Karlsruhe Institute of Technology

Studies of the usage of neural networks in particle physics analyses

Masterarbeit
von

Simon Jörger

An der Fakultät für Physik
Institut für Experimentelle Teilchenphysik

ETP-KA/2020-10

Reviewer:	Dr. Roger Wolf
Second Reviewer:	Prof. Dr. G. Quast

Datum der Abgabe: 01.04.2020

Erklärung zur Selbstständigkeit

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung vom 17.05.2010 beachtet habe.

Karlsruhe, den 01.04.2020, _____

Simon Jörger

Als Ansichtsexemplar genehmigt von

Karlsruhe, den 01.04.2020, _____

Dr. Roger Wolf

Contents

1. Introduction

Modern machine learning (ML) methods are becoming increasingly popular among natural sciences and are already widely used in high-energy physics to solve classification and regression tasks for large amounts of data. In the analysis of $H \to \tau\tau$ events based on multi-variate methods (MVA), neural networks (NN) are an integral part of the analysis and are used for the classification of signal and background events. NNs by construction learn to classify these events directly from the data presented to them and without additional human inputs. The classification of the NN is hereby not only based on selections on the input variables but also on non-trivial correlations among the used input variables, making the classification of the NN oftentimes more efficient than a selection-based analysis.
Due to this self-learning capabilities of the NN, it is necessary to understand on the one hand the decision made by them and on the other hand the input variables given to them in the training process. In the first part of this thesis, a strategy is developed and used to remove input variables with small impact on the classification of the NN by using Taylor coefficients [1] to assess their influence on the NN output. This is done for two reasons: To ease the computational effort needed to verify the input variables and to align the set of input variables used for the analysis across the multiple NNs used in the $H \to \tau\tau$ analysis. The input variables of the NN represent real physics measurements and as such are subject to systematic uncertainties. In the current analysis, no prior information about systematic uncertainties is implemented in the NN training. However, there are two motivations to do this: On the one hand, the prediction of the NN could be compromised by systematic uncertainties if the prediction is based on the information of input variables with large systematic uncertainties. On the other hand, the systematic uncertainties of a given input variable might be directly dependent on other parameters. These dependencies might only be poorly known or even unknown. In such a case, the NN output might become more reliable if it is made more robust against those input variables. In the second part of this thesis, two approaches to implement prior information about systematic variations are investigated: Firstly, an already known approach using adversarial NNs [2] is tested. Secondly, a novel approach is introduced in which a penalty term in the loss function is used to uncorrelate the NN output from a given input variable with systematic variations. This approach is first investigated with a simple pseudo-experiment and afterwards with a high-energy physics example.

2. The $H \to \tau\tau$ analysis

An integral part of the current standard model of particle physics (SM) is the Higgs boson. The successful discovery of this boson in 2012 [3] confirmed the SM as the so far best and most complete theory in particle physics. The discovery of the Higgs boson was confirmed by using highly pure final states with clear signals. Since then, it was discovered in many more – often more complicated – final states. One of these final states is the Higgs boson decay into two tau leptons. In this chapter an overview of the current MVA-based analysis of the $H \to \tau\tau$ decay will be given which implements NNs to classify the events. This analysis is based on data measured by the Compact Muon Solenoid (CMS) detector [4] located at the European Organization of Nuclear Research (CERN) [5].

2.1. The Higgs boson

As the name implies, the intrinsic spin of a Higgs boson is an integer value of 0. Other than the rest of the bosons in the SM, however, the Higgs boson does not mediate a gauge interaction. Mathematically, it is a remnant of the Higgs mechanism, an additional degree of freedom which was not absorbed by the vector bosons. The so-called SM Higgs boson is the result of the simplest possible formulation of the Higgs mechanism [7, 8, 9, 10, 11, 12]. All measurements indicate that the particle discovered in 2012 at $m_H = 125\,\text{GeV}$ does indeed behave like this SM Higgs boson [3, 13]. The Higgs boson does not carry any charges. The coupling of a Higgs boson to other particles solely depends on the mass of those particles. In fact it can be shown that the Higgs boson couples linearly to the mass of fermions and quadratically to the mass of the gauge bosons. This can be seen in figure 2.1 where the coupling constant of fermions and the square root of the coupling constant of gauge bosons is shown as a function of the mass of the particles.

In general there are four main processes via which a Higgs boson is produced at the Large Hadron Collider (LHC) at CERN [15]. Figure 2.2 shows the Feynman diagrams for all four processes. From these four, the gluon fusion (top left) and vector boson fusion (top right) have by far the largest production cross section. A comparison of all production cross sections are shown in figure 2.3. As can be inferred from this graph, the cross section of gluon fusion is larger than the cross section of vector boson fusion (VBF) by a factor of around 10. Nevertheless, the VBF production has a significant influence on the analysis as the unique event topology provides a more distinct signature that allows the suppression of background processes.

As all SM particles except gluons and photons have mass, the Higgs boson can decay into almost every kinematically possible particle of the SM directly and into all particles via loop interactions. The branching ratios of the Higgs boson can be seen in figure 2.4. As the coupling of the Higgs boson is dependent on the mass of particles, the branching ratios are dominated by vector bosons for higher masses and by bottom quarks for lower masses. The initial discovery of the SM Higgs boson in 2012 was driven by the decay of the Higgs boson into two Z bosons, which further decayed into four leptons, as well as the decay into two

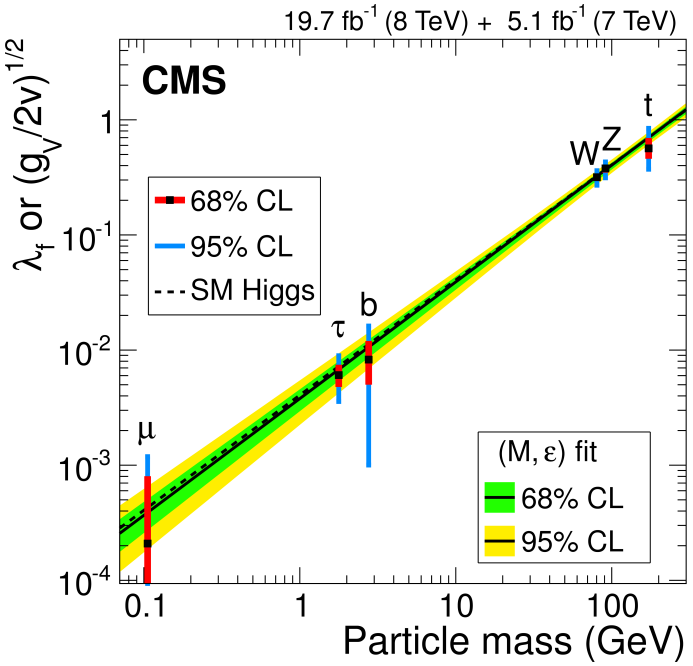


Figure 2.1.: Coupling of the SM Higgs boson. The y-axis has a different scale for fermions and bosons [6].

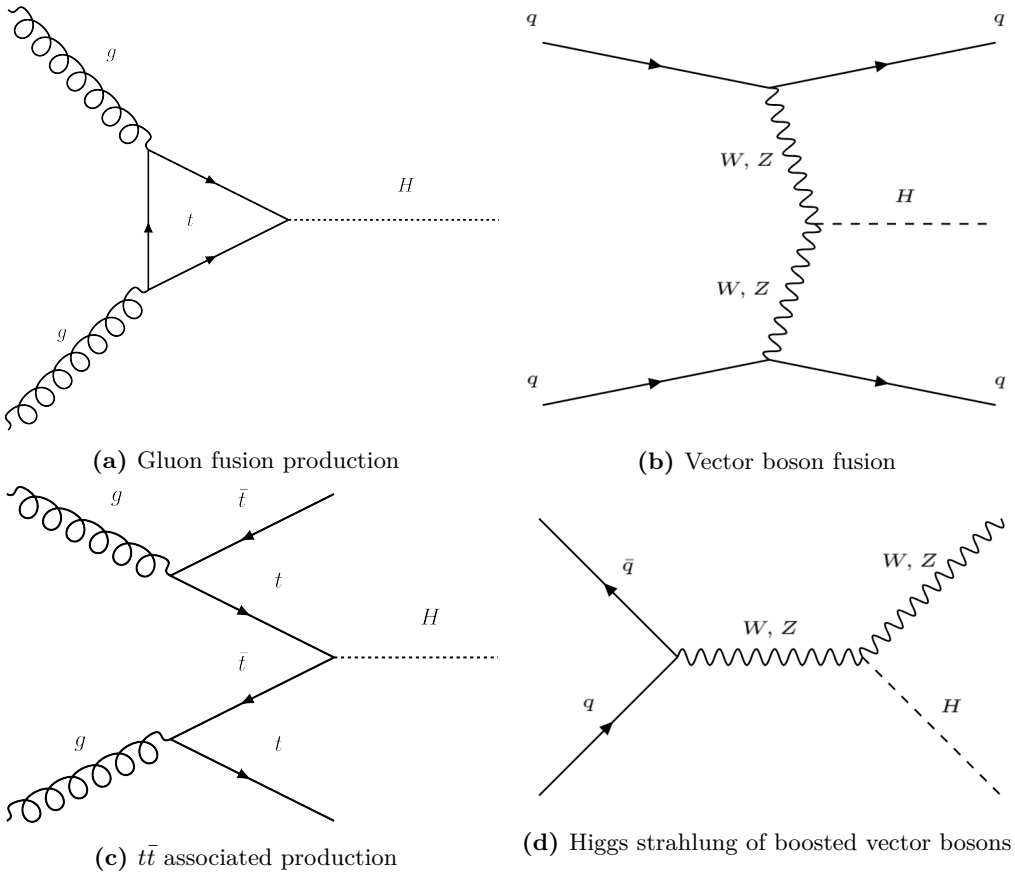
photons. Both decays have very low branching ratios compared to b quarks or W bosons, but they make up for this drawback by having very clean final states that are easier to reconstruct and distinguish from background processes. The decays into b quarks and W bosons face the problem of large backgrounds induced by processes with similar signatures.

At the relevant mass of $m_H = 125\,\text{GeV}$, the branching ratio into two tau leptons is approximately 6 %. The decay of the Higgs boson into two fermions is illustrated by the Feynman graph in figure 2.5. While the decay of the Higgs boson into two tau leptons is not necessarily rare and does not have the same problems of a large background as b quarks and W bosons, the reconstruction of this final state is still challenging, mostly due to the problematic decay of the two τ's into at least two undetectable neutrinos.

2.2. The CMS experiment

The data which this analysis uses was collected by the CMS experiment. The CMS experiment is stationed at the LHC, a proton proton collider with 27 km in circumference and a center of mass energy of currently $\sqrt{s} = 13\,\text{TeV}$. The CMS experiment is one of four major experiments located at the collider (see figure 2.6). The detector is a classic 4π detector encompassing the beam tubes with a point of collision at its center. It consists of four main components as seen in figure 2.7, that are aligned around the beam pipe. The detector weighs around 14000 t with a length of 21 m and 15 m in diameter. The superconducting solenoid produces a magnetic field of 3.8 T to bend the tracks of charged particles. The four main components from most inner to most outer component are:

- **Tracking system:** Nearest to the beam pipe is the tracking system of the detector. It consists of an array of silicon pixel and strip detectors. Charged particles lead to an electric signal in the silicon detectors along their trajectory. Each electric signal ("hit") is measured and the particle track is later reconstructed by following the path of the detectors that were hit. The interaction point of particles can also be reconstructed by extrapolating the paths of the particles. From the curvature of the track, the transverse momentum as well as the charge of the particle can be determined.



(a) Gluon fusion production

(b) Vector boson fusion

(c) $t\bar{t}$ associated production

(d) Higgs strahlung of boosted vector bosons

Figure 2.2.: Main processes for Higgs production at the LHC.

- **Electromagnetic calorimeter:** Next in line is the electromagnetic calorimeter (ECAL). It measures the energy of photons and electrons by initiating an electromagnetic shower when a photon or electron passes through. A scintillating material produces photons proportional to the energy of the particles in the shower. The emitted photons are then measured with photo diodes. The ECAL of the CMS experiment is homogeneous meaning that the same material is used for producing showers and scintillation. This way, no energy information is lost by being deposited in material that does not produce photons. The downside is that the scintillation of the material is much weaker than in specialized szintillator materials and thus harder to read out with photo diodes. The used material is lead tungstate crystals.

- **Hadronic calorimeter:** The hadronic calorimeter (HCAL) measures the energy of hadrons which pass through the ECAL. It does this by the same principle as used in the ECAL. The HCAL is a sampling calorimeter. Absorbing material such as brass alternates with scintillating material. In general an HCAL is much larger than the ECAL due to the larger interaction length of hadrons and has a larger uncertainty on the measured energy mostly due to the decay of hadrons into uncharged particles.

- **Muon system:** The muon system follows the superconducting solenoid which surrounds the HCAL. As muons are minimally interacting particles and can penetrate several meters of iron without interaction, they are the only SM particles beside neutrinos that reach this point. The muon system consists of gaseous tracking chambers, the so-called muon chambers. The momentum and charge of the particle can again be determined by the curvature of a muon in this tracking systems. This information is combined with the information of the innermost tracking system for consistency.

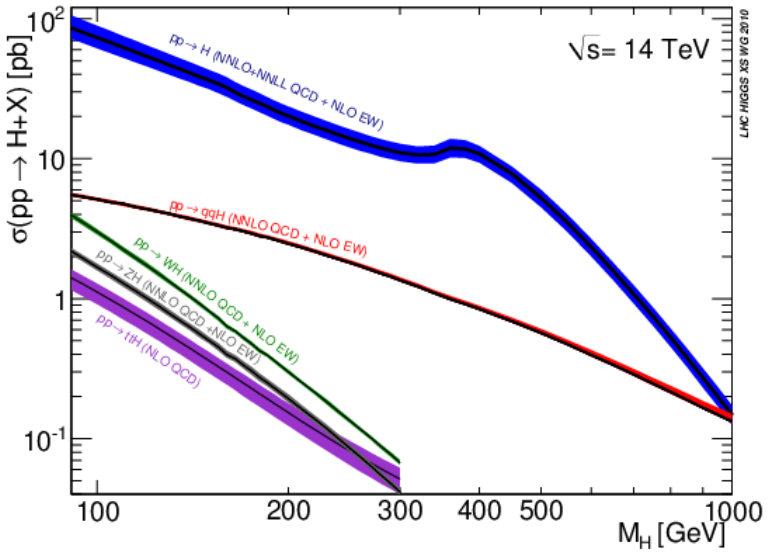


Figure 2.3.: Cross sections of various production processes for different masses of the Higgs boson [14].

As τ leptons have a mean lifetime of only 2.9×10^{-13} s, they cannot be directly measured in the detector. Instead the decay products of the di-tau system are measured. The decay of a τ lepton is shown in figure 2.8. There are four "final states" which can be directly measured in the detector: the $e\mu$ final state, the $e\tau_h$ final state, the $\mu\tau_h$ final state and the $\tau_h\tau_h$ final state. Here, τ_h represents a particle jet induced by a decay of a τ lepton.
After the measurement by all subdetectors, the data is processed by several triggers to reduce the immense influx of data from an event rate of 40 MHz to 100 Hz. As rare processes such as the production and decay of a Higgs boson are the focus of analyses, most proton-proton collisions that are measured at the experiment can be immediately discarded as they are only related to already known physics. The trigger system at CMS has two parts: The level one trigger system (L1) is a system of hardware triggers implemented in the form of field-programmable gate arrays (FPGA) directly on the detectors that immediately reduce the rate of events to approximately 100 kHz. This data is then sent to the high level trigger system (HLT) which is a computer farm of around 1000 standard computers performing simple analysis tasks to reduce the amount of data further and concentrate on only those events that are of interest. The events are then finally stored to disk for further offline analysis [19].

2.3. Event identification and uncertainties

The events measured at the CMS experiment can be classified into signal processes which are of interest and background processes which have a similar signature then the signal processes and can therefore obscure them. Furthermore, each measurement in a physics analysis is subject to systematic variations which shift the data in certain directions making the processes potentially harder to discern from each other. As such a good understanding of the systematic variations is paramount for a successful signal extraction.

2.3.1. Signal processes

The signal classes consist of three Higgs production processes in which the produced Higgs boson decays into a di-tau system: the VBF, Higgs Strahlung in which the vector boson decayed into two quarks, and gluon fusion. Those processes are selected due to their unique event topology which makes it easier to suppress background processes. The VBF and

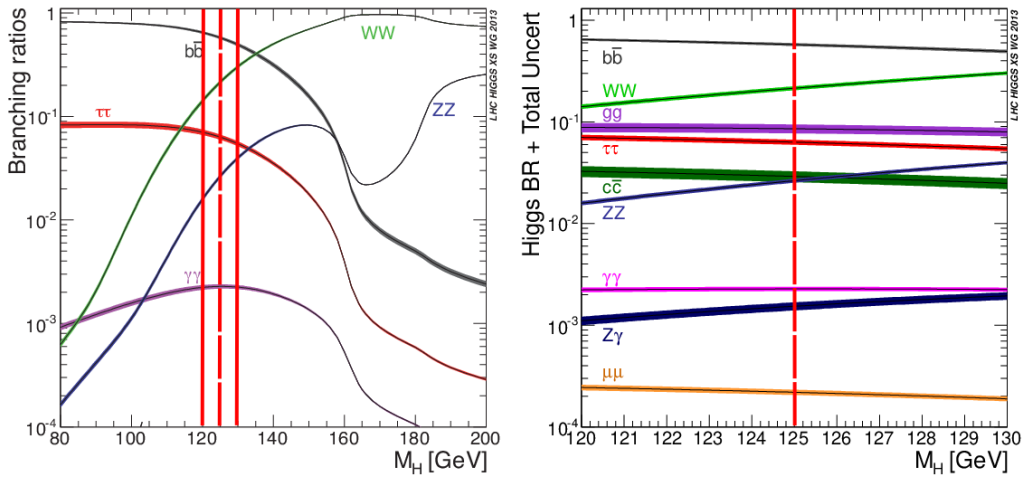


Figure 2.4.: SM Higgs boson branching ratios [16]. The highlighted area of the left graph is enlarged in the right graph. The mass of the SM Higgs boson is shown as a dashed red line.

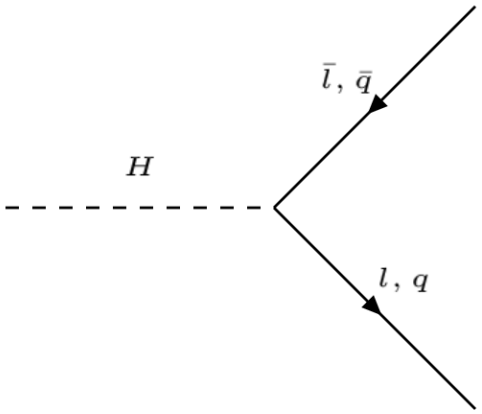


Figure 2.5.: Higgs decay into two fermions.

Higgs Strahlung process are combined into a single signal category due to very similar signatures. The two signal categories are then further split into smaller categories based on the reconstructed mass, transverse momentum and number of jets. This so-called stage 1.2 binning is depicted in figure 2.9. The stage 1.2 binning was slightly changed for this analysis. The red boxes around the categories symbolize which categories were condensed into one category for the training of the NN discussed in section 2.4.1. The condensation of categories was necessary to have a sufficient amount of data in each category for the training of the NN and for the statistical inference afterwards. For each signal category, a signal strength can be extracted and the contributions of each category can be combined to extract a single inclusive signal strength for the process $H \to \tau\tau$. For the analysis described in section 3, the signal categories of stage 1.2 binning were simplified to the so called stage 0 binning. It only consists of the two categories: gluon fusion and the combination of VBF and Higgs Strahlung. In theory, one could also combine those two signal categories into one single signal category. This inclusive analysis, however, was not used for the training of any NN presented here. The signal processes are the same for all four final states of the di-tau system measured in the detector and across all years.

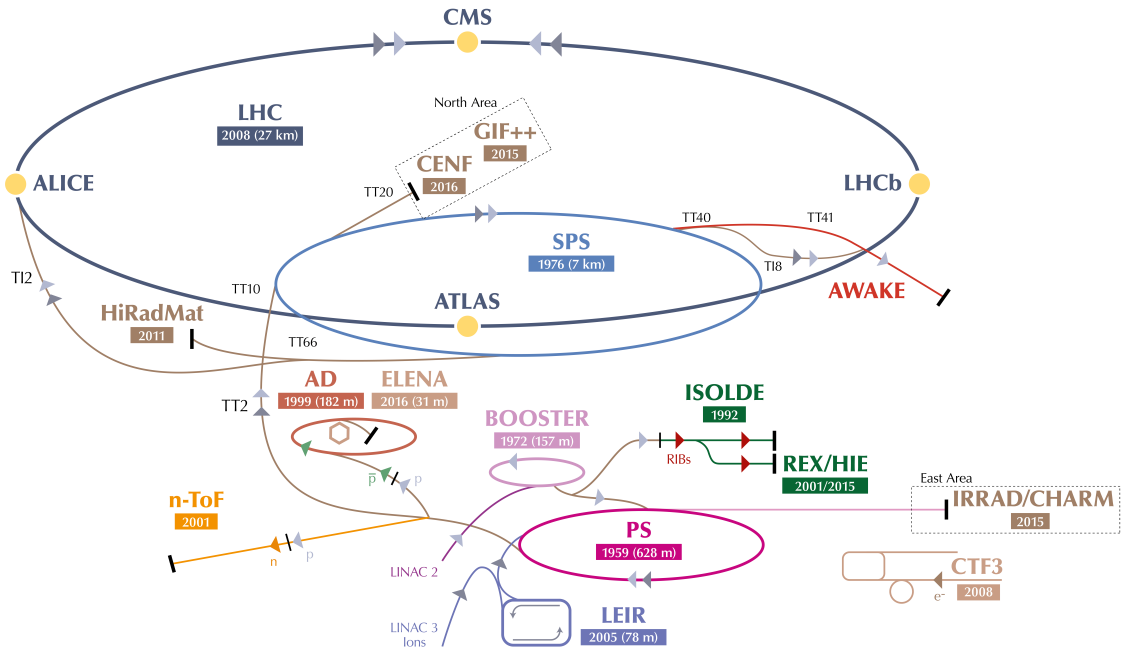


Figure 2.6.: The accelerator complex at CERN with all parts [17].

2.3.2. Background processes

There are several physics processes that have similar final states as the one described above and can thus be confused with signal processes. For this analysis, the following background processes are relevant:

- $\boldsymbol{Z \to \tau\tau}$ **:** Z bosons are produced very frequently via the Drell-Yan process at the LHC. The Z boson can decay into tau leptons with a branching ratio of 3.3%. The measured final states are the same as for the $H \to \tau\tau$ decay except for the invariant mass of the two tau leptons and the spin. As mentioned before though, accurately measuring the energy of the system is difficult due to neutrinos produced in the decay. This makes this background difficult to separate from the signal processes and thus it is the most dominant background process.

- $\boldsymbol{Z \to ll}$ **:** The ll denotes the decay of Z bosons to electrons, muons and neutrinos. Those decays have the same branching ratio as the τ decay due to lepton universality. In principle, this decay should be discernible from the signal processes due to different final states, but due to object misidentification in the reconstruction of the particles, the leptons could be falsely identified as τ leptons. Those misidentified leptons can also occur due to pile-up or initial state radiation. This background can be effectively suppressed with additional lepton vetos.

- **W + Jets:** This process can be mistaken for the signal process in two different ways: Firstly, a jet can be falsely identified as a hadronically decaying τ. Secondly, the W boson can decay into a τ lepton and – together with a misidentified jet – can mimic the final state of the signal process. This background is produced very frequently at the LHC. It can be suppressed in certain channels with additional vetos and cuts on the τ mass.

- $\boldsymbol{t\bar{t}}$ **:** Top quark production is not the most frequent process at the LHC. Nevertheless it is a relevant background for this analysis, as the top quark exclusively decays into W bosons which again can decay further into τ leptons potentially creating a di-tau system with a similar signature to the signal process. As in other processes, jets and

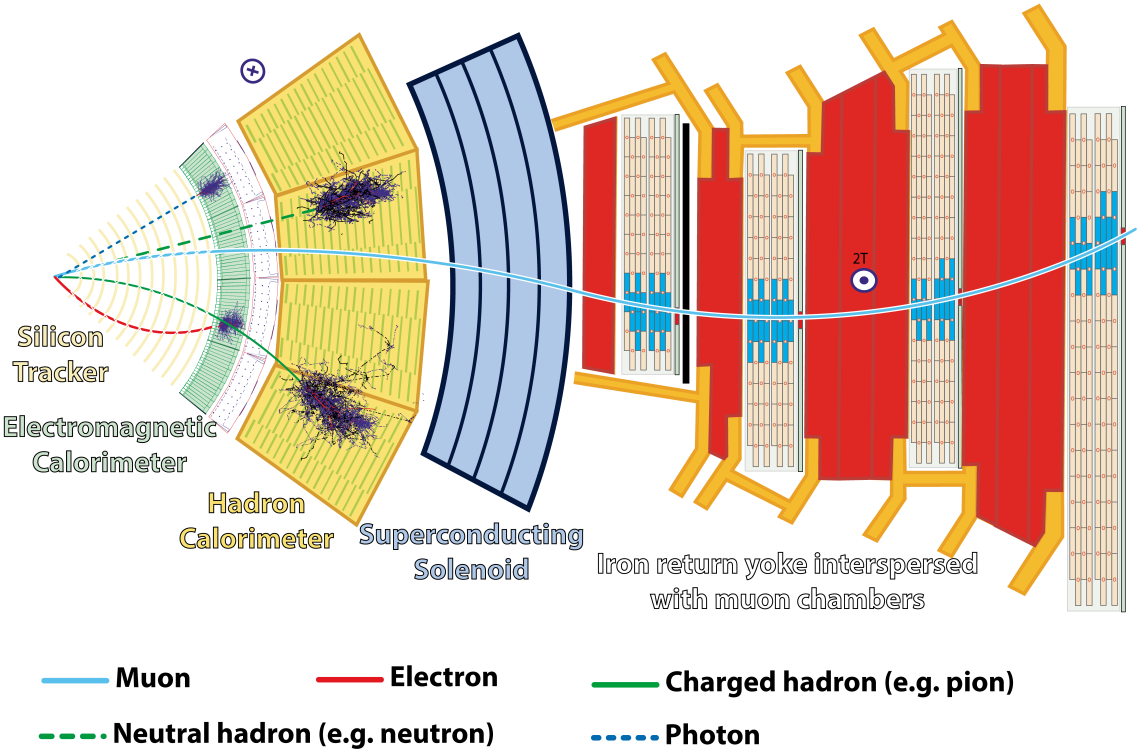


Figure 2.7.: Slice of the CMS detector with all major detector components and example tracks [18].

leptons produced by this process can also be misidentified as hadronically decaying τ leptons. This background can also be suppressed by b-tag vetos due to the presence of two b quarks in this decay.

- **QCD:** The QCD background sums up all final states with a high jet multiplicity. It is the most frequent and most versatile background, and most analyses at the LHC have to take this background into account. Some of this final state jets may be reconstructed as hadronically decaying taus, making it a background for this analysis as well.
- **Di-Boson:** This background is a combination of Z and W bosons decaying in such a way to be misidentified as the signal process. The contribution of this process is relatively small.

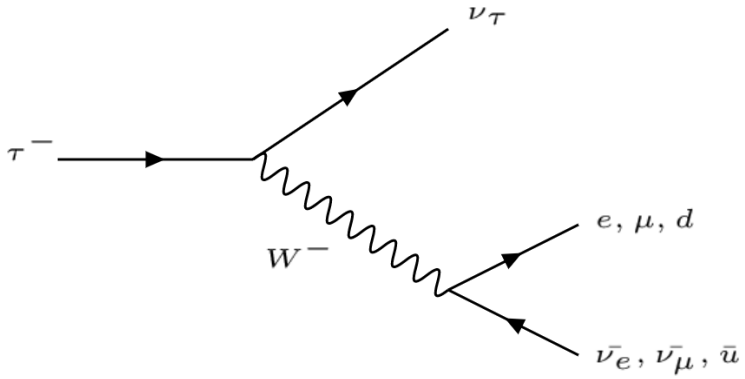
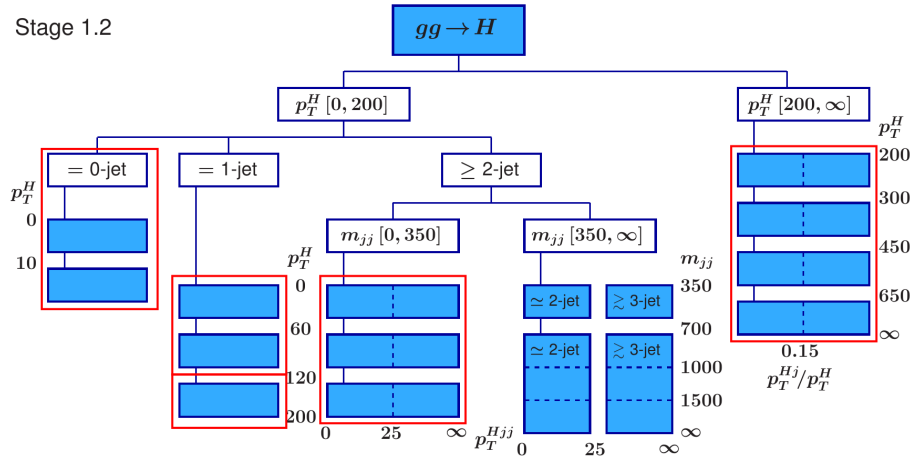
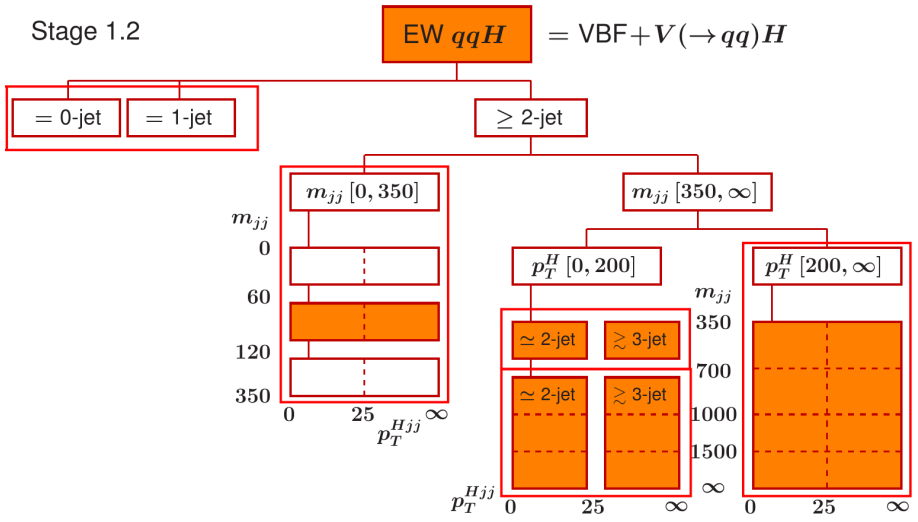


Figure 2.8.: Potentially decays of a τ^- lepton. Most decays involve two undetectable neutrinos.



(a) Binning for gluon fusion.



(b) Binning for vector boson fusion.

Figure 2.9.: Binning scheme for signal classes in classification with NNs [20]. The red boxes around the processes indicate one signal class during the NN classification as some processes were combined into one process to have enough events for training for the NN. The high mass category for $m_{jj} > 350$ of the gluon fusion was merged with the VBF process.

In general these backgrounds were estimated using Monte Carlo (MC) simulation techniques. Sometimes certain backgrounds were grouped together when training the NN into a `misc` category due to a low number of events or similar signatures.

While the MC simulation shows good agreement with the data, a background estimation that is derived from real data is in general preferable as the agreement to real data should be given by construction. The current analysis uses two of this data-driven methods to estimate most backgrounds given above:

- **τ-embedding:** All process with two genuine τ leptons – meaning not a τ due to misidentification in reconstruction – in the final state can be estimated using this technique. Taking events from data with a $\mu\mu$ final state, the muons are removed from the event record and replaced with simulated τ leptons. The simulation part is reduced to the decay of the τ leptons. Furthermore, due to the high number of $\mu\mu$ events in data, the overall statistics for all events are increased by the embedding method. A more detailed description can be found in [21].

