Università degli Studi di Napoli *Federico II*

DOTTORATO DI RICERCA IN FISICA

Ciclo: XXXIII

Coordinatore: Prof. Salvatore Capozziello

Inference Aware Neural Optimization for Top Pair Cross-Section Measurements with CMS Open Data

Settore Scientifico Disciplinare FIS/01

Dottorando	**Tutor**
Lukas LAYER	Prof. Dr. Tommaso Dorigo
	Dr. Alberto Orso Maria Iorio

Anni 2018/2022

Abstract

In recent years novel inference techniques have been developed based on the construction of summary statistics with neural networks by minimizing inference-motivated losses via automatic differentiation. The inference-aware summary statistics aim to be optimal with respect to the statistical inference goal of high energy physics analysis by accounting for the effects of nuisance parameters during the model training. One such technique is INFERNO (P. de Castro and T. Dorigo, Comp. Phys. Comm. 244 (2019) 170) which was shown on toy problems to outperform classical summary statistics for the problem of confidence interval estimation in the presence of nuisance parameters. In this thesis the algorithm is extended to common high energy physics problems based on a differentiable interpolation technique. In order to test and benchmark the algorithm in a real-world application, a complete, systematics-dominated analysis of the CMS experiment, "Measurement of the $\mathrm{t\bar{t}}$ production cross section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV" (CMS Collaboration, The European Physical Journal C, 2013) is reproduced with CMS Open Data. The application of the INFERNO-powered neural network architecture to this analysis demonstrates the potential to reduce the impact of systematic uncertainties in real LHC analysis.

Preface

The work presented in this thesis summarizes the main research project that I carried out as a Marie Skłodowska-Curie fellow in the context of an interdisciplinary European Innovative Training Network INSIGHTS, funded by the European Union's Horizon 2020 research and innovation program, call H2020-MSCA-ITN-2017, under Grant Agreement n. 765710. The main research project is the result of a collaboration with Tommaso Dorigo and Giles Strong. The purpose of this manuscript is to describe my original contributions to this work. The Chapters 2-5 are intended as an introduction to the main research project presented in Chapters 6 and 7 and do not describe my own work. The code for my main research project has been made public [1].

During my PhD as a Marie Skłodowska-Curie fellow of the INSIGHTS network, I also collaborated with members of other institutions and collaborations. The projects carried out within CMS and INSIGHTS that lead to a publication during my PhD are briefly described in the following.

Automatic log analysis with NLP for the CMS workflow handling [2]

In this work an approach is presented that considers the error log files of failing CMS Monte Carlo production workflows as regular text to leverage modern techniques from Natural Language Processing (NLP). In general, log files contain a substantial amount of text that is not human language. Therefore, different log parsing approaches are studied in order to map the log files' words to high dimensional vectors. These vectors are then exploited as feature space to train a model that predicts the action that an operator has to take. This approach has the advantage that the information of the log files is extracted automatically and the format of the logs can be arbitrary. The performance of the log file analysis with NLP is compared to previous approaches. I have contributed to this work by setting up a pipeline with Apache Spark to prepare the input data and I developed and optimized a state-of-the-art NLP model with TensorFlow.

Measurement of CKM matrix elements in single top quark t-channel production in proton-proton collisions at $\sqrt{s} = 13$ TeV [3]

In this work the first direct, model-independent measurement is presented of the modulus of the Cabibbo-Kobayashi-Maskawa (CKM) matrix elements $|V_{\mathrm{tb}}|$, $|V_{\mathrm{td}}|$, and $|V_{\mathrm{ts}}|$, in final states enriched in single top quark t-channel events. The analysis uses proton-proton collision data from the LHC, collected during 2016 by the CMS experiment, at a centre-of-mass energy of 13 TeV, corresponding to an integrated luminosity of 35.9fb^{-1}. In the standard model hypothesis of CKM unitarity, a lower limit of $|V_{\mathrm{tb}}| > 0.970$ is measured at the 95% confidence level. Several theories beyond the standard model are considered, and by releasing all constraints among the involved parameters, the values $|V_{\mathrm{tb}}| = 0.988 \pm 0.024$, and $|V_{\mathrm{td}}|^2 + |V_{\mathrm{ts}}|^2 = 0.06 \pm 0.06$, where the uncertainties include both statistical and systematic components, are measured. Within this work I have contributed to the setup of the profile likelihood fit with the CMS combine tool.

Clustering of Experimental Seismo-Acoustic Events Using Self-Organizing Map (SOM) [4]

In this work laboratory experiments that produce seismo-acoustic events relevant for understanding the degassing processes of a volcanic system are studied. A Self-Organizing Map algorithm is applied to cluster the feature vectors extracted from the seismo-acoustic data through the parameterization phase, and four main clusters have been identified. The results were consistent with the experimental findings on the role of viscosity, flux velocity and conduit roughness on the degassing regime. The neural network is capable to separate events generated under different experimental conditions. This suggests that the SOM is appropriate for clustering natural events such as the seismo-acoustic transients accompanying Strombolian explosions and that the adopted parameterization strategy may be suitable to extract the significant features of the seismo-acoustic signals linked to the physical conditions of the volcanic system. Within an industrial secondment at the Italian National Institute of Geography and Volcanology, I contributed to this work with data preparation, visualization and the application of unsupervised learning algorithms.

Changes in the Eruptive Style of Stromboli Volcano before the 2019 Paroxysmal Phase Discovered through SOM Clustering of Seismo-Acoustic Features Compared with Camera Images and GBInSAR Data [5]

In this work the two paroxysmal explosions that occurred at Stromboli on 3 July and 28 August 2019 are studied. After the first paroxysm an effusive activity began from the summit vents and affected the NW flank of the island for the entire period between the two paroxysms. We carried out an unsupervised analysis of seismic and infrasonic data of Strombolian explosions over 10 months (15 November 2018–15 September 2019) using a Self-Organizing Map (SOM) neural network to recognize changes in the eruptive patterns of Stromboli that preceded the paroxysms. We used a dataset of 14,289 events. The SOM analysis identified three main clusters that showed different occurrences with time indicating a clear change in Stromboli's eruptive style before the paroxysm of 3 July 2019. We compared the main clusters with the recordings of the fixed monitoring cameras and with the Ground-Based Interferometric Synthetic Aperture Radar measurements, and found that the clusters are associated with different types of Strombolian explosions and different deformation patterns of the summit area. Our findings provide new insights into Strombolian eruptive mechanisms and new perspectives to improve the monitoring of Stromboli and other open conduit volcanoes. I contributed to this work with data preparation, visualization and the application of unsupervised learning algorithms.

Toward Machine Learning Optimization of Experimental Design [6, 7]

In this work the research program of the MODE Collaboration (an acronym for Machine-learning Optimized Design of Experiments), which aims at developing tools based on deep learning techniques to achieve end-to-end optimization of the design of instruments via a fully differentiable pipeline capable of exploring the Pareto-optimal frontier of the utility function is described. The goal of MODE is to demonstrate those techniques on small-scale applications such as muon tomography or hadron therapy, to then gradually adapt them to the more ambitious task of exploring innovative solutions to the design of detectors for future particle collider experiments. I have been a founding member of the collaboration and the development of INFERNO is part of the program.

Calorimetric Measurement of Multi-TeV Muons via Deep Regression [8]

In this work the feasibility of an entirely new avenue for the measurement of the energy of muons based on their radiative losses in a dense, finely segmented calorimeter is presented. This is made possible by exploiting spatial information of the clusters of energy from radiated photons in a regression task. The use of a task-specific deep learning architecture based on convolutional layers allows us to treat the problem as one akin to image reconstruction, where images are constituted by the pattern of energy released in successive layers of the calorimeter. A measurement of muon energy with better than 20% relative resolution is shown to be achievable for ultra-TeV muons. I have contributed to this work with data preparation, visualization and assisted in the training of the models.

Acknowledgements

First and foremost, I want to thank my supervisor Tommaso Dorigo for his help and guidance during my PhD. The last years of this project were overshadowed by the Covid pandemic and various other challenges, and without his commitment and incredible enthusiasm for physics this project would not have been possible.

I am also very grateful to Jean-Roch Vlimant for his continuous support during and after my CMS service task, and for teaching me a lot about machine learning and data science.

I want to thank Giles Strong who collaborated with me, Tommaso and Benedetta Dal Sasso on the INFERNO project. I profited a lot from his excellent coding practices, as well as his knowledge about machine learning.

I also want to thank the members of the CMS groups at INFN Napoli and Padova, that were very welcoming and helped me to get started in Italy, as well as providing me valuable feedback for my research.

I am grateful to the INSIGHTS network, particularly to Glen Cowan, for the effort in organising the various training and research activities. INSIGHTS also enabled me to stay several periods at CERN, which is a unique place and was crucial for my research project. I would like to thank the other ESRs, in particular Rahul, Pim, Sitong and Artem, with whom I shared the time at CERN and in Napoli.

I am also grateful to P. De Castro for initial discussions on my research project and the members of gradHEP and MODE for helpful discussions on differentiable programming and inference, in particular J. Kieseler, A. Held, and N. Simpson.

I also want to thank the members of the INGV Napoli for hosting me during my secondment and the great scientific collaboration that developed from it.

I further would like to thank Olaf Behnke and Jean-Roch Vlimant for reviewing this thesis and for providing great comments that significantly improved the quality of the writing.

Finally, I want to thank my family and my friends for the support during the last years, particularly during the pandemic. Also without you this project would not have been possible.

