GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Bachelor's Thesis

submitted in partial fulfillment of the
requirements for the course "Applied Data Science"

Signal/ Background separation in resonant boosted HH and SH production in the $b\bar{b}VV^{(*)}$ decay channel with 1 lepton in final state

Peer Christian Drescher

Institute of Computer Science

Bachelor's Theses
of the Center for Computational Sciences
at the Georg-August-Universität Göttingen

06. February 2022

Georg-August-Universität Göttingen
Institute of Computer Science

Goldschmidtstraße 7
37077 Göttingen
Germany

☎ +49 (551) 39-172000
FAX +49 (551) 39-14403
✉ office@informatik.uni-goettingen.de
🌍 www.informatik.uni-goettingen.de

First Supervisor:	Prof. Dr. Stan Lai
Second Supervisor:	Prof. Dr. Arnulf Quadt
Thesis Number:	II.Physik-UniGö-BSc-2023/02

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 06. February 2022

Abstract

As Higgs boson pair production is often referred to as the ultimate test of the Standard Model of particle physics, many different analysis are conducted to find evidence this process. All these analysis have one thing in common: A large number of particle interactions and decays has to be recorded, reconstructed, identified and analysed. To automate such processes many different Data and Computer Science approaches are used.

In this thesis, a new Feed Forward Neural Network based model is proposed to separate and classify signal and background events for the analysis of Higgs pair production in the boosted decay channel $X \to HH \to b\bar{b}WW^*$ *with* 1 *lepton in the final state. To train the Neural Network simulated events in the range* $0.8\ \mathrm{TeV} \leq m_X \leq 5\ \mathrm{TeV}$ *at* $\sqrt{s} = 13\ \mathrm{TeV}$ *are used. The detector response for these simulated events is based on the ATLAS detector at CERN. The analysis currently uses a hand designed, cut-based approach, which is evaluated and compared to the proposed model. In addition to a training on all mass points simulated, the Neural Network is trained on two singular mass points, which are used to evaluate the model performance further. The final proposed model is shown to outperform the cut-based model.*

Keywords: *Particle Physics, Bachelor thesis, Higgs boson pair production, Machine Learning, Neural Networks*

Zusammenfassung

Higgs Boson Paarproduktion wird oft als ultimativer Test des Standardmodells der Teilchenphysik bezeichnet, weswegen viele verschiedene Analysen durchgeführt werden, um Nachweise zu finden. Alle diese Untersuchungen haben eine Sache gemeinsam: Eine große Menge an Wechselwirkungen und Zerfällen von Teilchen muss aufgezeichnet, rekonstruiert, identifiziert und analysiert werden. Heutzutage werden deshalb viele Ansätze der Informatik und Data Science angewandt, um diese Prozesse zu automatisieren.

In dieser Arbeit wird ein neues Modell, basierend auf einem Feed Forward Neural Network vorgestellt, welches Ereignisse aus dem Higgs Boson Paarproduktionskanal $X \to HH \to b\bar{b}WW^*$ *mit einem Lepton im Endzustand klassifizieren kann. Dieses Neuronale Netzwerk wird auf Simulationsereignissen in dem Bereich* $0.8\ \mathrm{TeV} \leq m_X \leq 5\ \mathrm{TeV}$ *bei einer Schwerpunktsenergie von* $\sqrt{s} = 13\ \mathrm{TeV}$ *trainiert. Simuliert werden die Ereignisse basierend auf dem ATLAS Detektor am CERN. Die Analyse des* $X \to HH \to b\bar{b}WW^*$ *Kanals benutzt derzeit ein per Hand definiertes, schnittbasiertes Modell, das in dieser Arbeit auch evaluiert und mit dem vorgeschlagenen Neuronalen Netzwerk verglichen wird. Um ein tieferes Verständnis in das Modell zu gelangen, wird zusätzlich zum Training auf allen simulierten Massenpunkten das Neuronale Netzwerk auch auf zwei einzelnen Massenpunkten trainiert. Es wird gezeigt, dass das vorgeschlagene Neuronale Netzwerk das schnittbasierte Modell übertrifft.*

Stichwörter: *Teilchenphysik, Bachelorarbeit, Higgs Boson Paarproduktion, Maschinelles Lernen, Neuronale Netzwerke*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Particle physics is such a vast field of research, that with new measurements and observations of the fundamental particles and interactions of our world, even the knowledge about the creation of our universe is expanded. To date, the best theory describing the fundamental particles and interactions is the Standard Model of particle physics (SM) [7–9].

The latest discovered particle of the SM is the Higgs boson, which was first observed in 2012 by the ATLAS and CMS experiment [10, 11]. Since then many properties of the Higgs boson have been measured and verified. Still missing is the observation of Higgs boson pair production, which is often referred to as the ultimate test of the SM, as it allows the direct measurement of the Higgs potential [12, 13]. In addition to the SM, many different Beyond Standard Models predict resonant Higgs boson pair production, which would, if experimentally proven, open up a new window of understanding about our universe.

Challenged by the sheer number of events having to be analysed, modern particle physics heavily relies on computer aided analyses. Therefore Data Science and modern approaches, such as Neural Networks, present a great opportunity. Machine and Deep Learning models are widely used for event selection, reconstruction and classification of events recorded [14–16].

In this thesis, a new deep learning model to separate and classify signal and background events for the analysis of Higgs pair production in the boosted decay channel $X \to HH \to b\bar{b}VV^*$ with 1 lepton in the final state is proposed. This channel was chosen, because it offers the second highest Higgs boson pair branching ratio. The $X \to HH \to b\bar{b}VV^*$ analysis is based on Ref. [4]. The model proposed uses Feed Forward Neural Networks and is trained on events simulated at $\sqrt{s} = 13$ TeV. Furthermore, no Beyond Standard Model (BSM) theories are considered in depth in this thesis.

Chapter 2

The Fundamentals of Particle Physics

2.1 Standard Model of Particle Physics

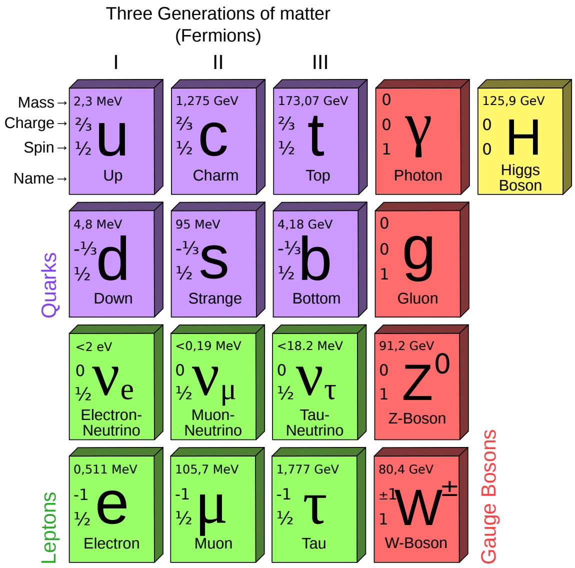


Figure 2.1: Schematic view of the Standard Model (SM) [1].

The Standard Model (SM) [7–9] is currently the most applicable theory to explain the elementary components and interactions of our universe. The SM predicts two different kind of particles: the ones that mediate interactions, bosons, and the ones that interact with said forces, fermions. To date, 12 fermions and 5 bosons have been observed experimentally. These particles can be seen in Figure 2.1.

Fermions

Fermions are spin $1/2$ particles with features such as mass, charge, colour, spin and weak isospin. Because of their differing masses, they are grouped into 3 generations, with the mass of fermions of the same type increasing between each generation. Furthermore, fermions are split into quarks and leptons.

Quarks are either up-type, meaning they possess charge $+2/3\ \mathrm{e}$ or down-type meaning they possess charge $-1/3\ \mathrm{e}$ with e being the elementary charge. Additionally, quarks carry colour charge. As a result, they cannot be observed alone, but only bound in hadrons. This phenomena is explained later by the strong interaction. The first generation of quarks are the up quark u and down quark d, followed by the charm quark c and strange quark s. Finally, the top quark t and bottom quark b form the third generation. Since quarks are massive, charged and carry colour they can interact with every boson the SM predicts.

Leptons on the other hand, do not carry colour and are integer charged. They are grouped into charged leptons with charge $\pm 1\ \mathrm{e}$ and neutral charged leptons called neutrinos. As a result, charged leptons can interact with the electromagnetic and weak interaction and neutrinos only with the weak interaction. The first generation of leptons are the electron e and the electron neutrino ν_e. Second are the muon μ and muon neutrino ν_μ. The third generation of leptons are the tauon τ and tauon neutrino ν_τ. As seen, every charged lepton has a corresponding neutrino.

Forces

In the SM, fermions can interact with one another as described by the respective quantum field theory (QFT), which correspond to local gauge symmetries. An excitation of the corresponding gauge field is a gauge boson. Interactions can be ranked by their respective strength, quantified by the coupling constant α_i. The value of these coupling constants are in fact not constant, but change with the energy scale. Consequently the coupling constants are referred to as running constants.

The electromagnetic interaction is described by quantum electrodynamics (QED) [17] and its mediating particle the photon γ. Photons are spin 1 vector bosons and carry neither electrical charge, colour nor mass. They interact with charged particles and therefore cannot couple to themselves. QED corresponds to the symmetry of the $\mathrm{U}(1)$ gauge group.

The weak nuclear force [18–20] on the other hand is symmetric under an $\mathrm{SU}(2)$ group transformation and is mediated by two particles: the charged $W^{\pm}$-boson and the neutral charged Z-boson. Both of these particles are massive vector bosons with spin 1. Coupling to the $W^{\pm}$ and Z bosons is possible via the weak isospin I_3. Only fermions with negative chirality and antifermions with positive chirality can interact with the charged weak nuclear force. $W^{\pm}$-bosons mediate between two particles with $I_3 = +1/2$ and $I_3 = -1/2$, thereby changing the quark and lepton flavour. Because of that, the W also has to be of integer weak isospin $I_3 = \pm 1$ or $I_3 = 0$. The $W^{\pm}$ and Z acquire their mass via the Higgs mechanism described later. But because of their masses, the coupling strength of the weak nuclear force is suppressed at low energies.

The electromagnetic and weak nuclear force can also be unified into the electroweak interaction [18,19]. This force is described by a symmetry of an $\mathrm{SU}(2)_L \times \mathrm{U}(1)_Y$ with L standing for coupling to particles with left-handed chirality and the hypercharge $Y = 2(Q - I_3)$, where Q is the electric charge.

The strong interaction describes the coupling of particles with colour. Gluons g mediate the strong interaction and are themselves colour charged. Consequently, only gluons and quarks can interact strongly. Gluons are massless spin 1 vector bosons, with no electric charge. There are three different colour charges: red, green and blue. A particle carrying a single colour is not observable, however colourless hadrons can be observed. These hadrons are bound quarks constantly interacting via the strong force with each other. For a hadron to be colourless, the quark colours have to match by either being colour - anti-colour pairs or three particles of all different (anti-)colours. There are a total of 8 gauge fields and gluons. These fields stem from a symmetry of an $\mathrm{SU}(3)$ group and are described by quantum chromodynamics (QCD) [21].

Higgs Mechanism

The last boson of the SM is the Higgs boson H. The Higgs boson is the second heaviest particle of the SM and is a scalar boson with spin 0. It is neither electrical nor colour charged. Nevertheless, the Higgs field and Higgs boson are the reason that fermions and gauge bosons possess mass [22]. The Higgs mechanism is expressed as a complex doublet of scalar fields in the Lagrange density of the electroweak interaction. The corresponding potential known as the Higgs potential can be expressed as follows [22]

$$V(\phi) = \mu^2(\phi^\dagger \phi) + \lambda(\phi^\dagger \phi)^2 \tag{2.1}$$

with $\lambda > 0$ and $\mu^2 < 0$ and is shown in 2.2.

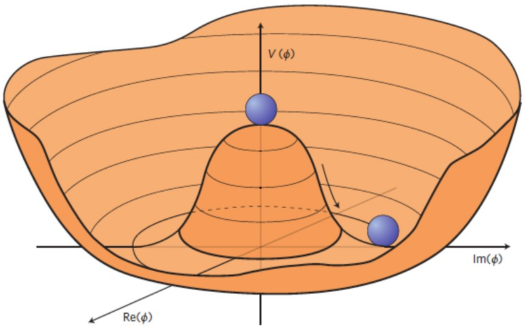


Figure 2.2: An illustration of the Higgs potential [2].

As visible in Figure 2.2, the global minimum of the Higgs potential is not at the origin, but located at $v = \sqrt{-\frac{\mu^2}{\lambda}}$. As a result the symmetry of the Higgs potential is broken, because the ground state of the Higgs potential does not respect the symmetry of the system. This allows the fermions, $W^{\pm}$ and Z bosons to couple to the Higgs field whose vacuum expectation is non zero and thereby obtain mass. Their masses can be expressed as functions of v

$$m_W = \frac{1}{2} g_W v \tag{2.2}$$

$$m_Z = \frac{1}{2} \frac{g_W}{\cos\theta_W} v \tag{2.3}$$

$$m_f = \frac{1}{\sqrt{2}} g_f v \tag{2.4}$$

with g_W being the weak coupling constant, θ_W the Weinberg angle (also known as weak mixing angle) and g_f the Yukawa coupling constant [23].

In general, the SM is in accordance with current measurements and the most successful description of particle physics today. However, the SM is not a complete theory as it does not explain every effect of our universe, most notably gravity and dark matter.

2.2 Higgs Boson Pair Production

The Higgs boson is the newest discovered particle of the SM. It was discovered in 2012 by the ATLAS and CMS expirements in LHC at CERN [10,11].

The Higgs boson is a scalar boson of spin 0 and has a mass of $m_H = 125.09 \pm 0.21(stat.) \pm 0.11(sys.)$ GeV [24,25] and a total cross section of $\sigma_H^{total} = 55.62$ pb measured at $\sqrt{s} = 13$ TeV [24,25]. Because of their large mass, Higgs bosons decay quickly. The branching ratios of the Higgs boson are shown in Figure 2.3 as a function of Higgs boson mass. As can be seen, the Higgs decay is dominated by $H \to b\bar{b}$, followed by $H \to WW$, because of their large mass.

The production of Higgs bosons at the LHC is dominated by gluon-gluon fusion (ggF), meaning two gluons give rise to a virtual top quark loop which fuses to a Higgs boson. Secondly Higgs bosons can be produced by vector boson fusion (VBF). The third most likely production channel is the W or Z associated production. Least likely is a production in association with a $t\bar{t}$ pair. All leading order Feynman diagrams of these production channels are depicted in Figure 2.4.

The last missing component of the Higgs potential (Eq. 2.1), the self-coupling constant λ, can be measured by Higgs boson pair production. In proton-proton collisions, the production of a pair of Higgs bosons is dominated by gluon-gluon fusion and is sensitve to the Higgs boson self coupling. As seen in Figure 2.5 a) the Higgs pair stems from a decaying virtual Higgs boson. Thus the Higgs bosons couple to each other and the self coupling constant λ is measurable. However, Higgs boson pair production is heavily suppressed due to destructive interference, for example by

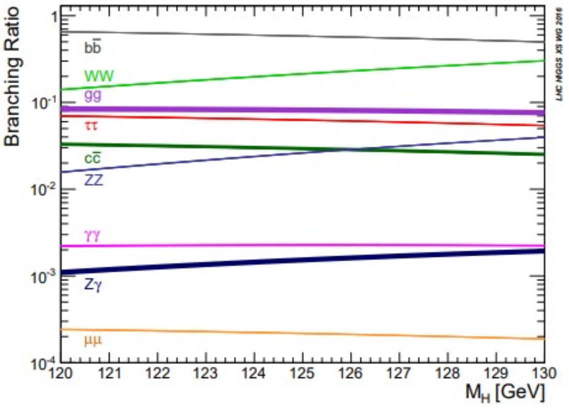
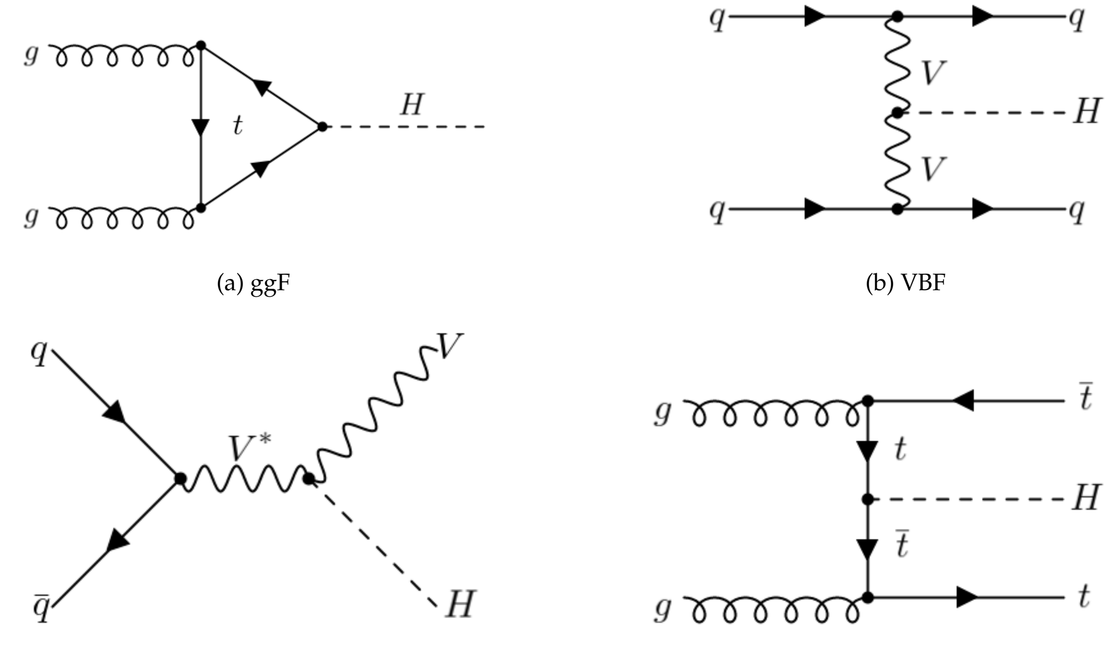


Figure 2.3: Higgs decay channel with branching ratios depicted as a function of the mass of the Higgs boson [3].

top Yukawa-coupling 2.5 b). Therefore, Higgs boson pair production has only a cross-section of about $\sigma_{HH}^{\mathrm{SM}} = 31.05\ \mathrm{fb}$ [26].

The aforementioned self interacting pair production is also referred to as non resonant. Many Beyond Standard Model theories (BSM) also predict one or more other particles which would be capable of decaying resonantly into two SM Higgs bosons. One example would be the extension of the SM with a second complex scalar doublet, as this is necessary for supersymmetry [27]. Another example is the Two-Real-Singlet Model (TRSM) [28], which is including two real scalar singlets to the SM. The TRSM even predicts two additional scalar boson, which can decay into two SM Higgs bosons, depending on the chosen mass scales. An example for such a process is shown in Figure 2.5 c).



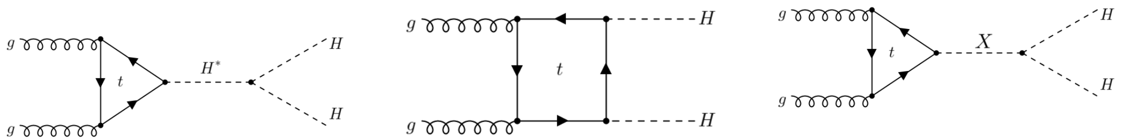
(a) ggF

(b) VBF

(c) VH

(d) $t\bar{t}H$

Figure 2.4: Lowest order Feynman diagrams for the production of Higgs bosons at LHC. a) gluon-gluon fusion ggF, b) vector boson fusion VBF, c) W, Z H and d) $t\bar{t}H$ production [4].



(a) virtual Higgs decay

(b) top Yukawa-coupling

(c) resonant Higgs boson pair production via heavy scalar X

Figure 2.5: Lowest order Feynman diagrams for the pair production of Higgs bosons at LHC [4].

Chapter 3

The Fundamentals of Neural Networks

Instead of a human defining an algorithm which calculates a solution to a problem, in Machine Learning, an algorithm is designed to learn a statistical model on which a decision can be made. This requires training data and a function which quantifies how good the model performs, known as a loss function (also known as cost function). By optimising the loss function, the model learns how to weight data to get the best output. Often machine learning algorithms use some form of regression to build a statistical model.

3.1 Neural Networks

Neural Networks are a class of Machine Learning algorithms first envisioned in the late 1950s [29] and designed after the way the human brain makes decisions. They are often referred to as Deep Learning.

Often Neural Networks are built up from perceptrons and activation functions. In combination a perceptron and activiation function perform a binary regression, and are called a unit. By joining many units in series and in parallel, a Neural Network is formed.

Multilayer Perceptron

A perceptron [29,30] works by summing n weighted input variables $\sum_{i=1}^{n} w_i \cdot x_i$ for $i = 1, ..., n$ and adding a bias b. Afterwards the activation function $f : \mathbb{R} \to \mathbb{R}$ is applied, leading to the unit output:

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right) \tag{3.1}$$

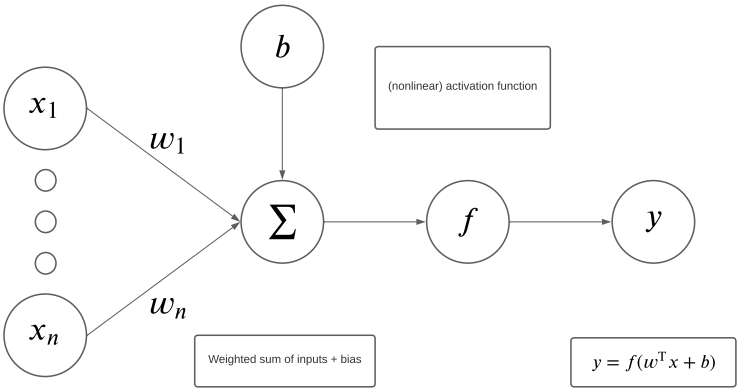

b
(nonlinear) activation function
x_1
w_1
Σ
f
y
x_n
w_n
Weighted sum of inputs + bias
$y = f(w^{\mathrm{T}}x + b)$

Figure 3.1: Diagram of a perceptron.

