

---

# Using Regression Neural Networks to correct the $b\bar{b}$ Mass in Higgs Decay

---



Ana Alexandre

University of Manchester

DESY Summer Programme Report

Supervised by Krisztian Peters and Jonathan Burr

September 4, 2019

## Abstract

Different neural networks were used to approximate the truth channel  $m_{b\bar{b}}$  distribution using the reco channel kinematical variables as input. Different targets and loss functions were considered and a comparative analysis was performed. A neural network training towards the difference between the reco and the truth channel variables, with a MAE loss, was the best of all the networks considered. It improved the peak location, from between 120GeV and 122GeV, to between 126GeV and 128GeV and improved the reco mean of 112.3GeV and standard deviation of 19.4GeV to a mean of 117.0GeV and a standard deviation of 18.6GeV. No other network was capable of improving the reco channel  $m_{b\bar{b}}$  distribution.

# 1 Introduction

Different measurements of the total mass of the universe have been inconsistent. In particular, luminosity estimates, which take only into account the visible universe, predict a lower mass than dynamical estimates, which consider only dynamical quantities such as rotation velocity. These discrepancies suggest that matter in the universe is not only visible matter, but also Dark Matter (DM). Observations of rotation curves of galaxies, which relate the circular velocity of the gas and the stars to their distance to the centre of the galaxy, weak gravitational lensing, the Oort discrepancy in the disk of the Milky Way and Weak modulation of strong lensing, all provide strong evidence for the presence of DM as the dominant form of matter in the universe [?]. However, the nature of DM remains an unsolved mystery of modern physics.

One theoretical possibility is that DM consists of electrically neutral, stable particles that interact weakly with the Standard Model (SM) particles. This theory is consistent with observations of the relic abundance of DM [?] and the Large Hadron Collider (LHC) could potentially find evidence of the predicted DM particles [?]. However, in order to successfully identify these particles, better analysis methods must be developed. In particular, in DM models where DM particles are produced along with a Higgs boson, it is necessary to successfully isolate a Higgs particle from the numerous background events [?].

Machine learning techniques have proven to be useful and powerful tools in several different areas, including in LHC physics analysis, having been used, for example, to determine particle properties, to improve the resolution of the CMS electromagnetic calorimeter and to isolate the  $H \rightarrow t\bar{t}$  decay signal [?]. In this report, the possibility to use machine learning techniques to correct the invariant mass of the Higgs decay to  $b\bar{b}$ , in order to better isolate the  $H \rightarrow b\bar{b}$  signal, is studied.

## 2 Searching for Dark Matter in the ATLAS Detector

### 2.1 The ATLAS Detector

ATLAS (A Toroidal LHC ApparatuS) is a general-purpose detector built to study the  $pp$  collisions at the LHC at CERN. ATLAS is forward-backward symmetric with respect to the interaction point and it covers a solid angle of almost  $4\pi$  [?]. The coordinate system used in the detector is centred at the interaction point, with the  $z$ -axis in the beam direction, and the  $x$  and  $y$ -axes are transverse to the beam direction, with the  $x$ -axis pointing from the origin to the centre of the LHC ring. The more commonly used coordinates are  $\theta$ , the polar angle, measured from the beam axis,  $\phi$ , the azimuthal angle, measured around the beam axis, and the pseudorapidity, defined as  $\eta = -\ln \tan(\theta/2)$ . The transverse momentum,  $p_T$ , is defined in the  $xy$  plane [?].

