THE EUROPEAN
PHYSICAL JOURNAL PLUS

ABCNet: an attention-based method for particle tagging

V. Mikuni[a] , **F. Canelli**

University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

Abstract In high energy physics, graph-based implementations have the advantage of treating the input data sets in a similar way as they are collected by collider experiments. To expand on this concept, we propose a graph neural network enhanced by attention mechanisms called ABCNet. To exemplify the advantages and flexibility of treating collider data as a point cloud, two physically motivated problems are investigated: quark–gluon discrimination and pileup reduction. The former is an event-by-event classification, while the latter requires each reconstructed particle to receive a classification score. For both tasks, ABCNet shows an improved performance compared to other algorithms available.

1 Introduction

One of the main goals in modern machine learning is to be able to extract the maximum amount of information available from a data set. Successful implementations take advantage of the data structure for model building. In high energy physics (HEP), particle collisions in experiments are reconstructed by combining the energy deposits left by particles after crossing different parts of a detector. The information provided by sub-detectors can be further combined to give a full description of each particle produced. At the Large Hadron Collider (LHC) [1], jets are ubiquitous objects produced in proton–proton collisions. Jets are the byproducts of the hadronisation of quarks and gluons, resulting in an often collimated spray of particles. After each collision, $\mathcal{O}(1000)$ or more particles can be produced, making the task of identifying the original hard scattering objects challenging. The luminosity increase at the LHC will also increase the amount of multiple interactions per bunch crossing (pileup). For instance, event collisions recorded thus far by the ATLAS [2] and CMS [3] detectors at LHC measured an average of about 30 extraneous interactions. With the future upgrade, up to 200 pileup events per bunch crossing are expected, requiring new methods for particle identification and pileup suppression. In this paper, a new method for event classification in HEP is introduced. The attention-based cloud net (ABCNet) takes into account the data structure recorded by particle collision experiments, treating each interaction as an unordered set of points that defines a point cloud. This description is advantageous since the byproducts of each particle collision are treated in a similar fashion as they are collected by particle detectors. To enhance the extraction of local information, an attention mechanism is used, following closely the implementation developed in [4]. Attention mechanisms have proved to

---

[a] e-mail: vinicius.mikuni@uzh.ch (corresponding author)

boost performance for different applications in machine learning by giving local and global context to the learning procedure. To show the performance and flexibility of the model, two critical problems are investigated: quark–gluon discrimination and pileup mitigation.

2 Related works

The main novelties introduced by ABCNet are the treatment of particle collision data as a set of permutation invariant objects, enhanced by attention mechanisms to filter out the particles that are not relevant for the tasks we want to accomplish. The usage of graph-based machine learning implementations is still a new concept in particle physics. Nevertheless, new implementations have already been proposed with promising results. ParticleNet [5] uses a similar approach, using point clouds for jet identification. The main difference between ABCNet and ParticleNet is that ABCNet takes advantage of attention mechanisms to enhance the local feature extraction, allowing for a more compact and efficient architecture. A theory-inspired approach was also developed in the framework of Deep Sets [6] using an infrared and collinear safe basis, developed in the context of Energy Flow Networks [7]. A message-passing approach for jet tagging discussed in [8]. Interaction networks were also studied in the context of high-mass particle decays with JEDI-net [9]. Other graph-based implementations have also been presented in the context of signal and background classification [10,11], particle track reconstruction [12], and particle reconstruction on irregular calorimeters. [13]. In the context of pileup rejection, the GGNN implementation [14] shows promising results by combining graph nodes with GRU cells.