During training, the weights w_i and bias b_i are optimised. As seen easily this can be vectorised to:

$$y = f\left(\vec{w}^T\vec{x} + b\right) \tag{3.2}$$

By running many identical units in parallel, a layer is formed. A layer takes n variables as input, runs them through m units, and returns m output variables. A layer is expressed by the weights matrix $W \in \mathbb{R}^{m\times n}$, the bias vector $\vec{b} \in \mathbb{R}^m$ and an activation function $f : \mathbb{R}^m \to \mathbb{R}^m$:

$$\vec{y} = f\left(W\vec{x} + \vec{b}\right) \tag{3.3}$$

Finally multiple layers are combined sequentially to form a Multilayer Perceptron (MLP). The number of layers is referred to as depth of the Neural Network, and the count of units in a layer as width. The first layer of a MLP Neural Network consists of the input variables and is referred to as input layer. The last layer of the Network is called the output layer and comprises the output variables, on which a decision is made. In between the input and output layer, layers are referred to as hidden layers, because from the outside, no insight is given into these layers. MLPs are referred to as a Feed Forward Neural Networks, because data flows from the input to the output without forming loops or recursion.

The activation function f is important for the performance of an MLP, because it defines a cut on the output variables of a layer [31]. Therefore, non-linear functions are most desirable, such as a binary step function at some threshold. On the other hand, the function should be differentiable (at least in almost all points), because of the gradient based optimisation of the Neural Network (see section 3.2). The preferred activation function nowadays is the Rectified Linear Unit (ReLU) [32]:

$$ReLU(x) = \max(0, x) \tag{3.4}$$

The ReLU cuts of the negative output from propagating further through the network, but does not normalise the positive values to a specific range. Non-differentiability at zero is problematic for the ReLU, since it can lead to instability during training. Because of this, similar, but fully differentiable functions like the Sigmoid Linear Unit (SiLU) have been developed (see Figure 3.2).

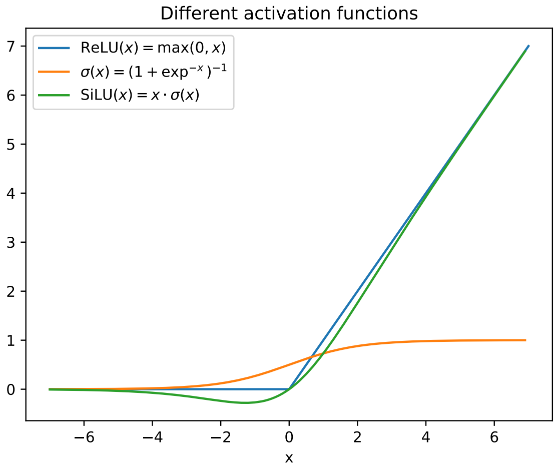


Figure 3.2: Rectified Linear Unit (ReLU), Sigmoid (σ) and Sigmoid Linear Unit (SiLU) activation functions.

Universal Approximation theory mathematically proves, for different problems and designs of Neural Networks, conditions in which a desired function can be approximated to an arbitrary precision. It was first shown, that for an MLP consisting of one layer with sufficient width and a non-polynomial activation function, any continuous function could be approximated to an arbitrary precision [33,34]. However, using only one layer of sufficient width maybe mathematically possible to represent a problem, but deeper networks are easier to train numerically. Therefore, for arbitrary deep MLPs using ReLUs as activation functions, it was shown, that the minimum width of a layer is $\max\{\text{input dim. of the target function} + 1, \text{output dim. of the target function}\}$ to approximate certain target functions [35–37]. In practice MLPs are chosen to be wide enough and the number of hidden layers is used to increase the model's overall generalisation power.

3.2 Optimisation

Optimising a Neural Network is heavily dependant on the task and the dataset provided [30]. In general, a dataset is the most useable if all included features and classes are represented equally.

Additionally the dataset should sample the input space uniformly. A dataset that does not fulfil this would include a bias, which will be learned and used by any model trained on this data. To train and evaluate a model on a dataset, the dataset is split into three groups: training, validation and testing. The training dataset is the largest one and as the name suggests contains the data used to train the model. During the training of the model, the validation dataset is used to cross check the performance of the neural network, and to stop the training when the model's performance on the validation set stagnates, indicating overfitting of the model. Lastly, the test dataset is used to evaluate the performance of the network. For that reason the test data can not be visible to the model during training, to avoid overfitting. The test dataset should be the second largest dataset, to provide enough statistics. In general three different kinds of algorithms are used to optimise Neural Networks: supervised, unsupervised and reinforcement learning.

Supervised learning [30] uses labelled data to compare the model output with the labels. This approach is often used for classification tasks.

Unsupervised learning [30] uses unlabelled data. The model is optimised to predict the data again. This first sounds undesirable, but by first encoding the data to a smaller latent space and then decoding it back to the original space, the model is trained to find patterns in the features. Disadvantageously, these patterns are usually not understandable for humans. Because of this design, such an approach is often used in recognition or text based tasks.

Reinforcement learning [38] is a completely different approach. It is based on the idea of punishing and rewarding an agent for taking an action in an environment. Thereby the agent is trained to find a compromise between exploring uncharted environments and exploiting already existing knowledge. Easy examples for reinforcement learning are computer controlled bots in a video game or self driving cars. Reinforcement learning is not applicable to the use case in this thesis and will therefore not be explained in more detail.

In this thesis, supervised learning is applied to train a Neural Network. The different steps to train a Neural Network are explained in detail, in the following Sections 3.2.1 - 3.2.4.

3.2.1 Loss Function

Since Neural Networks are trained by minimising a loss function [30], the loss has tremendous impact on the model performance. Furthermore, the loss function should be easily optimisable. This means the function should be differentiable in almost every point, have an easy to find global minima and in the best case be convex. In addition, a function with a clear global minima could still be inadequate, if many local minima exist, in which a optimiser could get stuck in. Because of these reasons the most used loss functions are the Mean-Squared-Error (MSE):

$$MSE(x,t) = \frac{1}{n}\sum_{i=1}^{n}(x_i - t_i)^2 \tag{3.5}$$

and the Cross-Entropy-Loss:

$$CrossEntropy(x, t) = -\sum_{i=1}^{n} t_i \cdot \log x_i \tag{3.6}$$

with x being the model prediction and t the truth label for n classes.

The Mean-Squared-Error is one of the most straightforward loss functions. It calculates the mean difference between the true label and prediction. By squaring the difference, large differences between the target and prediction are amplified more than small deviations, which increases training stability. Because of this behaviour, the MSE is often used for regression tasks and tasks that require a continuous model output [30].

The Cross-Entropy-Loss is used in classification tasks, for optimising the probability of an input belonging to a class. Often the Cross-Entropy-Loss is combined with a Softmax $\mathrm{Softmax}(x_i) = \frac{\exp^{x_i}}{\sum_j \exp^{x_j}}$ or Sigmoid $\sigma(x_i) = \frac{\exp^{x_i}}{1+\exp^{x_i}}$ activation function in the output layer, to force the output to be in $x \in [0, 1]^n$. The logarithmic nature of the Cross-Entropy-Loss punishes large deviations from the target. In addition, minimising the Cross-Entropy-Loss for a parameter is the same as maximising the likelihood for that parameter, which is a well known statistical optimisation [30].

3.2.2 Gradient Descent and Backpropagation

The loss of a Neural Network is optimised in its simplest form via first order Gradient Descent optimisation. For the loss function $f(\theta)$, the model parameter θ and learning rate Gradient Descent can be implemented as in Algorithm 1.

Algorithm 1 Gradient Descent

Input: θ_0 (model parameters), $L(\theta)$ (loss function), γ (learning rate)

```
for $t = 1, \cdots, T$ do
    $g_t = \nabla_\theta L_t(\theta_{t-1})$          ▷ compute gradients
    $\theta_t = \theta_{t-1} - \gamma g_t$          ▷ Update parameters
end for
```

Output: θ_T

First the training data is passed through the model and an output is predicted. This step is referred to as forward pass. After the loss has been calculated, the gradient of the loss is calculated for all model parameters. The parameters are then updated by subtracting the gradient of the loss from the parameters. The learning rate γ determines the amount of change applied to the model parameter per gradient descent step, thereby defining the convergence rate.

The problem with this approach is the need for the gradient of the loss function for all model parameters. Consider a Neural Network with 3 layers called $y = f(x)$, $z = g(y)$ and $h(z)$ with the output of the network given as $L(x) = h(g(f(x)))$. To calculate the gradient, the chain rule is needed:

$$\frac{\partial L}{\partial x}(x) = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial x} = h'(z)g'(y)f'(x) = h'(g(f(x))g'(f(x))f'(x) \tag{3.7}$$

To calculate this gradient, Backpropagation [39] is employed. This means that during the forward pass of the data, every layer/parameter saves its input. Then knowing its own gradient, any layer can calculate the gradient for each of its parameters dependent on the input. To obtain the full gradient, all layer gradients are then multiplied together.
As an example: For the layer h, the input z is saved, $h(z)$ calculated and passed forward. Afterwards for the layer h, the gradient $h'(z) = h'(g(f(x)))$ can be calculated, when the gradient h' is known. Independently for g and f, the respected gradients can be computed, to form the desired result: $\frac{\partial L}{\partial x}(x) = h'(z)g'(y)f'(x)$.
If each layer and function has a known derivative by design, Backpropagation is therefore relatively efficient to calculate.

To avoid the memory consumption of Backpropagation, Stochastic Gradient Descent is used [40]. In this alteration of the Gradient Descent algorithm, not all data is used to calculate the most desirable gradient per training's epoch, but only a sample of the training data, called a batch. By using the smaller batches, more gradients have to be calculated during an epoch, which results in suboptimal gradients, but reduces memory and computation consumption. In practice, the Stochastic Gradient Descent still convergences but requires more iterations than classical Gradient Descent. To further increase numerical stability and the convergence rate, weight decay/regularisation and momentum are employed in Stochastic Gradient Descent algorithms as in Algorithm 2.

Momentum [41,42] is based on the idea of retaining a velocity vector b_t pointing in the direction of persistent reduction over the previous iterations. The velocity vector classically is defined with momentum μ, the impact of the new gradient to the velocity vector and dampening τ, the amount of retained velocity [41,42]:

$$\mathrm{b}_t = \mu \mathrm{b}_{t-1} + (1-\tau)\nabla_\theta L_t(\theta_{t-1}) \tag{3.8}$$

Weight decay and regularisation are both based on the idea of reducing the impact of gradients with large magnitude to favour a "simpler" solution. Regularisation works by adding a term to the optimisation problem, resulting in a unique solution being forced. In this case the weighted L_2 norm of the input x is added. Weight decay retains an amount of the previous gradients during optimisation, to smooth out the current gradient, because in general all gradients should point in the overall same direction. In Stochastic Gradient Descent weight decay and L_2 regularisation are equivalent and defined as a weighting factor λ [43]:

$$\theta_t = (1-\lambda)\theta_{t-1} - \gamma\nabla L_t(\theta_t) \tag{3.9}$$

Algorithm 2 Stochastic Gradient Descent with momentum and regularisation

```
Input: $\theta_0$ (model parameters), $L(\theta)$ (loss function), $\gamma$ (learning rate), $\lambda$ (weight decay),
    $\mu$ (momentum), $\tau$ (dampening)
for $t = 1, \cdots, T$ do
    $g_t = \nabla_\theta L_t(\theta_{t-1})$                ▷ compute gradients
    $g_t = g_t + \lambda\theta_{t-1}$                      ▷ regularize gradients
    if $t > 1$ then                                         ▷ Calculate momentum
        $\mathrm{b}_t = \mu\mathrm{b}_{t-1} + (1-\tau)g_t$
    else
        $\mathrm{b}_t = g_t$
    end if
    $g_t = \mathrm{b}_t$
    $\theta_t = \theta_{t-1} - \gamma g_t$                 ▷ Update parameters
end for
Output: $\theta_T$
```

3.2.3 Adam, AMSGrad and decoupled weight decay

To further improve numerical stability and convergence rate, algorithms have been developed which control momentum and dampening on their own. One of the most prominent is Adam, named after Adaptive Moment estimation. As the name suggests, Adam estimates the first and second order moment to update the learning rate instead of using momentum and dampening [44]. This design was chosen to combine the advantages of AdaGrad and RMSProp in a memory efficient algorithm. The first excels at dealing with sparse gradients and the second one at dealing with non-stationary objects. Because of this, Adam has become widely used for training machine learning problems with large datasets and/or high-dimensional parameter spaces.

As advantageous as Adam is, two problems in the original definition lead to suboptimal performance in some cases. However these problems can easily be fixed.

The first problem stems from weight decay not being equal to L_2 regularisation in Adam as in Stochastic Gradient Descent. Originally Adam used L_2 regularisation, resulting in scaling the sums of the gradient of the loss function and the gradient of the L_2 regulariser. On the other hand, decoupled weight decay only adapts the gradient of the loss function. This results in weight decay regularising with a larger gradient magnitude than L_2 regularisation and is shown by Ref. [45] to be more favourable.

Secondly, the standard Adam algorithm does not converge in some cases. This problem is fixed by implementing AMSGrad [46] into Adam. In Adam, the second order moment estimate v_t can increase the learning rate if $g_t = \nabla_\theta L_t(\theta_{t-1}) < v_{t-1}$, because then $v_t = \beta_2 m_{t-1} + (1-\beta_2)g_t^2 < v_{t-1}$. This will result in a slowing of convergence and in the worst case non convergence. But by taking the maximum of all previous second order moment estimates $v^{\max} = \max(v^{\max}, v_t)$ the learning rate cannot decrease, resulting in overall faster convergence.

The following algorithm 3 is the implementation of Adam with AMSGrad and decoupled weight decay used as optimiser in this thesis:

Algorithm 3 Adam with AMSGrad and decoupled weight decay

```
Input: $\theta_0$ (model parameters), $L(\theta)$ (loss function), $\gamma$ (learning rate), $\beta_1, \beta_2$ (betas), $\lambda$ (weight decay), $\epsilon = 10^{-8}$ (to avoid dividing by 0)
$m_0 = 0$ (first moment)
$v_0 = 0$ (second moment)
$\hat{v}^{\max} = 0$
for $t = 1, \cdots, T$ do
    $g_t = \nabla_\theta L_t(\theta_{t-1})$                                   ▷ compute gradients
    $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$                          ▷ compute biasesd first order moment estimate
    $v_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2$                        ▷ compute biasesd second order moment estimate
    $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$                                   ▷ bias correct first order moment estimate
    $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$                                   ▷ bias correct second order moment estimate
    $\hat{v}^{\max} = \max(\hat{v}^{\max}, \hat{v}_t)$                      ▷ Update maximum $v^{\max}$
    $\theta_t = \theta_{t-1} - \gamma \frac{\hat{m}_t}{\sqrt{\hat{v}^{\max}} + \epsilon} - \gamma \lambda \theta_{t-1}$   ▷ Update parameters with decouled weight decay
end for
Output: $\theta_T$
```

3.2.4 Gradient Problems

During training, the gradients of the different parameters are of particular interest, because problems are directly visible via the Gradient Descent Optimisation. There are two different kinds of gradient problems: exploding and vanishing gradients.

Exploding Gradients

Exploding gradients are gradients that consistently grow larger during training, resulting in big changes to the parameters, which leads to unstable model performance. Exploding gradients are often a sign for implementation errors or suboptimal initialisation of model parameters in Feed Forward Neural Networks. In Recurrent Neural Networks, exploding gradients are much more common and often are a by product of the model architecture. To avoid exploding gradients gradient clipping and weight decay or L_2 regularisation can be used.

Vanishing Gradients

Vanishing gradients are much more common for Feed Forward Neural Networks. They often result from using the wrong activation function in a layer. As an example, the sigmoid function

$$\sigma(x) = \frac{1}{1+\exp^{-x}} = \frac{\exp^{x}}{1+\exp^{x}} \tag{3.10}$$

compresses its input to $[0, 1]$. This can be useful, but with

$$\frac{\partial}{\partial x}\sigma(x) = \frac{\exp^{-x}}{(1+\exp^{-x})^2} = \sigma(x)(1-\sigma(x)) \tag{3.11}$$

the gradients for $|\sigma(x)| \to 1$ become

$$\lim_{|\sigma(x)|\to 1} \frac{\partial \sigma}{\partial x}(x) = 0 \tag{3.12}$$

Because the input does not have to be relatively large or small to achieve this behaviour, these areas can easily be saturated during training, resulting in vanishing gradients (see Figure 3.3).

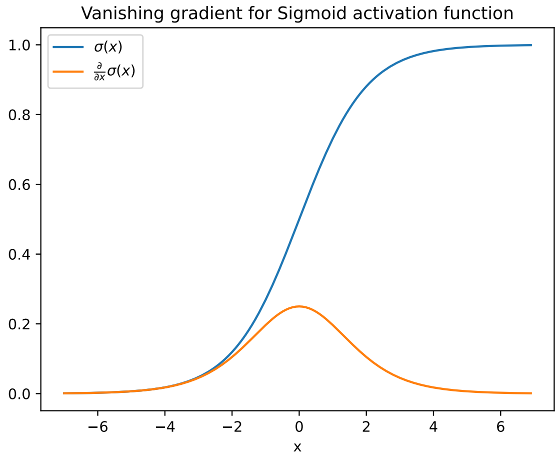


Figure 3.3: Sigmoid activation function and gradient.

To avoid vanishing gradients, ReLUs or similar functions are used as activation functions to connect layers. In addition, Batch normalisation between layers, gradient clipping and weight decay or regularisation can help [47].

Batch Normalisation

Batch normalisation [47, 48] was designed to keep data propagating through the network standard normalised. For $x_B \in \mathbb{R}^n$, an input batch with mean μ_B and variance σ_B^2, Batch normalisation is defined as in Ref. [48]:

$$\text{Batchnorm}(x_B) = \gamma \cdot \hat{x_B} + \beta = \gamma \cdot \frac{x_B - \mu_B}{\sqrt{(\sigma_B)^2 + \epsilon}} + \beta \tag{3.13}$$

with γ and β being learnable parameters to restore the representation power of the network. By setting $\gamma = \sqrt{(\sigma_B)^2}$ and $\beta = \mu_B$ the original input variable can even be recovered and therefore, during training, a desired representation is learned. Going back to the vanishing gradient problem, it is easily visible that batch normalised data is much more suited for activation functions compressing the output. Overall this architecture leads to an increase in training stability, speed and overall model performance.

3.2.5 Hyperparameters

During training, the following hyperparameters define the performance and training stability of a Feed Forward Neural Network using Multilayer Perceptrons (values provided are recommended default values):

- **Number of layers**: defines the depth of the Neural Network and therefore representation power.
- **Number of units used per layer**: controls the width of the NN and determines which functions can be fitted (Universal Approximation theory).
- **Number of batch normalisation layer**: helps with vanishing gradients and increases convergence rate and overall model performance.
- **Activation functions used per layer**: defines cuts on the layer output. For connections between layers, ReLUs are recommended. For the output layer, the activation function depends on the task and wanted output. For multi-class classification, the Softmax function is advised.
- **Loss function**: determines the way the model is trained. If the NN outputs probabilities, the Cross Entropy loss is recommended.
- **Optimiser**: defines how accurate and how fast the model is trained.
    - learning rate γ: determines the maximum change to the NN parameter per optimisation step.
    - weight decay $\lambda = 0.01$: defines how much the previous gradients influence the current gradient, to smooth out the overall gradient descent.
    - betas $\beta_1 = 0.9$, $\beta_2 = 0.999$ (if ADAM is used): determine the influence of previous moments estimates on the current estimation.

Chapter 4

The ATLAS Experiment and the Large Hadron Collider

4.1 The Large Hadron Collider

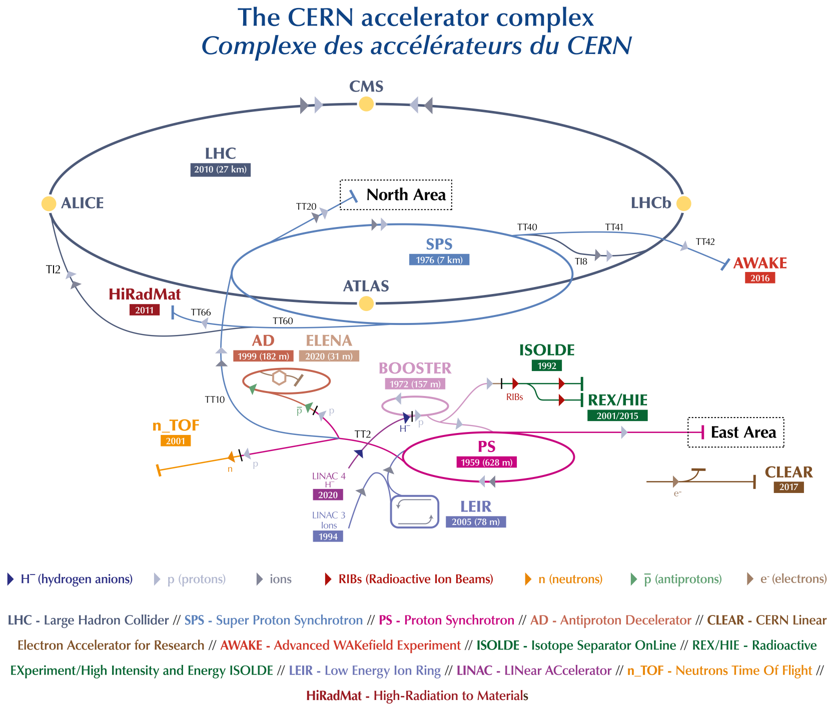

The CERN accelerator complex
Complexe des accélérateurs du CERN
CMS
LHC
2010 (27 km)
ALICE
North Area
LHCb
TT20
TT40
TT41
TT42
SPS
1976 (7 km)
TI8
AWAKE
2016
TI2
HiRadMat
2011
ATLAS
TT66
TT60
AD
1999 (182 m)
ELENA
2020 (31 m)
BOOSTER
1972 (157 m)
ISOLDE
1992
RIBs
REX/HIE
2001/2015
TT10
p̄
p
n_TOF
2001
TT2
H⁻
p
East Area
PS
1959 (628 m)
n
p
LINAC 4
H⁻
2020
CLEAR
2017
e⁻
LINAC 3
Ions
1994
LEIR
2005 (78 m)
H⁻ (hydrogen anions)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive EXperiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials

Figure 4.1: Schematic view of the CERN accelerator complex [5].

The Large Hadron Collider (LHC) [49] was built by the European Organisation for Nuclear Research (CERN) in Geneva to study particles of the SM and to search for BSM particles and interactions. CERN is not only encompassing the LHC but is a complex of many accelerators and experiments as seen in Figure 4.1. Admittedly, LHC is the biggest of them with a circumference of 27 km, built in tunnels about 100 m underground. The LHC was designed to collide protons at a centre-of-mass energy of $\sqrt{s} = 14$ TeV at a luminosity of $\mathcal{L} = 10^{34}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [49].