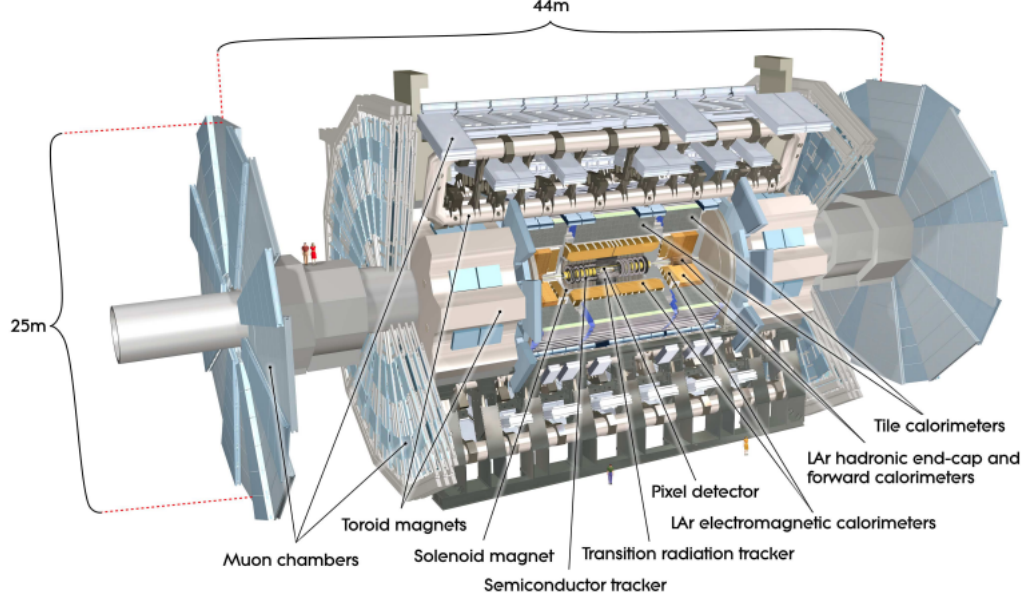


Figure 1: The ATLAS detector

The detector is made up of an inner detector, responsible for reconstructing charged-particle tracks and vertices, at  $|\eta| < 2.5$ , a calorimeter, covering  $|\eta| < 4.9$ , which measures electron, photon, jet and  $\tau$  lepton energies, and a muon spectrometer, with tracking chambers covering  $|\eta| < 2.7$  and detectors covering  $|\eta| < 2.4$ , meant to identify muons and measure their momentum. Around the inner detector, there is a thin superconducting solenoid, creating an homogeneous 2T magnetic field, and, around the calorimeters, arranged in an eightfold azimuthal symmetry, there are three large superconducting toroids. Figure ?? shows a schematic view of the ATLAS detector [?].

## 2.2 The $Z'$ 2HDM Model

The signal model considered in this report for the detection of DM particles is referred to as the  $Z'$  2HDM model and it is a Type-II two-Higgs doublet model with an additional  $U(1)_{Z'}$  gauge symmetry. In this model, a proton collision will produce a particle  $Z'$  which will produce a light scalar Higgs boson,  $h$ , and a pseudo-scalar Higgs boson,  $A$ , two of the five physical Higgs bosons present in the model. The particle  $A$  will then decay into DM particles  $\chi$  and  $\bar{\chi}$  [?]. These particles are invisible to the detector and thus result in missing momentum of magnitude denoted by  $E_T^{miss}$ . The most common decay channel of the Higgs  $h$  is to  $b\bar{b}$ , with a SM branching ratio  $BR = 57\%$  [?]. Therefore, DM particles,  $\chi$ , are produced along with a  $b\bar{b}$  pair, which is depicted in Figure ??, with an invariant mass of  $m_{b\bar{b}} = m_h$  and the detection of the signature  $b\bar{b} + E_T^{miss}$  is a good indication of DM production through this process [?].

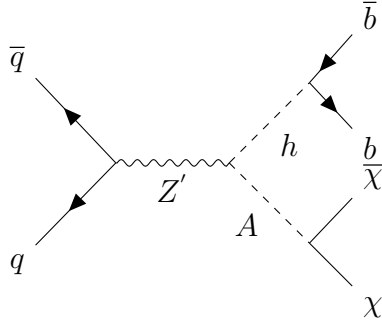


Figure 2: Feynman diagram of the leading order production of DM particles  $\chi$ , mediated by  $Z'$ , with production of a Higgs boson decaying into  $b\bar{b}$ .

The production of top-quark pairs,  $t\bar{t}$ , and the production of  $W$  and  $Z$  bosons with additional  $b\bar{b}$  production also result in the signature  $b\bar{b} + E_T^{miss}$  and, therefore, constitute the main source of SM background for this process [?]. Furthermore, a  $b$  quark can decay into a  $W^-$  boson, which can then decay into neutrinos,  $\nu_\mu$ ,  $\nu_\tau$  or  $\nu_e$ , muons,  $\mu$ , and  $\tau$  leptons, which can also decay into neutrinos and another  $W^-$  boson. All these processes are depicted in Figure ???. Neutrinos are not seen in the detector and muons are only detected by the muon spectrometer, whose information is not included for this analysis. Thus, interactions with muons and neutrinos as final products result in energy lost from the jet. As a result, the reconstructed momentum of the b-jets will be lower and so will the invariant mass of  $b\bar{b}$ . Consequently, the invariant mass distribution will be wider on the left side of the peak, as there will be a higher number of predictions lower than the Higgs mass. Both the background events and the wider peak due to lost energy make it extremely difficult to resolve the Higgs mass peak.

