ETP-Bachelor-KA/2021-06

Karlsruher Institut für Technologie

Multiklassifikation mit Bayesian Neural Networks in der $t\bar{t}H(b\bar{b})$-Analyse

Multiclassification with Bayesian neural networks in the $t\bar{t}H(b\bar{b})$-analysis

Bachelorarbeit

Dorian Guthmann

An der Fakultät für Physik
Institut für Experimentelle Teilchenphysik

Erstgutachter:	Prof. Dr. Ulrich Husemann
Zweitgutachter:	Priv.-Doz. Dr. Roger Wolf
Betreuender Mitarbeiter:	Nikita Shadskiy

Karlsruhe, 22. April 2021

Diese Arbeit wurde vom Erstgutachter der Bachelorarbeit akzeptiert.

Ort, Datum

.....

(**Prof. Dr. Ulrich Husemann**)

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Ort, Datum

. . . . . .

(**Dorian Guthmann**)

Inhaltsverzeichnis

1 Einleitung

Schon seit tausenden von Jahren fragen sich die Menschen, wie die Welt aufgebaut ist, in der sie leben. Bereits in der Antike, ca. 400 v. Chr., wurden vom griechischen Philosophen Demokrit kleinste unteilbare Teilchen eingeführt (atomos), aus denen die Welt aufgebaut sei [1]. In der modernen Physik beschreibt das Standardmodell der Elementarteilchenphysik die kleinsten bekannten Teilchen, aus denen die Welt aufgebaut ist. Für die experimentelle Erforschung des Standardmodells wurden riesige Teilchenbeschleuniger, wie der Large Hadron Collider (LHC) gebaut, mit denen nach und nach die heutzutage bekannten Teilchen des Standardmodells entdeckt wurden. Das zuletzt gefundene Teilchen des Standardmodells wurde 2012 durch die Experimente CMS [2] und ATLAS [3] am LHC entdeckt . Bei diesem Teilchen handelt es sich um das Higgs Boson. Die Untersuchung desselben ist Gegenstand aktueller Forschung der experimentellen Teilchenphysik.
Eine große Herausforderung für die Experimente, die Daten über das Higgs-Boson sammeln, ist die extrem kurze Lebensdauer des Teilchens. Aufgrund dieser kann das Teilchen nicht direkt betrachtet werden, sondern es sind nur Zerfallsprodukte beobachtbar. Mittels Messung, bzw. Rekonstruktion der Zerfallsprodukte können dann Informationen über das Higgs-Boson gewonnen werden. Bei der Rekonstruktion stellt sich das Problem, dass es viele verschiedene teilchenphysikalische Prozesse gibt, die ähnliche Endzustände haben und wesentlich häufiger vorkommen als Ereignisse in denen ein Higgs-Boson involviert ist [4]. Konkret wird in dieser Arbeit die mit $t\bar{t}H$ bezeichnete Produktion eines Higgs-Bosons mit einem Top-Quark-Paar im Rahmen des CMS-Experiments betrachtet [5]. Um eine gute Trennung zum Untergrund zu erreichen werden in der Analyse künstliche neuronale Netze eingesetzt.
Neuronale Netze sind künstliche Strukturen, die Informationsverarbeitung ermöglichen, welche der im menschlichen Gehirn ähnelt. Sie lernen durch Training an Datensätzen, wie die Prozesse zu unterscheiden sind und sollen dann in der Lage sein eine möglichst genaue Vorhersage zu machen, um welchen teilchenphysikalischen Prozess es sich bei Messdaten aus den Detektoren handelt.
Im Allgemeinen liefert ein neuronales Netz für beliebige Eingabedaten immer eine Vorhersage, ohne eine Angabe zu machen, wie sicher diese Vorhersage ist. Eine besondere Art der neuronalen Netze heißen bayesische neuronale Netze (BNN). Der Hauptunterschied zwischen herkömmlichen neuronalen Netzen und BNNs ist, dass die Informationsverarbeitung in BNNs durch Rechnung mit Wahrscheinlichkeitsverteilungen geschieht, mithilfe derer eine Aussage über die Unsicherheit der vom Netz gemachten Vorhersagen getroffenen werden kann. Im Rahmen der Masterarbeit [6] wurden für die $t\bar{t}H$-Analyse BNNs studiert, die

Vorhersagen darüber treffen können, ob ein gemessenes Ereignis zum $t\bar{t}H$-Prozess gehört, oder nicht.
Ziel dieser Arbeit ist es für die $t\bar{t}H$-Analyse BNNs einzuführen, die zwischen verschiedenen Klassen von Prozessen unterscheiden können, sowie eine Aussage über deren Nutzbarkeit und Leistungsfähigkeit im Vergleich zu herkömmlichen neuronalen Netzen zu treffen. Weiterführend sollen Auswertungsmethoden, welche die für BNNs einzigartige Vorhersagen-Unsicherheit miteinbeziehen, entwickelt werden. Im ersten Kapitel werden zunächst die physikalischen Grundlagen und die experimentelle Umgebung vorgestellt, die zum Verständnis der zu unterscheidenden physikalischen Prozesse notwendig sind. Kapitel 3 enthält eine Beschreibung der Funktionsweise von künstlichen neuronalen Netzen und BNNs, die später verwendet werden. Die zu klassifizierenden Daten und die Parameter der verwendeten neuronalen Netze, sowie die bisherigen, in der Analyse genutzten, Auswertungsmethoden für neuronale Netze werden in Kapitel 4 beschrieben. Darauf folgend wird in Kapitel 5 die Verlässlichkeit von BNNs für die Analyse überprüft. Ein Vergleich von herkömmlichen neuronalen Netzen und bayesischen neuronalen Netzen erfolgt in Abschnitt 5.2. In Kapitel 6 wird eine Möglichkeit die Unsicherheit der Ausgabe von BNNs zu nutzen vorgestellt.

2 Teilchenphysikalische Grundlagen

Dieses Kapitel behandelt die Physik und die Experimente, denen die in dieser Arbeit verwendeten Daten zu Grunde liegen. Im Folgenden wird in Abschnitt 2.1 auf das Standardmodell der Teilchenphysik eingegangen. Es stellt die fundamentale Theorie der Teilchen dar, die in dieser Arbeit behandelt werden. Darauf folgend wird in Abschnitt 2.2 kurz auf den Large Hadron Collider (LHC) und in Abschnitt 2.3 auf den an diesem befindlichen Compact-Myon-Solenoid (CMS)-Detektor eingegangen. Die in dieser Arbeit verwendeten Daten entstammen Simulationen der CMS-Kollaboration. Diese Simulationen bilden Detektordaten des CMS-Experiments am größten Teilchenbeschleuniger der Welt, dem LHC, nach. Die Experimente die dort stattfinden, sollen das Standardmodell weiter überprüfen oder zu dessen Erweiterung dienen. In Abschnitt 2.4 werden die teilchenphysikalischen Ereignisse, die für diese Arbeit relevant sind, also der sogenannte $t\bar{t}H(b\bar{b})$-Prozess, der für die Erforschung des Higgs-Bosons wichtig ist, sowie seine Untergründe, genauer erläutert.

2.1 Das Standardmodell

Das Standardmodell der Teilchenphysik beschreibt die bekannten Elementarteilchen des Universums sowie die Wechselwirkungen zwischen ihnen, mit Ausnahme der Gravitation. Die Elementarteilchen des Standardmodells sind in Bosonen und Fermionen aufgeteilt, wobei Fermionen die Teilchen mit halbzahligem Spin sind und Bosonen ganzzahligen Spin aufweisen. Die nachfolgenden Ausführungen dieses Kapitels stützen sich, falls nichts anderes angegeben wird, auf [7] und [8].

Die Fermionen des Standardmodells lassen sich in zwei Untergruppen einteilen, zum einen die Leptonen und zum anderen die Quarks.
Die Gruppe der Leptonen besteht aus sechs Teilchen, die alle Spin 1/2 in Einheiten von $\hbar$ haben und in drei verschiedene Leptonen-Familien $L_{e,\mu,\tau}$ eingeteilt werden. Zu jeder Familie gehört hierbei ein Paar von Leptonen, mit einem (schwereren) elektrisch geladenen Teilchen und einem leichteren elektrisch neutralen Neutrino. Die Partner der Neutrinos sind alle einfach negativ geladen. Zu den Leptonen gehören Elektron e, Elektron-Neutrino ν_e, Myon μ, Myon-Neutrino ν_μ, Tauon τ und Tau-Neutrino ν_τ.
Die andere Untergruppe der Fermionen bilden die Quarks. Eine einzigartige Eigenschaft der Quarks ist, dass sie Drittelladungen in Einheiten der Elementarladung aufweisen und im Vergleich zu den Leptonen noch einen weiteren Freiheitsgrad haben, welcher

Farbladung genannt wird. Die möglichen Farbladungen der Quarks sind rot, blau und grün, beziehungsweise anti-rot, anti-blau und anti-grün bei Antiquarks. Alle Quarks haben ebenso wie die Leptonen einen Spin von 1/2. Die verschiedenen Quarks besitzen die Flavors Up u, Down d, Charm c, Strange s, Bottom b und Top t. Von diesen sechs haben u, c und t die elektrische Ladung $+2/3$ und d, s und b $-1/3$. Quarks können aufgrund des sogenannten Confinements nur in gebundenen Zuständen mehrerer farbgeladener Teilchen, die Hadronen genannt werden, existieren. Ein Hadron muss immer neutrale Farbladung besitzen.
Werden bei einer hochenergetischen Kollision von Hadronen, z.B. Protonen (uud), gebundene Quarks auseinandergerissen, dann nimmt mit wachsender Entfernung die Bindungsenergie solange weiter zu, bis diese ausreicht um ein neues Quark-Antiquark-Paar zu bilden. Das geschieht mehrfach, bis die Energie aufgebraucht ist. Die Bewegungsrichtung der neuen Quarks ist durch die des ursprünglichen gegeben. Ein so entstehendes Bündel von Teilchen, welche sich in eine Richtung bewegen, wird Jet genannt.
Die Fermionen lassen sich nach ihrer Masse in drei Familien (bzw. Generationen) einteilen. Diese Einteilung ist in Abbildung 2.1 zu sehen. So stellen z.B. u, d, e und ν_e die erste Familie dar. Aus ihr besteht die gesamte bekannte sichtbare Materie. Teilchen der zweiten und dritten Familien werden beispielsweise durch Streuexperimente hoher Energie erzeugt. Sie sind allerdings mit Ausnahme der Neutrinos kurzlebig und zerfallen in leichtere Teilchen. Zu jedem der zwölf beschriebenen Fermionen gibt es ein entsprechendes Antiteilchen, welches umgekehrte elektrische Ladung hat. Somit unterscheidet das Standardmodell insgesamt 24 verschiedene Fermionen.

Die Bosonen des Standardmodells lassen sich ebenfalls in zwei Gruppen unterteilen, die Eichbosonen (Vektorbosonen) und die Skalarbosonen. Eichbosonen dienen der Beschreibung der Wechselwirkungen zwischen den Elementarteilchen. So gibt es im Standardmodell zu jeder fundamentalen Wechselwirkung mindestens ein Eichboson.
Die elektromagnetische Wechselwirkung wird durch das Photon γ vermittelt. Photonen koppeln an alle Teilchen, die eine elektrische Ladung haben. Das Photon ist ladungs- und masselos und hat wie die anderen Eichbosonen einen Spin von 1.
Die schwache Wechselwirkung wird durch die W^+- und W^--Bosonen, sowie durch das Z-Boson übertragen. Bei elementaren (elektrisch) ladungsändernden Prozessen wird eins der $W^\pm$-Bosonen ausgetauscht und bei ladungserhaltenden Prozessen wird ein Z-Boson ausgetauscht. $W^\pm$-Bosonen und Z-Bosonen können an Quarks und an Leptonen koppeln. Die Theorie der elektroschwachen Wechselwirkung vereinheitlicht die Wechselwirkung durch $W^\pm$-, Z- und γ-Teilchen.
Für die starke Wechselwirkung sind die Gluonen g zuständig. Sie verfügen selbst über Farb-Antifarb-Paarladungen. Um alle Farbladungsprozesse des Standardmodells zu erklären sind acht verschiedene Gluonen notwendig. Gluonen koppeln an alle Teilchen mit Farbladung. Das bedeutet, dass sie sowohl an Quarks als auch an sich selbst koppeln. Diese Selbstkopplung führt zum Confinement.
Damit bilden Photon, die beiden W-Bosonen, das Z-Boson und die acht Gluonen die zwölf Eichbosonen, die im Standardmodell zu finden sind.
Die dem Standardmodell zugrundeliegende Eichtheorie fordert, dass die Eichbosonen masselos sind. Für Photonen und Gluonen stimmt diese Aussage, allerdings besitzen die W- und Z-Bosonen der schwachen Wechselwirkung endliche Massen. Um diesen Widerspruch zu lösen wurde 1964 u.a. von Peter Higgs der Higgs-Mechanismus eingeführt (welcher erst später nach ihm benannt wurde), der ein skalares Higgs-Feld und ein zu diesem gehöriges Teilchen H postuliert, das an die Masse der Elementarteilchen koppelt [10, 11].
Im Jahr 2012 wurde das Higgs Boson am LHC in den Experimenten CMS und ATLAS nachgewiesen. Es ist das einzige Skalarboson im Standardmodell und hat einen Spin von 0.

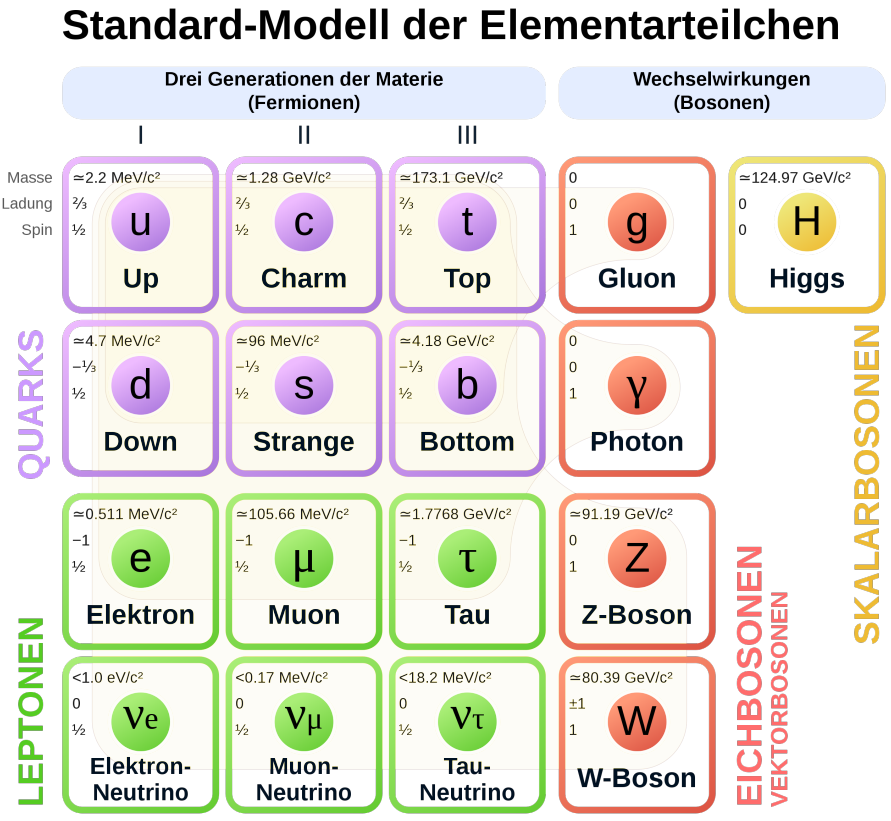

Abbildung 2.1: **Überblick über das Standardmodell** (Bildquelle: [9]) Zu sehen sind alle Teilchen des Standardmodells mit ihren jeweiligen Massen, elektrischen Ladungen und Spins. Auf der linken Seite sind die Fermionen zu sehen. Die Quarks sind in Violett dargestellt, die Leptonen darunter sind grün gekennzeichnet. Die drei Spalten stellen die im Text erwähnten drei Generationen der Fermionen dar. Auf der rechten Seite finden sich die Bosonen wieder. Dabei sind die Eichbosonen rot markiert und das Higgs-Boson gelb.

Mit seiner hohen Masse von ca. 125 GeV/c^2 ist es das zweitschwerste Elementarteilchen im Standardmodell nach dem Top-Quark [12].

2.2 Large Hadron Collider

Die nachfolgenden Beschreibungen basieren auf [13] und [14]. Der Large Hadron Collider (kurz LHC) am Conseil européen pour la recherche nucléaire (CERN) bei Genf in der Schweiz ist der größte Teilchenbeschleuniger der Welt. Es handelt sich um einen Synchrotron (Ringbeschleuniger) mit etwa 26,65 km Umfang, welcher Kollisionsenergien von bis zu 13 TeV erreicht. Die Teilchen, welche zum Zusammenstoß gebracht werden, sind entweder Protonen oder schwere Ionen, wie z.B. von Blei. Am LHC gibt es vier Stellen, an denen die Teilchenkollisionen durch große Experimente bzw. Detektoren beobachtet werden. Die vier Detektoren heißen A Large Ion Collider Experiment (ALICE), A Toroidal LHC ApparatuS (ATLAS), the Compact Muon Solenoid (CMS) und the Large Hadron Collider beauty (LHCb). Eine schematische Darstellung des LHCs mit den wichtigsten Interaktionspunkten ist in Abbildung 2.2 zu sehen. Die in den Detektoren gemessenen Signaturen der hochenergetischen Kollisionen lassen Rückschlüsse auf die physikalischen Prozesse zu. Ziel des LHCs

ist die experimentelle Untersuchung des Standardmodells, beziehungsweise Hinweise auf mögliche Erweiterungen des selbigen zu finden.

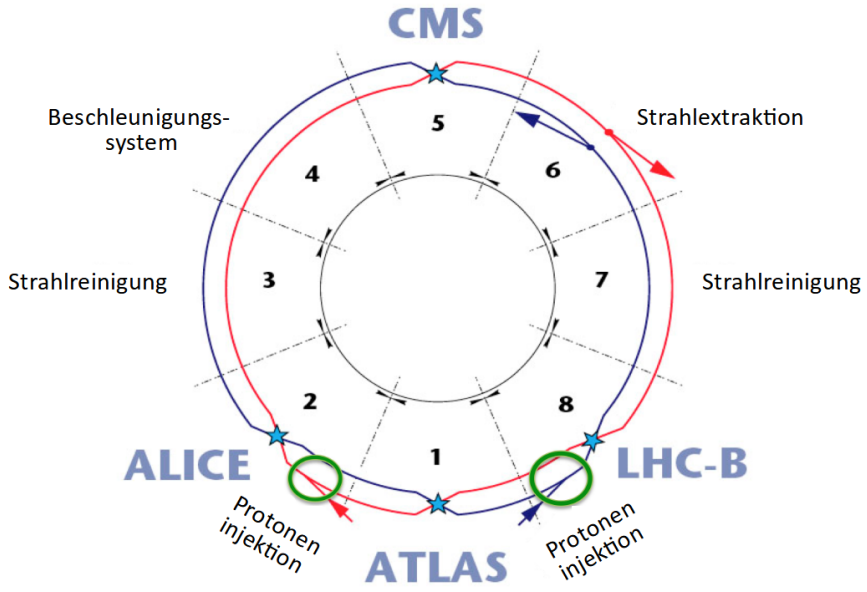

Abbildung 2.2: **Schematische Darstellung des Large Hadron Colliders** (Bildquelle: [15]) Der LHC ist in acht Bereiche unterteilt. Im Schaubild sind die Standorte der vier großen Detektoren zu sehen (blaue Sternchen). Rechts und links von ATLAS in Bereich 2 und 8 werden jeweils die Protonen in den Beschleuniger geleitet (grüne Kreise). Der für diese Arbeit relevante CMS-Detektor liegt in Bereich 5.

2.3 Der CMS-Detektor

Der Compact Muon Solenoid hat eine zylindrische Form, ist 15 Meter hoch und 21 Meter lang. Er besteht aus mehreren Schichten und kann vor allem Myonen besonders akkurat detektieren. Der Detektor umschließt die Kollisionsstelle der Protonenstrahlen, sodass die gestreuten Teilchen im Detektor landen. Mit ihm können Energie, Ort und Impuls der gestreuten Teilchen gemessen werden. Der nachfolgende Überblick beruht auf [16] und [17].

2.3.1 Aufbau

Das Herzstück des Detektors ist die supraleitende Magnetspule, welche eine magnetische Flussdichte von bis zu vier Tesla erzeugen kann. Der Zweck des Magnetfeldes ist es, geladene Teilchen innerhalb des Detektors abzulenken. Da negative und positive Ladungen im Magnetfeld in entgegengesetzte Richtungen abgelenkt werden, kann durch Bestimmung der Krümmungsrichtung der Trajektorien das Ladungsvorzeichen der gestreuten Teilchen bestimmt werden. Außerdem kann der Impuls von Teilchen präzise bestimmt werden. Je größer der Krümmungsradius der Teilchenbahn, desto kleiner ist der Impuls des beobachteten Teilchens.

Ein schematischer Aufbau des Detektors ist in Abbildung 2.3 zu sehen. Die innerste Detektorschicht ist der Spurdetektor, welcher selbst aus mehreren Schichten aus Siliziumhalbleitern besteht. Geladene Teilchen, welche die Spurdetektoren durchqueren, wechselwirken durch

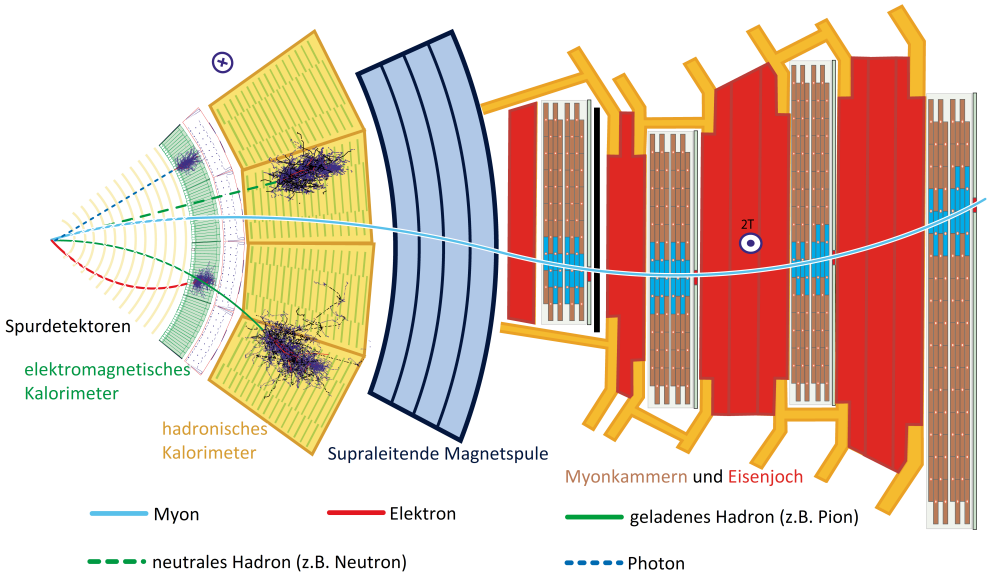

Abbildung 2.3: **Schematische Darstellung eines Ausschnitts aus dem CMS-Detektor** (Bildquelle: [18]) Es sind die verschiedenen Schichten des CMS-Detektors zu sehen. Der Ort der Protonenkollision ist ganz links am Rand angedeutet. Vom Interaktionspunkt ausgehend sind einige mögliche Trajektorien von verschiedenen Streuteilchen eingezeichnet.