Contents

1 Introduction

The search for new physics at the LHC, as well as the search for known physics signals not yet put in evidence, is in general terms a problem of distinguishing a small signal from large backgrounds in a multi-dimensional space of observed event features. Neural network classifiers as well as boosted decision trees are nowadays routinely used to construct powerful summary statistics as input for inference, e.g. for parameter estimation or hypothesis tests. However, the imperfect knowledge of the properties of the background and signal results in systematic uncertainties that have to be accounted for in statistical models by the inclusion of nuisance parameters [9]. Neglecting the nuisance parameters during the training of a classifier causes a reduction of the statistical power of the summary statistic during inference. Nuisance parameters are therefore one of the main limiting factors of the precision of high energy physics (HEP) analyses, while the statistical uncertainty can be reduced by collecting more data. The understanding and mitigation of systematic uncertainties is crucial for precise measurements at the LHC, in particular for model-independent searches that are becoming increasingly important in the search for new physics. Thus, addressing directly the statistical inference objective with modern machine learning techniques that account for the effect of nuisance parameters has attracted a lot of attention and various methods are being developed [10].

In recent years, a novel approach, called INFERNO [11], an acronym that stands for Inference-Aware Neural Optimization, has been developed to construct machine learning based summary statistics that are optimal for statistical inference. The INFERNO technique promises to fundamentally improve the power of neural network classification by embedding the effect of all systematic uncertainties on the parameter of interest in the loss of the neural network. On toy problems INFERNO clearly outperformed classical summary statistics for the problem of confidence interval estimation in the presence of nuisance parameters. The original code for the INFERNO algorithm has been developed for a synthetic example. However, the structure of the data and systematic uncertainties in HEP is special and therefore one of the main aspects of this work is the development of a framework where real LHC data can be used with the INFERNO algorithm. In order to benchmark the framework, a complete, systematics-dominated analysis of the CMS experiment, “Measurement of the $\mathrm{t\bar{t}}$ production cross section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV” [12] is reproduced with CMS Open Data. The study of top-quark production, decays, couplings and other properties with the highest possible precision is an important part of the physics program of the CMS experiment and is one of the most promising avenues for the discovery of new physics due to the critical role of the top-quark mass in the consistency of the Standard Model. The INFERNO algorithm will be applied to the reproduced top pair cross-section measurement and the inference with the obtained summary statistic will be compared to a classifier trained in a classical approach with a standard binary cross-entropy loss. Several studies with different setups are performed in order to understand under which conditions INFERNO can reduce the impact of systematic uncertainties in a realistic LHC analysis.

The work in this thesis is organized as follows: in Chapter 2 the foundations of the machine learning and statistical methods are laid that are necessary to understand the working principle of the INFERNO algorithm. In Chapter 3 the use of machine learning to build summary statistics for inference is reviewed from a statistical perspective and its limitation in the presence of nuisance parameters is discussed. Additionally, an overview over existing approaches to deal with nuisance parameters in machine learning is given. Subsequently the INFERNO algorithm is described and its performance on a synthetic example is discussed. In Chapter 4 the Standard Model of particle physics and its limitations are discussed, as well as possible extensions. Then an introduction to the phenomenology of proton-proton collisions is given and some of the key aspects of top physics at the LHC are reviewed. In Chapter 5, the design of the Large Hadron Collider at CERN and the CMS detector is discussed, as well as the event reconstruction and simulation of the CMS experiment. In Chapter 6, the reproduction of the analysis “Measurement of the $\mathrm{t\bar{t}}$ production cross section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV” with CMS Open Data is presented. The main result in this chapter is the measurement of the top pair cross-section based on a summary statistic obtained by training a neural network classifier with a binary cross-entropy loss with CMS Open Data. In Chapter 7, the extension of the INFERNO algorithm to HEP-like data and systematic uncertainties is described. Its performance is quantified in a performance study based on the reproduced analysis and compared to models trained with binary cross-entropy. Finally, an INFERNO model is trained with the most relevant systematic uncertainties. The resulting summary statistic is used to measure the top pair cross-section and compare the results to the classical approach described in Chapter 6.

2 Machine Learning and Statistical Inference at the LHC

Machine learning (ML) is a very broad field that has grown substantially in the last decade, driven mainly by the emergence of deep learning (DL). Deep learning has seen an enormous growth in its popularity and usefulness, mainly due to more powerful hardware, larger datasets and advanced techniques to train deeper networks. Particle physics offers a variety of use cases for machine learning techniques. The experimental high-energy physics (HEP) program has two main objectives: probing the Standard Model (SM) with increasing precision and searching for new particles associated with physics beyond the SM. Both tasks require the identification of rare signals in very large backgrounds. The increasing number of collisions at the High-Luminosity LHC will make this a significant challenge. Exploiting the full potential of machine learning in many different areas will be a key ingredient for the success of the LHC program. In the area of statistical data analysis, machine learning techniques have become particularly important to construct powerful summary statistics that are used as input for inference. Due to the importance of machine learning and inference for the topic of this thesis, the core concepts will be introduced in the following. The review is based on the standard literature for machine learning [13] and the Statistics and Machine Learning chapters of the Particle Data Group Review [9], which are also recommended for further reading.

2.1 Core Concepts of Machine Learning

Most machine learning algorithms can be described as the optimization of some objective, which can be formulated as minimizing a quantity called risk, that includes three main ingredients: the model family $\mathcal{F}$, the loss function $\mathcal{L}$, and a data distribution $p(u)$. The risk for a model $f \in \mathcal{F}$ is defined as its expected loss:

$$\mathcal{R}[f] := \mathbb{E}_{p(u)}[\mathcal{L}(u, f(u))] \equiv \int \mathcal{L}(u, f(u)), p(u)\mathrm{d}u \ . \tag{2.1}$$

In general, the goal of machine learning is to solve the optimization problem:

$$f^* = \arg\min_{f\in\mathcal{F}} \mathcal{R}[f] \tag{2.2}$$

where $\mathcal{F}$ includes all possible functions. In practice, the data distribution $p(u)$ is unknown, and the input for the machine learning algorithm are n i.i.d. samples from that distribution. This leads to the corresponding empirical risk

$$\mathcal{R}_{\text{emp}}[f] := \mathbb{E}_{\hat{p}(u)}[\mathcal{L}(u, f(u))] \equiv \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(u_i, f(u_i)) \tag{2.3}$$

where $\hat{p}(u) = \frac{1}{n}\sum_{i=1}^{n} \delta(u - u_i)$ is referred to as the empirical distribution of the samples u_i. The empirical risk minimization approximates f^* with the empirical analogue

$$\hat{f} = \arg\min_{f\in\hat{\mathcal{F}}} \mathcal{R}_{\text{emp}}[f] \tag{2.4}$$

where $\hat{\mathcal{F}}$ are all possible functions with the model parameters ϕ. In the infinite parameter limit, machine learning models are universal approximators, such that they cover all functions and $\mathcal{F} = \hat{\mathcal{F}}$. However, this is not necessarily true for models with finite parameters in real world applications. The “capacity” of a model characterizes the ability to fit a wide variety of functions and depends on the model architecture and its parameters such as the width and depth of neural network layers. To evaluate the abilities of a machine learning algorithm, a quantitative measure of its performance needs to be designed, which is usually specific to the learning task.

2.1.1 Generalization and Bias-Variance Tradeoff

The central challenge in machine learning is that the algorithm must perform well on previously unseen samples - not just on those which have been used for the training procedure. The ability to perform well on previously unobserved inputs is called “generalization”. More formally, while the empirical risk might be minimized $\mathcal{R}_{\text{emp}}[\hat{f}] \to 0$, the true risk might be large $\mathcal{R}[\hat{f}] \gg 0$. The gap $\mathcal{R}[\hat{f}] - \mathcal{R}_{\text{emp}}[\hat{f}]$ is typically referred to as the “excess risk”. In general it is not possible to evaluate $\mathcal{R}[\hat{f}]$

exactly because the true data distribution $p(u)$ is unknown, however, an independent testing dataset can be used to obtain an unbiased estimate of it. This motivates the splitting of the data in “train” and “test” sets. The factors determining how well a machine learning algorithm will perform are its ability to reduce the training error, while also keeping the gap between test and training error small. These two factors correspond to two important concepts in machine learning: underfitting and overfitting. Underfitting occurs when the model is too simple and is not able to perform well on the training set. Overfitting occurs when a sufficiently flexible model fits the training data very well, but the gap between the training error and test error is large and the model does not generalize well to unseen data. By altering a model’s capacity, one may control under and overfitting. Models with low capacity might not be sufficiently flexible to fit the training set. Models with high capacity can overfit by memorizing statistical fluctuations of the training set.

The statistical concepts of “bias” and “variance” are useful to formally characterize the concept of generalization. Bias and variance quantify two different sources of error in an estimator, where an estimator $\hat{\theta}$ is defined as a function of the data used to estimate the value of the parameter θ. The variance $V[\widehat{\theta}]$ provides a measure of the

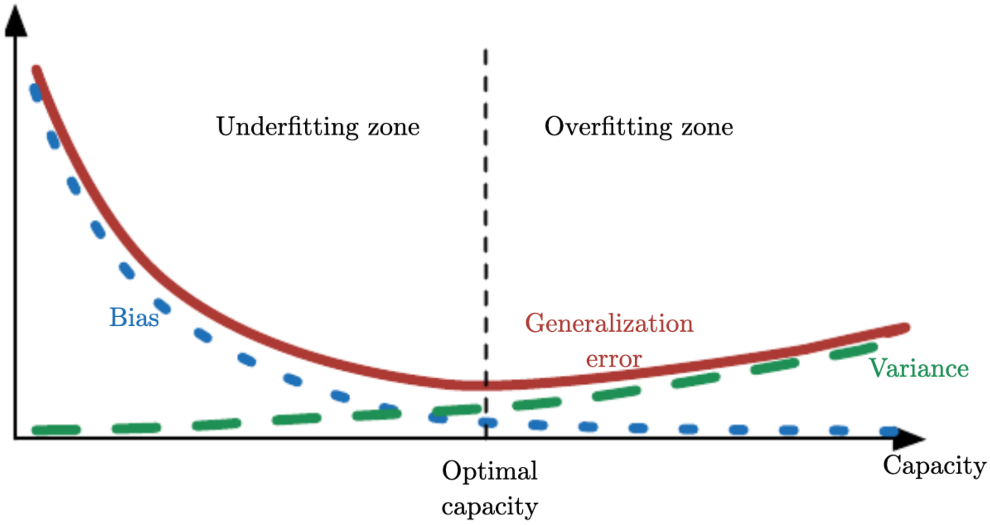


Figure 2.1: As capacity increases, bias tends to decrease and variance tends to increase, yielding an U-shaped curve for the generalization error [13].