- F_F **method:** All processes with jets misidentified as hadronically decaying τ leptons can be estimated with this method. The general principle of the F_F method is to measure a transfer factor ratio F_F in a so-called determination region free of genuine $\tau\tau$ events which is not used in the actual analysis. F_F is defined by the number of hadronic τ decays with certain identification requirements to the number of hadronic τ decays where those identification requirements were inverted. Afterwards a signal region is defined, which is mostly populated by signal events relevant for the analysis but also with a small number of background events that is estimated by this method. This signal region is then mirrored into an application region by inverting a certain requirement that was used to construct the signal region, e.g. the requirement of the τ leptons to be isolated. The requirement that is inverted is chosen in such a way that the application region is orthogonal to the signal region. The application region would then only uses events with e.g. non-isolated τ leptons. This region is therefore dominated by background events with only a small number of signal events. The universal assumption made in this technique is that F_F which was previously measured in the determination region is the same for the number of background events measured between the signal region and the application region. With this assumption the number of background events $N_{\text{Signal Region}}$ can simply be estimated as

$$N_{\text{Signal Region}} = N_{\text{Application Region}} \times F_F. \tag{2.1}$$

  A more detailed description of the technique and the requirements used in this analysis is given in [22].

Other than the signal processes, the background categories for the classification of the NN can vary between final states as certain processes are more relevant in certain final states. For the main analysis of $H \to \tau\tau$ events, the data-driven background categories are used for the training of the NN. For the study in section 3, however, the MC simulated backgrounds are used.

2.3.3. Systematic uncertainties

An important part in any physics analysis is the handling of systematic uncertainties. Systematic uncertainties are all uncertainties that are not caused by statistical fluctuations of the data. They are often caused by inaccuracies in measurements, simulation and theory. The correct handling of them is paramount to get a reliable and robust result for the accuracy of the measurement. In general, the systematic uncertainties in this analysis are applied on histogram level after the classification of the processes either via additional datasets containing the shifted values or via weights. It is to be noted that many variables are highly correlated and systematic uncertainties have to be propagated to all correlated variables. There are three main sources of systematic uncertainties in the current analysis that can then be further separated into categories.

The largest group are the shape uncertainties. They cause shifts in the shapes of the histograms and are applied via statistical weights or shifts for the events in each respective variables. Shape uncertainties are:

- **Energy scale uncertainties:** There are several uncertainties on measured energy scales. For this analysis the τ energy scale, the electron energy scale, the jet energy scale, the fake τ energy scale as well as the missing transverse momentum (MET) energy scale are considered.

- **Reweighting uncertainties:** Reweighting uncertainties are applied to the top quark p_T and the Drell-Yan mass m_{ll} and p_T. They account for higher-order effects and miss-modeling in the matrix element calculation in the simulation.
- **Reconstruction uncertainties:** Those uncertainties encompass the b-tagging efficiency mostly used to identify the $t\bar{t}$ background as well the the τ_h tracking efficiency in embedded event samples.
- F_F **method uncertainties:** The F_F method introduces several uncertainties that are dependent on which background is estimated using this method. This includes a statistical uncertainty due to a fit uncertainty, uncertainties due to non-closure corrections and uncertainties due to data-to-simulation correction factors. For a more complete overview see [23].
- **QCD estimate uncertainties:** This uncertainty is only applied in the $e\mu$ final state, as all other methods use the F_F method to estimate the QCD background.
- **Bin-by-bin uncertainties:** Bin-by-bin uncertainties stem from the limited number of the simulated events. The Barlow-Beeston approach [24] is used to measure the effect of them. Each bin is weighted with its associated Poisson error to produce alternative shapes and simulate the bin-by-bin uncertainty.
- **Other uncertainties:** Other uncertainties include the uncertainty due to prefiring and the $t\bar{t}$ contamination in the embedded samples. Prefiring means that a trigger for a detector is prematurely blocked because jets or photons were falsely matched to a previous event due to a shift in timing.

The second largest category of uncertainties are the normalization uncertainties. They are applied directly on the yield of each affected process via sampling of an additional term instead of the shape of the histogram.

- **Luminosity uncertainty:** These uncertainties are applied per year and are usually around 2.5%. They are introduced for simulated processes.
- **Electron, muon and tau ID efficiency:** The efficiency of reconstructing and identifying the leptons.
- **Trigger efficiencies:** As explained in the previous section, the preselection of events for data and simulation is done via triggers. Those triggers are also subject to a systematic uncertainty which has to be accounted for.
- **Background and fake factor normalization uncertainties:** The background normalization uncertainties vary between $4 - 6\%$ depending on the background and mostly account for the uncertainty in the cross section of the associated process. The fake factor normalization uncertainty stems from the substraction of the contribution of processes with real τ's in the final state in the application region.
- $l \to \tau$ **fake rate:** The uncertainty associated with the misclassification of leptons as τ's.

The last category are the uncertainties in the signal theory. They can have effects on both the yield and the shape. The theory uncertainties include:

- **Cross section and branching ratio uncertainties:** Uncertainties due to incomplete knowledge of the particle density functions (PDF) as well as the normalization and factorization scale. This directly affects the expectation for the cross section and branching ratio.

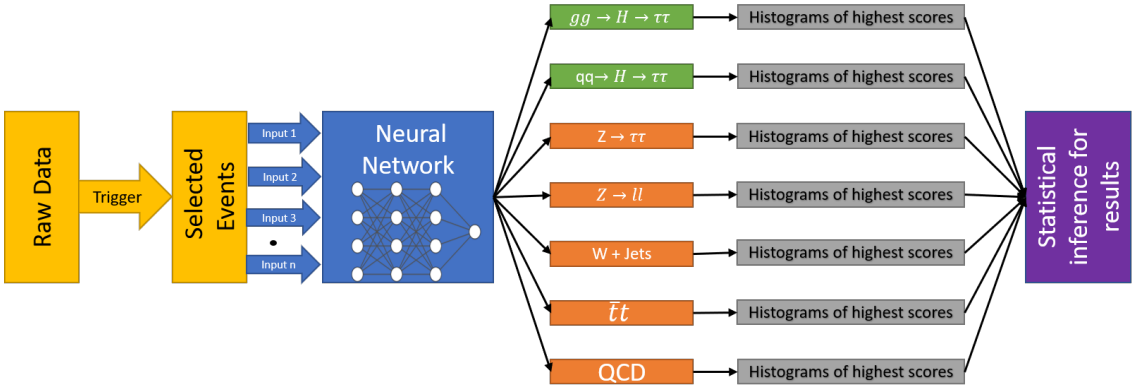


Figure 2.10.: Overview of the presented analysis. From the raw data, selected events are chosen based in criteria. Those selected events are then classified by NNs into signal and background processes. The histograms of those processes are then used in a statistical inference to extract the final results.

2.4. Multi-variate analysis based sensitivity enhancement

This section discusses the MVA-based strategy to analyse the SM Higgs boson decay into two taus. As already mentioned, the undetectable neutrinos that are part of the tau decay and the reconstruction of the tau leptons make the analysis of this decay channel non-trivial. MVA-based strategies are therefore used for the enhancement of the signal purity. The analysis uses data acquired by the CMS detector during 2016, 2017 and 2018 at a center of mass energy of $\sqrt{s} = 13\,\text{TeV}$. For the selection of the data, a combination of selection criteria are applied to reduce the impact of the background processes described in section 2.3.2. They consist of requirements on electrons, muons, and hadronically decaying tau leptons, such as a minimal distance to the primary interaction point (vertex) for the reconstructed tracks. Additionally, combinations of reconstructed signals at trigger level (cross-triggers) are exploited to further increase the signal acceptance. The event selection is performed in the same way as for [25] to ensure comparable results. A complete list of triggers used for the data acquisition can be found in [23]. The selection is applied to the recorded as well as the simulated data.
In the next step of the analysis, NNs are trained for the four final states of the di-tau system measured in the detector. The task of the NN is to separate background from signal. The output of the NN is then used to create histograms for further analysis.
After filling the histograms, systematic uncertainties can be applied to the histograms as discussed in section 2.3.3. The statistical inference is then based on the histograms in form of binned likelihood fits and hypothesis tests. This is discussed in section 2.4.2. A sketch of the analysis work flow can be seen in figure 2.10.

2.4.1. Classification of processes with NNs

The classification of each event into one of the processes mentioned above is done by a multi-classification NN. The architecture of the NN was chosen to be a feed-forward NN: Two hidden layers with 200 nodes each and hyperbolic tangent activation functions serve as the basis. The last activation function is a softmax activation function for process categorization. For the training, the chosen loss function is a categorical cross entropy function:

$$L_{\text{CE}} = -\sum_{n=1}^{N}\sum_{i} p_i \log\left(y_i\right) \tag{2.2}$$

where N is the batch size – meaning the number of events processed before the weights are updated –, p_i is the target and y_i is the output given by the network. Backpropagation is performed using the adaptive momentum estimation (ADAM) optimizer [26] with a learning rate of 10^{-4}. The ADAM optimizer calculates a learning rate for each parameter individually depending on the previous gradient of the function. The global learning rate 10^{-4} hereby serves as a maximum learning rate for the backpropagation. The weights are initialized using the Glorot initialization technique [27]. In order to minimize effects caused by overtraining, two regularization techniques are used simultaneously: Firstly, L2 regularization is applied [28]. This adds a penalty term to the loss function which reduces weights that are too large and allows smaller weights to have a larger influence on the classification. Secondly, the dropout technique is used in which a random number of nodes for each training step are deactivated to artificially decrease the size of the NN [28]. This guarantees a better generalization of the NN. Since the recorded data does not have any labels and to avoid bias in the NN, simulated data was used to train them. To simulate the fact that certain physics processes are much more likely to appear in the recorded data than others due to different cross section, each event was additionally weighted with a factor corresponding to the associated cross section of the event relative to each other. The weight was applied to the loss function to scale the loss according to the event weight w_n:

$$L_{\text{CE, weighted}} = -\sum_{n=1}^{N} w_n \sum_i p_i \log\left(y_i\right) \tag{2.3}$$

As certain cross sections and especially the cross section of signal events are relatively small compared to other processes, this makes the training dataset highly imbalanced. One could theoretically add a larger number of events for the processes which have a smaller cross section (and therefore weight). This, however, would cause another caveat: With a different number of events for each process it could happen that the NN trains mostly on the events which are most prevalent in the dataset. To mitigate this effect, a new method was introduced when creating a batch for a training step: The batches were created in such a way that from all classes the same number of events would be present in each batch. This way it can be guaranteed that all classes are fairly represented in the training and thus recognized by the NN. The number of samples per class is chosen to be 30. Additionally, the weights w_n of each event for a given output class within each batch are summed up and the inverse of the sum is applied to all events of this output class as a class weight w_c. This guarantees that categories with generally small event weights are also taken into account by the NN. The input variables of the NN are a mix of high-level inputs such as the fully reconstructed mass of the di-tau system [29] and low-level inputs such as the transverse momentum of a particle in the final state. In section 3.2, a method is described to effectively select those variables which have a high impact on the NN output. Before the variables are applied to the NN, a preprocessing algorithm is used to scale the range of the variables onto the acceptance range of the activation functions. The formula used is $(x - \mu)/\sigma$ with a mean μ and standard deviation σ for each variable in the dataset. This guarantees that the values are all within $[0, 1]$. Missing values are set to -10. This way, they cannot be confused by the NN with non-missing values. The training is monitored with an independent validation set of simulation. The training is stopped if the loss on the validation set did not improve within 50 epochs. An epoch was defined to be 1000 backpropagation steps.

As a softmax function is used in the final layer of the NN, signal and background processes are split into multiple independent output classes. The softmax activation function can be interpreted as the probability of an event belonging to a certain output class [28]. The highest output score for each event is then used to classify each event and fill histograms

based on the value of the given output score. Taking only the highest score is a necessary requirement to prevent events from being double counted in the statistical inference of the signal. A more in-depth explanation of NNs and especially the NNs used in the analysis are given in [30].

It should be noted that – in the current form of the analysis – the systematic uncertainties described in 2.3.3 are only applied after the NN was trained. This does not directly introduce an error, because the NN itself does not introduce a systematic uncertainty itself and simply propagates the existing ones. In general the NN can simply be seen as a function that combines all the lower-level input variables to higher-level variables. Of course this only applies if the training of the NN is done before the actual analysis and the NN is frozen afterwards. This is further discussed in [30].

While the NN does not introduce any new systematic uncertainties into the analysis, it is also not aware of any already existing ones. The NN is solely trained on the nominal data. Depending on the value and influence of the systematic uncertainties, this could have negative effects on the classification of the NN. If, for example, the most significant variable for the NN is a variable that indeed does have a strong separating power in the absence of systematic uncertainties, it could be thinkable that exactly this variable on which the NN heavily relies also has a very high systematic uncertainty. This would lead to a large amount of falsely classified events of the NN if the systematic uncertainty is then applied to the actual analysis. In contrast, an NN that is aware of systematic uncertainties can reduce the danger of using unreliable variables for the classification. In section 4, an already known technique is shown to reduce the dependence of an NN on variables with systematic uncertainties. Afterwards a novel technique is introduced to achieve the same reduction and this new technique is applied to an example of high energy physics.

2.4.2. Statistical inference

After filling the histograms for each potential signal and background process with the output of the NN, the statistical inference is performed on those histograms in form of binned likelihoods fits and in form of a hypothesis test where the hypothesis that a signal is present in the data (signal+background hypothesis) is tested against the hypothesis that only background events were measured (background-only hypothesis). In general, such tests can be done by comparing the likelihood functions $\mathcal{L}_{\mathrm{s+b}}$ and $\mathcal{L}_{\mathrm{b}}$ of each corresponding hypothesis against each other. The likelihood functions are constructed using Poisson distributions $\mathcal{P}$:

$$\mathcal{L}_{\mathrm{s+b}} = \prod_{i}^{N} \mathcal{P}(d_i|s_i + b_i) \prod_{j}^{n_i} \frac{s_i \cdot S_{ij} + b_i \cdot B_{ij}}{s_i + b_i} \tag{2.4}$$

$$\mathcal{L}_{\mathrm{b}} = \prod_{i}^{N} \mathcal{P}(d_i|b_i) \prod_{j}^{n_i} \frac{b_i \cdot B_{ij}}{b_i} \tag{2.5}$$

where N is the number of independent measurements, d an observation, s and b are expectations of signal and background respectively and n are the number of events in the data. S and B are probabilities to find a given event in a certain bin for signal and background.
As we are interested in whether the observed Higgs boson is the predicted SM Higgs boson, it is much more practical to measure the deviation of the observed signal with respect to the expectation given by theoretical predictions. Given the measured cross section σ and the predicted cross section σ_{SM}, a *signal strength modifier* μ can be defined as

$$\mu = \frac{\sigma}{\sigma_{\mathrm{SM}}} \tag{2.6}$$

This signal strength modifier is used to scale the signal s in equation 2.4:

$$s \to \mu s \tag{2.7}$$

Systematic uncertainties are introduced in form of nuisance parameters $\boldsymbol{\theta} = \{\theta_k\}$. The log-normal probabilities on one hand modify the yield of either signal or background directly through an additional term:

$$b_i \to b_i \cdot f(\theta_i, \sigma_i, x) = \begin{cases} \frac{b_i}{\sqrt{2\pi x \sigma_i}} e^{-(\ln(x)-\theta_i)^2/\sigma_i^2} & x > 0, \\ 0 & x \leq 0 \end{cases} \tag{2.8}$$

where θ_i is the best fit value for the yield with uncertainty σ_i. The same can be applied to the signal expectation s_i. Sampling is done by evaluating the test statistic with a randomly chosen x. Shape uncertainties on the other hand are applied via a morphing algorithm which alters the shape of the histograms of the NN [31]. Applying both μ and $\boldsymbol{\theta}$, the likelihood can be written as:

$$\mathcal{L}_{\mathrm{s+b}} = \mathcal{L}(d|\mu, \boldsymbol{\theta}) = \prod_i^N \mathcal{P}(d_i|\mu \cdot s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta})) \prod_j^{n_i} \frac{s_i \cdot S_{ij} + b_i \cdot B_{ij}}{s_i + b_i} \tag{2.9}$$

$$\mathcal{L}_{\mathrm{b}} = \mathcal{L}(d|0, \boldsymbol{\theta}) = \prod_i^N \mathcal{P}(d_i|b_i(\boldsymbol{\theta})) \prod_j^{n_i} \frac{b_i \cdot B_{ij}}{b_i}. \tag{2.10}$$

The likelihood is now dependent on the signal strength modifier μ and $\boldsymbol{\theta}$. In fact, the actual values of $\boldsymbol{\theta}$ are of no particular concern to the final result as long as the value can be considered reasonable. Thus, a test statistic focused on the actual parameter of interest (POI) μ is used. This test statistic corresponds to a *profile likelihood ratio*, defined as

$$q_\mu = -2 \ln \frac{\mathcal{L}(d|\mu, \hat{\theta}_\mu)}{\mathcal{L}(d|\hat{\mu}, \hat{\theta})}, \quad 0 \leq \hat{\mu} \leq \mu. \tag{2.11}$$

where $\hat{\theta}_\mu$ is the estimate of $\boldsymbol{\theta}$ which maximizes $\mathcal{L}$ for a given μ and $\hat{\mu}$ and $\hat{\theta}$ are the best fit values for each parameter when both parameters are fitted at the same time. The boundary conditions enforce a one-sided boundary, excluding negative signal strength modifiers. Equation 2.11 can be scanned for different values of μ. Lower values for q_μ corresponds to a better agreement of the signal+background hypothesis with the observation. An example for a profile likelihood scan can be seen in figure 2.11. The minimum of the resulting function is the best estimate for the signal strength modifier μ_{best}.
In addition to the best estimate for μ, the profile likelihood scan is also used to derive a statistical statement in form of a p-value p at which the null hypothesis (meaning the background-only hypothesis) can be discarded. The p-value denotes the probability at

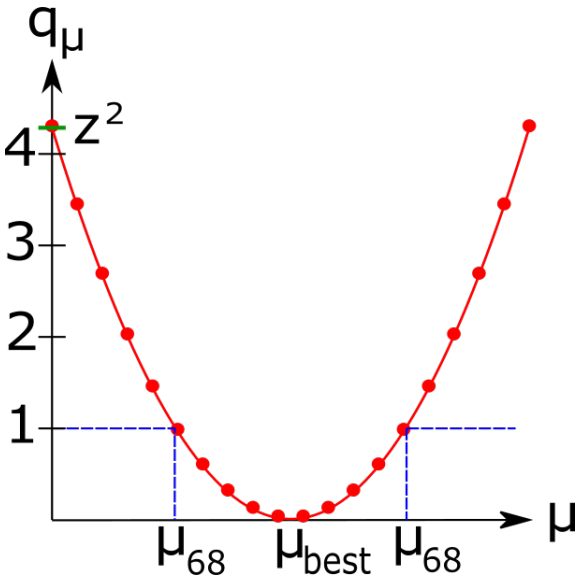


Figure 2.11.: Example of a profile likelihood scan. μ_{best} is the value for μ for which $q = 0$ while the upper and lower bound of μ_{best} in the 68% confidence interval can be calculated from the difference between μ_{best} and both μ_{68}'s. The intersection of the parabola with the y-axis is z^2 from which the p-value can be calculated.

which the observation occurs given the background-only hypothesis. It is calculated by integrating the probability density function (PDF) $f(q_\mu|\mu, \hat{\theta}_\mu)$ of the test statistics from minus infinity to the observed q value:

$$p_{\text{b}} = \int_{-\infty}^{q_{\text{obs}}} f(q_\mu|\mu, \hat{\theta}_\mu)\text{d}q \tag{2.12}$$

If the p-value is smaller than a predetermined critical value of α, the background-only hypothesis is excluded with a confidence level of $1 - \alpha$. The p-value is usually expressed in terms of quantiles of a normalized Gaussian distribution in units of σ. It is convention to claim an observation if the confidence level is above $5\,\sigma$, which means that the likelihood of the observation occurring given the background-only hypothesis is less than 2.87×10^{-7}. In the limit of large statistics, the test statistic q follows a χ^2 distribution [32]. In this case, the p-value can be determined by evaluating the test statistic at $\mu = 0$:

$$q_{\mu=0} = z^2 \tag{2.13}$$

The result z is evaluated in quantiles of a normalized Gaussian distribution. The square root of a value of e.g. $z^2 \approx 4.2$ as shown in figure 2.11 is thus equivalent to $p_{\text{b}} \approx 2.05\,\sigma$. Again, an observation can be claimed if the significance is above $5\,\sigma$.

3. Input variables for multi-variate analyses

A crucial component in the analysis strategy explained in section 2.4.2 is the NN that separates signal from background events. The samples used to train the NN contain a multitude of potential input variables [30]. Simulated data was used for the training of the NN. Section 3.1 gives an explanation why simply using all variables contained in the dataset might not be desirable and why the correct selection of input variables is critical for an effective classification of the data. Section 3.2 will then propose a method to reduce the number of input variables (pruning) based on Taylor coefficients [1] and section 3.3 shows the results of the pruning and the comparison of the signal strength constraints w.r.t. an unpruned set of input variables. The last section 3.4 will show an additional method that could only be used due to the previous pruning efforts.

3.1. Reason for pruning

Simulation describes the data in the best possible way. Nevertheless it cannot be guaranteed that the simulation perfectly captures all features in data. It is also possible that the simulation has additional features that are not present in data which might introduce bias to the analysis. These caveats are further amplified in the presence of systematic uncertainties as the simulation not only has to capture all features of the nominal dataset, but also all features that might be present in systematic variations. As such it is necessary to validate all input variables used for the classification of signal and background and confirm that simulation and data are in congruency. An established way to quantify the agreement between simulation and data is the saturated goodness of fit (GoF) test [34]. The saturated GoF test is comparable to the χ^2 test [30]. In fact, the only - but very important - difference between the χ^2 test and the saturated GoF test is the normalization, which provides a meaningful scale of the resulting test statistic. The strongest point of the saturated GoF test is its ability to consider all systematic and statistical uncertainties by bootstrapping from the distributions of the systematic shifts. As described in section 2.3.3 there are many systematic uncertainties to be considered for the test and it is essential to take those into account. In order to make a statistical statement, a p-value can be extracted from the GoF test, which in this case is the probability that the observed data can be explained by the model prediction. With this p-value, a threshold can be defined and variables below the defined threshold can either be discarded or an effort can be made to improve the description of the variable in simulation to bring it closer to the data. An example of a 1D and 2D GoF test can be seen in figure 3.1 and 3.2.

If GoF tests are used to verify the input variables, the problem of a large number of input variables becomes apparent: Using 30 input variables per year and final state would already result in 360 1D tests without taking correlations between the variables into account. A large number of those 30 input variables might imply correlations to each other as e.g.

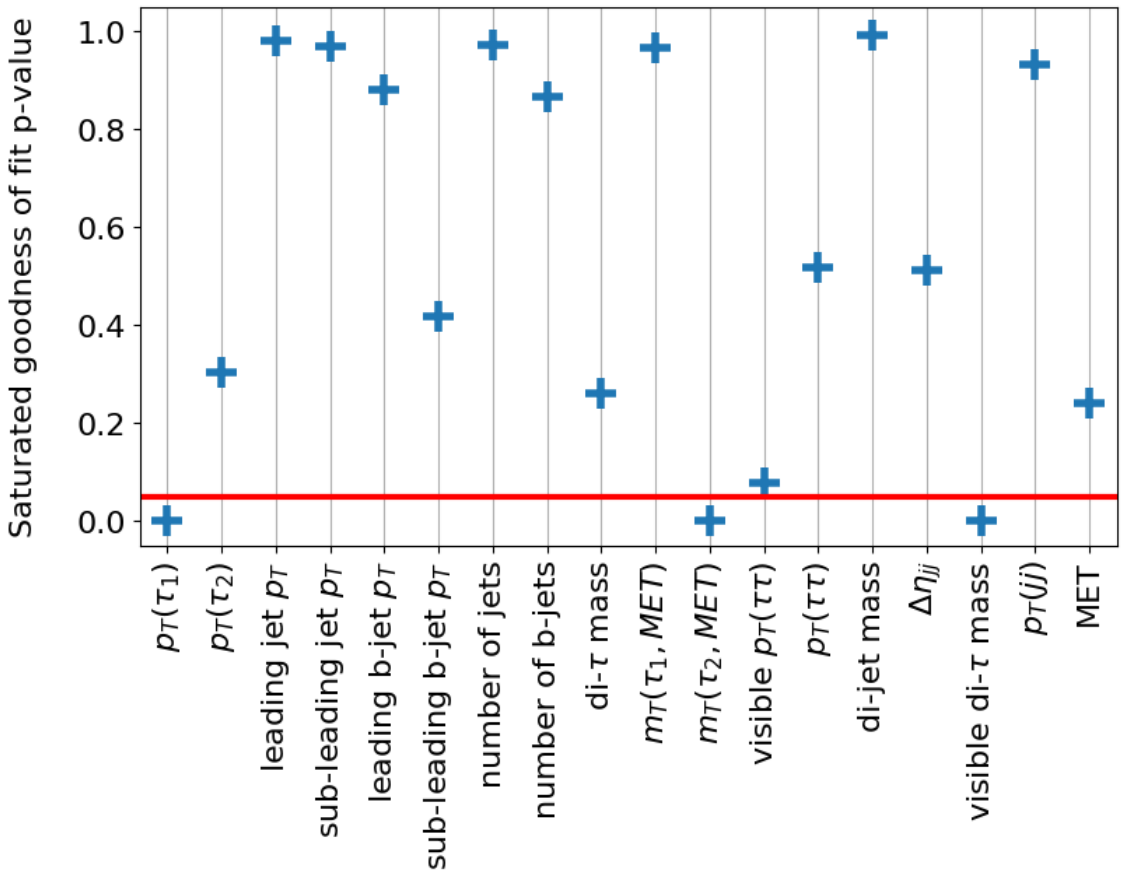


Figure 3.1.: Shown are the results of the one-dimensional GoF test for the input variables of an NN used in a previous $H \to \tau\tau$ analysis [33]. The red line shows the threshold for the p-value for which input variables were previously dropped. Only variables greater than this value were considered as inputs for the NN.

for the reconstructed mass of the visible components and the fully-reconstructed di-tau mass. The simulation also has to capture such correlations. Thus, it would be reasonable to at least consider the two-dimensional GoF test as well. For N input variables, the number of GoF tests needed to quantify all correlations between all variables corresponds to $(N^2 - N)/2$ tests. With 30 input variables, 3 separate years of data taking and 4 final states per year, this would result in 5220 GoF tests for $N = 2$, resulting in a huge computational effort. This is amplified by the fact that every time the modeling of the data is improved, the GoF tests have to be redone. This example makes it also clear that GoF tests for even higher orders would take an unreasonable amount of time and are therefore not considered. With this, there are three main reasons to reduce the number of input variables for the NN:

- Having a smaller number of input variables from the beginning greatly reduces the amount of computational effort needed to verify the input variables.
- If a variable, which is fully correlated to another variable, is added to the input space of the NN, the convergence of the training of the NN might become less stable while the efficiency of the NN will not improve.
- Additional variables increase the chance of having a variable in the input space which might still have slight misdescriptions in simulation which the GoF test did not capture. While this effect might be low for a single variable, this effect can add up with the addition of more variables.

An additional and important reason for the pruning of the input variables to the NN was

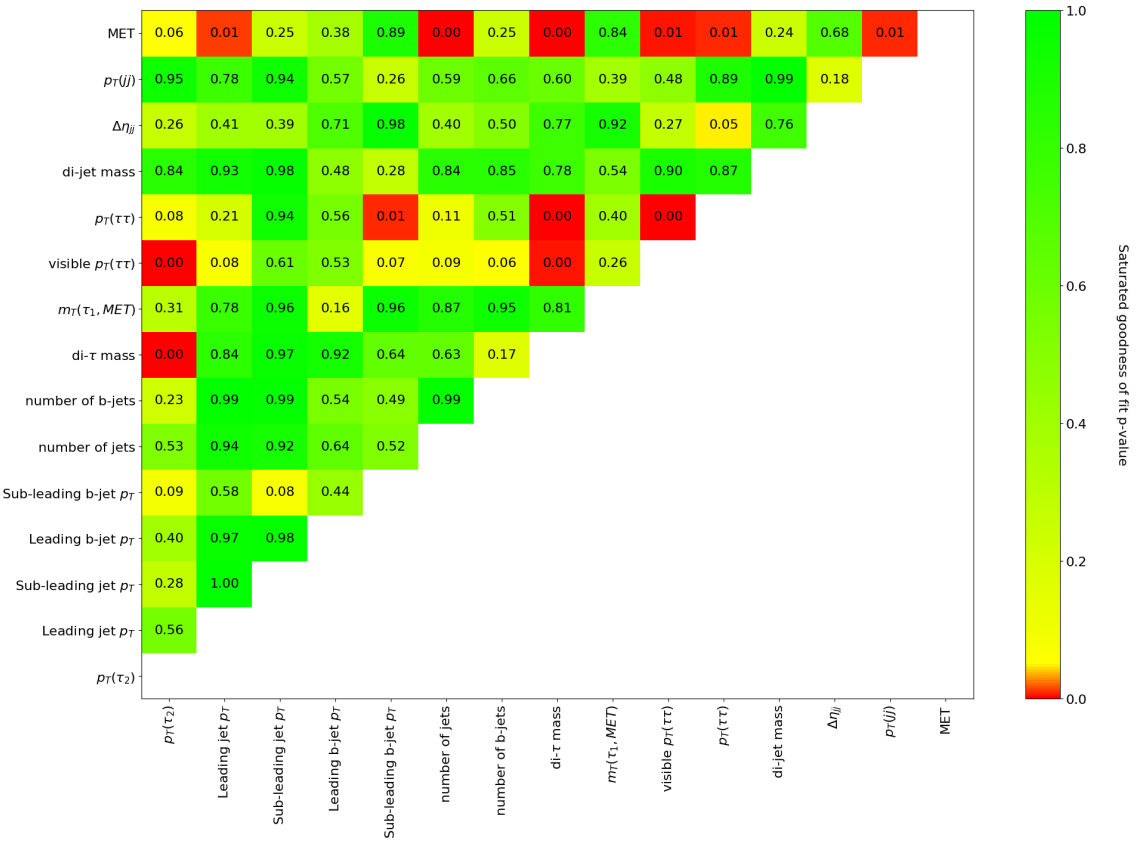


Figure 3.2.: Shown are the results of the two-dimensional GoF test for the input variables of an NN for the $\mu\tau$ final state used in a previous $H \to \tau\tau$ analysis [33]. If the p-value of a variable is below a threshold as indicated by the red color of the corresponding box, the variable was dropped from the analysis.

the unification of the variable sets per year and eventually per final state. For the analysis of $H \to \tau\tau$ events, 3 years of data were provided: 2016, 2017 and 2018. Each year has its own separated set of data. In the previous analysis [33], a NN was trained for each year with only the data from this year and a set of input variables mostly unique to this year. While this makes sense considering the results of the GoF tests for those years and from a pure machine learning point of view, choosing a different variable set for each year lacks motivation by physics. In general, the most significant variables to separate signal from background should not be dependent on the year from which the data was taken. While we do expect differences between each year, mostly due to changes in triggers, detector quality and luminosity, we would not expect a significant change of variables. Thus, the pruning of the input variables was used as a chance to unify the set of input variables across all years, which is not only more appropriate for a physics analysis but also further reduces the complexity of the task. In addition, it allows to train a single NN per channel across all years as later described in section 3.4.

3.2. Method of pruning

As described in the previous section, most input variables are correlated across each other. Those higher order correlations between input variables turn the pruning of variables non-trivial. A deceptively simple approach would be to test every combination of variables to find the combination of variables with a good result on the analysis objective while simultaneously having a relatively low number of variables. This approach is unfeasible in a reasonable amount of time due to the large amount of possible combination of input

variables. Instead of testing every possible combination of input variables, Taylor coefficients are used as described in [1]. The NN function can be approximated using a Taylor expansion in its input variables. The resulting Taylor coefficients are indicators of the importance of a variable for the NN, as larger values correspond to larger contributions to the overall output value. The actual value of the Taylor coefficient for an input variable depends on the output class. For each output class a different Taylor coefficient can be calculated. For each final state and output class, a ranking of Taylor coefficients was produced from a network that was trained on all 29 input variables available at the time. The ranking implies that the input variable with the highest coefficient as a single variable has the highest influence on the output of the NN. In general, the Taylor coefficients can be extended into k-dimensions, were k is the number of possible correlations between input variables [1]. In this pruning method, only the first dimension was considered, and no correlations between input variables were taken into account for the importance ranking. As it would be unclear which variables to add if a correlation has a large Taylor coefficient, a simple ranking could not be made when higher dimensions would have been considered as well. In this sense, the ranking should only be considered an approximation and not a definite rank of importance. The output classes for an NN of a final state are the signal processes for stage 0 binning and combinations of background processes given in section 2.4. The number of output classes for each NN of a final state varies between 5 and 8 according to the number of background processes considered for this final state. With a total of four final states, 28 of the Taylor rankings were produced. For each of those, the following training procedure was started:

1. Starting with the most significant variable in the ranking, the NN was trained until convergence.
2. The trained NN was tested on an independent test set.
3. Afterwards the next highest ranking variable was added to the NN.
4. The first 3 steps are repeated until all variables of the ranking have been added successively.

With a total of 29 input variables per final state and a two fold training, between 290 and 464 NNs were produced and tested for each final state.