Ionisation mit dem Silizium des Detektors und werden so registriert. Durch mehrere Schichten Silizium hintereinander kann so die Bahn eines Teilchens bestimmt werden. Als nächste Schicht folgt das elektromagnetische Kalorimeter (ECAL). Dieses misst die Energie von geladenen Teilchen. Es werden, vor allem durch Bremsstralung, Photonen erzeugt, aus denen durch Paarbildung wiederum Elektronen und Positronen entstehen, welche wieder Bremsstrahlung abgeben. Es entsteht ein elektromagnetischer Teilchenschauer. Die Stolzit-Kristalle ($PbWO_4$) des ECAL haben Eigenschaften eines Szintillators und absorbieren die ankommende Teilchenkaskade. Das Szintillationslicht wird detektiert und dient zur Bestimmung der Energie. Ungeladene hadronische Streuprodukte können das ECAL einfach passieren. Ihre Energie wird in der nächsten Schicht, dem hadronischen Kalorimeter (HCAL) gemessen. Das HCAL besteht abwechselnd aus Schichten von Absorbermaterial (hauptsächlich Messing) und fluoreszierenden Plastik-Szintillatoren. Im Absorbermaterial stoßen die eintreffenden Teilchen mit dem Detektormaterial und produzieren Sekundärpartikel, auf die die Szintillatoren reagieren. Die nächste Schicht ist die vorher erwähnte Magnetspule. Sie ist umgeben von einem Eisenjoch, in welchem die Myonkammern liegen. Das Eisenjoch dient dazu das Magnetfeld der Spule nach außen abzuschirmen. Die weiter innen liegenden Detektorschichten filtern alle Teilchen außer Myonen und Neutrinos. Die Kammern im Eisenjoch sind mit einem Gemisch aus Argon und Kohlendioxid gefüllt und ein elektrisches Feld ist angelegt. Die ankommenden Myonen ionisieren die Gase und erzeugen so einen Strom in der Kammer, der gemessen wird. Mithilfe der Myonenkammern werden die Bahnen und Impulse der Myonen gemessen. Da Neutrinos kaum mit Materie wechselwirken, verlassen sie den Detektor ungehindert. Auf die Produktion von Neutrinos kann jedoch über die Impulsbilanz in der Ebene des Detektors, die transversal zur Strahlrichtung liegt, geschlossen werden.

2.3.2 Koordinatensystem und wichtige Messgrößen

Das kartesische Koordinatensystem, das zur Bestimmung von Orten bzw. Richtungen im Detektor genutzt wird, hat seinen Ursprung im Kollisionspunkt der Teilchenstrahlen.

Die x-Achse zeigt auf den gedachten Mittelpunkt des LHCs. Die z-Achse ist parallel zu den einlaufenden Teilchenstrahlen und so orientiert, dass sie in Richtung des gegen den Uhrzeigersinn laufenden Teilchenstrahls zeigt. Die y-Achse verläuft in der Konsequenz so, dass sie nach oben zeigt. Die xy-Ebene entspricht deshalb einem Querschnitt durch den Detektor. Durch die Symmetrie des Detektors bietet es sich jedoch an, eine Transformation zu den Koordinaten η und ϕ vorzunehmen, wobei ϕ dem Azimutwinkel in der xy-Ebene entspricht und die Pseudorapidität η aus dem Polarwinkel θ zur z-Achse berechnet wird.

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right) \tag{2.1}$$

Mit diesem Koordinatensystem können nun verschiedene Messgrößen definiert werden. So ist zum Beispiel der Abstand zweier Punkte in der $\eta\phi$-Ebene $\Delta R(a, b)$ durch folgenden Ausdruck gegeben:

$$\Delta R(a, b) = \sqrt{(\eta_a - \eta_b)^2 + (\phi_a - \phi_b)^2}. \tag{2.2}$$

Eine weitere wichtige Messgröße ist der Transversalimpulsvektor $\boldsymbol{p}_\mathrm{T}$. Bei $\boldsymbol{p}_\mathrm{T} = (p_x, p_y)^T$ handelt es sich um den Impuls senkrecht zur Strahlrichtung. Der Betrag des Transversalimpulses berechnet sich durch

$$|\boldsymbol{p}_\mathrm{T}| = \sqrt{p_x^2 + p_y^2}. \tag{2.3}$$

Vor der Teilchenkollision ist der Transversalimpuls der Protonen verschwindend gering. Für einzelne Streuprodukte, die im Detektor landen, ist p_T eine wichtige Messgröße und liegt im GeV-Bereich. Pro Teilchenkollision muss jedoch wegen der Impulserhaltung gelten, dass die Summe der Transversalimpulse aller Streuprodukte wieder verschwindet. Falls Teilchen den Detektor unerkannt verlassen, wie beispielsweise die in Abschnitt 2.3.1 erwähnten Neutrinos, dann fehlt ein Bruchteil des detektierten Transversalimpulses. Die fehlende Transversalenergie wird mit $\not{E}_\mathrm{T}$ bezeichnet. Es gilt für n detektierte Transversalimpulse von Streuprodukten einer Kollision

$$\not{E}_\mathrm{T} = \left| -\sum_{i=1}^{n} \boldsymbol{p}_{\mathrm{T}i} \right|. \tag{2.4}$$

2.4 Signalprozess und Untergrundprozesse

Im Rahmen dieser Arbeit soll zwischen fünf verschiedenen teilchenphysikalischen Prozessen unterschieden werden. Diese werden im Folgenden kurz erläutert. Da diese Arbeit auf [6] und [19] aufbaut, werden hier dieselben Prozesse berücksichtigt, wie in diesen Arbeiten.

2.4.1 Signalprozess

Der Prozess, der im Weiteren als Signalprozess bezeichnet wird, ist die Erzeugung eines Top-Quark-Antiquark-Paares und eines Higgs-Bosons, bei einer Schwerpunktsenergie von $\sqrt{s} = 13$ TeV am CMS Experiment. Im weiteren Verlauf zerfällt das Higgs-Boson mit einem Verzweigungsverhältnis von 58% [4] in ein Bottom-Quark-Antiquark-Paar. Nur dieser $H \to b\bar{b}$-Prozess wird in dieser Arbeit berücksichtigt. Die Top-Quarks zerfallen jeweils in ein W-Boson und ein Bottom-Quark. Die W-Bosonen zerfallen nach absteigender Wahrscheinlichkeit geordnet, entweder in zwei Quark-Antiquark-Paare, ein Quark-Antiquark-Paar und ein geladenes Lepton- und Neutrino-Paar oder in zwei geladene Lepton- und Neutrino-Paare [5].
In dieser Arbeit wird der sogenannte semi-leptonische Zerfall betrachtet, der dem zweitgenannten Zerfall in ein Quark-Antiquark-Paar und ein Leptonen-Neutrino-Paar entspricht. Entsprechend besteht der Endzustand des Signalprozesses aus vier Bottom-Quarks, einem geladenen Lepton, einem Neutrino des gleichen Flavours und einem Quark-Antiquark-Paar. Ein Feynman-Diagramm des Signalprozesses ist in Abbildung 2.4 zu sehen. Der Prozess wird im weiteren Verlauf der Arbeit auch $\mathrm{t\bar{t}H(b\bar{b})}$ oder einfach $\mathrm{t\bar{t}H}$ genannt.

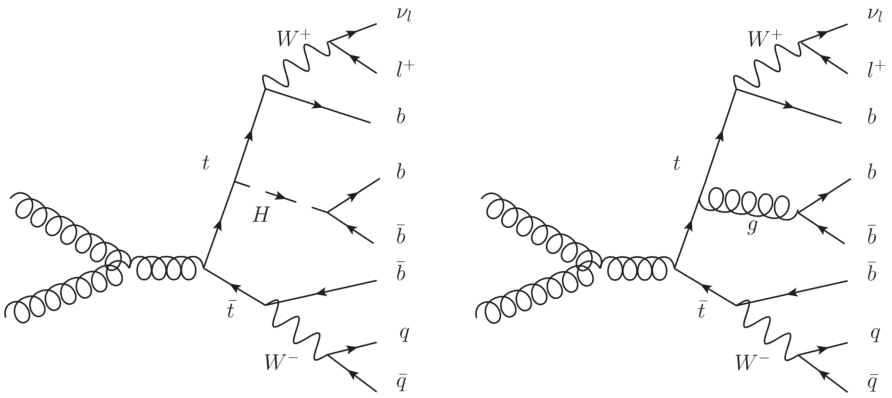

Abbildung 2.4: **Feynman-Diagramme des $t\bar{t}H(b\bar{b})$-Prozesses und des $t\bar{t}b\bar{b}$-Prozesses** (Bildquelle: [20]) Links wird ein Higgs Boson vom Top Quark abgestrahlt und rechts ein Gluon. Sowohl Higgs-Boson als auch Gluon zerfallen in einen Bottom-Quark-Antiquark Endzustand. Das Top-Quark-Antiquark-System zerfällt in beiden Fällen semi-leptonisch.

2.4.2 Untergrundprozesse

Abgesehen vom Signalprozess treten bei den Teilchenkollisionen bei CMS noch andere Prozesse auf, die den gleichen- oder einen ähnlichen Endzustand wie der $t\bar{t}H(b\bar{b})$-Prozess haben. In dieser Arbeit werden vier davon unterschieden. Die Prozesse, die $t\bar{t}H$ am meisten ähneln, sind die $t\bar{t}$ + heavy Flavour-Prozesse. Bei ihnen gibt es im Endzustand mindestens noch zwei Bottom-Quarks zusätzlich zu denen, die aus den Zerfällen der Top-Quarks kommen. In dieser Arbeit werden zwei verschiedene $t\bar{t}$ + heavy flavor-Prozesse betrachtet. Bei $t\bar{t} + b\bar{b}$ werden entweder beide b-Jets rekonstruiert oder nur einer, während der andere außerhalb der Akzeptanz liegt. Bei $t\bar{t}$ + 2b werden beide b-Jets als einer rekonstruiert [5]. Weitere Prozesse mit leichteren Quarks, die berücksichtigt werden, sind der $t\bar{t} + c\bar{c}$-Prozess, bei dem es Charm-Quarks zusätzlich im Endzustand gibt, sowie $t\bar{t}$ + light flavor, bei dem im Endzustand leichte Quarks (u, d, s) oder Gluonen vorliegen.
In Abbildung 2.4 ist auf der rechten Seite ein Feynman-Diagramm des $t\bar{t} + b\bar{b}$-Prozesses zu sehen. Die $t\bar{t}$+$c\bar{c}$- und $t\bar{t}$+light flavor haben nicht denselben Endzustand wie $t\bar{t}H$. Sie stellen aufgrund ihrer ähnlichen Signatur im Detektor und ihres höheren Wirkungsquerschnitts trotzdem Untergrundprozesse dar, die nicht zu vernachlässigen sind. Es gibt noch weitere Untergrundprozesse zu $t\bar{t}H$, die aber in dieser Arbeit nicht berücksichtigt werden [5].

3 Neuronale Netze

An den großen Detektoren an Teilchenbeschleunigern werden riesige Mengen an Daten aufgenommen. Um Erkenntnisse über die Physik zu gewinnen, welcher die beobachteten Prozesse unterliegen, müssen diese Daten zunächst ausgewertet werden. Ein Problem, das bei der Datenanalyse des $t\bar{t}H$-Prozesses auftritt, ist, dass es dominante Untergrundprozesse gibt, die bei der Erhebung der Daten ebenfalls erfasst werden, da diese dem Signalprozess stark ähneln. Um die relevanten Signaldaten von den Untergründen zu trennen, kommen in der Analyse neuronale Netze zum Einsatz. Neuronale Netze sind informationsverarbeitende Strukturen, die zunächst durch sogenanntes Training an eine bestimmte Aufgabe angepasst werden. Im Falle dieser Arbeit entspricht diese Aufgabe der Unterscheidung von teilchenphysikalischen Prozessen. Nach dem Training kann das neuronale Netz seine Aufgabe mit einer bestimmten Genauigkeit erfüllen.
Das folgende Kapitel enthält in Abschnitt 3.1 eine allgemeine Einführung zu künstlichen neuronalen Netzen (engl. artificial neural networks, oder kurz: ANNs) um anschließend, in Abschnitt 3.2 genauer auf die in dieser Arbeit verwendeten bayesischen neuronalen Netze (engl. bayesian neural networks, oder kurz BNNs) einzugehen.

3.1 Künstliche neuronale Netze

Die Grundidee von künstlichen neuronalen Netzen ist die Erzeugung eines informationsverarbeitenden Systems nach Vorbild des menschlichen Gehirns. Wie die biologischen Vorbilder bestehen ANNs aus Knotenpunkten, den sogenannten Neuronen, welche miteinander verbunden sind. Über die Verbindungen werden Informationen von Neuron zu Neuron geleitet, wo sie dann weiterverarbeitet werden. ANNs haben viele verschiedene Verwendungszwecke. Sie werden beispielsweise zur Steuerung von verschiedensten technischen Prozessen, zur Entscheidungsunterstützung oder zum Erkennen und Interpretieren von Mustern eingesetzt [21]. In dieser Arbeit werden sie verwendet, um Sets von simulierten Messwerten in verschiedene Klassen einzuordnen, wobei diese Klassen genau den in Abschnitt 2.4 beschriebenen Prozessen entsprechen. Falls nichts anderes angegeben wird, stützt sich dieses Unterkapitel auf [22].

3.1.1 Aufbau und Funktionsweise

Die Neuronen der neuronalen Netze dieser Arbeit werden in hintereinander liegende Schichten (engl. layer) eingeteilt. Diese Schichten werden dann miteinander vernetzt. Wenn

jedes Neuron einer Schicht mit allen Neuronen der nächsten Schicht verbunden ist, so spricht man von vollständig verbundenen Schichten (engl. fully connected layers bzw. dense layers). Solche Netze aus vollständig verbundenen Schichten werden in dieser Arbeit verwendet. Die erste Schicht des Netzes ist die Eingangsschicht (engl. input layer). Diese wird zur Einspeisung der Informationen verwendet. Das n-te Neuron der Eingangsschicht wird im Folgenden mit x_n, $n \in \{1, ..., N\}$ bezeichnet, wobei N die Anzahl der Eingangsneuronen ist. Nach der Eingangsschicht folgen die verdeckten Schichten (engl. hidden layers), welche die eingegebenen Informationen weiterverarbeiten. Ihre Neuronen werden mit h_j, $j \in \{1, ..., J\}$ benannt. Dabei ist J die Anzahl der Neuronen der verdeckten Schicht. Die letzte Schicht des Netzwerks ist die Ausgabeschicht (engl. output layer) mit M Neuronen. Sie gibt die verarbeiteten Informationen wieder aus. Ihre Neuronen werden mit o_m, $m \in \{1, ..., M\}$ bezeichnet.
Die Anzahl der Neuronen der Eingangsschicht N hängt von den Informationen ab, die dem Netz zur Verfügung gestellt werden. Für jede Eingabevariable gibt es ein Neuron in der Eingangsschicht. Sowohl die Anzahl der verdeckten Schichten als auch deren jeweilige Anzahl an Neuronen J kann beliebig gewählt werden. Ausgabeneuronen gibt es bei klassifizierenden Netzwerken genau so viele, wie es Klassen gibt, in welche die Eingabedaten eingeteilt werden sollen. Sind nur ein oder zwei Ausgabeneuronen vorhanden, so wird von binärer Klassifikation gesprochen. Wenn das Netzwerk zwischen mehreren Klassen unterscheiden kann, handelt es sich um Multiklassifikation. In Abbildung 3.1 ist ein neuronales Netz aus vollständig verbundenen Schichten, mit einer Eingangsschicht mit N Neuronen, einer verdeckten Schicht mit J Neuronen und einer Ausgabeschicht mit M Neuronen schematisch dargestellt. Der Aufbau eines spezifischen neuronalen Netzes wird auch Architektur des Netzes genannt. Die Informationsverarbeitung des neuronalen Netzes geschieht in den

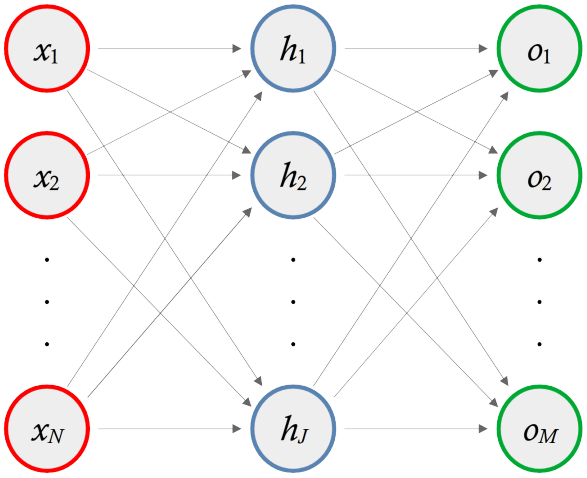

Abbildung 3.1: **Schematische Darstellung der Architektur eines ANNs.** Von der Eingangsschicht (rot) aus werden die Daten in Pfeilrichtung zur verdeckten Schicht (blau) weitergeleitet und verarbeitet. Derselbe Vorgang wiederholt sich zwischen der verdeckten Schicht und der Ausgabeschicht (grün). Dass das Netz aus vollständig verbundenen Schichten besteht, erkennt man daran, dass jedes Neuron einer Schicht mit allen Neuronen der nächsten Schicht verbunden ist.

Neuronen der verdeckten Schichten. Zur Veranschaulichung ist in Abbildung 3.2 das j-te Neuron einer verdeckten Schicht dargestellt. Die Informationen, die das ANN verarbeiten soll, erreichen die Eingabeneuronen in Form von Eingabevariablen. Der Zahlenwert x_n des

n-ten Neurons wird mit dem Gewicht w_{nj} an das j-te Neuron weitergegeben. Das Neuron bildet die Summe aus den Produkten der x_n und deren jeweiligen Gewichten. Zusätzlich kann noch ein Wert b_j, ein sogenannter Bias, zu der Summe addiert werden.

$$g_j = \sum_{n=1}^{N} x_n w_{nj} + b_j \tag{3.1}$$

Die Summe lässt sich auch als Skalarprodukt eines Vektors, der aus den Eingabevariablen besteht, und eines anderen Vektors, welcher die Gewichte enthält, schreiben.

$$g_j = \boldsymbol{x} \cdot \boldsymbol{w}_j + b_j \tag{3.2}$$

Auf die so erhaltene Zahl g_j wird dann die Aktivierungsfunktion f angewendet. Der erhaltene Funktionswert $f(g_j)$ wird vom Neuron weitergegeben.

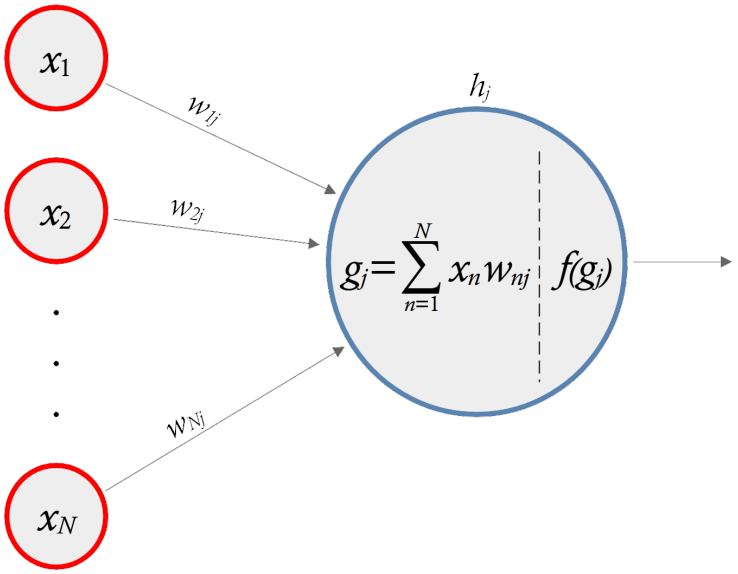

Abbildung 3.2: **Vergrößerte Darstellung des Neurons h_j der verdeckten Schicht ohne Bias.** (Abbildung an Darstellung in [22] angelehnt.) Die Neuronen der vorherigen Schicht geben die Werte x_i an das Neuron h_j weiter. Diese werden jeweils mit dem ihnen zugeordneten Gewicht w_{ij} multipliziert und anschließend zur Summe g_j aufaddiert. Auf g_j wird dann die Aktivierungsfunktion f angewendet und das Neuron gibt den Wert $f(g_j)$ an die nächste Schicht weiter.

3.1.2 Aktivierungsfunktionen

Je nach Schicht wird in den Neuronen eine andere Aktivierungsfunktion verwendet. Generell kann als Aktivierungsfunktion jede Funktion genutzt werden, die g_j wieder auf eine reelle Zahl abbildet. Oft haben Aktivierungsfunktionen eine sogenannte Schwelle θ, die dafür sorgt, dass das Neuron nur bei $g_j > \theta$ den Funktionswert $f(g_j)$ weiterleitet. Die Eingangsneuronen nutzen die Identitätsfunktion $f(x) = x$ als Aktivierungsfunktion. Sie leiten ihren Wert unverändert weiter. In den verdeckten Schichten kommen je nach Netzwerk verschiedene Aktivierungsfunktionen zum Einsatz. Die in dieser Arbeit verwendeten Netze nutzen die ReLU-Funktion

$$f^{\mathrm{ReLU}}(g_j) = \max(0, g_j). \tag{3.3}$$

Entsprechend wird g_j von ReLU unverändert weitergeleitet, wenn g_j größer ist als der Schwellenwert $\theta = 0$. Die ReLU-Funktion hat an der Stelle Null einen Knick, und ist damit nicht komplett linear. Somit eigenen sich neuronale Netze, welche ReLU verwenden auch für nichtlineare Probleme, wie sie beispielsweise in der Teilchenphysik oft auftreten.
Bei multiklassifizierenden neuronalen Netzen soll in der Ausgabeschicht jedes Neuron einen Wert o_m ausgeben, der die Wahrscheinlichkeit angibt, dass die eingegebenen Daten zur m-ten Klasse gehören, die das jeweilige Neuron darstellt. Um diese Interpretierbarkeit der Ausgabe zu gewährleisten wird die Softmax-Funktion

$$f^{\text{Softmax}}(\boldsymbol{g}, m) = \frac{\exp(g_m)}{\sum_{i=1}^{M} \exp(g_i)} \tag{3.4}$$

verwendet, die den Ausgabewert g_m auf das Intervall $[0; 1]$ abbildet. Die Gesamtheit aller M Ausgabewerte für g_m ist hierbei als Vektor $\boldsymbol{g} = (g_1, \ldots, g_M)^T$ zusammengefasst. Da im Nenner die Summe aller $\exp(g_m)$ steht, ist die Summe der Funktionswerte der Ausgabeschicht immer normiert und kann als Wahrscheinlichkeitsverteilung interpretiert werden. Ein Wert für o_m nahe an Eins bedeuetet, dass es wahrscheinlich ist, dass die Eingabe zur m-ten Klasse gehört. Ein Wert von o_m nahe an Null entspricht einer geringen Wahrscheinlichkeit für die Zugehörigkeit zur m-ten Klasse.