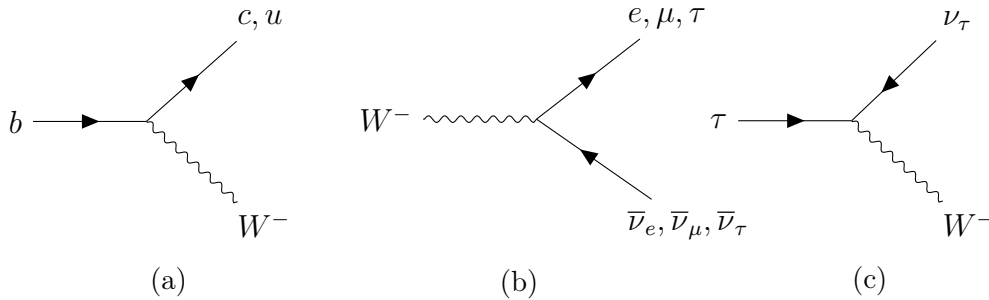


Figure 3: (a)  $b$  (b)  $W^-$  and (c)  $\tau$  decays, leading to the production of neutrinos and muons which result in  $E_T^{miss}$

## 2.3 Simulations and Jet Reconstruction

Using Monte Carlo algorithms, it is possible to simulate the interaction represented in Figure ?? and, in addition, it is possible to simulate the hadronic shower resulting from the  $b$  quark. A Feynman diagram representing a possible  $b$ -quark decay is shown in Figure ??. The simulation will calculate the final products of the decay and their kinematical variables ( $p_T, \eta, \phi, m$ ) including neutrinos and muons for

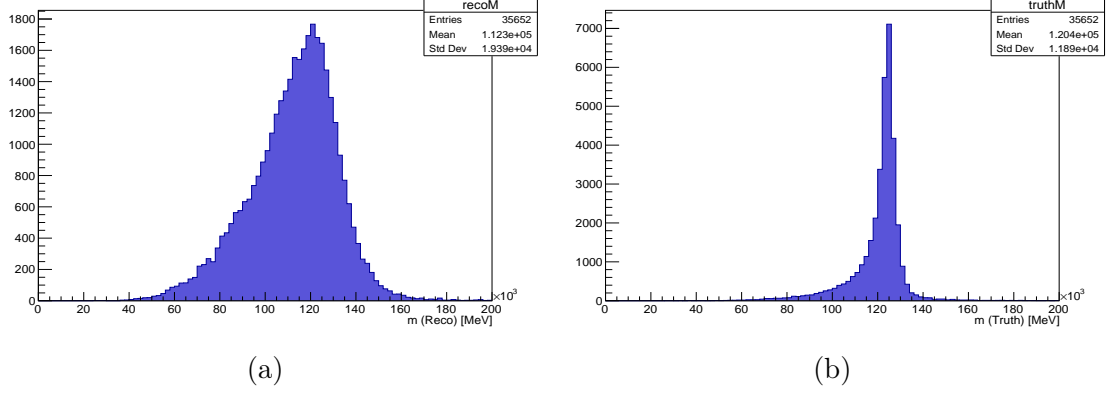


Figure 4: Invariant mass distributions calculated from  $(p_T, \eta, \phi, m)$  of the  $b\bar{b}$  jets. (a) Reconstructed distribution, peaking between 120GeV and 122GeV, with a mean of 112.3GeV and a standard deviation of 19.4GeV (b) Truth distribution, peaking between 124GeV and 126GeV, with a mean of 120.4GeV and a standard deviation of 11.9GeV. Data simulated with  $m_A = 400\text{GeV}$  and  $m_{Z'} = 800\text{GeV}$ .

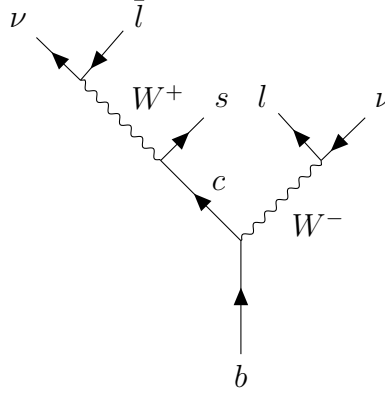


Figure 5: Feynman diagram representation of a possible  $b$ -quark decay

the truth channel but not for the reconstructed (reco) channel. For the reco channel, the simulated particles will interact with the detector simulation and the final  $(p_T, \eta, \phi, m)$  will include only particles detected by the calorimeter. The same jet clustering algorithm, the anti- $k_T$  algorithm [?], is then applied to both the truth and the reco channels. In Figure ??, the invariant mass distribution  $m_{b\bar{b}}$  is shown for the truth and the reco channel.

## 3 Regression Neural Networks

### 3.1 Neural Networks

Regression analysis is a statistical method used to determine the function  $y = f(x)$  which relates the independent variables  $x$  with the dependent variables  $y$ . This function  $f(x)$  can be approximated with the use of a neural network (NN) [?].

