Published for SISSA by Springer

Received: *February 12, 2021*
Revised: *March 26, 2021*
Accepted: *April 2, 2021*
Published: *April 30, 2021*

Combine and conquer: event reconstruction with Bayesian Ensemble Neural Networks

Jack Y. Araz and Michael Spannowsky

Institute for Particle Physics Phenomenology, Durham University, South Road, Durham, DH1 3LE, U.K.

E-mail: jack.araz@durham.ac.uk, michael.spannowsky@durham.ac.uk

Abstract: Ensemble learning is a technique where multiple component learners are combined through a protocol. We propose an Ensemble Neural Network (ENN) that uses the combined latent-feature space of multiple neural network classifiers to improve the representation of the network hypothesis. We apply this approach to construct an ENN from Convolutional and Recurrent Neural Networks to discriminate top-quark jets from QCD jets. Such ENN provides the flexibility to improve the classification beyond simple prediction combining methods by linking different sources of error correlations, hence improving the representation between data and hypothesis. In combination with Bayesian techniques, we show that it can reduce epistemic uncertainties and the entropy of the hypothesis by simultaneously exploiting various kinematic correlations of the system, which also makes the network less susceptible to a limitation in training sample size.

Keywords: Jets

ArXiv ePrint: 2102.01078

Open Access, © The Authors.
Article funded by SCOAP[3].

https://doi.org/10.1007/JHEP04(2021)296

JHEP04(2021)296

Contents

JHEP04(2021)296

1 Introduction

Deep Learning (DL) has gained tremendous momentum on the verge of the latest developments in data analysis. Whilst boosted decision trees (BDT) have been used in the context of High-Energy Physics for over 30 years, wide usage of Deep Neural Networks (DNNs) only surged very recently. Since then, especially in applications to LHC physics where a large amount of data with the need for its fast and automated analysis is gathered, there has been a profound improvement in the understanding of Neural Networks (NNs). The analysis of the internal structure of jets, highly complex collimated sprays of radiation [1], is a popular arena where reconstruction techniques evolved from sophisticated multi-variate approaches, e.g. HEPTopTagger [2–4], over theory-guided matrix-element methods [5–8] to data-driven NN techniques [9–12]. In particular top tagging has been the prime example to benchmark the performance of various NN classifiers [13–21]. Similar tagging algorithms have been used for Higgs [22, 23] and W-boson [24, 25] tagging and quark-gluon discrimination [26–30].[1] Thus, it became apparent that there is a wide range of use-cases for NNs in collider phenomenology, where particle tagging is just one of many applications.

A standard supervised learning algorithm produces a fitting function that aims to find an optimal contour of the decision boundary between competing hypotheses.[2] The given algorithm takes a labelled feature-tensor and attempts to find the global minimum of a given objective function, the so-called loss function, resulting in the prediction of the algorithm. This is achieved by convoluting the input feature vector with non-linear functions, so-called activation functions, and updating the weights of the initial hypothesis through the backpropagation algorithm. Whilst such an approach offers increased flexibility, in general,

[1] For a review of these methodologies and more see refs. [14, 31], and other examples [32–44].

[2] Here the word "fitting" is used to simplify the text. However, Deep Learning is not merely a fitting algorithm; it looks for a higher dimensional irreducible representation that the feature-space lives in.

it can suffer from three major predicaments [45]. First, the problem of statistics denotes the lack of training examples within a particular domain, which can cause the learning algorithm to get stuck in various minima with comparable accuracies in each training. The second problem is computational. As mentioned before, a learning algorithm often employs a stochastic search algorithm, e.g. gradient descent. Assuming the provision of sufficient data, the feature-space can be highly complex, creating a very non-trivial loss-hypersurface for which the algorithm is tasked to find the global minimum [46, 47]. Finally, the third problem is representational. As the nature of a "fitting" algorithm, it is not always possible to find a linear or non-linear representation of the actual function. Hence, it might be necessary to expand the representation space or employ various possible hypotheses to find a closer approximation of the actual function. Although the representation problem is directly linked to the previously mentioned issues, even with sufficient statistics and advanced algorithms, an optimization algorithm may not proceed after finding a hypothesis that can adequately explain the data [48].

The three most popular architectures for classification tasks in particle physics are currently Deep Neural Networks (DNN), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Each of these networks is designed to exploit different features and correlations of the input data. For instance, CNNs are special-purpose networks that are widely used for image recognition [14, 18]. This method sweeps through the image by dividing it into subvolumes. Each subvolume has been transferred to the next layer by passing through an activation function, allowing the network to filter the image's distinguishable features. RNNs are a different kind of specialised networks that keep track of the ordering of the feature vector and thus maintains a sense of "memory" by connecting each node in a graph via an ephemeral sequence. Long-short term memory (LSTM) networks have been employed to classify QCD events with high accuracy [17, 25, 49]. While each of these techniques can be powerful by itself, it is not clear whether they exploit the full amount of information contained in the feature vectors to perform an optimal classification between competing hypotheses. Thus, combining multiple networks into an Ensemble Neural Network (ENN) might allow to improve on their individual classification performances.

Ensemble learning is a paradigm which employs multiple neural networks to solve a problem. The main idea behind ensembling is to increase the generalisation of the data by harvesting many hypotheses trying to solve the same problem. Each of the networks mentioned above is specialised to learn a particular feature of the given data to achieve the same or similar generalisation. An ensemble of these networks can access all the information presented in each component network and optimise it according to more generic information through data [50–55].

While some techniques to combine classifiers have been used in the context of collider phenomenology before,[3] to our knowledge for the first time, we will present a parallel combining method to go beyond simple prediction combinations. As shown in previous stud-

[3]Ref. [56] shows that combining predictions of BDTs with specific rules can improve the discrimination of BSM models from the SM. Ref. [57] shows that injecting randomness to a hypothesis and combining its results can boost the accuracy of the classification. Refs. [58, 59] uses stack combining method for Higgs tagging at LHC and ref. [60] combines the predictions of multiple different learners.

JHEP04(2021)296

ies [45, 46, 51, 61], combining predictions of various networks can significantly improve the overall performance for classification or regression. However, if networks are only combined at the prediction level, they are each separately trained for a specific property of the data. Parallel combined ensembles allow the network to train on a combined higher dimensional latent-space to optimize the entire network accordingly. Hence, having access to all component networks allows improvement upon the representation of the problem. We will show that such an approach allows flexibility to improve background rejection beyond simple prediction combinations. Furthermore, we will show that it will drastically improve the network's error correlations beyond the component and prediction-based-combined networks.

With continuously improving performance indicators for NNs, e.g. measured through receiver operating characteristic (ROC) curves, it becomes increasingly important to obtain an understanding of how this is achieved and how reliable the performance is evaluated [62–67]. Bayesian neural networks allow to estimate intrinsic uncertainties of NN by treating their weights as distributions instead of a single trainable variable [68, 69]. Hence the network output is a distribution rather than a fix value. To estimate the uncertainties of a network, multiple measurements of the same test data are combined to calculate the mean prediction alongside its standard deviation. We will employ Bayesian techniques to show that parallel combining methods, i.e. as implemented in ENNs, can reduce the standard deviation of the predictions and epistemic uncertainties without requiring more data.

In section 2 we provide a discussion of Ensemble Neural Networks and review their applications and benefits in improving the classification performance. In section 3.1 we describe the procedure we employed to preprocess the input data before the training and in section 3.2 we present our results. Finally, in section 4 we compare uncertainties between component networks and their ensemble, and we offer a summary and conclusions in section 5.

2 Ensemble Neural Networks

Ensemble Neural Networks (ENNs) are protocols that aim to increase the generalizability of a hypothesis by combining multiple component networks. It has been shown that ENNs can provide the necessary resolutions or approximations that all three potential pitfalls for NNs mentioned in section 1 require [50–54]. Depending on the problem at hand, ensembling methods can be pooled under three paradigms [55]: (i) parallel combining, (ii) stacked combining and (iii) combining weak classifiers.

Combining classifiers spanning feature-spaces that contains different physical domains, can provide an expanded representation of the hypothesis space, see figure 1. Such methods are studied under so-called "parallel combining" method. Another technique, called "stacked combining", employs different classifiers to be trained on the same feature-tensor. Such techniques can provide simple solutions to the computational problem where multiple non-correlated hypotheses can approximate the underlying function more efficiently. The final and most widely studied method is "combining weak classifiers" where, as the name suggests, weak but successful classifiers' predictions are assembled to create a NN that reaches accuracies beyond its constituents [45]. Here successful means that the hy-

JHEP04(2021)296

pothesis has greater accuracy than random selection. Although existing methods under the paradigms (*ii*) and (*iii*) can successfully optimize over statistical and computational shortcomings of the NNs [70–77], they can not expand the representation of the hypothesis without acquiring an extended domain of the data. Hence one needs a dedicated approach to address the representation problem to learn over different types of correlations within distinct symmetries of the data.