3.1.3 Training

Damit ein neuronales Netz seine Aufgabe verlässlich erfüllen kann, muss es an diese angepasst werden. Der Anpassungsprozess heißt Training. Beim Training wird ein Trainingsdatensatz $D = \{(\boldsymbol{x}_1, o_1^{\text{wahr}}), \ldots, (\boldsymbol{x}_d, o_d^{\text{wahr}})\}$ verwendet um die Gewichte w_{nj} und den Bias b_j der Neuronen immer weiter anzupassen, bis ein optimales Verhalten des Netzes erreicht ist.
Der Trainingsdatensatz beinhaltet d Sätze von Eingangsvariablen $\boldsymbol{x}_i$ mit $i \in \{1, \ldots, d\}$, von denen schon bekannt ist, zu welcher Klasse o_i^{wahr} sie gehören. Für den Trainingsprozess wird eine Verlustfunktion (engl. loss-function) L eingeführt, die angibt, wie gut die Vorhersagen sind, die mit den jeweiligen Gewichten und Bias gemacht werden. In dieser Arbeit wird für ANNs die Verlustfunktion Kreuzentropie (engl. categorical cross entropy, oder kurz CCE)

$$L_i^{\text{CCE}} = -\sum_{m=1}^{M} o_{i,m}^{\text{wahr}} \cdot \ln(o_{i,m}) \tag{3.5}$$

verwendet. Dabei steht i für den Index des Elements des Datensatzes und m nummeriert wie zuvor die Ausgabeneuronen bzw. die Klassen. Zu Beginn des Trainings sind Bias und Gewichte zufällig initialisiert. Dann wird ein Netzdurchlauf eines Elements des Testdatensatzes durchgeführt. Die Ausgabeschicht des Netzes trifft eine Vorhersage, die mit der wahren Klasse abgeglichen wird. Die Verlustfunktion misst den Verlust, also wie stark das Netz vom wahren Wert abweicht. Um die Gewichte zu optimieren wird ein Optimierungsalgorithmus verwendet. Hier wird eine Methode namens Gradientenverfahren beschrieben. Diese nutzt die Ableitung der Verlustfunktion nach den Gewichten, um diese zu minimieren. Die neuen Gewichte werden bestimmt mit

$$w_{ij}^{\text{neu}} = w_{ij} - \gamma \frac{\partial L}{\partial w_{ij}}. \tag{3.6}$$

Der Parameter γ in dieser Gleichung wird Lernrate genannt. Diese wird vor dem Training als Hyperparameter des Netzes eingestellt und legt fest, wie stark die Gewichte pro Trainingsdurchgang durch den Gradienten angepasst werden. Ist der Netzdurchlauf des ersten

Elements von D abgeschlossen, wird der Prozess wiederholt bis alle Elemente das Netzwerk durchlaufen haben. Ein solcher Durchlauf wird Epoche genannt. Nach einer Epoche werden die Gewichte des Netzes den Berechnungen entsprechend angepasst und die nächste Epoche beginnt.
Der in dieser Arbeit verwendete Optimierungsalgorithmus heißt Adam [23]. Bei diesem Algorithmus handelt es sich um eine Erweiterung des Gradientenverfahrens. Bei Adam wird D vor dem Training in Unterdatensätze, sogenannte Mini-Batches, eingeteilt. Die Gewichte werden pro Mini-Batch aktualisiert, um das Training effizienter zu machen. Adam passt außerdem während des Trainings seine Lernrate an. Der Algorithmus, welcher Adam und dem Gradientenverfahren zugrunde liegt, heißt Fehlerrückführung (engl: backpropagation of error). Er beginnt die Berechnung der w^{neu} im Output-Layer und propagiert dann rückwärts durch das Netz. Der Vorteil dieses Vorgehens besteht darin, dass beim Berechnen der Gewichte in vorherigen Schritten berechnete Terme genutzt werden können. Dadurch wird die Rechenzeit verkürzt. Eine genaue Herleitung findet sich in [24].
Beim Training kann es passieren, dass das Netz sich zu sehr an bestimmte Charakteristiken des Trainingsdatensatzes anpasst, sodass es bei Anwendung auf Daten, die nicht zum Trainingsdatensatz gehören, schlechtere Ergebnisse liefert. Dieser Effekt wird Übertraining genannt [25]. Um dem zuvorzukommen wird D in einen Trainingsdatensatz, einen Validierungsdatensatz und einen Testdatensatz aufgeteilt. Das Netz wird nach jeder Epoche mit dem Validierungsdatensatz ausgewertet. Die hierbei ermittelten Werte der Verlustfunktion werden nicht für weiteres Training genutzt. Wenn der Verlust des Validierungsdatensatzes größer wird, während der Verlust des Trainings sinkt, ist das ein Zeichen für Übertraining. Um zu verhindern, dass das Netz trainingsdatenspezifische Eigenschaften lernt, wird die L2-Norm

$$\text{L2} = \lambda \cdot \sum_{i=1}^{W} w_i^2 \tag{3.7}$$

verwendet. In dieser Gleichung steht W für die Anzahl aller Gewichte des Netzes. Der L2-Parameter λ muss vor dem Training als Hyperparameter festgelegt werden. Der L2-Term wird zur Verlustfunktion addiert. Wenn das Quadrat aller Gewichte des Netzwerks zu groß wird, so liefert der L2-Term einen erheblichen Beitrag zur Verlustfunktion. Somit wird gewährleistet, dass einzelne Gewichte nicht zu groß werden und das Netz für andere Daten anwendbar bleibt [26]. Nach dem Training wird der Testdatensatz genutzt, um das Netz auszuwerten. Mit seiner Hilfe wird bewertet, wie gut die Klassifizierungsfähigkeit des Netzes ist, die durch das Training erzielt wurde.

3.2 Bayesische neuronale Netze

Die in Abschnitt 3.1 beschriebenen ANNs können zwar Vorhersagen über die Klasse der eingegebenen Daten treffen, jedoch geben sie keine Aussage darüber, wie zuverlässig die gegebenen Vorhersagen sind.
Um Vorhersagen über die Unsicherheiten der Netzausgaben zu treffen werden bayesische neuronale Netze (BNNs) eingesetzt. BNNs nutzen, anders als herkömmliche ANNs, keine Zahlen als Gewichte. Stattdessen verwenden sie Wahrscheinlichkeitsverteilungen, die beim Training angepasst werden. In Abbildung 3.3 ist ein BNN schematisch dargestellt. Im Folgenden wird eine Übersicht über BNNs gegeben, so wie sie in dieser Arbeit verwendet werden. Falls keine anderen Quellen angegeben sind, basieren die in diesem Kapitel gegebenen Informationen auf [27] und [28].

3.2.1 Satz von Bayes

Grundlage und namensgebend für BNNs ist der Satz von Bayes. Bei diesem handelt es sich um einen grundlegenden Satz der Wahrscheinlichkeitstheorie, welcher die bedingte

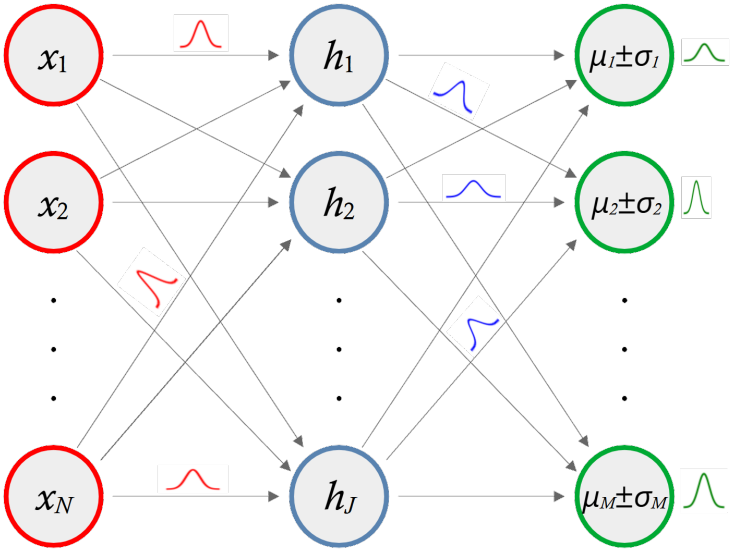

Abbildung 3.3: **Schematische Darstellung der Architektur eines BNNs.** Die Wahrscheinlichkeitsdichtefunktionen, die als Gewichte genutzt werden sind als Kurven über den entsprechenden Pfeilen, welche die Neuronen verbinden, dargestellt. Zur besseren Übersichtlichkeit ist nicht über jedem Pfeil eine Kurve zu sehen. Der Ausgabewert der Ausgabeneuronen ist mit seiner zugehörigen Unsicherheit im Kreis zu sehen, der das jeweilige Neuron repräsentiert. Die Verteilung, die von diesen Werten beschrieben wird, ist hinter den Ausgabeneuronen symbolisch dargestellt.

Wahrscheinlichkeit beschreibt. In dieser Arbeit wird der Satz genutzt, um die Gewichtsverteilungen des Netzwerks auszudrücken. Der Satz ist durch

$$p(\boldsymbol{w}|D) = \frac{p(D|\boldsymbol{w})p(\boldsymbol{w})}{p(D)} \tag{3.8}$$

gegeben. Hierbei steht der Vektor $\boldsymbol{w}$ für einen Satz von Gewichten des Modells. Die A-posteriori-Verteilung (engl. posterior distribution) $p(\boldsymbol{w}|D)$ entspricht den Gewichtsverteilungen. Diese werden berechnet aus der bedingten Wahrscheinlichkeitsdichte $p(D|\boldsymbol{w})$ (engl. likelihood), der A-priori-Verteilung $p(\boldsymbol{w})$ (engl. prior distribution) und der Wahrscheinlichkeitsverteilung der Daten $p(D)$. Die A-priori-Verteilung wird vor Beginn des Netztrainings mithilfe von allgemeinem Wissen über das Problem festgelegt. In dieser Arbeit wird dafür die Normalverteilung

$$p(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{3.9}$$

gewählt. Die Verteilung $p(D)$ hängt allein von den verwendeten Trainingsdaten D ab und ist deshalb konstant für alle Gewichtsverteilungen. Die Wahrscheinlichkeit für einen Satz von Gewichten die Eingabedaten $\boldsymbol{x}_i$ dem jeweils richtigen Output o_i^{wahr} zuzuorden, ist durch $p(D|\boldsymbol{w})$ beschrieben. Durch das Training werden die A-posteriori-Verteilungen ermittelt, mit denen das Netzwerk die Eingaben am besten klassifiziert.

3.2.2 Variational Inference

Bei der praktischen Bestimmung der A-posteriori-Verteilung wird nicht Gleichung (3.8) verwendet. Eigentlich müsste über alle möglichen Gewichtskonfigurationen integriert werden,

was die Gleichung im Allgemeinen unlösbar macht. Die gesuchte Verteilung muss daher mithilfe eines Algorithmus approximiert werden. Der Ansatz hierfür heißt Variational Inference. Zunächst wird eine Verteilung $p_\theta(\boldsymbol{w})$ definiert, die mit θ parametrisiert ist und $p(\boldsymbol{w}|D)$ annähert. Hierfür wird eine Normalverteilung (Gleichung (3.9)) mit den Parametern $\theta = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ angesetzt. Nun wird die Kullback-Leibler-Divergenz [29]

$$\mathrm{KL}[p_\theta(\boldsymbol{w}), p(\boldsymbol{w}|D)] = \int p_\theta(\boldsymbol{w}) \cdot \ln\left(\frac{p_\theta(\boldsymbol{w})}{p(\boldsymbol{w}|D)}\right) \cdot d\boldsymbol{w} \geq 0 \tag{3.10}$$

als Maß für die Ähnlichkeit der beiden Verteilungen genutzt. Diese ist bezüglich der Parameter θ zu minimieren. Ein neuronales Netz kann diese Minimierung vornehmen. Die Umformung von Gleichung (3.10) unter Verwendung von Gleichung (3.8) und anschließendes Zerlegen des Logarithmus in Summanden liefert die Verlustfunktion für dieses neuronale Netz

$$L_{\mathrm{BNN}} = \int p_\theta(\boldsymbol{w}) \cdot \ln\left(\frac{p_\theta(\boldsymbol{w})p(D)}{p(D|\boldsymbol{w})p(\boldsymbol{w})}\right) \cdot d\boldsymbol{w} \tag{3.11}$$

$$= -\int p_\theta(\boldsymbol{w}) \cdot \ln(p(D|\boldsymbol{w})) \cdot d\boldsymbol{w} + \mathrm{KL}[p_\theta(\boldsymbol{w}), p(\boldsymbol{w})] + \ln(p(D)) \cdot \int p_\theta(\boldsymbol{w}) \cdot d\boldsymbol{w}. \tag{3.12}$$

Der erste Term stimmt mit der Kreuzentropie, Gleichung (3.5), überein. Der zweite Term ist die KL-Divergenz zwischen den approximierten A-posteriori-Verteilungen $p_\theta(\boldsymbol{w})$ und den A-Priori-Verteilungen $p(\boldsymbol{w})$. Dieser Term erfüllt die gleiche Aufgabe wie der L2-Regularisierungsterm der ANNs. Deshalb haben klassifizierende BNNs und ANNs korrespondierende Verlustfunktionen, was sie vergleichbar macht. Der dritte Term ist eine Konstante und daher nicht notwendig für die Minimierung der Verlustfunktion. Das Integral $\int p_\theta(\boldsymbol{w}) \cdot d\boldsymbol{w}$ ergibt Eins und der Vorfaktor hängt nicht von θ ab. Folglich hat die im BNN verwendete Verlustfunktion die Form

$$L_{\mathbf{BNN}} = -\int p_\theta(\boldsymbol{w}) \cdot \ln(p(D|\boldsymbol{w})) \cdot d\boldsymbol{w} + \int p_\theta(\boldsymbol{w}) \cdot \ln\left(\frac{p_\theta(\boldsymbol{w})}{p(\boldsymbol{w})}\right) \cdot d\boldsymbol{w}. \tag{3.13}$$

3.2.3 Auswertung

Das BNN rechnet in der Praxis nicht mit Wahrscheinlichkeitsverteilungen, sondern nutzt Zufallszahlen, die entsprechend der jeweiligen Funktion verteilt sind. Eine einmalige Auswertung des Netzes ergibt daher wie bei ANNs eine Zahl. Allerdings ist zu beachten, dass BNNs im Vergleich zu ANNs wegen der genutzten Zufallszahlen nicht deterministisch sind. Jede Auswertung eines Satzes von Eingangsvariablen $\boldsymbol{x}$ durch ein BNN erzeugt in den Ausgabeneuronen ein anderes Ergebnis. Eine diskrete Verteilung wird durch mehrfaches Auswerten von $\boldsymbol{x}$ mit dem BNN produziert. Um die erhaltenen diskreten Verteilungen $\boldsymbol{o}_m$ als Parameter einer Normalverteilung schreiben zu können, werden aus ihnen der Mittelwert μ_m

$$\mu_m = \frac{1}{S}\sum_{s=1}^{S} o_{m,s} \tag{3.14}$$

und die Standardabweichung σ_m

$$\sigma_m = \sqrt{\frac{1}{S}\sum_{s=1}^{S}(o_{m,s} - \mu_m)^2} \tag{3.15}$$

berechnet. Dabei steht S für die Anzahl der Auswertungen der Daten durch das Netz. So erhält man für jedes Ausgabeneuron o_m einen Mittelwert μ_m, der der Vorhersage des ANNs entspricht, sowie zusätzlich eine Standardabweichung σ_m, die ein Maß der Unsicherheit des

Netzes für die jeweilige Vorhersage angibt.
Um die Ergebnisse statistisch repräsentativ zu machen sind Vielfachauswertungen der Daten durch das Netz nötig. Die BNNs dieser Arbeit werden deshalb immer 100 mal ausgewertet, falls nicht anders angegeben. Die Mehrfachauswertung von BNNs kann mit der einzelnen Auswertung von einem Ensemble an ANNs verglichen werden, wie in Abbildung 3.4 illustriert. In Kapitel 5.2 wird ein Vergleich von BNNs mit Ensembles von ANNs durchgeführt.

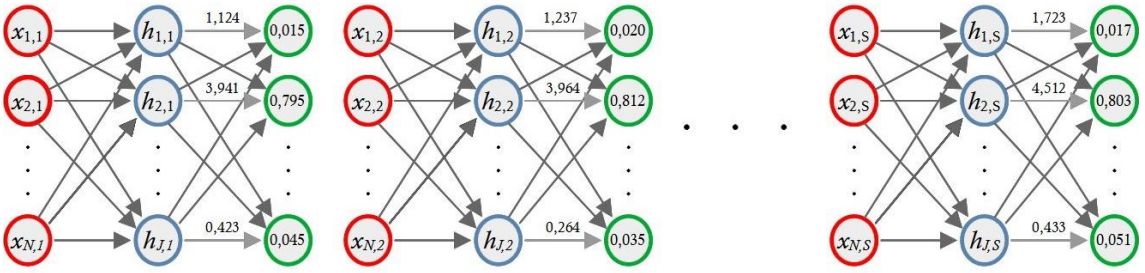

Abbildung 3.4: **Ensemble von mehreren ANNs.** (Abbildung an Darstellung in [6] angelehnt.) Zu sehen sind mehrere fertig trainierte ANNs. Einige Gewichte sind exemplarisch als Zahlen über den Pfeilen, welche die Neuronen verbinden, zu sehen. Die Ausgabewerte stehen in den entsprechenden Neuronen. Von einem solchen Ensemble an ANNs können, wie bei einem BNN, Mittelwert und Standardabweichung berechnet werden. Alle Zahlenwerte sind willkürlich gewählt.

4 Methodik dieser Arbeit

Für die praktische Umsetzung der multiklassifizierenden BNNs wird als Grundlage die Open Source Machine-Learning Frameworksammlung DRACO-MLfoy [30] genutzt. Sie beinhaltet Werkzeuge um klassifizierende neuronale Netze für teilchenphysikalische Prozesse zu trainieren und auszuwerten. DRACO-MLfoy basiert auf der Programmiersprache Python und nutzt die Bibliothek Keras [31] und die Backend-Schnittstelle Tensorflow [32], welche die Implementierung von neuronalen Netzen erlauben. Die Möglichkeit binär klassifizierende BNNs zu nutzen wurde im Zuge der Masterarbeit [6] zu DRACO-MLfoy hinzugefügt. Diese verwenden die Dense Variational-Schicht (DV) aus der Bibliothek Tensorflow Probability [33]. Im Rahmen der Bachelorarbeit [19] wurden verschiedene Optimierungsstrategien für diese binären BNNs untersucht.
In diesem Kapitel wird in Abschnitt 4.1 genauer auf die verwendeten Daten, sowie die Konfiguration der in dieser Arbeit verwendeten Netze eingegangen. In Abschnitt 4.2 werden anschließend die für diese Arbeit genutzten Auswertungsmethoden aus DRACO-MLfoy erläutert. Aufbauend darauf wird in Abschnitt 4.3 ein Überblick über die Methodik und Ziele dieser Arbeit gegeben.

4.1 Verwendete Daten, Eingangsvariablen und Hyperparameter

Um ein neuronales Netz trainieren zu können, muss vorher der Datensatz, der zum Trainieren, Validieren und Testen genutzt wird, festgelegt werden. Die verwendeten Eingangsvariablen und die einstellbaren Hyperparameter müssen ebenso vor dem Training festgelegt werden. Hyperparameter sind solche Parameter des Netzes, die nicht durch Training anpassbar sind und vor dem Training festgelegt werden. Die Spezifika dieser Randbedingungen des Trainings werden in diesem Abschnitt genauer erläutert.

4.1.1 Daten und Eingangsvariablen

Wie in [6] und [19] wird in dieser Arbeit ein Datensatz aus simulierten Daten von Teilchenkollisionen bei einer Schwerpunktsenergie von 13 TeV verwendet. Diese Spezifikationen entsprechen den experimentellen Bedingungen der Messungen am CMS-Experiment von 2017. Der Datensatz enthält ca. 1,8 Millionen Ereignisse. Mehr Informationen über die Simulation der betrachteten Prozesse finden sich in [5]. Tabelle 4.1 zeigt, wie der Datensatz in Trainings-, Validierungs- und Testdatensatz aufgeteilt wird und wie viele Ereignisse zu

für die betrachteten Prozesse vorhanden sind.

Tabelle 4.1: **Unterdatensätze mit jeweiliger Anzahl an Ereignissen.** Zu sehen ist die Zusammensetzung der verwendeten Daten. Die Gesamtanzahl der Ereignisse entspricht sowohl der Summe von Trainings-, Validierungs- und Testdatensatz, als auch der Summe der Ereignisanzahlen der einzelnen Prozesse.

Unterdatensatz	Anzahl der Ereignisse
$t\bar{t}H$	1.286.754
$t\bar{t} + b\bar{b}$	170.958
$t\bar{t}$ + 2b	46.032
$t\bar{t} + c\bar{c}$	60.568
$t\bar{t}$ + light flavor	221.751
Training	1.071.638
Validierung	357.213
Test	357.212
Gesamt	1.786.063

Für die neuronalen Netze dieser Arbeit werden 25 Eingangsvariablen genutzt. Diese sind in [19] im Variablenranking als die 25 relevantesten Eingangsvariablen aus einer großen Menge verschiedener Variablen identifiziert worden. Die Relevanz einer Variable wird dort als die Summe der Gewichte der jeweiligen Variable in der ersten verdeckten Schicht definiert. Je höher die Summe der zugehörigen Gewichte, desto höher wird die Relevanz der Variablen eingeordnet. Eine Liste der verwendeten Eingangsvariablen ist in Tabelle 4.2 zu finden.

Tabelle 4.2: **Selektierte Eingangsvariablen, übernommen aus [19].** Die Eingangsvariablen sind innerhalb ihrer jeweiligen Kategorie nach absteigender Relevanz sortiert. In Tabelle A.2 im Anhang werden die hier aufgeführten Größen genauer beschrieben.

betrachtete Objekte	Eingangsvariablen
Jets	M_3, N_{Jets}, b-Tag-Wert 4, $p_T(\text{j}_1)$, $M(\text{j}_1)$, $\overline{\text{b-Tag-Wert}}$, $p_T(\Delta R_{\text{min}}(\text{j},\text{j}))$, $\overline{\Delta R}(\text{j},\text{j})$, $\overline{\Delta\eta}(\text{j},\text{j})$, $\Delta R_{\text{min}}(\text{j},\text{j})$, b-Tag-Wert 3, $H_T(\text{j})$
b-Jets	$M_2(\text{t},\text{t})_{125}$, $p_T(\Delta R_{\text{min}}(\text{t},\text{t}))$, $\overline{\Delta\eta}(\text{t})$, $M_2(\Delta R_{\text{min}}(\text{t},\text{t}))$, $\Delta R_{\text{min}}(\text{t},\text{t})$, $\overline{\eta}(\text{t})$, $\overline{M}(\text{t})$, $\overline{\Delta R}(\text{t})$
Leptonen	$\eta(l)$, $p_T(l)$
Lepton & b-Jet	$M(\Delta R_{\text{min}}(\text{l},\text{t}))$, $\Delta R_{\text{min}}(\text{l},\text{t})$
Jets, Lepton & MET	$M(\text{j},\text{l},\text{MET})$

4.1.2 Netzkonfiguration und Hyperparameter

Der Test der ermittelten Variablenkofiguration für verschiedene Netzarchitekturen in [19] zeigt, dass diese bei binär klassifizierenden BNNs mit einer verdeckten Schicht und 50 bis 200 verdeckten Neuronen die besten Ergebnisse liefert. Deshalb werden für die multiklassifizierenden BNNs dieser Arbeit ebenfalls solche Netzarchitekturen verwendet. Die

Tabelle 4.3: **Hyperparameter der verwendeten BNNs.** Größen, die im Verlauf des Trainings angepasst werden, sind mit dem Superskript ^{T} versehen. Wenn Hyperparameter zu einem bestimmten Zweck verändert werden, so wird im entsprechenden Abschnitt nochmals gesondert darauf verwiesen.

