity - no loss of information from pixelation
  - straightforward to include any kind of features for each particle
  - constituent particles of a jet are intrinsically unordered – permutation symmetry!
  - however, most particle-based approaches (RNN/TreeNN/1D CNN) failed to exploit this symmetry

- The particle cloud approach
  - also particle-based
  - builds on permutation symmetry from the beginning
Learning from Particle Clouds

- Learning from point clouds
  - active research area in the ML community, mainly due to the rise of interest in autonomous driving
  - methods for point clouds can be imported and improved for particle clouds

- The key is to exploit the permutation symmetry
    - per-particle transformations + symmetric global function
    \[
    \mathcal{O}(\{p_1, \ldots, p_M\}) = F \left( \sum_{i=1}^{M} \Phi(p_i) \right),
    \]
    - generally, any permutation-invariant functions can be approximated with this form [Deep Sets, arXiv:1703.06114]
    - not explicitly exploiting the local structure of particle clouds

- our approach:
  - exploit both the local and global structures of particle clouds explicitly
    - convolution operation is the best candidate
  - use permutation-invariant convolution designed for point clouds
Convolution on point clouds: EdgeConv [arXiv:1801.07829]

- treating a point cloud as a graph: each point is a vertex
- for each point, a local patch is defined by finding its k-nearest neighbors
- designing a symmetric “convolution” function
  - define “edge feature” for each center-neighbor pair: \( e_{ij} = h_\Theta(x_i, x_j) \)
    - same \( h_\Theta \) for all neighbor points, and all center points, for symmetry
  - aggregate the edge features in a symmetric way: \( x_i' = \text{mean}_j e_{ij} \)
**Dynamic Graph CNN**

- *EdgeConv* shares many nice properties of regular CNNs
  - incorporates local neighborhood information
  - EdgeConv layers can be stacked to allow the network to learn both the local and global structures in a hierarchical manner

- **Dynamic Graph CNN (DGCNN)** [arXiv:1801.07829]
  - an *EdgeConv* operation outputs a set of learned features for each point
  - these features can also be interpreted as “coordinates” for each point (in a high-dim latent space)
  - such an interpretation enables us to recompute the “distances” between points and therefore dynamically update the k-nearest neighbor relations as going from one *EdgeConv* layer to another
  - found to be beneficial in the original paper
Based on EdgeConv and DGCNN, we developed ParticleNet, a customized architecture for jet tagging on particle clouds.
Performance Comparison

- The performance of ParticleNet is benchmarked on two jet tagging tasks using public datasets:
  - top tagging dataset: A. Butter, G. Kasieczka, T. Plehn and M. Russell [arXiv:1707.08966, link]

- Results compared with a few alternative algorithms
  - 1D CNN over particle sequence (P-CNN)
    - i.e., the CMS “DeepAK8” architecture [CMS-PAS-JME-18-002]
    - but using only information available in the datasets (e.g., no tracking)
  - 2D CNN over 64x64 jet image
    - state-of-the-art model for image recognition: ResNeXt-50 [arXiv:1611.05431]
    - #filters reduced by a factor or 4 to avoid overfitting and also to speed up training
  - Particle Flow Network (PFN) [arXiv:1810.05165]
    - not retrained — results are the ones reported in the original paper
Performance: Top Tagging

Top tagging:
- only four-momenta of the particles available in this dataset

Metrics:
- accuracy / area under the ROC curve (AUC)
- background rejection \(1/\varepsilon_b\) at signal efficiency of 30% and 50%
  - more relevant to physics analysis

Results:
- substantial improvement over P-CNN/PFN
  - \(~2\times\) background rejection
  - >40\% better than ResNeXt-50

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>AUC</th>
<th>(1/\varepsilon_b) at (\varepsilon_s = 50%)</th>
<th>(1/\varepsilon_b) at (\varepsilon_s = 30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNeXt-50</td>
<td>0.936</td>
<td>0.9837</td>
<td>302 ± 5</td>
<td>1147 ± 58</td>
</tr>
<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-Lite</td>
<td>0.937</td>
<td>0.9844</td>
<td>325 ± 5</td>
<td>1262 ± 49</td>
</tr>
<tr>
<td>ParticleNet</td>
<td>0.940</td>
<td>0.9858</td>
<td>397 ± 7</td>
<td>1615 ± 93</td>
</tr>
</tbody>
</table>
**Top Tagging Comparison**

- First direct comparison of a large number of ML algorithms on jet tagging, thanks to the public dataset prepared by G. Kasieczka et al.