LHC was first operational between 2010 and 2013 with a centre-of-mass energy between $\sqrt{s} = 7$ and 8 TeV. This first data collection period is called Run I. For Run II, the LHC was upgraded and a energy of $\sqrt{s} = 13$ TeV was reached. This run took place from 2015 until 2018, afterwards LHC was upgraded again. In 2022, the LHC Run III started again with $\sqrt{s} = 13.6$ TeV. Run III is supposed to end 2025.

To reach the required energy for LHC collisions, protons have to pass through different accelerators which increase the energy of the protons step-by-step. The first accelerator is the LINAC 2, a linear accelerator which accelerates ions containing protons to 50 MeV. In Run III, LINAC 4 is used instead which accelerates protons up to 160 MeV. After the LINAC, the protons are inserted into the Proton Synchrotron Booster. There, the protons are extracted and accelerated to 1.4 GeV. Afterwards the beam is fed into the Proton Synchrotron (PS) and the Super Proton Synchrotron (SPS). Protons are now accelerated to 450 GeV. Finally they are inserted into the LHC and accelerated to their currently final energy of about 6.5 TeV.

The collision data of the LHC is collected at 4 main experiments: ATLAS [50], CMS [51], ALICE [52] and LHCb [53]. ATLAS and CMS are designed as general purpose detectors, while ALICE and LHCb have specialised physics programs. The ALICE experiment focuses on quark deconfinement and quark-gluon plasma to study QCD. On the other hand, LHCb is designed to study b physics and thereby explore CP violation. In this thesis only data from the ATLAS project is considered.

4.2 The ATLAS Experiment

The ATLAS experiment, as mentioned above, is a general purpose detector. Therefore it is composed of many sub-detectors to measure every possible particle. These sub-detectors are ordered by increasing distance to the interaction vertex: the inner detector (ID), the calorimeters (CAL) and the muon spectrometer (MS). The detectors are arranged cylindrically around the proton beam as seen in Figure 4.2.

The ATLAS experiment uses a right-handed coordinate system with x pointing from the interaction vertex to the center of LHC, y upwards and z parallel to the beam. Because of the cylindrical design of ATLAS, the polar angle θ and azimuthal angle ϕ are often used instead of x and y. Additionally, θ is substituted with the pseudorapidity $\eta = -\ln\tan\frac{\theta}{2}$, because differences in pseudorapidity $\Delta\eta$ are approximately Lorentz invariant under boosts along z. The distance between two particles is

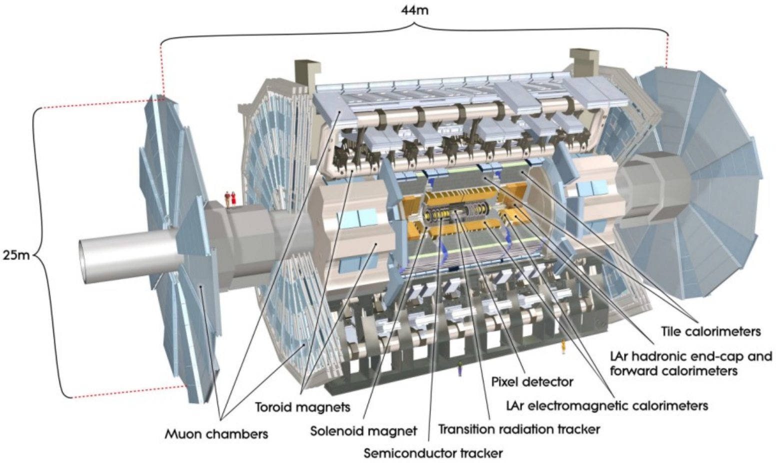

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

Figure 4.2: Schematic view of the ATLAS experiment [6].

defined as $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$.

The ATLAS detector covers the full azimuthal range and $|\eta| < 4.9$. Only some areas are not covered because of support structures and read out channels preventing the installation of further detectors.

4.2.1 The Inner Detector

The Inner Detector (ID) is the detector closest to the interaction point. It comprises an insertable B-Layer (IBL), pixel detector, the strip semiconductor tracker (SCT) and the transition radiation tracker (TRT). The whole ID is encapsulated in a $2\ \mathrm{T}$ magnetic field. This leads to the tracks of charged particles to bend, which allows to infer the charge of the particle from the direction of the curvature. In addition, the measurement of the tracks allows to deduce the transverse momentum p_T of the particle.

The IBL is used to track charged particles, especially B hadrons. The IBL is surrounded by the pixel detector and SCT. Both detectors use silicon semiconductors to measure charged particles. The pixel detector has a good 3D resolution by default. The SCT is comprised of stacked strip layers. These layers are rotated by $40\ \mathrm{mrad}$ to another to achieve a 2D resolution, but this is not as precise as the pixel detector. The pixel detector and SCT achieve a coverage of $|\Delta\eta| < 2.5$. Lastly particles pass the TRT. The TRT is used to identify electrons. For this task the TRT is built from layers of drift tubes filled with ionising gas. These tubes are interleaved with materials with different refraction indices, which generates distinct signatures for the particles and allows for their identification. The TRT covers $|\Delta\eta| < 2$.

Because the ID is located so close to high energy collisions, it is prone to radiation damage. To counter this the ID is cooled down with dry nitrogen to between -5 and $-10\ ^\circ\mathrm{C}$. The ID achieves an overall momentum resolution of $\frac{\sigma_{p_T}}{p_T} = 0.05\% p_T \oplus 1\%$ [GeV].

4.2.2 The Calorimeters

The calorimeters are used to measure the energy of particles interacting electromagnetically and hadronically. In ATLAS, sampling calorimeters are used. This means the calorimeters are built by alternating passive and active layers. The passive layers develop showers from the incoming particles. Therefore, lead, steel and tungsten are used as absorber materials. After passing the passive layers the energy of the shower particles is measured by liquid argon or scintillating tiles in the active layer. As a result most of the energy of the measured particles is depleted at the end of the calorimeter. Because of this, calorimeters are referred to as destructive detectors. The ATLAS calorimeter system is comprised of two barrel and two end-cap calorimeters.

The first and innermost barrel calorimeter is the Electromagnetic Calorimeter (ECAL). It is mainly used to measure the energy of electrons and photons. The shower produced in the passive layer stems from Bremsstrahlung and pair production of electrons and positrons by the photon. During these processes the electrons and photons of the shower sequentially loose energy. Therefore the process is limited because Bremsstrahlung only takes place until a threshold energy is reached. Because of this limit, the ECALs size is also assessable. The ECAL is built inside a liquid argon chamber, which is also used as active layers. For the passive layer lead is used. The ECAL covers the range $|\Delta\eta| < 1.475$ with an accuracy of $\frac{\sigma_E}{E} = \frac{10\%}{\sqrt{E}} \oplus 0.7\%$ [GeV].

The second barrel calorimeter is the hadronic tile calorimeter (HCAL). This detector is used to measure the energy of strongly interacting hadrons. The shower process is similar to the one in ECAL with the difference that hadrons interact with the absorber material instead of electrons or photons. This interaction is a strong interaction and produces either hadrons or gluons. However also electroweak decays are possible, resulting in hadronic showers often containing electromagnetic showers as well. The HCAL can measure energies with a resolution of $\frac{\sigma_E}{E} = \frac{50\%}{\sqrt{E}} \oplus 3\%$ [GeV] in a region of $|\Delta\eta| < 1.7$ in hadronic showers.

The end-cap calorimeters encompass a Forward calorimeter (FCal), an Electromagnetic end-cap calorimeter (EMEC) and hadronic end-cap calorimeter (HEC). Furthermore the FCal is a combination of one EM calorimeter and two hadron calorimeters. The FCal, EMEC and HEC are located at the end of the detector to extend the range in which particles can be tracked. They cover a combined range of $1.375 < |\Delta\eta| < 4.9$ and work on the same principals as the before mentioned calorimeters. Liquid argon and scintillating tiles are used as active layers as well as lead, steel and tungsten as absorber. The EMEC covers a range of $1.375 < |\Delta\eta| < 3.2$ with a resolution of $\frac{\sigma_E}{E} = \frac{10\%}{\sqrt{E}} \oplus 0.7\%$ [GeV]. The HEC measures hadrons with a resolution of $\frac{\sigma_E}{E} = \frac{50\%}{\sqrt{E}} \oplus 3\%$ [GeV] in a region of $1.375 < |\Delta\eta| < 3.2$. The FCal extends the coverage of ATLAS calorimeter to a range of $3.1 < |\Delta\eta| < 4.9$ at a resolution of $\frac{\sigma_E}{E} = \frac{100\%}{\sqrt{E}} \oplus 10\%$ [GeV].

4.2.3 The Muon Spectrometer

The Muon Spectrometer (MS) is the outermost detector of the ATLAS experiment. It is designed to detect particles that are not stopped and detected by the calorimeter. Such particles are predominantly muons μ. Muons are not detected in the calorimeters because they are not affected much by Bremsstrahlung due to their high mass. To detect them, gas filled drift chambers inside a magnetic field with 0.5 T for the barrel and 1 T for the end-cap are used. Muons that enter the drift chambers ionise the gas, which causes an ionisation cascade that is measurable. Because of the magnetic field the measured tracks are bend, so the transverse momentum of the particle can be reconstructed.

The MS is able to measure the momentum of muons in the barrel at $|\Delta\eta| < 1$, in the end-cap at $|\Delta\eta| < 2$ and at $2 < |\Delta\eta| < 2.7$ with higher granularity. Trigger chambers for muons are also part of the MS and cover $|\Delta\eta| < 2.4$. For muons with a transverse momentum of $p_T = 1$ TeV the relative resolution of the MS is around 10%.

4.2.4 The Trigger

Due to the high number of collisions taking place at LHC, a large amount of data is generated, approximately 60 TB/s. Therefore, a trigger system is used to reduce the amount of data which has to be stored and analyzed.

First, a hardware based Level-1 trigger (L1) is used. L1 makes fast decisions at a low resolution, mainly by searching for high energy measurements in the calorimeters or muon spectrometer to define regions of interest. This step already reduces the data stream from 40 MHz to $100 - 1000$ kHz.

Secondly, a software based High-Level trigger (HLT) is used. This trigger makes much more complex decisions, as it associates track information from the ID with energy deposits in the remaining detector, to reconstruct different physics objects. In the end, the data stream is reduced to $1 - 10$ kHz and permanently saved.

Chapter 5

The $X \to HH \to b\bar{b}VV^{(*)}$ channel

5.1 The Data

In this analysis, a search for resonant pair production of two Higgs bosons is performed in the fully boosted decay channel $X \to HH \to b\bar{b}VV^{(*)}$ with 1 lepton in the final state. As seen in Figure 5.1, this channel is characterised by one Higgs boson decaying to two b-jets, $H \to b\bar{b}$,and the other decaying via $H \to WW^{*}$. In the 1 lepton final state, one of the W bosons decays hadronically (W_{had}), while the other decays leptonically (W_{lep}). Furthermore, the event selection employed targets the leptonic decay $W \to \mu\nu$ for W_{lep}. In the fully boosted topology, the W_{had} is reconstructed as a single jet. Furthermore, it is expected that the lepton is reconstructed inside the W_{had}-jet. In addition, no distinction is made on the W bosons being off-shell or on-shell.

This topology is relevant for resonant HH production and targets high masses. This means only a fraction of the event energy m_X is used during the production of the masses m_H of the Higgs bosons HH. Therefore the search targets scenarios where m_X exceeds 1 TeV significantly are used. As a result, the further decay products have a high momentum and cannot be resolved by the detector, resulting in this fully boosted topology.

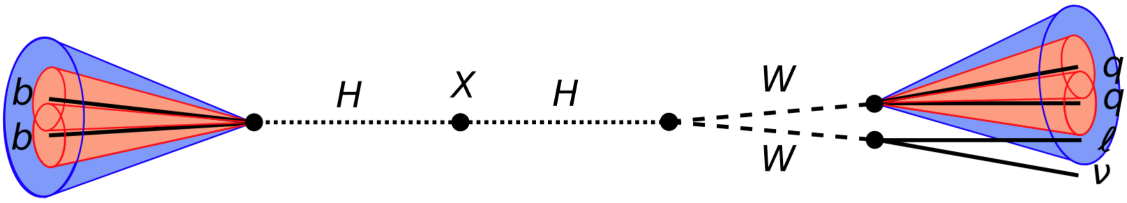


Figure 5.1: $X \to HH \to b\bar{b}VV^{(*)}$ schematic drawing with 1 lepton final state [4].

To train a model to separate signal from background, labelled events are needed. The events used in this analysis are generated using Monte Carlo simulations. During these simulations, labelled particle tracks and interactions are simulated with the corresponding detector readout (see

Section 5.1.1). Afterwards the detector responses are reconstructed for particles and other objects such as jets. A preselection is applied to the events (see Section 5.1.2). Kinematic distributions of the selected data is shown in Section 5.1.3. The Monte Carlo simulations, reconstruction and preselection follow the design implemented in Ref. [4].

5.1.1 Monte Carlo Simulation

The samples for the Higgs boson pairs are generated by simulating a spin-0 heavy resonance decaying to two SM Higgs bosons $X \to HH$ with $m_H = 125$ GeV. Samples are generated for $m_X = 0.8, 0.9, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.5, 3, 4$ and 5 TeV. For all of these events a cross section of 1 pb is assumed. Initial $X \to HH$ samples are calculated at leading order in α_S using MADGRAPH [54] with the NNPDF2.3LO PDF [55] set. Hadronization and parton showers are then simulated using HERWIG7 [56] with EVTGEN [57]. Detector responses are simulated using the AFII detector simulation [58]. During simulation, a filter is applied to require one decay to be $H \to b\bar{b}$ and the other either $H \to WW^{(*)}$ or $H \to ZZ^{(*)}$. In addition, the branching ratio of each H is constrained to be 50% to $b\bar{b}$ and 50% to $VV^{(*)}$ Finally by requiring that exactly one lepton and neutrino is present in the final state, $H \to ZZ^{(*)}$ samples are excluded.

As background the following processes are simulated: massive vector boson production together with jets (W and Z), top quark pair production ($t\bar{t}$), QCD, single top production including a t, s and Wt channel (single t), vector boson pair production (diboson) and single Higgs production (single H). The W, Z and diboson samples are generated at next-to-next-to leading order, $t\bar{t}$ and single t background samples are simulated at leading order. For the QCD samples, a data driven approach was used.

For a full overview of all simulations and filters used for the samples, see Ref. [4]. To note is that for this analysis SHERPA 2.2.11 is used instead of SHERPA 2.2.1 for the W and Z background samples.

5.1.2 Reconstruction and Preselection

During data taking, the detectors only readout particle tracks and energy deposits. These tracks and energy deposits are then associated with particle signatures, to make the data human readable. For this task, different reconstruction algorithms are used. Once particle signatures are reconstructed, an event preselection is applied to the data.

Jets

Because of the many different hadrons with similar signature, depositing energy in the hadronic calorimeter, not single objects are reconstructed, but groups of particles called jets. These jets are associated with the originally decaying particle like the W_{had} and in the best case only encompass decay products of this process. In this analysis TAR jets [59] are used. These TAR jets are clustered using the anti-k_t algorithm [60] with a size parameter of $R = 0.75$ as large-R jets. They are required to have $p_{\mathrm{T}} > 100$ GeV and $|\eta| < 2$. Furthermore, TAR jets are labelled as W_{had} and $H \to b\bar{b}$, where the W_{had} is the TAR jet closest to the lepton and $H \to b\bar{b}$ the p_{T} leading TAR jet, which is not labelled W_{had}.

Leptons

Due to the background modelling of the boosted topology, only muons (μ) are considered. They are reconstructed by matching a track in the Inner Detector (ID) with a Muon Spectrometer track [61]. Muons in this analysis are required to pass medium identification and tight track only isolation [62]. Furthermore, they must have $p_{\mathrm{T}} > 10$ GeV and $|\eta| < 2.5$.

Missing Transverse Momentum

Because neutrinos cannot be measured by the ATLAS detector, the missing transverse momentum in the final state is assumed to originate from the neutrinos. It is defined as the negative vector sum of all other reconstructed objects, such as jets, photons, electrons and muons. Moreover, in this analysis the missing transverse momentum is given by $p^{\mathrm{met}} = \left(E_{\mathrm{T}}^{\mathrm{miss}}, p_{\mathrm{x}}^{\mathrm{miss}}, p_{\mathrm{y}}^{\mathrm{miss}}, 0\right)$.

b-Tagging

Because the decay signature of hadrons containing a b quark differs significantly from other hadrons, it is possible to extract the quark flavour. To tag b quarks inside a jet, the *DL1r* algorithm is used [63].

Overlap Removal

Because in ATLAS all event reconstructions and identifications run independently on all tracks and energy deposits, overlap removal is needed to remove double counting of energy. As this is a geometrical problem, overlap removal is done by cutting nearby classifications based on the ΔR between two objects.
An overview of the overlap removal used is described in Ref. [4].

Preselection

All preselection criteria applied are listed again in Table 5.1. The criteria are designed to select events that conform with the boosted topology signature and have a precise detector readout. The cuts on the p_{T} of an object and the requirement for two TAR jets, force the event to be consistent with a boosted topology. In addition, cuts are applied on the pseudorapidity η of the object. The preselection is the same as for the cut-based $X \to HH \to b\bar{b}VV^{(*)}$ analysis [4], with the exception that the condition of the lepton being inside or near a jet $\Delta R(\mathrm{lep}, \mathrm{closestJet}) < 1$ is not used, because the selection would drastically reduce the separation power of the $\min \Delta R(\mathrm{lep}, \mathrm{closest\ jet})$ variable.

reconstruction object	Conditions
TAR jets	2 TAR Jets required $p_t^{\mathrm{TAR}} > 100$ GeV $\|\eta^{\mathrm{TAR}}\| < 2$ $p_T^{lead.Jet} > 500$ GeV <= 2 b-tagged PFlow jets
lepton	lepton has to be a muon $p_T^{\mathrm{lep}} > 10$ GeV $\|\eta^{\mathrm{lep}}\| < 2.5$ $\|d_0 \mathrm{significance}\| < 3$ $\|z_0 \sin\theta\| < 0.5$ mm pass *medium ID* [62] pass *FCTightTrackOnly* isolation [62]
$H \to b\bar{b}$	$p_t^{H\to\bar{b}b} > 500$ GeV

Table 5.1: Preselection criteria used.

5.1.3 Kinematic Properties and Distributions of the Dataset

After the simulation, reconstruction and preselection, the events are distributed in the different classes as seen in Figure 5.2. With around 45.28%, the W samples make up the bulk of events. Followed by $t\bar{t}$ events with 28.33%, $HH \to b\bar{b}WW^*$ events with 12.44%, QCD events with 5.95%, single t with 3.52%, Z events around 2.64%, diboson events at 1.72% and lastly single H with around 0.12%. For further analysis, four analysis classes are defined: W, $t\bar{t}$, $HH \to b\bar{b}VV^{(*)}$ and other background. The $HH \to b\bar{b}WW^*$, W and $t\bar{t}$ classes are the respected simulated events and the other background class combines the single H, diboson, Z, single t and QCD samples. These classes were chosen because of the number of events available and their respected significance as signal and background processes. The distribution of these classes can be seen in Figure 5.3. As the W, $t\bar{t}$ and $HH \to b\bar{b}WW^{(*)}$ classes are unchanged, still around 45.28% of events are W events, 28.33% $t\bar{t}$ events and 12.44% $HH \to b\bar{b}WW^*$ events. The other background class contains around 13.95% of events.

As depicted, the new $HH \to b\bar{b}WW^{(*)}$ class contains the least amount of events. Furthermore, the $HH \to b\bar{b}WW^{(*)}$ class is a combination of different mass points $m_X = 0.8$, 0.9 1, 1.2, 1.4, 1.6, 1.8, 2.5, 3, 4 and 5 TeV, but for simplicity a model has to predict only predict the class and not the mass point. The distribution of the $HH \to b\bar{b}WW^{(*)}$ mass point samples can be seen in Figure 5.4. As shown, the lower mass points $m_X = 0.8$, 0.9 and 1 TeV are drastically under-represented. However, the fully boosted topology does not favour such events, so this under-representation is not as drastic, as is it first seems.

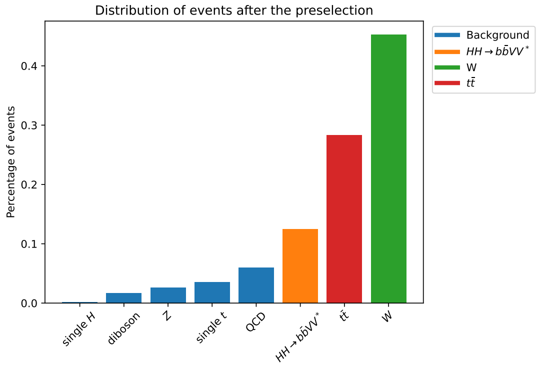


Figure 5.2: Distribution of event classes.

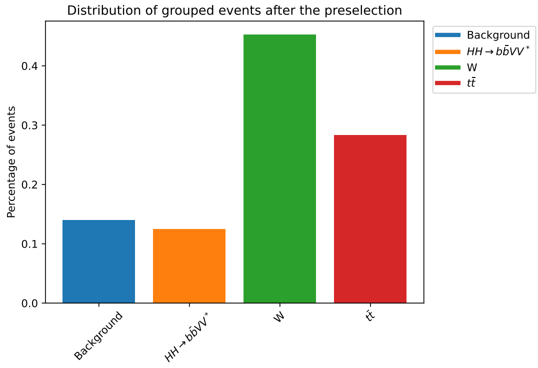


Figure 5.3: Distribution of analysis classes.

Using this simulated dataset to train a model would result in overfitting the W class, because of the bias towards the W class. The low number of $HH \to b\bar{b}WW^{(*)}$ signal events is especially problematic, resulting in an optimiser training on this data only after all other classes are sufficiently learned. As described in Section 3.2, a uniformly distributed dataset is most efficient to train a model. To achieve this without further simulations, the $HH \to b\bar{b}WW^{(*)}$ mass points and the four classes are reweighted.

First the mass points $m_X = 1.2$, 1.4, 1.6, 1.8, 2.5, 3, 4 and 5 TeV are weighted to the same number of events as the $m_X = 2$ TeV mass point. The samples with $m_X = 0.8$, 0.9 and 1 TeV are excluded from this reweighting, because of their drastically smaller number of events and lower probability to pass the preselection. The results of this normalisation can be seen for the mass points in Figure 5.5 and for the analysis classes in Figure 5.6.

Secondly, the $HH \to b\bar{b}WW^{(*)}$, $t\bar{t}$ and other background class are weighted to the same number of events as the W class. The results of this normalisation can be seen in Figure 5.7.