To compare the impact of adding a variable, the number of true positive T_p, false positive F_p and false negative F_n predictions as well as the efficiency ϵ, purity ρ and the F1 score given by equation 3.1 were calculated after each training and plotted in sequence for each added input variable.

$$\begin{aligned} \epsilon &= \frac{T_p}{T_p + F_n} \\ \rho &= \frac{T_p}{T_p + F_p} \\ F_1 &= 2 \cdot \frac{\rho \cdot \epsilon}{\rho + \epsilon}, \end{aligned} \tag{3.1}$$

Each event in the calculation of the efficiency and purity was weighted by its event weight and a class weight to address the different cross sections of each corresponding process. In figure 3.3 one such graph is shown for the output class `qqh` and the $\mu\tau_h$ final state. From this graph the most important variables can be determined by finding the point of saturation for a given output class. In this example it can be concluded that the first 12 variables are most important for the output class `qqh` and all subsequent variables do not add to the separation between background and signal. While the F1-scores for each

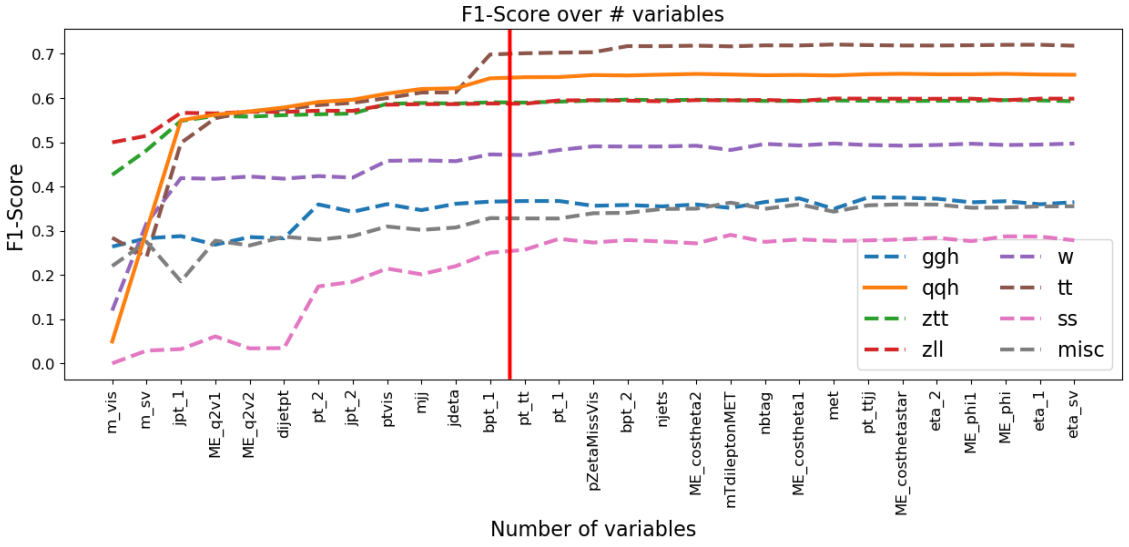


Figure 3.3.: F1 score in sequence of the added input variables. The ranking for the $\mu\tau_h$ final state and output class `qqh` is shown. Here the line for `qqh` (solid orange) is most important. One can see that the F1 score has converged after 12 variables have been added. Adding more variables does not seem to improve the classification for `qqh` any further.

other output class are also shown, no conclusion can be drawn for those output classes as the ranking of the Taylor coefficients were made from the NN function corresponding to the `qqh` output class. For each class the F1-score has been evaluated separately. This procedure of finding the saturation point can now be repeated for all output classes. In general, a perfect classifications of the background classes are of no particular interest as long as the signal classes, denoted by `ggh` (production of Higgs boson via gluon fusion) and `qqh` (production of Higgs boson via VBF) respectively, do not suffer from miss-classified background processes.
Therefore the graphs of the signal classes are the focus for the pruning. By evaluating those F1-score graphs, preliminary sets of variables have been determined that have been further processed by evaluating the confusionen matrices at certain points in the graphs. In the example shown, there is almost no improvement e.g. between the input variables `mjj` and `jdeta` (the physics meaning of each variables is explained in table 3.1). To determine the usefulness of adding those variables to the NN, the confusion matrices of the networks at those points are compared with each other. Only if an improvement for the signal classes can be seen in the confusion matrices (e.g. lower migration from background classes to the signal classes), the variable is considered for the final set of variables. In case of ambiguous behaviour of an NN, e.g. a variable was determined to be useful in the first, but not in the second fold, the graphs of the background class were also considered when choosing the variables.
After this third step, two sets of variables were defined for each final state: A core set containing the most influencial variables and an extended set that can be used to gain further improvement with the disadvantages described in section 3.1.

Evaluation of the pruning

In a first evaluation of the pruning, the confusion matrices of this pruned set were compared to the confusion matrices of the full set of variables. If the confusion matrices were almost the same, e.g. a difference of ≤ 0.03 for all categories, the signal strength constraints for stage 0 binning and inclusive binning were calculated as the final step of the procedure. The signal strength constraints of the NN with pruned variables and the NN containing all

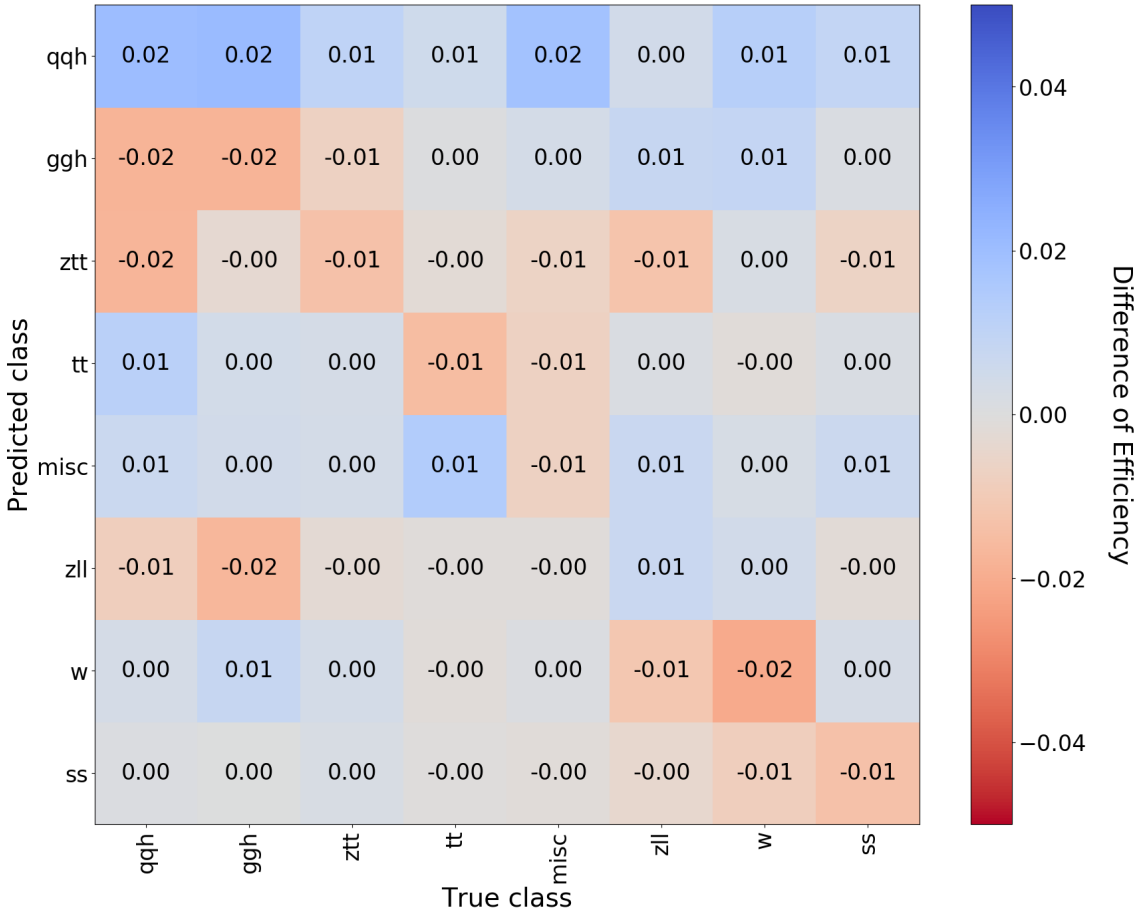
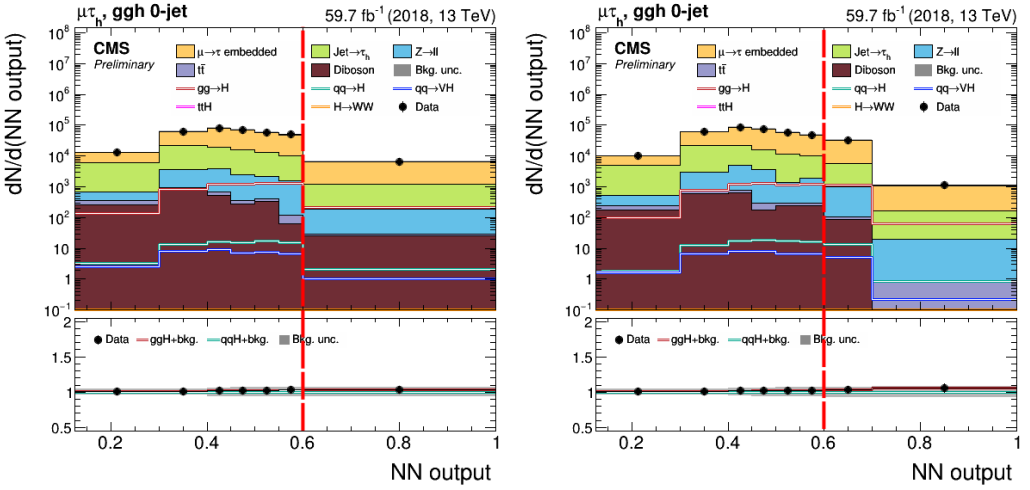


Figure 3.4.: The matrix shows the difference in efficiency per class for a pruned set of variables containing 11 variables and the full set of variables containing 29 variables for the $\mu\tau$ final state in 2017. A negative score indicates a loss in efficiency while a positive score indicates a better efficiency. The difference is no more than 0.02 for all classes and 0.00 in most cases. Nevertheless a difference in signal strength constraints could be measured, indicating that confusion matrices are not a completely reliable indicator for signal strength constraints.

possible variables were compared with each other. The pruning was considered successful if both the upper and lower bound of the signal strength constraints for the inclusive binning were within 10 % of the previous constraints containing all possible variables in the 68 % confidence interval (CI).

It should be noted that both confusion matrices and F1 scores are no clear indicator of the final signal strength constraint which also depends on the form of the NN distribution. An example for this in the $\mu\tau$ final state with data taken from 2017 can be seen in figure 3.4. The confusion matrix shows the difference between the pruned set of variables and the reference NN containing 29 variables. The difference is clearly no more than ± 0.02 for all categories. The pruned NN is sometimes even better then the reference NN. Nevertheless the signal strength constraints calculated are 8 % worse for the pruned set of variables as seen in table 3.3. The final signal strength constraints are described in the form of a binned profile likelihood fit. The likelihood for each bin is calculated separately, meaning that having highly pure bins is more important than just reaching a certain threshold for the classification, i.e. having a higher output value than all other classes. Figure 3.5 shows the post fit distribution for both NNs for one output class. It is clearly visible that the distribution of the full set of variables is sightly shifted towards higher NN scores. The NN has a higher number of events for very high NN scores, thus producing more pure



(a) Pruned set **(b)** Full set

Figure 3.5.: The two graphs show the post fit distribution for the $\mu\tau_h$ final state for 2018, for the `ggh 0-jet` signal category. The fit is done using an Asimov dataset. The left plot was made using an NN containing 11 variables and the right plot used an NN containing 29 variables. The number of bins for each plot is chosen algorithmically to guarantee a minimum amount of events in each bin. The plot with the full set of variables has a higher number of events for NN scores > 0.6, which results in an additional bin being used for the calculation of the signal strength constraints. This difference leads to a better signal strength constraint.

bins, which tightens the signal strength constraints in return. While the signal strength constraints are the best indicator whether the performance of the pruned NN is comparable to the reference NN, calculating the signal strength constraints for all NNs produced during the pruning procedure explained above would take an unreasonable amount of time. For this reason, the signal strength constraints could only be used as a final indicator and not as the main metric for this pruning method.

3.3. Result of pruning and validation

As mentioned previously, there are 3 datasets for each year which needed to be pruned. The pruning described in section 3.2 was first performed on data from 2017. The complete set of input variables available for the training can be found in table 3.1. After the pruning method was applied to 2017, for each final state, a core and extended set of variables was defined. After further examination, the defined set of variables was approximately the same for all but the $e\mu$ channel. As variables such as the transverse visible mass or the transverse momentum are general indicators of a Higgs boson decay and should not depend on the $\tau\tau$ final state, it is to be expected to have a large overlap of variables across final states. Thus the variable sets for the $\mu\tau_h$, $e\tau_h$ and $\tau_h\tau_h$ final states were synchronized to obtain a homogeneous set if input variables across years and final states. This was done at the cost of only adding one or two variables per final state in order to achieve a consistent set across all final states but $e\mu$. The $e\mu$ final state takes a slightly larger set of variables to achieve comparable results with the full set of variables. Because of that, no extended set was defined for this final state. The complete pruned set of variables for all final states can be found in table 3.2. As the initial goal of the pruning was to align the number of variables used per year, the core set and extended set derived from the 2017 data has been applied to the 2018 and 2016 datasets. The results were again compared to the signal strength constraints given by a full set of variables. As the signal strength constraints for the core set based on inclusive binning was within 10 % of the signal strength constraints

Table 3.1.: Shown are all variables that were considered with identifier and description. The right hand side shows which of those variables were used in the previous analysis for the year 2017 (2016). The 1 in an identifier always refers to the leading lepton or jet, while the 2 is the subleading lepton or jet. `pZetaMissViss` and `mTdileptonMET` are explained in more detail in [35]. The variables used in the Matrix Element Likelihood Analysis (MELA) are explained in more detail in [36, 37, 38].

Identifier	Description	Variable Selection			
		$\mu\tau_h$	$e\tau_h$	$\tau_h\tau_h$	$e\mu$
`pt_1`	Lepton transverse momentum	—(✓)	—(—)	✓(✓)	—(✓)
`pt_2`		✓(✓)	✓(✓)	—(✓)	—(✓)
`jpt_1`	Jet transverse momentum	✓(✓)	✓(✓)	—(✓)	✓(✓)
`jpt_2`		✓(✓)	✓(✓)	✓(✓)	✓(✓)
`bpt_1`	b-jet transverse momentum	✓(✓)	✓(✓)	✓(✓)	—(—)
`bpt_2`		✓(✓)	✓(✓)	✓(—)	—(—)
`njets`	Number of jets	✓(✓)	✓(✓)	✓(—)	✓(✓)
`nbtag`	Number of b-tags	✓(✓)	✓(✓)	✓(✓)	—(—)
`m_sv`	Fully reconstructed mass of the di-tau system [29]	✓(✓)	✓(✓)	✓(✓)	✓(✓)
`ptvis`	Visible transverse momentum	—(✓)	✓(✓)	—(✓)	✓(✓)
`pt_tt`	Ditau transverse momentum	✓(✓)	✓(✓)	✓(✓)	✓(✓)
`mjj`	Mass of dijet system	✓(✓)	✓(✓)	✓(✓)	✓(✓)
`jdeta`	Difference in pseudorapidity for dijet system	✓(✓)	✓(✓)	✓(✓)	✓(✓)
`m_vis`	Reconstructed mass of the visible di-tau system	—(✓)	—(✓)	✓(✓)	✓(✓)
`dijetpt`	Transverse momentum of dijet system	✓(✓)	✓(✓)	✓(✓)	✓(✓)
`met`	Missing transverse energy	—(✓)	✓(✓)	—(✓)	✓(—)
`eta_1`	Pseudo rapidity of the lepton or jet	—(—)	—(—)	—(—)	—(✓)
`eta_2`		—(—)	—(—)	—(—)	—(✓)
`pt_ttjj`	Ditau transverse momentum of jets	—(—)	—(—)	—(—)	✓(—)
`pZetaMissViss`	$e\mu$ specific variable	—(—)	—(—)	—(—)	✓(✓)
`mTdileptonMET`	$e\mu$ specific variable	—(—)	—(—)	—(—)	✓(✓)
`eta_sv`	SVFit pseudo rapidity	—(—)	—(—)	—(—)	—(—)
`ME_costheta1`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_costheta2`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_phi`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_phi1`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_q2v1`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_q2v2`	MELA specific variable	—(—)	—(—)	—(—)	—(—)
`ME_costhetastar`	MELA specific variable	—(—)	—(—)	—(—)	—(—)

Table 3.2.: Proposed variables to be used for the years 2016, 2017, and 2018. The core set is denoted by checkmarks (✓) and can be considered obligatory. The extended set is denoted by an "e" and can be used in addition to the core set. The variables are in no particular order

Identifier	Description	Variable Selection			
		$\mu\tau_h$	$e\tau_h$	$\tau_h\tau_h$	$e\mu$
pt_1	Lepton transverse momentum	✓	✓	✓	✓
pt_2		✓	✓	✓	✓
jpt_1	Jet transverse momentum	✓	✓	✓	✓
njets	Number of jets	✓	✓	✓	✓
nbtag	Number of b-tags	✓	✓	✓	✓
m_sv	SVFit mass	✓	✓	✓	✓
ptvis	Visible transverse momentum	✓	✓	✓	✓
mjj	Mass of dijet system	✓	✓	✓	✓
jdeta	Difference in pseudorapidity for dijet system	✓	✓	✓	✓
m_vis	Visible mass	✓	✓	✓	✓
dijetpt	Transverse momentum of dijet system	✓	✓	✓	✓
ME_q2v1	MELA specific variable	e	e	e	✓
ME_q2v2	MELA specific variable	e	e	e	✓
jpt_2	Jet transverse momentum	e	e	e	✓
pt_tt	Ditau transverse momentum	e	e	e	—
eta_1	Pseudo rapidity	—	—	—	✓
mTdileptonMET	$e\mu$ specific variable	—	—	—	✓
bpt_1	b-jet transverse momentum	—	—	—	✓

Table 3.3.: Relative comparison of signal strength constraints s based on inclusive binning for the core set c and extended set e given in table 3.2 w.r.t. the full set f given in table 3.1. The signal strength constraints were calculated using an Asimov dataset. The comparison was calculated using the formula $s_f/s_{c,e} - 1$.

2016				
Channel	Relative comparison in %			
	Core		Extended	
	upper bound	lower bound	upper bound	lower bound
Combined	0.07	0.08	0.04	0.04
$\mu\tau_h$	0.10	0.09	0.05	0.05
$e\tau_h$	0.10	0.10	0.05	0.05
$\tau_h\tau_h$	0.02	0.03	0.02	0.02
$e\mu$	0.05	0.04	0.05	0.04
2017				
Channel	Relative comparison in %			
	Core		Extended	
	upper bound	lower bound	upper bound	lower bound
Combined	0.06	0.06	0.04	0.04
$\mu\tau_h$	0.08	0.08	0.04	0.04
$e\tau_h$	0.07	0.07	0.06	0.05
$\tau_h\tau_h$	0.04	0.03	0.03	0.03
$e\mu$	0.08	0.08	0.04	0.04
2018				
Channel	Relative comparison in %			
	Core		Extended	
	upper bound	lower bound	upper bound	lower bound
Combined	0.06	0.06	−0.02	−0.01
$\mu\tau_h$	0.06	0.06	0.00	0.00
$e\tau_h$	0.09	0.09	−0.04	−0.01
$\tau_h\tau_h$	0.03	0.03	0.03	0.03
$e\mu$	0.02	0.03	0.00	0.01

of the full set of input variables, the core and extended set of variables from 2017 could be used for the datasets of 2016 and 2018 as well. The exhaustive method described in section 3.2 was therefore not applied to the datasets of 2016 and 2018.
In summary, a consistent set of variables has been formulated across all years and most final states with $e\mu$ as an exception. The number of variables is reduced from 30 to 11/16 ("core set"/"extended set") variables. The number of two-dimensional GoF tests was reduced from 5220 for 30 variables to 1260 (1440) for the extended set for the $\mu\tau_h$, $e\tau_h$ and $\tau_h\tau_h$ ($e\mu$) final states, substantially reducing the complexity of the two-dimensional GoF test while maintaining a good constraint on the signal strength and unifying the years in terms of input variables.

A comparison of the Asimov signal strength constraints calculated using MC simulation can be seen in table 3.3. The relative change of the signal strength constraints upper and lower bound are given by the right-hand columns. A positive percentage corresponds to a deterioration while a negative percentage corresponds to an improvement of the core set w.r.t. the set containing 29 variables. All results are comparable within 10 % while the combined limit never has a larger difference than 0.08 %. A small gain in signal strength constraints can be seen when using the extended set, especially in the constraints of the

Table 3.4.: Relative comparison of signal strength constraints s calculated using Asimov datasets. Shown is the comparison between NNs i trained individually on the years (standard) and an NN c trained on all years simultaneously (conditional). Both networks were trained on the core set of variables. The comparison was calculated using the formula $s_i/s_c - 1$.

2016		
Channel	Relative comparison in %	
	upper bound	lower bound
Combined	−0.06	−0.06
$\mu\tau_h$	−0.04	−0.04
$e\tau_h$	−0.07	−0.08
$\tau_h\tau_h$	−0.03	−0.03
$e\mu$	−0.12	−0.13
2017		
Channel	Relative comparison in %	
	upper bound	lower bound
Combined	−0.02	−0.02
$\mu\tau_h$	−0.02	−0.02
$e\tau_h$	−0.01	−0.02
$\tau_h\tau_h$	−0.01	−0.01
$e\mu$	−0.01	−0.01
2018		
Channel	Relative comparison in %	
	upper bound	lower bound
Combined	0.00	0.00
$\mu\tau_h$	0.00	0.00
$e\tau_h$	0.04	0.03
$\tau_h\tau_h$	0.00	0.00
$e\mu$	0.00	0.00

$\mu\tau_h$ and $e\tau_h$ final states and for the 2018 dataset in general.

3.4. Conditional networks with aligned input variables

As the input variables and output classes of the NN are now the same for all years per final state, the only difference between the NNs are the datasets used for training. One could simply combine all datasets and train the NN on this combined dataset without any distinction between the years. In reality, the datasets are not entirely consistent across all years, e.g., different triggers were applied for the data selection, the luminosity of the datasets differ or the detector conditions changed between the years. A full list of selection criteria for all datasets can be found in [23]. Therefore a variable has been introduced to distinguish between the datasets of each year. This switch has been directly implemented into the NN in form of an additional `era` variable. The `era` variable acts as an identifier from which year a given event of a dataset is obtained from. There are two potential ways of implementing this `era` variable. In the continuous case, a single `era` variable is used which will directly get the integer value of the year, e.g. "2017" or – when re-labeling – a value between 0 and 2. The problem with this implementation is that `era` is a discrete variable. By feeding it to the NN as a continuous variable, it is implied to the NN that there might be values in between the given years such as "2016.67". While this could arguably make sense for other integer valued variables such as the number of b-tagged jets, this does not represent the reality for the `era` variable. Furthermore, the continuous case might imply a ranking to the NN, e.g. having an `era` value of 2 might be more signal-like than having

Table 3.5.: Shown is the comparison between NNs i trained individually on the years and an NN r trained on all years simultaneously but signal classes were additionally randomized in two different ways. "Randomization" refers to using all available signal classes but randomizing the `era` variable for each of them. "No 2016" means that all signal event classes of 2016 were removed and the eras of the remaining signal classes were randomized (including `2016` as an era). The comparison was calculated using the formula $s_i/s_r - 1$.

2016				
Channel	Relative comparison in %			
	Randomization		No 2016	
	upper bound	lower bound	upper bound	lower bound
Combined	−0.07	−0.07	−0.09	−0.10
$\mu\tau_h$	−0.05	−0.06	−0.08	−0.08
$e\tau_h$	−0.10	−0.11	−0.16	−0.17
$\tau_h\tau_h$	−0.04	−0.05	−0.06	−0.07
$e\mu$	−0.12	−0.14	−0.13	−0.14
2017				
Channel	Relative comparison in %			
	Randomization		No 2016	
	upper bound	lower bound	upper bound	lower bound
Combined	−0.02	−0.02	−0.06	−0.06
$\mu\tau_h$	−0.01	−0.02	−0.04	−0.05
$e\tau_h$	−0.01	−0.01	−0.10	−0.10
$\tau_h\tau_h$	−0.03	−0.03	−0.04	−0.04
$e\mu$	−0.01	−0.02	−0.08	−0.08
2018				
Channel	Relative comparison in %			
	Randomization		No 2016	
	upper bound	lower bound	upper bound	lower bound
Combined	0.01	0.01	−0.01	−0.01
$\mu\tau_h$	0.01	0.00	−0.01	−0.02
$e\tau_h$	0.04	0.03	−0.02	−0.03
$\tau_h\tau_h$	0.02	0.01	0.00	0.00
$e\mu$	0.01	0.02	−0.06	−0.06

an `era` value of 0. Another way of implementing this switch is by using so-called one-hot encoding. With one-hot encoding, each category of the discrete variable gets its own input variable that is either 0 or 1, depending on whether the event belongs to this category or not. In the case of eras, there are 3 additional input variables, each representing one era. Only one of this variables will be 1 while the others will be 0 for each event of the combined dataset. A study on different techniques used to encode categorical data can be found in [39]. After training, the signal strength constraints were calculated for each year independently using the `era` variables provided by the NN to differentiate between each year. A comparison between the combined conditional network and the networks trained on each year separately can be seen in table 3.4. The constraints on the signal strength improved by 13 % in the $e\mu$ final state of 2016 and in general across the 2016 and 2017 datasets, while being comparable within 5 % for the 2018 dataset. The additional and well described amount of signal data from the 2018 dataset improves the constraints of especially the 2016 samples which suffered from an insufficient amount of signal samples in this training run. This also indicates that the datasets of each year are very close to each other and can be used to improve the signal strength constraints on other years. In addition to this improvement, the added benefit of easier control due to having only a single network and the physics motivation behind it, the unification of the training are well taken into account.

Recovering signal by randomization

Taking the improvement of one dataset via another dataset of a different year to the extreme, an additional technique can be implemented when combining the datasets: In case there is no or an insufficient amount of signal samples available for a given year, the NN can be trained to supplement the missing information of this year by using information provided by datasets of other years. This way, the NN can recover the signal of the missing year. By randomizing the `era` variables of the signal category for each year, the NN can be deprived of the information to which year a signal event belongs to, thus it is forced to generalize the signal categories while still maintaining the specializiation for the background categories. In table 3.5 this technique was used to recover the signal of 2016 data, which - as mentioned already - suffered from an insufficient amount of signal samples. Two cases were studied in this regard:
First, all samples that were available for the signal classes of 2016 were used alongside the signal events of 2017 and 2018 and the `era` information was shuffled. The second example was made to be an extreme case: All signal events of 2016 were removed from the training dataset while the background samples of 2016 were kept. As explained in section 2.4.1, in order to mitigate the imbalance of the training samples, the NN uses a balanced batch approach in which each batch has the same number of events for each class. This balanced batch approach was extended for the training of the conditional neural network to include the years. Each batch contains the same number of samples for each class and year, effectively tripling the amount of events per batch compared to a network trained on a single year. The number of steps per epoch was tripled as well to ensure a good convergence. Removing the signal events from a year would cause the overall number of signal events to drop per batch which might introduce a bias in the NN training. To compensate the overall loss of signal samples per batch, each batch got an additional amount of 2017 and 2018 signal events to compensate the missing 2016 samples. Table 3.5 shows the result of the training on randomized eras for the signal events. It can be clearly seen that the signal strength constraints based on an inclusive binning for the 2016 data improves compared to signal strength constraints of NNs trained on all years separately without randomization. Completely removing the signal samples of 2016 even further improves the signal strength constraints for 2016 by up to 17 % in the $e\tau_h$ final state. This improvement is, however, unexpected. The signal strength constraints are even better than

the signal strength constraints calculated without randomization as seen in table 3.4 which should not be the case as there was less data used overall for the training and especially less data from 2016 which should describe the signal samples for this year the best even though the overall amount of signal samples is low. This indicates that the improvements are caused by artifacts in the dataset, e.g. the tau identifier is not as well described in 2016 as in 2017 or 2018. This could lead to an underestimation of the background in 2016 as this background was not randomized and would use the tau identifier of 2016 while the signal samples would use the better tau identifier of 2017 or 2018. The signal would be easier to separate from background due to this artifact in the dataset causing the signal strength constraints to be tighter.
While this techniques can be useful in future scenarios, they were not used for the current $H \rightarrow \tau\tau$ analysis [23]. It should be clear that the improvement in the signal strength constraints comes vastly from an insufficient amount of data or from data that is not described very well. As such, the focus should be on improving the amount and the description of data available for the analysis instead of bypassing the problem altogether with this randomization technique which could lead to wrong interpretations of the results. However, it was shown that this technique can recover signal events in the classification of an NN for a year with insufficient amount of data.

4. Systematic uncertainties in Neural Networks

As shown in the previous sections, NNs are well suited for data analyses in high energy physics. Besides the $H \to \tau\tau$ analysis explained in section 2.4, another example for the usage of NNs on data acquired in physics experiments is the classification of particle jets induced by heavy flavor quarks [40, 41]. Neural networks are capable of identifying non-trivial correlations among input variables even to higher orders turning them very robust and efficient in classification tasks. In fact it can be shown that in the absence of any systematic uncertainties, neural networks are a maximum likelihood estimator (MLE), as shown in appendix A. In reality though, all physics measurements are subject to systematic uncertainties. Systematic uncertainties usually manifest themselves in form of a shift Δ_i of the input variables x_i. Training a neural network on a dataset that is unaware of any systematic variation can lead to an overestimation of the predictive power of certain input variables that may suffer from large systematic uncertainties. This can have a negative impact on the statistical inference that will later be done on the NN output $f(x)$ if systematic uncertainties are applied. Furthermore, certain input variables might be poorly modeled by the simulation. Systematic uncertainties for a given input variable x_i can also be underestimated or overestimated causing a too optimistic or too conservative evaluation of the data. Lastly, uncertainties that are dependent on other parameters $x_j, j \neq i$ can be unknown or not fully understood. An example for this is the correlation between the uncertainties of two input variables. All of this implies that a method of training is desired that is robust against systematic variations of input variables. The goal of the training of the neural network should be to not only give correct predictions for a given input, but also propagate systematic uncertainties from the input space $\mathbf{x} = \{x_i\}$ to the NN output $f(x)$ in order to achieve consistent and robust results in a high-energy physics analysis. The first paragraph gives a brief introduction of potential ways to implement systematic uncertainties in neural networks while the second section explains an already known approach of using an adversarial network to propagate systematic uncertainties through a NN. The third section will then introduce a novel approach on this topic by adding an additional term to the loss function in order to introduce systematic variations into the NN function.