3 GAPLayer

ABCNet follows closely the implementation described for GAPNet [4], with key differences to adapt the implementation to our problems of interest. For clarity, the description of the essential aspects of the implementation are described. The key aspect of GAPNet is the development of a graph attention pooling layer (GAPLayer) using the edge convolution operation proposed in [15], which defines a convolution-like operation on point clouds together with attention mechanisms to operate on graph-structured data described in [16]. The point cloud is first represented as a graph with vertices represented by the points themselves. The edges are constructed by connecting the points to their k-nearest neighbours, while the edge features, $y_{ij} = (x_i - x_{ij})$, are taken as the difference between features of each point x_i and its k-neighbours x_{ij}. A GAPLayer is constructed by first encoding each point and edge to a higher-level feature space of dimension F using a single-layer neural network (NN), with learnable parameters θ, in the following form:

$$x_i' = h(x_i, \theta_i, F)$$
$$y_{ij}' = h(y_{ij}, \theta_{ij}, F)$$

where $h()$ denotes the single-layer neural network operation. Self- and local coefficients are created by passing the transformed points and edges to a single-layer NN with output dimension of size one. Finally, the attention coefficients c_{ij} are created by combining the newly created coefficients in the following way:

$$c_{ij} = \text{LeakyRelu}(h(x_i', \theta_i', 1) + h(y_{ij}', \theta_{ij}', 1)) \tag{1}$$

where the nonlinear LeakyRelu operation is applied to the output of the sum. To align the attention coefficients between different points, a Softmax normalisation is applied to the coefficients c_{ij}. At this moment, each point is associated with k attention coefficients. To compute a single attention feature for each point, a linear combination with a nonlinear activation function is defined as

$$\hat{x}_i = \text{Relu}\left(\sum_j c_{ij} y'_{ij}\right). \tag{2}$$

To enhance the stability of the determination of the coefficients $\hat{x}_i$, a multihead mechanism can be used. A M-head process repeats the same procedure described above, determining $\hat{x}_i$ M times, differing only on the random weight initialisation. The M results are combined by taking the maximum of the M different $\hat{x}_i$. The outputs of each GAPLayer consist of attention features ($\hat{x}_i$) and graph features (y'_{ij}). The graph features are further aggregated in the form:

$$y_{ij}^{max} = max(y'_{ij}).$$

Due to stackability properties, a GAPLayer output can be further used as an input to a subsequent GAPLayer or multilayer perceptron (MLP).

4 Classification: quark–gluon tagging

Quark–gluon tagging refers to the task of identifying the origin of a jet as produced from the hadronisation of a gluon or a quark. The data set used for the studies are available from [7]. It consists of stable particles clustered into jets, excluding neutrinos, using the anti-k_T algorithm [17] with $R = 0.4$. The quark-initiated sample (signal) is generated using a $Z(\nu\nu) + (u, d, s)$ while the gluon-initiated data (background) are generated using $Z(\nu\nu) + g$ processes. Both samples are generated using PYTHIA8 [18] without detector effects. Jets are required to have transverse momentum $p_T \in [500, 550]$ GeV and rapidity $|y| < 1.7$ for the reconstruction. For the training, testing and evaluation of the method, the recommended splitting is used with 1.6M/200k/200k events, respectively. For every reconstructed jet, up to 200 constituents are saved. Each constituent contains the four momentum and the expected particles type (electron, muon, photon, or charged/neutral hadrons). A typical jet has $\mathcal{O}(10)$ to $\mathcal{O}(100)$ particles. To simplify the implementation, ABCNet uses the first 100 constituents sorted by p_T from highest to lowest. If the jet has less than 100 constituents, the event is padded with zeros; if there are more than 100 constituents, the event is truncated.

To enhance the non-local information extraction, global features can also be added to ABCNet. The approach is similar to the one described in [19], where global information is used to parameterise the network, improving the generalisation and performance as a function of the global parameters.

The features used to describe each constituent are listed in Table 1.

4.1 Network architecture

The network layout used is shown in Fig. 1. The first step is to calculate the distances between the constituents in the pseudorapidity–azimuth ($\eta - \phi$) space of the form $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$. From the distances, we create the first GAPLayer by associating each particle to its nearest 10 neighbours. While different choices for k were tested, the overall performance did not improve with the addition of more neighbours. The encoding channel size of

Table 1 Description of each feature used to define a point in the point cloud implementation for quark–gluon classification. The latter two features are the global information added to the network