Hyperparameter	Wert
Architektur	1 Eingangsschicht, 25 Neuronen 1 verdeckte DV-Schicht, 50 Neuronen 1 DV-Ausgabeschicht, 5 Neuronen
Batchgröße	5000 Ereignisse
Optimierungsalgorithmus	ADAM
Lernrate	10^{-3}
Aktivierungsfunktion verdeckte Schicht	RELU
Aktivierungsfunktion Ausgabeschicht	SOFTMAX
Initialisierung A-posteriori-Verteilung	$\mu_{\text{post}}^T = 0$, $\sigma_{\text{post}}^T = 1$
Initialisierung A-priori-Verteilung	$\mu_{\text{prior}}^T = 0$, $\sigma_{\text{prior}} = 3$
Anzahl der Netzauswertungen	100

multiklassifizierenden BNNs in dieser Arbeit verfügen, wenn nichts anderes angegeben wird, über 25 Neuronen für die Eingangsvariablen in der Eingangsschicht, eine verdeckte Schicht mit 50 Neuronen und, den fünf Klassen der Ereignisse entsprechend, fünf Ausgabeneuronen. In der verdeckten Schicht wird als Aktivierungsfunktion die in Abschnitt 3.1.2 beschriebene ReLU-Funktion benutzt. Für die Aktivierungsfunktion der Ausgabeschicht kommt die, ebenfalls in Abschnitt 3.1.2 eingeführte, Softmax-Funktion zum Einsatz. Für das Training werden die Daten in Batch-Pakete der Größe von 5000 Ereignissen eingeteilt. Der in Abschnitt 3.1.3 beschriebene Adam wird als Optimierungsalgorithmus verwendet. Für die Verlustfunktion kommt die in Abschnitt 3.2.2 hergeleitete Verlustfunktion für BNNs zum Einsatz.
Die Gewichte der verwendeten BNNs verfügen über drei trainierbare Parameter. Diese sind der Mittelwert μ_{prior} der A-priori-Verteilung sowie der Mittelwert μ_{post} und die Standardabweichung σ_{post} der A-posteriori-Verteilung. Diese sind zu Beginn des Trainings alle mit zufälligen Werten initialisiert. Die Standardabweichung der A-priori-Verteilung σ_{prior} ist nicht trainierbar. In [6] wurde mittels einer Studie der Netzauswertungen für verschiedene σ_{prior} gefunden, dass $\sigma_{\text{prior}} = 3$ eine gute Wahl für diesen Parameter ist.
Wenn nichts anderes angegeben wird, werden die BNNs in dieser Arbeit 100 mal ausgewertet, um den Mittelwert und die Standardabweichung der Vorhersage zu bestimmen. Die Hyperparameter sind in Tabelle 4.3 zusammengefasst.

4.2 Auswertungsmethoden für neuronale Netze

Da innerhalb der Programmierumgebung von DRACO-MLfoy die korrekte Benennung der zu unterscheidenden Prozessklassen aufgrund der in den Namen vorkommenden Sonderzeichen nicht vorhanden ist, sind diese in den nachfolgenden Diagrammen der Arbeit mit Abkürzungen bezeichnet. Die Zuordnung der Abkürzungen findet sich in Tabelle 4.4.

Eine wichtige Möglichkeit um die Leistungsfähigkeit eines neuronalen Netzes zu beurteilen ist die Ermittlung des ROC-AUC-Werts. Diesen erhält man, indem die Rate der richtig klassifizierten Ereignisse des Testdatensatzes gegen die Rate der falsch klassifizierten Ereignisse des Testdatensatzes aufgetragen werden. Durch diese Auftragung entsteht die ROC-Kurve. Das Integral der ROC-Kurve ergibt den ROC-AUC-Wert. Ein ROC-AUC-Wert von 0,5 entspricht dem willkürlichen Einordnen der zu klassifizierenden Ereignisse. Ein

Tabelle 4.4: **Abkürzungen der Prozesse in Diagrammen.**

Prozess	Abkürzung
$t\bar{t}H$	ttH
$t\bar{t} + b\bar{b}$	ttbb_b
$t\bar{t}$ + 2b	tt2b
$t\bar{t} + c\bar{c}$	ttcc
$t\bar{t}$ + light flavor	ttlf

ROC-AUC-Wert gleich eins hingegen bedeutet, dass das neuronale Netz die Testdaten alle richtig klassifiziert. Es gilt: Je näher der ROC-AUC-Wert an eins ist, desto leistungsfähiger ist das neuronale Netz.
DRACO-MLfoy bietet die Möglichkeit Wahrheitsmatrizen (engl. confusion matrices) für multiklassifizierende neuronale Netze zu erstellen. Diese stellen dar, wie oft die einzelnen Ereignisklassen beim Klassifizieren mit den jeweils anderen Ereignisklassen verwechselt-, und wie viele richtig zugeordnet werden. Mehr über Wahrheitsmatrizen, speziell über Wahrheitsmatrizen bei BNNs, ist in Kapitel 6 zu lesen. In Abbildung 4.1 ist ein Beispiel einer Wahrheitsmatrix eines BNNs zu sehen.
Eine Methode direkt die Klassifizierung der Testdaten darzustellen, ist das Histogrammieren der von der Softmax-Funktion ausgegebenen Werte für jedes Ausgabeneuron. Ein solches Histogramm zeigt, wie viele Ereignisse jeder Klasse welchem SOFTMAX-Ausgabewert in einem Ausgabeneuron zugeordnet werden. In den Abbildungen B.1 bis B.5 im Anhang sind die Histogramme der Ausgabeneuronen desselben BNNs zu sehen, zu dem die Wahrheitsmatrix in Abbildung 4.1 gehört. Auf der linken Seite ist dort die Verteilung der Mittelwerte für jedes Ausgabeneuron dargestellt, auf der rechten Seite die Verteilung der Standardabweichungen.
Die Diskriminatoren wiederum sind Histogramme, welche die Zuordnung von einzelnen Ereignissen in die verschiedenen Klassen darstellen. Anders als bei den zuvor beschriebenen Histogrammen werden in den Diskriminatoren nur diejenigen Ereignisse gezählt, die auch dem betrachteten Ausgabeneuron zugeordnet wurden. Das bedeutet, dass ein Ereignis nur zu dem Diskriminator des Neurons beiträgt, in dem der Ausgabewert des Netzes am nächsten an Eins ist. Die Diskriminatoren für das Netz, zu dem auch die Wahrheitsmatrix in Abbildung 4.1 gehört sind ebenfalls im Anhang in den Abbildungen C.1 bis C.5 zu finden.

4.3 Methodik und Ziele dieser Arbeit

In dieser Arbeit wird aufbauend auf den binär klassifizierenden BNNs aus [6] und [19] die Verwendung von multiklassifzierenden BNNs für die ttH-Analyse eingeführt. Die Abildungen 4.1 und B.1 bis C.5 zeigen Auswertungsergebnisse eines BNNs, die mithilfe der zur Verfügung stehenden Methoden in DRACO-MLfoy aus Abschnitt 4.2 dargestellt sind. Im weiteren Verlauf dieser Arbeit wird in Kapitel 5 zunächst überprüft, ob die Ausgabe der multiklassifizierenden BNNs wie angenommen einer Normalverteilung entspricht. Anschließend wird dort, wie in Abschnitt 3.2.3 beschrieben, ein Vergleich zwischen multiklassifizierenden BNNs und Ensembles aus multiklassifizierenden ANNs durchgeführt. In Kapitel 6 werden einige neue Auswertungsmethoden, die die Vorhersagenunsicherheit einbeziehen, vorgestellt und diskutiert.

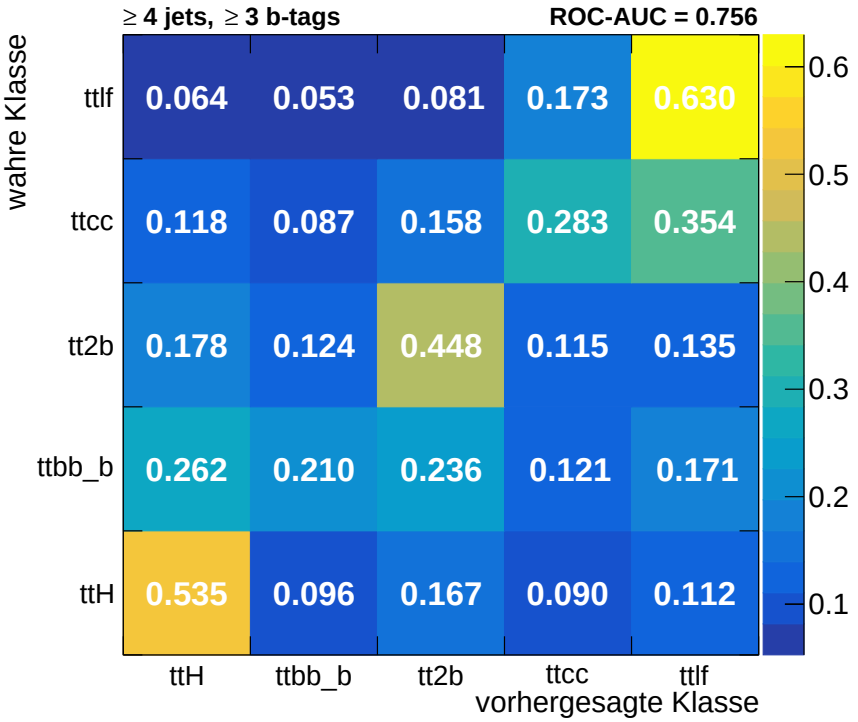

Abbildung 4.1: **Wahrheitsmatrix eines multiklassifizierenden BNNs.** Die wahren Klassen der Daten sind auf der vertikalen Achse aufgetragen, die vorhergesagten Klassen auf der horizontalen Achse. Das Diagramm zeigt die relative Häufigkeit mit der Ereignisse einer bestimmten wahren Klasse einer vorhergesagten Klasse zugeordnet werden. Oben rechts ist der ermittelte ROC-AUC-Wert des Netzes zu sehen. Mit einem Wert von 0.756 ist dieser zwar deutlich über 0.5 aber ebenfalls deutlich unter 1. Es ist erkennbar, dass das Netz besonders bei $t\bar{t}H$ und $t\bar{t} + \text{light flavor}$ die besten Ergebnisse liefert, da bei diesen Kategorien die Diagonalelemente groß sind. Die Klasse $t\bar{t} + b\bar{b}$ wird vom Netz besonders schlecht klassifiziert, da ihre Ereignisse sehr gleichmäßig allen Klassen zugeordnet wurden.

5 Studien zur Verlässlichkeit von multiklassifizierenden BNNs

Der größte Vorteil, den die Nutzung von bayesischen neuronalen Netzen gegenüber herkömmlichen neuronalen Netzen bietet, ist, dass sie ein Maß für die Unsicherheit der von ihnen getroffenen Vorhersagen mitliefern. In der $t\bar{t}H(b\bar{b})$-Analyse werden bisher ANNs verwendet, um Ereignisse zu klassifizieren. In diesem Kapitel wird deshalb untersucht, in wie weit multiklassifizierende BNNs für den Einsatz in der Analyse geeignet sind. Zunächst wird dafür in Abschnitt 5.1 überprüft, ob die Ausgabe des Netzes nach vielfachem Auswerten, wie angenommen, Normalverteilungen entsprechen, welche durch einen Mittelwert und eine Standardabweichung beschrieben werden können. Im Anschluss daran wird in Abschnitt 5.1 ein Vergleich zwischen multiklassifizierenden BNNs und einem Ensemble aus den bisher in der Analyse genutzten ANNs durchgeführt.

5.1 Nachweis der Normalverteilung der Ausgabe

Die Interpretation der Ausgabe der BNNs als Mittelwert und Standardabweichung ist nur sinnvoll, wenn die Verteilung der Ausgabewerte nach mehrfachem Auswerten die Form einer Normalverteilung hat. Innerhalb des Netzes ist durch die A-priori- und A-posteriori-Verteilung sowie die ReLU-Aktivierungsfunktionen der Neuronen, die Normalverteilung der weitergegebenen Daten sichergestellt. In der Ausgabeschicht wird allerdings zuletzt, bevor die Ausgabe der verarbeiteten Daten in den Ausgabeneuronen stattfindet, die nichtlineare Softmax-Funktion (Gleichung (3.4)) angewandt um die Ausgabe in das Wahrscheinlichkeitsintervall $[0; 1]$ zu transformieren. Durch diese nichtlineare Aktivierungsfunktion könnte die Ausgabe des Netzes die Form der Normalverteilung verlieren. Um sicherzustellen, dass die Normalverteilung erhalten bleibt, werden stichprobenartig die Verteilungen von 15 Ereignissen des Testdatensatzes für 2000 Netzauswertungen überprüft.
Um die Verteilung zu erhalten, die jedes der Ausgabeneuronen des Netzes produziert, werden die nach jeder Auswertung erhaltenen Werte in einem Histogramm dargestellt. Die Histogramme aller Ausgabeneuronen eines Ereignisses werden dann gemeinsam in einem Diagramm zusammengefasst, welches die komplette Netzausgabe für selbiges Ereignis darstellt. Zusätzlich werden die mit dem Mittelwert und der Ausgabeverteilungen parametrisierten Normalverteilungen eingezeichnet. Wenn die Form der Histogramme gut mit der Form der eingezeichneten Normalverteilungen übereinstimmt, so ist das ein klares Zeichen dafür, dass der Einfluss der Softmax-Funktion auf die Form der Verteilung vernachlässigbar ist. Das führt wiederum dazu, dass die Netzausgabe durch einen Mittelwert

und eine Standardabweichung vollständig beschrieben werden kann.
In Abbildung 5.1 ist das Diagramm des ersten Ereignisses zu sehen. Da das Maximum des $t\bar{t} + b\bar{b}$-Histogramms im Vergleich mit den anderen Prozessen am nächsten an der Eins liegt, wird dieses Ereignis vom Netz als $t\bar{t} + b\bar{b}$-Ereignis klassifiziert. Im Anhang sind in den Abbildungen D.1 bis D.14 dieselben Diagramme für die anderern 14 Ereignisse zu sehen.

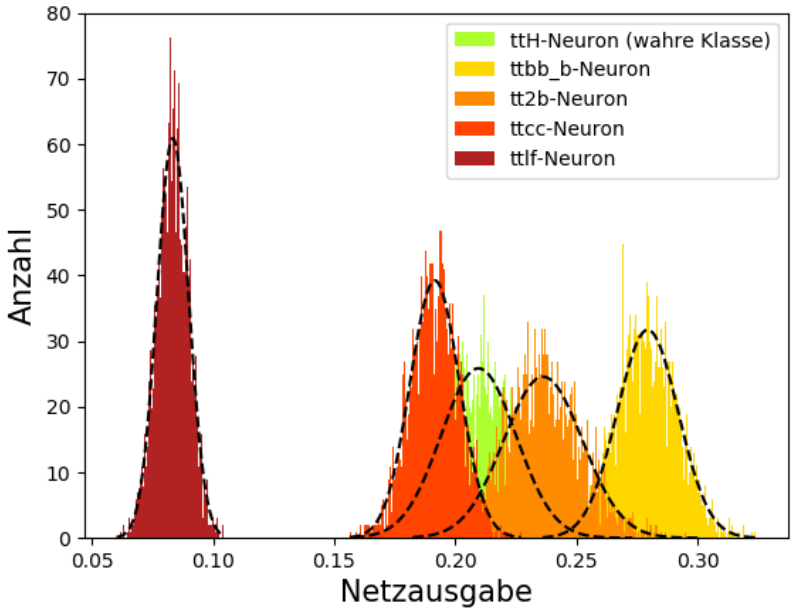

Abbildung 5.1: **Histogramme aller Ausgabeneuronen für das erste Ereignis des Testdatensatzes.** Die Histogramme der einzelnen Verteilungen sind in verschiedenen Farben dargestellt. Die zugehörigen Normalverteilungen sind als Strichlinien über den jeweiligen Histogrammen eingezeichnet. An der Reihenfolge der Verteilungen lässt sich erkennen, dass $t\bar{t} + b\bar{b}$ als der wahrscheinlichste Prozess eingestuft wird, danach folgen in absteigender Reihenfolge $t\bar{t}$ + 2b, $t\bar{t}$H, $t\bar{t} + c\bar{c}$ und $t\bar{t}$ + light flavor. Die Unsicherheit ist am geringsten bei $t\bar{t}$ + light flavor, da die Verteilung am schmalsten ist.

Aus den Abbildungen 5.1 und D.1 bis D.14 geht deutlich hervor, dass die vorhergesagten Normalverteilungen alle gut mit den Histogrammen übereinstimmen. Die parametrisierten Normalverteilungen passen zu den Ausgabeverteilungen. Somit ist die Interpretierbarkeit der Netzausgaben als Mittelwert und Standardabweichung gewährleistet.

Tabelle 5.1: **Hyperparameter der verwendeten ANNs.**

Hyperparameter	Wert
Architektur	1 Eingangsschicht, 25 Neuronen 1 verdeckte Schicht, 150 Neuronen 1 Ausgabeschicht, 5 Neuronen
Batchgröße	5000 Ereignisse
Optimierungsalgorithmus	ADAM
Lernrate	10^{-3}
Aktivierungsfunktion verdeckte Schicht	RELU
Aktivierungsfunktion Ausgabeschicht	SOFTMAX
L2-Parameter	10^{-3}, 10^{-4}, 10^{-5}

5.2 Vergleich eines BNN mit einem Ensemble von ANNs

In Abschnitt 3.2.3 wurde beschrieben, dass ein BNN durch ein Ensemble von ANNs nachgebildet werden kann. In diesem Kapitel wird ein Vergleich zwischen BNNs und ANNs durchgeführt. In [6] wird ein solcher Vergleich für binär klassifizierende neuronale Netze beschrieben, der zum Ergebnis hatte, dass BNNs und ANNs äquivalente Ergebnisse bei der Klassifizierung erzielen. Hier wird nun ein Vergleich von multiklassifizierenden neuronalen Netzen durchgeführt.
Für das BNN wird die in Abschnitt 4.1 beschriebene Standard-Netzkonfiguration genutzt. Die verwendeten BNNs haben pro Gewicht drei verschiedene trainierbare Parameter. Da die ANNs pro Gewicht über nur einen trainierbaren Parameter verfügen, werden in der verdeckten Schicht der ANNs 150 Neuronen verwendet. Somit wird erreicht, dass beide Netzarchitekturen die gleiche Anzahl an trainierbaren Parametern haben. Um den Mittelwert und die Standardabweichung der Ausgabeverteilungen des BNN zu ermitteln wird dieses 100 Mal ausgewertet. Um dieselbe statistische Repräsentativität bei dem ANN-Ensemble zu erreichen, besteht dieses aus 100 einzeln trainierten ANNs, die alle einmal ausgewertet werden. Der Mittelwert und die Standardabweichung des Ensembles werden dann aus den 100 verschiedenen Ausgaben berechnet. Die restlichen Hyperparameter der Netze sind gleich gewählt, sodass die neuronalen Netze gut vergleichbar sind. Die Gesamtheit der verwendeten Hyperparameter des BNNs ist in Tabelle 4.3 in Kapitel 4 zu finden. Die verwendeten Hyperparameter der ANNs sind in Tabelle 5.1 aufgelistet. Wie in [6] werden auch hier die L2 Parameter $\lambda = (10^{-3}, 10^{-4}, 10^{-5})$ verwendet, um den KL-Term aus Abschnitt 3.2.2 mit verschiedenen L2-Normen vergleichen zu können. Die BNN-Ausgabe, die für den Vergleich genutzt wird, ist die des BNNs, welches in Abschnitt 4.2 bereits gezeigt wurde.
Zunächst werden die erhaltenen ROC-AUC-Werte der Netze verglichen. Das BNN erzielt einen ROC-AUC-Wert von 0,756, siehe Abbildung 4.1 in Abschnitt 4.2. In der Abbildung 5.2 sind die Wahrheitsmatrizen der drei ANN-Ensembles zu finden. In diesen Abbildungen ist auch der jeweilige ROC-AUC-Wert des Ensembles eingetragen. Die ROC-AUC-Werte der verschiedenen ANN-Ensembles sind mit 0,761, 0,765 und 0,761 nur marginal besser als der des BNNs. Das spricht für eine ähnliche Klassifizierungsleistung. Die erhaltenen Wahrheitsmatrizen selbst sind ebenfalls alle sehr ähnlich zueinander. Es kommt beim BNN und den ANN-Ensembles gleichermaßen zu einer guten Klassifizierung von $t\bar{t}$ + light flavor-Ereignissen und $t\bar{t}H$-Ereignissen, aber zu vielen Verwechslungen von Ereignissen mit der

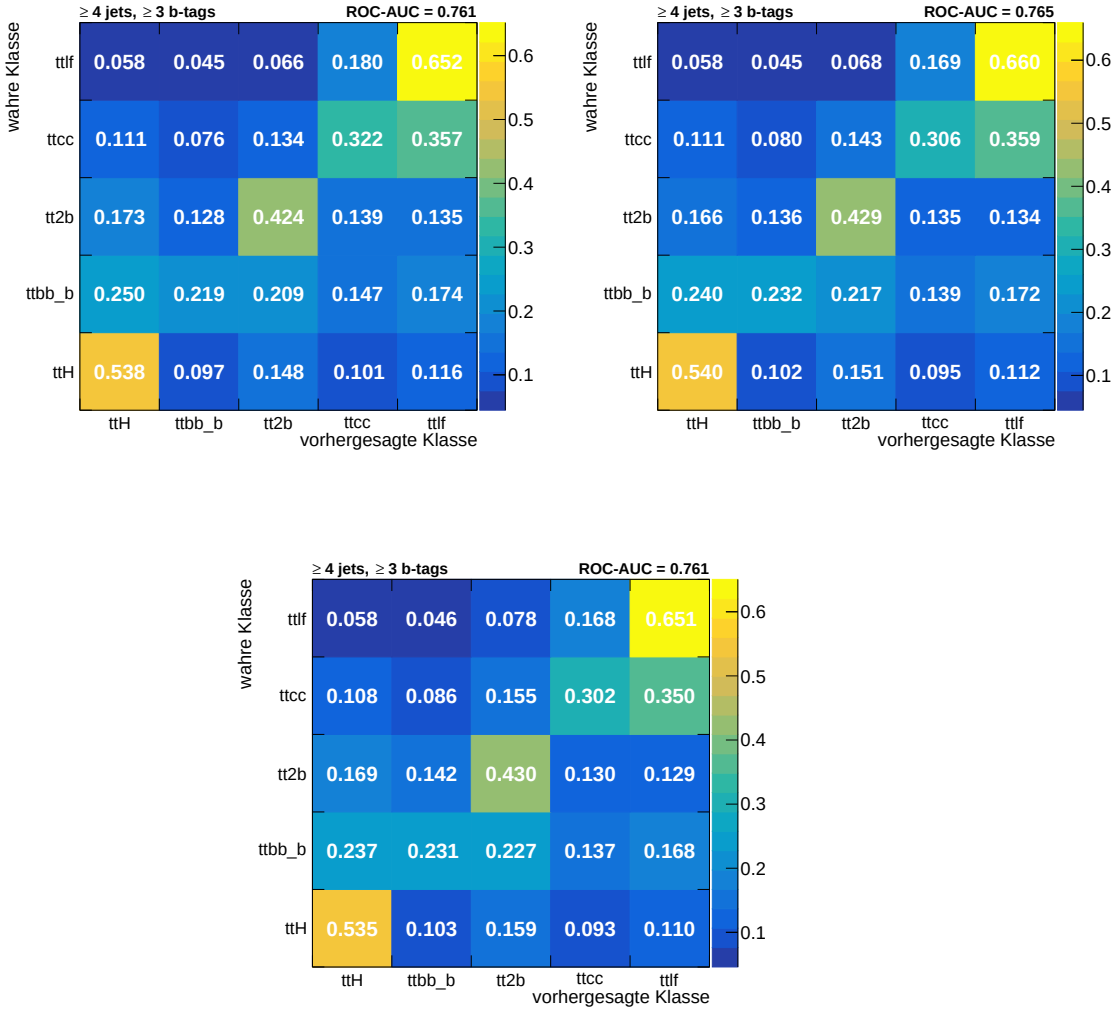

Abbildung 5.2: **Wahrheitsmatrizen der drei ANN-Ensembles** Die Wahrheitsmatrix oben links gehört zum Ensemble mit L2-Parameter 10^{-3}, oben rechts zu 10^{-4} und unten zu 10^{-5}.

wahren Klasse $t\bar{t} + b\bar{b}$ und $t\bar{t} + c\bar{c}$.
Um einen genaueren Vergleich zwischen dem BNN und den ANN-Ensembles zu erhalten werden wie in [6] 2D-Histogramme erstellt, welche die Korrelation der Vorhersagen von BNN und ANN-Ensembles zeigen. In den 2D-Histogrammen sind die erhaltenen Mittelwerte bzw. Standardabweichungen jeweils eines Ausgabeneurons für alle Ereignisse des Testdatensatzes dargestellt. Je mehr Werte auf der Winkelhalbierenden liegen, desto mehr stimmen die Vorhersagen von BNN und ANN-Ensemble überein. Die erstellten 2D-Histogramme des $t\bar{t}H$-Neurons sind in Abbildung 5.3 zu sehen. Die Histogramme für die anderen Neuronen sind im Anhang in den Abbildungen E.1 bis E.4 zu sehen.