deviation from the expected estimator value that any sampling of the data causes, while bias measures the expected deviation from the true value of the parameter:

$$\mathrm{b} = \mathbb{E}\left(\hat{\theta}\right) - \theta \ . \tag{2.5}$$

Ideally, an estimator has both a low bias and a relatively low variance. The mean-squared error (MSE) is a measure of an estimator's quality which combines bias and variance:

$$\mathrm{MSE} = \mathbb{E}\left[(\widehat{\theta} - \theta)^2\right] = V[\widehat{\theta}] + b^2 \ . \tag{2.6}$$

Desirable estimators have a small MSE, which means that they manage to balance both their bias and variance. The relationship between bias and variance is linked to the discussed concepts of capacity, underfitting and overfitting. When the generalization error is measured by the MSE, increasing capacity tends to increase variance and decrease bias. Typically, the generalization error has a U-shaped curve as a function of the model capacity, which is illustrated in Fig. 2.1. The optimal capacity corresponds to the minimum of the generalization error. Left to the optimal capacity, the bias dominates and the model tends to underfit, while right to the optimal capacity the variance dominates and the model tends to overfit, and the gap between training and generalization error increases. It is also possible for the model to have optimal capacity and yet still have a large gap between training and generalization errors, which can be reduced with more training examples. A common way to control the bias-variance trade-off is to use cross-validation. In the basic k-fold cross-validation approach, the training set is split into k smaller sets and k models are trained on $k-1$ folds. The models are then evaluated on the remaining part of the data and the performance is measured by the average of the computed values such that an estimate of the mean value and the variance of the predictions can be obtained. This allows to choose a suitable model with a low generalization error.

Surprisingly, recent studies have shown that overparametrized models can achieve good empirical generalization. Overparametrized models are highly complex with respect to the size of the training dataset, which results in them interpolating the training data. A wide range of interpolating models have been observed to generalize extremely well on unseen test data [14]. The double descent phenomenon shown in Fig. 2.2, indicates that highly overparametrized models can improve over the best underparametrized model in test performance. This counter-intuitive behaviour is

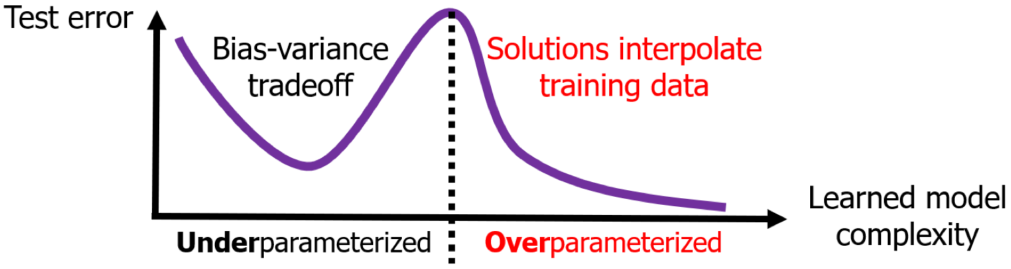


Figure 2.2: Double descent of test errors with respect to the complexity of the learned model. In this qualitative demonstration, the global minimum of the test error is achieved by maximal overparametrization [14].

studied by the theory of overparametrized ML, called TOPML, that tries to explain these findings from a statistical signal processing perspective.

2.1.2 No-Free Lunch Theorem and Regularization

The "no-free lunch" theorem for machine learning states that, averaged over all possible data-generating distributions, every classification algorithm has a similar error rate when classifying unseen data, thus no learning algorithm is universally better than any other [15]. However, by making assumptions about the kinds of probability distributions, learning algorithms can be designed to perform well on these particular distributions. Thus, the goal of machine learning is not to find a universal learning algorithm, but rather to understand what kinds of distributions are relevant and what kinds of machine learning algorithms perform well on data sampled from particular data-generating distributions.

The no-free lunch theorem and the bias–variance tradeoff motivates the addition of a regularization term to the loss function. A common form of regularization is Tikhonov regularization, which penalizes the loss of a model with parameters ϕ by the L2 norm of the parameters $||\phi||^2$. Another possible form of regularization is the restriction of the model class $\hat{\mathcal{F}}$, since in real-world problems some models perform better than others, depending on the concrete application. This form of regularization is typically encoded in the architecture of a neural network and the choices are referred to as inductive bias in the model. In addition to explicit regularization, it is also possible to regularize implicitly, for example with early stopping, where the loss is monitored on the training and test dataset. Early stopping terminates the training when the test loss does not improve anymore for a number of pre-defined epochs. The no-free lunch theorem makes clear that no optimal machine learning algorithm and no best form of regularization exists, instead a form of regularization has to be chosen that suits the particular task.

2.1.3 Supervised Learning

Supervised learning refers to the class of problems where the training dataset consists of input-output pairs $\{x_i, y_i\}_{i=1,\dots,n}$, where $x_i \in \mathcal{X}$ are the input features and $y_i \in \mathcal{Y}$ are the corresponding target labels. The resulting trained model is then used to predict the labels for a dataset where labels are not available, which can be seen as an estimate of the conditional probability $p(y|x)$, to predict y given x. Furthermore, it is typically assumed that the input features (x_i, y_i) are i.i.d. and are generated according to the distribution $p(x, y)$, that usually is unknown.

Regression

The goal of regression is the prediction of a label $y \in \mathcal{Y}$ given an input feature vector $x \in \mathcal{X}$. To solve this task, the learning algorithm is asked to output a function $f : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}$. When $\mathcal{Y}$ is a discrete variable, the task is referred to as classification. Regression and classification are closely related, since many classifiers predict continuous probabilities for each class, followed by an operation that results in a discrete label, e.g. logistic regression. A common loss function for regression is the mean-squared error (MSE):

$$\mathcal{L}_{\mathrm{MSE}}(y, f(x)) = (y - f(x))^2 \ . \tag{2.7}$$

It can be shown with calculus of variations that the optimal regressor for the MSE loss is the conditional expectation of y given x:

$$f^*_{\mathrm{MSE}}(x) = \mathbb{E}_{p(y|x)}[y] \ . \tag{2.8}$$

Here regression yields a function $f(x)$ that provides a point estimate for y, however, also alternative approaches to regression exist that try to model the full conditional distribution $p(y|x)$. A drawback of the MSE as a loss function is its sensitivity to outliers. Therefore also various other loss functions exist that can be used in regression problems, such as the Huber loss [16], that aims to be more robust to outliers than MSE.

Classification

The goal of classification is the prediction of a discrete number of class labels $y \in \mathcal{Y}$ given an input feature vector $x \in \mathcal{X}$. Typically, the learning algorithm produces a function $f : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{Y}|}$. If the number of discrete labels is two $y = \{0, 1\}$, the task is referred to as binary classification and the mean-squared error $\mathcal{L}_{\text{MSE}}(y, f(x))$ can be used as loss function. The resulting model approximates the optimal classifier f^*_{MSE} and the conditional expectation of equation 2.8 can be written as:

$$f^*_{\text{MSE}}(x) = \mathbb{E}_{p(y|x)}[y] \to p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 0)p(y = 0) + p(x|y = 1)p(y = 1)} \ . \tag{2.9}$$

This expression shows that the MSE loss for binary classification results in the Bayesian posterior probability that the label y is equal to class 1 given the feature vector x. As will be discussed further in 3.1.1, binary classification is equivalent to hypothesis testing in frequentist statistics. If the classification task considers multiple classes, the binary cross-entropy loss can be generalized to the cross-entropy loss:

$$\mathcal{L}_{\text{xe}}(y, f(x)) = -\sum_{c=1}^{|\mathcal{Y}|} \mathbf{1}(y = c) \log(f_c(x)) \tag{2.10}$$

where $f_c : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ and the indicator function selects the term in the sum for the corresponding class label y. This loss can be derived from maximizing the posterior of the Bayes theorem using a discrete set of class labels y, and enforcing the constraint $\sum_c f_c(x) = 1$ and $f_c(x) \geq 0$. The risk associated with the cross-entropy loss function is given by:

$$\mathcal{R}_{\text{xe}}[f] = \mathbb{E}_{p(x,y)}\left[-\sum_{c=1}^{|\mathcal{Y}|} \mathbf{1}(y = c) \log f_c(x)\right] \tag{2.11}$$

and calculus of variations can be used to show that the optimal classifier is the conditional expectation for y given x:

$$f^*_{\text{x.e.},c}(x) = p(y = c|x) \ . \tag{2.12}$$

In general, the cross-entropy between two distributions can be written as:

$$H[p, f] = \mathbb{E}_p[-\log f] \ . \tag{2.13}$$

This is related to the Kullback-Leibler (KL) divergence that measures the degree of dissimilarity between two probability distributions:

$$KL(p\|f_\phi) := \mathbb{E}_p[\log p(x) - \log f_\phi] = H[p, f_\phi] + H[p] \tag{2.14}$$

where $H[p] := \int p(x) \log p(x) dx$ is the entropy and independent of f_ϕ. The KL divergence is equal to zero if and only if $p = f$. Minimizing the cross-entropy $H[p, f_\phi]$ with respect to ϕ is equivalent to minimizing the KL divergence, since $H[p]$ does not depend on ϕ. It can further be shown that minimizing the empirical risk of the cross-entropy loss function:

$$\mathcal{R}_{\text{emp,xe}}[f_\phi] = -\frac{1}{n}\sum_{i=1}^{n} \log f_\phi(x) \tag{2.15}$$

is equivalent to maximum likelihood estimation for the parameters ϕ of the classifier f_ϕ with likelihood function $L(\phi) = \prod_{i=1}^{n} f_\phi(x_i)$. Thus, minimizing the cross-entropy loss yields the optimal parameters ϕ for the model f_ϕ. In general, any loss consisting of a negative log-likelihood is a cross-entropy between the empirical distribution defined by the training set and the probability distribution defined by the model. Therefore, maximum likelihood estimation in the context of classification can be seen as an attempt to minimize the dissimilarity between the empirical distribution defined by the training set and the model distribution, with the degree of dissimilarity measured by the KL divergence.