4.1. Implementation of systematic uncertainties in the NN

In general there are two ways of applying systematic uncertainties to the input space $\mathbf{x}$: Firstly, for any given systematic uncertainty $\mathbf{\Delta}$, one could simply add each set of variables $\mathbf{x} + \mathbf{\Delta} = \{x_i + \Delta_i\}$ to the training set. While this is the easiest and clearest approach, its feasibility is dependent on computation capacities and storage due to a large number of events and systematic uncertainties. High energy physics usually uses event numbers beyond 10^6 while also having a large number of systematic uncertainties as seen in section 2.3.3. A training dataset containing all shifted events would most likely use several terabyte of disc space for a single dataset and year. The time for training and evaluation of the

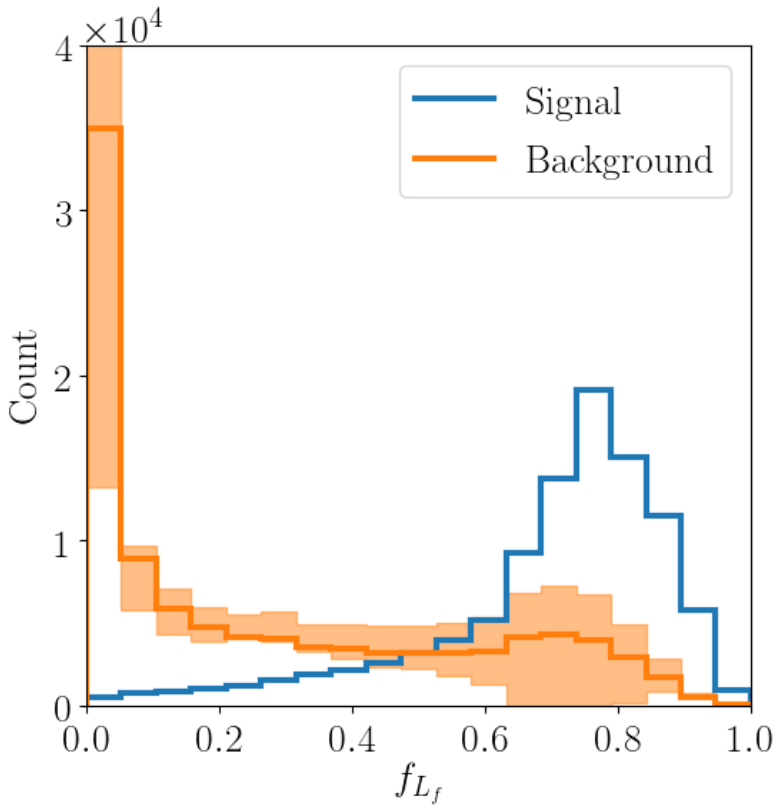
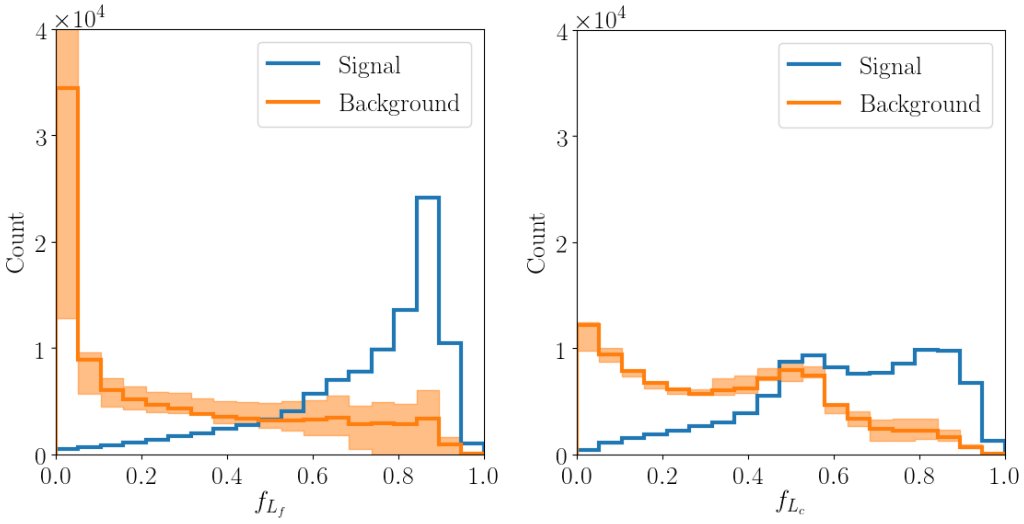


Figure 4.1.: Output of a NN trained with a dataset containing nominal data as well as data with systematic uncertainties. The dataset is visualized in figure 4.4. The bands around the background indicate the variation in output of the NN caused by the systematic variation of the input variables.

NN would also increase with each systematic uncertainty added to the training dataset, greatly increasing the computational effort to generate one NN for classification. In fact, many systematic uncertainties in the $H \to \tau\tau$ analysis are kept as statistical weights that are applied on histogram level only after the NN output score has already been calculated using the nominal value of the data. This way only the weights of a systematic uncertainty have to be stored alongside the event. Application and storage is much more efficient this way. This approach of applying weights to each event for systematic uncertainties will be called re-weighting in the following.

In machine learning the task of implementing systematic uncertainties falls into the broader theme of domain adaption techniques [42, 43]. The goal of domain adaption techniques is usually to find a representation of the data that is independent of the domain. A simple approach of domain adaption sometimes used in high energy physics for domain adaption is to train the classifier simply on data sets containing nominal as well as data, that were shifted according to the systematic uncertainties. As previously mentioned, this leads to a higher storage space needed. While this classifier certainly should perform better on test data containing systematic uncertainties, it can not be guaranteed that the resulting classifier is really robust against systematic shifts. An example of a classifier trained on nominal data as well as shifted data can be seen in figure 4.1. Furthermore, this is highly dependent on the assumption that the generated datasets containing the systematic variation are representative of the data on which the actual classification is later performed on. In physics the datasets are often produced by MC simulation and several plausible values for the systematic uncertainty can be found which means that there can be a disagreement between the shifted simulation given to the NN and the systematic



(a) f_{L_f} (setup 1) **(b)** f_{L_c} with $\lambda = 50$ (setup 2)

Figure 4.2.: The left graph shows the NN output for the NN trained without an adversarial network to decorrelate the systematic uncertainties (setup 1). The right graph shows the NN output for the NN trained with an adversarial network (setup 2). The uncertainty band caused by the systematic variations of the input data are much smaller than for the NN trained without an adversary. On the other hand, the predictive power of the NN has decreased due to the decorrelation against one input variable.

uncertainty actually contained in the data that will later be classified by the NN. With this in mind, an approach to completely or partially remove the information of a variable that has systematic variations would be favored. This removal of information will be called decorrelation in the following.

4.2. Decorrelation through adversarial neural networks

One such approach is trying to decorrelate certain input variables that are known to be affected by a systematic uncertainty using a secondary NN called "adversary" as proposed by [44]. Adversary NNs are – as the name implies – usually made to be in direct conflict with the main classifier network. This conflict can be used to make a classifier more robust by obliterating the information of certain variables with systematic uncertainties. Adversary NNs were first popularized by [45].

The general architecture of this approach can be seen in figure 4.3. The dataset for this task is defined with two input variables x_1 and x_2 and two classes called signal and background. A visualization of the dataset is given in figure 4.4. To introduce a systematic variation, x_2 of the background data is varied by ± 1. This way, three background datasets and one signal dataset are produced. This dataset $\mathbf{X}$ has labels $\mathbf{Y} \in [0, 1]$ according to whether an event belongs to background (0) or signal (1). Additionally, a variable $\mathbf{Z} \in [0, 1]$ is introduced to distinguish between input events with systematic uncertainties (1) and nominal data (0). The result of training the classifier without an adversary (setup 1) and testing the robustness against systematic uncertainties can be seen in figure 4.2 (left). The large bands imply a large difference between the classification score of the nominal data and the classification score of data with systematic uncertainty. It is clearly visible that data that received a down shift tend to be miss-classified as signal due to its more signal-like nature as expected giving the data used for the analysis. The highest discrepancy between nominal, up- and down variation can be seen in the first bin to the left. After the classifier finished training, its output $f_{L_f}(x)$ is given as inputs to the second NN. The adversarial

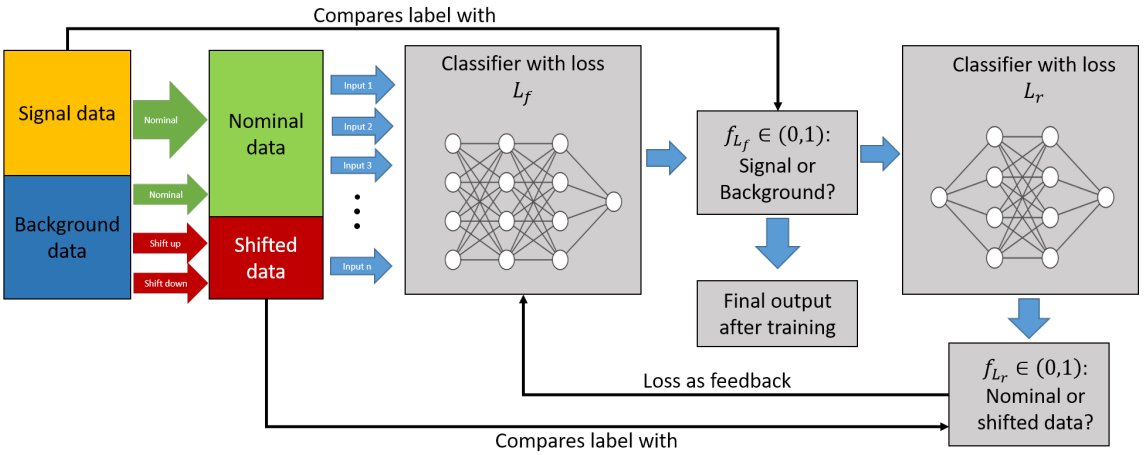


Figure 4.3.: The input variables X and Y are put into a classification NN. The output value of this NN is then taken as an input for the adversarial NN. The adversarial NN determines whether the output of the classification NN was from an event with a systematic variation or from a nominal event. The loss functions of the classification NN and the adversarial NN are then combined as seen in equation 4.1. The higher the loss L_r of the adversary NN is, the less it can successfully determine whether an event had a systematic variation or not and the lower the combined loss function L_c will become. As only the weights of the classification NN can be adjusted in this step, those weights are adjusted in such a way to make the output indistinguishable for the adversary NN, thus making it more robust against systematic variations.

Table 4.1.: Summary of the hyper-parameters of the NN architecture used for classification for all tasks described in chapter 4.

Number of hidden layers	2
Number of nodes per hidden layer	200
Activation functions of the hidden layers	Rectified linear unit (ReLU)
Optimizer algorithm	ADAM (learning rate $= 10^{-3}$) [46]
Validation split	50%
Weight initialization	Glorot (uniform) [47]

NN then tries to distinguish whether a given event has been shifted or not using the additional information of **Z**. In this phase the weights of the classifier are frozen and only the adversary is able to train. Afterwards, the weights of the adversary network are frozen and only the classifier can be trained again. After this initial training of both NNs, the loss functions of both classifier and adversary are combined to a single loss function

$$L_c = L_f - \lambda \cdot L_r. \tag{4.1}$$

Subtracting the loss function of the adversary from the classifier ensures the proper behavior: The larger the loss of the adversary, the better the combined loss function will be. A large adversary loss means that the adversary is unable to distinguish between events with or without systematic uncertainties. The frozen weights of the adversary guarantee that the classifier has to change to reduce the accuracy of the adversary, turning the classifier more robust. For this example, the architecture of the classifier can be seen in table 4.1. The architecture of the adversarial NN uses 64 hidden nodes and softmax activation in the output layer, but is the same otherwise. Pre-training of the classification NN was done for 20 epochs and a batch size of 1000. The adversarial NN was also pre-trained for 20 epochs with a batch size of 128 [44]. The combined training consisted of 1000 gradient steps with

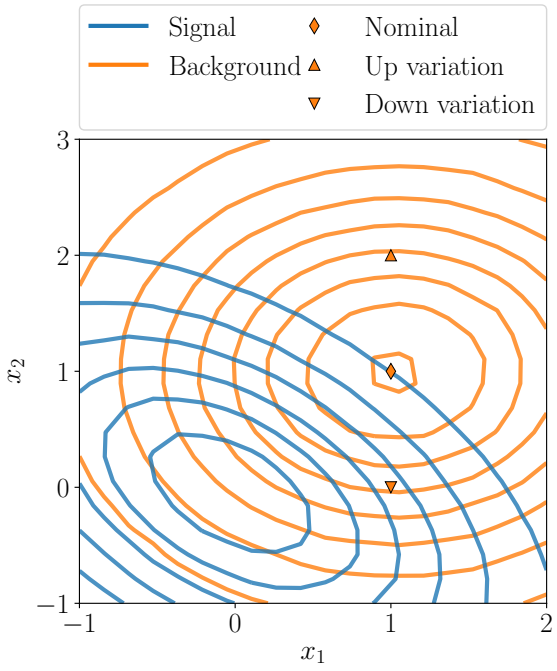


Figure 4.4.: Signal and background data are obtained from two-dimensional Gaussian distributions. The signal data is centered around $(0\ 0)$ with a coveriance matrix of $\begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$. The nominal background distribution is centered around $(1\ 1)$ with a covariance matrix of $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Additionally, the x_2 variable of the background distribution is shifted with ± 1 to simulate a systematic uncertainty

a batch size of 128. Before each gradient step of the combined loss function, the adversarial NN was trained for 1 epoch to ensure that the classification by the adversarial NN is not overpowered too quickly by the change in the classification NN. The single gradient step of the combined lossed combined with the training of the adversarial NN for one epoch will be called a combined epoch in this context.
The evolution of the validation loss functions L_r, L_f and L_c can be seen in figure 4.5. A histogram of the output of the classifier after training on the combined loss function L_c (setup 2) can be seen in figure 4.2 (right). It can be seen that there is almost no distinction by the classification NN between up and down variation as indicated by the much smaller uncertainties bands. As the used data set is only two-dimensional, the boundaries for the output scores of the classification NN can be visualized as seen in figure 4.6. Such a visualization will be called decision surface in the following. Comparing setup 1 and setup 2, one can see that the decision surface of the latter is tilted and x_2, the variable which has the systematic uncertainty, is no longer taken into account for the prediction. The NN effectively obliterates any information of this input variable. This can also be seen by monitoring the evolution of the Taylor coefficients [1] of the NN output function $f_{L_c}(x)$. Shown in figure 4.7 are the Taylor coefficients as a function of the combined epochs during the training of setup 2. It can be seen that the Taylor coefficient t_{x1} and t_{x2}, corresponding to the input variables x_1 and x_2, start at approximately the same value before the combined training is executed. During the training, the value of t_{x2} constantly decreases until reaching approximately 0.05 while t_{x1} actually increases to a value of approximately 0.23. From the given dataset, this is the behavior expected for the Taylor coefficients: The importance of x_2 slowly decreases as the information provided by the variable is successively ignored while more emphasis is put on x_1 by the classification NN. Additionally, the coefficient t_{x1x1}, which is the self-correlation of the x_1 variable, sharply rises in importance during the training before decreasing again to approximately the same value as before. This behavior

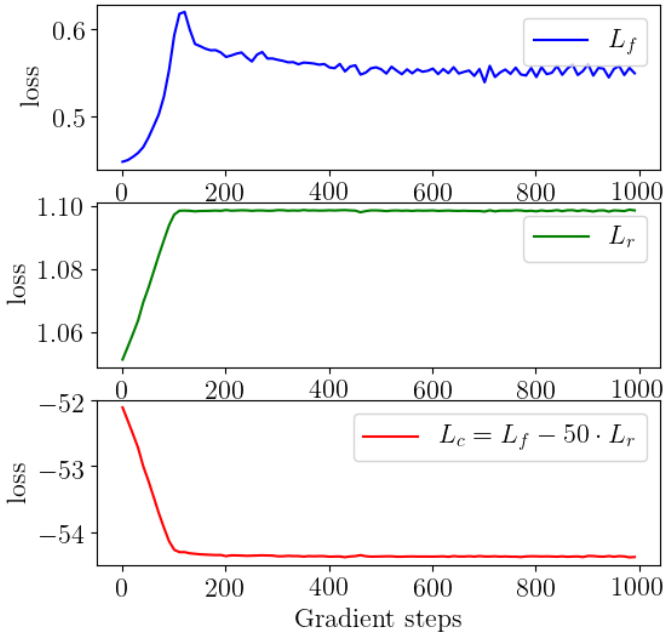
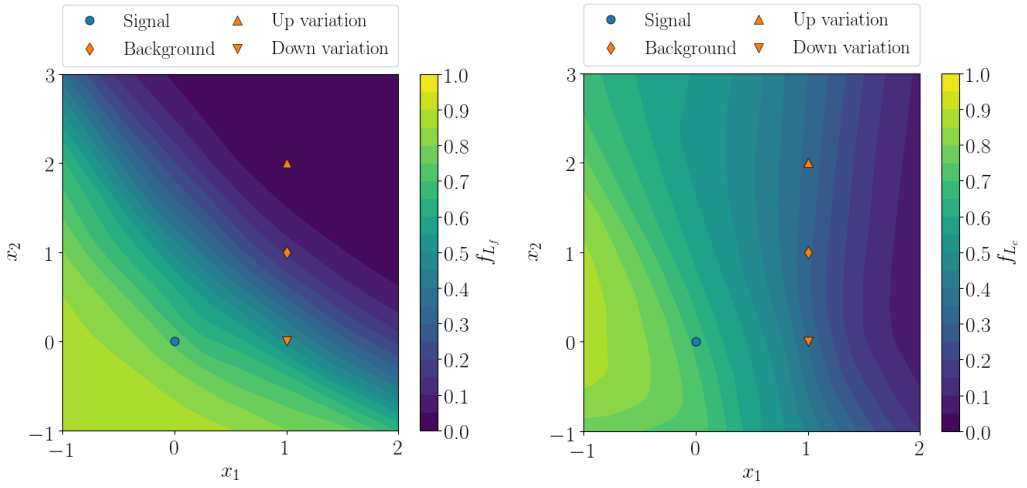


Figure 4.5.: The blue graph shows the loss of the classifier. The loss rises as the classification power of the classifier is reduced in order to decorrelate against the systematic uncertainties. The loss goes down later on as the classifier focuses more on the predictive power of x_1. The green graph shows the loss of the adversary which rises according to the model. The red graph is the combined loss.

is reflected in the loss function of L_f seen in figure 4.5 where the loss term of L_f peaks at approximately the same time t_{x1x1} peaks. The advantage of this approach is its flexibility. In general the adversarial NN is not restricted to be a classifier as shown in this example, but it can also be any kind of neural network or machine learning technique that quantifies its success in form of a loss function (e.g. a Gaussian Mixture Model was used by [44]). It can also decorrelate against continuous shifts and is applicable to examples of high energy physics. Despite of the loss in general predictive power, which is a result of depriving the NN of the information of the unreliable input variable, it could even be shown that in some cases the over all results where better due to the more robust classification of the NN [44]. Nevertheless, this approach comes with its own set of limitations: Firstly, having the optimal NN function after training is not a certainty since reaching the global minimum of the loss function is never guaranteed due to the nature of the back propagation algorithm. Thus, using one NN already results in a fine-tuning process of the corresponding hyper-parameters such as learning rate and regularization terms in order to achieve the best possible output for a given input space. In this approach, a second NN with an additional set of hyper-parameters is added on top of the first NN. Combining both with a hyper-parameter λ, which requires further fine-tuning, results in a training process which can be complicated to fine-tune due to the large amount of hyper-parameters present in the overall architecture. Both loss functions must be kept in a balance, otherwise one NN will simply dominate the overall result of L_c. Especially for the number of data and systematic uncertainties used in high energy physics, this challenge would not be easy to overcome. Secondly, the nature of the combined network does not allow the systematic uncertainties to be passed in the form of weights to the neural network. The input variables must be given to the architecture in form of separate datasets, which again could lead to storage and computation problems. Over all, while showing good results on the examples, the method of using adversarial networks might not be universally applicable to high energy physics.



(a) f_{L_f} **(b)** f_{L_c} with $\lambda = 50$

Figure 4.6.: The left graph shows the decision surface for a classifier trained without an adversary. The boundary is clearly tilted to take both x_1 and x_2 for the decision into account. The down-shifted background clearly reaches into the signal-like region. In the right-hand graph the decision surface is tilted. Now x_2 is mostly ignored in the NN decision and the output is almost exclusively based on the value of x_1.

4.3. Decorrelation with the addition of a penalty term

Given that an additional NN introduces a fine-tuning problem, a solution would be to incorporate the systematic uncertainties directly into the loss function. If decorrelation can be considered a technique to make the NN aware of certain input variables being subject to systematic variations, then a natural formulation of the loss function in form of a penalty term comes to mind. Penalizing the loss function is an established way of achieving a regularization of the output. In the same way it should be possible to penalize an otherwise optimal loss function in order to decorrelate against certain input variables that are considered. An optimal loss function in the absence of systematic variations (meaning only statistical uncertainties are considered) for classification task is, for example, the CE loss function as an NN trained on CE is the MLE for the given problem (see appendix A). In general, the goal should be that $f(x)$ is insensitive to shifts of the input variables. Measuring a difference of outputs is already a well-established loss function called mean squared error. In addition, instead of only using shifted input variables, it is possible to use re-weighting to apply systematic uncertainties with this new penalty term. As the weights are applied on a histogram level, $f(x)$ is transformed into a histogram with counts per bin $\mathcal{N}_k(f(x))$ where k is the corresponding bin. After producing the histogram, the weights can be applied to produce the shifted counts per bin $\mathcal{N}_k(f(\mathbf{x}+\mathbf{\Delta}))$. Taking inspiration from the mathematical formulation of the mean squared error, the penalty term can then be formulated as

$$\Lambda(\mathbf{x}, \mathbf{\Delta}) = \frac{1}{n_k} \sum_k \left(\frac{\mathcal{N}_k(f(\mathbf{x})) - \mathcal{N}_k(f(\mathbf{x}+\mathbf{\Delta}))}{\mathcal{N}_k(f(\mathbf{x}))} \right)^2, \tag{4.2}$$

where n_k is the total number of bins. The total loss function can be written as

$$L_\Lambda = L' + \lambda \Lambda(\mathbf{x}, \mathbf{\Delta}) \tag{4.3}$$

where λ and the number of bins n_k are hyper-parameters that need to be fine-tuned. L' can be any loss function for classification. In this example, L' refers to the CE loss function. If the count per bin for shifted values has a large difference to the nominal values, Λ will

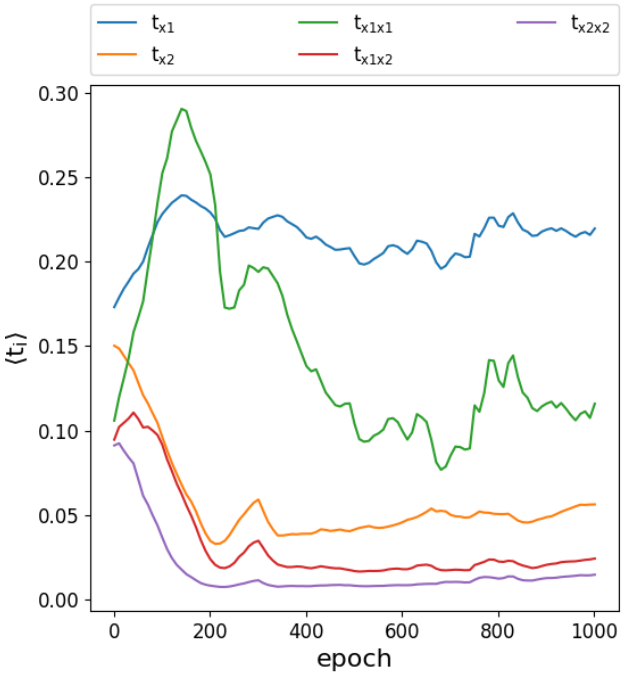


Figure 4.7.: Shown are the first and second order Taylor coefficients of the NN output function f_{L_c} for $\lambda = 50$. As expected, the Taylor coefficient for x_2 decreases during the training while the importance of x_1 slightly increases.

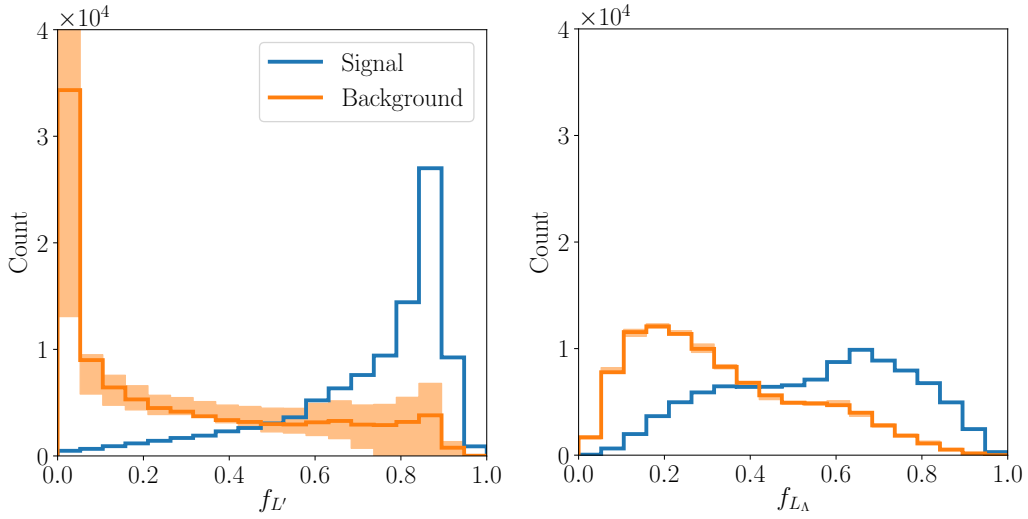
be large and L_Λ will receive an increased value. On the other hand, bin counts close to each other will result in a low penalty term. In consequence, a minimization of the loss function should lead to the desired effect of decorrelation of input variables with systematic uncertainties. The classifier should become more robust in the presence of systematic uncertainties. As the penalty term deprives the NN of additional information usually provided by the input variable we decorrelate against, $f_{L_\Lambda}(x)$ is expected to have a slightly diminished classification power compared to $f_{L'}(x)$.

While we did get additional hyper-parameters with λ and n_k that need to be fine-tuned, the total number of hyper-parameters compared to equation 4.1 is significantly reduced since an additional NN is not required. The penalty term Λ can in principle be extended to include multiple systematic uncertainties. By simply adding penalty terms, each with their own λ_m where m is the number of uncorrelated systematic uncertainties, one can decorrelate against many systematic uncertainties at the same time.

Although this formulation is very intuitive from a theoretical point of view, the implementation poses some technical challenges. Producing a histogram from $f(x)$ can be mathematically formulated as

$$\mathcal{M}_k = \theta(\mathbf{f}(\mathbf{x}) - a) \cdot \theta(\mathbf{f}(\mathbf{x}) + b)\,, \tag{4.4}$$

where a_k and b_k are the bin edges of bin k and θ is the Heaviside theta function. As all NNs learn via backpropagation, the complete function from the final loss function to the weights of the NN must be differentiable. The derivative of the Heaviside theta function is zero and undefined at the edges though and as such can not be directly used to bin the NN output. Instead, a filter function is used which approximates the value of each event corresponding to a bin. A Gaussian function $\mathcal{G}_k(x)$, which is normalized to $\max(\mathcal{G}_k(x)) = 1$, has been chosen as such a filter function. Taking the mean of the Gaussian as the center and the standard deviation as the half-width of bin k, an event with a value $f(x)$ that is exactly in the center of bin k will get a value of 1, while an event further away will receive



(a) $f_{L'}$ **(b)** f_{L_Λ} with $\lambda = 20$

Figure 4.8.: The left graph shows the NN output for a NN trained on cross-entropy and data without systematic uncertainties. The right graph shows the NN output for a NN trained with a penalty term. The bands around the background distribution indicate the effect of the variations on $f(x)$. The bands are clearly reduced for a NN trained with a penalty term.

a value close to 0. Approximating each bin with such a Gaussian filter gives over all a reasonable approximation of the total count per bin. The approximation can be written as

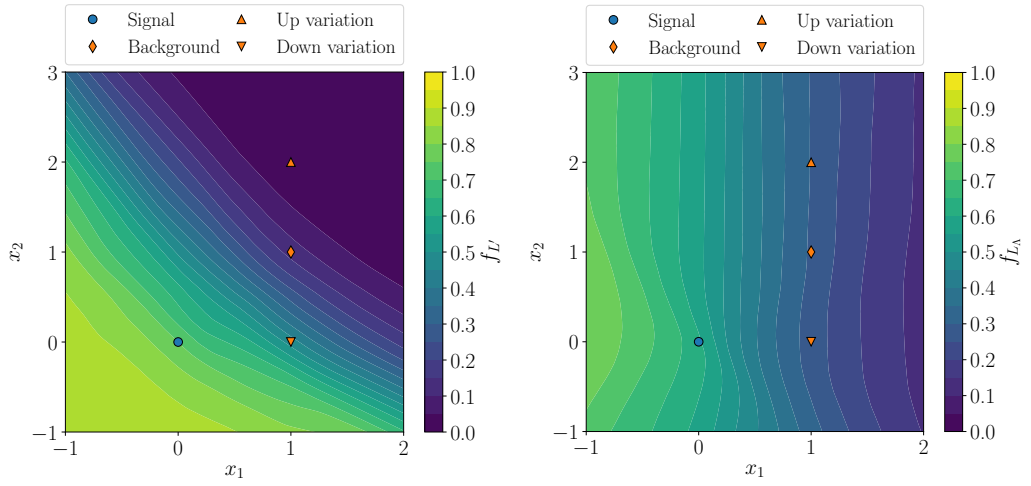
$$\mathcal{N}_k(f(x)) = \sum_b \mathcal{G}_k(f(x))\,, \tag{4.5}$$

where b are the samples in a training batch. Due to boundary effects on the edges of the defined histogram range, the total number of counts per batch $\mathcal{N}_{\text{tot}}(f(x))$ can slightly vary between batches.

4.3.1. Decorrelation of a simple pseudo-experiment with one uncertainty

Testing this loss function on the same pseudo dataset that is used for the adversarial NN setup given in figure 4.4, this approach can be illustrated and compared to the decorrelation effect of the adversarial setup described in section 4.2. Like in section 4.2, the NN used for classification is a simple feed-forward NN. The architecture can again be seen in table 4.1. As the goal is to create a histogram with the output of a training batch, the batch size was chosen sufficiently large with 10^3. For L' the normal binary cross entropy was used. The two new hyper-parameters were chosen to be $\lambda = 20$ with $n_k = 10$ equidistant bins in the range $[0, 1]$. The optimal values for those parameters has to be found by manual optimization. The chosen values demonstrate the decorrelation against x_2 with Λ while still maintaining a reasonable classification with L'. The training was stopped if the loss value on the validation dataset did not improve within 10 epochs and the NN with the best value was chosen for testing.

Using 5×10^4 events for training and 10^5 events for testing, the shape of $f_{L_\Lambda}(x)$ can be seen in figure 4.8. This shapes can now be directly compared to figure 4.2 of the adversarial approach. The expectation was that the penalty term achieves the same reduction in up- and down-variation of the background as the adversarial approach. The figures seem to indicate that $f_{L_\Lambda}(x)$ has even lower variations of the background than $f_{L_c}(x)$. In both cases the separation of background and signal events is less pronounced as for $f_{L'}$. To see



(a) $f_{L'}$ **(b)** f_{L_Λ} with $\lambda = 20$

Figure 4.9.: The decision surfaces are strikingly similar to figure 4.6.

whether Λ reduces the impact of x_2 on $f_{L_\Lambda}(x)$ as well, Taylor coefficients were plotted for different values of λ after convergence as seen in figure 4.10. As expected, the values for t_{x1} and t_{x2} are approximately the same for $\lambda = 0$. The value of t_{x2} decreases with a higher value of λ while the importance of x_1 increases first up until $\lambda = 3$ and then slowly decreases as well. t_{x1x1} shows the same behavior as for $f_{L_c}(x)$. For lower values of λ it rapidly increases peaking at $\lambda = 3$ and then rapidly decreases again. At the chosen value of $\lambda = 20$, the Taylor coefficient of t_{x1} is more than double the value of t_{x2} and t_{x1x1}. That the Taylor coefficients show a similar behavior as functions of values of λ for L_Λ as they did as functions of combined epochs and a fixed λ for L_c indicates that the value of the Taylor coefficients with an adversary setup is dependent on the number of combined epochs trained. A higher number of combined epochs in the adversarial setup correlates to a higher value for λ in the penalty term setup. The NN for the adversary setup slowly converges towards $\lambda = 50$. This is also reflected in the fact that the Taylor coefficients are converged after 500 epochs and stay approximately the same. The Taylor coefficients for $f_{L_\Lambda}(x)$ fluctuate much less than for $f_{L_c}(x)$ because the Taylor coefficients are only calculated after full convergence of the NN when a particular value of λ is reached.
The similarity of both approaches can further be seen when comparing the decision surfaces in figure 4.9 and figure 4.6. In both approaches the decision plane is tilted in such a way that x_2 is mostly unimportant for $f(x)$ especially between the critical range of $[0, 1]$ for x_1. The decision surface of $f_{L_c}(x)$ shows some additional nuances towards more signal-like region for values of $x_1 < 0$ and $x_2 < 2$ where the decision surface is less symmetrical to x_1. This indicates that not all information of x_2 is disregarded by the approach in this region. $f_{L_\Lambda}(x)$ does also show a slight asymmetrical decision surface for lower values of x_1 and x_2 but to a much lesser extent than $f_{L_c}(x)$. Furthermore, $f_{L_c}(x)$ has lower values for the more background-like region of $x_1 > 1$ then $f_{L_\Lambda}(x)$ indicating a greater confidence in classifying those values as background. This is reflected in the shapes of both outputs as the first bins (from the left) are more populated for $f_{L_c}(x)$ than for $f_{L_\Lambda}(x)$.