In Abbildung 5.3 ist zu erkennen, dass die Korrelation der Mittelwerte bei einem L2-Parameter von 10^{-3} enger an der Winkelhalbierenden bleibt, als bei den L2-Parametern 10^{-4} und 10^{-5}. Die Korrelation der Standardabweichungen sind bei einem L2-Parameter von 10^{-3} in allen Neuronen sehr stark unterhalb der Winkelhalbierenden ausgeprägt. Das bedeutet, dass das ANN-Ensemble mit dieser L2-Norm Verteilungen, die im Vergleich zu

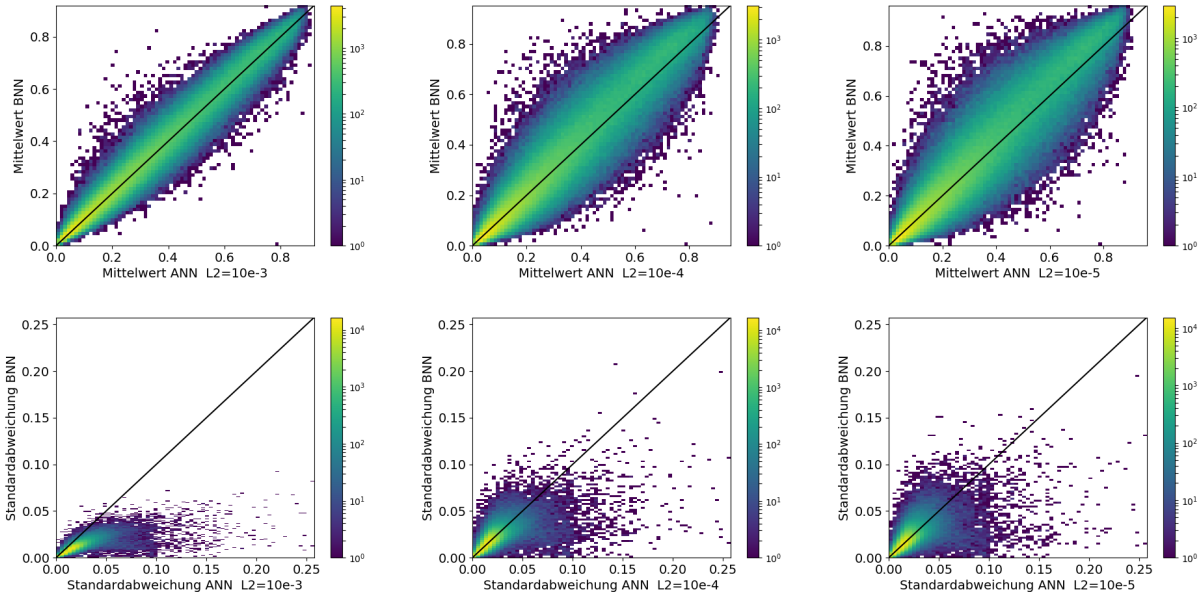

Abbildung 5.3: **Korrellation der Vorhersagen von BNN und ANN-Ensembles des $t\bar{t}H$-Neurons.** Die vorhergesagten Werte für Mittelwert und Standardabweichung aus dem BNN sind gegen die vorhergesagten Mittelwerte und Standardabweichungen des ANN-Ensembles aufgetragen. Zusätzlich ist die Winkelhalbierende (schwarz) eingezeichnet. Die oberen drei Histogramme stellen jeweils die Korrelation der Mittlwerte dar, während die unteren drei die Korrelation der Standardabweichungen darstellen. Die beiden linken Histogramme nutzen die Werte des ANNs mit dem L2-Parameter $\lambda = 10^{-3}$, die beiden mittleren $\lambda = 10^{-4}$ und die beiden rechten $\lambda = 10^{-5}$.

denen des BNNs, hohe Standardabweichungen haben, ausgibt. Die 2D-Histogramme der Standardabweichungen bei L2-Parametern von 10^{-4} und 10^{-5} haben für alle Ausgabeneuronen ähnliche Formen. Sie sind nicht so einseitig ausgeprägt, wie die Verteilung bei $\lambda = 10^{-3}$, sondern insgesamt symmetrischer um die Winkelhalbierende verteilt. Bei Betrachtung der einzelnen Neuronen ist das allerdings nicht so. Bei den $t\bar{t}$ + 2b- und $t\bar{t} + c\bar{c}$-Neuronen ist die vom BNN ausgegebene Standardabweichung bei $\lambda = (10^{-4}, 10^{-5})$ höher.
Im Falle der binären Klassifikation in [6] haben das ANN-Ensemble und das BNN für $\lambda = 10^{-4}$ die ähnlichsten Auswertungsergebnisse geliefert. Im Falle der Multiklassifikation in fünf Klassen lässt sich aus den Korrelationsplots folgern, dass es nicht vernachlässigbare Unterschiede gibt. Ein Beispiel hierfür ist die große Streuung der Unsicherheiten, die in den 2D-Histogrammen der Standardabweichung zu erkennen ist. Trotzdem werden insgesamt von den Netzen ähnliche Ergebnisse erzielt, was an der Ähnlichkeit der Wahrheitsmatrizen zu erkennen ist. Um herauszufinden, wie sich die erhaltenen Unterschiede der Netze auf die Endresultate auswirken, müsste das BNN bzw. die ANNs in einem Maximum-Likelihood-Fit an Daten verglichen werden, was im Rahmen dieser Arbeit allerdings nicht stattfinden konnte.
In Abbildung E.3 lässt sich eine Besonderheit der Ausgabe des ANN-Ensembles feststellen. Die Histogramme sind dort ab einem ANN-Mittelwert von etwa 0,45 nach oben abgeschnitten, während die vom BNN ausgegebenen Mittelwerte bis zu 0,7 erreichen.
Zusammenfassend lässt sich bemerken, dass bei L2-Parametern von 10^{-4} und 10^{-5} ANN-Ensembles und BNN für $t\bar{t}H$, $t\bar{t} + b\bar{b}$ und $t\bar{t}$ + light flavor ähnliche Unsicherheiten liefern. Für Klassen $t\bar{t}$ + 2b und $t\bar{t} + c\bar{c}$ liefert das BNN größere Unsicherheiten. Bei einem L2-Parameter von 10^{-3} tendiert das BNN zu kleineren Unsicherheiten. Im Vergleich mit den

binär klassifizierenden neuronalen Netzen aus [6] fällt auf, dass die 2D-Verteilungen, die bei Multiklassifikation auftreten, weiter um die Winkelhalbierende streuen als bei binärer Klassifikation. Das liegt daran, dass das Netz mehr Optionen hat, die eingegebenen Daten einzuteilen. Deshalb liegen die maximal erreichten Mittelwerte bei der Multiklassifikation auch im Bereich [0,8; 0,9], während bei binärer Klassifikation auch häufiger Ausgabewerte sehr nahe der Eins auftreten können.
Da das BNN im Vergleich ähnlich gute Voraussagen liefert wie ANN-Ensembles, ist es auch bei Multiklassifikation sinnvoller BNNs einzusetzen, wenn eine Unsicherheit auf die Vorhersagen gewünscht ist. Ein BNN zu trainieren und 100 mal auszuwerten nimmt etwa genau so viel Zeit in Anspruch, wie zwei ANNs zu trainieren und jeweils einmal auszuwerten. Das bedeutet, das für ähnliche statistische Repräsentativität bei ANN-Ensembles etwa die 50-fache Rechenzeit benötigt wird, als bei einem einzelnen BNN, das 100 mal ausgewertet wird.

6 Klassenmigration aufgrund von Vorhersagenunsicherheiten

Wie schon zuvor beschrieben, ist die große Stärke von bayesischen neuronalen Netzen eine Unsicherheit zu den getroffenen Vorhersagen mitzuliefern. Die bisher genutzten Auswertungsmethoden aus Abschnitt 4.2 nutzen diese allerdings nicht. In diesem Kapitel wird eine Möglichkeit vorgestellt und diskutiert, wie die erhaltenen Unsicherheiten in Auswertungsmethoden eingebunden werden können.
In Abschnitt 6.1 geht es zunächst darum, wie die Unsicherheit in die Vorhersage eingebracht wird. In Abschnitt 6.2 wird die Migration von Ereignissen durch die Betrachtung des ersten Sigmaintervalls der Unsicherheiten untersucht. Anschließend wird in Abschnitt 6.3 eine Erweiterung der in Abschnitt 4.2 beschriebenen Wahrheitsmatrizen eingeführt, die die Unsicherheiten der Vorhersagen miteinbezieht. Alle Abbildungen in diesem Kapitel sind mit der Auswertung desselben BNN-Trainings wie die Abbildungen aus Abschnitt 4.2 erstellt.

6.1 Nutzung von Unsicherheiten in der Auswertung

In Kapitel 5.1 wurde nachgewiesen, dass die Ausgaben der Ausgabeneuronen Normalverteilungen entsprechen. Die Vorhersagen des Netzes nutzen, was Klassifizierung der Ereignisse angeht, nur die Mittelwerte der erhaltenen Normalverteilungen, indem diese verglichen werden. Die Klasse, deren ausgegebener Mittelwert μ_m am nächsten an Eins ist, bekommt das Ereignis zugeordnet. Mehr Informationen über die Auswertung von BNNs sind in Abschnitt 3.2.3 zu finden. Klassischerweise werden in der Physik die Werte von Größen mit Unsicherheiten als Mittelwert und Standardabweichung angegeben. Dasselbe Konzept wird nun verwendet, um die Unsicherheiten in die Klassifizierung einzubringen. Dementsprechend wird, bevor die Klassifizierung basierend auf den ausgegebenen Mittelwerten stattfindet, die jeweils erhaltene Unsicherheit σ_m zum erhaltenen Mittelwert addiert

$$\mu_m^+ = \mu_m + \sigma_m, \tag{6.1}$$

bzw. von diesem subtrahiert

$$\mu_m^- = \mu_m - \sigma_m. \tag{6.2}$$

Die so erhaltenen neuen Werte $\mu_m^\pm$ werden dann zur Klassifizierung des Ereignisses genutzt. Der Bereich zwischen μ_m^+ und μ_m^+ ist das erste Sigmaintervall. Da die Standardabweichungen verschiedener Ausgabeneuronen im Allgemeinen unterschiedlich groß sind, kann sich durch

die Addition bzw. Subtraktion der Standardabweichung die vorhergesagte Klasse eines Ereignisses ändern. Wie die Änderung der vorhergesagten Klasse zustande kommt, ist in Abbildung 6.1 veranschaulicht.

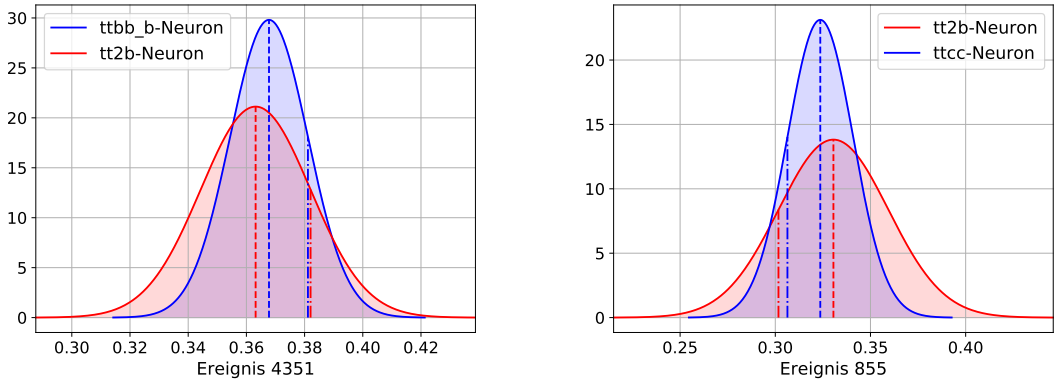

Abbildung 6.1: **Verteilungen der Neuronen mit den beiden größten Mittelwerten zweier Ereignisse des Testdatensatzes.** Es sind jeweils die Verteilungen der Neuronen mit den beiden höchsten Mittelwerten zu sehen. Die jeweiligen μ_m sind durch Strichlinien dargestellt. Die nach der Addition bzw. Subtraktion erhaltenen $\mu_m^{\pm}$ sind durch Strichpunktlinien dargestellt. Sie stellen gleichzeitig die Grenzen des Sigmaintervalls nach rechts bzw. links dar.

Auf der linken Seite von Abbildung 6.1 sind die Verteilungen der $t\bar{t} + b\bar{b}$- und $t\bar{t}$ + 2b-Neuronen eines Ereignisses zu sehen, welches seine vorhergesagte Klasse bei der Addition von einer Standardabweichung, siehe Gleichung 6.1, ändert. Wenn nur die Mittelwerte betrachtet werden, ist deutlich erkennbar, dass das Ereignis aufgrund des größten Mittelwerts als $t\bar{t} + b\bar{b}$ klassifiziert wird. Das rechte Sigmaintervall des $t\bar{t}$ + 2b-Neurons schließt das rechte Sigmaintervall des $t\bar{t} + b\bar{b}$-Neurons vollständig ein. Das führt dazu, dass bei Addition der Standardabweichung für das $t\bar{t}$ + 2b-Neuron ein höherer Wert zur Klassifikation verwendet wird als für das $t\bar{t} + b\bar{b}$-Neuron, wodurch sich die vorhergesagte Klasse ändert. Im rechten Diagramm sind die Verteilungen der $t\bar{t}$ + 2b und $t\bar{t} + c\bar{c}$-Neuronen eines anderen Ereignisses zu sehen, dessen Klasse sich, bei Subtraktion einer Standardabweichung, siehe Gleichung 6.2, ändert. Bei diesem Ereignis schließt das linke Sigmaintervall des $t\bar{t}$ + 2b-Neurons das linke Sigmaintervall des $t\bar{t} + c\bar{c}$-Neurons vollständig ein. Deshalb erhält man nach der Subtraktion einer Standardabweichung für $t\bar{t}$ + 2b einen kleineren Wert als für $t\bar{t} + c\bar{c}$, was zur Änderung der vorhergesagten Klasse von $t\bar{t}$ + 2b nach $t\bar{t} + c\bar{c}$ führt.
Die Änderung der Klasse eines Ereignisses durch die Addition bzw. Subtraktion einer Standardabweichung wird im Folgenden auch als Migration eines Ereignisses bezeichnet.

6.2 Migration zwischen Ereignisklassen

Migration kann bei einem Ereignis nur stattfinden, wenn das Sigmaintervall der Vorhersage eines Neurons auf der linken oder rechten Seite vollständig von dem der Vorhersage eines anderen Neurons eingeschlossen wird. Das kann entweder der Fall sein, wenn das Netz zwei Klassen mit ähnlicher Standardabweichung als ähnlich wahrscheinlich einstuft, oder wenn sehr breite Verteilungen und sehr schmale Verteilungen in der Ausgabe eines Ereignisses vorkommen.
Um zu überprüfen, wie oft es zu Migration von Ereignissen kommt, werden zunächst die

Diskriminatoren, die μ_m zur Klassifizierung nutzen, mit den Diskriminatoren, die $\mu_m^{\pm}$ zur Klassifizierung nutzen, verglichen. Die Abbildungen der unveränderten Diskriminatoren sind im Anhang in den Abbildungen C.1 bis C.5 zu sehen. Die Diskriminatoren mit addierter bzw. subtrahierter Standardabweichung sind ebenfalls im Anhang in den Abbildungen F.1 bis F.5 zu finden.
Die Diskriminatoren weisen im Vergleich nur marginale Unterschiede auf. Daraus lässt sich schließen, dass Migrationen von Ereignissen nicht häufig auftreten. Um genaue Informationen darüber zu erhalten, wie viele Ereignisse des Testdatensatzes migrieren, und welche Klassenänderungen vermehrt auftreten, wurden zu den bekannten Auswertungsmethoden aus Abschnitt 4.2 neue Diagramme hinzugefügt, die Migration von Ereignissen in BNNs detailliert zeigen. Die Migrationsdiagramme bestehen aus einer Säule links, in der der Name der betrachteten Klasse steht. Rechts sind durch kleinere Blöcke die anderen Ereignisklassen symbolisiert. In der Mitte sind pro kleinem Block zwei Pfeile dargestellt. Die Zahlen in den Pfeilen stehen für die Anzahl der Ereignisse, die von der Klasse links in die Klasse rechts übergehen bzw. umgekehrt. Die Übergagsanzahlen sind dabei nicht auf die selbe Luminosität normiert, es werden lediglich die einzelnen Migrationen gezählt. In Abbildung 6.2 sind solche Migrationsdiagramme für das $t\bar{t}H$-Neuron dargestellt. Die Migrationsdiagramme der anderen Klassen sind im Anhang in den Abbildungen G.1 bis G.4 zu finden. Da ein Diagramm pro Ausgabeneuron erzeugt wird, sind die Werte für Migration zwischen zwei Klassen immer in zwei Diagrammen zu finden.

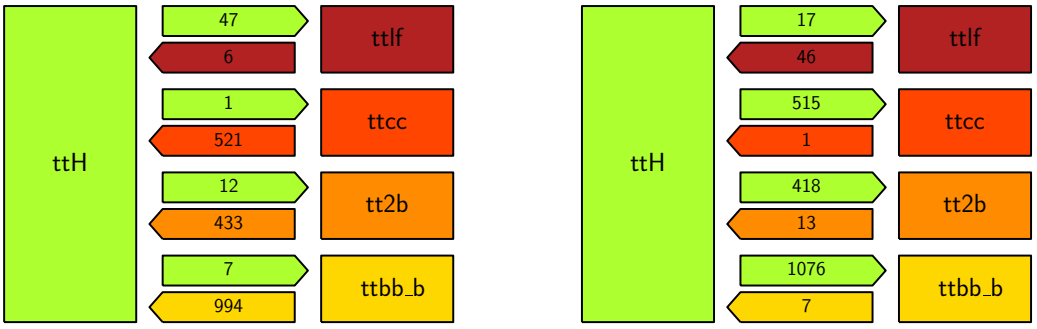

Abbildung 6.2: **Migrationsdiagramme für die $t\bar{t}H$-Klasse.** Das Diagramm links zeigt die Migration bei Addition einer Standardabweichung. Rechts ist das Migrationsdiagramm für die Subtraktion einer Standardabweichung zu sehen.