Variable	Description
$\Delta\eta$	Difference between the pseudorapidity of the constituent and the jet
$\Delta\phi$	Difference between the azimuthal angle of the constituent and the jet
$\log p_T$	Logarithm of the constituent's p_T
$\log E$	Logarithm of the constituent's E
$\log \frac{p_T}{p_T(jet)}$	Logarithm of the ratio between the constituent's p_T and the jet p_T
$\log \frac{E}{E(jet)}$	Logarithm of the ratio between the constituent's E and the jet E
ΔR	Distance in the $\eta - \phi$ space between the constituent and the jet
PID	Particle type identifier as described in [20]
m(jet)	Jet mass
p_T(jet)	Jet transverse momentum

the GAPLayer F is selected to be 32 with a 1-head. The attention features created by the GAPLayer are then passed through two MLPs with node sizes (128,128). The distances used for the second GAPLayer are calculated using the full-feature space produced in the output of the last MLP, allowing the network to learn distances in the transformed feature space. To achieve a robust estimation, the encoding channel size is selected to be 64 with the number of heads determined to be two. The newly created attention features are passed through two MLPs of node sizes each of 128. In parallel, ABCNet also takes additional global inputs in the form of the jet mass and transverse momenta. The global inputs are first transformed by means of a single-layer MLP with small node size of 16. The two graph features and the output of each MLP are concatenated with the transformed global features and fed to a MLP of node size 128. An average pulling is applied, and the result is further passed to 2 additional MLPs of node sizes (128,256) interleaved by two dropout layers. A Softmax operation is applied to the output result.

4.2 Results

The performance of ABCNet is compared to the methods implemented in [5] and [7], using the same data set.

The figures of merit used for the comparison are:

- Accuracy: Ratio between the number of correct predictions and the total number of test examples.
- AUC: Integral of the area under the receiver operating characteristic distribution.
- $1/\epsilon_B$: One over the background efficiency for a fixed value of the signal efficiency (50% or 30%)
- Parameters: Number of trainable weights for the model.

The results of the comparisons are listed in Table 2. Even though the accuracy obtained by ABCNet is numerically the same as the one reported by ParticleNet, ABCNet excels on the other figures of merit, improving the background rejection, at 30% signal efficiency, by 15–20%. The use of attention coefficients allow the model complexity of ABCNet to be reduced, having 40% less parameters compared to ParticleNet.
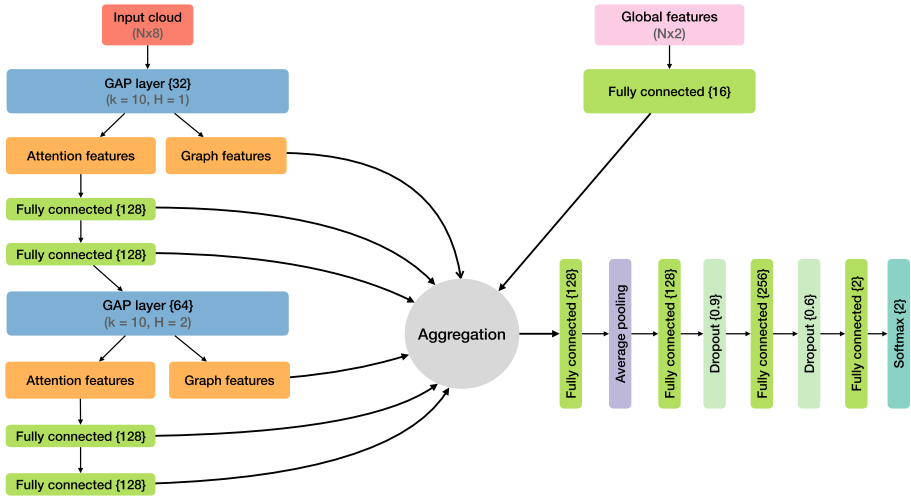
Fig. 1 ABCNet architecture used for quark–gluon tagging. Fully connected layer and encoding node sizes are denoted inside "{}". For each GAPLayer, the number of k-nearest neighbours (k) and heads (H) is given