While ENNs are known to improve on the statistics and computational problems [55], see section 1, its benefits for the representation problem, which is in most collider phenomenological applications often is prevalent, is underrated. We propose the use of ENNs for the event reconstruction at high-energy collider experiments under the paradigm of parallel combining. We will further show that this approach improves on the representation problem.

For this purpose, we will use two high-level classifiers, a CNN and a RNN which are often used for image recognition and text recognition respectively. Both of these models are generalising a specific property of a jet, i.e. the spatial position of the substructure of a jet and the sequential order of a cluster history respectively. Naively, one could take the mean prediction of both classifiers, which will lead to a generalisation of the problem in the higher-dimensional hypothesis space. Although this can improve the performance, both component networks are optimised for their own feature space. In this study, we show that instead of combining the component networks' predictions, optimising the network over the combined latent-feature space can lead to a more substantial and stable performance improvement for the problem at hand.

Thus, we propose to initialise multiple high-level classifiers separately. For the example of section 3, these are chosen to be CNN and RNN classifiers. Each the CNN and RNN provide a vector in the latent-feature space corresponding to the flattened image for the CNN and the higher dimensional representation sequence for the RNN. Concatenating these vectors will lead to a larger latent-space, including information from both image-type and sequence-type data. Training with this higher dimensional feature space with extra handles for the NN architecture, such as more layers or nodes to generalise this latent-feature space, can lead to two significant improvements. Firstly, each component network's weights will be optimised with respect to the combined hypothesis space hence will have access to more features of the base theory. Secondly, the ability to access a larger latent-feature space will make it possible to increase the complexity of the model for a larger hypothesis-space.

Figure 1 shows a schematic representation of this approach where one source of input is divided into multiple branches to be analysed within different architectures. Depending on the nature of the problem, one can employ multiple network architectures such as fully connected networks (blue), CNNs (purple), RNNs (green) or even more complex structures which, for the sake of simplicity, are not shown explicitly. The merging stage represents the concatenation process where instead of the prediction of each model, one can combine the latent-space of each network after its individual i^{th} layer and continue training on this new feature space. Hence, the network's output will be the prediction optimised over each distinct feature of the problem.

JHEP04(2021)296

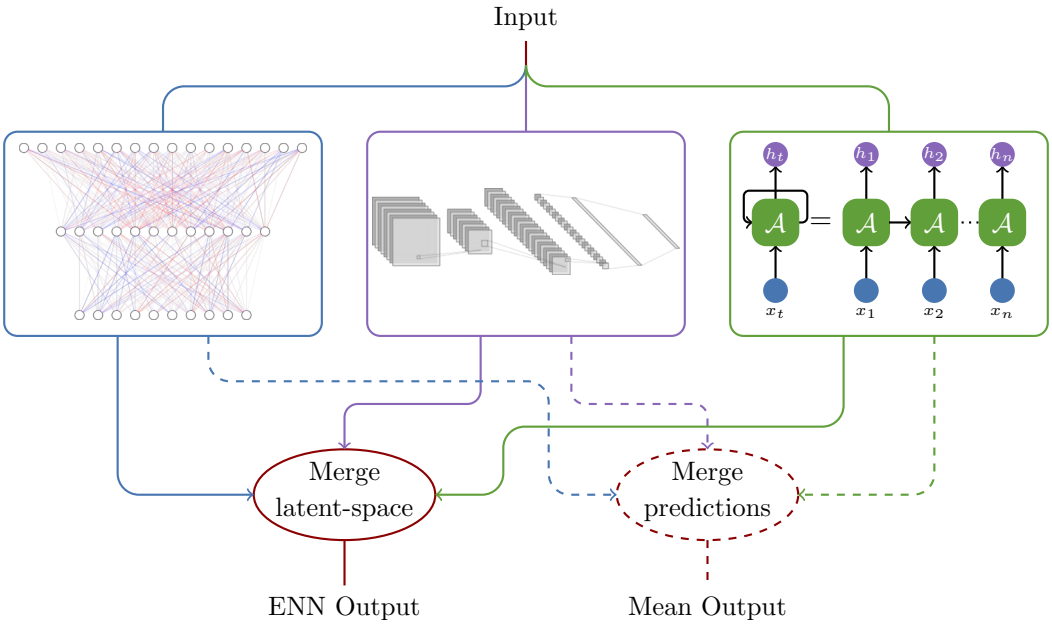

Figure 1. A schematic representation of ensemble neural networks where blue box represents a NN with dense layers, purple represents convolutional neural network and green represents a recurrent NN with inputs x_i and output values h_i for an operator $\mathcal{A}$. Solid line at the bottom guides towards latent-space concatenation which leads to ensemble prediction. Dashed lines represent the same for mean prediction of each network.

Whilst the network architectures discussed often unveil a strong performance improvement over conventional cut-based reconstruction strategies; one wonders if combining any NN will increase accuracy. To answer this question one needs to investigate the bias-variance-covariance decomposition. The prediction of an ensemble estimator, constructed by averaging the prediction of each component estimators, assuming that they are independent from each other, can be cast as

$$f_{\text{ens}}(\mathbf{x}) = \frac{1}{N}\sum_{i}^{N} f_i(\mathbf{x})\,, \tag{2.1}$$

where N is the number of component estimators, $f_i(\mathbf{x})$ is the prediction of the i^{th} estimator and $\mathbf{x}$ is the feature-tensor. For such an object, the generalization error is given by [61, 78]

$$\text{Err}(f_{\text{ens}}) = \text{Err}\left\{\frac{1}{N}\overline{\text{Var}}(\mathbf{x}) + \left(1-\frac{1}{N}\right)\overline{\text{Cov}}(\mathbf{x}) + \overline{\text{Bias}}(\mathbf{x})^2\right\}\,, \tag{2.2}$$

where the three terms correspond to variance, covariance and the squared bias of the feature-tensor respectively. Although such construction assumes a very simplistic case, it shows that the generalization error of the average prediction of multiple hypotheses is also affected by the covariance. This shows that if the component hypotheses are negatively correlated with each other the average prediction will decrease the generalization error further. However, as the average bias will remain the same, the generalization error can

JHEP04(2021)296

only be reduced to the bias term. Thus an ENN can improve the generalization error if and only if the given component estimators' errors are not completely correlated [50, 79].

3 Top tagging through ensemble learning

Using CNNs, the pixelated energy deposits in the calorimeters of multi-purpose high-energy experiments have been repeatedly shown to provide a strong discriminatory power between the radiation profile of top quarks versus QCD jets. In the $\eta - \phi$ plane, each pixel corresponds to one or more particles, and so-called colour or luminosity of a pixel can be measured by a combined intrinsic property of these particles such as energy or transverse momentum. This will allow the CNN to learn translationally invariant features of the top and jet system. RNNs instead maintain a sense of timing and memory in a given sequence used as input features. Due to the nature of the clustering algorithm, each jet has an embedded tree structure, where subjets are recombined with respect to a particular rule. Thus, CNNs and RNNs exploit different features of top and QCD jets to discriminate them from each other. We use the complementarity of both methods to combine them in an ENN that has an improved performance over both approaches individually. An implementation of the code we use for preprocessing and network training is provided at this link.[4]

3.1 Dataset & preprocessing

As a case study, we will investigate the top tagging capabilities of NNs by employing a CNN and a RNN. To achieve this, we used the dataset provided in [60, 80], which consists of 14 TeV top signal and mixed quark-gluon background jets generated and showered by Pythia 8 [81]. The detector simulation for showered events is obtained through the Delphes 3 package [82] using the default ATLAS detector card. The fat jets are reconstructed using `anti-kT` algorithm [83] as defiend in FastJet [84], using radius variable $R = 0.8$. The fat-jet transverse momentum has been limited to $[550, 650]$ GeV range in order to be able to benchmark the NN architectures precisely depending on the nature of jet substructure within a specific p_T-range. The resulting fat jets are further limited to be within $|\eta_j| < 2$. Finally, the fat jets in the top signal sample have been matched with truth level tops requiring $\Delta R(j, t_{truth}) < 0.8$. This dataset consists of 1.2 million training, 400,000 validation and test events respectively. This dataset has been divided into two subsets within our framework, one for CNN type training and one for RNN type training. For both of the datasets provided PFlow-objects are clustered into a fat-jet as described above.

The CNN dataset has been prepared with leading `anti-kT` fat jet constituents which are ordered by their corresponding transverse momentum. Each jet in the event has been centred with respect to total p_T weighted centroid where the jet vector has been centred at $(\eta, \phi) = (0, 0)$. Furthermore, the coordinate system has been rotated such that the principal axis is at the direction of positive pseudo-rapidity for all constituents. These modified constituents are fitted into pixels on $\eta - \phi$ plane, divided into 37×37 pixels between $(\eta, \phi) = ([-1.5, 1.5], [-1.5, 1.5])$ where the pixel value has been set as total p_T within that pixel. To

[4] https://gitlab.com/jackaraz/EnsembleNN.