2.1.4 Unsupervised Learning

Unsupervised learning addresses the class of problems that use unlabeled training datasets $\{x_i\}_{i=1,\ldots,n}$, where $x_i \in \mathcal{X}$ are the input features. Typically, it is assumed that the input features (x_i) are i.i.d. and are generated according to the distribution $p(x)$, that usually is not known. A concept related to unsupervised learning is that of self-supervised learning, which aims to learn useful features without requiring supervision labels for every sample in the input data. Common tasks for unsupervised learning are clustering and representation learning.

The goal of clustering is to group the data x into k clusters. Some clustering algorithms require the specification of k. Often, clustering uses a distance measure $d(x_i, x_j)$, which for example can be the L_p norm $\|x_i - x_j\|_p$. A popular clustering algorithm is the k-means algorithms, where the number of clusters k is specified beforehand which results in sets $S = \{S_1, \ldots, S_k\}$ that minimize the variance of each cluster.

Another important topic in machine learning and statistics is data representation and in particular how to construct a low-dimensional summary statistic that contains the relevant information from high-dimensional data for a particular task. An example of a linear dimensionality reduction is principal component analysis (PCA) [17]. A common algorithm for representation learning and nonlinear dimensionality reduction is the auto-encoder:

$$f = g \circ e : \mathcal{X} \to \mathcal{X} \tag{2.16}$$

where $e : \mathcal{X} \to \mathcal{Z}$ is called the encoder and $g : \mathcal{Z} \to \mathcal{X}$ is called the decoder. Typically the dimensionality of $\mathcal{Z}$ is smaller than the one of $\mathcal{X}$, and $z = e(x)$ yields a compressed representation of the input, also called a bottleneck. A possible loss function for an auto-encoder is the reconstruction error:

$$\mathcal{L}_{\text{a.e.}}\,(x, f(x)) = \|x - f(x)\|^2 \,. \tag{2.17}$$

After the training phase, the encoder part $e(x)$ can be used independently of the decoder to obtain low-dimensional representations of the data.

2.1.5 Stochastic Gradient Descent

Most machine learning algorithms are powered by the important Stochastic Gradient Descent algorithm (SGD), which is an extension of the Gradient Descent algorithm. The parameters θ of a parametrized model $f(x, \theta)$ and a loss function $\mathcal{L}(x, \theta)$ can be optimized with the Gradient Descent algorithm by performing iterative updates:

$$\theta_t = \theta_{t-1} - \lambda \nabla_\theta \mathcal{L}(x, \theta) \tag{2.18}$$

where λ is a small, real-valued hyperparameter called learning rate. The algorithm further requires an appropriate initialization of the parameters $\theta_{t=0}$. By defining $\delta\theta \equiv \theta_t - \theta_{t-1}$ and considering a small variation of the loss function $\delta(\mathcal{L}(x, \theta))$ the following relation can be obtained for small λ:

$$\delta(\mathcal{L}(x, \theta)) \approx \delta\theta \cdot \nabla_\theta \mathcal{L}(x, \theta) = -\lambda |\nabla_\theta \mathcal{L}(x, \theta)|^2 \tag{2.19}$$

which shows that the loss function decreases monotonically, and the parameter values are moved in the direction of loss function minimization.

Typically in machine learning large training sets are necessary for good generalization, which has the drawback that the training process becomes computationally expensive. The idea of Stochastic Gradient Descent is that the gradient is an expectation, which may be approximately estimated using a small set of samples. Thus, SGD is based on GD but replaces the exact gradient term $\nabla_\theta \mathcal{L}(x, \theta)$ with a stochastic approximation, where the loss function is evaluated using N i.i.d. sub-samples of the total dataset, also called mini-batches:

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)} \mathcal{L} \approx \frac{1}{N} \sum_i^N \nabla_\theta \mathcal{L}_i \tag{2.20}$$

and $\mathcal{L}_i$ is the loss function for data sample i. An important advantage of SGD is the computational cost, since only a small subset of data has to be evaluated at each update, and thus the cost per SGD update does not depend on the training set size. Moreover, vectorization libraries and GPU architectures can be exploited.

Based on the SGD algorithm, more advanced optimization algorithms can be constructed. The loss is often highly sensitive to some directions in parameter space and insensitive to others. The method of momentum can mitigate this by computing a running average of current and past gradients with a forgetting factor that controls how far back the averaging goes. Moreover, the convergence of SGD can be improved by making the learning rate λ depend on the individual θ_i, for example by using the gradient norm squared $(\nabla_{\theta_i} \mathcal{L})^2$. The popular ADAM (Adaptive Moment Estimation) algorithm makes use of both the momentum and gradient norm concepts by computing running averages of both the gradient and the gradient norm squared, each with a separate forgetting factor [18].

2.1.6 Boosted Decision Trees

Many different algorithms exist for supervised learning. Some of the most important are logistic regression, the k-NN algorithm, support vector machines, random forests, boosted decision trees, and neural networks. Due to its importance in HEP, the boosted decision tree algorithm will be briefly described, while neural networks are discussed in the next section.

Boosted decision trees (BDT) are popular algorithms for supervised learning that work particularly well with tabular data and are used in many HEP applications. A decision tree takes a set of input features and splits the input data recursively based on those features. Each split at a node is chosen such that it maximizes information gain or minimizes entropy. The information gain is the difference in entropy before and after the potential split, where the entropy is maximized for a 50/50 split and minimized for a 1/0 split. The splits in the decision tree are created recursively until a stop criterion is met, for example, the depth of the tree or no more information gain. Boosting is a method of combining many weak learners, such as single decision trees, into one classifier. Usually, each tree is created iteratively and the tree's output $h(x)$ is given a weight w relative to its accuracy. The ensemble output then corresponds to the weighted sum:

$$\hat{y}(x) = \sum_t w_t h_t(x) \ . \tag{2.21}$$

After an iteration each data sample is given a weight based on its misclassification. The more often a data sample is misclassified, the more important it becomes. The goal of the algorithm is to minimize an objective function:

$$O(x) = \sum_i l(\hat{y}_i, y_i) + \sum_t \Omega(f_t) \tag{2.22}$$

where $l(\hat{y}_i, y_i)$ is the loss function and $\Omega(f_t)$ is a regularization function that penalizes the complexity of the t^{th} tree. There are many different ways of iteratively adding learners to minimize a loss function. Some of the most common algorithms are AdaBoost [19], GradientBoost [20], and XGBoost [21]. Some common tree parameters that are usually tuned to increase accuracy and prevent overfitting are:

- the maximum depth of a tree, that specifies how tall a tree can grow,
- the number of maximum features, that specifies how many features can be used to build a given tree, where features are randomly selected from the total set of features,
- and the minimum number of samples per leaf, that specifies how many samples are required to create a new leaf.

Common boosting parameters are the learning rate that specifies how much to adjust the data weights after each iteration and the number of trees, which is the same as the number of iterations. Benefits of the BDTs are that the training and prediction is fast and the results are interpretable. Moreover, they are not sensitive to the scale of the input data, which means that the features can be a mix of categorical and continuous data. However, BDTs are sensitive to overfitting and have limited application in deep learning.

2.2 Deep Learning

Modern deep learning is characterized by the composition of a large number of various types of layers that are optimized with the SGD algorithm. Training a deep neural network with more than one layer that generalizes well can be challenging and can require large training datasets, powerful hardware, and a suitable network architecture.

2.2.1 Feed Forward Neural Networks

A Feed Forward Neural Network consists of L layers $f = f^{(L)} \circ \cdots \circ f^{(1)}$. The l^{th} layer is a function that maps a d_{l-1} dimensional input to a d_l dimensional output: $f^{(l)} : \mathbb{R}^{d_{l-1}} \rightarrow \mathbb{R}^{d_l}$. For $l < L$, the functions $f^{(l)}$ are called hidden layers, and the number of neurons d_l is called the width of the hidden layer. The layers in a Feed Forward Neural Network are defined as:

$$f^{(l)}(u) = \sigma^{(l)}\Big(W^{(l)}u + b^{(l)}\Big) \tag{2.23}$$

where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ is called the weight matrix, the components of the vector $b^{(l)} \in \mathbb{R}^{d_l}$ are referred to as the biases, $u \in \mathbb{R}^{d_{l-1}}$ is the input from the previous layer, $W^{(l)}u$ denotes a matrix-vector product, and $\sigma^{(l)}$ is a non-linear activation function that is usually applied element-wise. The parameters of the network are given by the set of all weights and biases, $\phi = \Big(W^{(1)}, \ldots, W^{(L)}, b^{(1)}, \ldots, b^{(L)}\Big)$.

The activation function σ in neural networks is a nonlinear function and its choice depends on the model architecture and application. A popular choice in deep learning is a Rectified Linear Unit (ReLU):

$$\sigma(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

for which the computational cost is small and the gradient does not vanish. For classification tasks the logistic function

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{2.25}$$

is a popular choice for the final layer. Another important function in deep learning is the softmax function that can be used to normalize the elements of a discrete vector u and to interpret the output of a model as a probability over a set of n discrete categories. It is commonly used at the last layer in multi-class classifiers. Given a real-valued input vector $u \in \mathbb{R}^n$, the softmax function computes the output vector $v \in \mathbb{R}^n$ and the i-th component is given by:

$$v_i = \frac{\exp(u_i)}{\sum_{j=1}^{n} \exp(u_j)} \ . \tag{2.26}$$

The output vector has the property that $v_i \in (0, 1)$ and $\sum v_i = 1$. The components of the input vector u are also called logits, due to their connection to the logistic function used in logistic regression.