Table 2 Comparison between the performance achieved with ABCNet and different available implementations

|	Acc	AUC	$1/\epsilon_B$ ($\epsilon_S = 0.5$)	$1/\epsilon_B$ ($\epsilon_S = 0.3$)	Parameters
ResNeXt-50	0.821	0.9060	30.9	80.8	1.46M
P-CNN	0.827	0.9002	34.7	91.0	348k
PFN	–	0.9005	34.7 ± 0.4	–	82k
ParticleNet-Lite	0.835	0.9079	37.1	94.5	**26k**
ParticleNet	**0.840**	0.9116	39.8 ± 0.2	98.6 ± 1.3	366k
ABCNet	**0.840**	**0.9126**	**42.6 ± 0.4**	**118.4 ± 1.5**	230k

The uncertainty quoted corresponds to the standard deviation of nine trainings with different random weight initialisation. If the uncertainty is not quoted, then the variation is negligible compared to the expected value

4.3 Visualisation

A simple way to check what ABCNet is learning is to look at the self-coefficients of each point of the point cloud. First, we pre-processes the images in a similar fashion as [21], using the following steps:

– Centre: All jet images are translated in the $\eta - \phi$ space to a common centre at (0,0). The centre of the jet is taken as its p_T-weighted centroid.
– Particle scale: Each particle constituent has its transverse momentum scaled such that $\sum^{jet} p_{T,i} = 1$, where i is the i-th constituent of the jet.
– Overall scale: The final image is created by superimposing the individual event images and dividing the resulting distribution by the number of events in the test sample.

Other steps were adopted in [21]; however, since the goal is to have a simple visual cue, they were not used. The resulting jet images are shown in Fig. 2 for quark- and gluon-initiated jets on the upper and lower rows, respectively. The leftmost images correspond to the jets after the pre-processing. The subsequent columns show the same distribution, but only considering
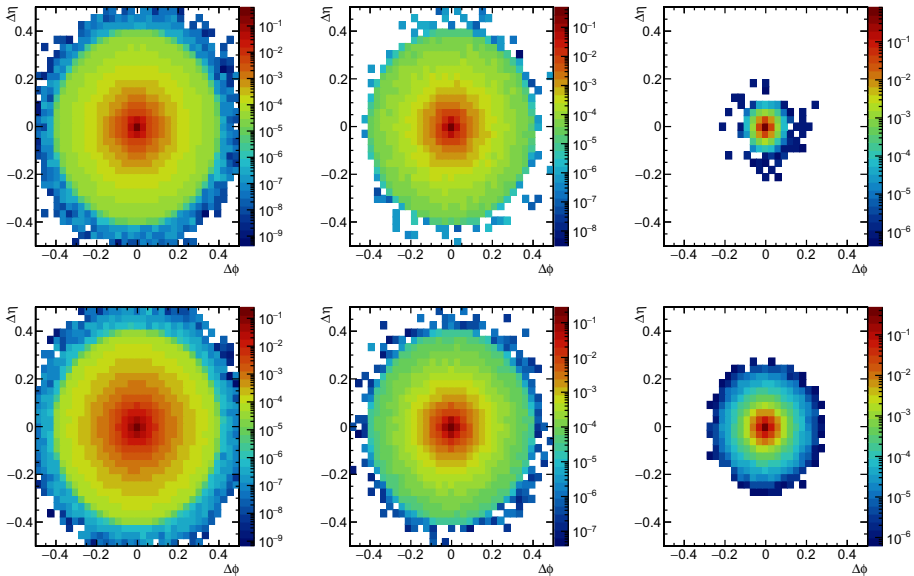
Fig. 2 Distribution of the p_T-scaled distribution of the jet constituents averaged over all images in the test sample. The leftmost images are the quark (top) and gluon (bottom) jet averages after the pre-processing. The first 5% of the jet constituents with the highest self-attention coefficients for the first and second GAPLayers are shown on the images in the centre and right, respectively