Durch Aufaddieren der einzelnen migrierten Ereignisse lässt sich ermitteln, das bei der Addition insgesamt 4679 Ereignisse des Testdatensatzes migriert sind. Das entspricht 1,31% aller Testereignisse. Durch Subtraktion einer Standardabweichung migrieren 1,37% der Testereignisse. Dieser geringe Anteil verdeutlicht, dass die Diskriminatoren in Anhang C (Abbildungen C.1 bis C.5) sich kaum von den Diskriminatoren in Anhang F (Abbildungen F.1 bis F.5) unterscheiden. Es ist leicht zu erkennen, dass beim Übergang zwischen zwei Klassen immer eine Migrationsrichtung dominant ist. Ebenso ist in den Migrationsdiagrammen aller Klassen zu erkennen, dass die Subtraktion einer Standardabweichung eine Migration ähnlicher Größe erzeugt wie die Addition einer Standardabweichung, allerdings die Migrationsrichtung umgekehrt ist. Das ist in Abbildung 6.2 beispielsweise beim Übergang von $t\bar{t}H$ nach $t\bar{t}$ + 2b sehr eindeutig: Bei der Addition gehen 433 $t\bar{t}$ + 2b-Ereignisse zur $t\bar{t}H$-Klasse über und nur zwölf umgekehrt, wohingegen bei der Subtraktion 418 $t\bar{t}H$ Ereignisse zu $t\bar{t}$ + 2b übergehen, und nur 13 in die andere Richtung migrieren. Dieser

Trend ist in den Migrationsdiagrammen aller Klassen erkennbar.
Um einen noch besseren Überblick über die Gesamtmigration zu bekommen, die zwischen den Klassen stattfindet, wird ein Sankey-Diagramm verwendet [34]. Sankey-Diagramme stellen im Allgemeinen Mengenflüsse dar und kommen deshalb beispielsweise bei der Analyse von Wählerwanderung in der Politikwissenschaft zum Einsatz. In dieser Arbeit werden sie verwendet, um die Migration durch Addition oder Subtraktion einer Standardabweichung zwischen allen Klassen in einem Diagramm kompakt darzustellen. In Abbildung 6.3 ist das Sankey-Diagramm für die Migration durch Addition zu sehen, in Abbildung 6.4 das für Migration durch Subtraktion.
Da sowohl durch Addition als auch durch Subtraktion einer Standardabweichung nur ein kleiner Bruchteil der Testereignisse die vorhergesagte Klasse wechselt, sind in den Sankey-Diagrammen nur die Ereignisse dargestellt, die ihre vorhergesagte Klasse wechseln. Sowohl die Gesamtanzahl als auch die Anzahl von migrierten Ereignissen, sind unter dem eigentlichen Diagramm eingetragen.
Das Sankey-Diagramm besteht aus Blöcken links und rechts, die die Klassen symbolisieren. Die Höhe dieser Blöcke ist proportional zu dem Anteil der Klasse an den migrierten Ereignissen. Die linke Seite stellt die Verteilung der klassifizierten Ereignisse vor der Addition bzw. Subtraktion der Standardabweichung dar. Die rechte Seite stellt die Verteilung der migrierten Ereignisse nach der Migration dar. Die prozentualen Anteile der migrierten Ereignisse der Klassen an der Gesamtanzahl der Testereignisse sind in den jeweiligen Blöcken als Prozentsätze zu lesen. Die Verbindungskurven zwischen den Blöcken symbolisieren, wie viele Ereignisse von einer Klasse zur anderen migrieren. Die Dicke der Kurven ist dabei proportional zur Anzahl der Ereignisse, deren Migration sie darstellt.
Die Sankey-Diagramme stellen die Information der Migrationsdiagramme zusammengefasst dar. Die beiden größten Übergänge finden sowohl bei der Addition, als auch bei der Subtraktion einer Standardabweichung zwischen den $t\bar{t} + c\bar{c}$ und $t\bar{t}$ + light flavor-Klassen und den $t\bar{t}H$ und $t\bar{t} + b\bar{b}$-Klassen statt. Das passt gut zu der Wahrheitsmatrix in Abbildung 4.1, die zeigt, dass diese Prozesse oft verwechselt werden. Deshalb sind für diese Prozesse bei einigen Ereignissen ähnliche Vorhersagen zu erwarten. Die Klasse $t\bar{t}$ + 2b ist die einzige Klasse, bei der die abgehenden Ereignisse und die neu hinzukommenden Ereignisse, sowohl für Addition, als auch für Subtraktion einer Standardabweichung, nicht völlig andere Größenordnungen haben, was bedeutet, dass die $t\bar{t}$ + 2b Klasse durch Migration ähnlich viele Ereignisse verliert, wie sie neue dazu bekommt.
Aus den Sankey-Diagrammen in den Abbildungen 6.3 und 6.4 lässt sich ebenfalls folgern, dass die Klassenänderungen durch Addition einer Standardabweichung grob entgegengesetzt zu den Klassenänderungen durch Subtraktion einer Standardabweichung sind. Das lässt sich daran erkennen, dass die beiden Sankey-Diagramme durch vertikale Spiegelung eine Form annehmen, die sehr ähnlich zum jeweils anderen, ungespiegelten Diagramm ist. Der genaue Grund für diese Symmetrie konnte in dieser Arbeit nicht herausgearbeitet werden. Eine detaillierte Betrachtung der Vorhersagen der migrierten Ereignisse im Rahmen zukünftiger Arbeiten könnte mehr Erkenntnisse darüber bringen, was der genaue Grund für die entgegengesetzte Migration durch Addition und Subtraktion von einer Standardabweichung ist.

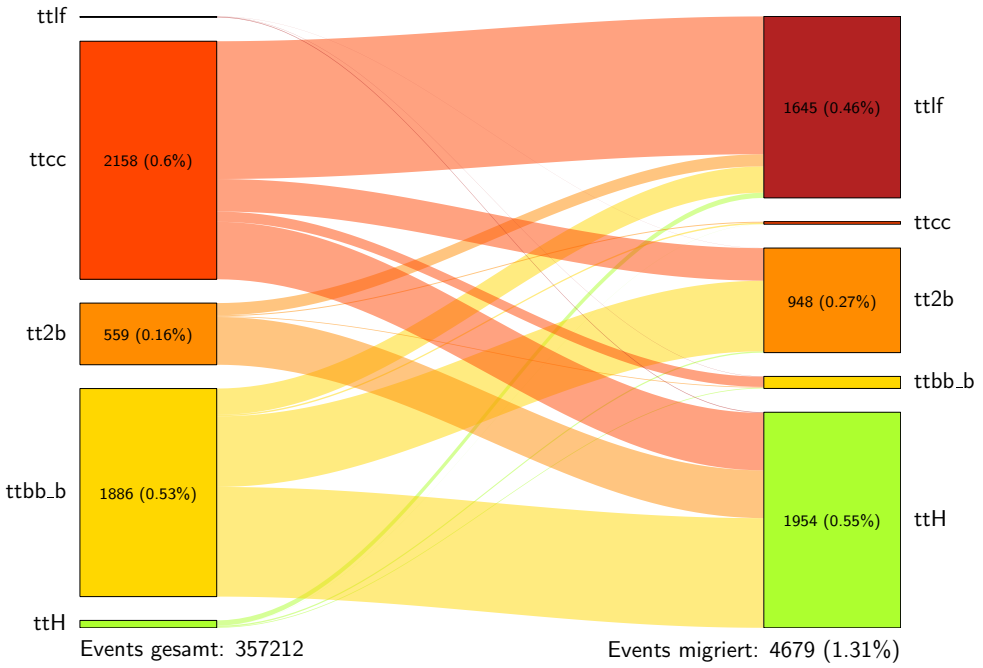

Abbildung 6.3: **Sankey-Diagramm der Migration durch Addition einer Standardabweichung.** Das Diagramm zeigt nur die Verteilungsänderung der 4679 durch Addition einer Standardabweichung migrierenden Ereignisse. Alle Ereignisse des Testdatensatzes, die nicht migrieren sind nicht dementsprechend dargestellt. Die Richtung der Migration im Diagramm ist von links (alte Verteilung) nach rechts (neue Verteilung). Die Prozentzahlen im Diagramm beziehen sich auf den Anteil am Testdatensatz.

6.3 Wahrheitsmatrix mit Unsicherheiten

In Abschnitt 4.2 wurde die Wahrheitsmatrix eingeführt. Sie stellt die relativen Häufigkeiten dar, mit der das neuronale Netz Ereignisse einer wahren Klasse einer bestimmten vorhergesagten Klasse zuordnet. Die in diesem Kapitel ausführlich diskutierte Migration von Ereignissen kann nun verwendet werden, um eine Unsicherheit der relativen Häufigkeiten der Wahrheitsmatrix anzugeben.
Um die Unsicherheiten zu bestimmen, werden zunächst alle Ereignisse ermittelt, die durch Addition einer Standardabweichung migrieren. Von diesen Ereignissen sind die wahre Klasse, die vorhergesagte Klasse vor der Migration und die vorhergesagte Klasse nach der Migration bekannt. In der herkömmlichen Wahrheitsmatrix ohne Einbeziehen der Standardabweichung entspricht jedes Feld in der Matrix einer möglichen Kombination aus wahrer Klasse und vorhergesagter Klasse. Es wird eine Matrix erstellt, die die gleichen Dimensionen wie die eigentliche Wahrheitsmatrix hat und mit Nullen gefüllt ist. Ihre Einträge stehen an den selben Stellen für die gleiche Kombination von wahrer Klasse und vorhergesagter Klasse. Im Falle dieser Arbeit entspricht die neue Matrix einer 5×5-Matrix. Für jedes Ereignis, das migriert, wird in der neuen Matrix von dem Eintrag, dem dieses Ereignis ursprünglich zugeordnet wurde, eine Eins subtrahiert. Ebenfalls wird zu dem Eintrag dem das Ereignis durch die Migration zugeordnet wird eine Eins addiert. Wenn der Prozess für alle migrierten Ereignisse stattgefunden hat, wird die neue Matrix zeilenweise

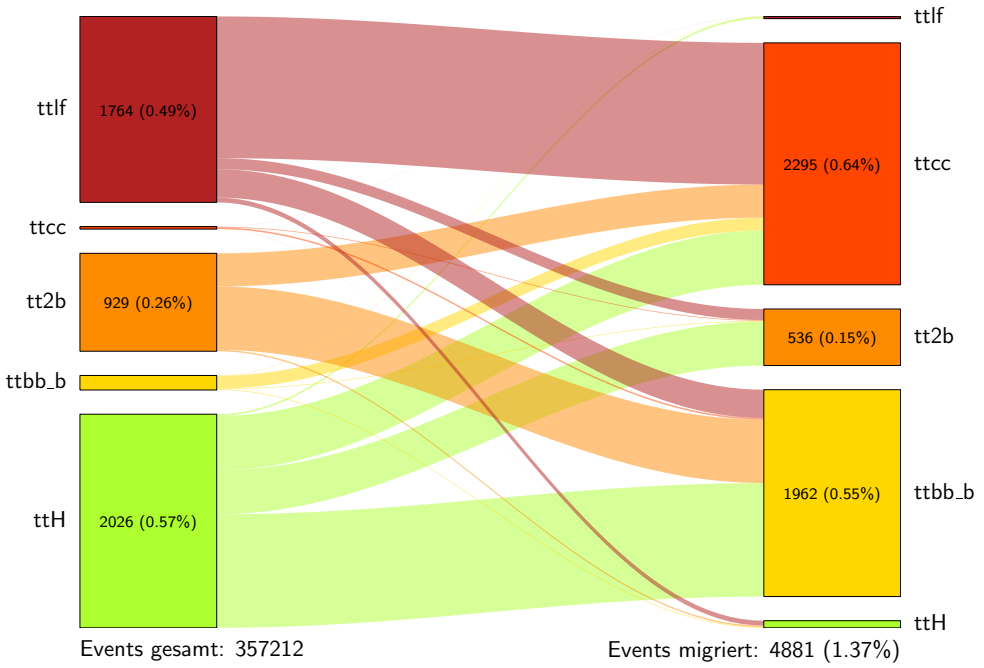

Abbildung 6.4: **Sankey-Diagramm der Migration durch Subtraktion einer Standardabweichung.** Das Diagramm zeigt die Verteilungsänderung der 4881 durch Subtraktion einer Standardabweichung migrierenden Ereignisse. Die Richtung der Migration im Diagramm ist wie in Abbildung 6.3 von links nach rechts.

normiert, sodass die Anzahlen der migrierten Ereignissen zu relativen Häufigkeiten der jeweiligen wahren Klasse umgeformt werden. Um die Unsicherheiten durch Subtraktion einer Standardabweichung zu ermitteln, wird die selbe Prozedur für die durch Subtraktion migrierenden Ereignisse wiederholt. Die Zahlen, die in den neuen Matrizen stehen, entsprechen nun den Veränderungen der relativen Häufigkeiten in den Feldern, die durch die Migration bei der Addition bzw. Subtraktion von einer Standardabweichung zustande kommen.

Die beiden erhaltenen Matrizen werden zusammen mit der ursprünglichen Wahrheitsmatrix dargestellt. In Abbildung 6.5 ist die Wahrheitsmatrix mit Unsicherheiten zu sehen. Der obere Wert in jedem Feld entspricht dabei der relativen Häufigkeit in der ursprünglichen Wahrheitsmatrix in Abbildung 4.1. Die beiden unteren Werte jedes Feldes sind die errechneten Veränderungen durch Migration, wobei durch ein großes Plus die Veränderung durch Addition, und durch ein großes Minus die Veränderung durch Subtraktion einer Standardabweichung dargestellt ist. Wenn hinter dem großen Plus bzw. Minus ein kleines negatives Vorzeichen der Veränderung der relativen Häufigkeit vorangestellt ist, so verkleinert sich diese durch die Migration. Wenn ihr kein Vorzeichen vorangestellt ist, so vergrößert sich die relative Häufigkeit durch Migration. Die relativen Häufigkeiten liegen somit durch die Einbeziehung einer Standardabweichung in dem Sigma-Bereich, den die Veränderungen ausgehend vom oberen, ursprünglichen Wert aus aufspannen.

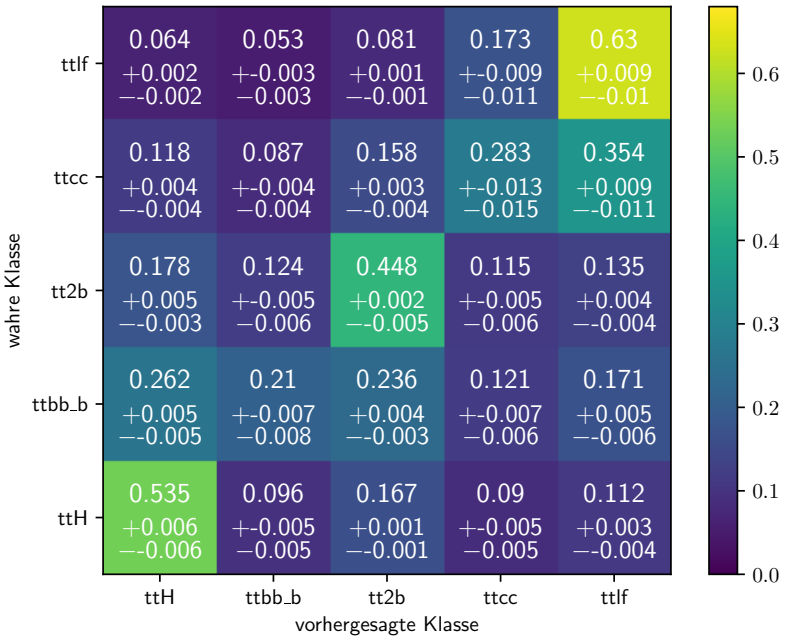

Abbildung 6.5: **Die Wahrheitsmatrix mit den Veränderungen der relativen Häufigkeiten durch zusätzliches Betrachten einer Vorhersagenunsicherheit.** Bei jedem Eintrag wird der Bereich der Unsicherheit zwischen einem Wert kleiner und einem Wert größer der ursprünglichen relativen Häufigkeit aufgespannt. Die Änderungen der relativen Häufigkeit durch Addition und Subtraktion sind betragsmäßig in allen Feldern nah aneinander. Sie haben aber immer unterschiedliche Vorzeichen.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden erfolgreich multiklassifizierende bayesische neuronale Netze eingeführt, die zur Klassifizierung von teilchenphysikalischen Prozessen in der $t\bar{t}H$-Analyse zum Einsatz kommen können. Sie bringen, im Vergleich zu herkömmlichen neuronalen Netzen, den großen Vorteil mit sich, Aussagen über die Unsicherheiten der getroffenen Vorhersagen zu machen.

Die Form der Ausgabe der eingeführten Netze wurde überprüft, um sicherzustellen, dass die mit Mittelwert und Standardabweichung parametrisierten Normalverteilungen, die im Netz genutzt werden, nicht durch die nichtlineare Softmax-Funktion in ihrer Form verändert werden, sodass die Ausgabe des Netzes auch vollständig durch einen Mittelwert und eine Standardabweichung beschrieben werden kann.

Weiterhin wurde durch Vergleich der ROC-AUC-Werte und der Wahrheitsmatrizen gezeigt, dass multiklassifizierende BNNs bei der Klassifizierung vergleichbare Ergebnisse liefern wie ANNs, die bisher als Standardwerkzeug in der $t\bar{t}H$-Analyse zum Einsatz kommen. Ebenfalls wurde bestätigt, dass die Berechnung von Unsicherheiten von Vorhersagen mittels BNNs ähnlich gut, aber wesentlich zeiteffizienter ist, als ein ganzes ANN-Ensemble zu trainieren.

Ebenfalls wurde die durch Einbeziehung der Unsicherheiten entstehende Migration von Ereignissen diskutiert. Mit Hilfe der Migration von Ereignissen wurde die Möglichkeit implementiert, eine Wahrheitsmatrix zu erzeugen, die nicht nur die relative Häufigkeit der richtig und falsch klassifizierten Ereignisse darstellt, sondern auch die Unsicherheiten dieser relativen Häufigkeiten zeigt.

Auf dieser Arbeit aufbauend können noch weitere Studien und Auswertungsmethoden mit BNNs untersucht werden, die die Migration von Ereignissen nutzen. Ein Beispiel für eine Möglichkeit dafür wäre die Nutzung von Migration um die Unsicherheit des ROC-AUC-Wertes zu bestimmen. Ebenfalls sollte der Grund für die Symmetrie, die bei der Addition und Subtraktion von Standardabweichungen in der Anzahl der migrierten Ereignisse auftritt, untersucht werden. Im Rahmen der Bachelorarbeit [19] wurden aufbauend auf der Masterarbeit [6] größere Netzarchitekturen für binär klassifizierende BNNs getestet. Für die multiklassifizierenden BNNs, die in dieser Arbeit eingeführt wurden, besteht ebenfalls die Möglichkeit im Rahmen weiterführender Studien größere Netzarchitekturen zu testen.

Literatur

[1] J. Honerkamp. *Die Vorsokratiker und die moderne Physik: Vom Wesen und Werden einer strengen Wissenschaft*. Berlin [u.A.]: Springer, 2020.

[2] S. Chatrchyan et al. „Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC“. In: *Physics Letters B* 716.1 (Sep. 2012), S. 30–61. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.021. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.021.

[3] G. Aad et al. „Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC“. In: *Physics Letters B* 716.1 (Sep. 2012), S. 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.020.

[4] LHC Higgs Cross Section Working Group. *Handbook of LHC Higgs cross sections: 4. Deciphering the nature of the Higgs sector*. Bd. 2. CERN Yellow Reports: Monographs, 2017. DOI: 10.23731/CYRM-2017-002.

[5] The CMS Collaboration. *Measurement of $t\bar{t}H$ production in the $H \to b\bar{b}$ decay channel in 41.5 fb^{-1} of proton-proton collision data at $\sqrt{s}$ = 13 TeV*. Techn. Ber. CMS-PAS-HIG-18-030. CERN. 2019. URL: https://cds.cern.ch/record/2675023.

[6] N. Shadskiy. „Treating Uncertainties with Bayesian Neural Networks in the measurement of ttH (H→ $b\bar{b}$) production“. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2020. URL: https://publish.etp.kit.edu/record/21982.

[7] B. Povh et al. *Teilchen und Kerne: Eine Einführung in die physikalischen Konzepte, 8. Auflage*. Berlin [u.A.]: Springer, 2014.

[8] W. Demtröder. *Experimentalphysik 4: Kern- Teilchen- und Astrophysik, 5. Auflage*. Berlin: [u.A.]: Springer Spektrum, 2017.

[9] *Standard Modell der Elementarteilchen*. (besucht am: 07.02.2021). URL: https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles-de.svg#/media/Datei:Standard_Model_of_Elementary_Particles-de.svg.

[10] P. Higgs. „Broken symmetries, massless particles and gauge fields“. In: *Physical Review Letters* Band 12 (1964).

[11] P. Higgs. „Broken symmetries and the masses of gauge bosons.“ In: *Physical Review Letters* Band 13 (1964).

[12] *Offizielle Seite des CERN zum Higgs Boson: "The Higgs boson"*. (besucht am: 07.02.2021). 2014. URL: https://home.cern/science/physics/higgs-boson.

[13] J. Resag. *Quanten, Quarks und der LHC*. Berlin: [u.A.]: Springer Spektrum, 2010.

[14] *Facts and figures about the LHC*. (zuletzt besucht am: 18.02.2021). URL: https://home.cern/resources/faqs/facts-and-figures-about-lhc#:.

[15] A. D. Rosso. *Particle kickers*. (besucht am: 07.02.2021). 2014. URL: https://cds.cern.ch/record/1706606/files/Diagram_image.png?version=1.

[16] *Offizielle Website des CMS-Detektors*. (zuletzt besucht am: 08.02.2021). URL: https://cms.cern/detector.

[17] The CMS Collaboration. „The CMS experiment at the CERN LHC“. In: *Journal of Instrumentation* 3 (2008). DOI: 10.1088/1748-0221/3/08/s0800.

[18] D. Barney. *CMS Detector Slice*. (besucht am: 08.02.2021). 2016. URL: http://cds.cern.ch/record/2120661/files/.

[19] Y. C. Cung. „Untersuchung von Optimierungsstrategien für Bayesian Neural Networks im Rahmen der ttH(bb)-Analyse am CMS-Experiment am CERN“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2020. URL: https://publish.etp.kit.edu/record/21999.

[20] K. El Morabit. „A study of the multivariate analysis of Higgs boson production in association with a top quark-antiquark pair in the boosted regime at the CMS experiment“. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2015. URL: https://publish.etp.kit.edu/record/21342.

[21] R. Kruse et Al. *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze, 2. Auflage*. Wiesbaden: Springer Vieweg, 2015.

[22] K. Choo et Al. *Machine Learning kompakt: Ein Einstieg für Studierende der Naturwissenschaften*. Wiesbaden: Springer Spektrum, 2020.

[23] D. P. Kingma und J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[24] R. Rojas. *Neural Networks - A Systematic Introduction*. Berlin [u.A.]: Springer, 1996.

[25] X. Ying. „An Overview of Overfitting and its Solutions“. In: *Journal of Physics: Conference Series* 1168 (Feb. 2019), S. 022022. DOI: 10.1088/1742-6596/1168/2/022022. URL: https://doi.org/10.1088/1742-6596/1168/2/022022.

[26] A. Y. Ng. „Feature selection, L 1 vs. L 2 regularization, and rotational invariance“. In: *Proceedings of the Twenty-First International Conference on Machine Learning* (Sep. 2004). DOI: 10.1145/1015330.1015435.

[27] Y. Gal. „Uncertainty in Deep Learning“. Diss. University of Cambridge, 2016.

[28] R. M. Neal. *Bayesian Learning for Neural Networks*. New York: Springer-Verlag, 1996.

[29] S. Kullback und R. A. Leibler. „On Information and Sufficiency“. In: *The Annals of Mathematical Statistics* 22.1 (1951), S. 79–86. ISSN: 00034851. URL: http://www.jstor.org/stable/2236703.

[30] J. van der Linden et al. *DRACO-MLfoy*. 2020. URL: https://github.com/kit-cn-cms/DRACO-MLfoy.

[31] F. Chollet et al. *Keras*. https://keras.io. 2015.

[32] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[33] J. V. Dillon et al. „TensorFlow Distributions“. In: *CoRR* abs/1711.10604 (2017). arXiv: 1711.10604. URL: http://arxiv.org/abs/1711.10604.

[34] H. Losbichlerm et al. „Neue Visualisierungsformen auf dem Prüfstand“. In: *Controlling Management Review* 60 (2016), S. 46–53. DOI: 10.1007/s12176-016-0010-2.

Anhang

A Beschreibung der Eingangsvariablen

In Tabelle A.2 sind die verwendeten Eingangsvariablen beschrieben. Eine allgemeine Bedeutung der Abkürzungen, die in den Variablen-Namen zu finden sind, ist sich in Tabelle A.1 zu sehen.