JHEP04(2021)296

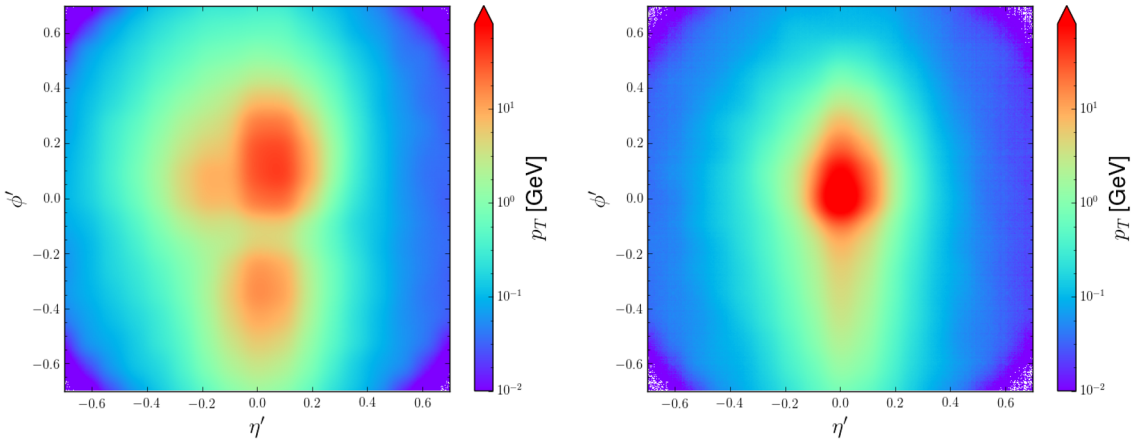

Figure 2. Left panel shows averaged top signal image on modified $\eta - \phi$ plane and the left panel shows the same for dijet sample. Colour represents the combined transverse momentum of the constituents within a pixel. Each image includes 10,000 events.

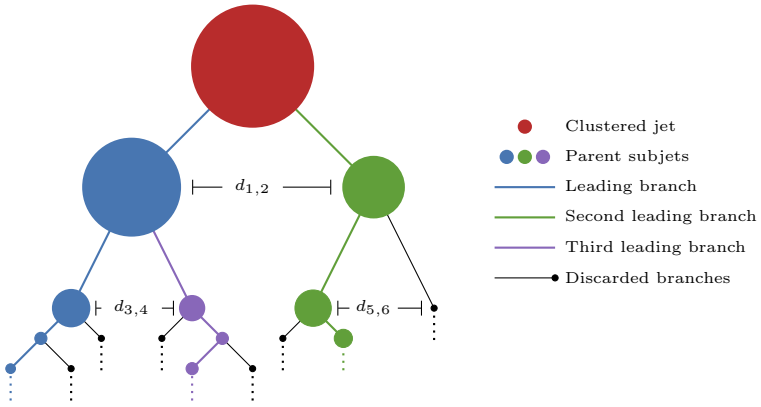

Figure 3. A schematic representation of the cluster history where blue represents leading branch with respect to the relative magnitude of transverse momentum, green is the second leading branch and purple is the third leading branch. Black lines shows the discarded branches. Finally dark red represents the initial clustered jet. The size of the circles represents the relative magnitude of transverse momentum.

get the leading constituent into the first quadrant, the vertical half of the image with higher total p_T flipped to the right, and similarly, the horizontal half of the image with higher p_T flipped to the top. Figure 2 shows the averaged top signal (left) and dijet background (right) images for $10,000$ events projected on modified $\eta - \phi$-plane. Colour represents the magnitude of the transverse momentum in the corresponding pixel. Note, this image has been zoomed-in to highlight the relevant portion of the image. Since the network requires the input data within $[0, 1]$ range, each image has been normalized by 1 TeV before training.

The RNN dataset has been constructed using leading `anti-kT` fat-jet where the constituents of the this jet are re-clustered with the same radius parameter using the `Cambridge/Aachen` (`C/A`) clustering algorithm [85]. In order to construct the training sequence, three leading branches have been extracted from the clustering history where

JHEP04(2021)296

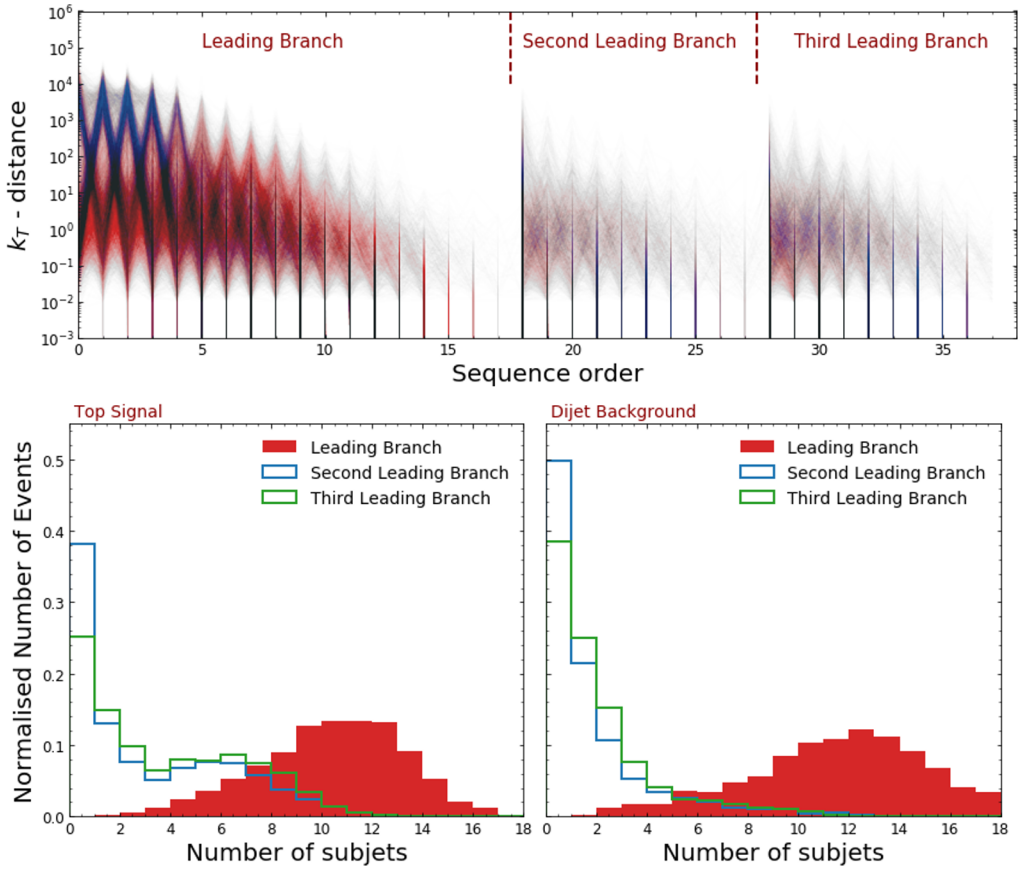

Figure 4. Top panel shows combined k_T-distances in RNN sequence for 4000 events. Red represents the dijet events and blue represents top signal events. Dominated colours shows which event has high occurance in a particular sequence. Bottom two panel shows the number of subjets in each branch where left panel shows it for top signal and right panel shows for dijet background.

their respective transverse momentum defined the branches. Initial two leading branches are constructed by the first two subjets in the clustering history where the subjet with larger p_T has been chosen to be the leading branch. The third leading branch has been chosen within the parent subjets of the first leading subjet. The parent with the lowest p_T is considered as the third leading branch. Figure 3 shows a schematic representation of this selection where blue stands for the leading branch following the subjets with relatively higher momentum than the consecutive parent subjet. Green is the second leading branch and purple is the third leading branch following the same pattern as the leading branch. Black lines represent the discarded branches which have less p_T compared to the corresponding parent subjet. Finally, red represents the `C/A`-jet. The sequence has been constructed using k_T-distances in the clustering history, defined as

$$d_{i,j} = \min\left(p_{T,i}^2, p_{T,j}^2\right) \frac{\Delta R^2}{R} \ .$$

Here i, j is the number of the parent subjets, ΔR is the relative angular distance between two subjets and R is the clustering radius given as 0.8. For each parent subjet in a branch,

JHEP04(2021)296

the $d_{i,j}$ value is stored with its chronological order. $d_{1,2}$ and $d_{3,4}$, see figure 3, are included as part of the leading branch sequence. In order to compose the RNN sequence, we first used the mass of the `anti-kT`-jet and then the mass of `C/A`-jet constructed using Mass Drop Tagger [86] ($\mu = 0.8$, $y_{\text{cut}} = 0.09$). Then we added the first 18, 10, 10 k_T-distances of the leading, second leading and third-leading branches, respectively. Branches with fewer subjets then padded with zeros. Upper panel of figure 4 shows the k_T sequence for 2000 top signal and 2000 dijet background events. Each event has been represented via high transparency; hence the vibrant colours show the high occurrences of the particular events where blue and red stands for top and dijet samples. The bottom two panels of figure 4 show the number of subjets in each branch where the left and right panels show for top and dijet samples, respectively.[5] Before passing the input feature vectors to the network for training, the dataset has been standardized using `RobustScaler` within Scikit-Learn package [87] using 100,000 mixed events from the training sample.