A neural network can be represented as a function  $g(x_i)$  which will act on a one-dimensional vector,  $x_i$ , where  $i$  is an index which labels each datapoint in the dataset, and from it compute an output vector,  $y_i = g(x_i)$  [?]. This function  $g(x_i)$  can be decomposed into several different functions referred to as layers. The first layer will act directly on the input  $x_i$  and the following layers will consecutively perform operations on the input. The final layer will compute the final output  $y_i$  [?]. The simplest layer for a neural network layer is a linear function, represented by:

$$f(x_i, W, b) = Wx_i + b, \quad (1)$$

where  $W$  is a matrix whose parameters are referred to as the weights and  $b$  is called the bias vector. The information in  $b$  can be included in the matrix  $W$  by adding an extra column at the end of it. Each row of the matrix  $W$  is referred to as a node, each of them calculating a single value of the output vector. The number of nodes in each layer is set before training and should be optimised. Several layers of a typical neural network consist of linear functions [?].

To provide non-linearity to a NN, an activation function is applied to the output of a linear layer and will only give a non-zero output for values within a certain range. A common choice for an activation function is the Rectified Linear Unit, ReLU [?], given by

$$f(x) = \max(0, x). \quad (2)$$

The purpose of the NN is to improve the function  $g(x_i)$  in order to increase the network's ability to approximate  $f(x_i)$ . The optimisation of the NN can be described as the minimisation of a loss function, which computes how successfully the network is acting. In particular, if the computed value of the loss function is high then  $g(x_i)$  is not a good approximation for  $f(x_i)$  [?]. A commonly used loss function in regression problems is the Mean Square Error (MSE), which is given by

$$L_i = (y_i^{pred} - y_i^{true})^2, \quad (3)$$

where  $y_i^{pred}$  is the output predicted by the NN and  $y_i^{true}$  is the known true value of  $f(x_i)$ . Another common loss function is the Mean Absolute Error (MAE), given by

$$L_i = |y_i^{pred} - y_i^{true}|. \quad (4)$$

The complete loss function [?] is given by

$$L = \frac{1}{N} \sum_i L_i. \quad (5)$$

In order to minimise the loss function, one can measure the gradient of the loss function at each point and then update it using gradient descent, such that

$$W' = W - s \bullet G, \quad (6)$$

where  $G$  is a matrix that contains all the gradients of  $L$  and  $s$  is a parameter referred to as the learning rate [?]. The learning rate is also set before training and must be carefully chosen to prevent divergent behaviour.

### 3.2 Training a Neural Network

To train a NN to work on a particular dataset, it is necessary to split it into a training set, a test set and a validation set. The network will train using only the training set. That is, the gradients will only be calculated and updated when the training set is being used [?]. In one epoch, the network will move through all the input vectors  $x_i$  in the training set. A network is typically trained for several epochs and, throughout training, the validation set can be used to test how the network is capable of approximating  $f(x_i)$ .

It is possible that during training, the network adapts to the training set instead of learning how to approximate  $f(x_i)$  for the entire dataset. This is visible in a plot of the loss function for the training and the validation sets. Initially, the loss of both should decrease. However, after  $n$  epochs, the loss of the validation set will begin to increase again, despite the decrease of the training loss. This phenomenon is referred to as overfitting [?]. A technique to prevent it is dropout, which consists of randomly dropping nodes in the network at a rate  $p$  during training [?]. This will prevent the network from adapting to the training set. During validation, the dropped nodes are frozen so that they are not used in the loss calculation. Once the network is trained, it can be tested using the test set. In this case, all the nodes will be used and no dropout will be applied in order to test all the parameters learned.

## 4 Using Neural Networks to correct the $b\bar{b}$ mass

As it was seen in section ??, the reco mass peak is much wider than the truth peak. In fact, while the truth distribution peaks between 124GeV and 126GeV and has a mean of 120.4GeV and a standard deviation of 11.9GeV, the reco distribution peaks between 120GeV and 122GeV has a mean of 112.3GeV and a standard deviation of 19.4GeV. Therefore, to resolve the mass peak from the background, it would be useful to develop a tool that can start from the reco peak and approximate the truth peak. For this purpose, a NN, with either 3 or 4 linear layers, was trained to output the true  $(p_T, \eta, \phi, m)$  taking as input the reconstructed  $(p_T, \eta, \phi, m)$  of the entire jet and two leptons resulting from the  $b$  decay, as shown in Figure ??.