Tabelle A.1: **Erklärung der Abkürzungen, die in Tabelle A.2 verwendet werden.**

Abkürzung	Bedeutung
j	Jet(s)
t	b-Tag-Jets
l	Lepton
M	invariante Masse
p_T	Transversalimpuls
η	Pseudorapidität
ΔR	Abstand in der $\eta\phi$-Ebene
MET	fehlende Transversalenergie

Tabelle A.2: **Beschreibung der Eingangsvariablen, übernommen aus [19].**

Variable	Beschreibung
M_3	invariante Masse der 3 Jets mit größtem p_T
N_{Jets}	Anzahl der Jets im Ereignis
b-Tag-Wert i, $\overline{\text{b-Tag-Wert}}$	i-t größter- bzw. Durchschnitt der b-Tag-Werte
$p_T(\text{j}_1)$, $M(\text{j}_1)$	p_T und invariante Masse des Jets mit größtem p_T
$p_T(\Delta R_{\text{min}}(\text{j},\text{j}))$, $\Delta R_{\text{min}}(\text{j},\text{j})$, $p_T(\Delta R_{\text{min}}(\text{t},\text{t}))$, $\Delta R_{\text{min}}(\text{t},\text{t})$	p_T-Wert bzw. ΔR-Wert des Jets bzw. b-Jets mit dem kleinsten ΔR
$\overline{\Delta R}(\text{j},\text{j})$, $\overline{\Delta\eta}(\text{j},\text{j})$	Mittelwert von ΔR bzw. $\Delta\eta$
$H_T(\text{j})$	Summe der Transversalimpulse der Jets
$M_2(\text{t},\text{t})_{125}$, $M_2(\Delta R_{\text{min}}(\text{t},\text{t}))$	invariante Masse mit b-Tag, die am nächsten an 125 GeV ist, bzw. der b-Jets mit dem geringsten ΔR
$\overline{\Delta\eta}(\text{t})$, $\overline{\eta}(\text{t})$, $\overline{M}(\text{t})$, $\overline{\Delta R}(\text{t})$	Mittelwert invarianter Masse, η, $\Delta\eta$, ΔR der b-Jets
$\eta(\text{l})$, $p_T(\text{l})$	η bzw. p_T des Leptons
$M(\Delta R_{\text{min}}(\text{l},\text{j}))$, $\Delta R_{\text{min}}(\text{l},\text{j})$	invariante Masse von Lepton und b-Jet bzw. ΔR zwischen Lepton und b-Jet mit kleinstem ΔR-Wert
$M(\text{j},\text{l},\text{MET})$	invariante Masse aller Objekte des Ereignisses

B Histogramme der Mittelwerte und Unsicherheiten der Auswertung des Testdatensatzes

Hier sind die Histogramme zu sehen, auf die in Abschnitt 4.2 verwiesen wird. Auf der linken Seite ist für das gegebene Ausgabeneuron die Verteilung der Mittelwerte der Testdaten aller Klassen zu sehen. Die Skala der horizontalen Achse geht für die Mittelwerte über das gesamte Interval von null bis eins.

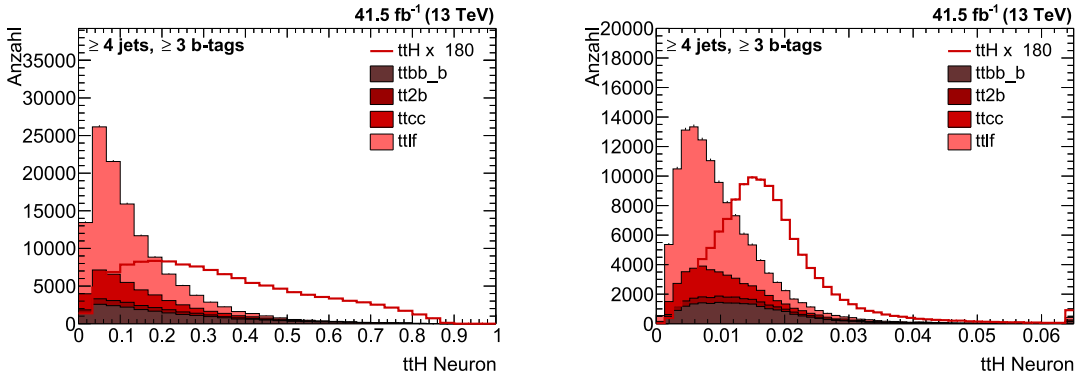

Abbildung B.1: **Netzausgabe des $t\bar{t}$H-Neurons**

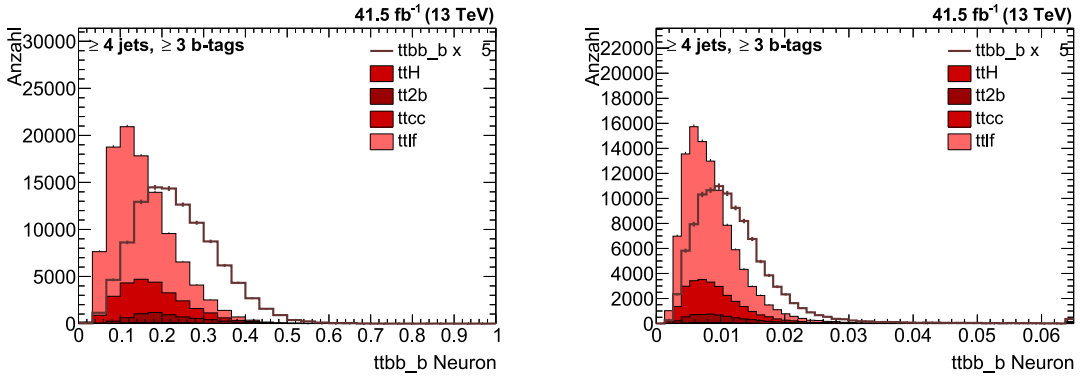

Abbildung B.2: **Netzausgabe des $t\bar{t} + b\bar{b}$-Neurons**

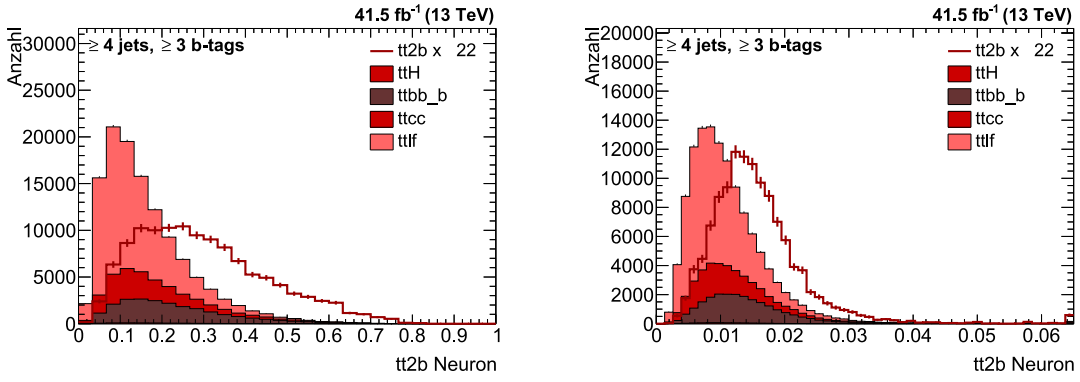

Abbildung B.3: **Netzausgabe des $t\bar{t}$ + 2b-Neurons**

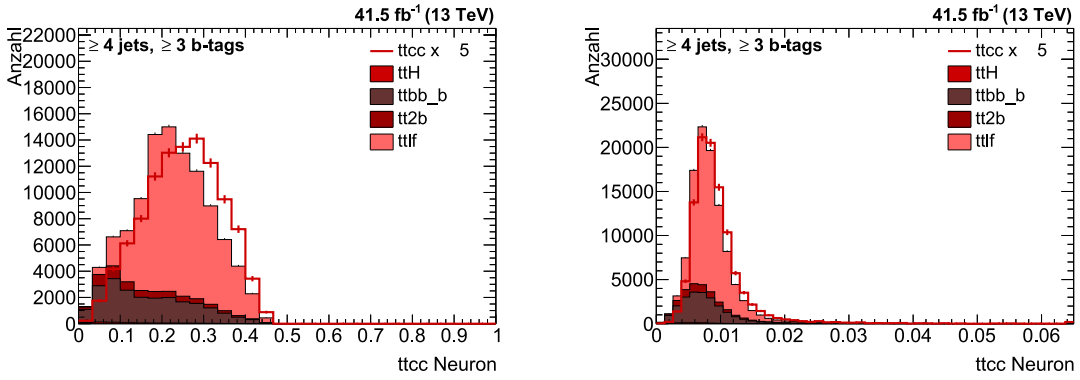

Abbildung B.4: **Netzausgabe des $t\bar{t} + c\bar{c}$-Neurons**

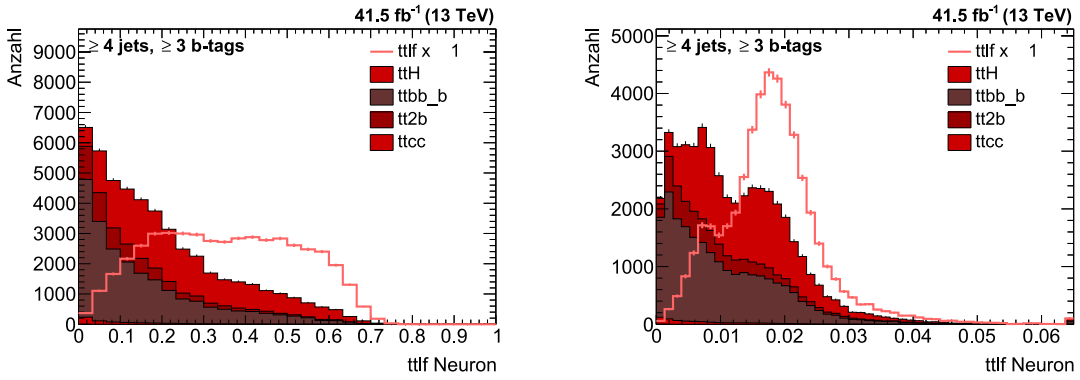

Abbildung B.5: **Netzausgabe des $t\bar{t}$ + light flavor-Neurons**

C Diskriminatoren zur Auswertung eines multiklassifizierenden BNNs

Hier sind die Diskriminatoren zu sehen, auf die in Abschnitt 4.2 verwiesen wird.

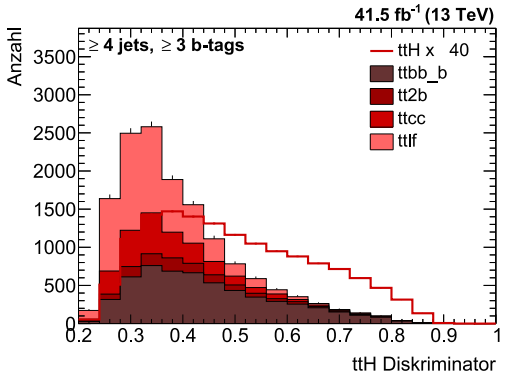

Abbildung C.1: **Diskriminator der $t\bar{t}$H-Klasse**

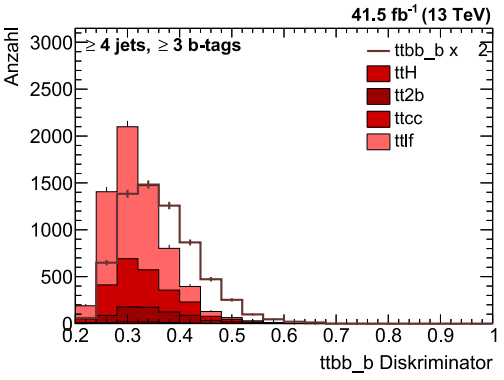

Abbildung C.2: **Diskriminator der $t\bar{t}+b\bar{b}$-Klasse**

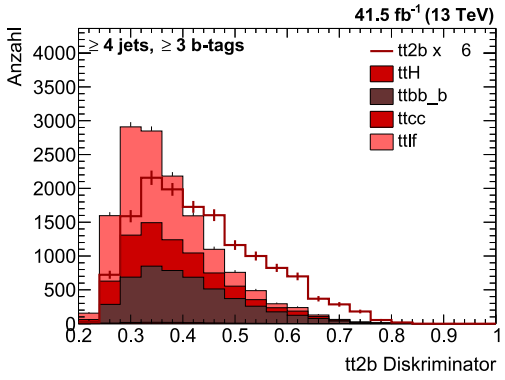

Abbildung C.3: **Diskriminator der $t\bar{t}$ + 2b-Klasse**

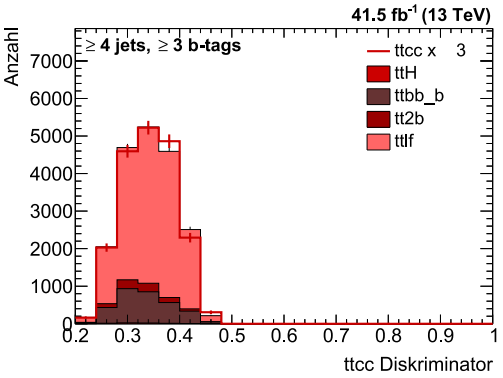

Abbildung C.4: **Diskriminator der $t\bar{t}+c\bar{c}$-Klasse**

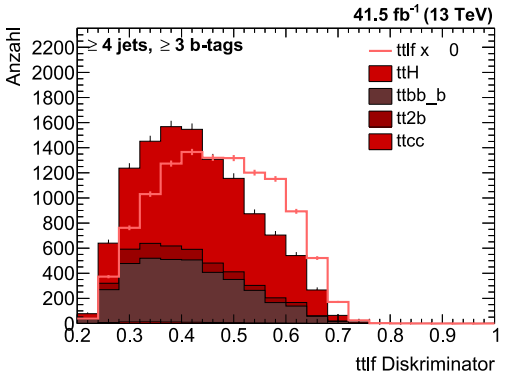

Abbildung C.5: **Diskriminator der $t\bar{t}$ + light flavor-Klasse**

D Histogramme zu den Ereignissen zwei bis 15

Im Folgenden sind die in Kapitel 5.1 erwähnten Histogramme der Netzausgaben für die Ereignisse 2 bis 15 zu sehen. Sie wurden auf gleiche Art wie Abbildung 5.1 erstellt (Beschreibung siehe Kapitel 5.1).

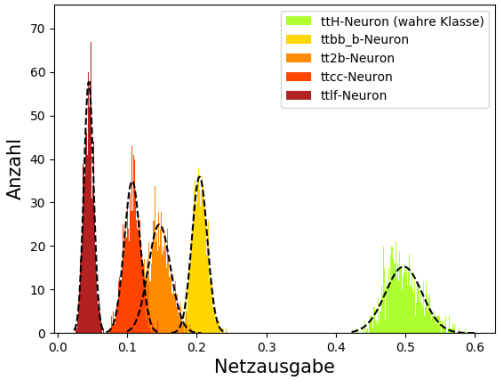

Abbildung D.1: **Ereignis 2**

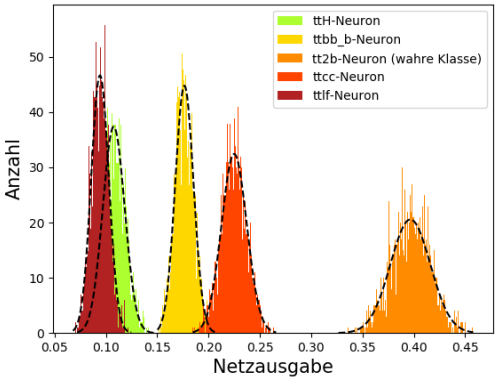

Abbildung D.2: **Ereignis 3**

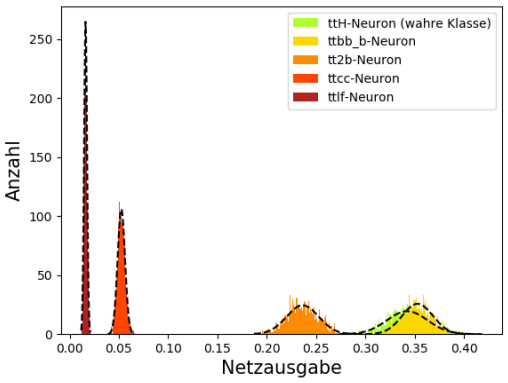

Abbildung D.3: **Ereignis 4**

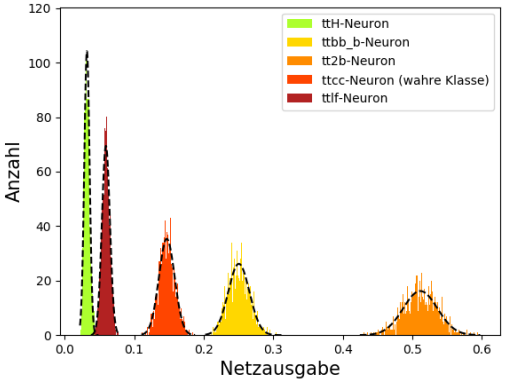

Abbildung D.4: **Ereignis 5**

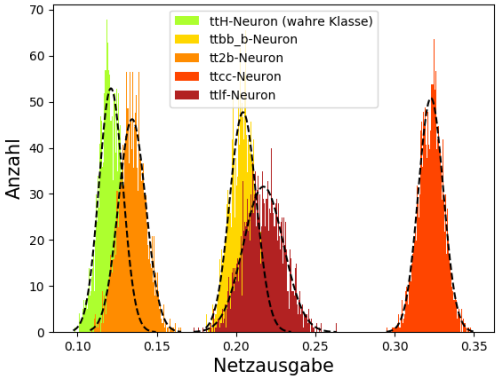

Abbildung D.5: **Ereignis 6**

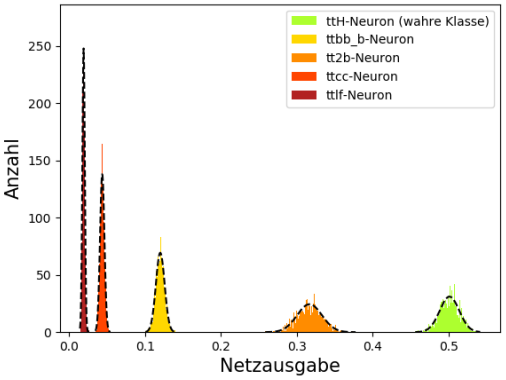

Abbildung D.6: **Ereignis 7**

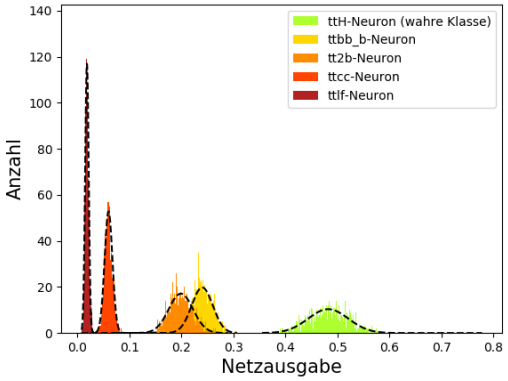

Abbildung D.7: **Ereignis 8**

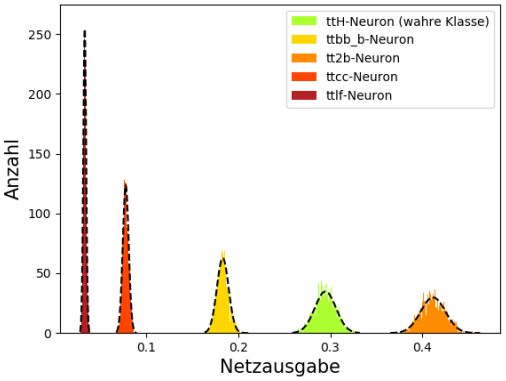

Abbildung D.8: **Ereignis 9**

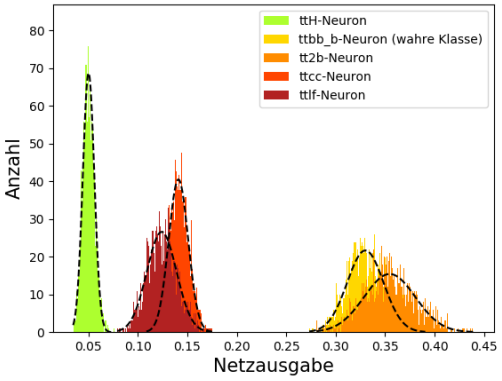

Abbildung D.9: **Ereignis 10**

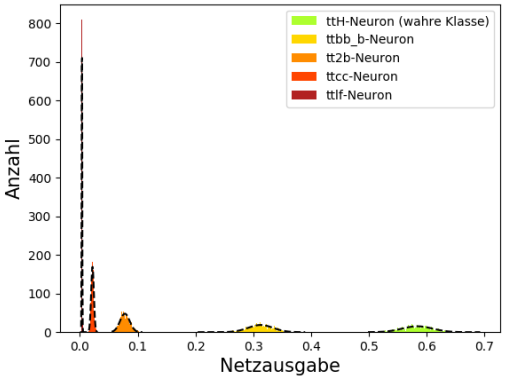

Abbildung D.10: **Ereignis 11**

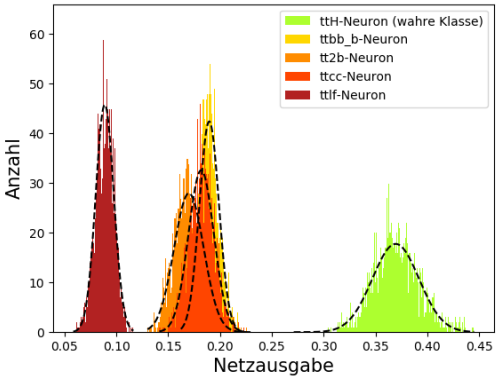

Abbildung D.11: **Ereignis 12**

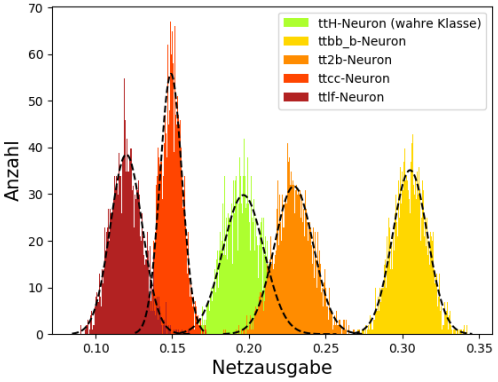

Abbildung D.12: **Ereignis 13**

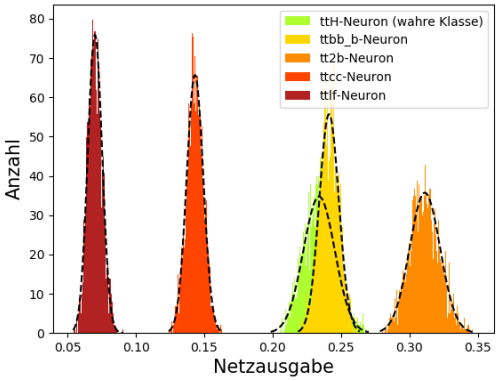

Abbildung D.13: **Ereignis 14**

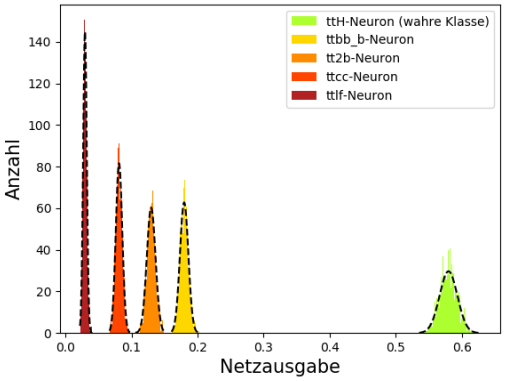

Abbildung D.14: **Ereignis 15**

E 2D-Histogramme zum Vergleich von BNN und ANN-Ensembles

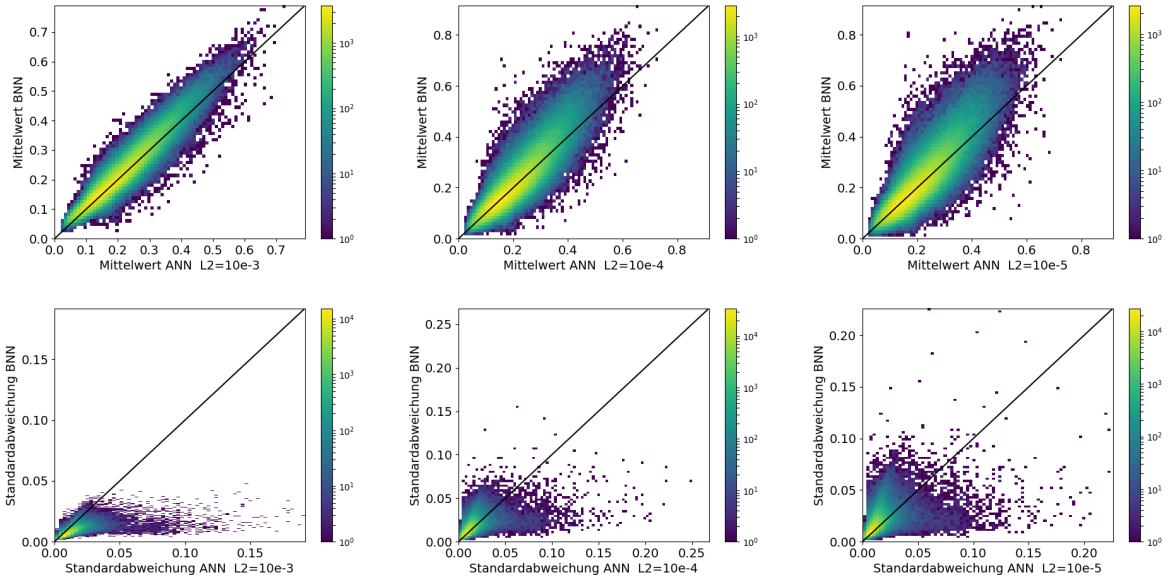

Abbildung E.1: **Korrelation der Vorhersagen der $t\bar{t}+b\bar{b}$-Neuronen von BNN und ANN-Ensembles** Der Aufbau der Abbildung ist gleich wie in Abbildung 5.3. Hier werden nur die Korrelationen der Mittelwerte und Standardabweichungen der $t\bar{t} + b\bar{b}$-Neuronen dargestellt.