particles whose self-attention coefficients, resulting from the first (middle column) and second (right column) GAPLayers, are higher than a certain value. This value is chosen such that only the first 5% of all particles with the largest self-attention coefficients are selected. The self-coefficients from the first GAPLayer have the effect of giving higher attention to high-p_T particles, while soft-QCD with large angular variation has less importance. The second GAPLayer, where nearest neighbours are calculated in the feature space, has different distributions for quark-initiated and gluon-initiated jets. Quark-initiated jets have the highest coefficients in a confined radius with $\sim \Delta R = 0.1$ around the centre, while gluon-initiated coefficients spam a bigger area around the centre with $\sim \Delta R = 0.3$. That behaviour is expected since gluon jets have a larger colour factor compared to quark jets, typically resulting in a broader angular distribution compared to quark jets.

5 Pileup reduction using part segmentation

Another crucial problem in particle physics is how to identify the particles originated from high-p_T collisions and separate them from unwanted additional interactions. Two traditional methods to accomplish this task are the SoftKiller [22] and the Pileup Per Particle Identification (PUPPI) [23] algorithms. These two algorithms are chosen since they represent the most common algorithms for pileup mitigation at the LHC. To test the performance of ABCNet in this context, we change the scope of a single jet classifier to a particle-by-particle classification (part segmentation). In this case, a probability is estimated for object, determining how likely it is for each particle to originate from the leading vertex (LV). The sample used for this study is available from [24], containing a set of $q\bar{q}$ light-quark-initiated jets coming

Table 3 Variable description for each feature used to define a point in the point cloud implementation for the pileup mitigation problem. The latter two features are the global information added to the network

Variable	Description
η	Particle's pseudorapidity
ϕ	Particle's azimuthal angle
$\log p_T$	Logarithm of the particle p_T
Q	Boolean flag identifying if the particle is charged
$\log \frac{p_T}{p_T(\text{jet})}$	Logarithm of the ratio between the particle p_T and the associated jet p_T
$\log \frac{E}{E(\text{jet})}$	Logarithm of the ratio between the particle E and the associated jet E
w_{PUPPI}	PUPPI weight for the particle
$w_{\text{SoftKiller}}$	Boolean flag identifying if the particle passes the SoftKiller p_T requirement
NPU	number of pileup interactions
NPART	Number of reconstructed particles associated with jets

from the decay of a scalar particle with mass $m_\phi = 500$ GeV. The samples were generated using PYTHIA8 at $\sqrt{s} = 13$ TeV. The pileup events were generated by overlaying soft QCD processes onto each event. Stable particles are clustered into jets, excluding neutrinos, using the anti-k_T algorithm with R=0.4. At parton level, a p_T requirement of at least 95 GeV was applied. Only jets satisfying p_T>100 GeV and $\eta \in [-2.5, 2.5]$ are considered. For each event, up to two leading jets as ordered in p_T are stored. Two thousand events are generated, each with a different number of pileup interactions (NPU) ranging from 0 to 180. For the training and testing samples, events are randomly selected from the generated samples according to a Poisson distribution with average pileup <NPU> = 140, motivated by the expected pileup levels for future collisions at the LHC. The training and evaluation are done with 80% and 10% of the events with <NPU> = 140, respectively. For testing, two samples are created: one corresponding to the remaining 10% of the events and <NPU> = 140 and the other a sample with independent events generated at different NPU levels. For each event, up to 500 particles are stored as long as they are matched to one of the two leading jets. The features used to define each particle are described in Table 3. The feature choice is similar to the ones used for the classification task. The main difference is that for this sample, the PID information is not available, but replaced by a flag that identifies if a particle is charged or not. Since more than one jet can be reconstructed, a global zero is used for all events, instead of using the jet axis as a reference point. While no selection is applied to the particles used in ABCNet, the PUPPI weights and the SoftKiller decision flag are also used as input features. The global information added to the parameterisation is NPU and the number of reconstructed particles associated with jets.