Different targets and loss functions were considered in training the NN. In particular, the NN was trained towards the true  $(p_T, \eta, \phi, m)$ , the true  $(E, p_x, p_y, p_z)$ , defined such that

$$\begin{aligned} p_x &= p_T \sin \phi, \\ p_y &= p_T \cos \phi, \\ p_z &= p_T \sinh \eta, \\ E &= \sqrt{p^2 + m^2}, \end{aligned} \tag{7}$$

and also towards the difference between the reco and the true  $(p_T, \eta, \phi, m)$ . Both the MSE and the MAE losses were used. In order to evaluate the training of the NN, the losses were computed during training and, after training, the NN was tested and the invariant mass distribution of the output was plotted. Additionally, a quantity referred to as the resolution, defined as

$$res = \frac{p_T^{true} - p_T^{pred}}{p_T^{true}}, \tag{8}$$

was also calculated as a measure of the NN, and compared to the original resolution between the reco  $p_T$  and the true  $p_T$ .

## 4.1 Training towards Transverse Momentum

Firstly, a NN with 3 layers with 240, 120 and 60 nodes, respectively, and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . The network converged at a lower loss but it was incapable of predicting the correct invariant mass distribution. Figure ?? shows a plot of the loss function as a function of the epochs, where the training of the NN is visible. Figure ?? shows a histogram of both the output and the original resolutions. It can be seen from it that the network has not only not improved the resolution but it has actually learned a resolution which is worse than the original.

As a consequence, the invariant mass distribution calculated from the output  $p_T$  was wider than the reco peak, having a mean of 132.1GeV and a standard deviation of 20.5GeV. Furthermore, from the  $p_T$  distributions, shown in Figure ??, it can be seen that the network is over predicting values at higher  $p_T$  regions and that it is under predicting values at lower  $p_T$  regions. As a consequence, the invariant mass peak has shifted to a value much higher than the truth peak, peaking between 140GeV and 142GeV, as can be seen in Figure ??.

## 4.2 Training towards energy

The target and input  $(p_T, \eta, \phi, m)$  were converted to  $(E, p_x, p_y, p_z)$  using equation ?. The NN was then trained towards these targets and its output was converted back to  $(p_T, \eta, \phi, m)$ . It was able to learn the  $(E, p_x, p_y, p_z)$  distributions. However,

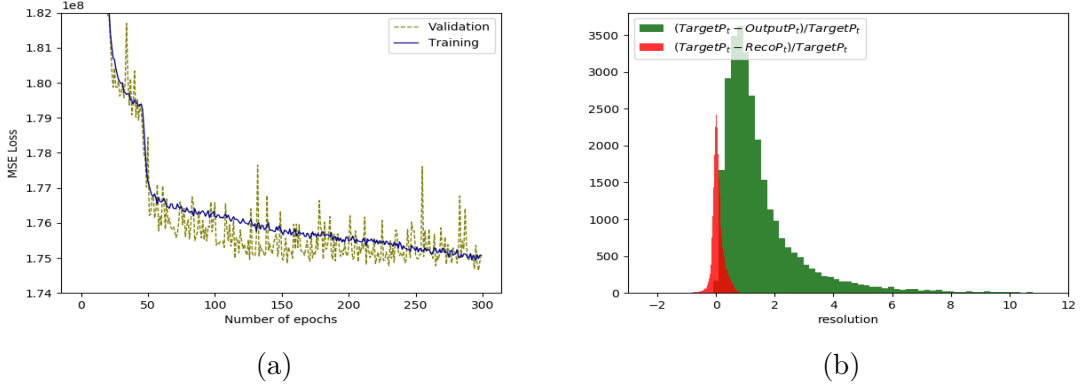


Figure 6: A NN with 3 layers with 240, 120 and 60 nodes, respectively, and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$  (a) Plot of the loss function as a function of epochs. (b) Comparison between the output resolution and the original resolution.