3.2 Network architecture & training

In order to study the effects of ensembling multiple architectures, here we will first introduce two "comparable" but independent architectures for the CNN and RNN-type of datasets presented in section 3.1. Our NN architecture relies on Keras library [88] embedded in TensorFlow version 2.2 [89].

The CNN dataset has been trained by a network receiving 37×37-pixel input via a 2D convolutional layer with eight features and four stride pixels alongside with zero paddings. This layer's output is normalized within a batch normalization layer and passed on to a max-pooling layer with a pool size of 2×2, leaving a reduced 18×18 image with eight features. Finally, these images have been flattened and passed to a fully connected dense layer with sixteen nodes with a dropout probability of 25%. A rectified linear unit (`ReLu`) activation function has been used for each layer. A dense output layer has then followed the network with a single node and sigmoid activation for classification.

Furthermore, the RNN dataset has been trained in a slightly more complex architecture starting with an LSTM layer, including 64 nodes. The activation and recurrent activation function for the LSTM layer have been chosen as hyperbolic tangent and sigmoid functions. It has been followed by three fully connected dense layer with 64, 64 and 32 nodes respectively and each dense layer followed by a dropout layer with 25% probability. As before, the `ReLu` activation function has been used for each dense layer. The network output has been generated from a final dense layer with a single node and sigmoid activation function.

Both networks are aimed to minimize a binary cross-entropy loss function via `Adam` optimizer [90] with a learning rate of 10^{-4}. Networks are trained for 500 epochs, and the learning rate has been reduced half for every 20 epochs if there is no improvement on the validation dataset's loss value. If the network didn't improve the validation loss for 250 epochs, the training terminated automatically.

[5] It is important to note that we also test our sequence by constructing it out of jets clustered by `kT` and `anti-kT` algorithms; however, the discriminative power has been observed to be less than the sequence clustered by `C/A` algorithm.

JHEP04(2021)296

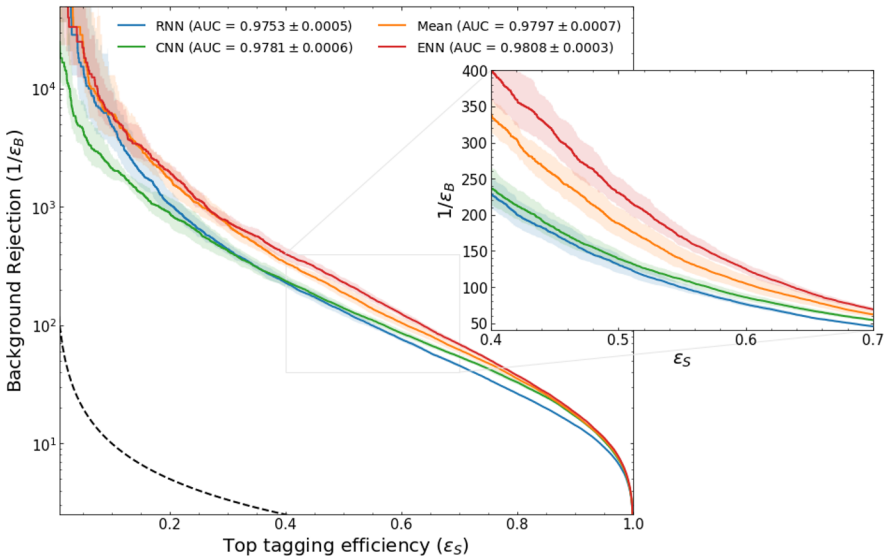

Figure 5. Receiver operating characteristic curve has been shown where CNN, RNN, the mean prediction of both and ENN architectures represented by green, blue, orange and red curves. The epistemic uncertainty has been represented by the transparent area around each curve for one standard deviation. Black curve represents the random choice. The inner plot zooms into the slice of $\varepsilon_S \in [0.4, 0.7]$.

Since the goal of this study is to question if a more extensive representation can generalize the given problem much better than its component hypotheses, we employed two types of ensembling methods. As a reference case, we studied the mean of both CNN and RNN predictions. As mentioned in section 1, such ensembles have shown to go beyond the accuracies of their component networks. For the main case, we will study an extended architecture where CNN and RNN architectures are concatenated before their output layer; hence resulting in a latent-space of 48 features. To find an optimal generalization of this latent-feature space, they are further connected to a fully connected dense layer with 96 nodes, employing `ReLu` activation function and L2 kernel regularization with a penalty strength of 0.05. This dense layer has been padded with 25% dropout layers before and after. Then connected to an output layer as before, activated via a sigmoid function.

In order to estimate the inherent uncertainty on each model, the test data has been divided into randomly selected 50,000 non-overlapping partitions. Figure 5 shows the ROC curve for each model. RNN and CNN are represented with blue and green curves alongside the inherent uncertainty for one standard deviation. The orange curve shows the mean prediction of these two models, which already indicates a higher generalization power than each component network. Finally, the red curve shows the minimalistic ENN configuration. Although the concatenated latent-feature space's training is minimal, it still reveals improvement beyond the mean prediction. The inner plot of figure 5 zooms into the slice of tagging efficiency within $[0.4, 0.7]$ to emphasize this improvement. Figure 5 also shows the area under the curve (AUC) value for each curve where the improvement in mean prediction and ENN is also visible.

JHEP04(2021)296

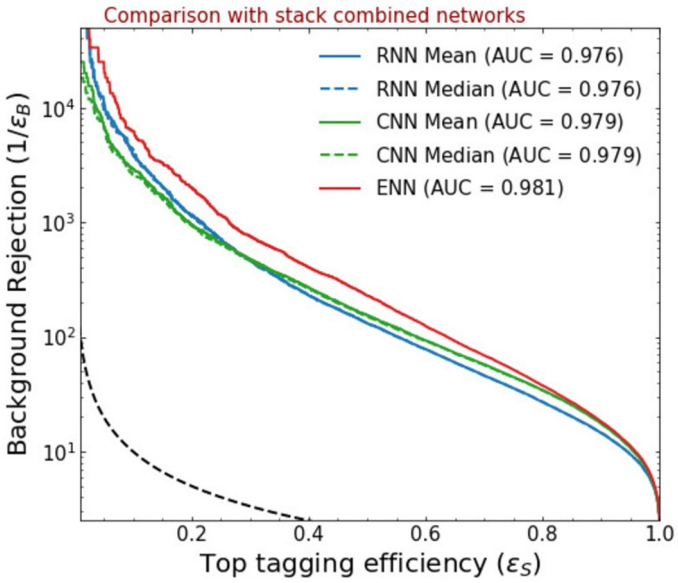

Figure 6. Comparison between stack and parallel combined networks have been shown. The receiver operating characteristic curve for CNN (green) and RNN (blue) shows the stack combined results for ten randomly initialized networks. The solid line shows the ROC curve for the predictions' mean, and the dashed line shows the same for the median of the predictions. The solid red line shows the ROC curve for the parallel combined ensemble network. The black curve represents a random choice.

As mentioned before, for the ENN to show a significant performance improvement over all pooled networks, it is important for the component networks to show mutually a comparable performance. As seen from figure 5, both the ENN and the mean prediction is dominated by the CNN above a tagging efficiency of 0.8 and dominated by the RNN below a tagging efficiency of 0.15. This shows that the performance of the network is solely dependent on the correlation between the component networks. In the region $\varepsilon_S \in [0.8, 1]$ both RNN and CNN captures the 3-prong substructure presented in figure 2 and 4 where in the region $\varepsilon_S \in [0, 0.3]$ a dipole type substructure is captured. Hence, both networks are highly correlated, which reflects in ENN's prediction as well. As seen from the interval $[0.4, 0.7]$ of the ROC curve, the ENN-improvement is maximized when the component accuracies are similar.