More in S. Macaluso’s talk

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>Acc</th>
<th>(1/\epsilon_B (\epsilon_S = 0.3))</th>
<th>#Param</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>single</td>
<td>mean</td>
</tr>
<tr>
<td>CNN [16]</td>
<td>0.981</td>
<td>0.930</td>
<td>914±14</td>
<td>995±15</td>
</tr>
<tr>
<td>ResNeXt [30]</td>
<td>0.984</td>
<td>0.936</td>
<td>1122±47</td>
<td>1270±28</td>
</tr>
<tr>
<td>TopoDNN [18]</td>
<td>0.972</td>
<td>0.916</td>
<td>295±5</td>
<td>382±5</td>
</tr>
<tr>
<td>Multi-body (N)-subjettiness 6 [24]</td>
<td>0.979</td>
<td>0.922</td>
<td>792±18</td>
<td>798±12</td>
</tr>
<tr>
<td>Multi-body (N)-subjettiness 8 [24]</td>
<td>0.981</td>
<td>0.929</td>
<td>867±15</td>
<td>918±20</td>
</tr>
<tr>
<td>TreeNiN [43]</td>
<td>0.982</td>
<td>0.933</td>
<td>1025±11</td>
<td>1202±23</td>
</tr>
<tr>
<td>P-CNN</td>
<td>0.980</td>
<td>0.930</td>
<td>732±24</td>
<td>845±13</td>
</tr>
<tr>
<td>ParticleNet [47]</td>
<td>0.985</td>
<td>0.938</td>
<td>1298±46</td>
<td>1412±45</td>
</tr>
<tr>
<td>LBN [19]</td>
<td>0.981</td>
<td>0.931</td>
<td>836±17</td>
<td>859±67</td>
</tr>
<tr>
<td>LoLa [22]</td>
<td>0.980</td>
<td>0.929</td>
<td>722±17</td>
<td>768±11</td>
</tr>
<tr>
<td>Energy Flow Polynomials [21]</td>
<td>0.980</td>
<td>0.932</td>
<td>384</td>
<td></td>
</tr>
<tr>
<td>Energy Flow Network [23]</td>
<td>0.979</td>
<td>0.927</td>
<td>633±31</td>
<td>729±13</td>
</tr>
<tr>
<td>Particle Flow Network [23]</td>
<td>0.982</td>
<td>0.932</td>
<td>891±18</td>
<td>1063±21</td>
</tr>
<tr>
<td>GoaT</td>
<td>0.985</td>
<td>0.939</td>
<td>1368±140</td>
<td>1549±208</td>
</tr>
<tr>
<td><strong>ParticleNet-Lite</strong></td>
<td><strong>0.984</strong></td>
<td><strong>0.937</strong></td>
<td><strong>1262±49</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ParticleNet</strong></td>
<td><strong>0.986</strong></td>
<td><strong>0.940</strong></td>
<td><strong>1615±93</strong></td>
<td></td>
</tr>
</tbody>
</table>
Performance: Quark/Gluon Tagging

- Quark/gluon tagging
  - two versions with different information:
    - 4-momentum only
    - 4-momentum + particle ID

- Results
  - best performance obtained with ParticleNet in both cases (w/ and w/o PID)
  - addition of PID inputs has a large impact on the performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>AUC</th>
<th>$1/\varepsilon_b$ at $\varepsilon_s = 50%$</th>
<th>$1/\varepsilon_b$ at $\varepsilon_s = 30%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNeXt-50</td>
<td>0.821</td>
<td>0.8960</td>
<td>30.9</td>
<td>80.8</td>
</tr>
<tr>
<td>P-CNN</td>
<td>0.818</td>
<td>0.8915</td>
<td>31.0</td>
<td>82.3</td>
</tr>
<tr>
<td>PFN</td>
<td>-</td>
<td>0.8911 ± 0.0008</td>
<td>30.8 ± 0.4</td>
<td>-</td>
</tr>
<tr>
<td>ParticleNet-Lite</td>
<td>0.826</td>
<td>0.8993</td>
<td>32.8</td>
<td>84.6</td>
</tr>
<tr>
<td>ParticleNet</td>
<td>0.828</td>
<td>0.9014</td>
<td>33.7</td>
<td>85.4</td>
</tr>
<tr>
<td>P-CNN (w/ PID)</td>
<td>0.827</td>
<td>0.9002</td>
<td>34.7</td>
<td>91.0</td>
</tr>
<tr>
<td>PFN-Ex (w/ PID)</td>
<td>-</td>
<td>0.9005 ± 0.0003</td>
<td>34.7 ± 0.4</td>
<td>-</td>
</tr>
<tr>
<td>ParticleNet-Lite (w/ PID)</td>
<td>0.835</td>
<td>0.9079</td>
<td>37.1</td>
<td>94.5</td>
</tr>
<tr>
<td>ParticleNet (w/ PID)</td>
<td><strong>0.840</strong></td>
<td><strong>0.9116</strong></td>
<td><strong>39.8 ± 0.2</strong></td>
<td><strong>98.6 ± 1.3</strong></td>
</tr>
</tbody>
</table>