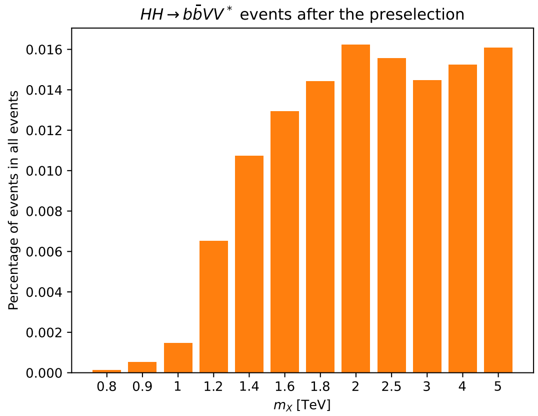


Figure 5.4: Distribution of $HH \to b\bar{b}WW^{(*)}$ mass point events

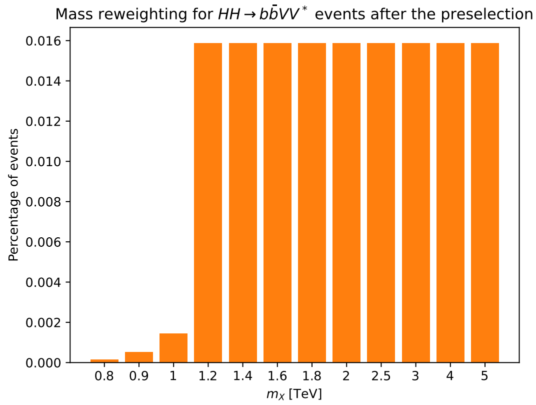


Figure 5.5: Distribution of reweighted $HH \to b\bar{b}WW^{(*)}$ mass point events.

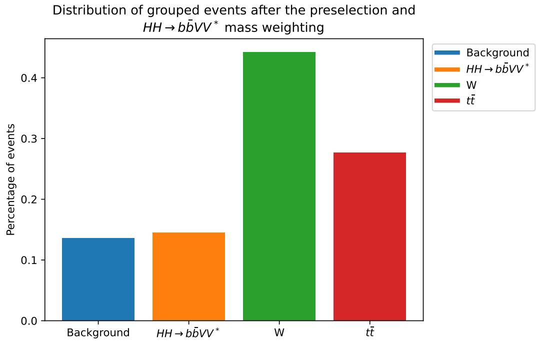


Figure 5.6: Distribution of analysis classes with the $HH \rightarrow b\bar{b}VV^{(*)}$ mass points reweighted.

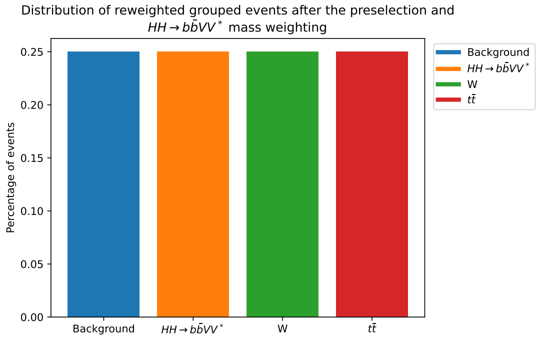


Figure 5.7: Distribution of analysis classes with the $HH \rightarrow b\bar{b}VV^{(*)}$ mass points reweighted and all classes normalised.

It is expected that models trained on this reweighted dataset would not prefer any class or $HH \rightarrow b\bar{b}VV^{(*)}$ mass point during training. Excluded are the $HH \rightarrow b\bar{b}VV^{(*)}$ mass points $m_X = 0.8$, 0.9 and 1 TeV. However, a non weighted dataset with uniformly distributed samples in every class and mass point is expected to outperform this rewighted dataset.

The variables contained in the dataset include low-level kinematic features like the four-vectors of reconstructed objects, high-level features like jet substructure and meta variables like the number of b-tags, which are defined through other algorithms.
The low-level features included in the dataset are:

- **Mass** of the reconstructed jets and lepton: $\boldsymbol{m^{H\rightarrow b\bar{b}}}$, $\boldsymbol{m^{W_{\text{had}}}}$ and $\boldsymbol{m_{\text{T}}^{W_{\text{lep}}}}$
- **Pseudorapidity** of all reconstructed objects: $\boldsymbol{\eta^{H\rightarrow b\bar{b}}}$, $\boldsymbol{\eta^{W_{\text{had}}}}$ and $\boldsymbol{\eta^{\text{lep}}}$
- **Transverse momentum** of all reconstructed objects: $\boldsymbol{p_T^{H\rightarrow b\bar{b}}}$, $\boldsymbol{p_T^{W_{\text{had}}}}$ and $\boldsymbol{p_T^{\text{lep}}}$
- **Missing transverse momentum** for the neutrino is included via $\boldsymbol{E_T^{\text{miss}}}$

The azimuthal angle ϕ for the reconstructed objects is not included in the low-level variables, because it is assumed that events are uniformly distributed over ϕ. However this also means the dataset does not contain four-vector momentum for the reconstructed objects.

The following high-level features are included

- **Jet substructure variables based on Energy Correlation Functions**: $\boldsymbol{C_2^{H\rightarrow b\bar{b}}}$ and $\boldsymbol{C_2^{W_{\text{had}}}}$ [64]
- **Jet substructure variables based on N-subjettiness**: $\boldsymbol{\tau_{42}^{H\rightarrow b\bar{b}}}$ and $\boldsymbol{\tau_{42}^{W_{\text{had}}}}$ [65]
- **Position of the lepton to the jets**: $\boldsymbol{\min \Delta R(\text{lep}, \text{closest jet})}$
- **Mass of the whole process combined** via the visible + met mass:
  $\boldsymbol{m^{HH}_{\text{vis+met}}} = \sqrt{(p^{h\rightarrow b\bar{b}} + p^{W_{\text{had}}} + p^{\text{lep}} + p^{\text{met}})^2}$

These variables where chosen because of their provided separation power, nonetheless detailed analysis on all different kinds of jet substructure variables was not conducted.

Lastly the meta variables contain:

- **Number of b-tagged Jets**: $\boldsymbol{N^{\text{jets}}_{b-\text{tagged}}}$
- **Number of b-tags per jet**: $\boldsymbol{N^{H\rightarrow b\bar{b}}_{b-\text{tags}}}$ and $\boldsymbol{N^{W_{\text{had}}}_{b-\text{tags}}}$
- **Number of tracks per jet**: $\boldsymbol{N^{H\rightarrow b\bar{b}}_{\text{tracks}}}$ and $\boldsymbol{N^{W_{\text{had}}}_{\text{tracks}}}$

To further improve numerical and training stability, all variables are scaled to be within $[0, 1]$ using a *MinMax-Scaler* transformation. The transformation is defined for the variable x as:

$$x_{\text{scaled}} = \frac{x - \min(x)}{\max(x) - \min(x))} \tag{5.1}$$

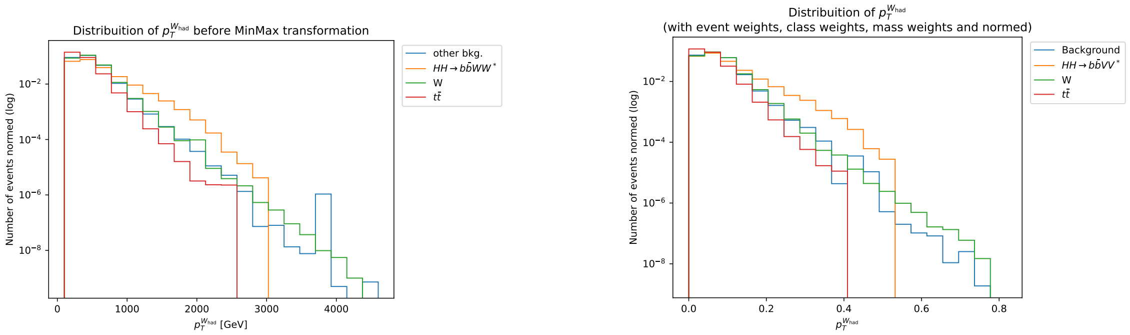
Figure 5.8 shows the distribution of $p_{\text{T}}^{W_{\text{had}}}$ before and after MinMax-Scaler transformation. As can be seen, only the value range changed and not the distribution itself. Therefore, no separation power is lost by applying this transformation.

The separation power of some variables, such as the number of b-tags inside the $H \rightarrow b\bar{b}$ candidate is very easy to see (Figure 5.9). Only two peaks can be observed, with one being dominated by W and the other by $HH \rightarrow b\bar{b}WW^*$ events. Therefore, a separation between these events is visible.

For continuous variables such as $\min \Delta R(\text{lep}, \text{closest jet})$ (Figure 5.10), still one area is dominated by an event, but the separation power constantly changes. $\min \Delta R(\text{lep}, \text{closest jet})$ contains around zero predominantly $HH \rightarrow b\bar{b}WW^*$ events, than shortly $t\bar{t}$ events are slightly the most prominent events, however W events dominate starting from 0.2.

For the classification of $HH \rightarrow b\bar{b}WW^*$ events variables, that contain large ranges, are especially interesting such as the distributions of $m^{W_{\text{had}}}$ (Figure 5.11) and $p_T^{H\rightarrow b\bar{b}}$ (Figure 5.12). The distribution of $p_T^{H\rightarrow b\bar{b}}$ particular could be of great use for $HH \rightarrow b\bar{b}WW^*$ classification, because in the range 0.2 until 0.7 $HH \rightarrow b\bar{b}WW^*$ are the most probable event.

Distributions for all variables can be found in the Appendix .1.



(a) before the MinMax-Scaler transformation

(b) after the MinMax-Scaler transformation

Figure 5.8: Comparison of $p_{\mathrm{T}}^{W_{\mathrm{had}}}$ (a) before and (b) after the MinMax-Scaler transformation with the class and mass point reweighting applied.

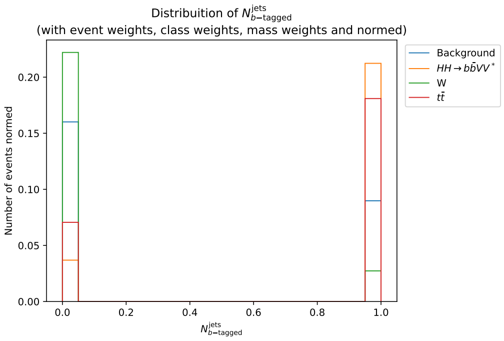
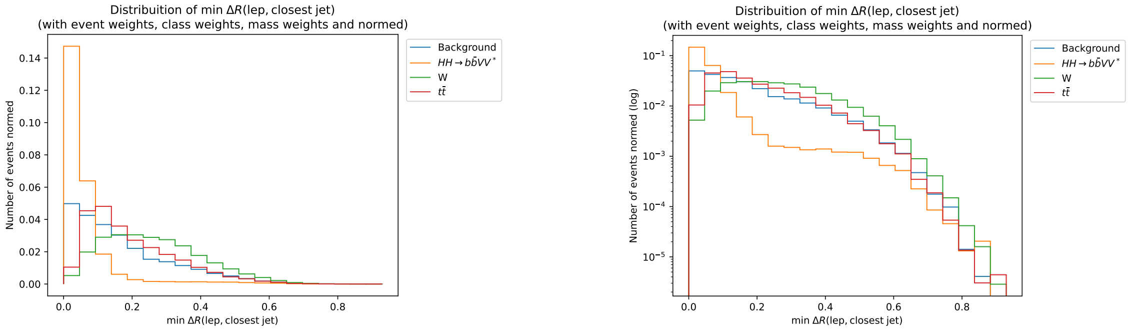


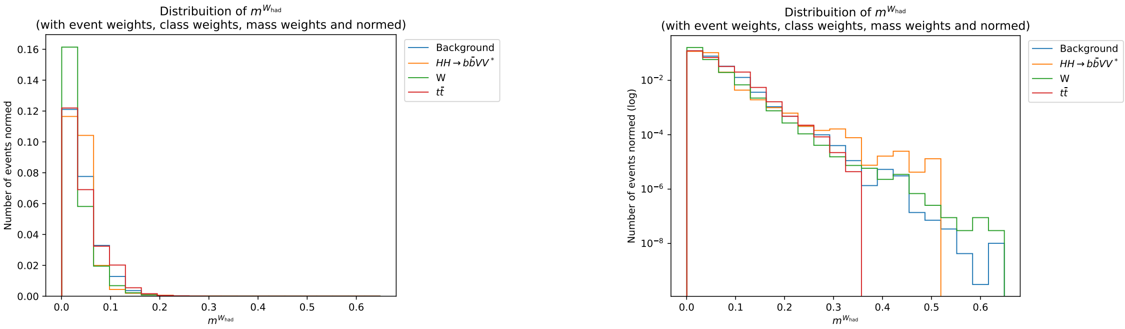
Figure 5.9: Normalised distribution of $N_{b-\mathrm{tagged}}^{\mathrm{jets}}$ after the MinMax-Scaler transformation with the class and mass point reweighting applied.



(a)

(b)

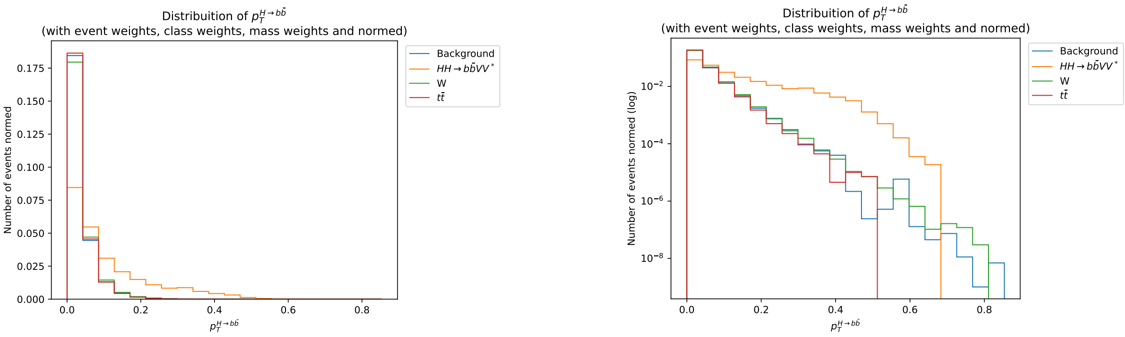
Figure 5.10: Normalised distribution of $\min\ \Delta R(\mathrm{lep}, \mathrm{closest\ jet})$ after the MinMax-Scaler transformation with the class and mass point reweighting applied, in (a) linear and (b) log scale.



(a)

(b)

Figure 5.11: Normalised distribution of $m^{W_{\mathrm{had}}}$ after the MinMax-Scaler transformation with the class and mass point reweighting applied, in (a) linear and (b) log scale.



(a)

(b)

Figure 5.12: Normalised distribution of $p_T^{H\to b\bar{b}}$ after the MinMax-Scaler transformation with the class and mass point reweighting applied, in (a) linear and (b) log scale.

Chapter 6

Optimization of Signal Extraction

6.1 The cut-based selection

Originally a cut-based, binary selection was proposed to separate the $HH \to b\bar{b}WW^*$ signal from all backgrounds [4]. These cuts were human designed and are based on the physical and statistical knowledge about the process.

For an event to pass as a signal $X \to HH \to b\bar{b}WW^*$ event, it is first required that the $H \to b\bar{b}$ candidate is reconstructed as a single, b-tagged TAR jet. Furthermore, this b-tagged $H \to b\bar{b}$ candidate is required to be the only b-tagged TAR jet in the event reconstruction. However, no distinction is made on the $H \to b\bar{b}$ candidate having one or two b-tags. This is done to increase the signal efficiency, as the efficiency of the used b-tagging working point would result in discarding a non-negligible number of signal events, when two b-tags per $H \to b\bar{b}$ candidate would be required. Nonetheless, the signal significance for $H \to b\bar{b}$ candidates with two b-tags is better. Therefore, keeping the one and two b-tagged $H \to b\bar{b}$ candidates separated in two regions, allows to exploit the signal significance in the two b-tagged region and the added signal efficiency of the single b-tagged events [4].

In addition to these conditions, three window cuts are applied to separate signal and background events. These cuts are designed to take advantage of differences in variables stemming from the boosts of the final state products. For the $X \to HH \to b\bar{b}WW^*$ process, the following variables are used: $m^{H\to b\bar{b}}$, $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ and $C_2^{H\to b\bar{b}}$. To perform a cut on these variables, the dependency on the p_{T} of the related object is used. As an example, the $H \to b\bar{b}$ mass cut depends on $p_{\mathrm{T}}^{H\to b\bar{b}}$. Moreover, two cuts are used as windows instead of a single cut. Therefore a region is defined in which the signal resides.

The window cuts were constructed for an object by the following steps:

1. All events that pass the preselection in 5.1.2 and contain exactly one b-tagged TAR jet are

grouped in a two-dimensional histogram consisting of the p_{T} of the considered objected ($m^{H\to b\bar{b}}$, $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ or $C_2^{H\to b\bar{b}}$) and variable of interest. To include all signal events, a range of 0 GeV $\leq p_{\mathrm{T}} \leq 5000$ GeV with a bin width of 10 GeV is used.

2. The p_{T} bins are then merged until the statistical uncertainty per bin is less then 5%, to ensure sufficient statistics.

3. In every bin the smallest window containing a certain percentage of signal events is constructed around the median of the variable of interest.

4. These upper and lower bounds for all p_{T} bins are then used to fit a function, which is used as a smooth window cut.

For an event to be classified as a signal $X \to HH \to b\bar{b}WW^*$ event, it has to pass the following criteria:

1. pass the 70% efficiency $m^{H\to b\bar{b}}$ window cut with 1 or 2 b-tags in the $H \to b\bar{b}$ candidate
   or
   fail the 80% $m^{H\to b\bar{b}}$ window cut with 1 b-tag in the $H \to b\bar{b}$ candidate

2. pass the 80% efficiency $C_2^{H\to b\bar{b}}$ window cut

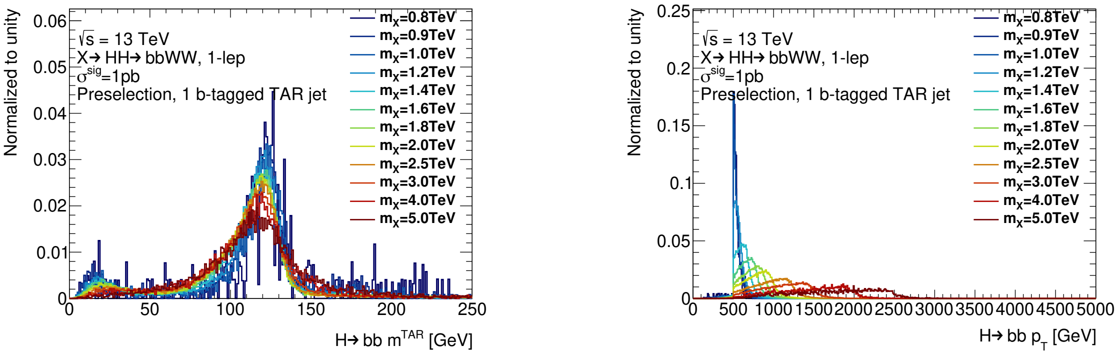
3. pass the 80% efficiency $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ window cut

For the $m^{H\to b\bar{b}}$ window cut, the relation between the mass of the $H \to b\bar{b}$ candidate $m^{H\to b\bar{b}}$ and the corresponding transverse momentum $p_{\mathrm{T}}^{H\to b\bar{b}}$ is exploited. Figure 6.1 shows the distribution of $m^{H\to b\bar{b}}$ and $p_{\mathrm{T}}^{H\to b\bar{b}}$ over all simulated m_X mass points. As seen, all $m^{H\to b\bar{b}}$ signals peak around m_H even though the shape of the peak varies with m_X, and for low m_X, a second very small peak in the low $m^{H\to b\bar{b}}$ region can be observed. Furthermore, the correlation between m_X and $p_{\mathrm{T}}^{H\to b\bar{b}}$ is shown.

Due to $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ being dependent on two variables, multiple potential transverse momenta are considered: $p_{\mathrm{T}}^{\mathrm{lep}}$, $p_{\mathrm{T}}^{W_{\mathrm{had}}}$ and $|p_{\mathrm{T}}^{H_{\mathrm{vis}}}| = |p_{\mathrm{T}}^{\mathrm{lep}} + p_{\mathrm{T}}^{W_{\mathrm{had}}}|$. It was found that $p_{\mathrm{T}}^{H_{\mathrm{vis}}}$ performs the best (for a detailed analysis see Ref. [4]). The distribution of $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ can be seen in Figure 6.2a.

For the $C_2^{H\to b\bar{b}}$ window cut, the relation between the $H \to b\bar{b}$ candidates jet substructure variable $C_2^{H\to b\bar{b}}$ and transverse momentum $p_T^{H\to b\bar{b}}$ is used. Figure 6.2b shows the distribution of $C_2^{H\to b\bar{b}}$. As seen, this variable does not separate as significantly as the $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ signal and background.

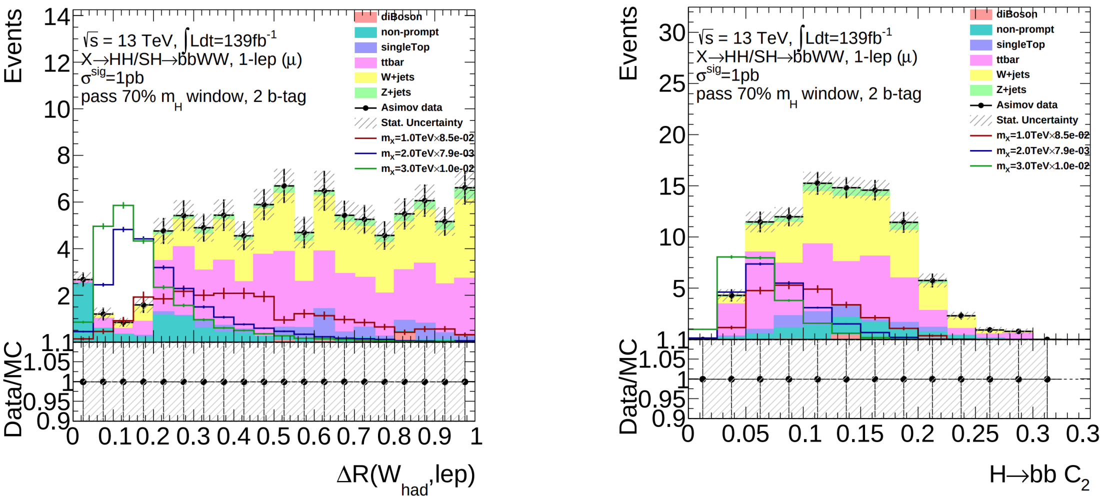
The cuts, except for the $C_2^{H\to b\bar{b}}$ cuts, are fitted on the $X \to HH \to b\bar{b}WW^*$ events discussed in Section 5. For the $C_2^{H\to b\bar{b}}$ cuts $X \to SH \to b\bar{b}WW^*$ events are used, as it was found, that cuts derived on these events perform equal or better to cuts fitted on $X \to HH \to b\bar{b}WW^*$ events (for a detailed analysis see Ref. [4]). The final cuts for $m^{H\to b\bar{b}}$, $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ and $C_2^{H\to b\bar{b}}$ can be seen in Figure 6.3.