The universal approximation theorem states that a Feed Forward Network with a single layer is sufficient to represent any function, but the layer may be impossibly large and may fail to learn and generalize correctly [22]. Often the use of deeper models can reduce the required number of neurons to approximate the desired function. In general, a neural network can be visualized as a graph, which is illustrated in Fig. 2.3.

2.2.2 Neural Network Variants: CNNs, RNNs and GNNs

Many variants of neural networks have been developed that are particularly suitable for specific data-structures and applications. Among the most important ones for HEP applications are Convolutional Neural Networks, Recurrent Neural Networks and Graph Neural Networks that will be briefly described in the following.

Convolutional Neural Networks (CNNs) are commonly used for image-like data. CNNs convolute the input image u and a filter W, also referred to as “kernel”. The parameters of the kernel are learnable and the convolution operation traverses over the input and calculates the inner product of the kernel W with the part of the input in the receptive field, which has the same spatial shape as the kernel and

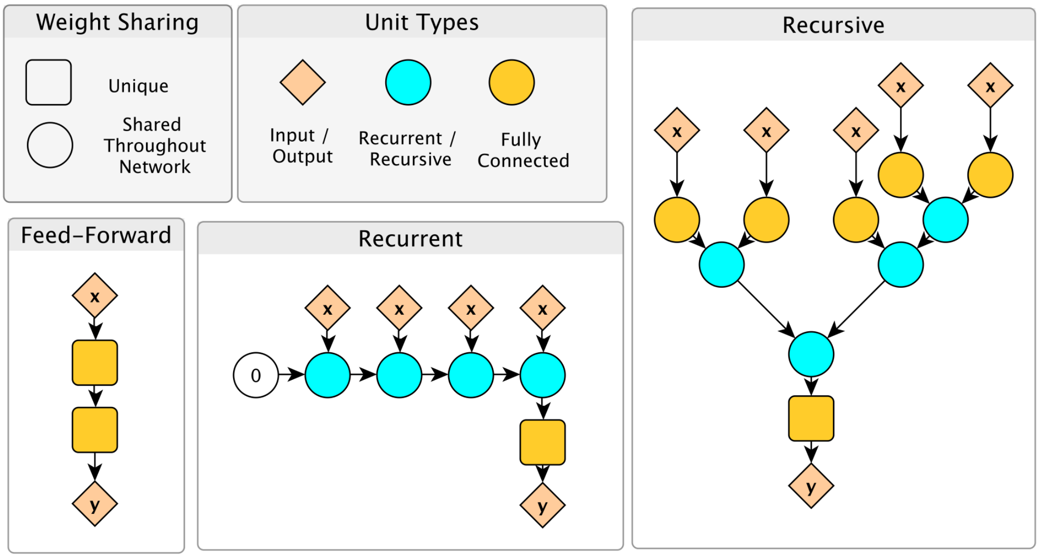


Figure 2.3: Sketch showing feed-forward, recurrent, and recursive neural network architectures. Diamonds represent inputs and outputs, while processing units are represented with circles and squares. The figure is taken from [23].

is centered at the target pixel. Each pixel can have a vector of features associated with it. The components of the features, indexed by c and c', are called channels. The convolution operation is often denoted with a $*$ and can be written as:

$$v_c(j) = (W * u_c)(j) = \sum_{c'} \sum_{i} W_{c,c'}(i) u_{c'}(“j - i”) \tag{2.27}$$

where “$j - i$” denotes the pixel index corresponding to the translation from pixel j to i. The result of a kernel convolution is also an image, and the image for a fixed channel index is referred to as a “feature map”.

Larger kernel sizes allow the filters to learn more complicated patterns, with the drawback of having more model parameters. In practice, kernel sizes of 3 are very common. A kernel size of 1 is referred to as a 1 × 1 convolution, which can be used to perform linear operations on the input features, such as increasing or decreasing the number of features. This is an important technique to extract more powerful features that can be used by the following layers, and also to compress features. Moreover, a typical CNN architecture often uses pooling, which effectively down-samples the image such that it can be processed at different resolutions. An important feature of CNNs is equivariance, which means that if the input image is shifted, then the output is also shifted by a similar amount. The CNN can be interpreted as a fully connected Feed Forward Network with shared weights that maintains the equivariance property.

Recurrent Neural Networks (RNNs) are a class of neural networks that are particularly suitable for sequential data. RNNs process sequences in such a way that information across the entire sequences can be accumulated and the model can operate on the previous states of the system. RNNs are commonly used for three types of tasks:

- **One-to-many tasks**: this task takes a single input and generates a sequence. An example is the generation of sequence data, such as a sentence or waveform, given a category.
- **Many-to-one tasks**: this task takes a sequence and generates an output. An example for this task is sequence-labeling.
- **Many-to-many tasks**: this task takes a sequence and generates a sequence where the length of input and output sequence may be the same. An example for this task is sequence to sequence mapping.

For sequential data where x_t represents each step in a sequence with $t \in [1, n]$ and h_t denotes the hidden state of the system, a simple RNN cell, which is a set of operations at each time step, may look like:

$$\begin{aligned} h_t &= g_h(Wx_t + Vh_{t-1} + b) \\ y_t &= g_o(Uh_t) \end{aligned} \tag{2.28}$$

where $W \in \mathbb{R}^{d_h \times d_i}$, $V \in \mathbb{R}^{d_h \times d_h}$, $U \in \mathbb{R}^{d_o \times d_h}$ are matrices and g_h and g_o represent functions. The dimension of input, hidden state and output are denoted as d_i, d_h, and d_o, and $b \in \mathbb{R}^{d_h}$ is a bias term. An RNN applies the same functions g_h and g_o repeatedly for each element of the sequence, which is similar to the shared weights in a convolutional filter. In practice, since recursive networks can grow very deep, simple recursive units encounter problems with vanishing or exploding gradients. Long sequences can be handled using a technique known as gating, where activation functions and transformations are applied selectively, or inputs can be ignored entirely. This mitigates the exploding and vanishing gradient problem at the expense of a more complicated recurrent unit. Examples for these units are long-short-term-memory (LSTM) units [24], and gated recurrent units (GRU) [25]. An extension of RNNs are bi-directional RNNs [26], that train two instead of one RNN on the input sequence, one on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and more detailed learning of the problem.

Graph Neural Networks (GNNs) are a class of neural networks that are suitable for graph-structured data. A graph consists of multiple nodes and edges between them. Graphs are highly flexible and allow to describe many types of structured data including images and sequences. An example for graph-structured data in HEP is the point cloud data type, which is an unordered set of points, for example the raw hits in a detector. Graph-based neural networks can be seen as a generalization of many types of machine learning models such as Recurrent Neural Networks and Convolutional Neural Network, which is studied in detail in the formulation of geometric deep learning [27]. In general, there are three types of prediction tasks on graphs: graph-level, node-level, and edge-level [28]. In a graph-level task the goal is to predict a single property for a whole graph. A node-level task aims at predicting some property for each node in a graph and in an edge-level task the goal is to predict the property or presence of edges in a graph. For a mathematical review of GNNs it is referred to reference [29].

2.2.3 Automatic Differentiation

Optimizing the parameters of a neural network often requires the calculation of gradients with respect to millions of parameters. Automatic differentiation (AD) is a technique for efficiently and accurately evaluating derivatives of numeric functions expressed as computer programs. It is much faster for the calculation of partial derivatives with a large number of inputs than traditional approaches, such as symbolic and numerical differentiation, and does not suffer from increasing errors when calculating higher derivatives. Moreover, symbolic differentiation has problems to convert a program into a single expression, and numerical differentiation suffers from round-off errors. The term $\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}(f(x,\theta))$, which needs to be computed in the Stochastic Gradient Descent algorithm, requires computing partial derivatives with respect to the parameters θ_i. If f is a composite model $f = f_n(f_{n-1}(\cdots, \theta_{n-1}), \theta_n))$, and if the functions f_i are differentiable, the chain rule can be applied:

$$\nabla_{\theta_i} \mathcal{L} = \frac{\partial \mathcal{L}(f(x,\theta))}{\partial \theta_i} = \frac{\partial \mathcal{L}}{\partial x_n} \cdot \frac{\partial x_n}{\partial x_{n-1}} \cdots \frac{\partial x_i}{\partial \theta_i} \tag{2.29}$$

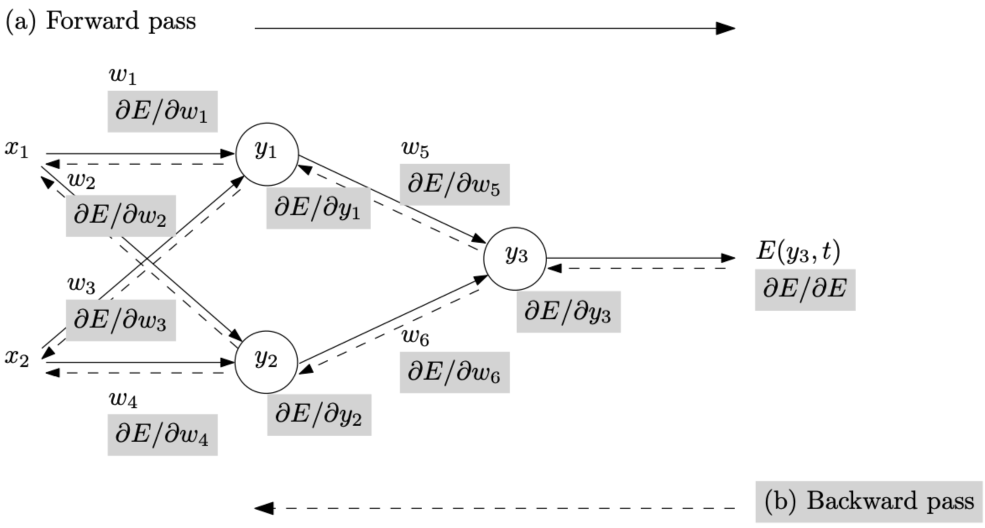

(a) Forward pass
w_1
$\partial E/\partial w_1$
x_1
y_1
w_2
$\partial E/\partial w_2$
w_5
$\partial E/\partial w_5$
$\partial E/\partial y_1$
y_3
$E(y_3, t)$
$\partial E/\partial E$
w_3
$\partial E/\partial w_3$
$\partial E/\partial y_3$
x_2
y_2
w_6
$\partial E/\partial w_6$
w_4
$\partial E/\partial w_4$
$\partial E/\partial y_2$
(b) Backward pass

Figure 2.4: Overview of backpropagation. (a) Training samples x_i are passed forward, generating corresponding activations y_i and an error E between the output y_3 and the target output t is computed. (b) The error is propagated backward, giving the gradient with respect to the weights $\nabla_{w_i} E = \left(\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_6}\right)$, which can then be used in the gradient-descent algorithm. The figure is taken from [30].

where x_n denotes the output of n-th composite function f_n. The computation of $\nabla_{\theta_i}\mathcal{L}$ for f_i requires the computation of a gradient at all preceeding functions. Accumulating the gradients of the differentiable functions in the reverse order of the composite model is referred to as backpropagation, which is widely used to train neural networks. An example for backpropagation is shown in Fig. 2.4.