5.1 Network architecture

The network architecture for the part segmentation problem is similar to the setup used previously. The main differences are:

– Number of considered neighbours increased from 10 to 50.
– Additional MLPs after the attention features and after the pooling layer.
– Usage of only 1-head GAPLayers.
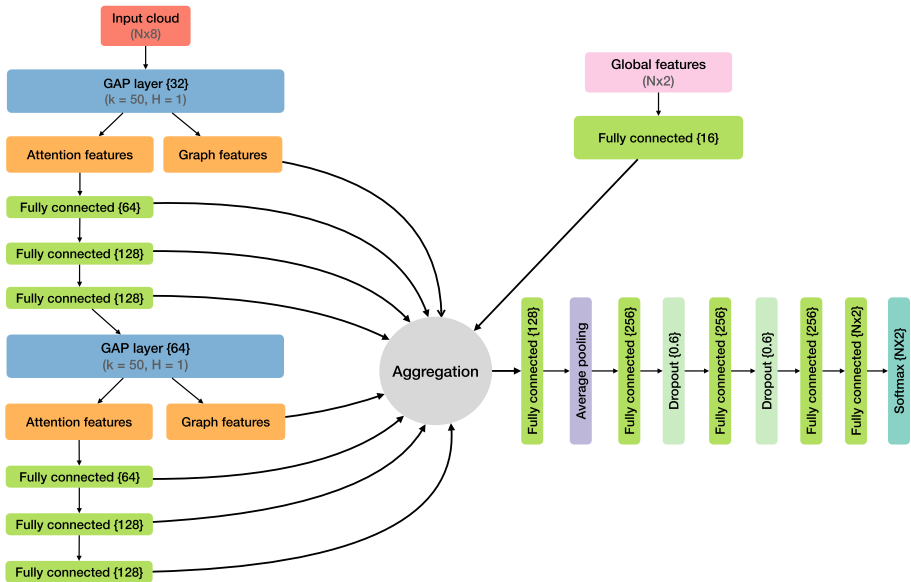
Fig. 3 ABCNet architecture used for pileup identification. Fully connected layer and encoding node sizes are denoted inside "{}". For each GAPLayer, the number of k-nearest neighbours (k) and heads (H) are given

The increase in expansion of neighbours and MLPs is chosen to increase the model's capacity to cover the larger amount of points per event. The architecture is shown in Fig. 3.

5.2 Results

The performance of ABCNet is compared to the performance achieved using PUPPI and SoftKiller. The default parameters for those methods are the same as the ones used in [24]: $R_0 = 0.3$, $R_{min} = 0.02$, $w_{cut} = 0.1$, $p_T^{cut}(\text{NPU}) = 0.1 + 0.007 \times \text{NPU}$ (PUPPI), grid size = 0.4 (SoftKiller). First, the jet mass is reconstructed with the <NPU> = 140 evaluation sample, applying the different mitigation algorithms. Inspired by PUPPI, the output probabilities from ABCNet are used to reweight the four-momentum of each particle. The reconstructed dijet mass and the dijet mass resolution are shown in Fig. 4. The resolution is defined as:

$$\text{mass resolution} = \frac{m_{\text{reco}} - m_{\text{true}}}{m_{\text{true}}}.$$

In Table 4, the width of the jet mass resolution, extracted by fitting the distributions in Fig. 4 (right) with a Gaussian function, is also listed.