Table 3: Performance comparison on the quark-gluon tagging benchmark dataset. The ParticleNet, ParticleNet-Lite, P-CNN and ResNeXt-50 models are trained on the quark-gluon tagging dataset starting from randomly initialized weights. A total of 9 independent trainings are performed for the ParticleNet model, and the table shows the result from the median-accuracy training, with the standard deviation of the 9 trainings quoted as the uncertainty. Due to limited computational resources, the training of other models is performed only once, but the uncertainty due to random weight initialization is expected to be fairly small. The performance of PFN on this dataset is reported in [55], and the uncertainty corresponds to the spread in ten trainings. Note that a number of PFN models with different levels of PID information are investigated in [55], and "PFN-Ex", also using experimentally realistic PID information, is shown here for comparison.
Another aspect of machine learning models is the complexity
- number of parameters
- computational cost (e.g., inference time)

The high performance of ParticleNet comes at the cost of speed
- ~10x slower than P-CNN/PFN – but not prohibitively slow
- also: current implementation of EdgeConv not as optimized as regular convolutions

Good balance between performance and speed for ParticleNet-Lite
- 40% improvement in performance
- only a few times slower than P-CNN/PFN

PFN: extremely fast and still good performing
- suitable for extremely time critical tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Time (CPU) [ms]</th>
<th>Time (GPU) [ms]</th>
<th>$1/\varepsilon_b$ at $\varepsilon_s = 30%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNeXt-50</td>
<td>1.46M</td>
<td>7.4</td>
<td>0.22</td>
<td>1147 ± 58</td>
</tr>
<tr>
<td>P-CNN</td>
<td>348k</td>
<td>1.6</td>
<td>0.020</td>
<td>759 ± 24</td>
</tr>
<tr>
<td>PFN</td>
<td>82k</td>
<td><strong>0.8</strong></td>
<td><strong>0.018</strong></td>
<td>888 ± 17</td>
</tr>
<tr>
<td>ParticleNet-Lite</td>
<td>26k</td>
<td>2.4</td>
<td>0.084</td>
<td>1262 ± 49</td>
</tr>
<tr>
<td>ParticleNet</td>
<td>366k</td>
<td>23</td>
<td>0.92</td>
<td><strong>1615 ± 93</strong></td>
</tr>
</tbody>
</table>
SUMMARY

- Particle cloud: new approach to represent jets for ML
  - inspired by point clouds, a jet can be viewed as a cloud of particles, with manifest permutation symmetry

- ParticleNet: custom neural network architecture for jet tagging using particle clouds
  - significantly improved performance compared to existing ML approaches

- Particle cloud provides a natural and flexible representation for jets as well as the whole collision events, with a lot of potential yet to be fully explored
  - particle cloud classification (labelling the whole cloud) -> jet tagging
  - particle cloud segmentation (labelling each point) -> jet grooming / pileup identification?
  - and more...?
BACKUPS
COMPARISON WITH V1
## Input Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>TOP</th>
<th>QG</th>
<th>QG-PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta \eta$</td>
<td>difference in pseudorapidity between the particle and the jet axis</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\Delta \phi$</td>
<td>difference in azimuthal angle between the particle and the jet axis</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\log p_T$</td>
<td>logarithm of the particle’s $p_T$</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\log E$</td>
<td>logarithm of the particle’s energy</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\log \frac{p_T}{p_T(\text{jet})}$</td>
<td>logarithm of the particle’s $p_T$ relative to the jet $p_T$</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\log \frac{E}{E(\text{jet})}$</td>
<td>logarithm of the particle’s energy relative to the jet energy</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\Delta R$</td>
<td>angular separation between the particle and the jet axis $(\sqrt{(\Delta \eta)^2 + (\Delta \phi)^2})$</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$q$</td>
<td>electric charge of the particle</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>isElectron</td>
<td>if the particle is an electron</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>isMuon</td>
<td>if the particle is a muon</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>isChargedHadron</td>
<td>if the particle is a charged hadron</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>isNeutralHadron</td>
<td>if the particle is a neutral hadron</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>isPhoton</td>
<td>if the particle is a photon</td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
Conventional convolution only operates on regular grids and cannot be applied on point clouds

- point clouds are **irregular**
  - how to define a “local” patch?
- point clouds are **unordered**
  - conventional convolution operation \( \sum_i K_i x_i \) is not invariant under permutation of the points \( x_i \)