In ref. [60], it has been reported that, depending on the architecture, it is possible to improve the prediction quality up to 15% by using the stack combination method. To assess the parallel combining method's performance with respect to the stack combining method, we retrained RNN and CNN architectures ten times by reinitializing the networks' weights for each training. Then the mean(median)-of-ensemble calculated by combing the predictions of each training. Figure 6 shows the comparison between the parallel combined ensemble method (solid red curve) and stacked combined RNN (blue) and CNN (green) architectures. The mean and median combination has been shown with solid and dashed curves. Although we observe a slight improvement in the performance of the stack combined

JHEP04(2021)296

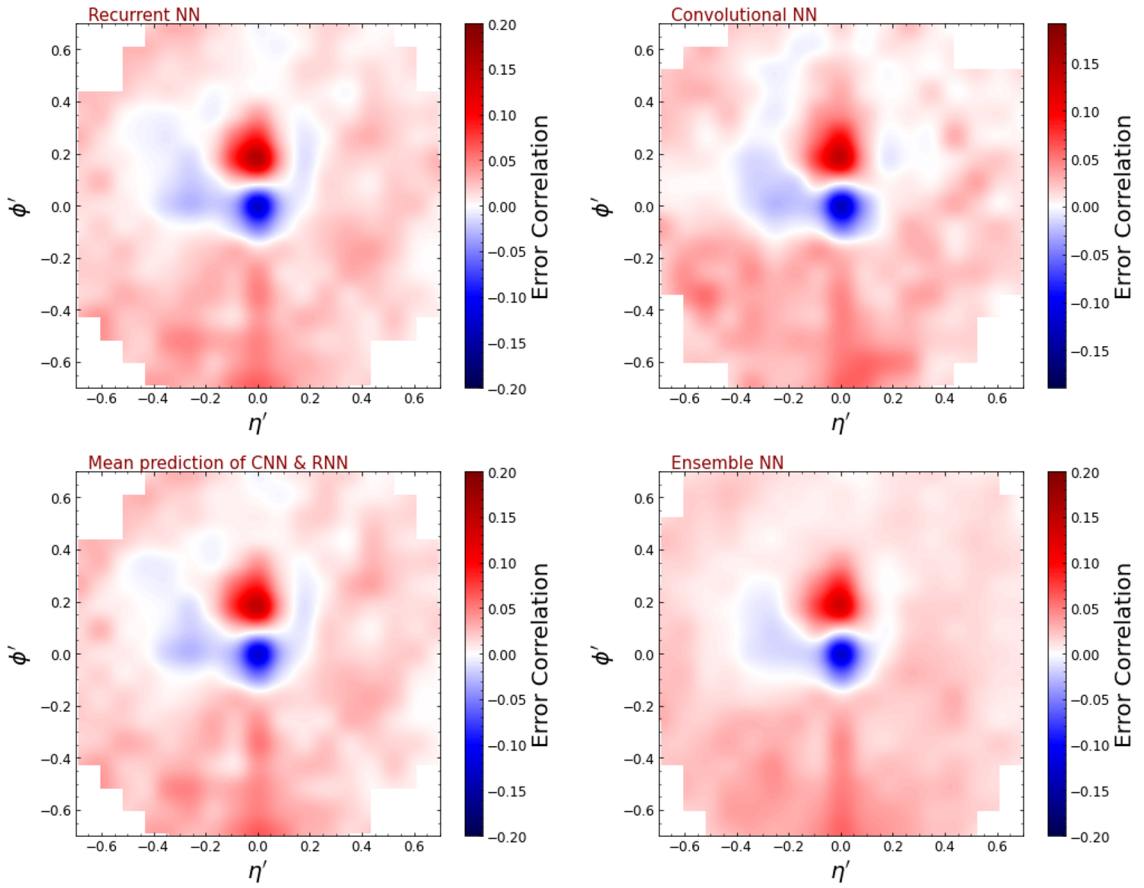

Figure 7. Squared error correlation mapped on 50,000 randomly selected test images for RNN (upper left), CNN (upper right), mean (lower left) and ENN (lower right).

networks with respect to their component network, this improvement does not match with the ENN architecture that this study proposes. We also do not observe a significant difference between using the mean or median for the stack combined methods.

As discussed in section 2, combining hypotheses with non-correlated errors may improve an ensemble's prediction. In order to test this, figure 7 shows the correlations of the squared error, $(y - \hat{y})^2$ mapped on the test images where y is the truth label and $\hat{y}$ is the prediction of the corresponding network. Figure 7 shows RNN (upper left panel), CNN (upper right panel), mean prediction (lower left panel) and ENN (lower right panel). Each correlation has been estimated by using randomly selected 50,000 test images. One can immediately see the shrinking area of the blue negative correlation distribution. Although the correlations between the RNN and the CNN mapping look similar, the mean prediction improves the two hypotheses' non-overlapping portions. The ENN goes beyond the mean prediction's achievement by drastically shrinking the blue region and removing the fluctuations in the red (positively correlated) region, hence increasing the correlations between squared error and the image pixels. As expected, similarly correlated regions changed neither for mean prediction nor for ENNs. Thus, combining all available neural

JHEP04(2021)296

networks would not improve the accuracy if their error is highly correlated. Instead, one can benefit from this methodology by employing networks with comparable accuracies and different error correlation to improve the latent-feature space accuracy.

4 Bayesian Deep Learning

For all phenomenological applications it is important to assess the intrinsic uncertainties of a NN model. Two major uncertainties can be modelled within the context of DL [62, 69]. The irreducible noise in the observations called aleatoric uncertainties and the uncertainties intrinsic to the proposed hypothesis called epistemic uncertainties. Given sufficient data, epistemic uncertainties can be explained and reduced. The decomposition of the variance of a binary hypothesis is given as [91, 92],

$$Var(y) \;=\; \underbrace{\langle \hat{y}^2 \rangle - \langle \hat{y} \rangle^2}_{\text{epistemic}} \;+\; \underbrace{\langle\; \hat{y}\;(1-\hat{y})\;\rangle}_{\text{aleatoric}}, \tag{4.1}$$

where $\hat{y}$ represents the network's predictive distribution, the first term represents the epistemic uncertainties while the second term is the aleatoric uncertainty. In addition to the uncertainties, the entropy of the network's prediction, also, gives strong indications about the underlying uncertainties of the system where higher entropy points to higher uncertainty. The entropy of binary classification is given as [93],

$$\mathcal{S} = -\left(\hat{y}\log_2\left(\hat{y}\right) + (1-\hat{y})\log_2\left(1-\hat{y}\right)\right)\,, \tag{4.2}$$

where the first term stands for the classification of the class 1 (top signal) and the second term for the classification of class 0 (dijet background).

In order to analyse the uncertainties underlying our neural network, we used the TensorFlow Probability package version 0.10.0 [94]. We limited ourselves to prediction uncertainties by only changing each network's output layer to Dense Flipout layer [95] with sigmoid activation.[6] The kernel divergence function has been chosen to be mean Kullbeck-Leiber divergence. We employed the same network architectures presented in section 3.2. As before, all networks are trained for 500 epochs with `Adam` optimizer. The initial learning rate has been given as 10^{-4} and reduced to its half in every 20 epochs if validation loss has not been improved. The final prediction has been reported using randomly chosen 100,000 test samples where each network output has been sampled 100 times.

Although the notion of "mean prediction" is ambiguous in the Bayesian context, in order to have a baseline, we defined the mean prediction of CNN and RNN networks as the mean of each 100 samples. This serves as the linear combination ensemble baseline which has not been trained on any latent-feature space beyond its component networks. To reveal our ensembling technique's full effect, we used an ensemble learner with one dense layer including 96 nodes, as before, and another ensemble learner with an additional dense layer with 96 nodes.[7]

[6] It is important to note here that, to get the complete model uncertainties from each layer, one can modify the entire network with Bayesian layers. This will double the number of trainable parameters in

JHEP04(2021)296

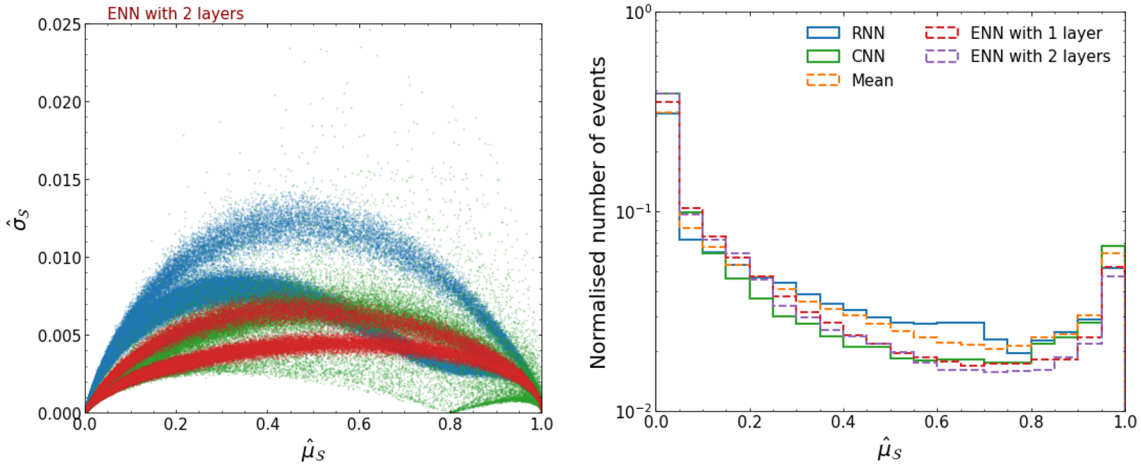

Figure 8. Mean entropy distribution with respect to the standard deviation of the entropy for RNN (blue), CNN (green) and ENN (red) where ensemble having two dense layers (left panel). Mean entropy distribution with respect to percentage of binned events (right panel).

	RNN	CNN	Mean	ENN (1 layer)	ENN (2 layers)
$\hat{\mu}_S < 0.5$	71.92%	75.22%	72.61%	78.05%	79.55%

Table 1. Percentage of events for each network structure, i.e. RNN, CNN, ENN and Mean, with mean entropy below 0.5.