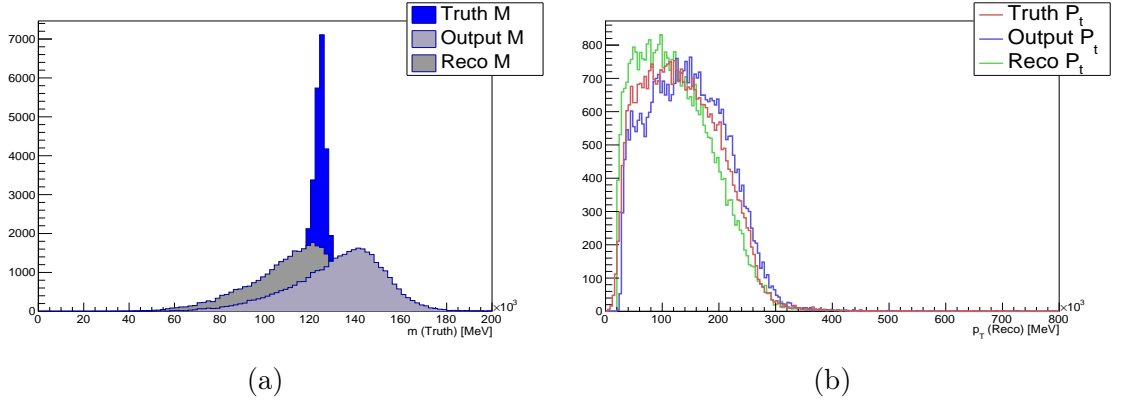


Figure 7: A NN with 3 layers with 240, 120 and 60 nodes, respectively, and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . (a) Histograms of the  $b\bar{b}$  pair invariant mass for the reco, the truth and the NN output data, which peaks between 140GeV and 142GeV and has a mean of 132.1GeV and a standard deviation of 20.5GeV. (b) Histograms of the  $p_T$  distributions for the reco, the truth and the NN output.

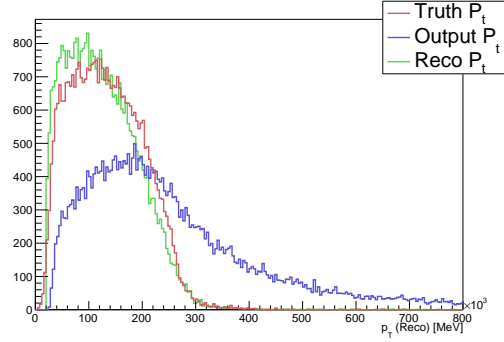


Figure 8: Plot of the truth, reco and output  $p_T$  distributions for a NN with 4 layers with 400, 300, 200 and 100 nodes, respectively, and a learning rate of 0.001 trained, using MSE, for 500 epochs towards the true  $(E, p_x, p_y, p_z)$ .

when converted back to  $(p_T, \eta, \phi, m)$ , the NN could not produce an invariant mass distribution comparable to the reco and truth ones. The distribution produced, despite peaking between 126GeV and 128GeV, has a mean of 128.8GeV and a standard deviation of 42.6GeV. In fact, the  $p_T$  distribution produced by the network, shown in Figure ??, is much more spread to higher values than the reco or truth distributions.

### 4.3 Training towards the difference

As an attempt to improve the invariant mass peak, a NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5, and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . Figure ?? shows a plot of the loss function as a function of the epochs, where the training of the NN is visible. Figure ?? shows a histogram of both the output and the original resolutions. This network was also capable of converging and it showed some improvement from the last NN in its resolution. Even though it is still not improving from the original resolution, the resolution of this output is closer to the original resolution than the previous.

The invariant mass distribution, shown in Figure ??, calculated from the output  $p_T$  peaked between 132GeV and 134GeV and it had a mean of 124.9GeV and a standard deviation of 19.2GeV. Even though the distribution was wider than the reco peak, the position of the outputted peak shows improvement in comparison to the previous NN. From the  $p_T$  distributions, shown in Figure ??, it can be seen that this NN demonstrates a behaviour of over and under predicting similar to the NN from the previous section. Consequently, the invariant mass peak is shifted to a value higher than the truth peak.

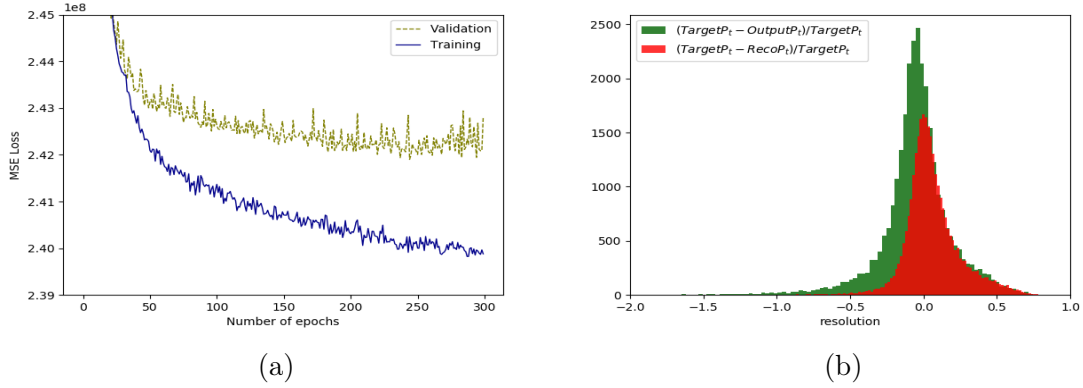


Figure 9: A NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5 and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . (a) Plot of the loss function as a function of epochs. (b) Comparison between the output resolution and the original resolution.