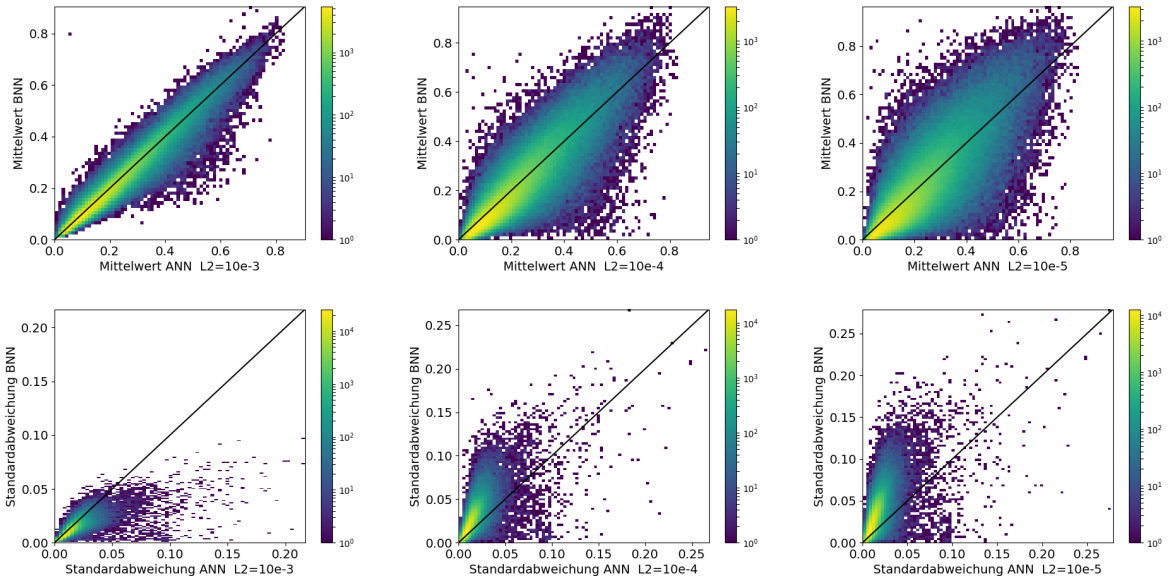

Abbildung E.2: **Korrelation der Vorhersagen der $t\bar{t}$ + 2b-Neuronen von BNN und ANN-Ensembles** Der Aufbau der Abbildung ist gleich wie in Abbildung 5.3. Hier werden nur die Korrelationen der Mittelwerte und Standardabweichungen der $t\bar{t}$ + 2b-Neuronen dargestellt.

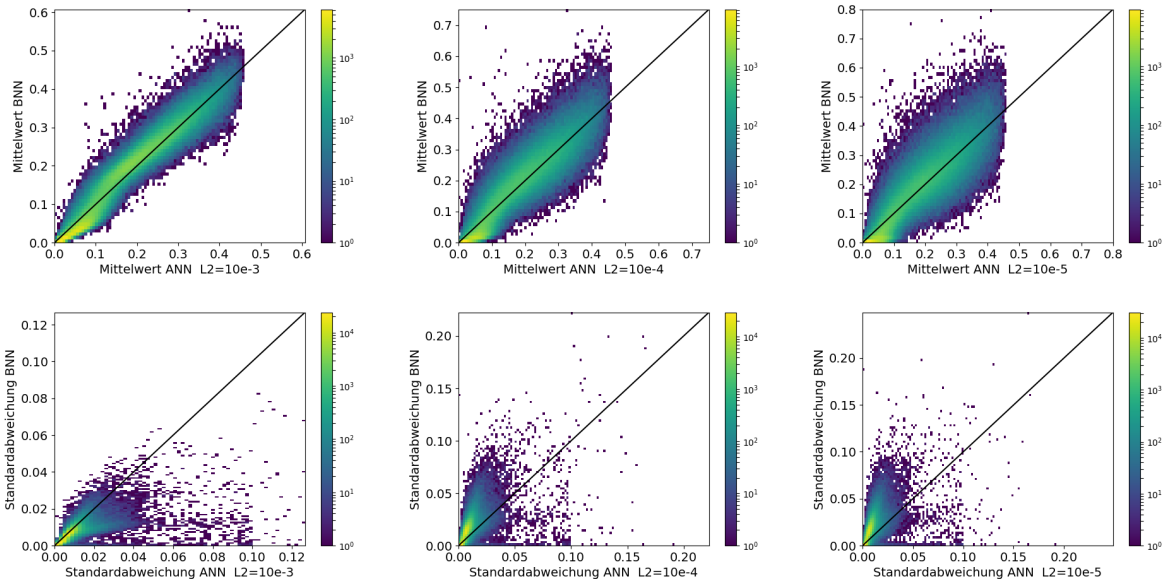

Abbildung E.3: **Korrelation der Vorhersagen der $t\bar{t}+c\bar{c}$-Neuronen von BNN und ANN-Ensembles** Der Aufbau der Abbildung ist gleich wie in Abbildung 5.3. Hier werden nur die Korrelationen der Mittelwerte und Standardabweichungen der $t\bar{t}+c\bar{c}$-Neuronen dargestellt.

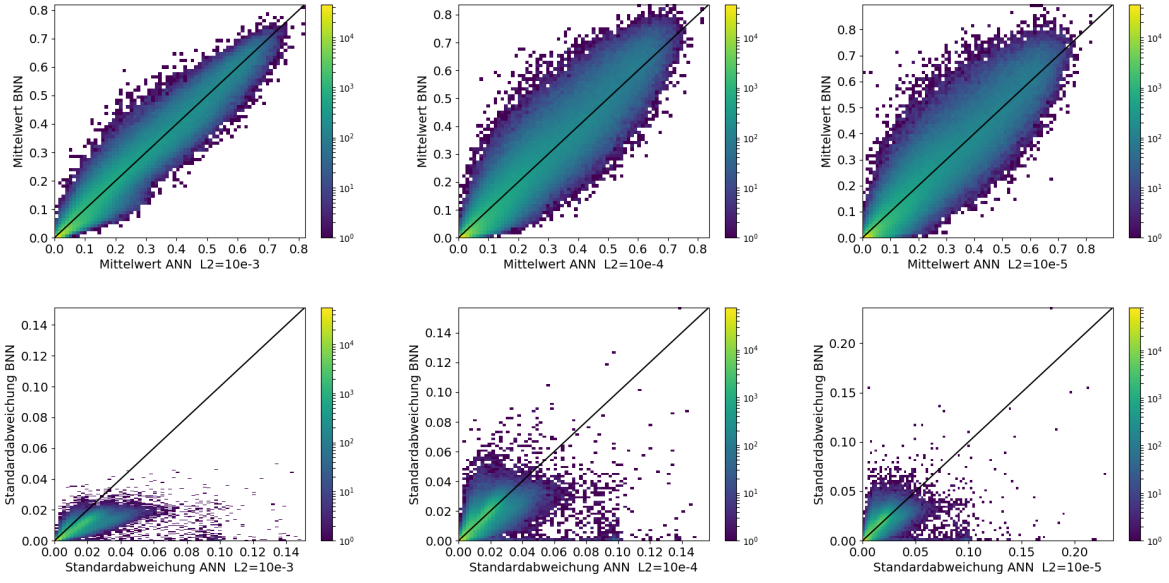

Abbildung E.4: **Korrelation der Vorhersagen der $t\bar{t}$ + light flavor-Neuronen von BNN und ANN-Ensembles** Der Aufbau der Abbildung ist gleich wie in Abbildung 5.3. Hier werden nur die Korrelationen der Mittelwerte und Standardabweichungen der $t\bar{t}$ + light flavor-Neuronen dargestellt.

F Diskriminatoren, die zur Klassifizierung die Standardabweichung zur Klassifizierung nutzen

In den hier gezeigten Diskriminatoren zeigen die Unterschiede der Diskriminatoren bei Addition bzw. Subtraktion einer Standardabweichung. Die Diskriminatoren, die die Klassifi-

kation ohne Einbeziehung der Standardabweichung zeigen sind in den Abbildungen C.1 bis C.5 zu sehen. Das linke Diagramm jeder Abbildung zeigt die Klassifizierung mit vorheriger Addition einer Standardabweichung, das rechte jeweils die Klassifikation mit vorheriger Subtraktion einer Standardabweichung.

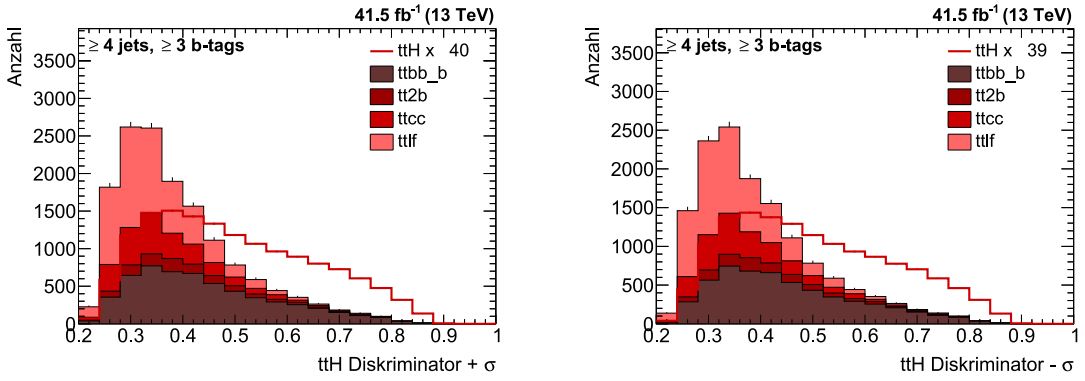

Abbildung F.1: **$t\bar{t}$H-Diskriminatoren mit Standardabweichung.**

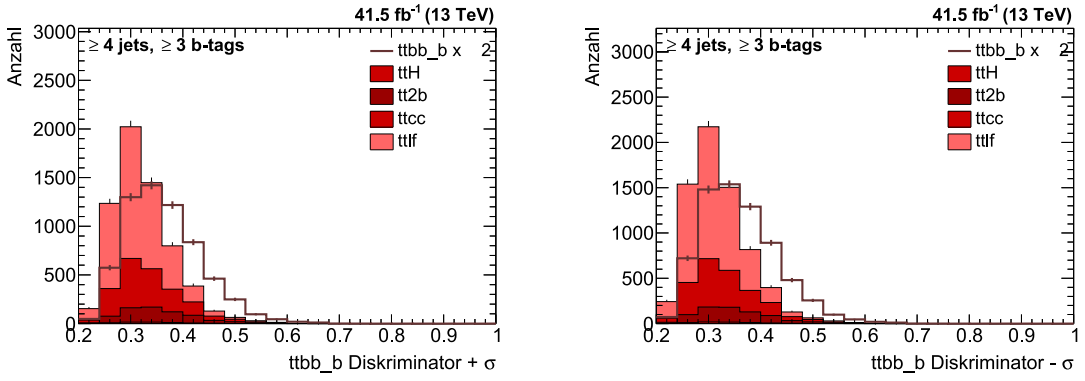

Abbildung F.2: **$t\bar{t}$ + $b\bar{b}$-Diskriminatoren mit Standardabweichung.**

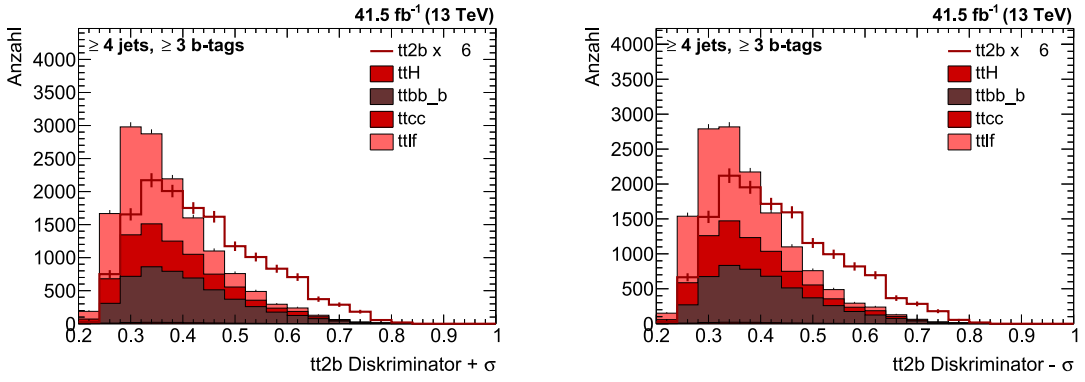

Abbildung F.3: **$t\bar{t}$ + 2b-Diskriminatoren mit Standardabweichung.**

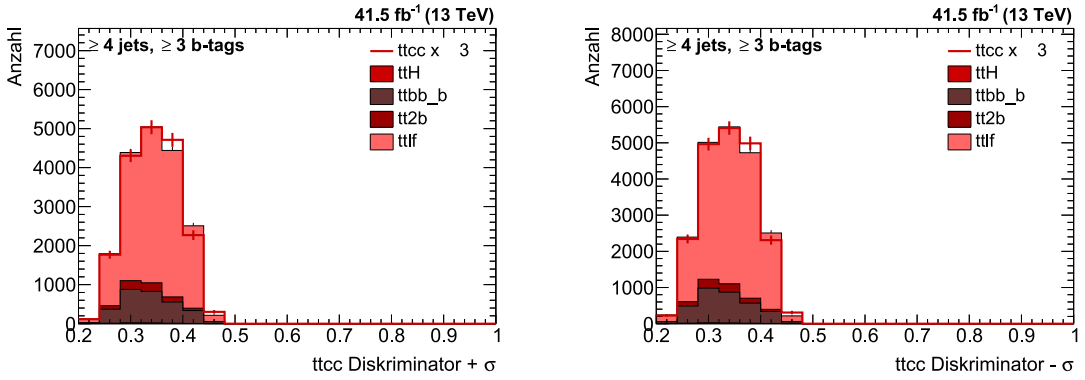

Abbildung F.4: **$t\bar{t}$ + $c\bar{c}$-Diskriminatoren mit Standardabweichung.**

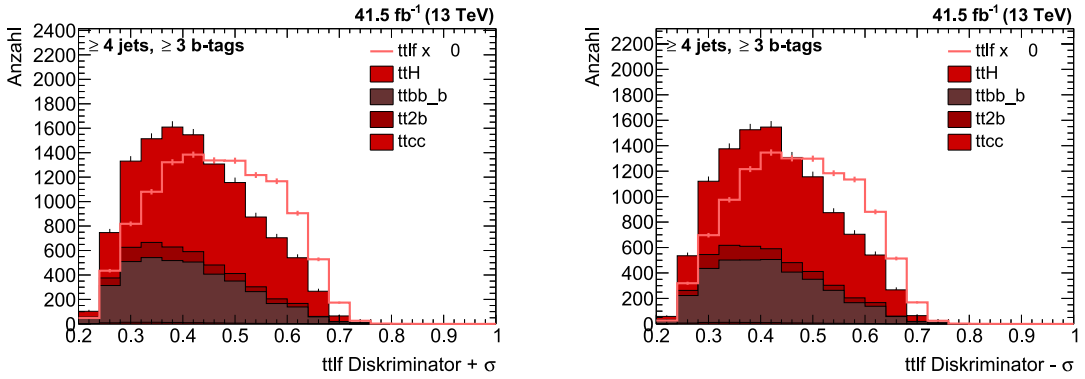

Abbildung F.5: **$t\bar{t}$ + light flavor-Diskriminatoren mit Standardabweichung.**

G Migrationsdiagramme

Hier sind die Migrationsdiagramme abgebildet, auf die in Abschnitt 6.2 verwiesen wird.

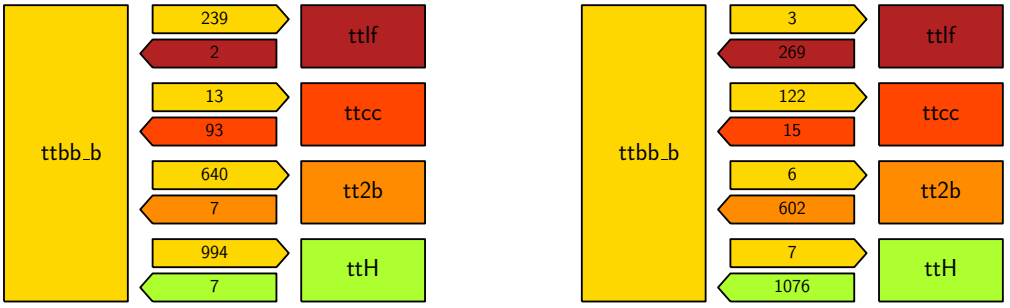

Abbildung G.1: **Migrationsdiagramme für $t\bar{t}$ + $b\bar{b}$**

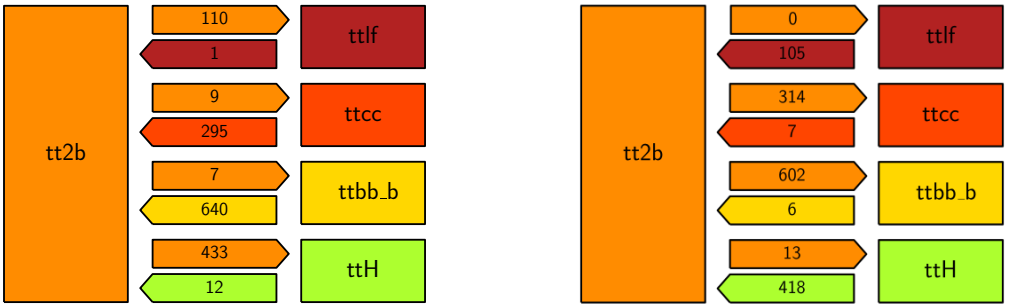

Abbildung G.2: **Migrationsdiagramme für $t\bar{t}$ + 2b**

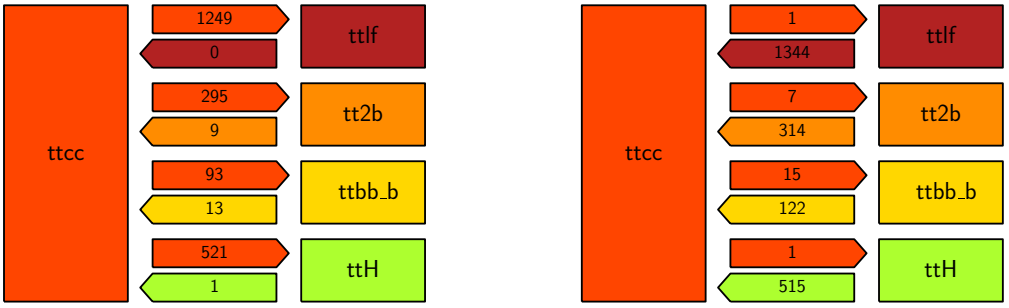

Abbildung G.3: **Migrationsdiagramme für $t\bar{t} + c\bar{c}$**

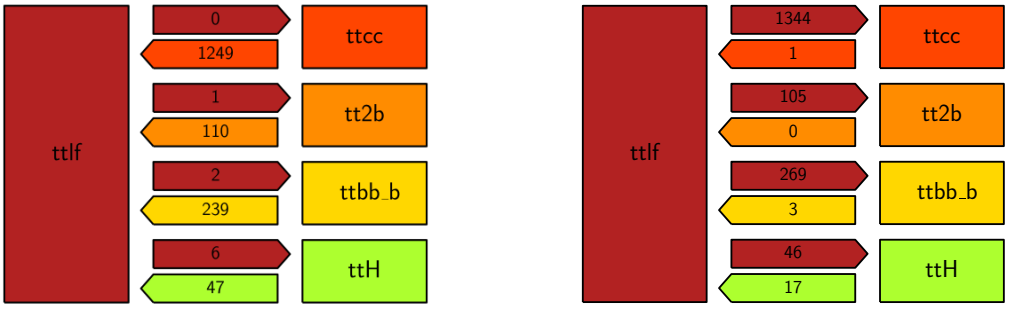

Abbildung G.4: **Migrationsdiagramme für $t\bar{t}$ + light flavor**