The left panel of figure 8 shows the mean entropy, $\hat{\mu}_{\mathcal{S}}$, distribution with respect to the standard deviation in entropy, $\hat{\sigma}_{\mathcal{S}}$, where RNN, CNN and two-layer ENN has been represented with blue, green and red points. In order to simplify the plot, the mean prediction and the one-layer ENN model is not shown. One can immediately conclude that the ensemble learner has a considerable limitation on the standard deviation of the entropy where CNN reaches beyond 0.025, RNN to 0.015 but ENN limits the standard deviation below 0.0075. The right panel of figure 8 shows the percentage of events per mean entropy. As before, the RNN and CNN architectures are represented by blue and green solid curves. The separation between two curves increases between the entropy values $0.2 - 0.8$ where RNN has been observed to have more events with mid-range entropy values than CNN, but the last bin reveals that CNN has more events with maximum entropy. The dashed orange curve represents the mean of the two predictions where only slight improvement has been observed beyond the RNN. Furthermore, for the two ensemble learners, represented by dashed red and purple curves, one can immediately see the reduction in the number of events for the mid-range entropy values. One can also see that when sufficient complexity is provided, an ensemble learner further improves the hypothesis's entropy, i.e. reduces its

each layer. Thus in order to simplify our problem, we are only concentrating on prediction uncertainties.

[7]It is important to note that we did not observe a significant improvement over ROC AUC by adding an extra dense layer. Thus further optimization beyond adding an extra layer required to improve the accuracy of an ensemble learner. Since this is beyond our scope, we limit ourselves to simplistic architecture.

JHEP04(2021)296

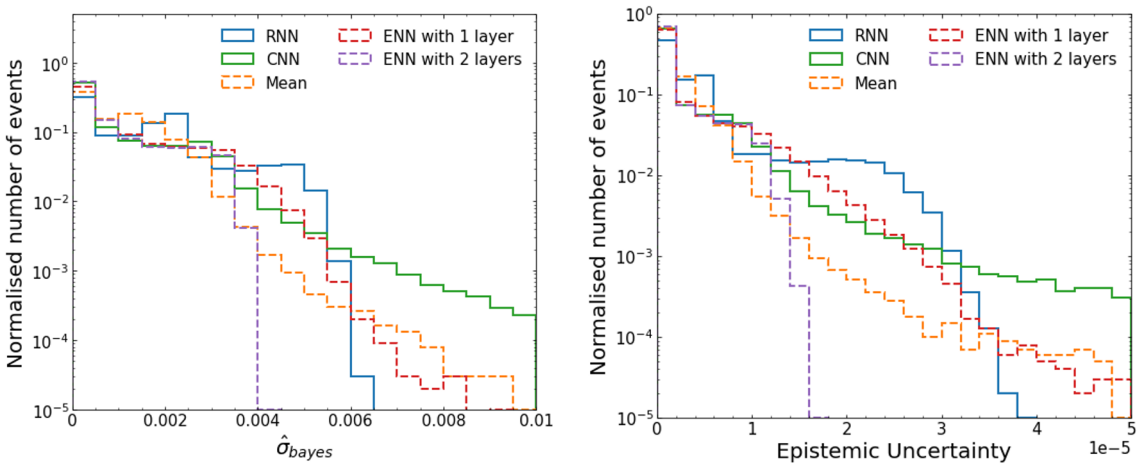

Figure 9. Left panel shows the normalised number of events per standard deviation in prediction. Right panel shows the same for epistemic uncertainty. In each histogram RNN, CNN, mean, one-layer ENN and two-layer ENN has been represented with blue, green, orange, red and purple curves.

values for both $\hat{\mu}_{\mathcal{S}}$ and $\hat{\sigma}_{\mathcal{S}}$. This is also summarized in table 1, where more than 78% of the events for both ensemble learners reach a mean entropy $\hat{\mu}_{\mathcal{S}}$ of less than 0.5, while RNN, CNN, and mean prediction remain below 75.3%.

We also analyzed the standard deviation in the hypothesis prediction, which is crucial to maintain small in order to achieve consistent predictions. The left panel of figure 9 shows the fraction of events per standard deviation in prediction where the same colour scheme applied as before. Given a sufficiently complexity problem, the ENN is observed to reduce $\hat{\sigma}_{bayes}$ significantly, compared to each component network and the mean combination of those networks respectively. While the mean prediction reaching up to $\hat{\sigma}_{bayes} \sim 0.01$, the ENN limits the standard deviation below 0.004, which is similar to the standard deviation mean entropy. On the right panel of the figure 9, we show the epistemic uncertainty as given in the first term of eq. (4.1) using the same colour labelling. Again, we find a significant reduction of the uncertainties with ensemble learners. These results show that learning over various symmetries leads to a more accurate representation of the given problem without requiring more data.

An optimization problem requires a sufficient amount of training examples in order to be able to generalize the given hypothesis successfully. As shown in ref. [62], lack of variety in training examples will cause uncertainty and the standard deviation of the prediction to increase. Figure 10 shows the change in the standard deviation in the prediction of CNN (green), RNN (blue) and two-layer ENN (red) architectures. As before, each network output has been sampled 100 times for 100,000 test samples. The solid, dashed, and dotted lines show each network's prediction trained with 610,000, 300,000 and 200,000 randomly chosen training samples. It has been observed that while $\hat{\sigma}_{bayes}$ gets significantly larger in RNN and CNN architectures, ENN is less susceptible to the lack of training examples. This shows that the ability to access different symmetries within a given data provides the necessary tools for ENN architecture to generalize the hypothesis better with fewer training examples.

JHEP04(2021)296

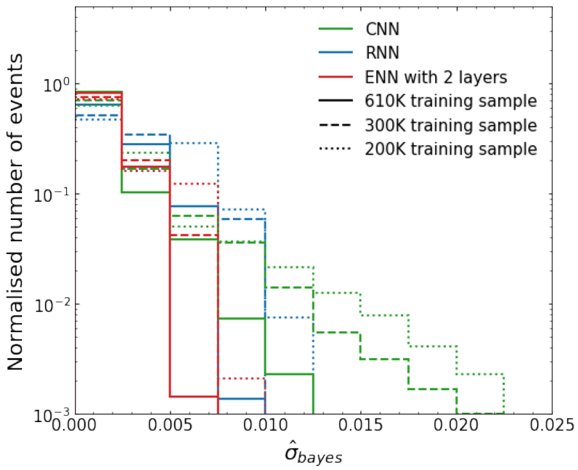

Figure 10. The histogram shows the effect of using different training sample sizes on the standard deviation of the prediction. Red, green and blue curves show ENN with two layers, CNN and RNN architectures and solid, dashed and dotted curves shows the training sample sizes of 610,000, 300,000 and 200,000, respectively.

Thus we observed that employing different domains of data that are specialised for specific properties, and optimising a neural network with combined properties of these component learners drastically reduces the system's uncertainties. Such an ensemble network has been shown to learn the system's correlations much more accurately compared to its individual component networks.

5 Conclusion

We presented Ensemble Neural Networks for the reconstruction and classification of collider events and applied them to the discrimination of boosted hadronically decaying top quarks from QCD jets. An ENN can improve the accuracy of the network beyond the individual contributions of its component networks by reducing the variance of the prediction given that the errors of component networks are not highly correlated. In this study, we showed that such techniques can be used in the event reconstruction of collider events in order to overcome the representation problem of neural networks and to improve the prediction accuracy and uncertainties.

Special-purpose networks, such as CNNs or RNNs have been repeatedly shown to be highly accurate for the classification of LHC events. These networks are specialised to learn and optimise their models with respect to the correlations of the given data. In the case of the classification of fat jets, these correlations can be represented through calorimeter images where a network learns the spatial distribution of a jet's constituents. On the other hand, clustering algorithms produce a sequential tree structures which can be employed to learn distinct kinematic features of top decays and QCD backgrounds. An ensemble learner is a paradigm that allows the combination of these properties in one algorithm. Instead of optimising the network separately with respect to the distinct

JHEP04(2021)296

symmetries of images or cluster sequences, it allows optimisation through combined latent-feature space. We showed that combining convolutional and recurrent neural networks and training the network further over their latent-feature space leads to higher accuracy for the classification task. Further, we found that such technique explicitly reduces the variations in error correlations of the component networks hence improving the domains where the component networks are not highly correlated.

Although ENN comes with a great advantage, it is crucial to emphasize the trade-off of building such an architecture. ENN is only valid if its component networks can capture different correlations in the data. As shown in section 3.2, ENN can not improve the regions where the errors of the components are highly correlated. Hence, in such cases, it would be equally beneficial to focus on improving the performance of individual state-of-the-art NN architectures. It is also important to note that this does not render the stack combining method invalid. For complex loss hypersurfaces, it is quite challenging for a learning algorithm to find the global minimum. If there is no other architecture available that can exploit different features of the given data, then the stack combining method will achieve a much closer approximation by sampling different regions of the hypothesis-space.