(a) $m^{H\rightarrow b\bar{b}}$

(b) $p_{\mathrm{T}}^{H\rightarrow b\bar{b}}$

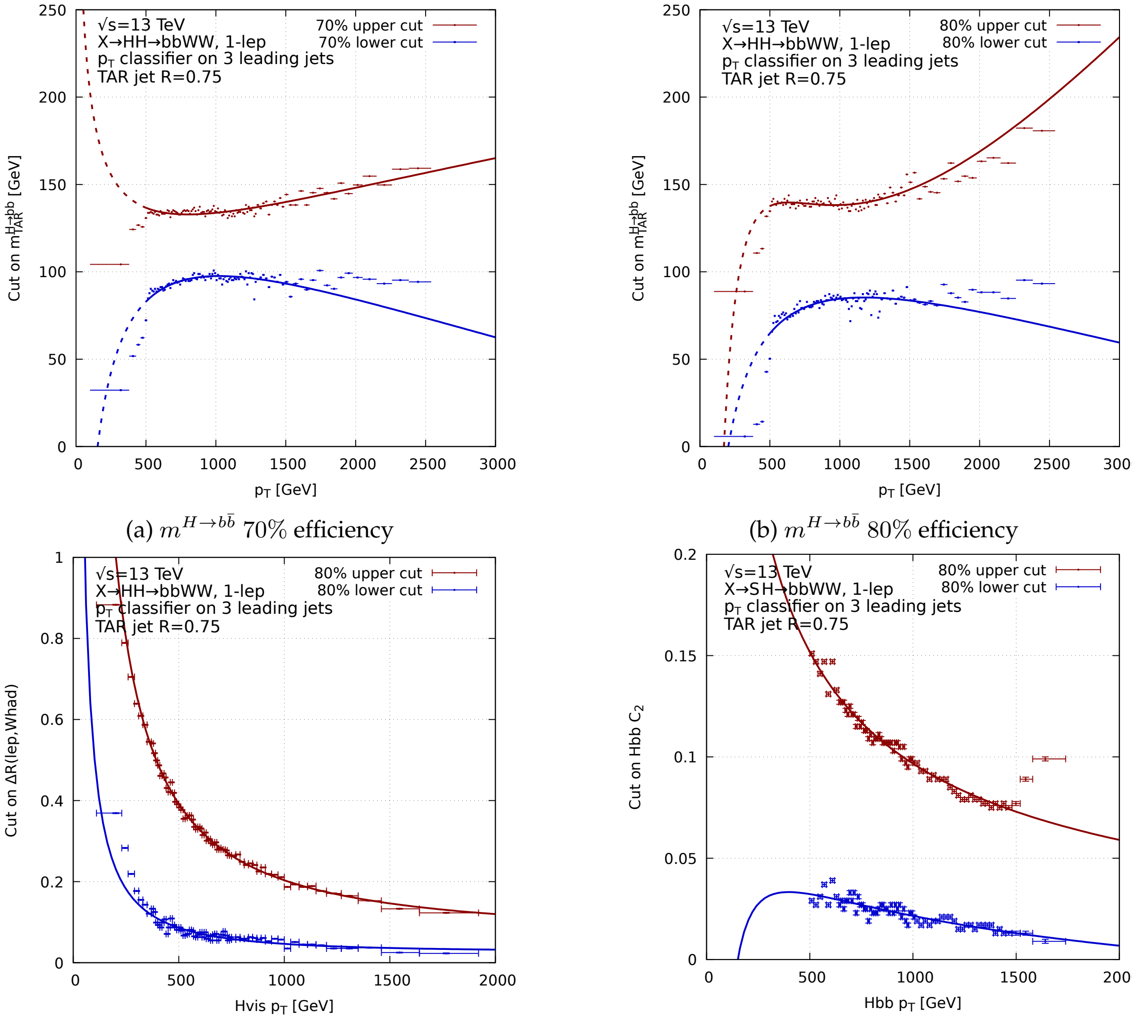
Figure 6.1: Normalised distributions of $m^{H\rightarrow b\bar{b}}$ and $p_{\mathrm{T}}^{H\rightarrow b\bar{b}}$ of the $H \rightarrow b\bar{b}$ candidate for the individual signal samples. The preselection except the $p_{\mathrm{T}} > 500$ GeV cut is applied. Furthermore, exactly one b-tagged jet in the event is required [4].



(a) $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$

(b) $C_2^{H\rightarrow b\bar{b}}$

Figure 6.2: Distributions of $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ and $C_2^{H\rightarrow b\bar{b}}$ for optimising the selection of HH production in SRp2, the dominant signal region. The shown Asimov data is the sum of expected background events. The shown signal is scaled to match 25% of the integral of all backgrounds [4].



(a) $m^{H\to b\bar{b}}$ 70% efficiency

(b) $m^{H\to b\bar{b}}$ 80% efficiency

(c) $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ 80% efficiency

(d) $C_2^{H\to b\bar{b}}$ 80% efficiency

Figure 6.3: Cuts on $m^{H\to b\bar{b}}$, $\Delta R\,(\mathrm{lep}, W_{\mathrm{had}})$ and $C_2^{H\to b\bar{b}}$ at the used efficiencies. The dashed part of the fit will not be used in the analysis due to the applied $p_{\mathrm{T}}^{H\to b\bar{b}} > 500$ GeV cut in the preselection [4].

6.2 Neural Network Setup and Training

In contrast to the cut-based approach, a model is proposed to not only separate signal and background binary, but predict the four analysis classes $HH \to b\bar{b}WW^*$, W, $t\bar{t}$ and background from Section 5.1.3. To achieve this goal, a Feed Forward Neural Network is proposed. This Neural Network is trained on the Monte Carlo samples described in Section 5.1.1.

6.2.1 Architecture

The model architecture proposed is a Feed Forward Neural Network utilising Multilayer Perceptrons. Based on the Universal Approximation theory [35], a width of 500 units per layer and a depth of 5 hidden layers was chosen, in order to be wide enough to approximate a possible underlying continuous function. After each layer of perceptrons, Batch Normalisation is applied to avoid vanishing gradients (see Section 3.2.4). Lastly a ReLU is used as an activation function, before the output of the layer is propagated to the next layer.

As input all 21 variables presented in Section 5.1.3 are used. For events to be fed into the model, they have to pass the preselection and have the class and m_X reweighting applied (see Section 5.1.3).

Because four classes have to be predicted, the model outputs four probabilities, with each representing one class. To achieve this, the final layer consists out of 4 perceptrons with a Softmax activation function. For an input x_i, $(i = 1, \ldots, 4)$, the Softmax function is defined as:

$$y_i = \mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{4} e^{x_j}} \tag{6.1}$$

resulting in $0 \leq y_i \leq 1$ and $\sum\limits_{i=1}^{4} y_i = 1$.

By taking the maximum output probability, an event can be predicted to originate from one of the analysis classes: $HH \to b\bar{b}WW^*$, W, $t\bar{t}$ or other background. However, the output probability for one class is dependent on the other classes, resulting in only one output probability being the highest. In this case this behaviour is desired, as an event can only come from one class.

A complete overview over the model architecture is listed in Table 6.1.

6.2.2 Training

The model is trained on the data discussed in Section 5.1.3. This means the four classes W, $t\bar{t}$, $HH \to b\bar{b}WW^*$ and other background are reweighted to be uniformly distributed. In addition, the $HH \to b\bar{b}WW^*$ samples for the mass points $m_X = 1.2$, 1.4, 1.6, 1.8, 2.5, 3, 4 and 5 TeV are weighted to be also uniformly distributed, except for the mass points $m_X = 0.8$, 0.9 and 1 TeV. During training 70% of the event samples are visible to the model and 20% of the data is reserved for testing. Furthermore, 10% of the training data is used as validation during training and not for optimisation.

To train the model, the ADAM optimiser with decoupled weight decay and AMSGRAD is used (see Section 3.2.3 and Algorithm 3). As a loss function, the Cross Entropy loss was chosen, due to the reasons presented in Section 3.2.1. For four classes with the model prediction $x \in [0, 1]^4$ and

Layer	Dimension
Perceptrons Batchnorm ReLU	$\mathbb{R}^{21} \rightarrow \mathbb{R}^{500}$
Perceptrons Batchnorm ReLU	$\mathbb{R}^{500} \rightarrow \mathbb{R}^{500}$
Perceptrons Batchnorm ReLU	$\mathbb{R}^{500} \rightarrow \mathbb{R}^{500}$
Perceptrons Batchnorm ReLU	$\mathbb{R}^{500} \rightarrow \mathbb{R}^{500}$
Perceptrons Batchnorm ReLU	$\mathbb{R}^{500} \rightarrow \mathbb{R}^{500}$
Perceptrons Softmax	$\mathbb{R}^{500} \rightarrow \mathbb{R}^{4}$

Table 6.1: Overview of the Feed Forward Neural Network architecture proposed.

truth label $t \in \{0,1\}^4$, the Cross Entropy loss is defined as:

$$\text{CrossEntropyLoss}(x, y) = -\sum_{i=1}^{4} t_i log(x_i) \tag{6.2}$$

To get a better insight into the model's decision making, three different trainings were conducted. The first two trainings use all background classes, but each using only a singular mass point m_X for the $HH \rightarrow b\bar{b}WW^*$ signal class. The mass point $m_X = 2$ TeV was chosen, as it is the mass point with the most Monte Carlo event samples of the signal class and therefore not affected by the $HH \rightarrow b\bar{b}WW^*$ reweighting. The mass point $m_X = 1.4$ TeV was also chosen, due to its lower number in samples and to see if for lower mass points, different variables are used by the model. Lastly, a model is trained on all events with the mass point and class reweighting applied. This training is the main model proposed, as for this model, the best performance over all mass points is expected.

Chapter 7

Performance Studies

7.1 Cut-based Model

As the cut-based model separates the $HH \to b\bar{b}WW^*$ and all background events in two classes, in the following "complete background" will refer to the W, $t\bar{t}$ and the other background class combined.

To evaluate the performance of the cut-based model, the confusion matrix shown in Figure 7.1a is a good first indicator. For a truth class t and predicted class p, the following quantity is calculated:

$$\frac{\text{Number of } t \text{ events predicted as class } p}{\text{Number of events in } t} \tag{7.1}$$

In this case, the matrix contains $HH \to b\bar{b}WW^*$ and complete background as classes. The values on the diagonal are referred to as efficiency. The efficiency for the $HH \to b\bar{b}WW^*$ class is defined as:

$$\text{efficiency}\left(HH \to b\bar{b}WW^*\right) = \frac{\text{Number of correctly predicted } HH \to b\bar{b}WW^* \text{ events}}{\text{Number of } HH \to b\bar{b}WW^* \text{ events}} \tag{7.2}$$

and analogously for the complete background class.

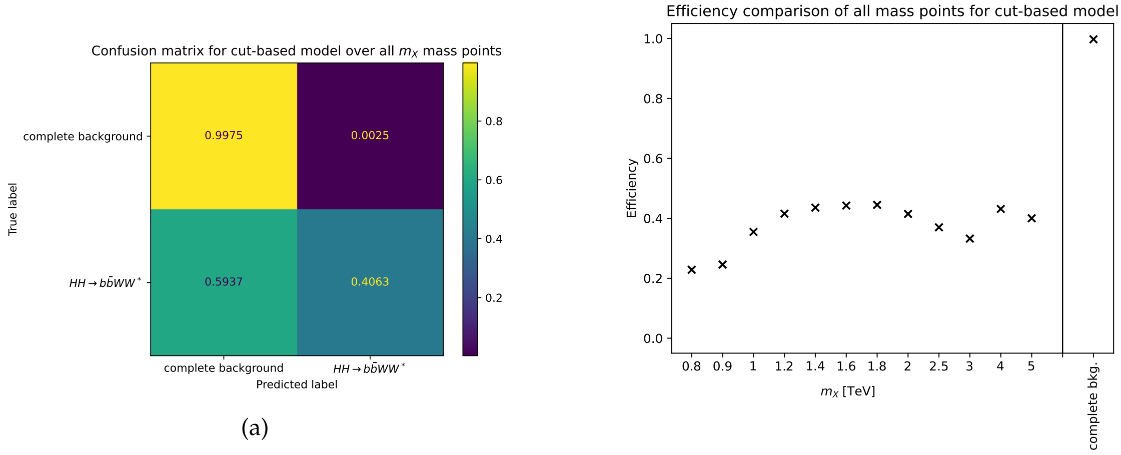
As seen, the complete background is predicted correctly with an efficiency of around 99.75% and is therefore almost perfectly predicted. In contrast, the $HH \to b\bar{b}WW^*$ class is not even predicted for half of the events correctly, with the cut-based model only achieving a efficiency of around 40.63% for $HH \to b\bar{b}WW^*$ events.

When examining the $HH \to b\bar{b}WW^*$ efficiency for the different mass points (see Figure 7.1b), it can be seen that some mass points do not even achieve an efficiency of 40%. Drastic outliers are the mass points $m_X = 0.8$ and 0.9 TeV, with the first one only reaching an efficiency of around 22.82% and the latter around 24.56%. However, starting from the mass point of $m_X = 1$ TeV, a notable increase in efficiency is observable, until $m_X = 1.8$ TeV at around 44.5%. Afterwards, the efficiency drops again up to $m_X = 3$ TeV at an efficiency of about 33.23%. Interestingly, at $m_X = 4$ TeV the efficiency peaks again to around 43.13% and plummets again to around 40.03% at the mass point $m_X = 5$ TeV.

This behaviour at first may seem quite undesired, but when taking into account, that only around 12.43% of the events are $X \to HH \to b\bar{b}WW^*$ samples, an efficiency of approx. 40.63% is respectable in comparison to random guessing. In addition, the even smaller number of samples for the mass points $m_X = 0.8$, 0.9 and 1 TeV explain the lower efficiency for these mass points, as there were fewer mass points to generate the cuts.

All in all, the cut-based model still reaches an accuracy of approximately 92.4%, with the accuracy being a measure for overall model performance and being defined as:

$$\text{accuracy} = \frac{\text{Number of correctly classified events}}{\text{Number of events}} \tag{7.3}$$



(a)

(b)

Figure 7.1: (a) Confusion matrix of the cut-based model over all mass points m_X.
(b) Comparison of efficiency of the cut-based model for all mass points m_X and complete background.

7.2 Single Mass Point Neural Network

Comparing the cut-based model with the Neural Network, it becomes apparent how different the output of the two designs is. The cut-based model only outputs a binary result, i.e. if an event is a signal event or not. However, the Neural Network outputs a probability per class, resulting in an output distribution per class. For the output of class i, it is expected that class i is predicted to be near 1 and the other classes near 0. Furthermore, different decisions can be made to classify an event based on the output distributions. The simplest way would be to classify an event by the maximum predicted class probability. Cuts on the output distributions can also be applied in addition or separately to the maximum, to change the model prediction.

Training on $m_X = 2$ TeV

First a training was conducted with the full background classes W, $t\bar{t}$ and other background, but only the single mass point $m_X = 2$ TeV for the $X \to HH \to b\bar{b}WW^*$ class. This mass point was chosen, as it contains the most events of all signal samples and should therefore be the easiest to learn by the model. In Figure 7.2 the output distribution for the different classes can be seen. As desired, the distribution for the wanted output class peaks at 1, while all others are predicted near 0. However, as seen on the right side of Figure 7.2, for every analysis class some events are completely misclassified. Mainly misclassified are events from the other background class, which is not surprising as the other background class contains four different background processes with changing objects and kinematics and thus non-uniform distribution of samples.

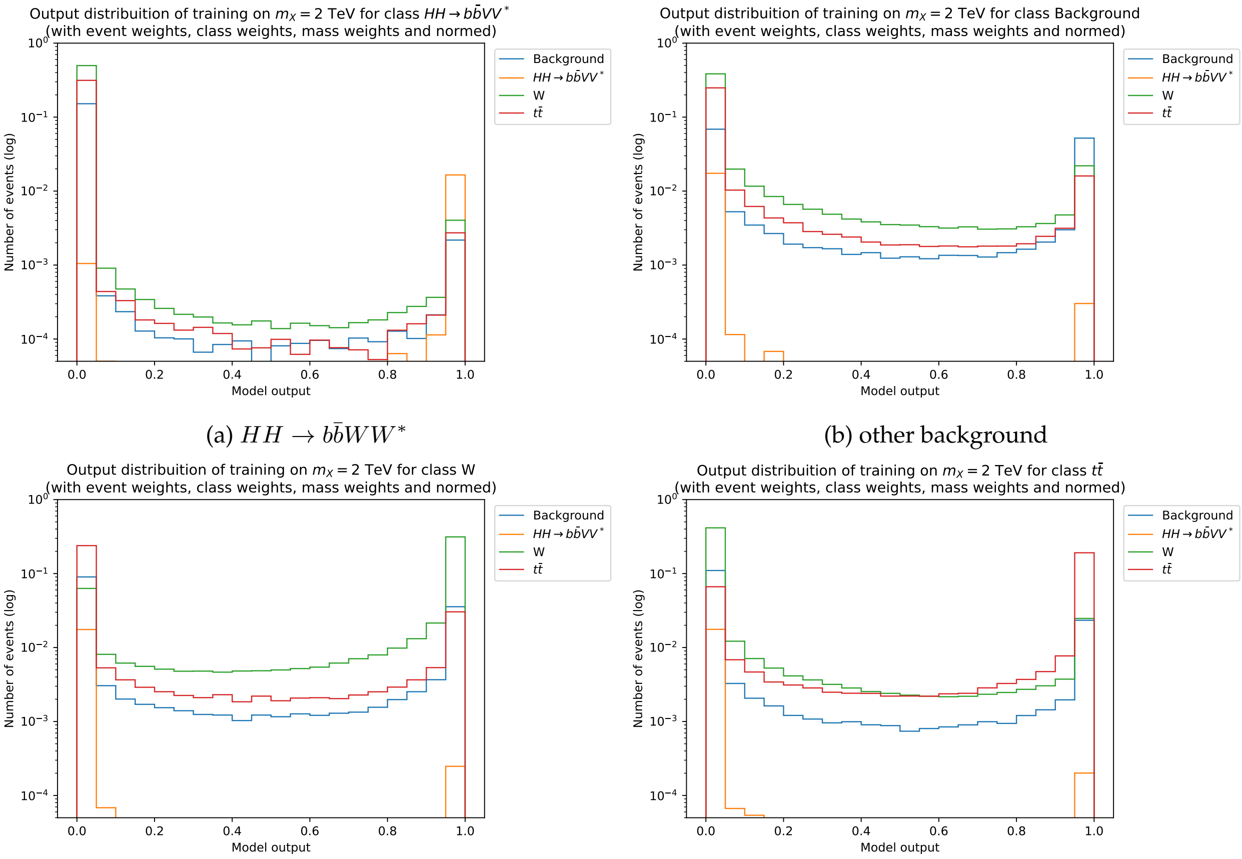
These observations are confirmed when taking a look at the confusion matrix (see Figure 7.3a). To classify an event, the class with the highest predicted output probability is used. As seen again, the other background class is misclassified the most, with more other background events being classified as being W or $t\bar{t}$ samples than being in the other classes. However, it is also shown that the $HH \to b\bar{b}WW^*$ signal class has an efficiency of approx. 92.9%. Interestingly, $HH \to b\bar{b}WW^*$ events are almost as often misclassified as $t\bar{t}$ and other background events.

Figure 7.3b shows the permutation importance [66, 67] of the different input variables. The permutation importance is calculated by first computing the loss s of the model, referred to as score in this context. Then, for the input variable i a random permutation is chosen, thereby corrupting this variable in the dataset. Finally a new score s_i is computed and the permutation importance is defined as:

$$\text{permutation importance}(i) = |s - s_i| \tag{7.4}$$

If for a variable the permutation importance is greater then 0, this implies that the variable has a considerable influence over the models decision. On the other hand, if the permutation importance is 0, the variable has an negligible effect on the models performance.
In this case, the input variables with the biggest impact on the model's performance are $N^{\text{jets}}_{b-\text{tagged}}$,

$p_{\mathrm{T}}^{W_{\mathrm{had}}}$, min $\Delta R(\mathrm{lep}, \mathrm{closest\ jet})$ and $m^{W_{\mathrm{had}}}$ in decreasing order.



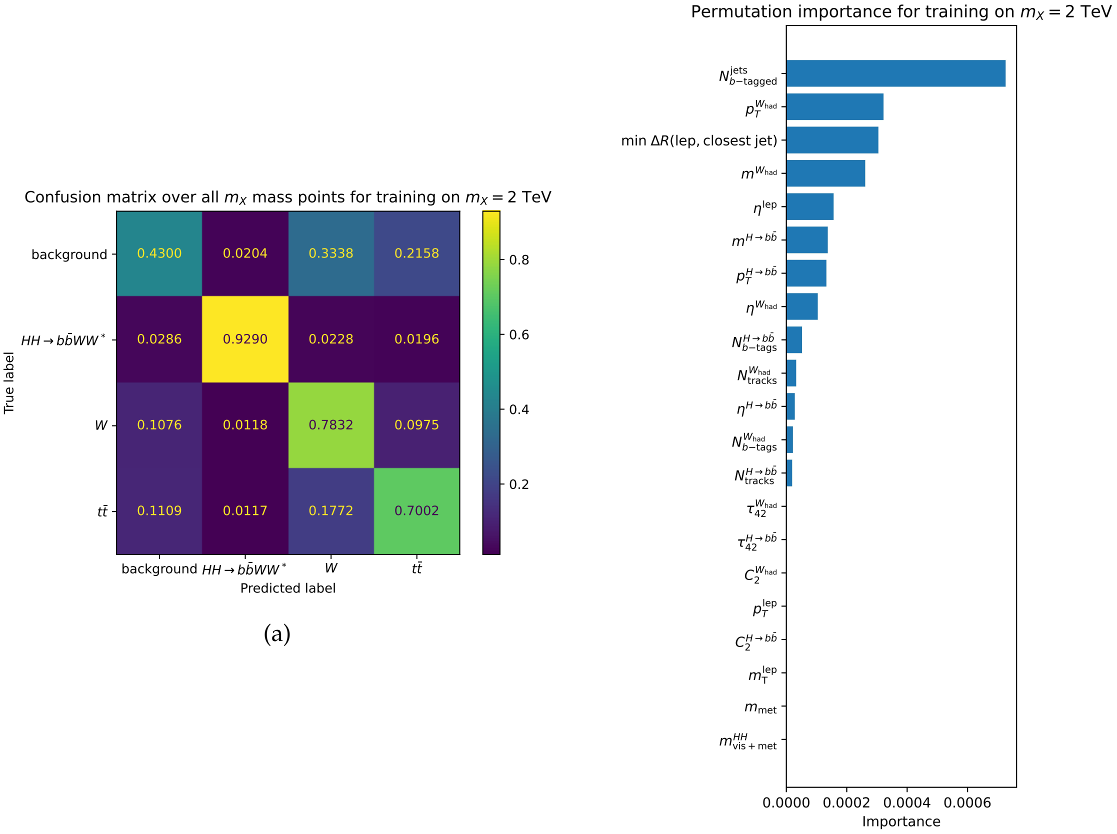
(a) $HH \to b\bar{b}WW^*$

(b) other background

(c) W

(d) $t\bar{t}$

Figure 7.2: Distribution of Neural Network output trained on only the $m_X = 2$ TeV mass point.