More generally, there are two modes of AD: the forward and backward mode. For a composite function $f(x, \theta)$, the forward mode evaluates the chain rule in the same order of the forward evaluation of f by first computing $\partial f_1/\partial x$, then $\partial f_2/\partial f_1$, and finally $\partial f_n/\partial f_{n-1}$. The backward mode evaluates the chain rule in the reverse direction: starting from the last function $\partial f_n/\partial f_{n-1}$, then $\partial f_{n-1}/\partial f_{n-2}$, and finishing with $\partial f_1/\partial x$. The backpropagation algorithm in equation 2.29 can be implemented using the backward AD mode.

2.2.4 Vanishing Gradients and Regularization

As shown in equation 2.29, the gradient of the i-th function f_i is a product of gradients of the preceeding functions. If the gradients of these functions are very large or very small, the magnitude can either increase or decrease exponentially with the number of layers, which is referred to as exploding and vanishing gradient problems that are particularly critical for deep neural networks that consist of many composite functions. One way to mitigate an exploding gradient is to use gradient clipping, which treats the maximum gradient as a model hyperparameter. Moreover, many architecture designs are motivated by the vanishing and exploding gradient problem, such as LSTM cells. Moreover, the initialization of model parameters and normalization of input data can cause vanishing and exploding gradients. Thus, it is often useful to center the input values around zero with a similar level of covariance across the inputs. Since the mean and covariance of the data representations in hidden layers are evolving during the training phase, it is also often useful to explicitly normalize features in between hidden layers, for example with batch normalization that fixes the means and variances of each layer's inputs. Another common form of regularization for neural networks is called dropout, which randomly omits some of the model units during training and can be interpreted as a type of model averaging.

2.3 Inference

Statistical inference refers to the problem of making inferences about a probabilistic model given a sample of data. The two main approaches to statistical inference are frequentist and Bayesian. Frequentist statistics interprets probability as the frequency of the outcome of a repeatable experiment and, importantly, a probability for a hypothesis or for the value of a parameter is not defined. In Bayesian statistics, the interpretation of probability is more general and includes a degree of belief and it is therefore possible to define a probability density function (PDF) for the true value of a parameter. The frequentist and Bayesian approaches are fundamentally different interpretations of probability, however, for many inference problems they give similar numerical values. The following description of basic statistical methods is based on the Statistics chapter of the Particle Data Group review [9].

For an experiment with data $\boldsymbol{x}$, a hypothesis H is a statement about the probability for the data $p(\boldsymbol{x}|H)$. If the probability $p(\boldsymbol{x}|H)$ is a function of the hypothesis H, it is called the likelihood $L(H)$ of H. If the hypothesis is characterized by one or more parameters $\boldsymbol{\theta}$, the likelihood function can be written as $L(\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})$. In the Bayesian approach, inference is based on the posterior probability for H given the data $\boldsymbol{x}$, which quantifies the degree of belief that H is true given the data $\boldsymbol{x}$. This can be obtained from Bayes' theorem:

$$p(H|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|H)\pi(H)}{\int p(\boldsymbol{x}|H')\pi(H')dH'} \tag{2.30}$$

where $p(\boldsymbol{x}|H)$ is the likelihood for H. The quantity $\pi(H)$ is the prior probability for H, which represents the degree of belief for H before the measurement. The integral in the denominator serves as a normalization factor. If H is characterized by continuous parameters $\boldsymbol{\theta}$ then the posterior probability is a PDF $p(\boldsymbol{\theta}|\boldsymbol{x})$.

The statistical methods described in the following are implemented in the ROOT-based software packages RooFit/RooStats [31, 32]. Recently also a statistics package called pyhf [33] has been developed that implements many statistical methods based on modern tensor arithmetic provided by the popular ML frameworks, TensorFlow, PyTorch and JAX.

2.3.1 Parameter Estimation

In the frequentist approach, parameters are estimated with the maximum likelihood method or least squares fits. For a set of measured quantities $\boldsymbol{x}$ and the likelihood $L(\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})$ characterized by parameters $\boldsymbol{\theta}$, the maximum likelihood (ML) estimators for $\boldsymbol{\theta}$ are defined as the values that give the maximum of L. To avoid numerical problems, it is often better to work with the log-likelihood $\ln L$, that is maximized for the same parameter values of $\boldsymbol{\theta}$. An important property of maximum likelihood estimators is that they are consistent, efficient and for large datasamples unbiased. If the data consist of i.i.d. values, the joint PDF of the data sample factorizes and the likelihood function is given by:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}) \ . \tag{2.31}$$

If the probability to observe n events follows a Poisson distribution with mean μ and the independent observations $\boldsymbol{x}$ all follow $f(\boldsymbol{x}; \boldsymbol{\theta})$, then the likelihood can be written as:

$$L(\boldsymbol{\theta}) = \frac{\mu^n}{n!} e^{-\mu} \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}) \tag{2.32}$$

which is often called the extended likelihood. Under a change of parameters from $\boldsymbol{\theta}$ to $\boldsymbol{\eta}$, the ML estimators $\hat{\boldsymbol{\theta}}$ transform to $\boldsymbol{\eta}(\hat{\boldsymbol{\theta}})$, which means that the ML estimators are invariant under change of parameters. The inverse V^{-1} of the covariance matrix $V_{ij} = \mathrm{cov}\left[\widehat{\theta}_i, \widehat{\theta}_j\right]$ for a set of ML estimators can be estimated with the relation:

$$\left(\widehat{V}^{-1}\right)_{ij} \geq -\left.\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}\right|_{\widehat{\theta}} . \tag{2.33}$$

For a small number of samples, this equation can result in a misestimation of the variances. In the asymptotic limit (i.e., for large data samples) $\ln L$ is parabolic and s times the standard deviations σ_i of the estimators for the parameters can be obtained from the hypersurface defined by the parameters $\boldsymbol{\theta}$ such that:

$$\ln L(\boldsymbol{\theta}) = \ln L_{\max} - s^2/2 \tag{2.34}$$

where $\ln L_{max}$ is the value of $\ln L$ at the solution point. The minimum and maximum values of θ_i on the hypersurface define an approximate s-standard deviation confidence interval for θ_i.

In Bayesian statistics, all knowledge about $\boldsymbol{\theta}$ is contained in the posterior PDF $p(\boldsymbol{\theta}|\boldsymbol{x})$. The posterior can for example be summarized with the mean value and covariance matrix.

2.3.2 Nuisance Parameters

In the presence of systematic uncertainties, the models are not perfect and the estimated parameters $\boldsymbol{\theta}$ can have a systematic bias. The imperfections can be addressed by including additional parameters such that the more general model $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\nu})$ depends on the parameters of interest $\boldsymbol{\theta}$ and the nuisance parameters $\boldsymbol{\nu}$. The presence of nuisance parameters increases the statistical uncertainties for the parameters of interest. This happens because the estimators for the nuisance parameters and the parameters of interest will in general be correlated, which enlarges the contour defined by equation 2.34. To reduce the impact of the nuisance parameters they can be constrained with control measurements. If the control measurements $\boldsymbol{y}$ are statistically independent from $\boldsymbol{x}$ and are described by a model $p_y(\boldsymbol{y}|\boldsymbol{\nu})$, the joint model for both $\boldsymbol{x}$ and $\boldsymbol{y}$ is the product of the probabilities for $\boldsymbol{x}$ and $\boldsymbol{y}$, and the likelihood function for the full set of parameters becomes:

$$L(\boldsymbol{\theta}, \boldsymbol{\nu}) = p_x(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\nu}) p_y(\boldsymbol{y}|\boldsymbol{\nu}) \ . \tag{2.35}$$

Using all of the parameters $(\boldsymbol{\theta}, \boldsymbol{\nu})$ in equation 2.34 to calculate the statistical errors for the parameters $\boldsymbol{\theta}$ is equivalent to using the profile likelihood [34], which depends only on $\boldsymbol{\theta}$. The profile likelihood is defined as:

$$L_{\mathrm{p}}(\boldsymbol{\theta}) = L(\boldsymbol{\theta}, \widehat{\widehat{\boldsymbol{\nu}}}(\boldsymbol{\theta})) \tag{2.36}$$

where $\widehat{\widehat{\boldsymbol{\nu}}}$ denote the profiled values of the parameters $\boldsymbol{\nu}$, defined as the values that maximize L for the specified parameters $\boldsymbol{\theta}$.

In the Bayesian treatment of nuisance parameters one can obtain the posterior PDF for $\boldsymbol{\theta}$ by integrating over the nuisance parameters:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \int p(\boldsymbol{\theta}, \boldsymbol{\nu}|\boldsymbol{x}) d\boldsymbol{\nu} \tag{2.37}$$

which often needs Markov Chain Monte Carlo techniques to compute the integrals.