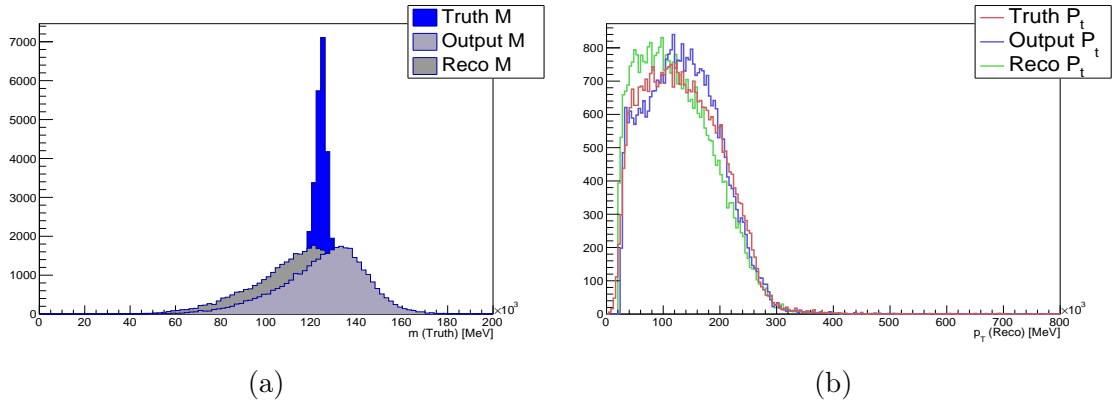


Figure 10: A NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5 and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . (a) Histograms of the  $b\bar{b}$  pair invariant mass for the reco, the truth and the NN output data, which peaks between 132GeV and 134GeV and has a mean of 124.9GeV and a standard deviation of 19.2GeV. (b) Histograms of the  $p_T$  distributions for the reco, the truth and the NN output.

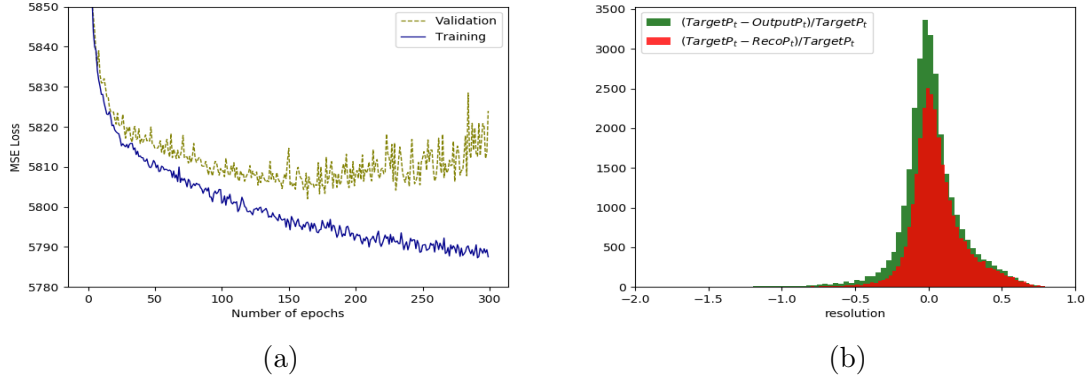


Figure 11: A NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5 and a learning rate of 0.001 was trained, using MSE, for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . (a) Plot of the loss function as a function of epochs. (b) Comparison between the output resolution and the original resolution.

#### 4.4 Training towards the difference with the MAE loss function

A different loss function, the MAE, was used to train a NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5 and a learning rate of 0.001 for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . Figure ?? shows the convergence of the loss as a function of epochs. Figure ?? shows a histogram of both the output and the original resolutions, where it can be seen that the use of the MAE loss has improved the resolution of the NN in comparison to the NNs in previous sections.

Moreover, the invariant mass peak calculated from the output  $p_T$  is between 126GeV and 128GeV, closer to the truth peak than the reco peak, as seen in figure ??. Furthermore, the NN was capable of a slight improvement to the width of the distribution, as it has a mean of 117.0GeV and a standard deviation of 18.6GeV. The correction of the peak distribution suggests that the use of MAE is a better choice for this problem. In fact, from the  $p_T$  distributions shown in figure ??, it can be seen that the NN is predicting  $p_T$  values mostly in between the reco  $p_T$  and truth  $p_T$  values, as expected.