(a)

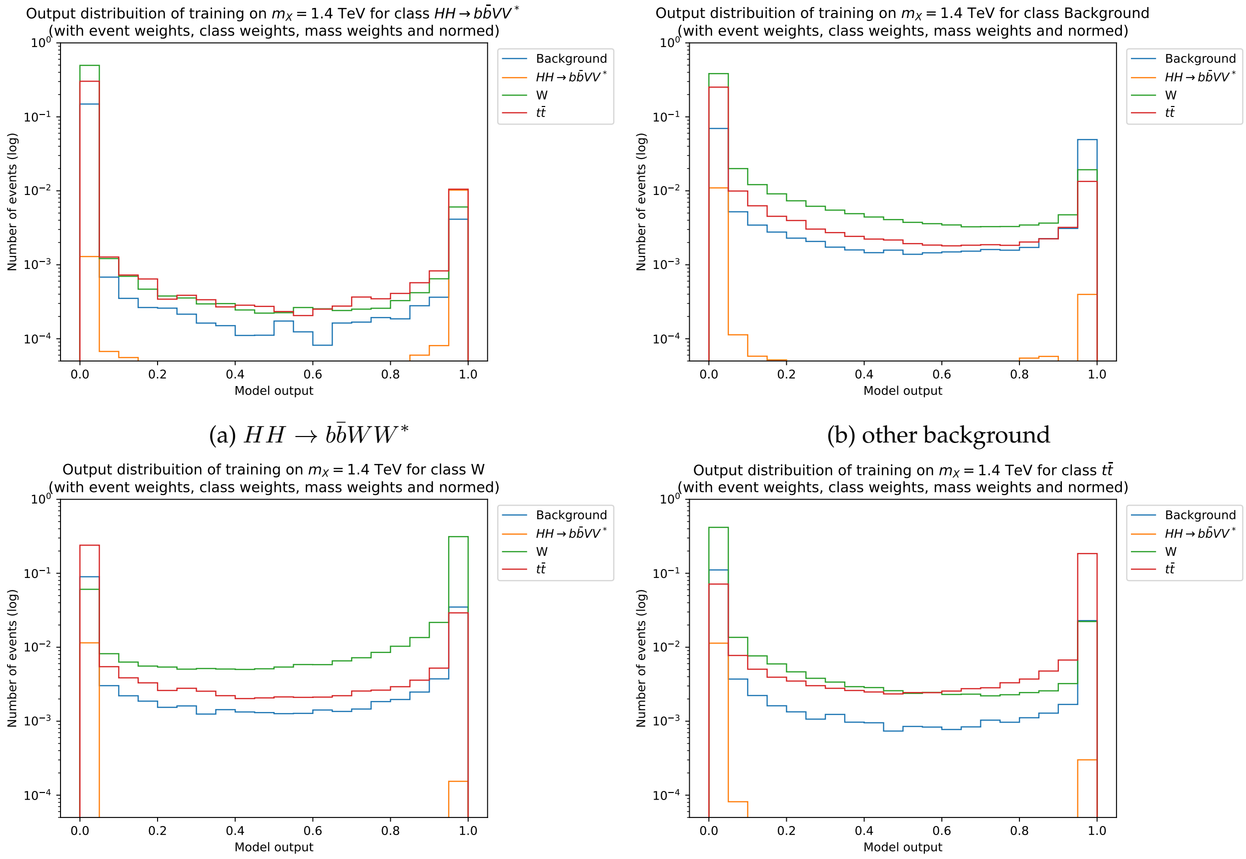
(b)

Figure 7.3: (a) Confusion matrix of the Neural Network trained on only the $m_X = 2$ TeV mass point. (b) Permutation importance of the different input variables for the Neural Network, trained only on $m_X = 2$ TeV. Classification is done by using the maximum predicted class probability.

Training on $m_X = 1.4$ TeV

Because different kinematics are expected for lower mass points $m_X < 2$ TeV, a second mass point at $m_X = 1.4$ TeV was chosen, which contains a significantly lower number of events. Comparing the output distributions for the Neural Network trained on $m_X = 2$ TeV (Figure 7.2) and the one trained on $m_X = 1.4$ TeV (Figure 7.4), the classes W, $t\bar{t}$ and other background perform almost identically. However, for the $HH \to b\bar{b}WW^*$ events, the model predicts not only the correct signal, but also some W events as $HH \to b\bar{b}WW^*$ events. Taking a look at the confusion matrix in Figure 7.5a, the overall trend from the distribution is also present.

Except for the W class, the model trained on $m_X = 1.4$ TeV is outperformed by the Neural Network trained on $m_X = 2$ TeV, although the $m_X = 1.4$ TeV model is only performing significantly worse for the $HH \to b\bar{b}WW^*$ class. Additionally, a small change in the input variables used is also notable, with the variables with the highest permutation importance being: $N^{\text{jets}}_{b-\text{tagged}}$, min $\Delta R(\text{lep}, \text{closest jet})$, $p_{\text{T}}^{W_{\text{had}}}$, and $m^{W_{\text{had}}}$ (see Figure 7.5b).



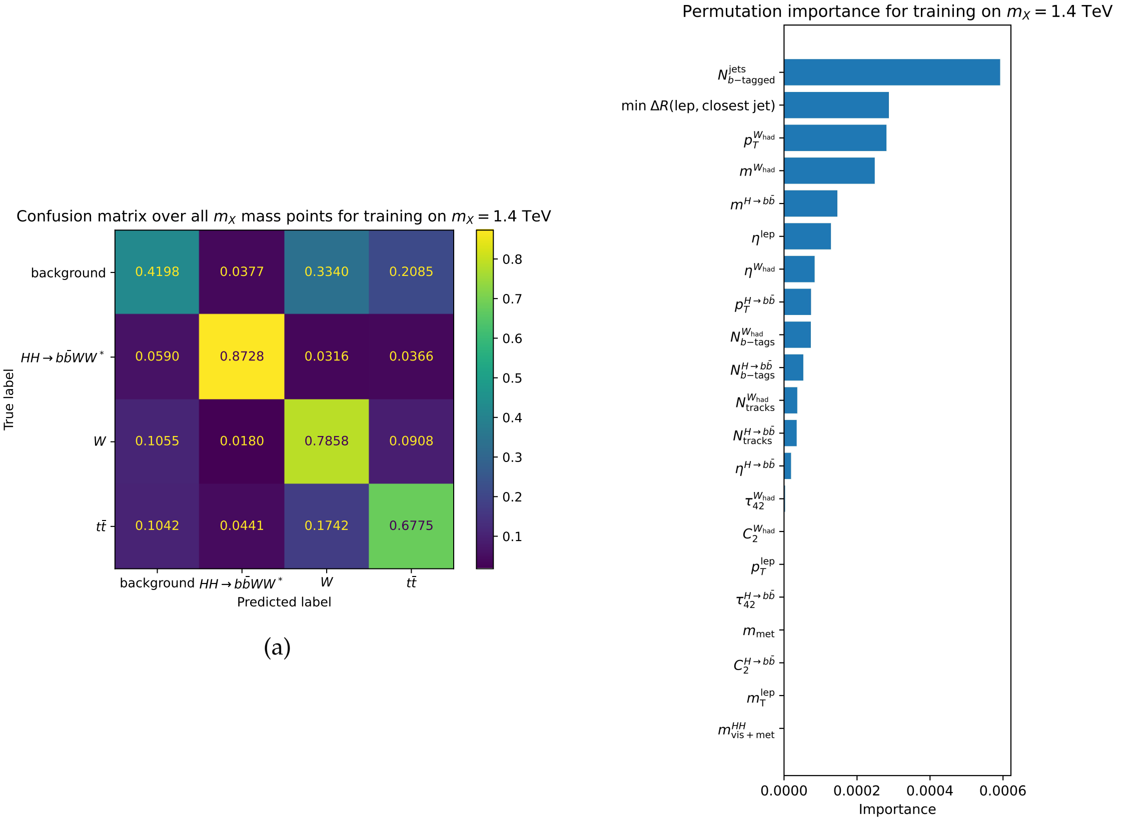
(a) $HH \to b\bar{b}WW^*$

(b) other background

(c) W

(d) $t\bar{t}$

Figure 7.4: Distribution of Neural Network output trained on only the $m_X = 1.4$ TeV mass point.



	background	$HH\to b\bar{b}WW^*$	W	$t\bar{t}$
background	0.4198	0.0377	0.3340	0.2085
$HH\to b\bar{b}WW^*$	0.0590	0.8728	0.0316	0.0366
W	0.1055	0.0180	0.7858	0.0908
$t\bar{t}$	0.1042	0.0441	0.1742	0.6775

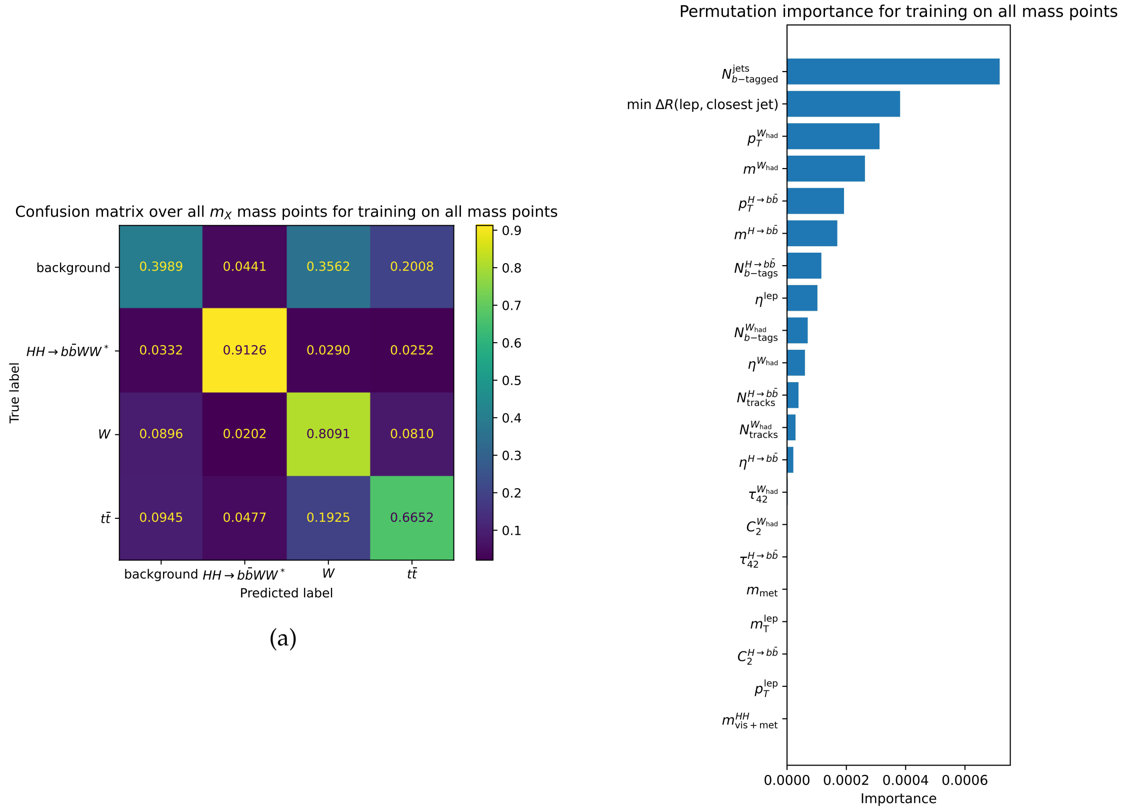
(a)

(b)

Figure 7.5: (a) Confusion matrix of the Neural Network trained on only the $m_X = 1.4$ TeV mass point. (b) Permutation importance of the different input variables for the Neural Network, trained only on $m_X = 1.4$ TeV. Classification is done, by using the maximum predicted class probability.

7.3 Multi Mass Point Neural Network

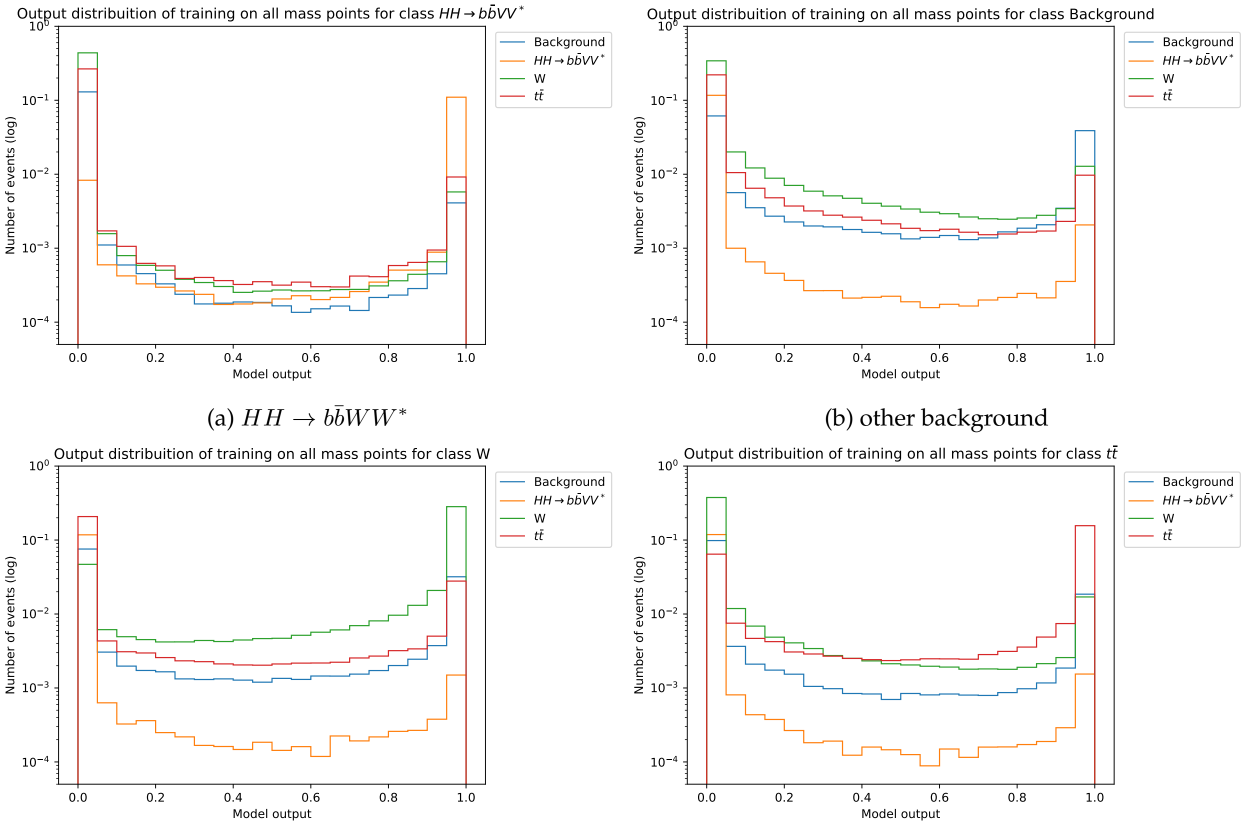
A final training was conducted on all mass points together with the mass point and class reweighting from Section 5.1.3 applied. As seen in Figure 7.6a, this model performs the best at predicting the $HH \to b\bar{b}WW^*$ and W class. On the other hand this model performs the worst for the classes $t\bar{t}$ and other background. When examining the output distributions (Figure 7.7) a similar behaviour as the $m_X = 2$ TeV can be seen, with the exception that an increase in $HH \to b\bar{b}WW^*$ event misclassification is visible in all classes. Most important to the model's decision are the following variables (see Figure 7.6b): $N_{b-\text{tagged}}^{\text{jets}}$, $\min\,\Delta R(\text{lep}, \text{closest jet})$, $p_{\text{T}}^{W_{\text{had}}}$, and $m^{W_{\text{had}}}$. These are the same variables as the $m_X = 1.4$ TeV training, but a change in importance can be seen.



(a)

(b)

Figure 7.6: (a) Confusion matrix of the Neural Network trained on all mass points. (b) Permutation importance of the different input variables for the Neural Network, trained only on all mass points. Classification is done by using the maximum predicted class probability.



(a) $HH \rightarrow b\bar{b}WW^*$

(b) other background

(c) W

(d) $t\bar{t}$

Figure 7.7: Distribution of Neural Network output trained on all mass points.

7.4 Comparison of the models

As discussed before the Neural Network trained on all mass points has the highest efficiency for $HH \to b\bar{b}WW^*$ and W events, at the cost of misclassifying other events. To compare the different models and cut-based approach more in depth, the efficiency for each mass point is of interest, which can be seen in Figure 7.8. The model trained on all mass points outperforms all other models, when all mass points are considered. However the singular mass point models achieve in some mass points an equal or even better efficiency. The singular mass point models peaks are around the mass point used in training: $m_X = 1.4$ and 2 TeV. For mass points with drastically different kinematics to the one trained on, the singular mass point models mostly cannot classify a signal event efficiently. Nonetheless, for mass points with similar kinematics the singular mass point models generalise quite well. As an example, the model trained on $m_X = 2$ TeV misclassifies almost all $HH \to b\bar{b}WW^*$ events with $m_X \leq 1.2$ TeV, but achieves the same efficiency as the model trained on all mass points at $m_X = 2$ and 5 TeV. The efficiency of the model trained on only $m_X = 1.4$ TeV for the low mass points $m_X = 0.8$ to 1.6 TeV is especially surprising. This behaviour indicates similar physical behaviour between the low mass points.

When only examining the model trained on all mass points, a steady increase in efficiency from $m_X = 0.8$ to 3 TeV can be observed. Afterwards the efficiency drops at $m_X = 4$ TeV and increases again at $m_X = 5$ TeV. Similarly the model trained on $m_X = 2$ TeV also has a lower efficiency for $m_X = 4$ TeV than for $m_X = 5$ TeV. Contradictory, the model trained on $m_X = 1.4$ TeV has a higher efficiency at $m_X = 4$ TeV, with this mass point generally being an outlier in the overall model performance. These observations imply a change in the separation power of the variables and therefore, a change in kinematics of the physics objects at $m_X = 4$ TeV. With the efficiency of the model trained on $m_X = 1.4$ TeV increasing at $m_X = 4$ TeV, the separation power or kinematics for $m_X = 4$ TeV have to be more similar to the once observed in lower mass points than neighbouring mass points.

Comparing the background efficiencies of the different models is difficult, because the Neural Network predicts three background classes and the cut-based model only one. As seen in Figure 7.8 and in the Sections above, the Neural Network's the efficiency for a single background class is not as good as the background efficiency of the cut-based approach. However, as most misclassified background classes are classified as another background class (see Figures 7.3a, 7.5a and 7.6a), combining the three classes into the complete background class, results in a significant increase in efficiency. Therefore, accuracy is also calculated twice for the Neural Networks: once with all 4 classes predicted, referred to as all classes accuracy and once with the background classes combined, referred to as binary accuracy. Both accuracies are listed in Table 7.1. As expected, the binary accuracy is the overall highest for the Neural Network trained on all mass points. However, when all four classes are considered, the accuracy is much lower.

The Receiver Operating Characteristic (ROC) curve shows the $HH \to b\bar{b}WW^*$ signal and the

model	binary accuracy	all classes accuracy
cut-based model	92.4%	–
trained on all mass points	96.03%	72.39%
trained on $m_X = 2$ TeV	94.39%	70.42%
trained on $m_X = 1.4$ TeV	90.29%	69.45%

Table 7.1: Accuracy of the different models. All statistics are calculated over all mass points.

complete background efficiency for a model at different cut thresholds on the model output. In addition, the Area Under the Curve (AUC) is presented, which is the integral of the ROC and for a perfect model would be 1. Figure 7.9 shows the ROC curves for the four analysis classes of the model trained on all mass points. As normal for ROC curves, the background efficiency decreases with increasing signal efficiency. However, for the background classes, the curves and AUC exhibit a much worse performance. In particular, the other background class approaches the performance of an offset random guesser (dotted line).

Figures 7.10, 7.11, 7.12 and 7.13 show the ROC curves for the three models. Based on these curves, the same observations as before can be made. In addition the Neural Network's signal or background efficiency can be compared to the cut-based model. As seen, at the same signal efficiency as the cut-based model, the model trained on all mass points only surpasses the background efficiency of the cut-based model starting from $m_X \geq 1.2$ TeV. Comparing the AUCs for the different models (see Table 7.2), once again the model trained on all mass points outperforms the singular mass point models.

	AUC			
model	$HH \to b\bar{b}WW^*$	W	$t\bar{t}$	other background
trained on all mass points	0.98	0.883	0.882	0.748
trained on $m_X = 2$ TeV $m_X = 2$ TeV	0.868	0.879	0.875	0.731
trained on $m_X = 1.4$ TeV	0.685	0.874	0.865	0.702

Table 7.2: AUC for the Neural Networks. The cut-based model is not included, as the statistics are not applicable. All statistics are calculated over all mass points.

All in all, the Neural Network trained on all mass points classifies signal events the best. Compared to the cut-based model a higher binary accuracy is achieved (see Table 7.1), with the benefit of relatively efficient classification of W and $t\bar{t}$ events.