4.3.2. Decorrelation of a simple pseudo-experiment with two uncertainties

To test the behavior of the penalty term for multiple uncertainties, a second dataset was created. It was established in the first example that decorrelation against an input variable completely or partially removes the information provided by this variable. As a consequence, simply applying a systematic variation to x_1 of the previous dataset and decorrelating

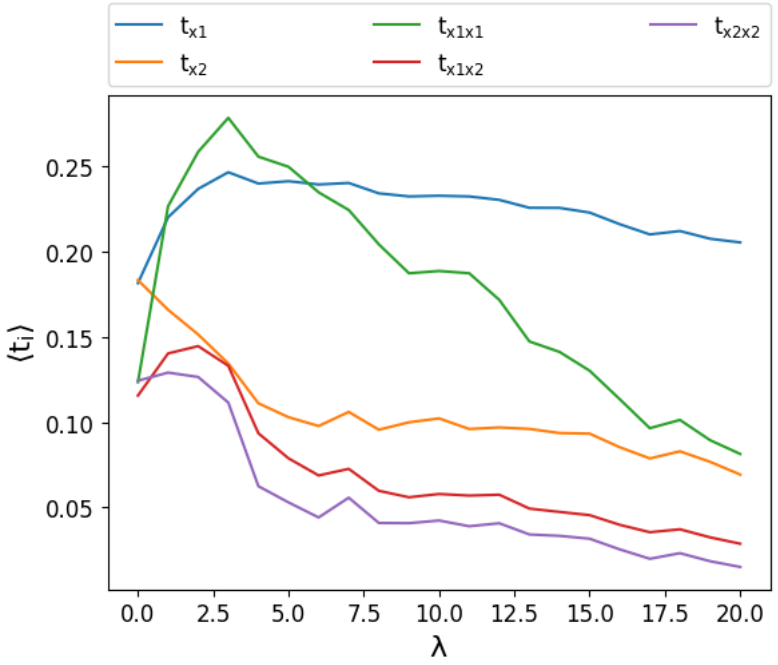


Figure 4.10.: Shown are the first and second order Taylor coefficients of the NN output function f_{L_Λ} for different values of λ. As expected, the Taylor coefficient for x_2 decreases for higher values of λ while the importance of x_1 slightly increases.

against this systematic variation as well would deprive the NN of all information that would be given by the two input variables. To mitigate this, a third input variable x_3 was introduced in the dataset that does not have a systematic variation while x_1 and x_2 now are both shifted. The datasets were produced using three-dimensional Gaussian distributions. The parameters for the signal distributions are

$$\mu_S = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.6}$$

$$\sigma_S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

And for the background distributions:

$$\mu_B = \begin{pmatrix} 1 \pm 1 \\ 1 \pm 1 \\ 1 \end{pmatrix} \tag{4.8}$$

$$\sigma_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.9}$$

where μ_i and σ_i are the means and covariance matrices respectively. A two-dimensional representation of the data can be seen in figure 4.11. It should be noted that the datasets for background and signal events, as well as the datasets for each systematic variation are generated independently from each other and are statistically independent.
The architecture of the NN is the same as in section 4.3.1 and section 4.2 to guarantee comparable results. Each uncertainty was treated with a separate penalty term which was added to the CE loss function L'. This way the decorrelation against the input variables

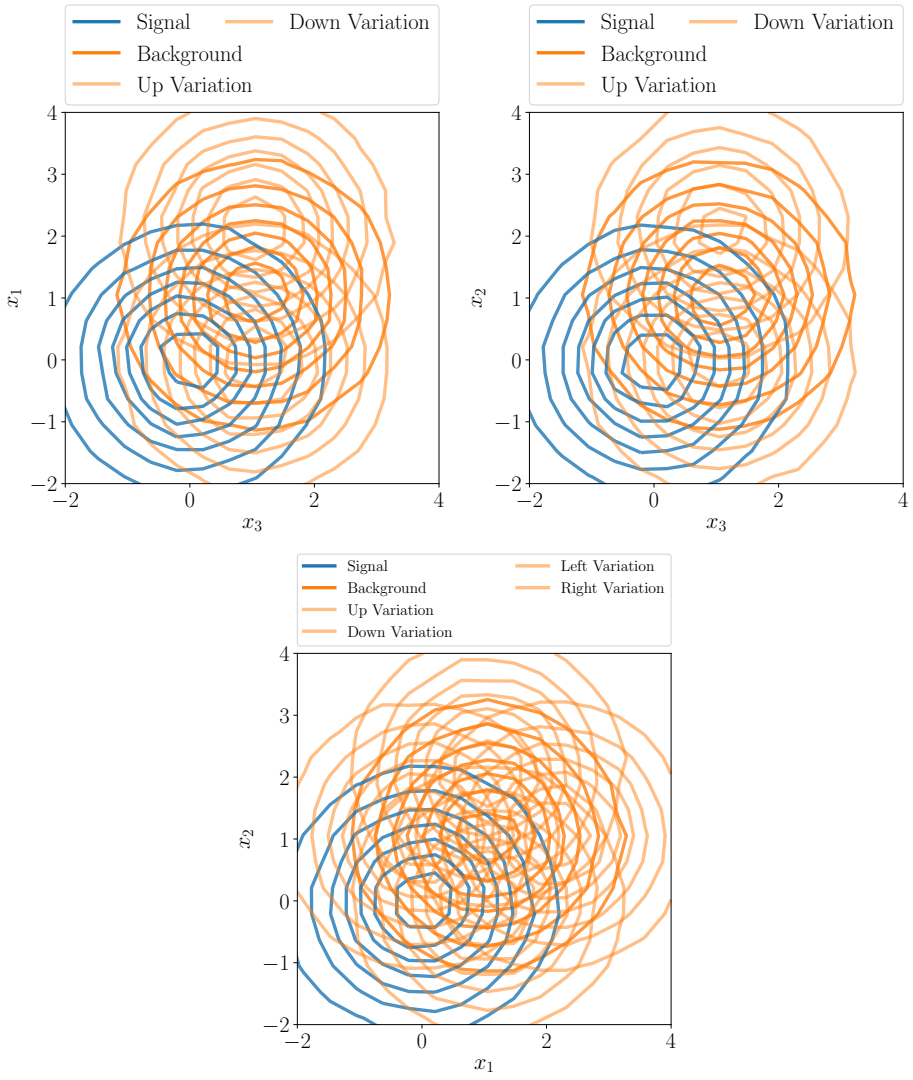
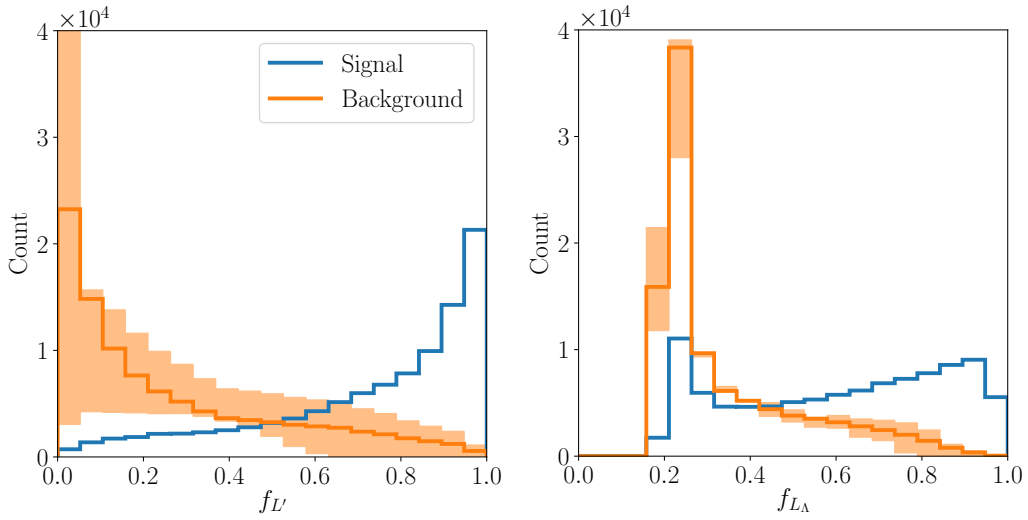


Figure 4.11.: Two-dimensional representation of the used dataset. The datasets are created using three-dimensional Gaussian distributions. Variables x_1 and x_2 have an uncertainty on their mean of ± 1 while the variable x_3 does not have an uncertainty.

was done independently for each systematic variation. The results for $f_{L_\Lambda}(x)$ and $f_{L'}(x)$ can be seen in figure 4.12. It can indeed be seen that the uncertainty bands are reduced for $f_{L_\Lambda}(x)$. In contrast to figure 4.8 though, there is still some systematic variation left, especially in the highly populated bins around 0.2. This can be attributed to the fact that the loss function is balancing classification and the values of two separate penalty terms to achieve the best result, which is a compromise between classification of events and decorrelation of the two systematic variations. A higher value of λ might reduce the uncertainty bands even more at the cost of a further reduced separation power.

From the produced dataset, the expectation would be that the NN now mostly disregards the information provided by x_1 and x_2 to separate the data and focuses mostly on the information of x_3. This assumption can again be verified by looking at the evolution of the Taylor coefficients as a function of different values of λ as seen in figure 4.13. At a value of $\lambda = 0$, the first order Taylor coefficients for all variables are approximately the same. Confirming the assumption made above, the Taylor coefficients of x_1 and x_2 then decrease slowly with an increasing value of λ. The Taylor coefficient for x_3, on the other hand, increases until around $\lambda = 5$ before slowly decreasing as well. The maximum could be an indicator that this value of λ is optimal in terms of a balance between the separation



(a) $f_{L'}$ **(b)** f_{L_Λ} with $\lambda = 20$

Figure 4.12.: Both histograms show the output distribution when both systematic variations are applied at the same time. The uncertainty bands for $f_{L_\Lambda}(x)$ are largely reduced in comparison to $f_{L'}(x)$. On the other hand, more events are closer to a value of 0.5 for $f_{L_\Lambda}(x)$, indicating a less clear classification decision of the NN.

power and the decorrelation against the systematic uncertainties. Again, the value of this Taylor coefficient stays well above the values of the other Taylor coefficients, marking it as the most important value for the classification. The second order Taylor coefficients t_{x3x3} rises as well, reminiscent of the rise of t_{x2x2} in figure 4.10 before peaking at around $\lambda = 3$ and then slowly decreasing.

As the variables are not correlated to each other and the datasets are statistically independent, the decision surfaces can again be plotted by projecting the three-dimensional space onto a two-dimensional plane. The projection is done by taking the average of all output scores along one axis. The decision surfaces for $f_{L_\Lambda}(x)$ can be seen in figure 4.14. Similar to the example with one systematic variations, the $x_1 - x_3$ and $x_2 - x_3$ decision planes (top row) are now tilted and symmetrical to the x_3 axis. Both decision surfaces are again showing that the information of x_1 and x_2 are almost completely obliterated. A more nuanced look into the information that is still provided by x_1 and x_2 is the $x_1 - x_2$ decision plane. As expected, most scores are around 0.5 for $x_3 > 0$ indicating that the NN could not clearly classify those events to either background or signal based on the information provided by x_1 and x_2. In fact the surface shape indicates that, for most values, the sensitivity of the NN for this particular plane is based solely on higher order correlations between variables instead of the first order information. As σ_S and σ_B are both unity matrices, there is no correlation to be expected between variables. The low scores for the second-order Taylor coefficients does indeed confirm this result. Nevertheless, the diagonal from upper left to lower right corner could be considered more background-like and the diagonal from lower left to upper right more signal-like, albeit all the scores are very close to 0.5 as already mentioned. Only in the lower-left corner at values of $x_1 < -1$ and $x_2 < -1$ is the first order information of x_1 and x_2 not completely obliterated and can be used by the NN to classify events as signal-like.

4.3.3. Decorrelation of a high energy physics example

The toy example illustrates the decorrelating power of the penalty term. A more realistic dataset is the Higgs boson machine learning challenge released by the ATLAS collaboration [48]. This dataset consists of $H \to \tau\tau$ signal processes and background processes

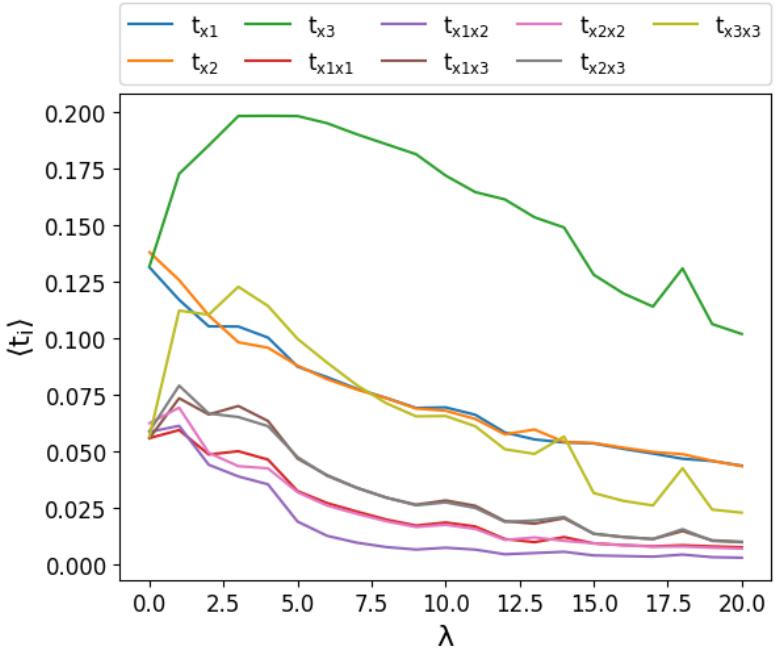


Figure 4.13.: Shown are the first and second order Taylor coefficients of the NN output function f_{L_Λ} for different values of λ for two systematic variations. As expected, the Taylor coefficients for x_1 and x_2 are decreasing with increasing λ while the importance of x_3 is rising with a maximum of at $\lambda = 5$ and clearly staying above the other values.

obscuring the signal. The original challenge is to separate signal from background using any machine learning technique. The data is a simplified synthetic set from simulated collisions of high-energy proton beams as they are used at the CERN LHC. Since the task at hand was to only separate signal from background, systematic uncertainties were originally not part of the dataset. The dataset consists of a training set with 250000 events and a test set of 550000 events. Each event has 30 input variables, the exact physical meanings can be seen in [48]. Additionally each event has a weight associated with it. Those weights are not the same as used for the re-weighting later on. They are associated with the cross section of each process and event and must be applied in addition to the weights that are used to apply the systematic variations in the penalty term. All input variables will be used for this example.

In order to introduce a systematic uncertainty into the set of variables, the transverse momentum of the reconstructed hadronic τ decay p_t^τ is shifted by a small amount. The uncertainty is chosen to be $\pm 3\,\%$ in accordance with the actual measurements [49]. The uncertainty on the transverse momentum comes from the finite resolution of the detectors measuring the energy. As p_t^τ has a requirement of $p_t^\tau > 20\,\text{GeV}$ for all events in the dataset, simply applying a shift of $\pm 3\,\%$ to the data set causes migration effects on the edges of the distribution of the input variables. Events from regions that are higher than 20 GeV can freely migrate downwards while no events can migrate upwards. Therefore, the p_t^τ requirement is raised to $p_t^\tau > 22\,\text{GeV}$. With this requirement, events can migrate upwards as well as downwards. This solution introduces a caveat though: As the shift is based on a percentage and the variation is dominated by migration effects at the lower p_t^τ boundary, the down variation of the uncertainty has a lower total amount of events than the nominal case while the up variation has a higher total amount. In other words: Changing the lower boundary requirement for the p_t^τ introduces an additional normalization uncertainty that needs to be taken into account later on. This normalization uncertainty is amplified for the background due to the steeply falling distribution as seen in figure 4.15 (upper left).

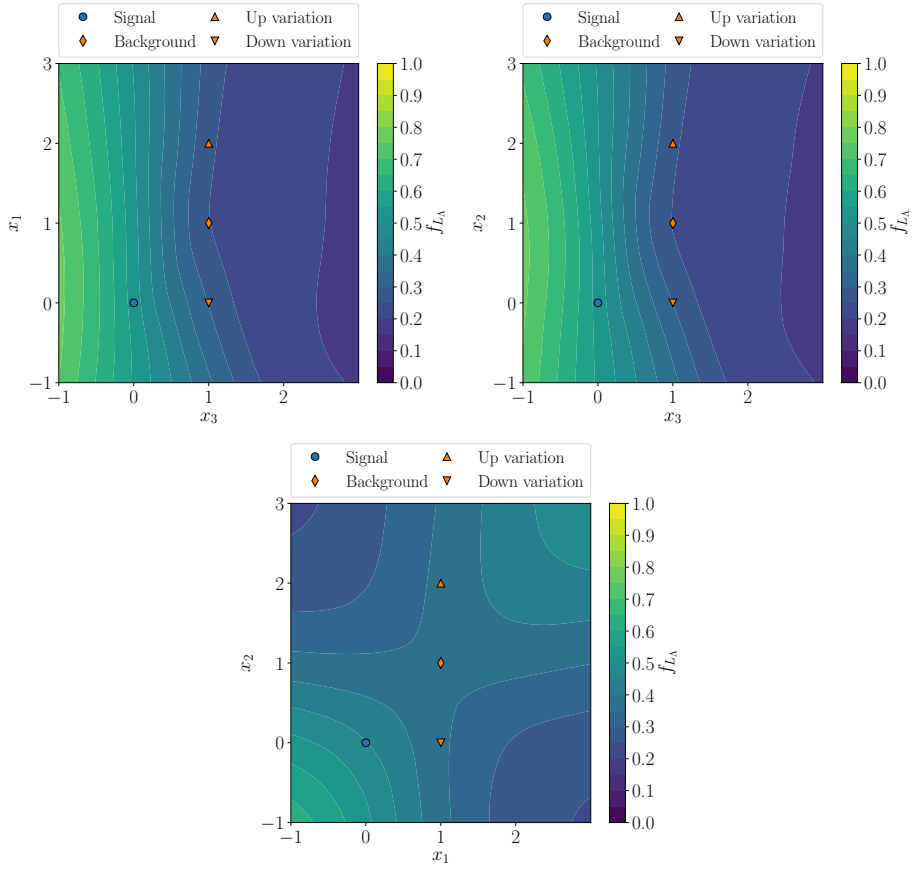


Figure 4.14.: The decision surfaces projected onto a two-dimensional plane for two systematic variations in x_1 and x_2 respectively. Especially the $x_1 - x_2$ plane indicates that most information of those variables are not considered for the decision of the NN.

The effect of the normalization uncertainty can be seen in the ratio to nominal. The ratio is approximately constant for lower p_t^τ values indicating that the uncertainty introduced by shifting p_t^τ is dominated by the normalization uncertainty. The distribution of signal events (upper right) does not show a constant ratio and can therefore be considered to be dominated by the shape uncertainty.

As with the simple pseudo-experiment example, the systematic uncertainty is applied on histogram level via statistical weights instead of resampling the signal and background datasets completely. By construction, re-weighting conserves all correlations across variables in the input space. This means that the uncertainty introduced only to p_t^τ is propagated to other input variables that are correlated to p_t^τ. Thus, all distributions that are correlated to p_t^τ will be subject to a systematic variation as desired. An example for correlated input variables are the missing transverse momentum and the invariant di-τ mass [48]. Both can be seen in figure 4.15 (lower row). The weights for each event can be calculated from the histograms given in figure 4.15 (upper row). It should be noted that the introduction of a fixed uncertainty of $\pm 3\,\%$ is of course a simplified case of a systematic uncertainty since in data from an actual physics measurement, intermediate values are normally also realized and given.

For the training, the normalization uncertainty is not considered. By construction, the penalty term of the loss function cannot reduce the effect of a normalization uncertainty on the NN output. In principle, the normalization uncertainty could be expressed as a weight and an additional penalty term could be constructed that could be used to decorrelate against this uncertainty. However, the weight of the normalization uncertainty would apply

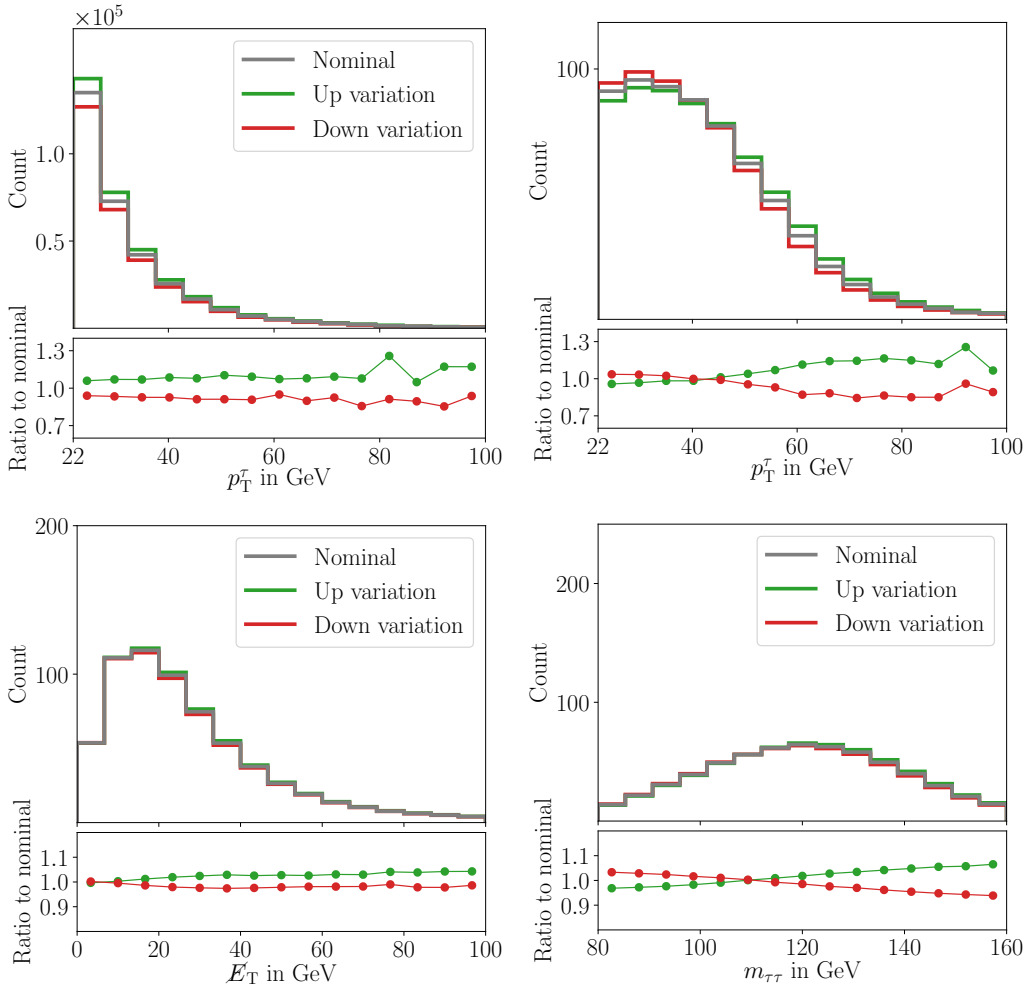
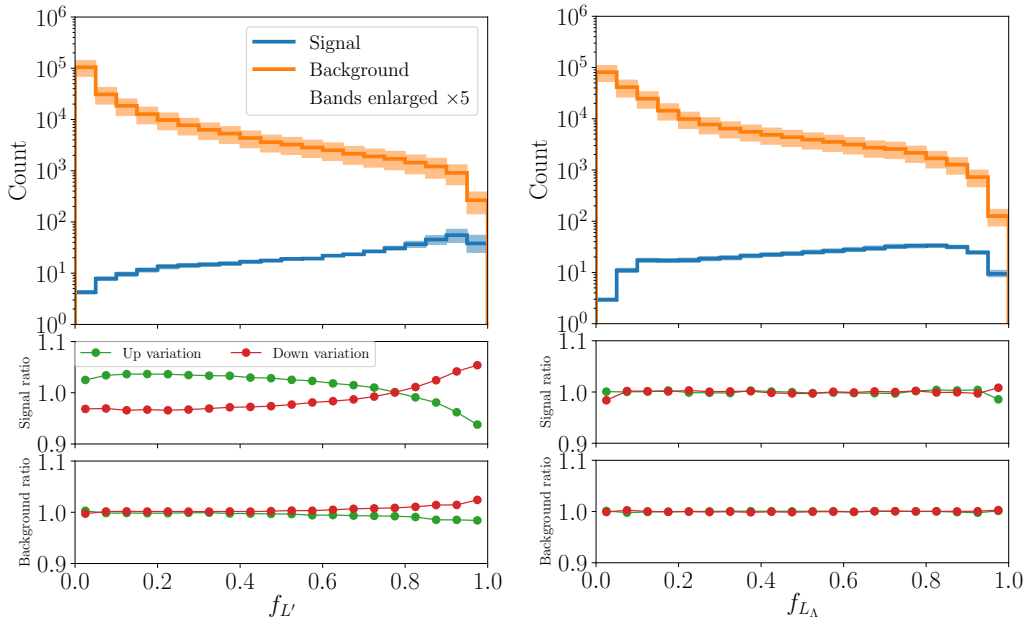


Figure 4.15.: The upper row shows the distribution of the transverse momentum p_t^τ for background (upper left) and signal (upper right). The variations introduced by weights are also shown in this plots. The ratio of the up and down-shifted histograms w.r.t. the nominal histogram is given in the lower panels of the graphs. The lower row shows the impact of the reweighting on other variables that are correlated to p_t^τ and thus are also affected by the systematic shifts of p_t^τ.

to all events within background and signal with the exact same value. It can therefore be considered a global constant. This is a crucial difference to the application of the weights of the systematic uncertainty: Those weights were calculated per bin, thus each bin has a different weight. Applying the same weight to all events would result in a gradient of the penalty term that would shift all NN weights of all events in the exact same direction. As a result, no information would be gained by the NN and the penalty loss would not be reduced, effectively rendering the whole term without a function. Thus, normalization uncertainties cannot be reduced with this NN architecture. Instead, the normalization parameters for signal events s and background events b are calculated from the training dataset and the histograms constructed for each batch are divided by the calculated normalization parameters to effectively obscure any normalization uncertainty from the NN while training. The normalization parameters calculated from the training



(a) $f_{L'}$ **(b)** f_{L_Λ} with $\lambda = 1000$

Figure 4.16.: The left graph shows the NN output for a NN trained on cross entropy and data without systematic uncertainties. The right graph shows the NN output for a NN trained with a penalty term. The bands around the background distribution indicate the systematic uncertainties which are a combination of normalization and shape uncertainty. The bands are enlarged by a factor of 5 for better visibility. The normalization parameters can be seen in equation 4.12. The bands are reduced for a NN trained with a penalty term. The ratio plots beneath the histograms give the ratio per bin between histograms with applied shape uncertainty and nominal histograms. The normalization uncertainty was not considered in the ratio plots. The ratios of f_{L_Λ} are overall closer to a value of 1 as expected.

dataset are:

$$\begin{aligned} N_{\text{s,up}} &= 1.02 \\ N_{\text{s,down}} &= 0.98 \\ N_{\text{b,up}} &= 1.07 \\ N_{\text{b,down}} &= 0.93 \end{aligned} \tag{4.10}$$

For each normalization uncertainty, a penalty term is formulated in the form of

$$\begin{aligned} \Lambda(\mathbf{x}, \boldsymbol{\Delta}) &= \frac{1}{n_k} \sum_k \left(\frac{\mathcal{N}_k(f(\mathbf{x})) - \mathcal{N}_{k,\text{shape}}(f(\mathbf{x}+\boldsymbol{\Delta}))}{\mathcal{N}_k(f(\mathbf{x}))} \right)^2 \\ \mathcal{N}_{k,\text{shape}}(f(\mathbf{x}+\boldsymbol{\Delta})) &= \mathcal{N}_k(f(\mathbf{x}+\boldsymbol{\Delta}))/N_{\text{Norm}}, \end{aligned} \tag{4.11}$$

where N_{Norm} is a the corresponding normalization parameter given in equation 4.10. This way, the training solely focuses on reducing the shape uncertainty introduced by the shift of p_t^τ and there is no residual value of the penalty term caused by the normalization uncertainty.

The architecture of the NN is the same as used in the simple pseudo-experiment described in section 4.3.1. The hyper-parameters for the loss function are $\lambda = 1000$ and $n_k = 20$. λ is chosen to be large as the shape uncertainty is quite low and consequently the values of Λ are easily overshadowed by the value of L'. The bins are chosen to be equidistant in the range $[0, 1]$ and the batch size is 10^3. The training was stopped after 50 epochs if the validation

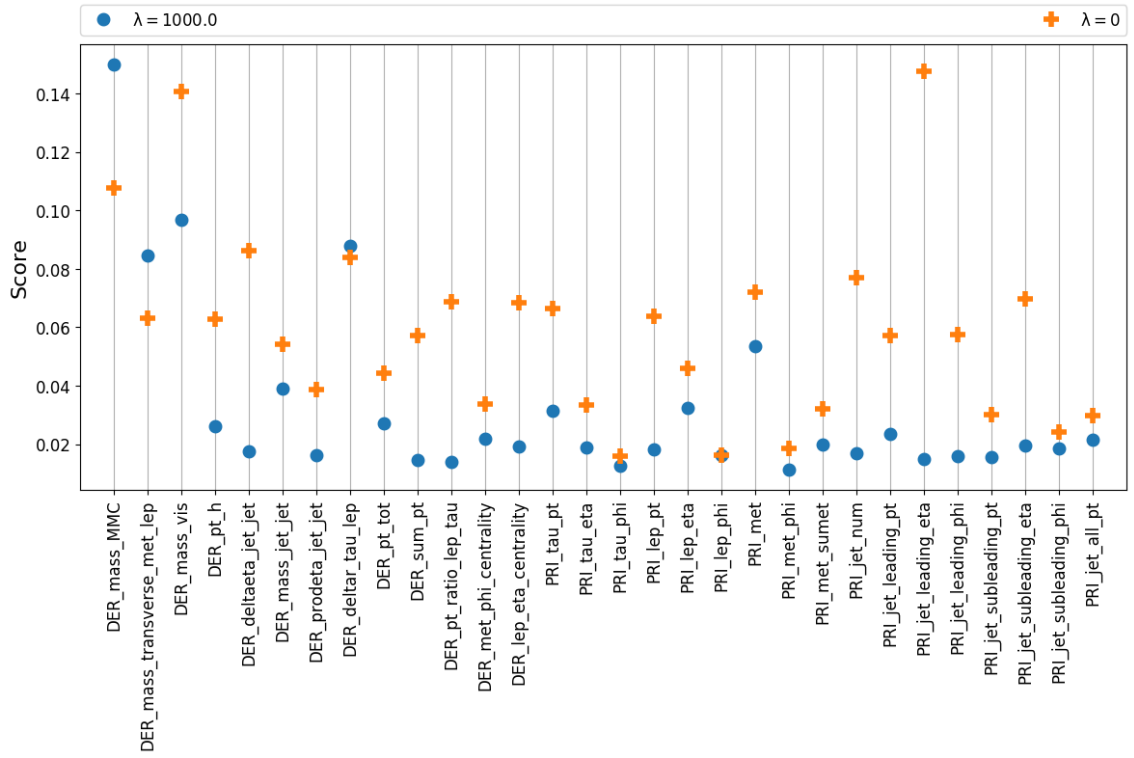


Figure 4.17.: Shown are the first order Taylor coefficients for the high energy physics example for NNs with $\lambda = 1000$ and $\lambda = 0$. The Taylor coefficients were only calculated once after a NN finished training. A different focus on variables can be seen for f_{L_Λ} compared to $f_{L'}$ with most variables having lower Taylor coefficients for f_{L_Λ}.

loss value did not improve in this time frame to avoid overfitting the NN. Afterwards, the network was tested on the test set for independent results. The histograms of the NN output with all systematic shifts are shown in figure 4.16. To make the small uncertainty shift more visible, the shifted histograms are divided by the normalization parameters calculated on the test dataset. The corresponding ratios of the shifted histograms and the nominal histograms are shown in the lower half of the figure. The normalization parameters for the test dataset are:

$$\begin{aligned} N_{\text{s,up}} &= 1.02 \\ N_{\text{s,down}} &= 0.98 \\ N_{\text{b,up}} &= 1.07 \\ N_{\text{b,down}} &= 0.93 \end{aligned} \tag{4.12}$$

As the uncertainty shift of $\pm 3\,\%$ is rather small, the effect of decorrelation is less visible in comparison to the pseudo-experiment histograms in figure 4.8. Nevertheless one can see a reduction of the dependence of $f_{L_\Lambda}(x)$ on the systematic variations of p_T^τ especially in the ratio plots. The normalization uncertainty can be seen in form of a constant value for up and down shift independent of the bin. The effect of this uncertainty is more pronounced in the background category due to a higher number of events at the p_t^τ threshold. The graph also indicates that the normalization uncertainty is in fact more dominant than the shape uncertainty which is reduced by the penalty term for higher values of λ.