## 5 Conclusion

In conclusion, the particular network architectures tried in this study are not enough to approximate the truth mass distribution of  $b\bar{b}$ . Table ?? shows a summary of the results from the different networks. In fact, all networks considered demonstrated a behaviour of over prediction for higher  $p_T$  values and under prediction for lower  $p_T$

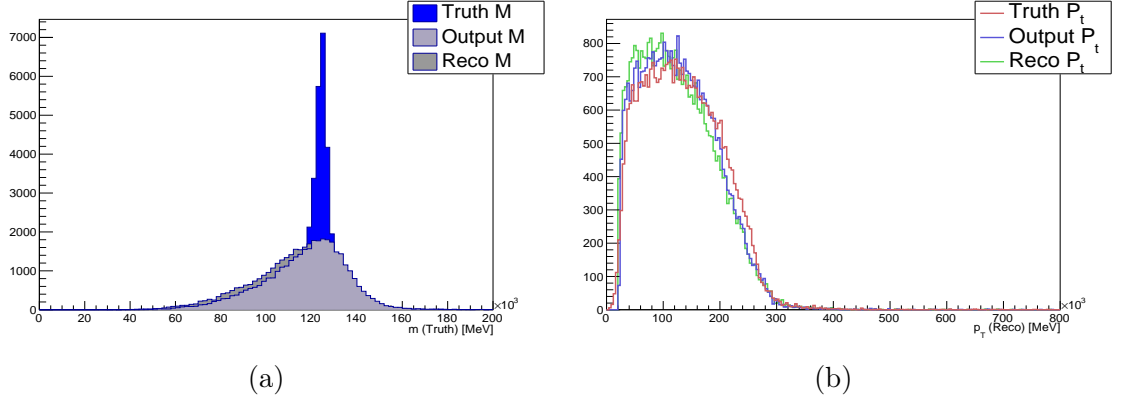


Figure 12: A NN with 3 layers with 400, 200 and 100 nodes, respectively, two dropout layers both with a dropout rate of 0.5 and a learning rate of 0.001 was trained, using MAE, for 300 epochs towards the difference between the reco and the true transverse momentum 4-vector  $(p_T, \eta, \phi, m)$ . (a) Histograms of the  $b\bar{b}$  pair invariant mass for the reco, the truth and the NN output data, which peaks between 126GeV and 128GeV and has a mean of 117.0GeV and a standard deviation of 18.6GeV. (b) Histograms of the  $p_T$  distributions for the reco, the truth and the NN output.

Target	Loss Function	Peak (GeV)	Mean (GeV)	Std (GeV)
$(p_T, \eta, \phi, m)$	MSE	140 - 142	132.1	20.5
$(E, p_x, p_y, p_z)$	MSE	126 - 128	128.8	42.6
Reco - Truth	MSE	132 - 134	124.9	19.2
Reco - Truth	MAE	126 - 128	117.0	18.6

Table 1: Summary table of the results: peak location, mean and standard deviation of the  $m_{b\bar{b}}$  distribution for each of the target and loss function combinations considered.

values which shifted the  $m_{b\bar{b}}$  peak to higher values.

However, a NN which trains towards the difference between reco and truth  $(p_T, \eta, \phi, m)$  using the MAE loss is enough to predict a mass distribution which peaks at a value closer to the truth peak and has a standard deviation lower than the reco distribution. This suggests that it might be possible to train a NN to solve the proposed problem but a different architecture would be necessary for significant improvement of the  $m_{b\bar{b}}$  distribution.

## Bibliography

- [1] G. Bertone, D. Hooper, and J. Silk. Particle dark matter:evidence, candidates and constraints. *Physics Reports*, 405:279–390, 2005.
- [2] G. Steigman and M. S. Turner. Cosmological constraints in the properties of weakly interacting massive particles. *Nuclear Physics B*, 253:375–386, 1985.
- [3] D. Abercrombie et al. Dark matter benchmark models for early lhc run-2 searches: Report of the atlas/cms dark matter forum. 2015.
- [4] ATLAS Collaboration. Search for dark matter produced in association with a higgs boson decaying to  $b\bar{b}$  at  $\sqrt{s} = 13\text{TeV}$  with the atlas detector using 79.8  $\text{fb}^{-1}$  of proton-proton collision data. *ATLAS-CONF-2018-039*, 2018.
- [5] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad. Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560:41–48, 2018.
- [6] ATLAS Collaboration. The atlas experiment at the cern large hadron collider. *JINST*, 3(S08003), 2008.
- [7] L. Carpenter, A. DiFranzo, M. Mulhearn, C. Shimmin, S. Tulin, and D. Whiteson. Mono-higgs-boson: A new collider probe of dark matter. *PHYSICAL REVIEW D*, 89(7)(075017(24)), 2014.
- [8] M. Cacciari, G. P. Salam, and G. Soyez. The anti- $k_t$  jet clustering algorithm. *JHEP*, 804(63), 2008.
- [9] D. F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568 – 576, 1991.
- [10] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10)(78-87), 2012.
- [11] W. Rawat and Z. Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, 2017.
- [12] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science*, 7700:437–478, 2012.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.