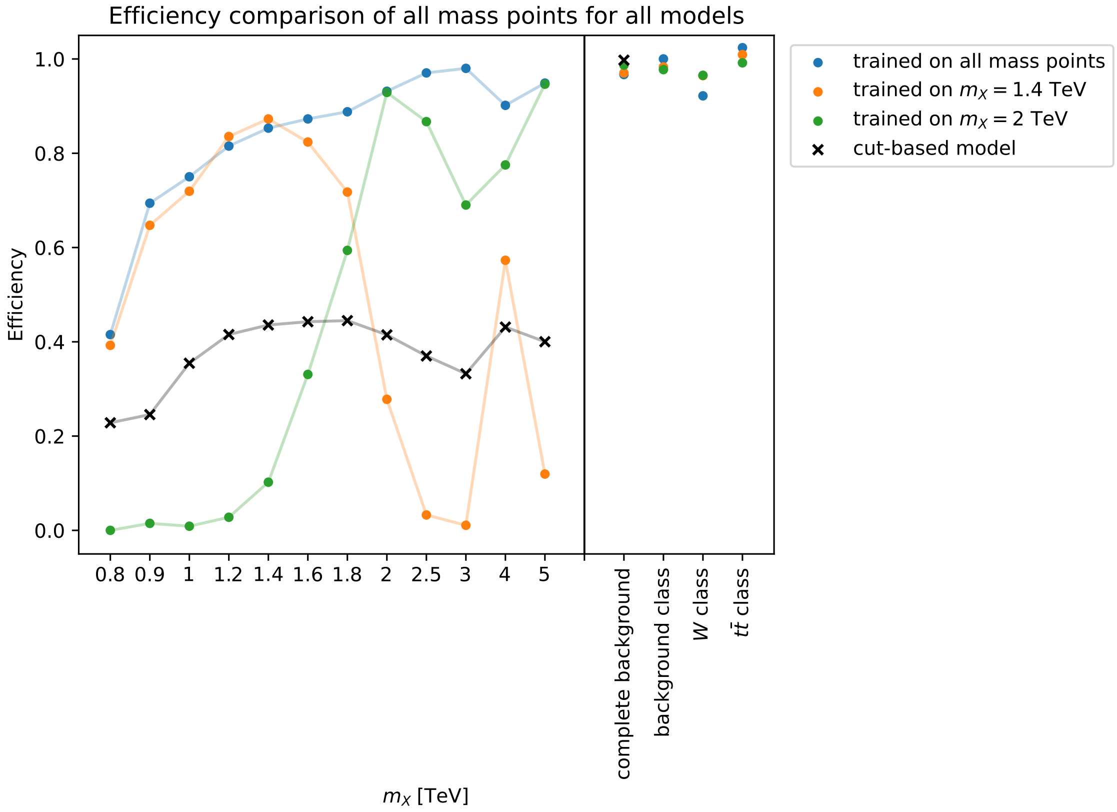


Figure 7.8: Efficiency comparison of all models for all mass points and background, W and $t\bar{t}$ classes. Classification is done, by using the maximum predicted class probability. Because the cut-based model only separates between signal and complete background, the complete background for the NN is defined as: events not classified as signal.

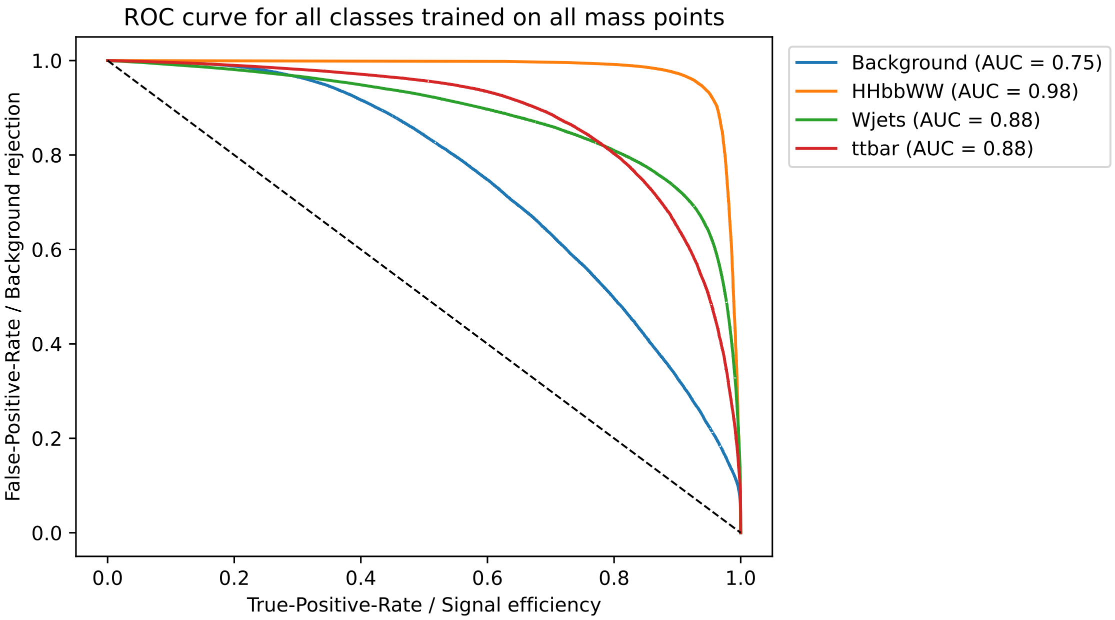


Figure 7.9: Receiver Operating Characteristic (ROC) curves of the multi mass point model for all classes.

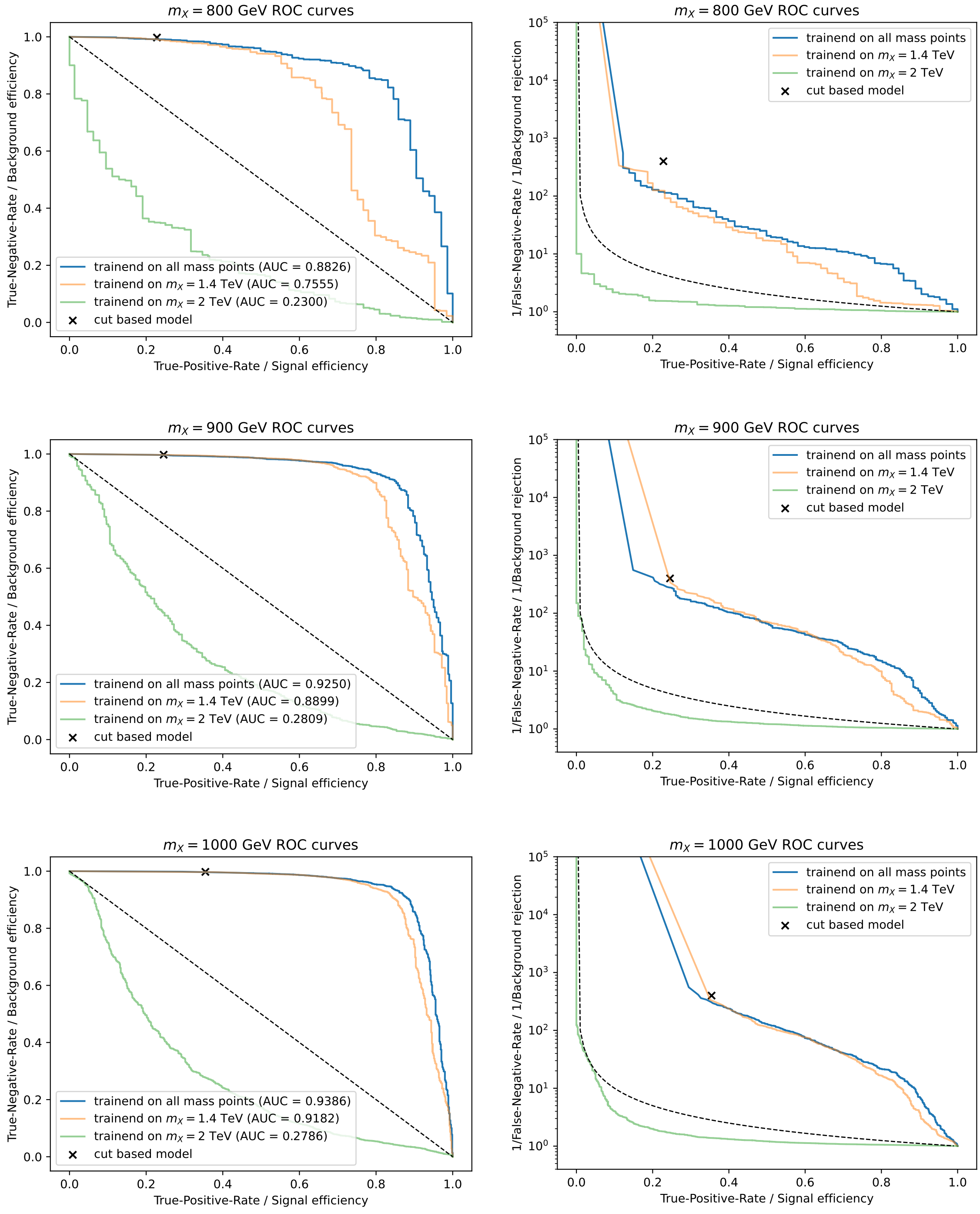


Figure 7.10: Receiver Operating Characteristic (ROC) curves for the different mass points and all models trained.

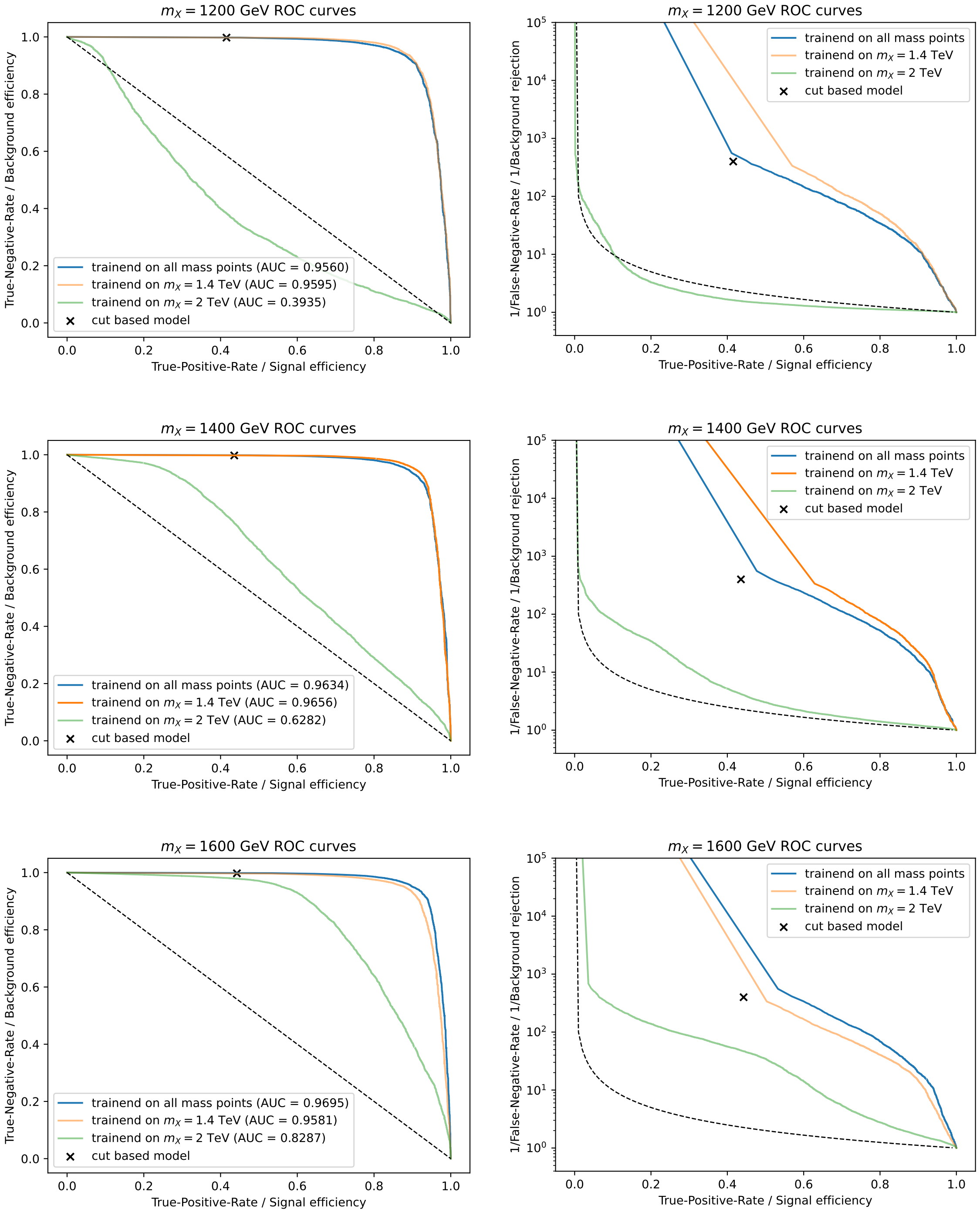


Figure 7.11: Receiver Operating Characteristic (ROC) curves for the different mass points and all models trained.

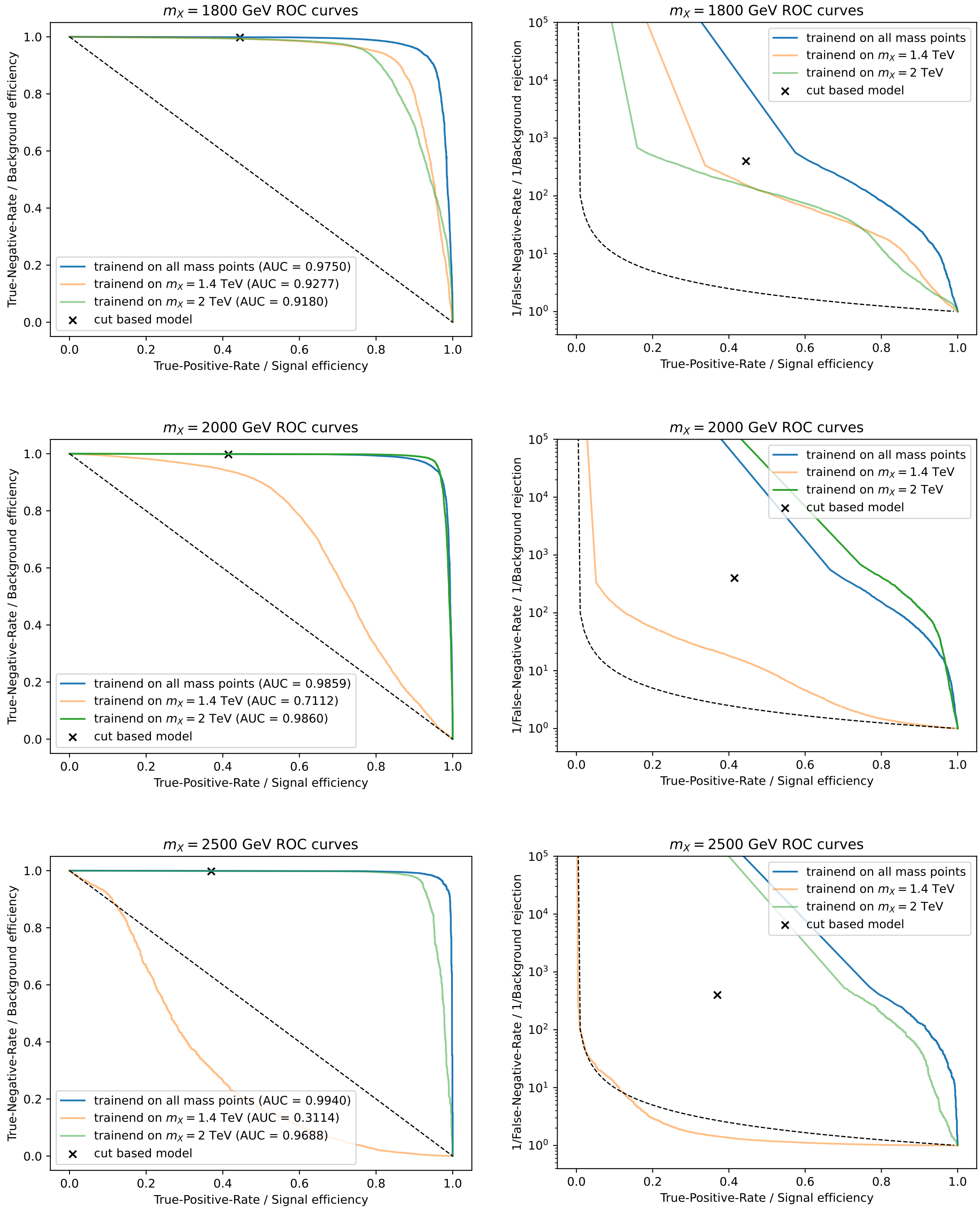


Figure 7.12: Receiver Operating Characteristic (ROC) curves for the different mass points and all models trained.

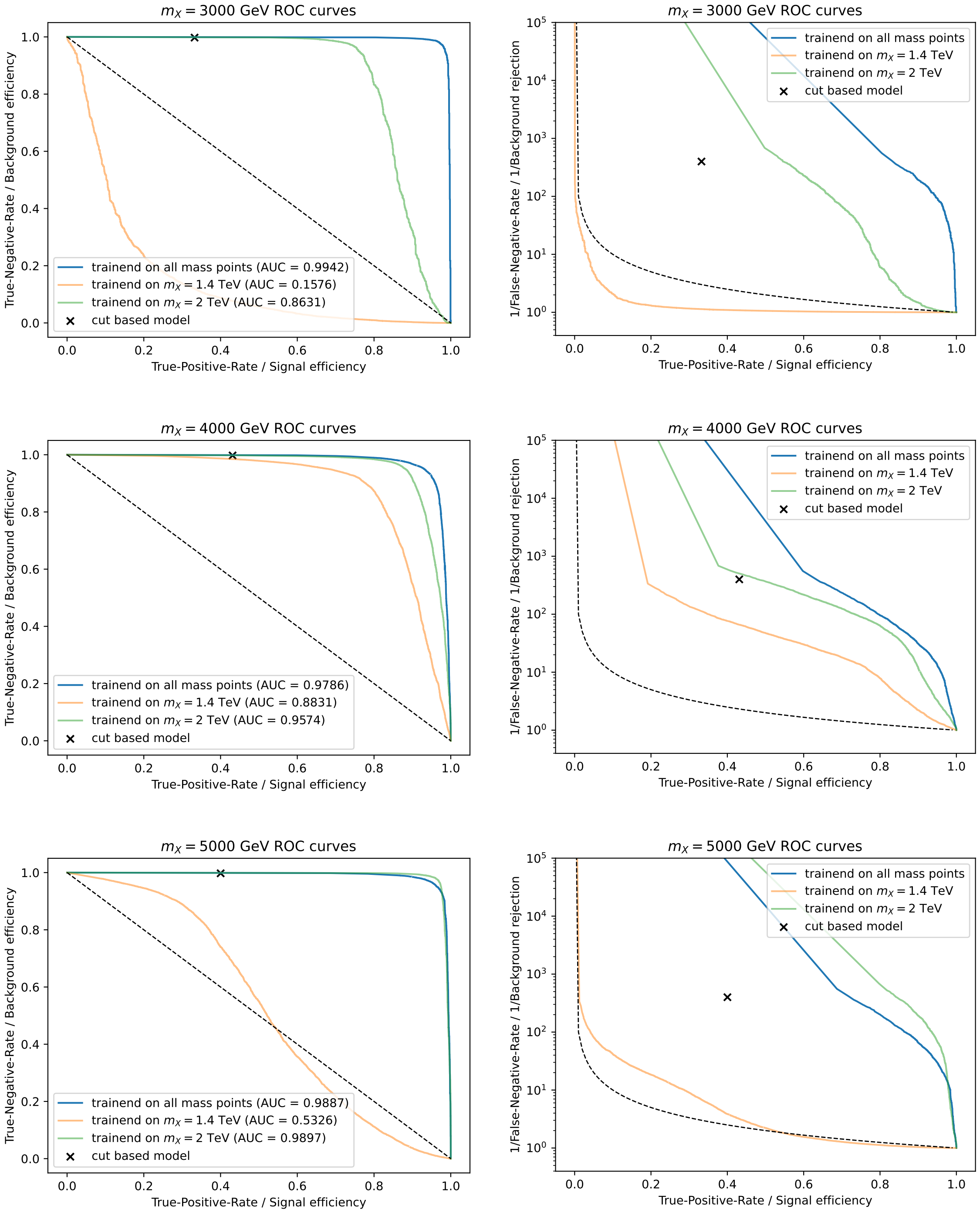


Figure 7.13: Receiver Operating Characteristic (ROC) curves for the different mass points and all models trained.

Chapter 8

Conclusion and Outlook

In Summary, a new model was proposed to separate $HH \rightarrow b\bar{b}WW^{*}$, W, $t\bar{t}$ and other background events in the boosted $X \rightarrow HH \rightarrow b\bar{b}VV^{*}$ analysis with 1 lepton in the final state. The model is based on a Feed Forward Neural Network and was trained on all mass points. Additionally, the model was trained on the mass points $m_X = 2$ TeV and $m_X = 1.4$ TeV, to evaluate changes in model performance.

In comparison, the new Neural Network model trained on all mass points outperformed the currently used cut-based model at classifying signal events on all mass points. However, a slight reduction in background efficiency is observed, but is negligible.

Comparing the Neural Networks at different mass points, a drastic change in signal efficiency is observable for the models trained on singular mass points. This can be attributed to a change in separation power of the input variables, due to a change in the underlying kinematics. Therefore, using multiple Neural Networks for different mass points is not expected to outperform the proposed method, as no significant performance increase between the Neural Network trained on all data and the Neural Network trained on singular mass points was observed. Furthermore, using a Parameterised Neural Network is also not expected to change the performance much, but would be expected to generalise more easily [68].

Concluding Machine and Deep Learning models can be of great benefit for the $X \rightarrow HH \rightarrow b\bar{b}WW^{*}$ analysis, as they could replace human designed algorithms with more efficient ones, as presented in this thesis. Additionally the better separation and classification of events should allow a new definition of regions, in which the $X \rightarrow HH \rightarrow b\bar{b}WW^{*}$ cross section can be better measured.

Bibliography

[1] "Standard model of elementary particles," https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg, accessed: 2022-09-23.

[2] J. Ellis, "Higgs Physics," *European School of High-Energy Physics*, 2013.

[3] "Standard model higgs boson decay branching ratios and total width," https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWGCrossSectionsFigures, 2016, accessed: 2022-09-23.