ABCNet improves jet mass resolution compared to both PUPPI and SoftKiller by 75% and 83%, respectively. The robustness of each algorithm is also tested by comparing. The Pearson linear correlation coefficient (PCC) between the true jet mass and corrected jet masses for different NPU is generated. Figure 5 shows the result of the comparison using the test sample with NPU from 0 to 180. To investigate the power of ABCNet to generalise, a training sample with <NPU> = 20 is created and trained using the same architecture described previously. For both trainings, ABCNet shows a superior performance compared to PUPPI and SoftKiller for the entire NPU range. Furthermore, ABCNet is also remarkably robust for
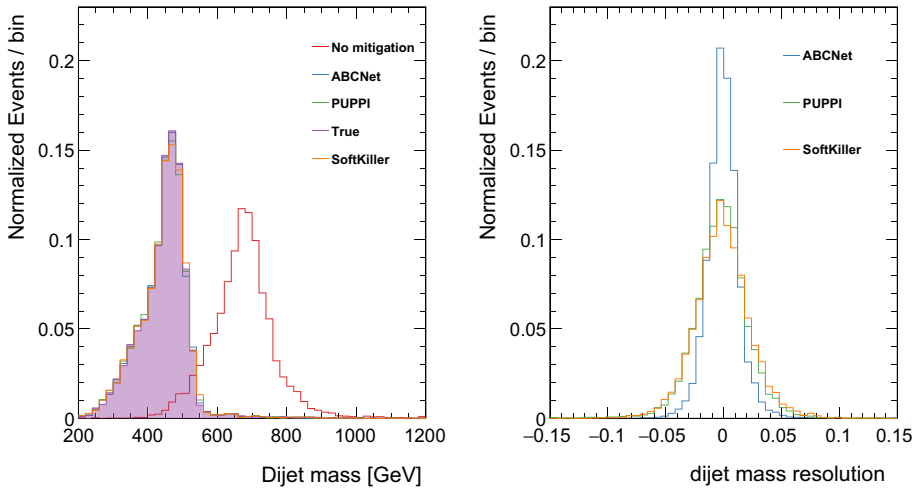
Fig. 4 Distribution of the dijet mass using the different pileup mitigation algorithms (left) and the jet mass resolution (right). A narrower resolution peak means better performance. All distributions are normalised to unit

Table 4 Resolution width for different pileup mitigation strategies. The resolution width is extracted by fitting the distributions shown in Fig. 4 (right) with a Gaussian function

Algorithm	Resolution width
SoftKiller	0.022
PUPPI	0.021
ABCNet	**0.012**

pileup variations outside the training region due to the addition of the global parameters to the method.

6 Training details

ABCNet is implemented using Tensorflow v1.4 [25]. A Nvidia GTX 1080 Ti graphics card is used for the training and evaluation steps. For all tasks described in this paper, the Adam optimiser [26] is used. The learning rate starts from 0.001 and decreases linearly by a factor 10 every seven epochs, until reaching a minimum of 1e-7. The training is performed with a mini batch size of 64 to a maximum number of 50 epochs. The epoch with the highest accuracy on the evaluation is saved in the case of the quark–gluon classification task. For the pileup identification, the epoch with the lowest loss is stored.

7 Conclusion

In this document, a new machine learning implementation for data classification in HEP is introduced. The attention-based cloud net (ABCNet) takes advantage of the data structure commonly found in particle colliders to create a point cloud interpretation. An attention mechanism is implemented to enhance the local information extraction and provide a simple way
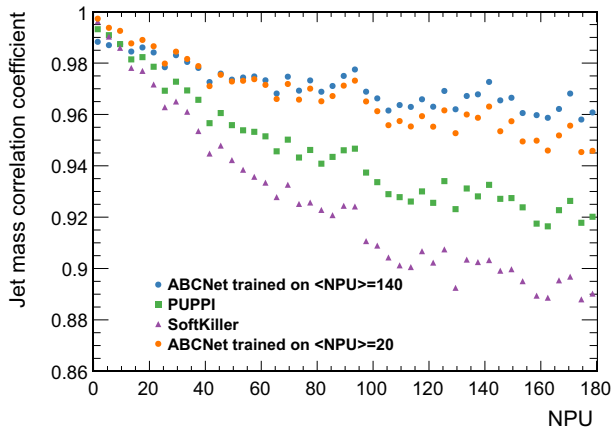
Fig. 5 PCC for each pileup mitigation algorithm for different NPU. ABCNet is trained on $<NPU> = 140$ (blue) or $<NPU> = 20$ (orange)

to investigate what the method is learning. To capture the global information, direct connections for global input features can be directly added. ABCNet can be used for event-by-event classification problems or generalised to particle-by-particle classification. To exemplify the architecture flexibility, two example problems are investigated: quark–gluon classification and pileup mitigation. For both problems, ABCNet achieved an improved performance compared to other available methods. By using a graph architecture and interpreting each point in a point cloud as a particle, ABCNet can be readily adapted to other applications in HEP like jet-flavour tagging, boosted jet identification, or particle track reconstruction.