A detailed understanding of the inner workings of Deep Learning techniques is often missing. To develop confidence in their applicability in measurements and searches for new physics, it is of vital importance to understand and, if possible, reduce the uncertainties of the networks. Bayesian techniques are designed to quantify such uncertainties. We found that ENNs can drastically reduce the uncertainty in the prediction of the network, without increasing the amount of training data. We also showed that such methods reduce the entropy of the system as well as the epistemic uncertainties and it reduces the network's susceptible to small-sized training samples. ENNs can thus provide much more accurate predictions than their component networks. The methodology employed in this study can be applied to a broad scope of application in HEP phenomenology. Instead of expanding the data domain, learning through combined underlying correlations of the problem has been shown to be very effective.

While ensemble learners can reduce the variance of the hypothesis, we did not observe any improvement in the data's bias or aleatoric uncertainties. Although reducing the network's epistemic uncertainties and variance is a crucial step, aleatoric uncertainties are observed to be larger than the epistemic uncertainties. It has been shown that Genetic-Algorithm-based Selective Ensembles can reduce the biases as well as the variance of the system [50], it is an obvious next step to employ such techniques to reduce biases as well as the variance of the network.

Open Access. This article is distributed under the terms of the Creative Commons Attribution License (CC-BY 4.0), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

JHEP04(2021)296

References

[1] S. Marzani, G. Soyez and M. Spannowsky, *Looking inside jets: an introduction to jet substructure and boosted-object phenomenology*, *Lect. Notes Phys.* **958** (2019) 1 [arXiv:1901.10342] [INSPIRE].

[2] T. Plehn, G.P. Salam and M. Spannowsky, *Fat jets for a light Higgs*, *Phys. Rev. Lett.* **104** (2010) 111801 [arXiv:0910.5472] [INSPIRE].

[3] T. Plehn, M. Spannowsky, M. Takeuchi and D. Zerwas, *Stop reconstruction with tagged tops*, *JHEP* **10** (2010) 078 [arXiv:1006.2833] [INSPIRE].

[4] T. Plehn, M. Spannowsky and M. Takeuchi, *How to improve top tagging*, *Phys. Rev. D* **85** (2012) 034029 [arXiv:1111.5034] [INSPIRE].

[5] D.E. Soper and M. Spannowsky, *Finding top quarks with shower deconstruction*, *Phys. Rev. D* **87** (2013) 054012 [arXiv:1211.3140] [INSPIRE].

[6] D.E. Soper and M. Spannowsky, *Finding physics signals with shower deconstruction*, *Phys. Rev. D* **84** (2011) 074002 [arXiv:1102.3480] [INSPIRE].

[7] D.E. Soper and M. Spannowsky, *Finding physics signals with event deconstruction*, *Phys. Rev. D* **89** (2014) 094005 [arXiv:1402.1189] [INSPIRE].

[8] S. Prestel and M. Spannowsky, *HYTREES: combining matrix elements and parton shower for hypothesis testing*, *Eur. Phys. J. C* **79** (2019) 546 [arXiv:1901.11035] [INSPIRE].

[9] J. Brehmer, K. Cranmer, G. Louppe and J. Pavez, *Constraining effective field theories with machine learning*, *Phys. Rev. Lett.* **121** (2018) 111801 [arXiv:1805.00013] [INSPIRE].

[10] J. Brehmer, F. Kling, I. Espejo and K. Cranmer, *MadMiner: machine learning-based inference for particle physics*, *Comput. Softw. Big Sci.* **4** (2020) 3 [arXiv:1907.10621] [INSPIRE].

[11] G. Louppe, M. Kagan and K. Cranmer, *Learning to pivot with adversarial networks*, arXiv:1611.01046 [INSPIRE].

[12] C.K. Khosa and V. Sanz, *Anomaly awareness*, arXiv:2007.14462 [INSPIRE].

[13] L.G. Almeida, M. Backović, M. Cliche, S.J. Lee and M. Perelstein, *Playing tag with ANN: boosted top identification with pattern recognition*, *JHEP* **07** (2015) 086 [arXiv:1501.05968] [INSPIRE].

[14] 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].

[15] 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].

[16] J. Pearkes, W. Fedorko, A. Lister and C. Gay, *Jet constituents for deep neural network based top quark tagging*, arXiv:1704.02124 [INSPIRE].

[17] S. Egan, W. Fedorko, A. Lister, J. Pearkes and C. Gay, *Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC*, arXiv:1711.09059 [INSPIRE].

[18] S. Macaluso and D. Shih, *Pulling out all the tops with computer vision and deep learning*, *JHEP* **10** (2018) 121 [arXiv:1803.00107] [INSPIRE].

[19] 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].

JHEP04(2021)296

[20] L. Moore, K. Nordström, S. Varma and M. Fairbairn, *Reports of my demise are greatly exaggerated: N-subjettiness taggers take on jet images*, *SciPost Phys.* **7** (2019) 036 [arXiv:1807.04769] [INSPIRE].

[21] A. Blance, M. Spannowsky and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, *JHEP* **10** (2019) 047 [arXiv:1905.10384] [INSPIRE].

[22] S.H. Lim and M.M. Nojiri, *Spectral analysis of jet substructure with neural networks: boosted Higgs case*, *JHEP* **10** (2018) 181 [arXiv:1807.03312] [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] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, *Jet substructure classification in high-energy physics with deep neural networks*, *Phys. Rev. D* **93** (2016) 094034 [arXiv:1603.09349] [INSPIRE].

[25] G. Louppe, K. Cho, C. Becot and K. Cranmer, *QCD-aware recursive neural networks for jet physics*, *JHEP* **01** (2019) 057 [arXiv:1702.00748] [INSPIRE].

[26] J. Gallicchio and M.D. Schwartz, *Quark and gluon jet substructure*, *JHEP* **04** (2013) 090 [arXiv:1211.7038] [INSPIRE].

[27] 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].

[28] T. Cheng, *Recursive neural networks in quark/gluon tagging*, *Comput. Softw. Big Sci.* **2** (2018) 3 [arXiv:1711.02633] [INSPIRE].

[29] 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].

[30] S. Bright-Thonney and B. Nachman, *Investigating the topology dependence of quark and gluon jets*, *JHEP* **03** (2019) 098 [arXiv:1810.05653] [INSPIRE].

[31] A.J. Larkoski, I. Moult and B. Nachman, *Jet substructure at the Large Hadron Collider: a review of recent advances in theory and machine learning*, *Phys. Rept.* **841** (2020) 1 [arXiv:1709.04464] [INSPIRE].

[32] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images — deep learning edition*, *JHEP* **07** (2016) 069 [arXiv:1511.05190] [INSPIRE].

[33] O. Kitouni, B. Nachman, C. Weisser and M. Williams, *Enhancing searches for resonances with machine learning and moment decomposition*, *JHEP* **04** (2021) 070 [arXiv:2010.09745] [INSPIRE].

[34] X. Ju and B. Nachman, *Supervised jet clustering with graph neural networks for Lorentz boosted bosons*, *Phys. Rev. D* **102** (2020) 075014 [arXiv:2008.06064] [INSPIRE].

[35] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, arXiv:2008.06545 [INSPIRE].

[36] S. Farrell et al., *Next generation generative neural networks for HEP*, *EPJ Web Conf.* **214** (2019) 09005 [INSPIRE].

[37] J. Lin, W. Bhimji and B. Nachman, *Machine learning templates for QCD factorization in the search for physics beyond the Standard Model*, *JHEP* **05** (2019) 181 [arXiv:1903.02556] [INSPIRE].

[38] K. Datta, A. Larkoski and B. Nachman, *Automating the construction of jet observables with machine learning*, *Phys. Rev. D* **100** (2019) 095016 [arXiv:1902.07180] [INSPIRE].

JHEP04(2021)296

[39] R.T. D'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, *Learning multivariate new physics*, *Eur. Phys. J. C* **81** (2021) 89 [arXiv:1912.12155] [INSPIRE].

[40] R.T. D'Agnolo and A. Wulzer, *Learning new physics from a machine*, *Phys. Rev. D* **99** (2019) 015014 [arXiv:1806.02350] [INSPIRE].

[41] B. Nachman and J. Thaler, *E pluribus unum ex machina: learning from many collider events at once*, arXiv:2101.07263 [INSPIRE].

[42] T. Faucett, J. Thaler and D. Whiteson, *Mapping machine-learned physics into a human-readable space*, *Phys. Rev. D* **103** (2021) 036020 [arXiv:2010.11998] [INSPIRE].

[43] C.K. Khosa, L. Mars, J. Richards and V. Sanz, *Convolutional neural networks for direct detection of dark matter*, *J. Phys. G* **47** (2020) 095201 [arXiv:1911.09210] [INSPIRE].

[44] C.K. Khosa, V. Sanz and M. Soughton, *Using machine learning to disentangle LHC signatures of dark matter candidates*, arXiv:1910.06058 [INSPIRE].

[45] T.G. Dietterich, *Ensemble methods in machine learning*, in *Multiple classifier systems*, Springer, Berlin, Heidelberg, Germany (2000), pg. 1.