2.3.3 Confidence Intervals

Confidence intervals are intervals constructed such that they cover the true value of a parameter with a specified probability [9]. Frequentist intervals can be obtained with a procedure proposed by Neyman [35]. The boundary of the interval is given by a function of the data, that would fluctuate if the experiment was repeated many times. The coverage probability refers to the fraction of intervals that contain the true parameter value. Confidence intervals are constructed such that they have a coverage probability greater than or equal to a given confidence level, regardless of the true parameter's value. To illustrate the procedure, a PDF $f(x;\theta)$ is considered, where x represents the outcome of the experiment and θ is the parameter of interest for which a confidence interval will be constructed. With a pre-defined probability $1-\alpha$, for every value of θ a set of values $x_1(\theta,\alpha)$ and $x_2(\theta,\alpha)$ can be found such that the coverage condition is fulfilled:

$$P(x_1 < x < x_2; \theta) = \int_{x_1}^{x_2} f(x;\theta) dx \geq 1 - \alpha \ . \tag{2.38}$$

This is illustrated in Fig. 2.5, where a horizontal line segment $[x_1(\theta,\alpha), x_2(\theta,\alpha)]$ is drawn for several values of θ. The union of the intervals for all values of θ, indicated in the figure as $D(\alpha)$, is referred to as a confidence belt. When an experiment is performed to measure x and a value x_0 is obtained, one draws a vertical line through x_0. The confidence interval for θ is then defined as the set of all values of θ for which

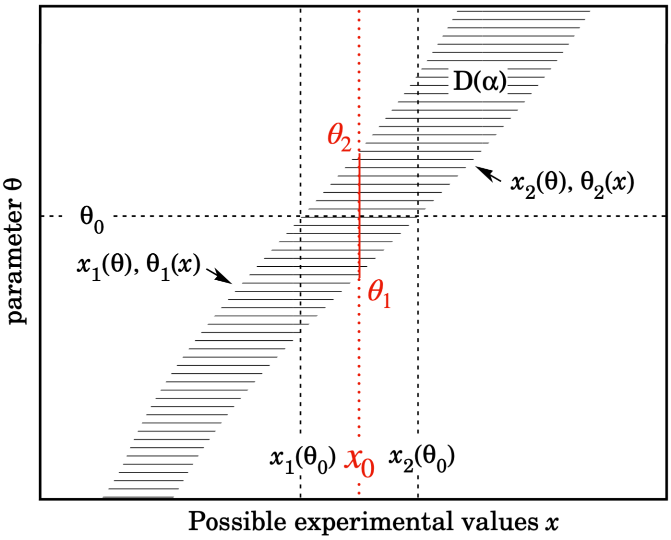

D(α)
θ_2
$x_2(\theta)$, $\theta_2(x)$
θ_0
$x_1(\theta)$, $\theta_1(x)$
θ_1
$x_1(\theta_0)$ x_0 $x_2(\theta_0)$
parameter θ
Possible experimental values x

Figure 2.5: Construction of the confidence belt. The figure is taken from [9].

the corresponding line segment $[x_1(\theta, \alpha), x_2(\theta, \alpha)]$ is intercepted by this vertical line. The confidence interval $[\theta_1, \theta_2]$, marked in red in Fig. 2.5, then has a confidence level (CL) equal to $1 - \alpha$.

If the experiment is repeated a large number of times, the interval $[\theta_1, \theta_2]$ covers the fixed value θ in a fraction $1 - \alpha$ of the experiments. The values of x_1 and x_2 are not determined uniquely by the condition of coverage. Central intervals can be chosen such that the probabilities to find x below x_1 and above x_2 are each $\alpha/2$, while for upper limits the lower bound can be set to zero and the upper to α, adapting equation 2.38. Due to the freedom to decide which values to include in the Neyman construction, also likelihood ratio ordering can be used to determine which values of x should be included in the confidence belt. The test statistic based on the likelihood ratio is defined as:

$$\lambda(\theta) = \frac{f(x; \theta)}{f(x; \hat{\theta})} \tag{2.39}$$

where $\hat{\theta}$ is the value of the parameter which maximizes $f(x; \theta)$ in the physical region. This results in the Feldman-Cousins intervals [36]. The Feldman-Cousins intervals have the advantage that they do not exclude parameter values to which one has little sensitivity and the prescription determines whether the interval is one- or two-sided such that the coverage probability is preserved. If the model contains nuisance parameters ν, they can be incorporated by profiling the likelihood as discussed in the section above.

An important example of constructing a confidence interval is when the data consist of random variables that follow a Gaussian distribution. In this case, the shape of the likelihood function around the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ can be used to approximate confidence intervals. Using Wilks' theorem [37], it can be shown that the confidence region for $\boldsymbol{\theta}$, that covers the true values with a certain, fixed probability, can be determined using the following relation:

$$\ln L(\boldsymbol{\theta}) \geq \ln L_{\max} - \Delta \ln L \tag{2.40}$$

where $\Delta \ln L$ depends on the number of parameter dimensions and the desired coverage. For example, the values of θ inside the 1-σ confidence region and for a one dimensional parameter, are given by $\Delta \ln L = 0.5$.

In the Bayesian approach, the Bayesian posterior probability can be used to determine regions with a given probability of containing the true value of a parameter of interest, however evaluating the posterior is often computationally expensive.

2.3.4 Hypothesis Tests

Frequentist hypothesis tests determine whether a hypothesis is accepted or rejected depending on the outcome of an experiment. A frequentist test of a hypothesis H_0 is a rule that states for which data $\boldsymbol{x}$ the hypothesis is rejected. A critical region, w, is defined such that the probability to find $\boldsymbol{x}$ under H_0 in w is smaller than a given probability α, referred to as the significance level of the test: $P(\boldsymbol{x} \in w|H_0) \leq \alpha$. The hypothesis H_0 is rejected if the data is observed in the critical region. As will be discussed in 3.1.1, signal-versus-background classification corresponds to hypothesis testing. The choice of the critical region is not unique and should take into account the probabilities for the data predicted by an alternative hypothesis H_1. Rejecting H_0 if it is true is called a type-I error, while not rejecting H_0 if an alternative H_1 is true is called a type-II error. The probability for a type-I error is by construction no greater than α, while the probability for a type-II error is given by $\beta = P(\boldsymbol{x} \notin w|H_1)$. The quantity $1 - \beta$ is referred to as the power of the test of H_0 with respect to the alternative H_1. A sketch of a hypothesis test is shown in Fig. 2.6. The important Neyman–Pearson lemma states that the power of a test of H_0 with respect to the

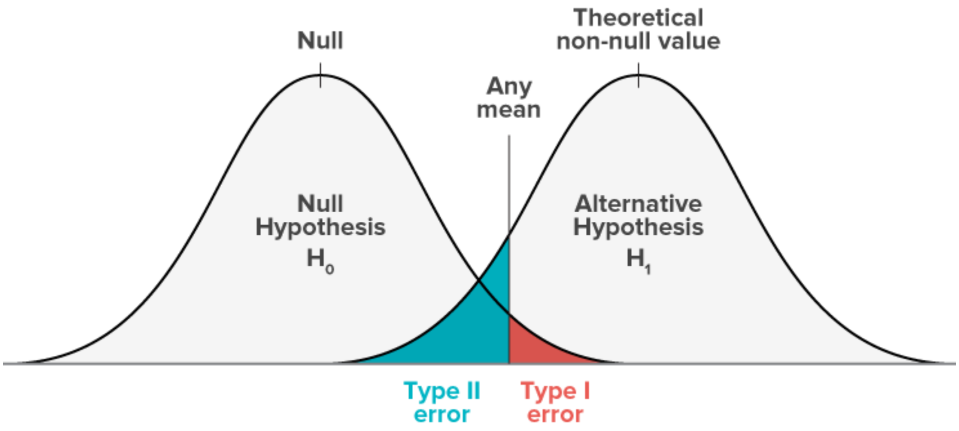


Figure 2.6: Sketch of a hypothesis test for the null hypothesis H_0 and the alternative hypothesis H_1. The figure is taken from [38].

alternative H_1 can be maximized by choosing the critical region w such that for all data values $\boldsymbol{x}$ inside w, the likelihood ratio

$$\lambda(\boldsymbol{x}) = \frac{f(\boldsymbol{x}|H_1)}{f(\boldsymbol{x}|H_0)} \tag{2.41}$$

is greater than or equal to a given constant c_α, and outside the critical region one has $\lambda(\boldsymbol{x}) < c_\alpha$. The value of c_α depends on the significance level α of the test [9]. The Neyman–Pearson lemma is equivalent to the statement that the likelihood ratio is the optimal test statistic. Type-I and type-II errors are also known as false positives and false negatives and the receiver operating characteristic (ROC) curve, that is a popular metric in machine learning, can be obtained by plotting the true positive rate against the false positive rate at various thresholds c_α.

To quantify the level of agreement between the data and a hypothesis without considering an alternative hypothesis, one can define a statistic t whose value reflects the level of agreement between the data and the hypothesis. The hypothesis H_0 will then determine the PDF $f(t|H_0)$ for the statistic. The p-value quantifies the significance of a discrepancy between the data and H_0. If t is defined such that large values correspond to a poor agreement with the hypothesis, then the p-value can be written as:

$$p = \int_{t_{\text{obs}}}^{\infty} f(t|H_0)dt \tag{2.42}$$

where t_{obs} is the experimentally observed value of the statistic. A hypothesis test can be formulated by defining the critical region such that that obtaining a p-value $p \leq \alpha$ implies that the data outcome was in the critical region. When searching for a new phenomenon in HEP, one often tries to reject the hypothesis H_0 that the data is consistent with the Standard Model. If the p-value of H_0 is sufficiently low, then one accepts that some alternative hypothesis is true. Often the p-value is converted into an equivalent significance Z, defined such that a Z standard deviation upward fluctuation of a Gaussian random variable would have an upper tail area equal to p:

$$Z = \Phi^{-1}(1-p) \tag{2.43}$$

where Φ is the cumulative distribution of the standard Gaussian, and Φ^{-1} is its inverse function. In HEP usually the level of significance for a discovery is $Z = 5$, corresponding to a p-value of 2.87×10^{-7}. However, in general the actual degree of belief that a new particle is discovered will depend on other factors as well, e.g. that the analyzers followed best scientific practices.