To get a better understanding of the variables used by the NN for the classification, Taylor coefficients are again calculated. The Taylor coefficients for $f_{L_\Lambda}(x)$ and $f_{L'}(x)$ for all variable used during training can be seen in figure 4.17. As expected, the invariant mass `DER_mass_vis` of the hadronic τ and the lepton is one of the most important variables

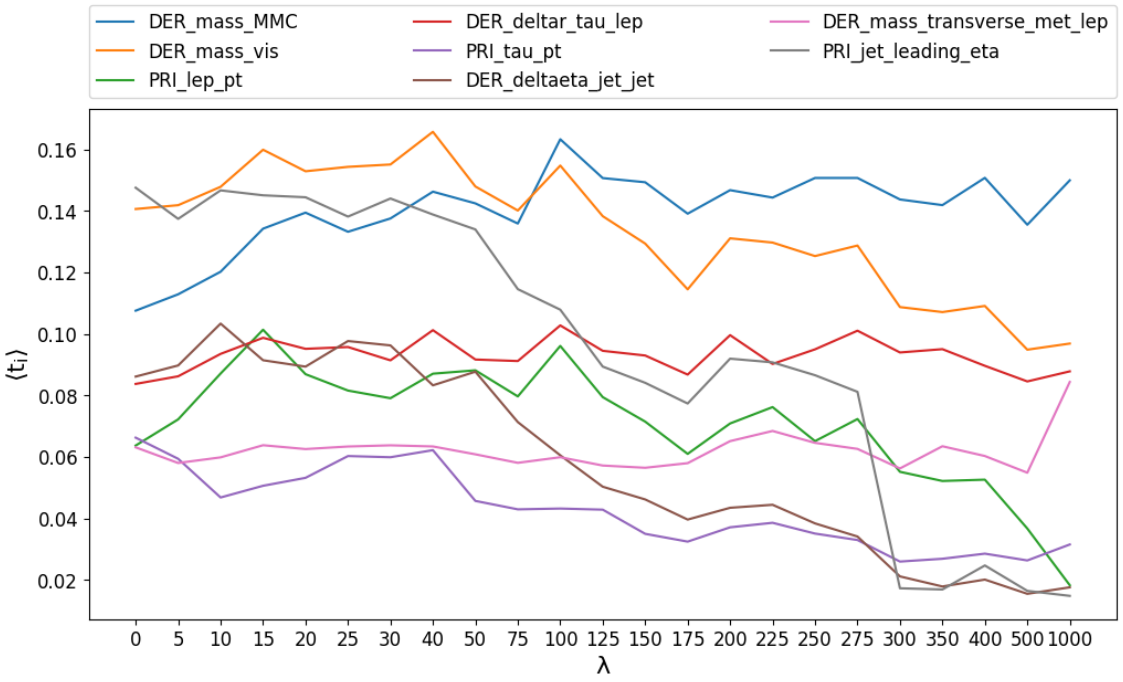


Figure 4.18.: Shown is the evolution of the first order Taylor coefficients for different values of λ. The variables shown are only a selection of all variables. The Taylor coefficients were only calculated once after a NN finished training. The Taylor coefficients decrease as a function of λ for most variables with noteworthy exemptions like `DER_mass_MMC`.

and the importance is only slightly lower for $f_{L_\Lambda}(x)$. The importance of `DER_mass_MMC`, which is the estimated mass m_H of the Higgs boson candidate, was already high for $f_{L'}(x)$ but increases further for $f_{L_\Lambda}(x)$. The variable is affected by the shift of p_t^τ as seen in figure 4.15, but the shape uncertainty mostly affects the signal events while the background events are mostly affected by the normalization uncertainty. A potential reason for the increase in importance is that there are more events in the background distribution than the signal distribution, thus the background distribution has a higher impact on the training (this will be also shown later on). Therefore, variables with background distributions like `DER_mass_MMC` become more important for the classification. The transverse mass `DER_mass_transverse_met_lep` between the missing transverse energy and the lepton also increases slightly for potentially the same reason. The R separation between the hadronic tau and the lepton `DER_deltar_tau_lep` stays approximately the same indicating that the importance of the variable is not influenced by the penalty term. The importance of most other variables decreases, with some dropping a significant amount. The evolution of the Taylor coefficients $\langle t \rangle$ as a function of different values of λ for a selection of variables is shown in figure 4.18. The Taylor coefficients for p_t^τ steadily decreases as a function of λ, though $\langle t \rangle$ for this variable was never high to begin with. On the other hand, the transverse momentum p_t^l of the second lepton slightly rises in importance with increasing λ before slowly decreasing like p_t^τ for $\lambda > 100$ and even dropping below p_t^τ for $\lambda = 1000$. The strongest decline in importance can be seen for the pseudorapidity η_{jet} of the leading jet. This variable is considered to be as important as variables associated with masses for $\lambda = 0$, but starts to slowly decline for $\lambda > 50$ before completely dropping off and converging to 0 for values of $\lambda > 275$. Confirming the previous assessment, the values of `DER_mass_transverse_met_lep` and `DER_deltar_tau_lep` indeed stay approximately the same for all values of λ with `DER_mass_transverse_met_lep` only gaining importance for values of $\lambda > 500$. Overall the importance of variables seems to shift towards mass terms that are less affected by the uncertainty in the background distributions while variables

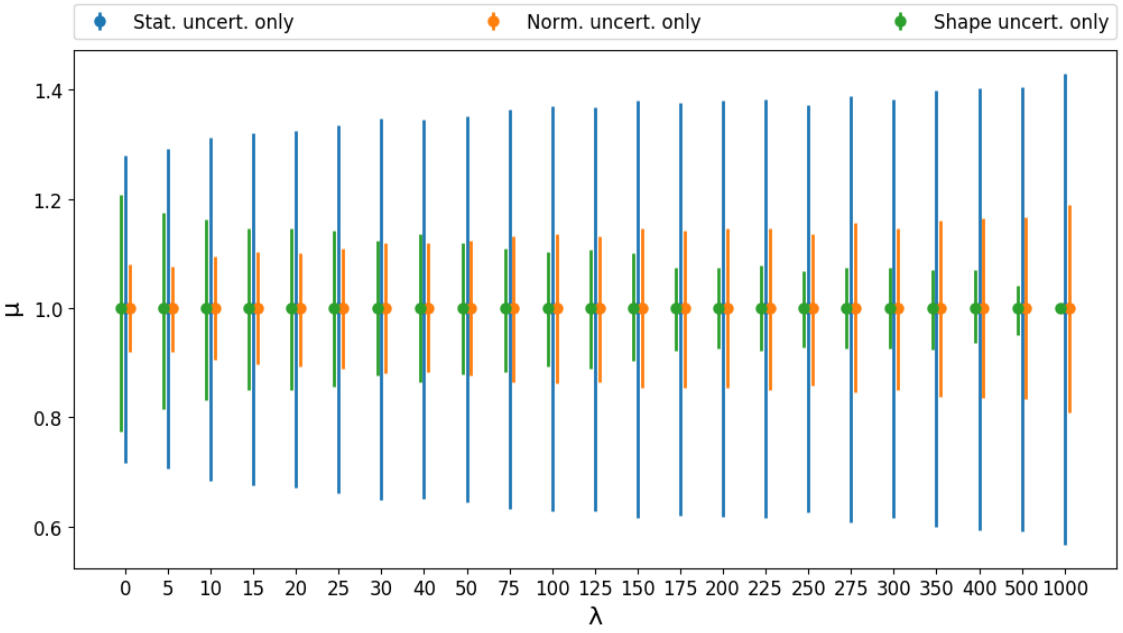
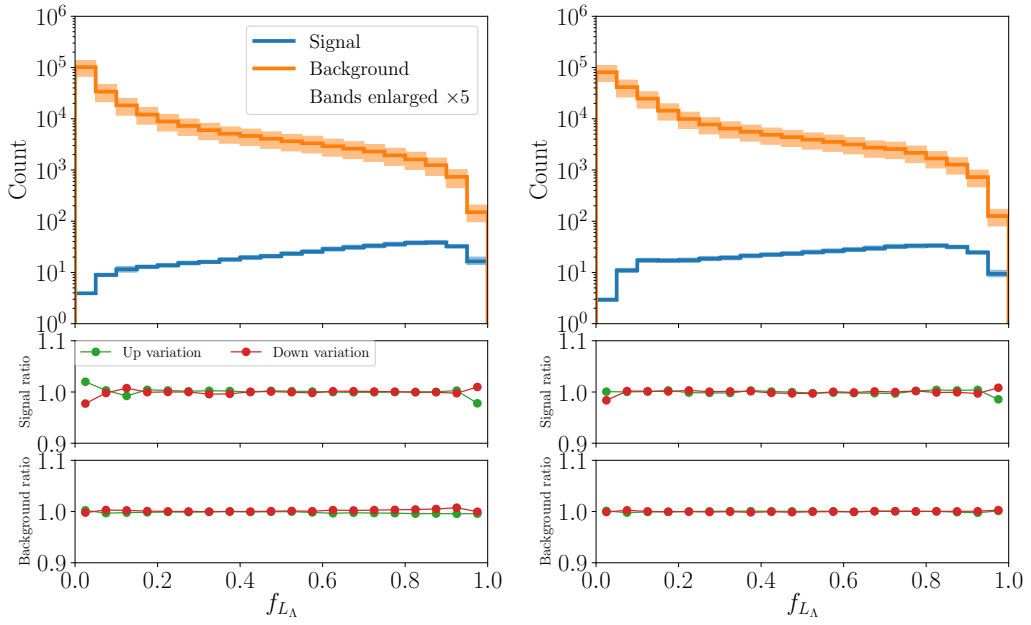


Figure 4.19.: Shown is the calculated signal strength μ and the uncertainties $\Delta\mu$ associated with it. As μ was calculated using an Asimov dataset, the value is always $\mu = 1$ and the important test statistics are the calculated constraints $\Delta\mu$ on the signal strength. The combined uncertainty is splitted into statistical uncertainty, shape uncertainty and normalization uncertainty too see the reduction in the shape uncertainty.

associated with features of jets or transverse momentums generally lose importance. It should be noted that this is not a unique result and the interpretation is therefore not universally applicable.

As a final comparison, the statistical inference process described in section 2.4.2 is applied to the histograms given in figure 4.16. As previously discussed, when constructing a profiled likelihood fit, systematic uncertainties can be accounted for with nuisance parameters $\boldsymbol{\theta} = \{\theta_i\}$. Usually every known systematic uncertainty Δ_i is incorporated by one nuisance parameter θ_i. As this will be a simple hypothesis test where the signal hypothesis will be tested against a null hypothesis, only a single parameter of interest μ representing the signal strength is needed. To calculate the uncertainty $\Delta\mu/\mu$ on μ, an Asimov dataset, where data is replaced by simulation, was used. This fixes the signal strength to $\mu = 1$ and the uncertainty simplifies to $\Delta\mu$. Fitting the profiled likelihood with a single nuisance parameter θ and splitting the uncertainty into a statistical part $\Delta\mu_{\text{stat}}$ and a systematic part $\Delta\mu_{\text{sys}}$, the results for $f_{L'}$ for the 68% CI are $\Delta\mu_{\text{stat}} = \pm 0.28$ and $\Delta\mu_{\text{sys}} = ^{+0.11}_{-0.08}$. The calculated uncertainties for f_{L_Λ}, on the other hand, are $\Delta\mu_{\text{stat}} = \pm 0.43$ and $\Delta\mu_{\text{sys}} = \pm 0.19$. While this might look like it is contradicting the previous findings as we would expect the systematic uncertainty to decrease while the statistical uncertainty increases, it is not completely unexpected if we take the previous discussion about normalization uncertainties into account. As already discussed, the normalization uncertainty introduced by raising the p_t^τ boundary is dominant in the background classes. Since background classes are a majority in every bin of the distribution, e.g. with more than 65 thousand counts in the first bin from the left, this normalization uncertainty becomes overall the dominant uncertainty. As the penalty term cannot reduce the effects of normalization uncertainties, the constraint of the nuisance parameters is dominated by the normalization uncertainty and we cannot expect to find a large difference in systematic uncertainty between $f_{L'}$ and f_{L_Λ} if only a single nuisance parameter is used for both uncertainties.
Instead we split the uncertainty into two independent nuisance parameters one for the



(a) f_{L_Λ} with $\lambda = 100$ **(b)** f_{L_Λ} with $\lambda = 1000$

Figure 4.20.: Comparison of two shapes trained with a different values of λ. Both signal ratios fluctuate in the first and last bin. However in the background ratio, only the left graph shows fluctuations, especially in the last 8 bins.

normalization uncertainty θ_{norm} and one for the shape uncertainty θ_{shape}. Additionally, the influence of λ on the uncertainties is investigated by plotting μ as a function of λ. When splitting the nuisance parameters we would now expect the normalization parameter to stay approximately the same for both $f_{L'}$ and f_{L_Λ} while θ_{shape} should decrease for higher values of λ. It should be noted though that splitting this uncertainties into two completely independent nuisance parameters is not fully correct as both uncertainties are correlated and thus not independent from each other. With this caveat in mind, the results can be seen in figure 4.19. Splitting the uncertainties does indeed meet the expectations as the shape uncertainty decreases steadily for increasing values of λ until it completely vanishes for $\lambda = 1000$. The statistical uncertainty on the other hand increases as a function of λ. This indicates that the systematic uncertainty is statistics driven when training with a penalty term. This is the desired effect of the penalty term as statistical uncertainties can be reduced by using more statistics while systematic uncertainties are irreducible by data alone. On the other hand, the normalization uncertainty increases for higher values of λ as well becoming the dominant systematic uncertainty for $\lambda > 50$. It should be noted that the value of λ has no influence on the normalization parameters given in equation 4.12. The increase of the normalization uncertainty can be attributed to the change in shape due to the decorrelation by the penalty term. Comparing the shapes seen in figure 4.16, the distribution of both signal and background are less steep showing that the NN is less powerful in the separation between background and signal events. Especially the first few and last few bins show a notable difference. Those bins are usually highly pure bins as an event needs the highest score possible to be selected for this bin. A reduction of purity in those bins could lead to a stronger correlation of systematic uncertainties to the signal strength. The correlation to the signal strength for the normalization uncertainty does indeed increase from $\rho = 0.26$ for $\lambda = 0$ to $\rho = 0.34$ for $\lambda = 1000$. The correlation of the shape uncertainty on the other hand decreases from $\rho = 0.61$ for $\lambda = 0$ to $\rho = 0.00$ for $\lambda = 1000$, again showing the decorrelation power of the penalty term.

Another point which can be proven now is the assumption that the constraint on the

signal strength is dominated by the background distribution. For this, the ratios of shifted histogram and nominal histogram for signal and background are compared for $\lambda = 100$ and $\lambda = 1000$ in figure 4.20. There is a difference between the shape of the shifted histogram and the nominal histogram in the first and last bin of the signal ratio for both graphs. This indicates that the large difference in shape uncertainty between the two values of λ, as seen in figure 4.19, cannot or only partially be attributed to the uncertainty of the signal shape. On the other hand, only the background ratio of the left graph has some bins where nominal and shifted histograms do not have the same amount of data. The right graph in contrast has almost perfect agreement between nominal and shifted histogram. The difference between the right graph and the left graph is most apparent in the last 8 bins from the left. Decorrelating the uncertainty of the last 8 bins from all other bins, we find that the correlation of all the other bins to the signal is only $\rho = 0.05$ combined, while the correlation of the last 8 bins is $\rho = 0.33$ combined. Furthermore, the correlation for the last bin – which also showed a discrepancy for the signal ratio – is only 0.02 and lower than the second to last or third to last bin. This indicates that the background ratio is indeed the dominant part for the calculation of the signal strength constraints.

5. Summary and Outlook

This thesis consists of two parts that both investigate the importance and reliability of input variables to NNs.
In the first part of the thesis, the input variables used for the current MVA-based $H \to \tau\tau$ analysis are investigated and reduced to a set of input variables with high impacts on the classification of the data. The pruning is based on the importance of each input variable, which was calculated using Taylor coefficients. The F1-scores, efficiencies and purities of each NN were used to determine after which variable the classification of the NN does not improve anymore. It was shown that using the F1-score and confusion matrices as metrics gives reliable results for an effective pruning of the input variables. For each final state, a core set and extended set of variables were defined, reducing the number of variables from 29 to 11 and 16 respectively. The NNs trained on the core set and extended set of variables given in table 3.2 achieve similar results to the reference NN as seen in table 3.3.
Furthermore, the input variables of the core and extended set were aligned across all years and most final states. This was used to train NNs with a combined dataset of all years for each final state, which reduces the number of NNs needed for classification from 12 to 4. The discrete input variable `era` was used to discern to which year a given event belongs to. The classification achieved by those NNs is comparable to the NNs trained on separated datasets. The synchronization of variables gives the possibility to introduce another technique to the training of the NNs: By randomizing the `era` variable of the signal events for each year, the NNs could be deprived of the information from which year a signal event was coming from. This was used to recover the signal from years where no data of signal events were available. It was shown that the signal strength constraints of the NNs trained with this randomization technique were again comparable to the reference NNs.
While the pruning technique presented here can be applied to most datasets, the results of the pruning are specific to the dataset used for this thesis. A dataset with a different event selection might have a different importance ranking for the input variables and thus a different core and extended set. This caveat should be taken into consideration when applying the results of this thesis to any datasets with different event selection. Furthermore, it was shown that the improvement in signal strength constraints for the randomization techniques must be caused by artifacts in the data. When using conditional NNs, an effort should be made that there are no artifacts in the dataset of one year that is not present in the dataset of other years as those artifacts might lead to an over- or underestimation of the final result of the analysis, which could cause a bias under the wrong circumstances.

In the second part of the thesis, two techniques are presented that implement prior knowledge of systematic uncertainties directly into the NN training. The first technique leverages adversarial NNs to decorrelate the output of the NN against input variables affected by systematic variations. It is shown in figure 4.2 that the setup with an adversarial NN does indeed reduce the dependency of the NN output on systematic variations. Taylor coefficients are used to visualize the reduction of importance of the input variable affected by systematic

uncertainties. However, due to the complexity and limitations of these adversarial NNs, they cannot be universally used in high-energy physics analyses to incorporate systematic variations in NNs.
The second technique is a novel strategy to implement the systematic variations. In this technique, systematic variations are introduced in the training of the NN via a penalty term of the loss function. The usage of a penalty term instead of an additional NN reduces the complexity of the task to only a few additional hyper-parameters. Additionally, the penalty term allows the inscription of systematic variations via statistical weights. The technique was demonstrated on the same pseudo-experiments used for the first technique and the results for the shape of the histograms and the Taylor coefficients are very similar. This technique also allows to incorporate multiple systematic variations by adding a penalty term to the loss function for each uncorrelated systematic uncertainty. This was demonstrated by enhancing the pseudo-experiment with an additional variable and introducing a second systematic uncertainty to the dataset. The shape and Taylor coefficients again show a reduction of the dependency of the NN output on the systematic variations of the input variables. Lastly, the new technique was tested on a high-energy physics example in form of the Higgs boson machine learning challenge released by the ATLAS collaboration [48]. In this more complex example, the technique again successfully reduces the impact of the systematic uncertainty, which affects multiple input variables this time, on the NN output as seen in figure 4.19. The results show that systematic uncertainties are converted into statistical uncertainties. As systematic uncertainties play a dominant role in measurements of high-energy physics, it is of increasing interest in this field that the output of an NN is less prone to systematic uncertainties.
While this technique does indeed produce a more robust NN, it also reduces the ability of the NN to separate background and signal events. It was shown that the decorrelation against input variables obliterates most information of this input variable for the NN. Fine-tuning the hyper-parameter λ can limit the amount of information obliterated by the penalty term, but the decorrelation against the systematic uncertainty will also be limited. To improve this behavior, a loss function would be needed, which adjusts the level of reliability given to an input variable without human input. The loss function would need to find a balance between the systematic uncertainties and separating power of an input variable. This could be achieved e.g. by a loss function which is based on the likelihood-based analysis of the NN output shown in section 2.4.2.

Appendix

A. On the relation between the maximum likelihood estimate and the cross entropy for neural networks

A.1. Maximum likelihood estimator

The likelihood function for the outcome of a statistical sample of length N is given by the product of the probabilities $P_j(\mathbf{x}|\theta)$ to make the individual observations $\{j : \mathbf{x}\}$ with $\mathbf{x} = \{x_i\}$ for a given parameterset $\theta = \{\theta_k\}$ of the hypothesized model:

$$L(\theta) = \prod_{j=0}^{N} P_j(\mathbf{x}|\theta) \tag{A.1}$$

The maximum likelihood principle defines the best estimators $\hat{\theta} = \{\hat{\theta}_j\}$ of θ (MLE) as those parameters that maximize $L(\theta)$. To find an extreme value of $L(\theta)$, the derivatives of Eq. (A.1) with respect of the parameters θ have to be calculated. Since $L(\theta)$ can become very small for large sample lengths N its handling is usually facilitated using the logarithm of $L(\theta)$. Due to the properties of the logarithmic derivative

$$\frac{\mathrm{d}}{\mathrm{d}x}\Big(\ln\big(f(x)\big)\Big) = \frac{1}{f(x)}\,\frac{\mathrm{d}}{\mathrm{d}x} f(x) \tag{A.2}$$

the MLE based on the logarithmic likelihood is equivalent to the MLE based on the likelihood itself. In practice the logarithmic likelihood function is moreover often multiplied by a factor -1, turning the maximum into a minimum. The negative logarithmic likelihood function (NLL) finally is defined as:

$$\mathcal{L} = -\ln\left(L(\theta)\right) \tag{A.3}$$

The MLE is always efficient and consistent [50, 51].

A.2. Maximum Likelihood function for a neural network

For illustrative purposes only we give the functional form of the neural network (NN) output. For this purpose, and without loss of generality, we use a simple NN with a single hidden layer $\mathbf{h} = \{h_k\}$, with nodes k, to be used for binary classification with a single output $\hat{y}$. The output function for this NN can be written as

$$\begin{aligned} \hat{y} &= \mathcal{O}\left(\sum_{k=0}^{m} h_k(x_i, \theta_{ik}, \theta_{bk})\, \theta'_k + \theta'_b\right) \\ h_k &= \mathcal{H}\left(\sum_{i=0}^{n} x_i \theta_{ik} + \theta_{bk}\right) , \end{aligned} \tag{A.4}$$

where $\mathcal{H}$ and $\mathcal{O}$ are the activation functions of the hidden and output layer, respectively, m is the number of nodes of the hidden layer and n the number of inputs x_i. The parameters $\{\theta_{i,bk}\}$ and $\{\theta'_{k,b}\}$ correspond to the trainable parameters of the hidden and output layer, respectively. In the following we require $\mathcal{O}$ to be the sigmoid function with values between 0 and 1, indicating the degree of belief for associating a single observation with inputs $\mathbf{x}$ to a given ground truth class, without knowing this truth. We thus interpret Eq. (A.4) for the value of $\hat{y}$ as a model for the probability

$$P(\mathbf{x}|\theta) = \hat{y} \qquad \text{with:} \qquad \theta = \{\theta_{i,bk}\} \cup \{\theta'_{k,b}\} , \tag{A.5}$$

to associate an observation with inputs $\mathbf{x}$ to a given ground truth class. To generalize this probability interpretation to all possible observations irrespective of their ground truth we introduce the prior probability

$$y = \begin{cases} 1 & \text{event in class} \\ \\ 0 & \text{else,} \end{cases} \tag{A.6}$$

which is 1 if the sample belongs to the class in consideration and 0 if it belongs to the complement class. In this way the probability of Eq. (A.5) can be rewritten as:

$$P(\mathbf{x}|\theta) = \hat{y}^y (1-\hat{y})^{1-y} = \begin{cases} \hat{y} & \text{event in class} \\ \\ 1-\hat{y} & \text{else.} \end{cases} \tag{A.7}$$

The dependency of $P(\mathbf{x}|\theta)$ on $\mathbf{x}$ and θ is given by the definition of $\hat{y}$ in Equation (A.4). For N independent observations j Equation (A.7) expands to

$$\begin{aligned} L(\theta) &= \prod_{j=0}^{N} P_j(\mathbf{x}|\theta) \\ &= \prod_{j=0}^{N} \hat{y_j}^{y_j} (1-\hat{y_j})^{1-y_j} \end{aligned} \tag{A.8}$$

$$\mathcal{L}(\theta) = -\sum_{j=0}^{N} \Big(y_j \ln(\hat{y_j}) + (1-y_j) \ln(1-\hat{y_j}) \Big) ,$$

which can be interpreted as the likelihood function to observe a given permutation of observations j being associated to the given class or its complement class in a sample of length N. In this interpretation the parameters θ defined in Equation (A.5) can be identified as the adjustable parameters of a likelihood model equivalent to the parameters θ of the hypothesized model assumed for Equation (A.1). The binary cross entropy (CE) for the true distribution of a given parameter y and a probability distribution $P(\mathbf{x}|\theta)$ is defined as

$$H(y, P(\mathbf{x}|\theta)) = -\sum_{j=0}^{N} y_j \ln\left(P(\mathbf{x}|\theta)\right) + (1 - y_j) \ln\left(1 - P(\mathbf{x}|\theta)\right) \tag{A.9}$$

Comparing Eq. (A.9) to Eq. (A.8) one can see that both equations are identical. So minimizing the CE w.r.t. θ is equivalent with finding the MLE of the specific likelihood model described above. A more complete and general mathematical description can be found in Ref. [28]. The arguments outlined above can be extended to multi-classification problems based on the softmax activation function for the output layer.

A.3. Caveats

In the above discussion of the equivalence of the CE and the MLE estimate the following points should not be omitted:

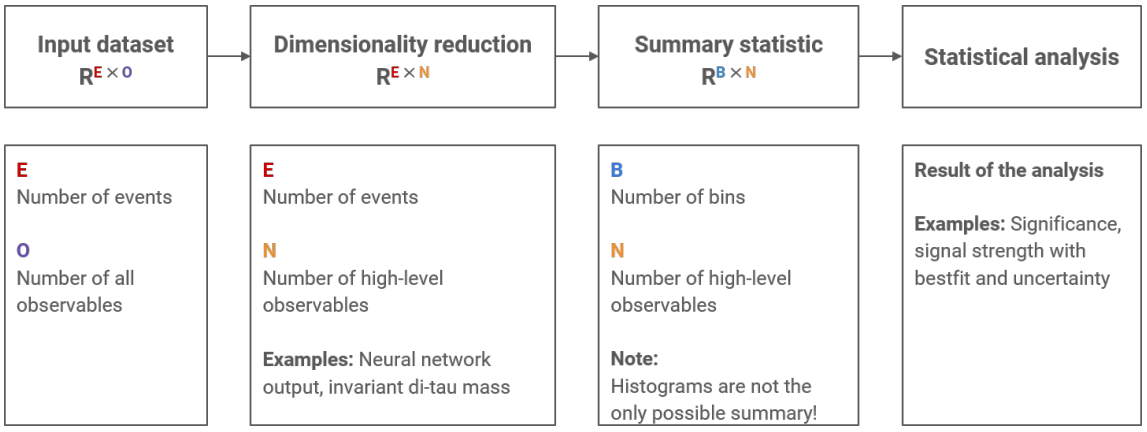
- In the interpretation given above the CE corresponds to the likelihood for a permutation of observations j being associated to a given class or its complement class in a sample of length N, without further uncertainties.

- While Eqs. (A.8) and (A.9) are equivalent, this is strictly only true within the boundary conditions and interpretation given above. The CE, which has its origin from information theory, cannot be related to a statistical likelihood function, in general.

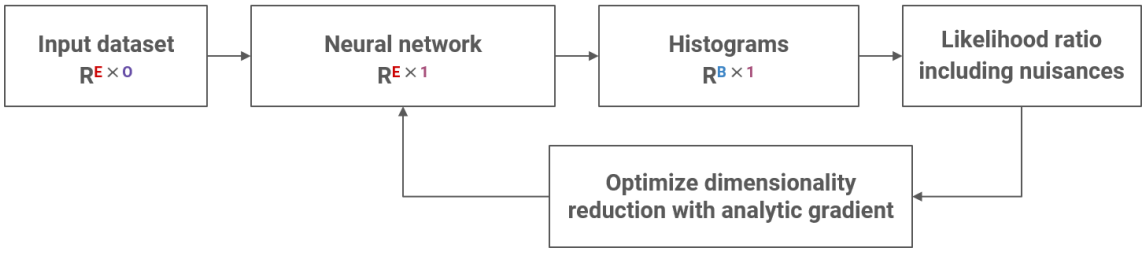
- The probability interpretation for $\hat{y}$ in the sense of a degree of belief associated to the observation $\mathbf{x}$ relies on a strictly Bayesian probability interpretation. In the frequentist approach and the example given above a single sample is either associated to the given class or to its complement. A probability assignment $P(\mathbf{x}|\theta)$ to an observation different from 0 or 1 makes no sense. Also since the NN is deterministic the same set of inputs $\mathbf{x}$ will always lead to the same value of $\hat{y}$, with a probability of 1.

B. Optimization of the NN on a likelihood-based analysis

The results of the NN with additional penalty term as introduced in section 4.3 does not necessarily indicate whether the NN performs better in the statistical inference of its output compared to a NN trained without consolidation of systematic uncertainties. As can be seen in figure 4.19, the reduction of the systematic uncertainty introduces a larger statistical and normalization uncertainty and does not lead to a better overall constraint of the signal strength constraints. In general the objective of the loss function of NNs do not necessarily coincide with the goal of the actual physical analysis described in section 2.4.2. In figure B.1 (top row), a rough description of the HEP analysis in its current implementation for the $H \to \tau\tau$ analysis is given. As one can see, the objective of the loss function of the NN is separated by an additional step from the actual end result. The end results is usually



(a) Current end-to-end analysis.



(b) New end-to-end approach.

Figure B.1.: The top graph shows the current implementation of the $H \to \tau\tau$ analysis. There is a clear disconnect between the objective function of the NN and the general objective of the analysis. While correlated, certain analysis tools are not considered when minimizing the NN loss. The bottom graph shows the potential new approach of the analysis. The complete analysis will be used in the NN loss function and the physics objective will be directly used to optimize the weights of the NN.

defined by the significance or signal strength with the best fit values and constraints, which takes all systematic uncertainties into account or uses additional cuts and binning option to further enhance and improve the results. Optimally, we would want to include those systematic uncertainties and binning options already in the NN training to achieve the best fit results on a given dataset.

The main loss function for NN classification tasks is the CE loss function. In a binary classification task, the cross entropy is defined as

$$H(y, P(\mathbf{x}|\theta)) = -\sum_{j=0}^{N} y_j \ln\left(P(\mathbf{x}|\theta)\right) + (1 - y_j) \ln\left(1 - P(\mathbf{x}|\theta)\right) \,. \tag{B.1}$$

with $P(\mathbf{x}|\theta)$ as the NN output and y_j the true label for the given event. This objective function separates the two classes to the left or right side of a histogram maximizing the efficiency. While this is indeed an MLE as seen in appendix A, it is not necessarily what the statistical inference cares about in the end, e.g. the binned profiled log likelihood does not care about the exact orientation of the distributions in the histogram. In other words: It is of no particular interest what the histogram looks like as longs as the significance in the end is the highest given the data. This leads to an obvious solution: One could take the significance of the end results as the final result for the loss function and propagate the error all the way to the NN output. This way, the NN loss function would be directly related to the

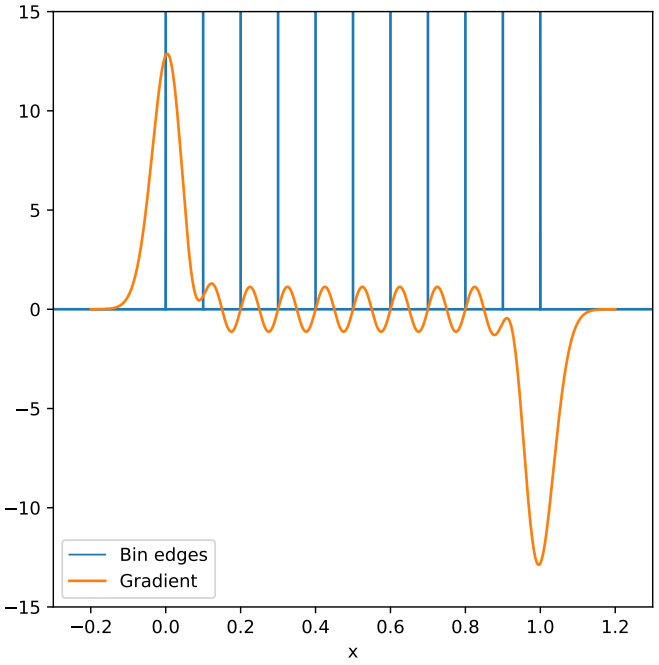


Figure B.2.: The binning is done according to equation B.3. Extending this formula for 10 bins results in the combined gradient shown above.

end objective of the statistical inference and the weights of the NN are directly optimized for maximization of the significance. This solution brings another advantage: The significance in HEP physics is often calculated using a profiled log likelihood scan as described in section 2.4.2. As the implementation of systematic uncertainties in the statistical inference of binned profiled log likelihood are well understood and an accepted practice in HEP, if this approach is used as a loss function for the NN, systematic uncertainties are naturally included in the NN. The end result would be an NN that is specialized to complete the criteria of a physics analysis. The general approach can be seen in figure B.1.