[46] L. Hansen and P. Salamon, *Neural network ensembles*, *IEEE Trans. Pattern Anal. Machine Intell.* **12** (1990) 993.

[47] A.L. Blum and R.L. Rivest, *Training a* 3*-node neural network is NP-complete*, *Neural Networks* **5** (1992) 117.

[48] K. Hornik, M. Stinchcombe and H. White, *Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks*, *Neural Networks* **3** (1990) 551.

[49] C. Englert, M. Fairbairn, M. Spannowsky, P. Stylianou and S. Varma, *Sensing Higgs boson cascade decays through memory*, *Phys. Rev. D* **102** (2020) 095027 [arXiv:2008.08611] [INSPIRE].

[50] Z.-H. Zhou, J. Wu and W. Tang, *Ensembling neural networks: many could be better than all*, *Artificial Intel.* **137** (2002) 239.

[51] A. Krogh and J. Vedelsby, *Neural network ensembles, cross validation and active learning*, in *Proceedings of the* 7^{th} *international conference on neural information processing systems, NIPS '*94, MIT Press, Cambridge, MA, U.S.A. (1994), pg. 231.

[52] M.P. Perrone and L.N. Cooper, *When networks disagree: ensemble methods for hybrid neural networks*, in *How we learn; how we remember: toward an understanding of brain and neural systems*, World Scientific, Singapore (1995), pg. 342.

[53] J. Xie, B. Xu and Z. Chuang, *Horizontal and vertical ensemble with deep representation for classification*, arXiv:1306.2759.

[54] L. Rokach, *Ensemble-based classifiers*, *Artificial Intel. Rev.* **33** (2009) 1.

[55] R.P.W. Duin and D.M.J. Tax, *Experiments with classifier combining rules*, in *Multiple classifier systems*, Springer, Berlin, Heidelberg, Germany (2000), pg. 16.

[56] J. Conrad and F. Tegenfeldt, *Applying rule ensembles to the search for super-symmetry at the Large Hadron Collider*, *JHEP* **07** (2006) 040 [hep-ph/0605106] [INSPIRE].

[57] P. Baldi, P. Sadowski and D. Whiteson, *Enhanced Higgs boson to* $\tau^+\tau^-$ *search with deep learning*, *Phys. Rev. Lett.* **114** (2015) 111801 [arXiv:1410.3469] [INSPIRE].

[58] A. Alves, *Stacking machine learning classifiers to identify Higgs bosons at the LHC*, 2017 *JINST* **12** T05005 [arXiv:1612.07725] [INSPIRE].

JHEP04(2021)296

[59] A. Alves and F.F. Freitas, *Towards recognizing the light facet of the Higgs boson*, *Mach. Learn. Sci. Tech.* **1** (2020) 045025 [arXiv:1912.12532] [INSPIRE].

[60] A. Butter et al., *The machine learning landscape of top taggers*, *SciPost Phys.* **7** (2019) 014 [arXiv:1902.09914] [INSPIRE].

[61] N. Ueda and R. Nakano, *Generalization error of ensemble estimators*, in *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, IEEE, (1996), pg. 90.

[62] S. Bollweg, M. Haußmann, G. Kasieczka, M. Luchmann, T. Plehn and J. Thompson, *Deep-learning jets with uncertainties and more*, *SciPost Phys.* **8** (2020) 006 [arXiv:1904.10004] [INSPIRE].

[63] S. Marshall et al., *Using Bayesian optimization to find asteroids' pole directions*, *AAS/Division for Planetary Sciences Meeting Abstracts* **50** (2018) 505.01D.

[64] J. Mukhoti, P. Stenetorp and Y. Gal, *On the importance of strong baselines in Bayesian deep learning*, arXiv:1811.09385.

[65] B. Nachman, *A guide for deploying deep learning in LHC searches: how to achieve optimality and account for uncertainty*, *SciPost Phys.* **8** (2020) 090 [arXiv:1909.03081] [INSPIRE].

[66] B. Nachman and J. Thaler, *Neural resampler for Monte Carlo reweighting with preserved uncertainties*, *Phys. Rev. D* **102** (2020) 076004 [arXiv:2007.11586] [INSPIRE].

[67] C. Englert, P. Galler, P. Harris and M. Spannowsky, *Machine learning uncertainties with adversarial neural networks*, *Eur. Phys. J. C* **79** (2019) 4 [arXiv:1807.08763] [INSPIRE].

[68] Y. Gal and Z. Ghahramani, *Dropout as a bayesian approximation: representing model uncertainty in deep learning*, arXiv:1506.02142.

[69] A. Kendall and Y. Gal, *What uncertainties do we need in bayesian deep learning for computer vision?*, arXiv:1703.04977.

[70] J.F. Kolen and J.B. Pollack, *Back propagation is sensitive to initial conditions*, in *Proceedings of the* 3rd *International Conference on Neural Information Processing Systems, NIPS'90*, Morgan Kaufmann Publishers Inc., San Francisco, CA, U.S.A. (1990), pg. 860.

[71] K. Cherkauer, *Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks*, in *Working notes of the AAAI workshop on integrating multiple learned models*, (1996), pg. 15.

[72] K. Tumer and J. Ghosh, *Error correlation and error reduction in ensemble classifiers*, *Connection Sci.* **8** (1996) 385.

[73] L. Breiman, *Bagging predictors*, *Machine Learn.* **24** (1996) 123.

[74] M. Gams, *New measurements highlight the importance of redundant knowledge*, in *Proceedings of the fourth european working session on learning*, (1989), pg. 71.

[75] B. Parmanto, P. Munro and H. Doyle, *Improving committee diagnosis with resampling techniques*, in *Advances in neural information processing systems*, volume 8, D. Touretzky, M.C. Mozer and M. Hasselmo eds., MIT Press, U.S.A. (1996), pg. 882.

[76] Y. Freund and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, *J. Comput. Syst. Sci.* **55** (1997) 119.

[77] Y. Freund and R.E. Schapire, *Experiments with a new boosting algorithm*, in *Proceedings of the thirteenth international conference on machine learning*, Morgan Kaufmann, San Francisco, CA, U.S.A. (1996), pg. 148.

JHEP04(2021)296

[78] G. Brown, J.L. Wyatt and P. Tiňo, *Managing diversity in regression ensembles*, *J. Mach. Learn. Res.* **6** (2005) 1621.

[79] P. Domingos, *A unifeid bias-variance decomposition and its applications*, in *Proceedings of the seventeenth international conference on machine learning, ICML '00*, Morgan Kaufmann, San Francisco, CA, U.S.A. (2000), pg. 231.

[80] G. Kasieczka, T. Plehn, J. Thompson and M. Russel, *Top quark tagging reference dataset*, Zenodo, March 2019.

[81] T. Sjöstrand et al., *An introduction to PYTHIA* 8.2, *Comput. Phys. Commun.* **191** (2015) 159 [arXiv:1410.3012] [INSPIRE].

[82] DELPHES 3 collaboration, *DELPHES 3, a modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [arXiv:1307.6346] [INSPIRE].

[83] M. Cacciari, G.P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, *JHEP* **04** (2008) 063 [arXiv:0802.1189] [INSPIRE].

[84] M. Cacciari, G.P. Salam and G. Soyez, *FastJet user manual*, *Eur. Phys. J. C* **72** (2012) 1896 [arXiv:1111.6097] [INSPIRE].

[85] S. Bentvelsen and I. Meyer, *The Cambridge jet algorithm: features and applications*, *Eur. Phys. J. C* **4** (1998) 623 [hep-ph/9803322] [INSPIRE].

[86] J.M. Butterworth, A.R. Davison, M. Rubin and G.P. Salam, *Jet substructure as a new Higgs search channel at the LHC*, *Phys. Rev. Lett.* **100** (2008) 242001 [arXiv:0802.2470] [INSPIRE].

[87] F. Pedregosa et al., *Scikit-learn: machine learning in Python*, *J. Mach. Learn. Res.* **12** (2011) 2825.

[88] F. Chollet et al., *Keras*, https://keras.io, (2015).

[89] M. Abadi et al., *TensorFlow: large-scale machine learning on heterogeneous distributed systems*, arXiv:1603.04467 [INSPIRE].

[90] D.P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, arXiv:1412.6980 [INSPIRE].

[91] Y. Kwon, J.-H. Won, B.J. Kim and M.C. Paik, *Uncertainty quantification using Bayesian neural networks in classification: application to biomedical image segmentation*, *Comput. Statist. Data Anal.* **142** (2020) 106816.

[92] N. Tagasovska and D. Lopez-Paz, *Single-model uncertainties for deep learning*, arXiv:1811.00908.

[93] D.J.C. MacKay, *Information theory, inference & learning algorithms*, Cambridge University Press, Cambridge, U.K. (2002).

[94] M. Abadi et al., *Tensorflow: a system for large-scale machine learning*, arXiv:1605.08695.

[95] Y. Wen, P. Vicol, J. Ba, D. Tran and R.B. Grosse, *Flipout: efficient pseudo-independent weight perturbations on mini-batches*, arXiv:1803.04386.

JHEP04(2021)296