[4] K. Abeling, *Search for resonant Higgs boson pair production in the $b\bar{b}WW^*$ decay channel in the boosted 1-lepton final state using the full Run 2 ATLAS dataset*, April 2022, II.Physik-UniGö-Diss-2022/01.

[5] E. Mobs, "The CERN accelerator complex in 2019. Complexe des accélérateurs du CERN en 2019," 2019, accessed: 2023-01-22, General Photo. [Online]. Available: https://cds.cern.ch/record/2684277

[6] ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST*, vol. 3, 2008.

[7] F. Halzen and A. D. Martin, *Quarks and Leptones: An Introductory Course in Modern Particle Physics*. Wiley, 1984.

[8] M. E. Peskin, *An Introduction To Quantum Field Theory*. CRC Press, 1995.

[9] M. Thomson, *Modern Particle Physics*. Cambridge University Press, May 2013.

[10] ATLAS Collaboration, "Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 1–29, 2012.

[11] CMS Collaboration, "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012.

[12] ATLAS Collaboration, "Combination of searches for higgs boson pairs in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *Physics Letters B*, vol. 800, no. 3, p. 135103, 2020.

[13] CMS Collaboration, "Combination of searches for higgs boson pair production in proton-proton collisions at $\sqrt{s} = 13$ TeV," *Physics Review Letter*, vol. 122, p. 121803, 2019.

[14] D. Guest, K. Cranmer, and D. Whiteson, "Deep learning and its application to LHC physics," *Annual Review of Nuclear and Particle Science*, vol. 68, no. 1, pp. 161–181, 2018.

[15] K. Albertsson *et al.*, "Machine learning in high energy physics community white paper," *Journal of Physics: Conference Series*, vol. 1085, no. 2, p. 22008, 2018.

[16] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih, "Machine learning in the search for new fundamental physics," *Nature Reviews Physics*, vol. 4, no. 1, p. 399–412, 2021.

[17] R. P. Feynman, "Mathematical formulation of the quantum theory of electromagnetic interaction," *Phys. Rev.*, vol. 80, pp. 440–457, 1950.

[18] S. Weinberg, "A model of leptons," *The European Physical Journal C*, vol. 19, no. 21, pp. 1264–1266, 1967.

[19] A. Salam, "Weak and electromagnetic interactions," *Conf. Proc. C*, vol. 680519, pp. 367–377, 1968.

[20] S. L. Glashow and S. Weinberg, "Natural conservation laws for neutral currents," *Phys. Rev. D*, vol. 15, no. 7, pp. 1958–1965, 1977.

[21] G. Altarelli, "A QCD primer," in *AIP Conference Proceedings*, 2002.

[22] J. Goldstone, A. Salam, and S. Weinberg, "Broken symmetries," *Phys. Rev.*, vol. 127, no. 3, pp. 965–970, 1962.

[23] F. Englert and R. Brout, "Broken symmetry and the mass of gauge vector mesons," *Phys. Rev. Lett.*, vol. 13, no. 9, pp. 321–323, 1964.

[24] LHC Higgs Cross Section Working Group, "Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector," 2016.

[25] ATLAS Collaboration and CMS Collaboration, "Combined Measurement of the Higgs Boson Mass in pp Collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS Experiments," *Phys. Rev. Lett. 114(19)*, 2015.

[26] J. Baglio *et al.*, "$gg \to hh$: Combined uncertainties," *Phys. Rev. D*, vol. 103, p. 056002, 2021.

[27] T. D. Lee, "A theory of spontaneous t violation," *Phys. Rev. D*, vol. 8, no. 4, pp. 1226–1239, 1973.

[28] T. Robens, T. Stefaniak, and J. Wittbrodt, "Two-real-scalar-singlet extension of the SM: LHC phenomenology and benchmark scenarios," *The European Physical Journal C*, vol. 80, no. 2, p. 151, 2020.

[29] R. F., "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychol Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009.

[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.

[32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010.

[33] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[34] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 3, pp. 303–314, 1989.

[35] S. Park, C. Yun, J. Lee, and J. Shin, "Minimum width for universal approximation," *International Conference on Learning Representations*, 2021.

[36] B. Hanin and M. Sellke, "Approximating continuous functions by relu nets of minimal width," vol. 2, 2017.

[37] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 3, p. 6232–6240, 2017.

[38] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 237–285, 1996.

[39] P. Werbos and P. John, "Beyond regression : new tools for prediction and analysis in the behavioral sciences," *Journal of Artificial Intelligence Research*, 1974.

[40] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[41] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, 1964.

[42] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, 2013, pp. 1139–1147.

[43] S. Hanson and L. Pratt, "Comparing biases for minimal network construction with back-propagation," *Advances in Neural Information Processing Systems*, vol. 1, 1988.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, vol. 9, 2014.

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *International Conference on Learning Representations*, vol. 3, 2017.

[46] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *International Conference on Learning Representations*, 2018.

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, vol. 37, 2015, p. 448–456.

[48] S. Ioffe *et al.*, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, vol. 3, 2015.

[49] L. Evans and P. Bryant, "LHC machine," *Journal of Instrumentation*, vol. 3, no. 8, p. S08001, 2008.

[50] ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *Journal of Instrumentation*, vol. 3, no. 8, p. S08003, 2008.

[51] CMS Collaboration, "The CMS experiment at the CERN LHC," *Journal of Instrumentation*, vol. 3, no. 8, p. S08004, 2008.

[52] ALICE Collaboration, "The ALICE experiment at the CERN LHC," *Journal of Instrumentation*, vol. 3, no. 8, p. S08002, 2008.

[53] LHCb Collaboration, "The LHCb detector at the LHC," *Journal of Instrumentation*, vol. 3, no. 8, p. S08005, 2008.

[54] J. Alwall, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations," *Journal of High Energy Physics*, vol. 07, no. 79, 2014.

[55] R. D. Ball *et al.*, "Parton distributions with LHC data," *Nuclear Physics B*, vol. 867, no. 244, 2013.

[56] J. Bellm *et al.*, "Herwig 7.0/herwig++ 3.0 release note," *The European Physical Journal C*, vol. 76, no. 196, 2016.

[57] D. J. Lange *et al.*, "The evtgen particle decay simulation package," *Nuclear Instruments and Methods A*, vol. 462, no. 152, 2001.

[58] ATLAS Collaboration, "Performance of the Fast ATLAS Tracking Simulation (FATRAS) and the ATLAS Fast Calorimeter Simulation (FastCaloSim) with single particles," CERN, Geneva, Tech. Rep., 2014, all figures including auxiliary figures are available

at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2014-001. [Online]. Available: https://cds.cern.ch/record/1669341

[59] ATLAS Collaboration, "Track assisted techniques for jet substructure," CERN, Geneva, Tech. Rep., 2018. [Online]. Available: https://cds.cern.ch/record/2630864

[60] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *Journal of High Energy Physics*, vol. 2, 2008.

[61] ATLAS Collaboration, "Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ tev," *The European Physical Journal C*, no. 76, 2016.

[62] ATLAS Collaboration, "Muon reconstruction and identification efficiency in ATLAS using the full run 2 pp collision data set at $\sqrt{s} = 13$ TeV," *The European Physical Journal C*, vol. 81, no. 7, p. 578, 2021.

[63] ATLAS Collaboration, "Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run," 2017. [Online]. Available: https://cds.cern.ch/record/2273281

[64] A. J. Larkoski, G. P. Salam, and J. Thaler, "Energy correlation functions for jet substructure," *Journal of High Energy Physics*, vol. 3, 2013.

[65] J. Thaler and K. V. Tilburg, "Identifying boosted objects with n-subjettiness," *Journal of High Energy Physics*, vol. 3, 2011.

[66] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, 2019.

[67] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[68] P. Baldi *et al.*, "Parameterized neural networks for high-energy physics," *The European Physical Journal C*, vol. 76, no. 5, 2016.

.1 Distribution of used features and variables

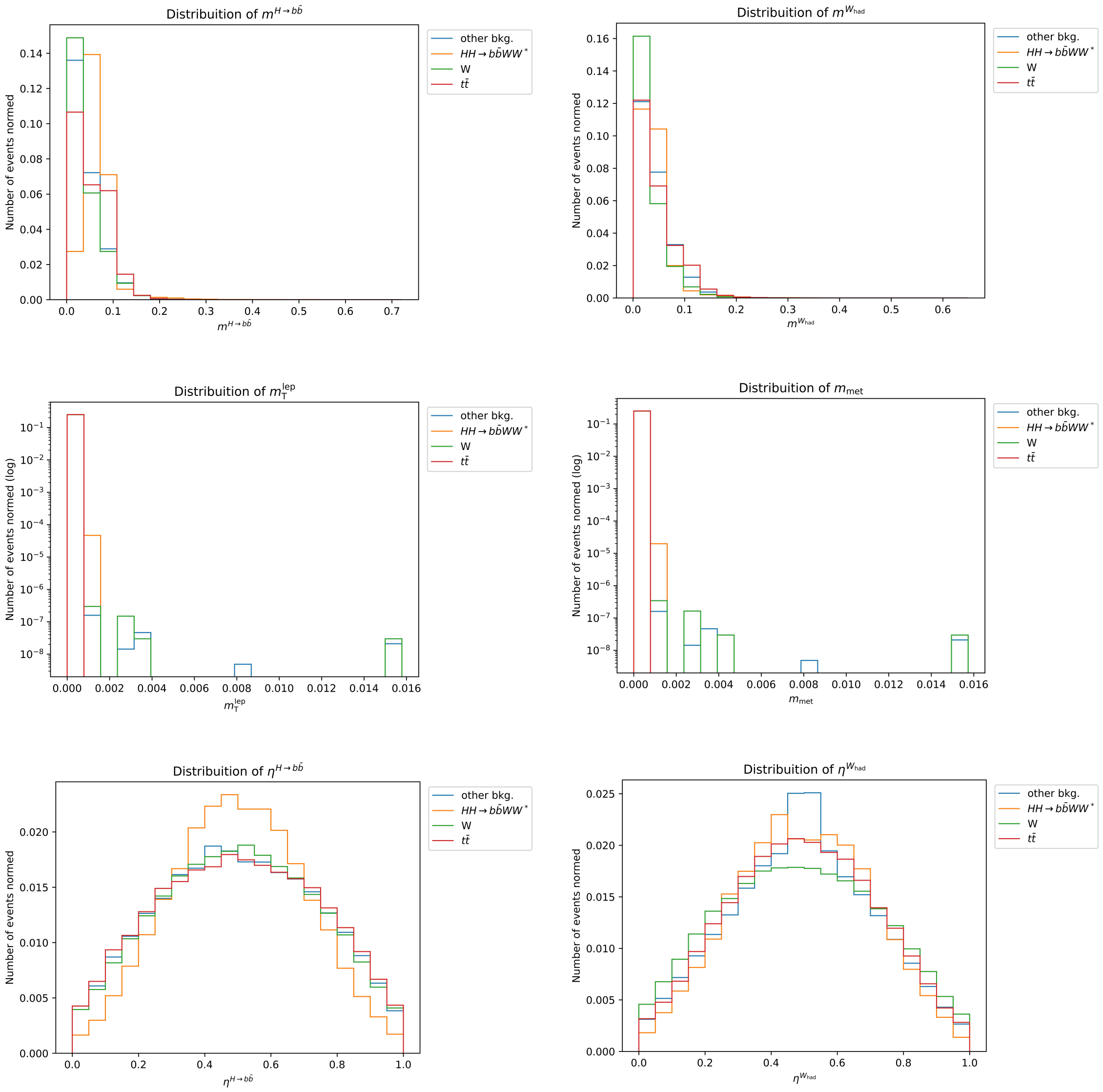


Figure 1: Normalised distribution of all low-level features with mass and class reweighting applied

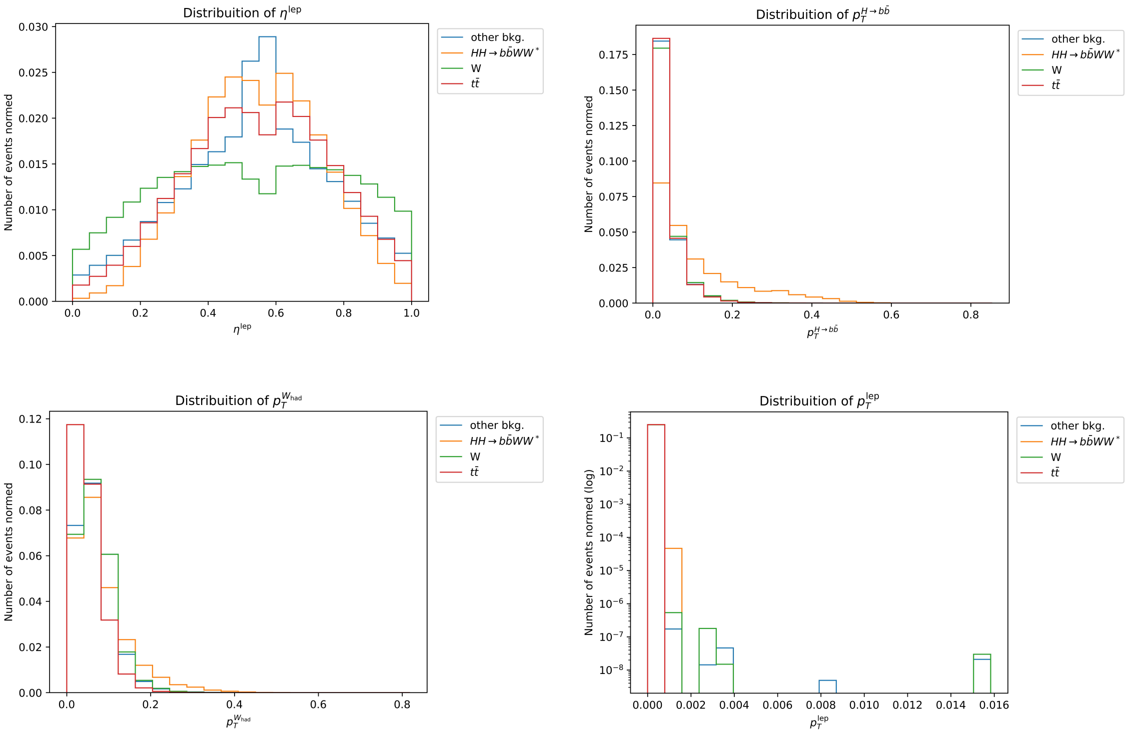


Figure 2: Continuation: Normalised distribution of all low-level features with mass and class reweighting applied

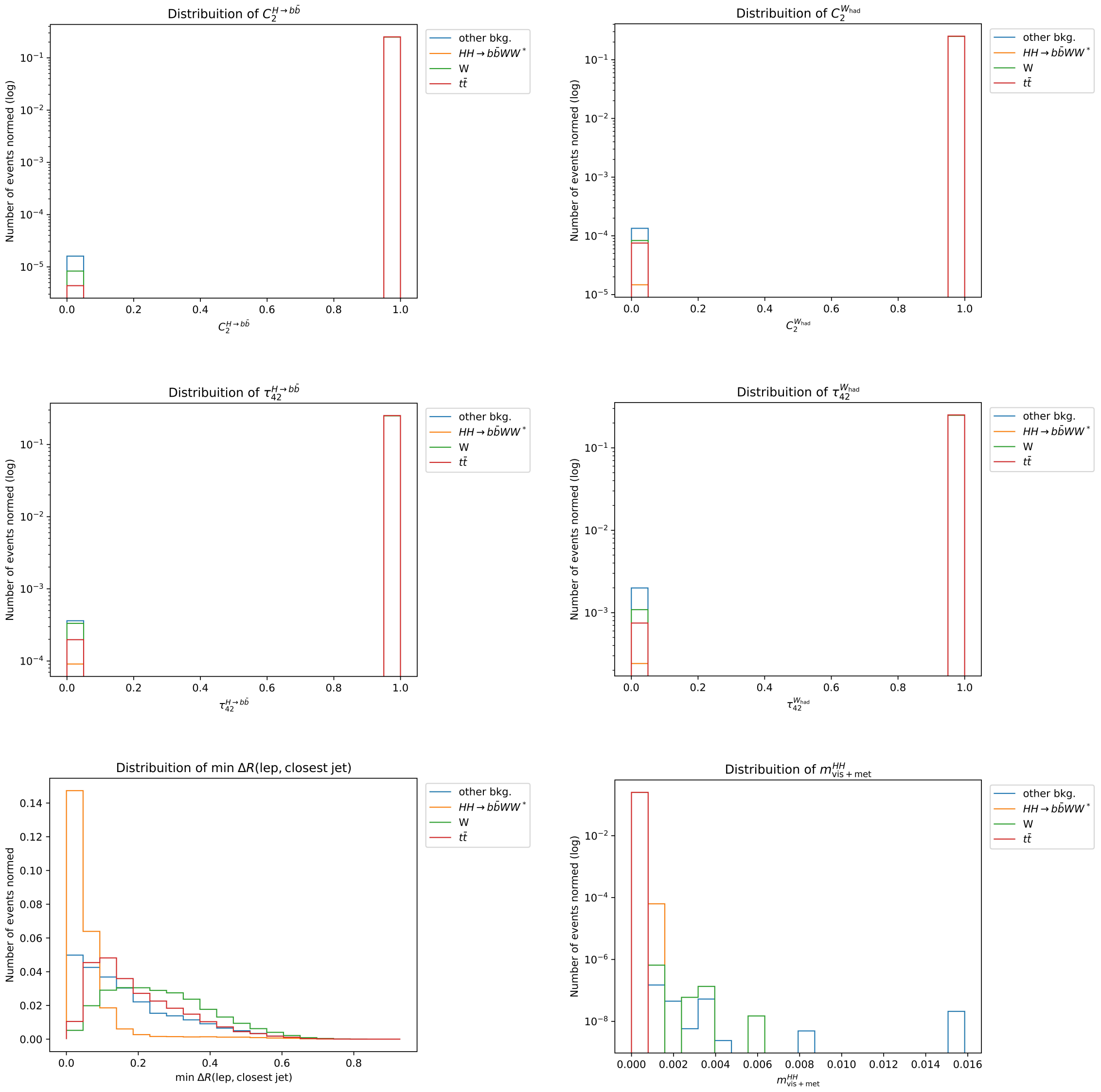


Figure 3: Normalised distribution of all high-level features with mass and class reweighting applied

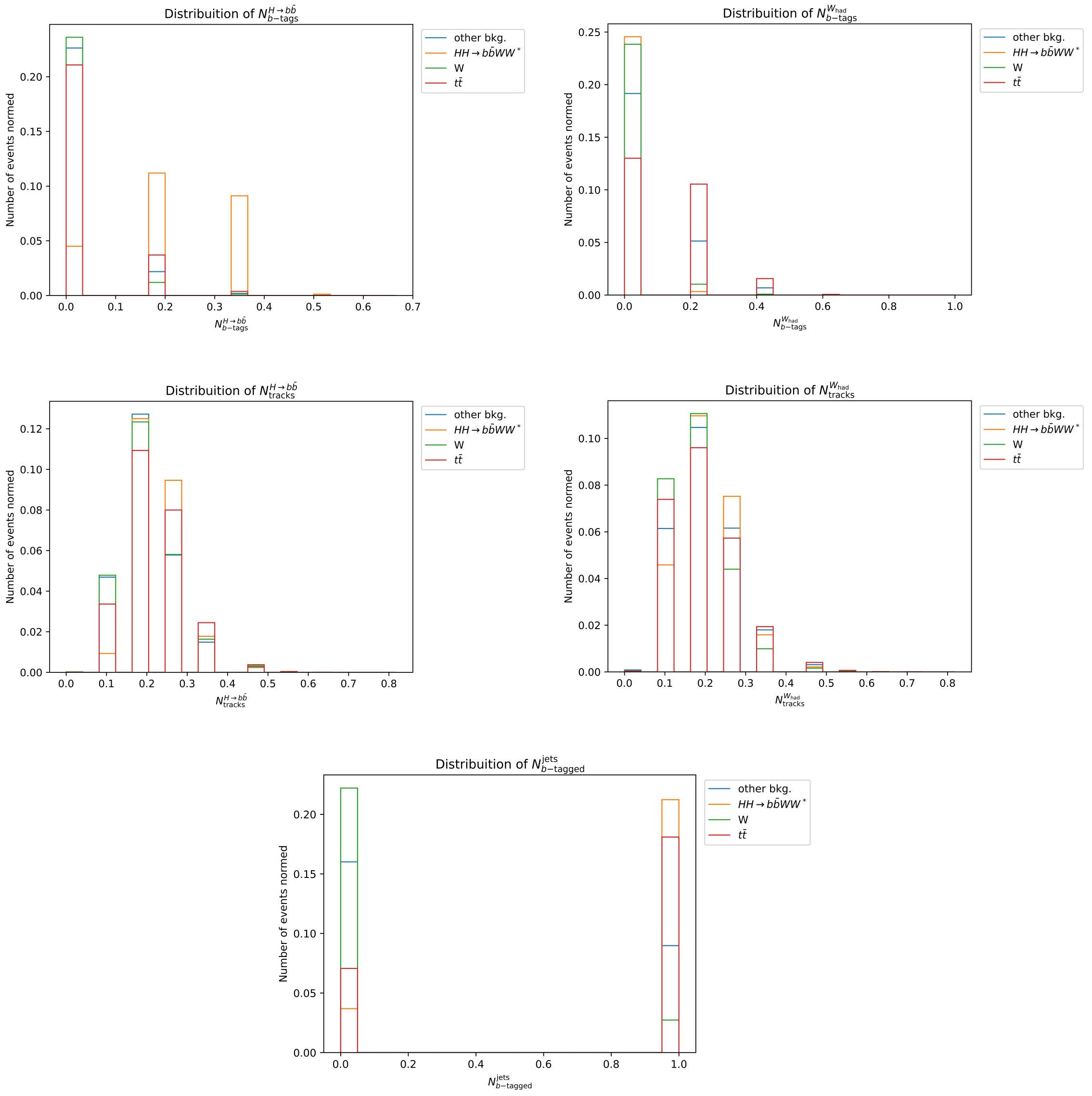


Figure 4: Normalised distribution of all meta variables with mass and class reweighting applied

Acknowledgements

I would like to take this opportunity, to thank you, Stan, for providing me the opportunity, to work on this thesis in such a nice working group and always having a suggestion on what to further analyse. I also owe Kira a big "Thank You!", for always helping me out and discussing any obscure idea. In general, I would like to thank the whole Göttingen particle physics and the ATLAS $HH \to b\bar{b}WW^*$ working group for the welcoming arms. Especially I want to thank Zobi, Leon and Joshua, for taking the time to listen to my weekly reports. Lastly, I want to thank Prof. Dr. Arnulf Quadt for his time as second referee.