For the actual statistical inference, the NN output needs to be binned again. Of course one could simply take the approach with the Gaussian filters again explained in section 4.3. As mentioned though, those Gaussian filters only approximate the actual bin count. This was not necessarily a problem in the previous task of decorrelation, but as the goal is now to make a precise measurement, a binning approach which is not an approximation would be more appropriate to use. As previously explained, equation 4.4 does not have a well-defined gradient and can not be differentiated for back propagation. While this is true from a mathematical point of view, practically this can simply be by-passed by injecting a custom derivative of the Heaviside theta function in the code. It is possible to simply assign the function a derivative which will be used in back propagation. A reasonable derivative for a the binning function is already known: The differentiation of the Gaussian filters did exactly what was expected of them. Taking the derivative of the Gaussian distributions used in section 4.3 and using this derivative for equation 4.4, a completely differentiable NN function can be formulated. The definition for the binning function is

$$\mathcal{M}_k = \theta(\mathbf{f}(\mathbf{x}) - a) \cdot \theta(\mathbf{f}(\mathbf{x}) + b) \tag{B.2}$$

$$\mathcal{M}'_k = -\mathcal{G}_k(\mathbf{f}(\mathbf{x})) \cdot \frac{\mathbf{f}(\mathbf{x}) - m}{\varphi^2}\,, \tag{B.3}$$

where m is the center and φ the half-width of the bin k. An example of the resulting

combined derivative with 10 bins can be seen in figure B.2. In the combined gradient, values close to the center of the bin are pushed more towards the center by this gradient while values close to the edge of the bin are pushed away from the center. On the edges of the distribution, large gradient hills have formed due to the combination of all residual positive or negative values from all other Gaussian gradients used and due to the absence of terms to the left or right of those peaks that would lower their values. The exact form of the gradient does not matter as long as the gradient assigns a different gradient to events in one bin to allow events to be passed from one bin to another. The peaks on the edges are also beneficial in this case as they push the overall distribution more to the center and thus limiting edge effects that can disrupt the training process, e.g. when all events are pushed into a single bin at the edge. After binning the values into n_k bins, the systematic uncertainties can again be applied via reweighting to get an up- and down-shifted histogram. This systematic uncertainties σ can then directly be applied in the following.

For simplicity sake, the approach is described on a binary classification task with a signal s and a background b. As described in section 2.4.2, the statistical analysis is done by introducing the signal strength modifier $\mu = \frac{\sigma}{\sigma_{\text{SM}}}$. The signal strength modifier shifts the signal expectation s. For $s = 0$ we get the background-only hypothesis and for $s = 1$ the background plus signal hypothesis. Using Poisson distributions $\mathcal{P}$, the likelihood can be formulated as

$$\mathcal{L}(\text{data}|\mu, \theta) = \prod_i^N \mathcal{P}(\text{data}_i|\mu \cdot s_i + b_i + \theta \cdot \sigma_i) \tag{B.4}$$

where σ_i are the systematic uncertainties as mentioned above. The systematic uncertainties increase or decrease the amount of background seen. The test statistic we want to calculate can than be formulated as

$$q_\mu = -2 \ln \left(\frac{\mathcal{L}(\text{data}|\mu, \hat{\theta}_\mu)}{\mathcal{L}(\text{data}|\hat{\mu}, \hat{\theta})} \right) \qquad 0 \leq \hat{\mu} < \mu \,. \tag{B.5}$$

$\hat{\mu}$ and $\hat{\theta}$ are the global best fit parameters, while $\hat{\theta}_\mu$ is the θ which maximizes $\mathcal{L}$ for a given μ. If Wilk's theorem [32] holds true, which means there is a very large sample size N, the significance z in quantiles of a normalized Gaussian distribution can then be calculated by calculating q_μ at $\mu = 0$:

$$z = \sqrt{q_{\mu=0}} = \sqrt{-2 \ln \left(\frac{\text{Poisson}(\text{data}|b + \hat{\theta}_\mu \cdot \sigma)}{\text{Poisson}(\text{data}|\hat{\mu} \cdot s + b + \hat{\theta} \cdot \sigma)} \right)} \,. \tag{B.6}$$

Maximizing this significance as the loss function (for technical reasons a minus sign is added to minimize the loss function), the physics objective would be directly optimized. From a technical point of view, one would have to calculate both global fit parameters $\hat{\mu}$ and $\hat{\theta}$ as well as $\hat{\theta}_\mu$ and the ratio of them to calculate a single significance and gradient step. This is not only computationally expensive even for a single nuisance parameter θ but also leads to a highly complex loss function which changes its topology according to the new best fit parameters. While finding a global minimum cannot be guaranteed for any gradient descent optimization, even finding a local minimum turns out to be a complicated and unstable task in such an environment due to the changing topology of the loss function.

A less computational intense and complex version of this loss function was first proposed by [52]. The first difference to the test statistic defined in B.5 is that an Asimov dataset is used instead of actual observed data. This is a quite common occurence in high energy physics as usually data is only used after the statistical model is finalized. This way any bias due to model fine-tuning can be avoided. In an Asimov dataset or in the Asimov likelihood $\mathcal{L}_A$, the data is replaced by the nominal expectation $s + b$ measured from the NN output:

$$\mathcal{L}_A(s + b|\mu, \theta) = \prod_i^N \mathcal{P}(s_i + b_i|\mu \cdot s_i + b_i + \theta \cdot \sigma_i) \tag{B.7}$$

This formulation automatically fixes the minimum of q at $\mu = 1$. In fact, only the minimization of a variable is of interest and the actual value of this variable is not important for the minimization task. Therefore, the denominator of B.5 is no longer needed, as this denominator becomes constant for an Asimov likelihood. The best global fit parameters will always be $\hat{\mu} = 1$ and $\hat{\theta} = 0$. As a constant term will simply be eliminated in the back propagation, it is no longer needed. Furthermore, the valuable Fisher information matrix $I(\mu, \theta)$ can now be obtained by calculating B.7 at $\mu = 1$, taking the negativ logarithmic expression of this and computing the Hessian matrix with regards to all nuisance parameters μ and θ:

$$I(\mu, \theta) = \frac{\partial^2}{\partial\mu\partial\theta} \left(-\ln \mathcal{L}_A(s + b|\mu, \theta)\right) \tag{B.8}$$

In case of only one nuisance parameter θ, this is a simple 2×2 matrix. If $\hat{\theta}$ and $\hat{\mu}$ is now assumed to be unbiased estimators of the values of both nuisance parameters (which is given by construction, see [52]), we can apply the Cramér-Rao lower bound [32] to the covariance matrix and the inverse of the Fisher information matrix:

$$\mathrm{cov}(\mu, \hat{\theta}) \geq I(\mu, \theta)^{-1} \tag{B.9}$$

This means that the inverse of the computational easily calculated Fisher information matrix I^{-1} gives a accurate approximation of the expected variance of each term as well as the correlations between the nuisance parameters. In fact the diagonal elements of the inverse Fisher information matrix $I^{-1}_{k,k}$, which are the expected variances, can now be used to optimize the NN on any given optimization goal for all nuisance parameters. Of course the most logical diagonal element to pick for a physics analysis is the $I^{-1}_{\mu\mu}$ element, which directly correlates to the measured significance. This element corresponds to the curvature of the parabola which is normally calculated by scanning equation B.5 for different values of μ and fit a parabola on those values. This again only holds true in the limit of large sample sizes N. An example for the similar results of a parabola centered on $\mu = 1$ with curvature $I^{-1}_{\mu\mu}$ and a parabola fitted by scanning μ can be seen in figure B.3. As is typical when using an Asimov dataset, instead of the significance, the constraints on the signal strength for the 68 % CI are calculated by taking the value of μ at $q = 1$. Both upper and lower limits can be calculated this way. The lower the constraints, the better the separation of the data with regards to the systematic uncertainties. As we do not scan q anymore, we can simply take the value of μ for the parabola seen in figure B.3 at $q = 1$, which will give us a good approximation on the real constraints. All in all, the final loss function L will be

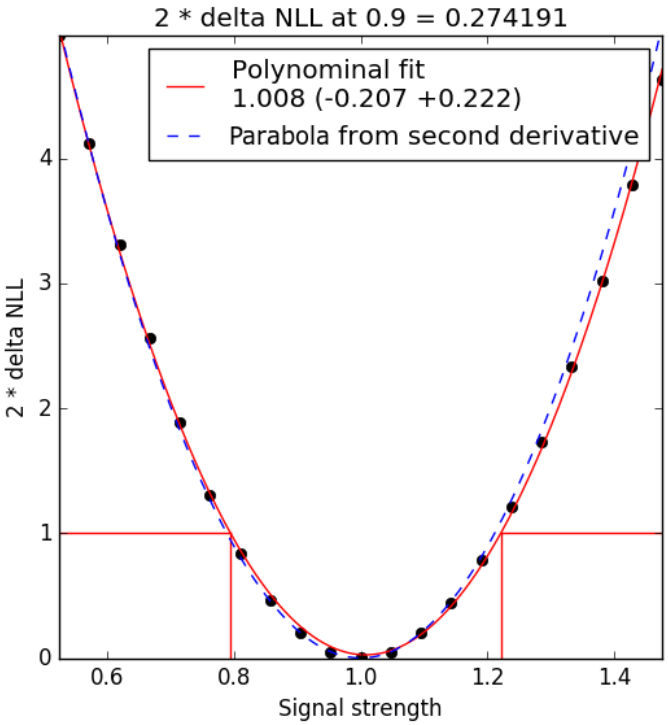
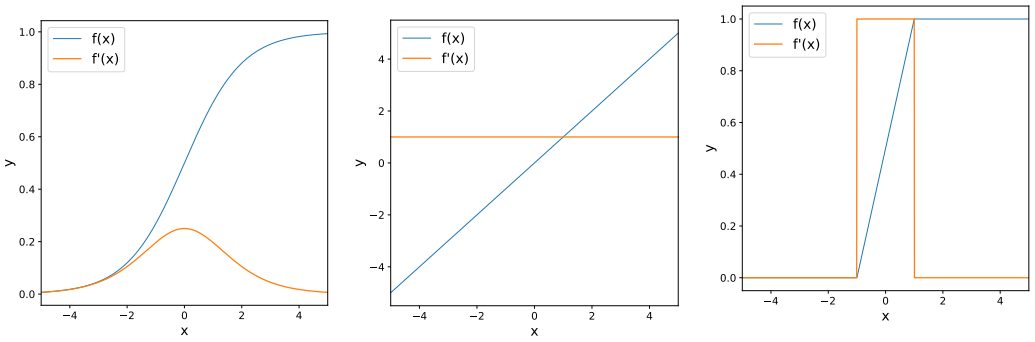


Figure B.3.: The graph shows the points and fitted parable given by a normal profiled log likelihood fit. The different values are calculated by injecting different values for μ into equation B.5. The blue dotted line is the parable one gets by calculating the inverse Fisher-matrix and taking the diagonal element of μ as the curvature for a parable.

$$L = \sqrt{I_{\mu\mu}^{-1}}\,. \tag{B.10}$$

Minimizing this value instead of equation B.6 results in a much more stable and less computational intensive training than before while still approximating the actual physical objective in a reasonable manner.

The architecture of the neural network is again kept simple and is largely identical to the NN described in section 4.3.1. However, a closer look at the NN output has to be taken: As the NN outputs has to be binned in a certain range, the activation function of the output layer should already map the output values to the accepted range. The simplest solution would be the sigmoid activation functions which outputs values between $[0, 1]$. A caveat of this choice is the derivative of the sigmoid function. As seen in figure B.4 (a), the derivative has large values near $x = 0$ and values close to 0 otherwise. For binary classification tasks where the distributions are sorted either to the left or the right hand side, this behavior is welcome as it aids this process by giving values close to 0.5 a higher value for the back propagation. In this case though, the shape of the distributions do not matter. In fact, the NN is not given any information on how to sort between signal and background. The NN could potentially choose any bin for the signal and any other bin for the background. This also includes the bins in the middle. Since the sigmoid activation encourages higher changes for value around 0.5, while suppressing changes of values near 0 and 1, the natural sorting of the loss function might get biased by the sigmoid activation function. If we take this one step further, a more natural choice for the loss function would be a combination of a sigmoid function and a simple linear function:



(a) Sigmoid activation. **(b)** Linear activation. **(c)** Activation according to equation B.11.

Figure B.4.: Comparison between sigmoid activation, linear activation and equation B.11. Also shown are the derivatives of each activation function.

$$f(x) = \begin{cases} 0 & x < -1 \\ x & -1 < x < 1 \\ 1 & x > 1 \end{cases} \tag{B.11}$$

$$f'(x) = \begin{cases} 0 & x < -1 \\ 1 & -1 < x < 1 \\ 0 & x > 1 \end{cases} \tag{B.12}$$

This activation function shown in figure B.11 (c) is not biased for any value inside its defined range $[-1, 1]$. The range of this activation function can be decided on based on the expected output values of the last NN layer. Taking this even further, one could simply take a linear activation function with no boundaries. This would mean that any value would be produced by the NN and the produced histogram of the loss function would have to take this into account. A simple solution to this problem would be to define over- and underflow bins for this histogram. Over- and underflow bins are also commonly used in analysis packages like `ROOT` to catch values outside of the defined range of the analysis. In other words: Using over- and underflow bins would be closer to the usual analysis routine done for the statistical inference described in section 2.4.2. All of the activation functions described here can be used to solve this problem, but for simplicity sake the sigmoid function was used for this analysis.

Additionally, it was found that the initialization of the weights of the NN can have a great impact on the over all performance of the training. In particular it was found that certain initializations, depending on the random seed used, can have their output values all in a range that would result in – for example – only zero values from the sigmoid activation functions. While this is not a problem for a normal CE loss, for the caculation of the loss L the output values would all be sorted into the same bin. This would result in a non-invertible Fisher information matrix I and thus the training would fail on the very first step as no gradient for back propagation can be formulated. To by-pass this problem and ensure that all weight initialization seeds result in approximately the same starting point for all NNs, a preliminary training was implemented. This preliminary training trains the NN on the CE loss function for 1000 steps. This way it can be ensured that the output values of the NN are all distributed over the complete range of $[0, 1]$. After 1000 steps the loss function will then be simply switched to L.

The simple dataset already used in section 4.2 and 4.3.1 is used again here to test this end-to-end implementation. In addition to the shifts, an imbalance of signal and background

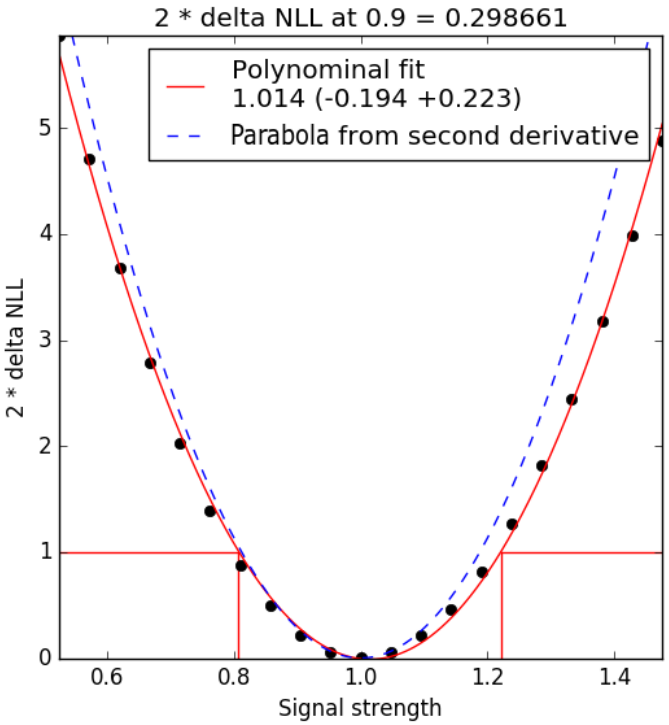
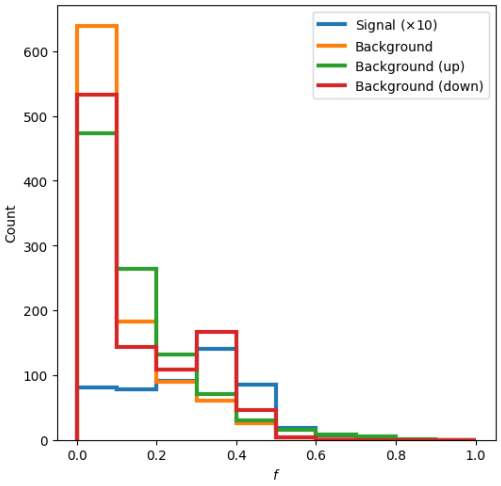


Figure B.5.: The graph shows the results of the statistical inference when directly applied to the distributions of the dataset given by figure 4.4. This results can be used to quantify the results of the new loss L.

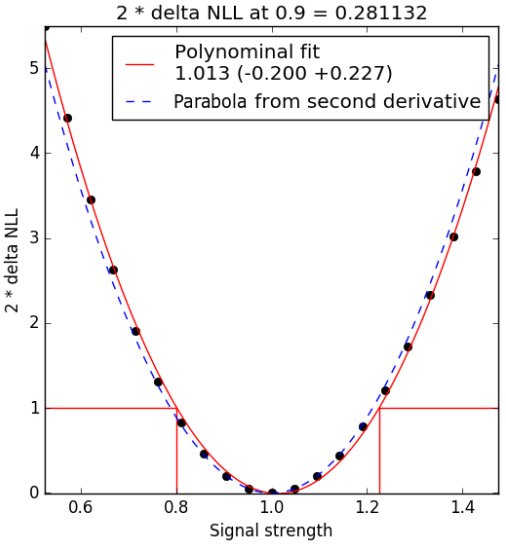
samples is introduced to the dataset to further enhance the complexity of the task and bring it closer to an actual physical measurement. The expectation of the signal samples is set to 100, while the expectation of the background samples is set to 1000. To compare the results of the statistical inference of the CE loss and the new loss L to a fixed point, first a statistical inference is performed on the dataset without the usage of an NN. As the exact distributions used for creating the datasets are known, one can simply use the probability density functions (PDF) to get the probability for signal and background events for any bin in a finely binned two-dimensional histogram by integrating the PDFs over the bin edges. For this study, 50×50 bins for the statistical inference of the two-dimensional histogram are used. Afterwards the 2500 bins are flattened to get a one dimensional histogram. Using 2500 bins for a binned profiled log likelihood fit is close to using an unbinned profiled log likelihood fit which theoretically would result in the best fit values. The result of the statistical inference of this histogram should therefore be a well-defined reference point on which to measure the success of the training of the NN. The results of this comparison test are shown in figure B.5.
Afterwards, the NN is trained with the new loss function L. The number of bins is chosen to be $n_k = 10$ and to be equidistant in the range $[0, 1]$. The resulting histogram of the NN output can be seen in figure B.6 (left). Other than the penalty term of section 4.3, this loss function does not necessarily reduce the uncertainty bands of the background distribution, which is to be expected considering that the loss L does not have any indicator to decorrelate against the shifted backgrounds. It can also be seen that the peak of the signal and background distribution is chosen rather arbitrarily by the NN. As long as they are not in the same bin, the actual distant between the distributions does not matter. This is again the behavior we would expect as the distant of the bins does not matter for the calculation of the signal strength constraints as long as the distributions are separated. In general, the distributions of signal and background are kept close to each other by the NN. A reason for this might be that exchanging events between the two distributions is easier

this way and thus only small changes of the weights are needed to further minimize the constraints. The signal strength constraints calculated by using the statistical inference framework from the $H \to \tau\tau$ analysis is shown in figure B.6 (right). Comparing this to the signal strength constraints given in figure B.5 one can clearly see that the results are comparable. The small difference in constraints can be attributed to the number of bins $n_k = 10$ used for training. A higher number of bins is computational more expensive, but leads to a better constraint. This can be seen in figure B.6 (bottom row). Figure B.7 shows the signal strength constraints for a NN trained with CE loss and the nominal dataset. One can see that the constraints are getting worse compared to figure B.6 and figure B.5. The new end-to-end training can thus be considered superior for this simple pseudo-experiment.

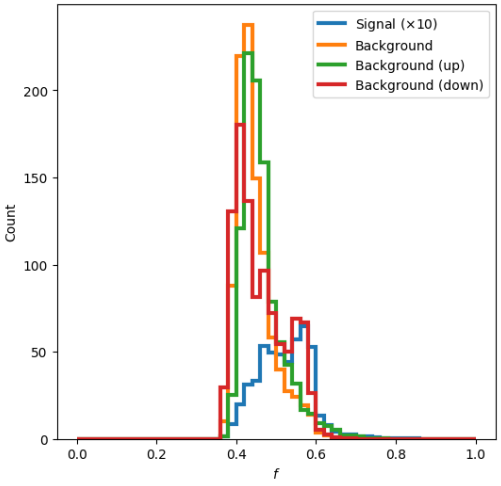
As already discussed, the cross entropy finds the MLE for a given problem if there are no systematic uncertainties present in the dataset. This case will be called "statistics only" in the following. As a MLE is both consistent and efficient, the MLE usually can be seen as the best possible estimator for a given problem. On the other hand, the statistical inference loss L should also find the MLE for a given problem simply by construction. To test this hypothesis, the signal strength constraints of cross entropy and L are compared in a statistics only case. Using the dataset from above but removing all systematic uncertainties, both loss functions are compared. As shown in figure B.8, we indeed find that L and CE have comparable signal strength constraints. Of course the histograms produced by both loss functions are widely different. This illustrates that there is no unique solution to this problem and the solution depends on the choice of the loss function.



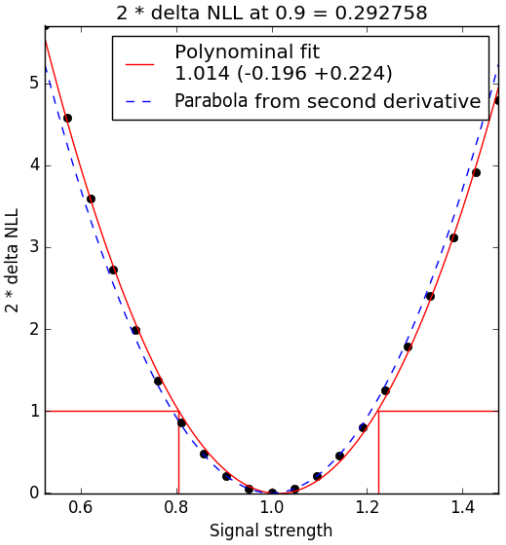
(a) Histogram of NN output for 10 bins.



(b) Calculated constraints using statistical inference.

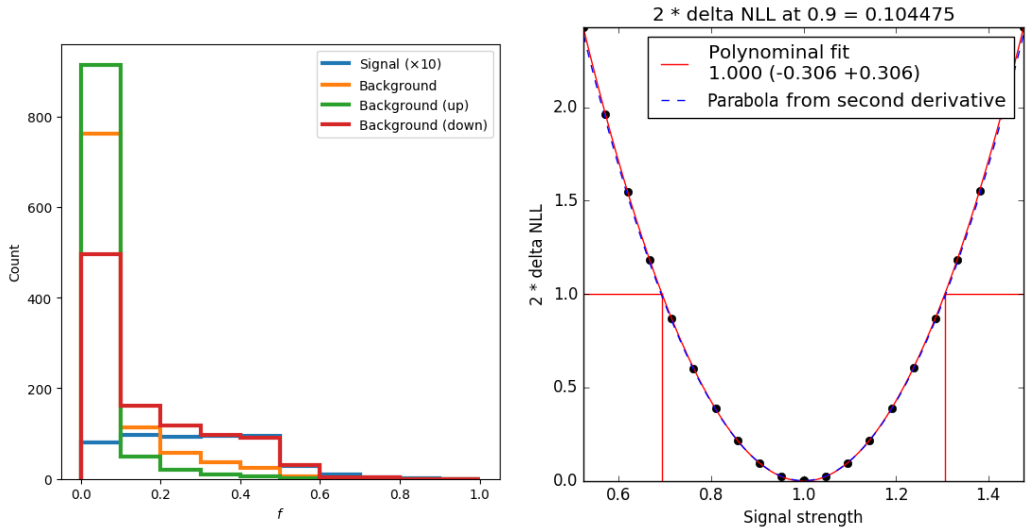


(c) Histogram of NN output for 50 bins.



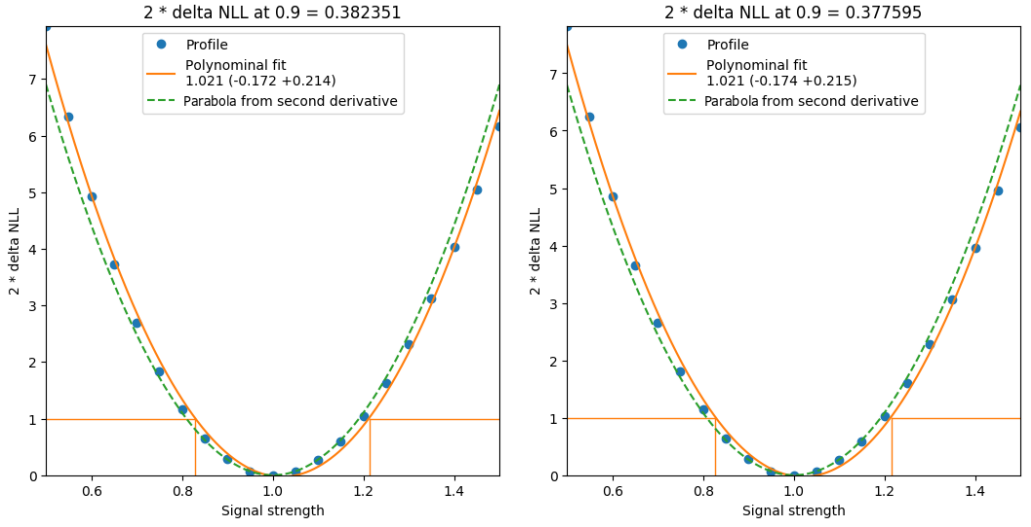
(d) Calculated constraints using statistical inference.

Figure B.6.: The left plots show the NN output for the loss function L for 10 and 50 bins. The distributions show a clear separation between signal and background. Up and down variation are still visible in contrast to figure 4.8. The right plots show the signal strength constraints calculated using statistical inference on the histogram given on the left side. The constraints are comparable to the constraints given in figure B.5. One can see a slight improvement of signal strength constraints with a higher number of bins (bottom row).



(a) Histogram of NN output using cross-entropy. **(b)** Calculated constraints using statistical inference.

Figure B.7.: The left plot shows the NN output for the CE loss function. The right graph shows the statistical inference done on the histogram given on the left side. The signal strength constraints are worse than the constraints given in figure B.6 and figure B.5.



(a) End-to-end training **(b)** Cross-entropy training

Figure B.8.: Both signal strength constraints given in the graphs show comparable results for $n_k = 10$ bins. Both NN find the MLE for this case, even though their solutions can be entirely different.

C. Input variables selection plots

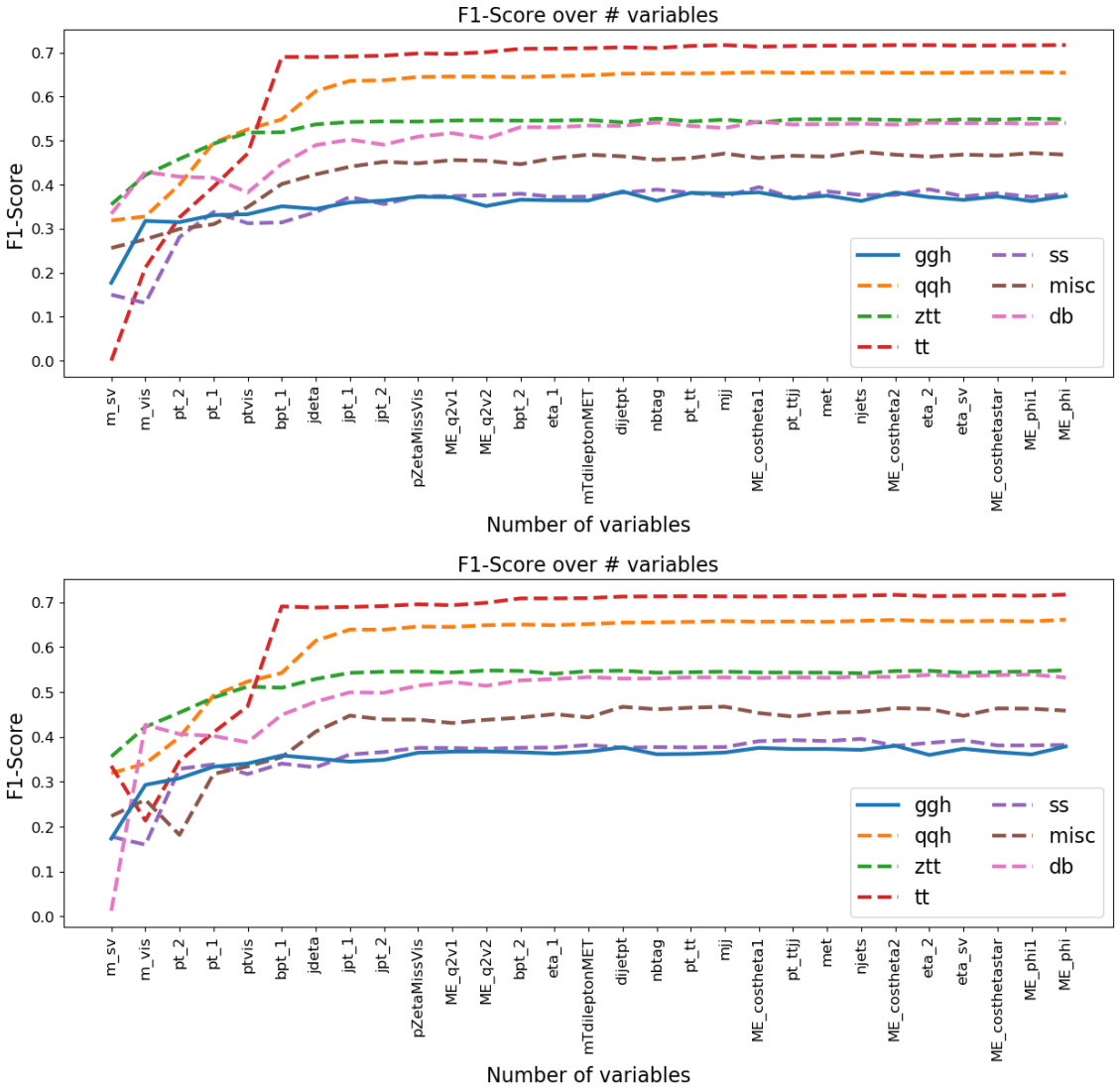


Figure C.9.: F1 score of the ggh output class (marked as the solid line) and the $e\mu$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

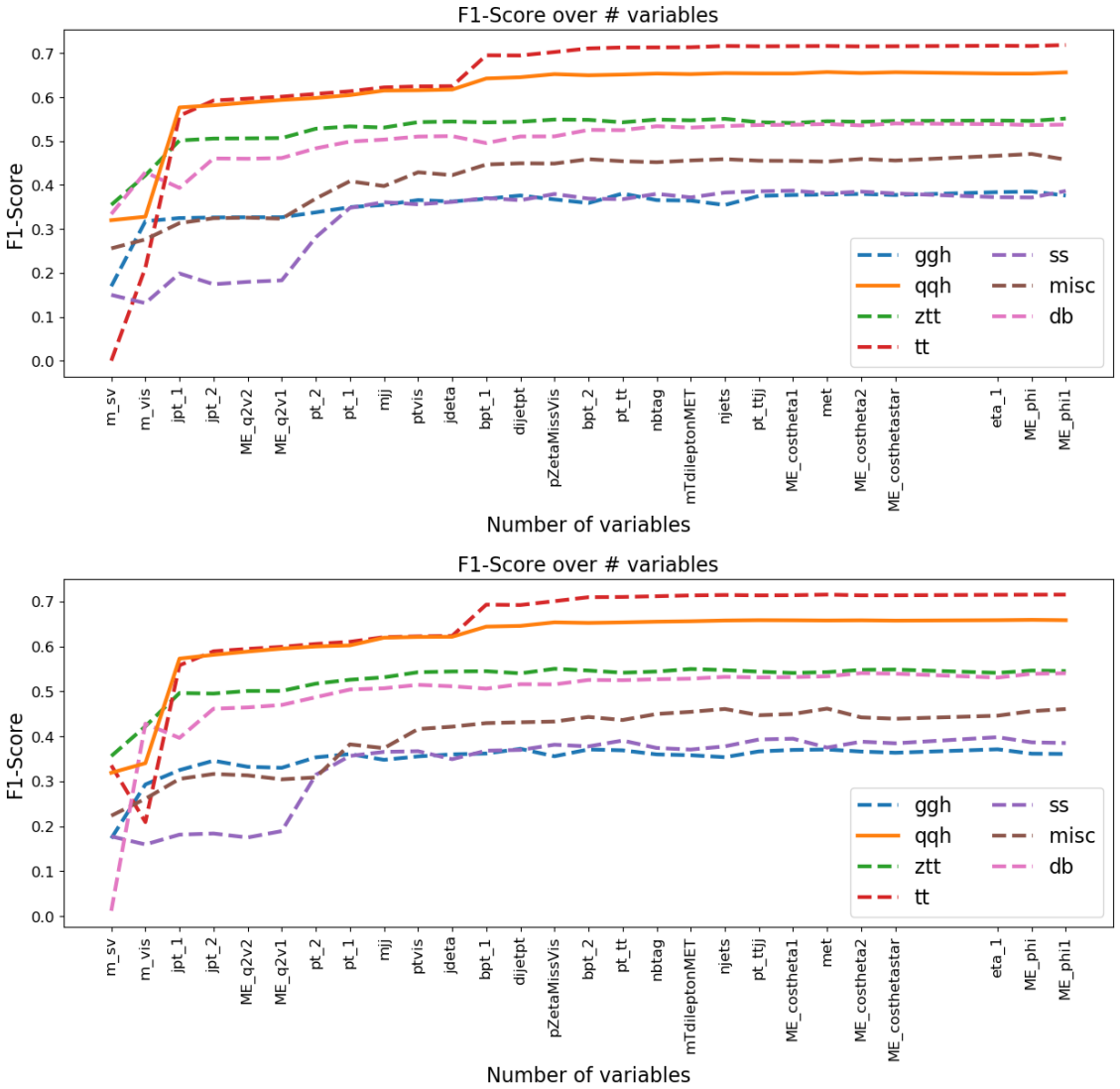


Figure C.10.: F1 score of the `qqh` output class (marked as the solid line) and the $e\mu$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