References

1. L. Evans, P. Bryant, LHC machine. JINST **3**, S08001 (2008)
2. ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. JINST, **3**, S08003 (2008)
3. C.M.S. Collaboration, The CMS Experiment at the CERN LHC. JINST **3**, S08004 (2008)
4. C. Chen, L.Z. Fragonara, A. Tsourdos. GAPNet: Graph attention based point neural network for exploiting local feature of point cloud. arXiv e-prints, arXiv:1905.08705 (2019)
5. H. Qu, L. Gouskos, ParticleNet: Jet Tagging via Particle Clouds. arXiv e-prints, arXiv:1902.08570 (2019)
6. M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R.R. Salakhutdinov, A.J. SmolaSmola, Deep sets, in *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon, U.V. Luxburg, S. Bengio,

H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Curran Associates, Inc., Red Hook, 2017), pp. 3391–3401

7. P.T. Komiske, E.M. Metodiev, J. Thaler, Energy flow networks: deep sets for particle jets. J. High Energy Phys. **2019**(1), (2019)

8. A. Lister J. Pearkes S. Egan, W. Fedorko, C. Gay. Neural message passing for jet physics, in *Proceedings of the Deep Learning for Physical Sciences Workshop at NIPS* (2017)

9. E.A. Moreno, O. Cerri, J.M. Duarte, H.B. Newman, T.Q. Nguyen, A. Periwal, M. Pierini, A. Serikova, M. Spiropulu, J.-R. Vlimant, JEDI-net: a jet identification algorithm based on interaction networks (2019)

10. M. Abdughani, J. Ren, L. Wu, J.M. Yang, Probing stop pair production at the LHC with graph neural networks. JHEP **08**, 055 (2019)

11. N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, P.W. Bhimji, M.M. Bronstein, S.R. Klein, J. Bruna, Graph neural networks for icecube signal classification. CoRR arXiv:1809.06166 (2018)

12. S. Farrell, et al., Novel deep learning methods for track reconstruction, in *4th International Workshop Connecting The Dots 2018 (CTD2018)* Seattle, Washington, USA, March 20–22, 2018 (2018)

13. S.R. Qasim, J. Kieseler, Y. Iiyama, M. Pierini, Learning representations of irregular particle-detector geometry with distance-weighted graph networks. Eur. Phys. J. C **79**(7), 608 (2019)

14. J.A. Martínez, O. Cerri, M. Spiropulu, J.R. Vlimant, M. Pierini, Pileup mitigation at the large hadron collider with graph neural networks. Eur. Phys. J. Plus **134**(7), 333 (2019)

15. Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds. CoRR, arXiv:1801.07829 (2018)

16. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks (2017)

17. M. Cacciari, G.P. Salam, G. Soyez, The anti-k_T jet clustering algorithm. JHEP **04**, 063 (2008)

18. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, An introduction to PYTHIA 8.2. Comput. Phys. Commun. **191**, 159–177 (2015)

19. P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, D. Whiteson, Parameterized neural networks for high-energy physics. Eur. Phys. J. C **76**(5), 235 (2016)

20. M. Tanabashi et al., Review of particle physics. Phys. Rev. D **98**(3), 030001 (2018)

21. P.T. Komiske, E.M. Metodiev, M.D. Schwartz, Deep learning in color: towards automated quark/gluon jet discrimination. JHEP **01**, 110 (2017)

22. M. Cacciari, G.P. Salam, G. Soyez, SoftKiller, a particle-level pileup removal method. Eur. Phys. J. C **75**(2), 59 (2015)

23. D. Bertolini, P. Harris, M. Low, N. Tran, Pileup per particle identification. JHEP **10**, 059 (2014)

24. P.T. Komiske, E.M. Metodiev, B. Nachman, M.D. Schwartz, Pileup mitigation with machine learning (PUMML). JHEP **12**, 051 (2017)

25. M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems (2015)d. Software available from http://www.tensorflow.org

26. D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. arXiv e-prints, arXiv:1412.6980 (2014)