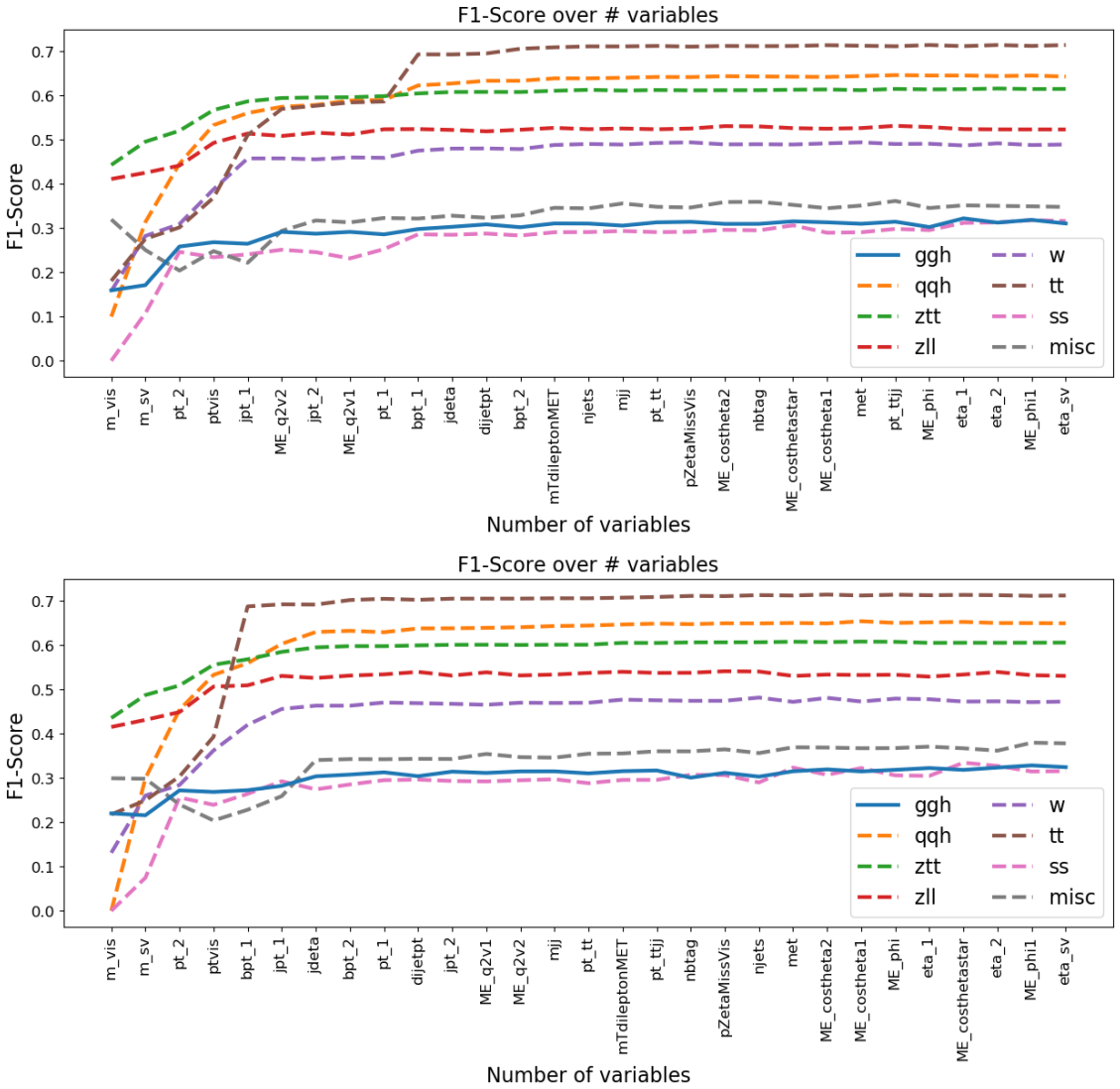


Figure C.11.: F1 score of the ggh output class (marked as the solid line) and the $e\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

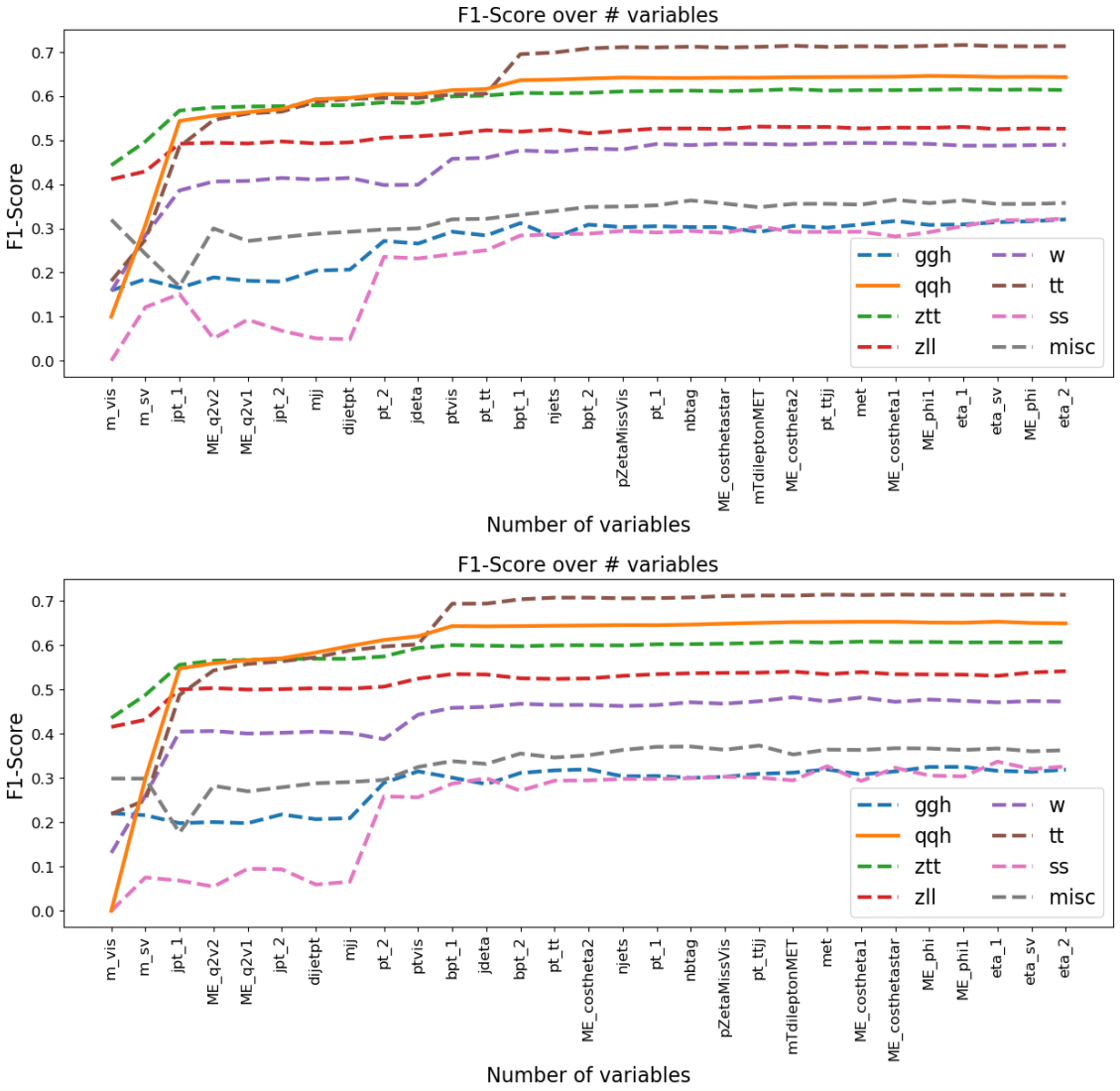


Figure C.12.: F1 score of the `qqh` output class (marked as the solid line) and the $e\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

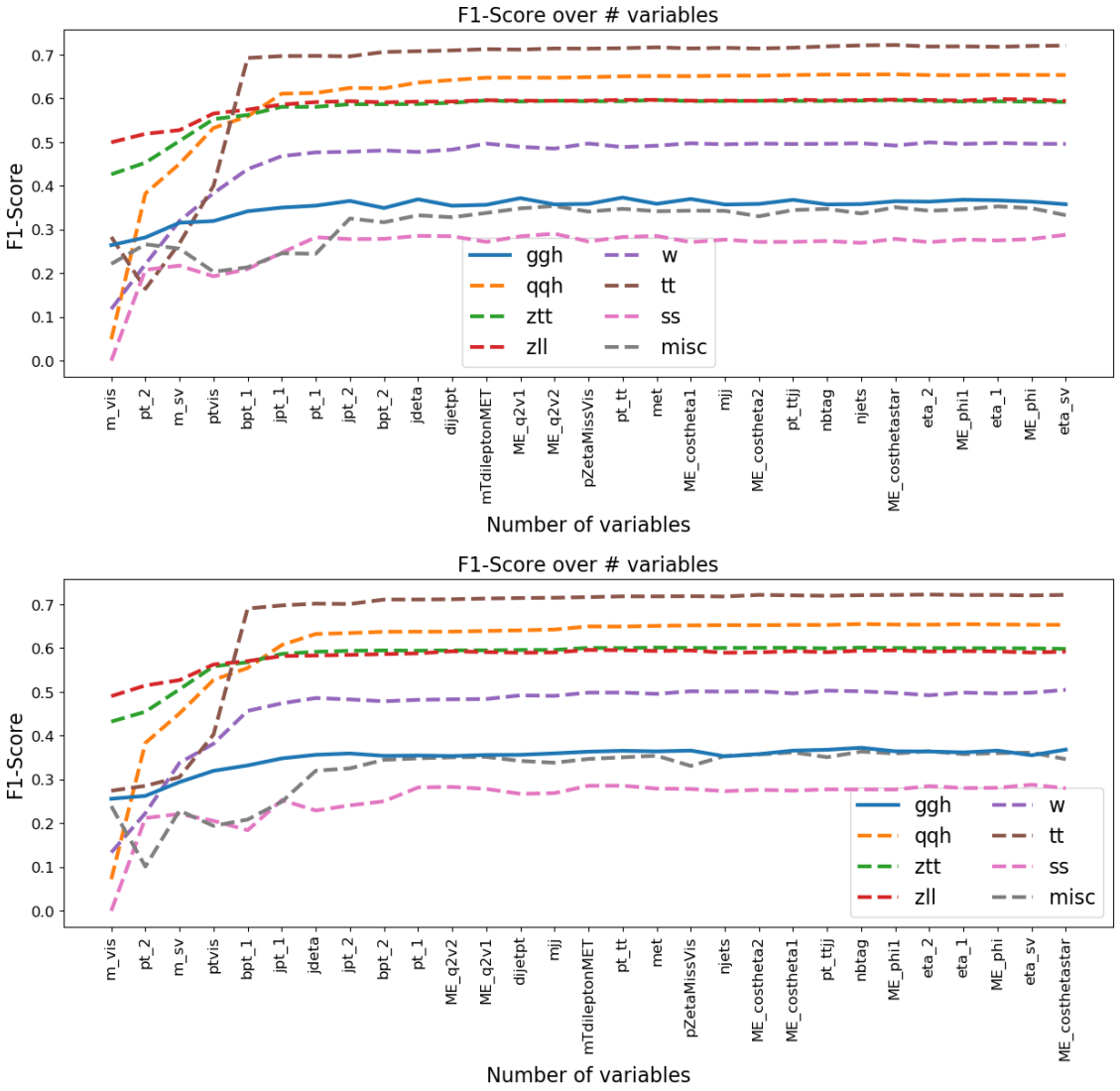


Figure C.13.: F1 score of the ggh output class (marked as the solid line) and the $\mu\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

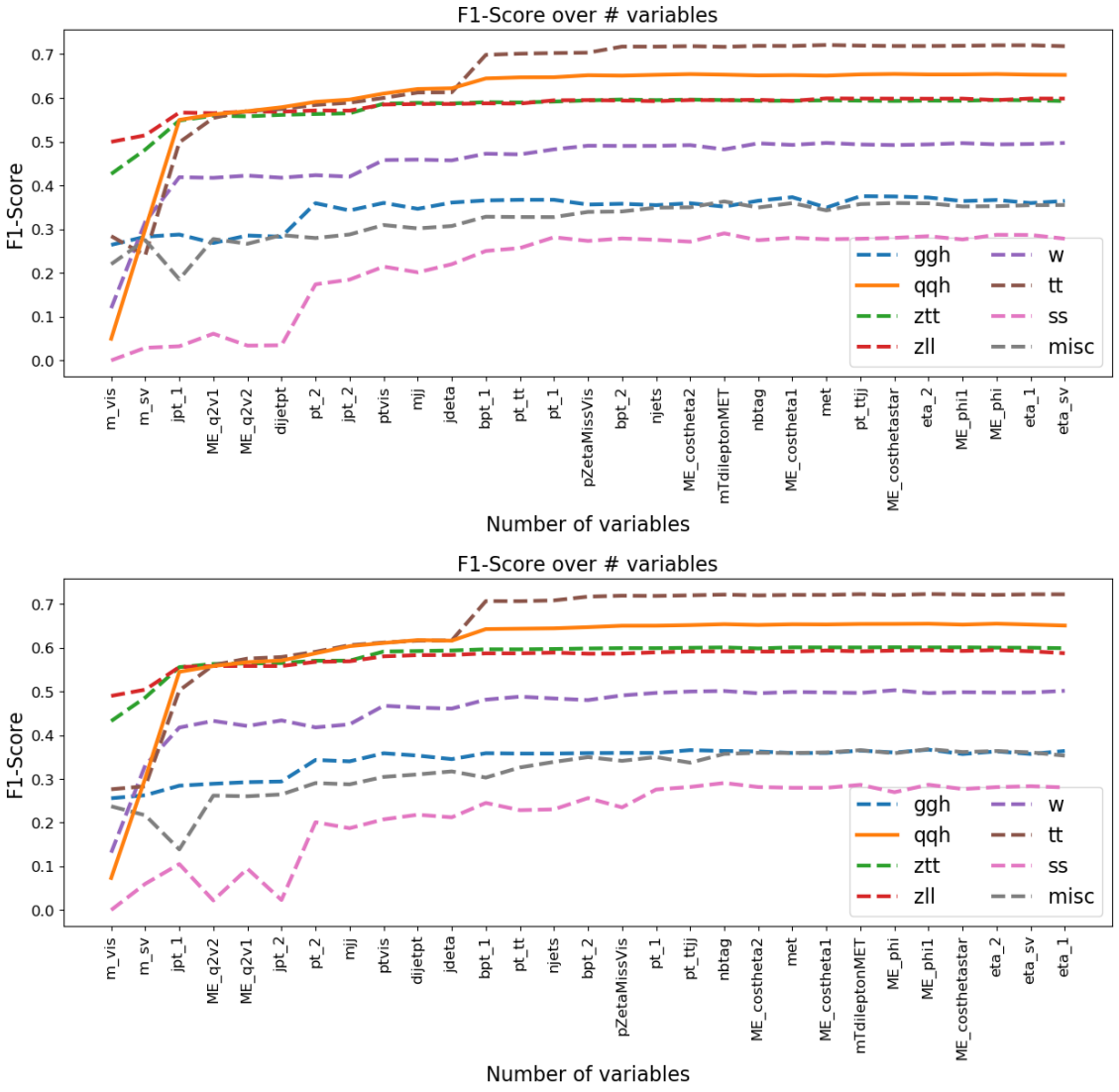


Figure C.14.: F1 score of the qqh output class (marked as the solid line) and the $\mu\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

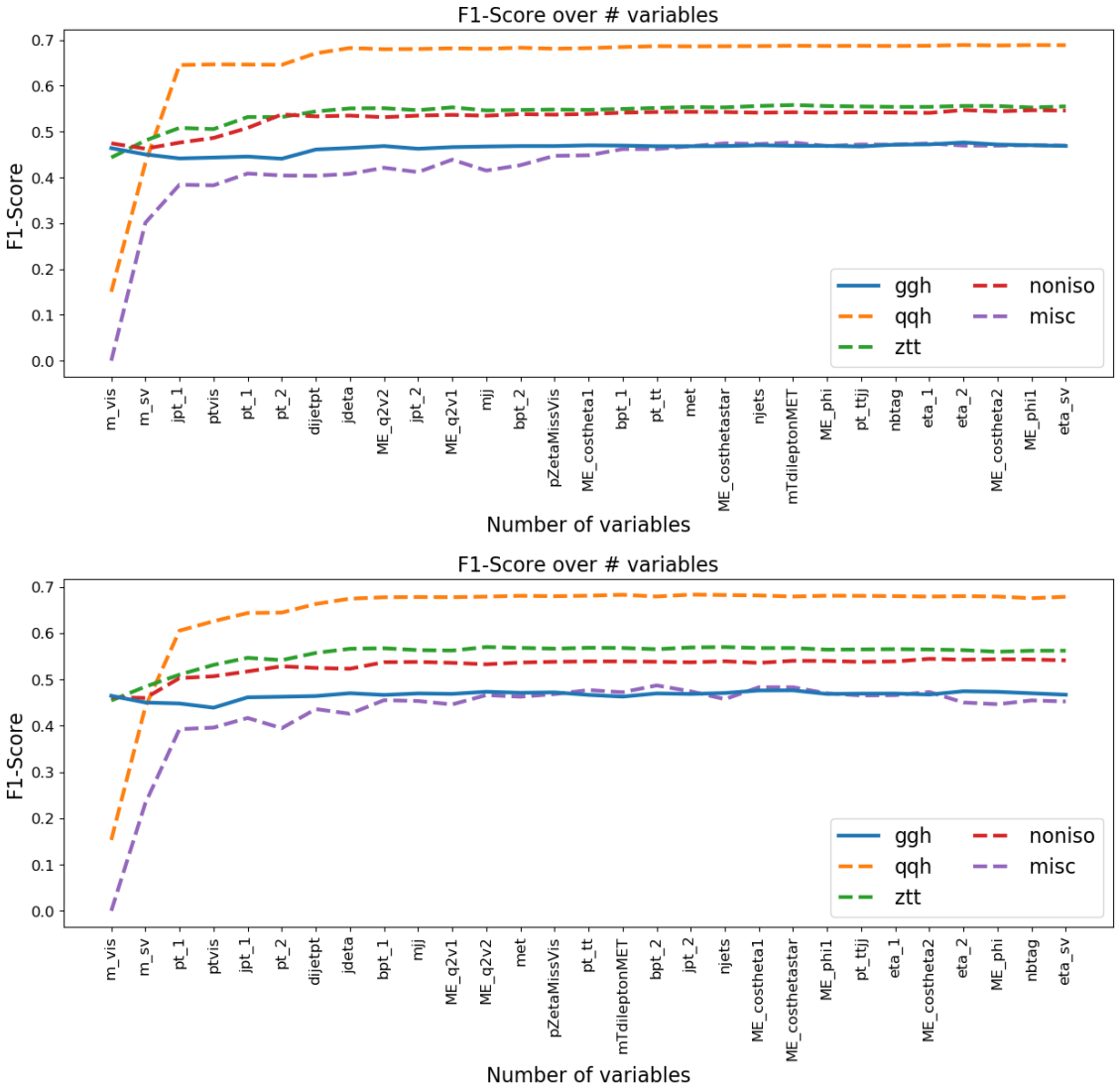


Figure C.15.: F1 score of the ggh output class (marked as the solid line) and the $\tau_h\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

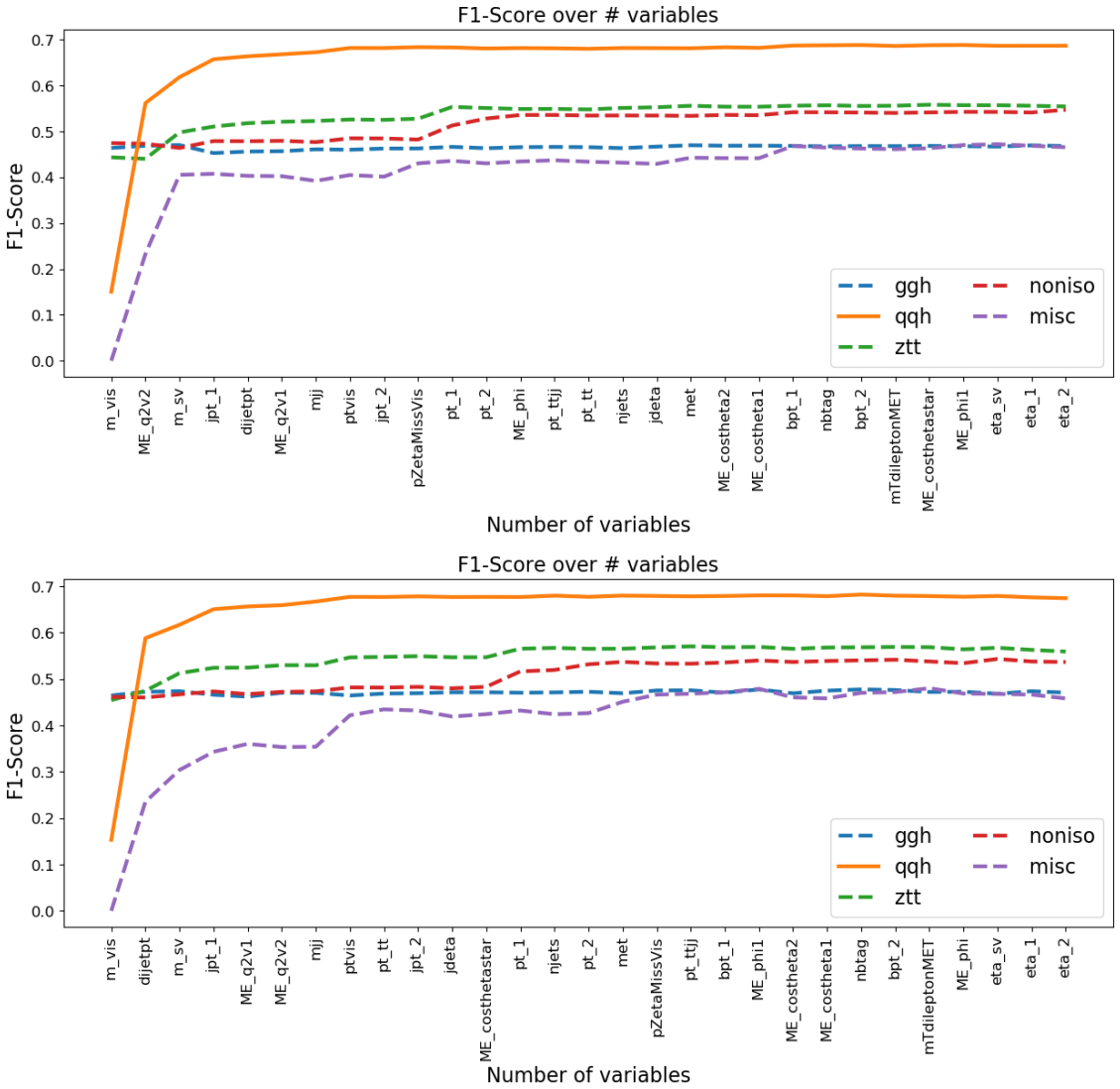


Figure C.16.: F1 score of the `qqh` output class (marked as the solid line) and the $\tau_h\tau_h$ final state as function of the input variables for two independent trainings and Taylor coefficient rankings.

Bibliography

[1] S. Wunsch et al. Identifying the relevant dependencies of the neural network response on characteristics of the input space. *Comput. Softw. Big Sci.*, 2(1):5, 2018.

[2] G. Louppe et al. Learning to Pivot with Adversarial Networks. 2016.

[3] S. Chatrchyan et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30 – 61, 2012.

[4] G. L. Bayatian et al. *CMS Physics: Technical Design Report Volume 1: Detector Performance and Software.* Technical Design Report CMS. CERN, Geneva, 2006. There is an error on cover due to a technical problem for some items.

[5] European Organization of Nuclear Research. `https://home.cern/`. Accessed: 2020-02-06.

[6] CMS Public Physics results. `http://cms-results.web.cern.ch/cms-results/public-results/publications/HIG/index.html`. Accessed: 2020-02-06.

[7] P. W. Higgs. Spontaneous symmetry breakdown without massless bosons. *Phys. Rev.*, 145:1156–1163, May 1966.

[8] F. Englert and R. Brout. Broken Symmetry and the Mass of Gauge Vector Mesons. *Phys. Rev. Lett.*, 13:321–323, 1964. [,157(1964)].

[9] P. W. Higgs. Broken symmetries, massless particles and gauge fields. *Phys. Lett.*, 12:132–133, 1964.

[10] G. S. Guralnik et al. Global Conservation Laws and Massless Particles. *Phys. Rev. Lett.*, 13:585–587, 1964. [,162(1964)].

[11] P. W. Higgs. Broken Symmetries and the Masses of Gauge Bosons. *Phys. Rev. Lett.*, 13:508–509, 1964. [,160(1964)].

[12] T. W. Kibble. Symmetry Breaking in Non-Abelian Gauge Theories. *Physical Review*, 155(5):1554–1561, March 1967.

[13] R. Wolf. *The Higgs Boson Discovery at the Large Hadron Collider.* Springer International Publishing, 2015.

[14] CERN Twiki. `https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWGCrossSectionsFigures#Higgs_production_cross_sections`. Accessed: 2020-02-06.

[15] O. S. Brüning et al. *LHC Design Report.* CERN Yellow Reports: Monographs. CERN, Geneva, 2004.

[16] CERN Twiki. `https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWGCrossSectionsFigures#Higgs_production_cross_sections`. Accessed: 2020-02-06.

[17] E. A. Mobs. The CERN accelerator complex. Complexe des accélérateurs du CERN. Oct 2016. General Photo.

[18] D. Barney. CMS Detector Slice. CMS Collection., Jan 2016.

[19] Triggering and Data Acquisition. `http://cms.web.cern.ch/news/triggering-and-data-acquisition`. Accessed: 2020-02-06.

[20] CERN TWiki. `https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWGFiducialAndSTXS`. Accessed: 2020-02-06.

[21] An embedding technique to determine genuine $\tau\tau$ backgrounds from CMS data. Technical Report CMS-PAS-TAU-18-001, CERN, Geneva, 2018.

[22] M. Flechl, M. Spanring, F. Spreitzer. Data-driven background estimation of fake-tau backgrounds in di-tau final states with 2016 and 2017 data. *CMS Note*, 2018/257, 2018.

[23] J. Andrejkovic, J. Bechtel, S. Brommer, M. Flechl, A. Gottmann, O. Hlushchenko, T. Lenz, M. Meyer, A. Raspereza, M. Spanring, F. Spreitzer, R. Wolf, S. Wozniewski, S. Wunsch, S. Joerger, M. Scham. Measurement of higgs(125) boson properties in decays to a pair of tau leptons with 2016 and 2017 data using machine-learning techniques. *CMS Note*, 2019/177, 2019.

[24] R. Barlow and C. Beeston. Fitting using finite monte carlo samples. *Computer Physics Communications*, 77(2):219 – 228, 1993.

[25] Observation of the SM scalar boson decaying to a pair of τ leptons with the CMS experiment at the LHC. Technical Report CMS-PAS-HIG-16-043, CERN, Geneva, 2017.

[26] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. arXiv:1412.6980.

[27] Y. Bengio and X. Glorot. Understanding the difficulty of training deep feed forward neural networks. *International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 01 2010.

[28] I. Goodfellow et al. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[29] L. Bianchini et al. Reconstruction of the higgs mass in events with higgs bosons decaying into a pair of τ leptons using matrix element techniques. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 862:54 – 84, 2017.

[30] S. Wunsch. *A Novel Strategy for the Standard Model $H \to \tau\tau$ Analysis with Emphasis on Minimizing Systematic Uncertainties in Presence of Modern Multi-Variate Methods*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2017.

[31] A. M. Sirunyan et al. Search for additional neutral MSSM Higgs bosons in the $\tau\tau$ final state in proton-proton collisions at $\sqrt{s} = 13$ TeV. *JHEP*, 09:007, 2018.

[32] G. Cowan. *Statistical Data Analysis*. Oxford science publications. Clarendon Press, 1998.

[33] J. Andrejkovic, J. Bechtel, S. Brommer, M. Flechl, A. Gottmann, O. Hlushchenko, T. Lenz, M. Meyer, A. Raspereza, M. Spanring, F. Spreitzer, R. Wolf, S. Wozniewski, S. Wunsch. Measurement of higgs(125) boson properties in decays to a pair of tau leptons with 2016 and 2017 data using machine-learning techniques. *CMS Note*, 2018/255, 2018.

[34] Robert D. Cousins. Generalization of the chisquare goodness-of-fit test for binned data using saturated models, with application to histograms. 2013. `http://www.physics.ucla.edu/~cousins/stats/cousins_saturated.pdf` (visited on 27/11/2019).

[35] D. Jang. *Search for MSSM Higgs decaying to τ pairs in $p\bar{p}$ collision at $\sqrt{s}$ = 1.96 TeV at CDF*. PhD thesis, Rutgers U., Piscataway, 2006.

[36] Y. Gao et al. Spin determination of single-produced resonances at hadron colliders. *Physical Review D*, 81(7), Apr 2010.

[37] Updated results on the new boson discovered in the search for the standard model Higgs boson in the ZZ to 4 leptons channel in pp collisions at sqrt(s) = 7 and 8 TeV. Technical Report CMS-PAS-HIG-12-041, CERN, Geneva, 2012.

[38] S. Bolognesi et al. Spin and parity of a single-produced resonance at the lhc. *Physical Review D*, 86(9), Nov 2012.

[39] K. Potdar et al. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9, 10 2017.

[40] The ATLAS collaboration. Performance of b-Jet Identification in the ATLAS Experiment. *JINST*, 11, 2016.

[41] The CMS collaboration. Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. *JINST*, 13(05):P05011, 2018.

[42] J. Blitzer et al. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 120–128, Sydney, Australia, July 2006. Association for Computational Linguistics.

[43] S. J. Pan et al. Domain adaptation via transfer component analysis. *Trans. Neur. Netw.*, 22(2):199–210, February 2011.

[44] G. Louppe et al. Learning to pivot with adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 982, 2017.

[45] I. J. Goodfellow et al. Generative Adversarial Networks, 2014.

[46] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[47] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[48] C. Adam-Bourdarios et al. The Higgs boson machine learning challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, volume 42 of *JMLR: Workshop and Conference Proceedings*, pp. 37, Montreal, Canada, December 2014.

[49] M. Aaboud et al. Cross-section measurements of the Higgs boson decaying into a pair of τ-leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector. *Phys. Rev.*, D99:072001, 2019.

[50] R. Barlow. *Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences, Edition 1*. John Wiley and Sons, New Jersey, NY, 2013.

[51] G. Bohm and G. Zech. *Introduction to Statistics and Data Analysis for Physicists*. Deutsches Elektronen-Synchrotron, Hamburg, DE, 2010. `http://www-library.desy.de/preparch/books/vstatmp_engl.pdf`.

[52] P. De Castro and T. Dorigo. INFERNO: Inference-Aware Neural Optimisation. *Comput. Phys. Commun.*, 244:170–179, 2019.