To find a p-value for $\boldsymbol{\theta}$ when the model also contains nuisance parameters $\boldsymbol{\nu}$, a test statistic q_θ can be constructed such that larger values of it correspond to increasing incompatibility between the data and the hypothesis. For an observed value of the statistic $q_{\theta,obs}$, the p-value of $\boldsymbol{\theta}$ is given by:

$$p_\theta(\boldsymbol{\nu}) = \int_{q_{\theta,\mathrm{obs}}}^{\infty} f(q_\theta|\boldsymbol{\theta},\boldsymbol{\nu})dq_\theta \ . \tag{2.44}$$

In the frequentist approach, $\boldsymbol{\theta}$ is rejected only if the p-value is less than α for all possible values of the nuisance parameters. This requires to define a test statistic q_θ such that its distribution $f(q_\theta|\boldsymbol{\theta},\boldsymbol{\nu})$ is independent of the nuisance parameters. While exact independence is only possible in special cases, it can be achieved approximately with the profile likelihood ratio. This is given by the profile likelihood divided by the value of the likelihood evaluated with the ML estimators $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\nu}}$:

$$\lambda_p(\boldsymbol{\theta}) = \frac{L(\boldsymbol{\theta},\widehat{\boldsymbol{\nu}}(\boldsymbol{\theta}))}{L(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\nu}})} \ . \tag{2.45}$$

Wilks' theorem [37] can be used to show that in the asymptotic limit the distribution of $-2\ln\lambda_{\mathrm{p}}(\boldsymbol{\theta})$ approaches a χ^2 distribution independent of the values of the nuisance parameters $\boldsymbol{\nu}$.

In the Bayesian approach, all knowledge about the model is contained in its posterior probability. A challenge in Bayesian hypothesis tests is the definition of the prior probability, since the posterior $p(H|x)$ is proportional to the prior probability $p(H)$, and often there is no consensus about the prior probabilities, e.g. for the existence of a new particle. However, it is possible to construct a quantity called the Bayes factor [39], which quantifies the degree to which the data prefers one hypothesis over another and is independent of their prior probabilities.

2.4 Applications in HEP

Machine learning has many different applications in particle physics. Some of the diverse areas where modern machine learning algorithms are used in HEP include event classification, tracking, triggering, object reconstruction, particle identification and calibration, fast simulation, as well as detector monitoring and production workflows. A more complete overview over the various applications in HEP can be found in [40]. In the following some selected examples for machine learning applications in HEP are discussed, based on the reviews in [23] and [41]. For a collection of relevant literature it is referred to [42].

2.4.1 Event Selection

One of the main applications of machine learning in high energy physics is supervised learning for the selection of signal events. Applying a selection on the classifier score often improves the signal efficiency and reduces the background, which overall improves the sensitivity of the measurement. It has been demonstrated that algorithms trained with physics-inspired high-level features are outperformed by deep networks based on features to which less pre-processing has been applied (low-level features). In the left panel of Fig. 2.7 the performance of deep networks in signal-background classification is compared to shallow networks with low- and high-level features [43]. The study demonstrates that deep networks with only low-level features outperform shallow networks that rely only on physics-inspired features such as reconstructed invariant masses. This suggests that by using only high-level features some of the information is lost. Feature engineering generally improves the performance of shallow neural networks and BDTs. However, with the rise of deep learning this is often no longer necessary and rather limits performance compared to working with the low-level features. By training all layers of a deep neural network simultaneously

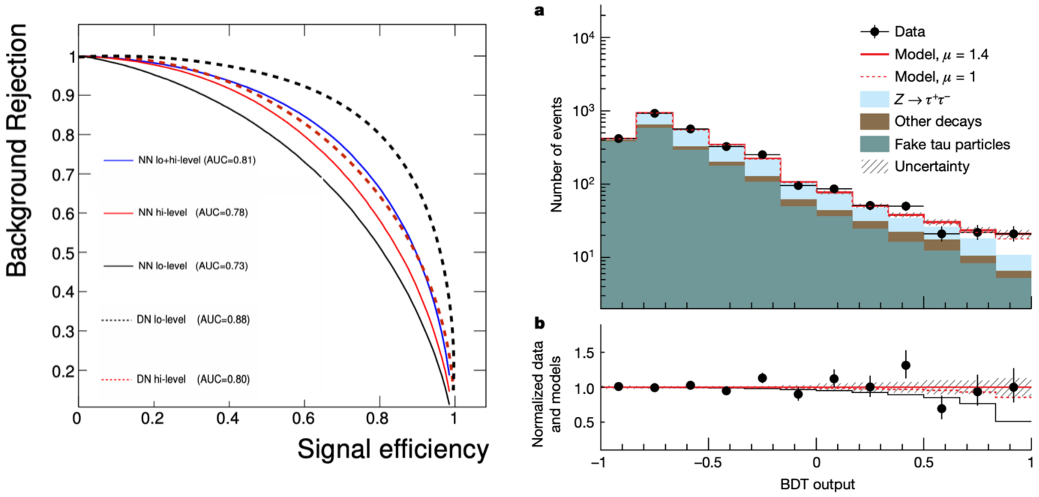


Figure 2.7: Left panel: comparison of the performance in signal-background classification between deep neural networks (DN), and shallow networks (NN) with low- and high-level features [43]. Right panel: the BDT-score distribution to separate signal and background for a search for the Higgs boson decaying to a tau-lepton pair by the ATLAS experiment is shown for one kinematic region [44].

one can interpret the intermediate layers as learned representations of the data, such that the neural network automatically engineers useful features.

Besides training machine learning algorithms in order to apply selections on the output score of the classifier, a common use case of supervised classification is the construction of low-dimensional event summaries, which allow to perform statistical inference on the parameters of interest. The learnt summary statistics can efficiently combine high-dimensional information from each event into one or a few variables, which may be used as the basis of statistical inference. These techniques improved the sensitivity of the measurements and were used in various analysis, among them the discovery of the Higgs boson [41]. An example for such an analysis is the measurement of the Higgs boson decaying into tau-leptons by the ATLAS collaboration [44]. In this analysis the data sample is divided into six kinematic regions. A BDT was trained to separate signal and background in each region using 12 discriminating input features. An example BDT output distribution obtained in one region is displayed in the right panel of Fig. 2.7. The combined analysis of all six regions provided strong evidence that the Higgs boson couples to tau-leptons, with about 40% better sensitivity achieved through the use of a BDT algorithm. The simulation that was used in this work was the basis of the 2014 Kaggle Higgs

Machine Learning Challenge and now is often used as a benchmark for novel algorithms. Moreover, multi-class classification has become a popular technique to define analysis categories.

2.4.2 Jet Identification

Machine learning has also been applied to a wide range of jet classification problems, in order to identify jets from heavy or light quarks, gluons, and W, Z, and H bosons. The application of machine learning improved the identification by using the low-level particle features within a jet. Since jets typically contain between 10 and 50 particles, the number of features is different for each jet. RNNs have proven to be highly successful at processing long sequences of data and thus are particularly suitable for jet classification problems. Applying an RNN to jet classification requires the particles in the jet to be ordered sequentially, such as sorting them according to the value of the transverse momentum. A set of features for each particle can then be provided to train the RNN to discriminate jets originating from different sources. Both ATLAS and CMS have developed flavor-tagging neural networks that rely on individual particle features. The ATLAS recurrent-network-based approach combined with traditional high-level features reduces the background by roughly a factor of two [46]. The CMS DeepJet neural network [45], shown in Fig. 2.8, uses three separate branches to process charged candidates, neutral candidates and secondary vertices. The algorithm applies 1×1 convolutional layers to perform automatic feature engineering for the different classes of jet constituents. The information for each sequence of constituents is combined with three LSTM layers. The full jet information is then combined using a fully connected layer. The DeepJet algorithm outperformed all previous flavour tagging approaches developed by the CMS collaboration. The newest generation of flavour tagging algorithms is based on GNNs and further improves the performance of jet identification [47, 48].

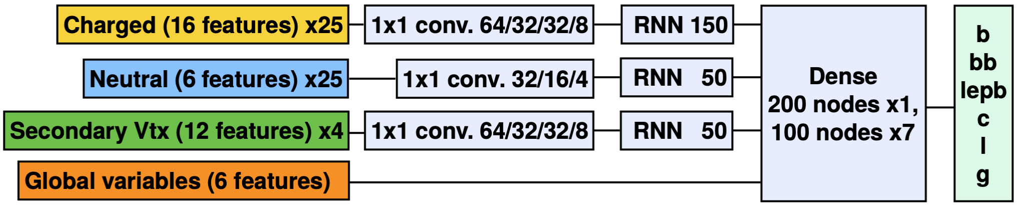


Figure 2.8: Sketch of the CMS DeepJet architecture that makes use of Recurrent Neural Networks. The figure is taken from [45].

2.4.3 Design Optimization of Detectors

The rise of deep learning also enables ambitious new programs to optimize complete workflows, such as the end-to-end optimization of detectors. Optimizing the design of detectors is a challenging task due to the complex interplay of physical processes with the detector material and the large choice of detector elements and geometries. However, as will be discussed in more detail in Chapter 4, so far the LHC has not detected any new physics, and therefore making optimal use of financial resources to build new detectors might become crucial for the progress of high energy physics in the future. The MODE collaboration [6] (an acronym for Machine-learning Optimized Design of Experiments) aims at developing tools based on deep neural networks and modern automatic differentiation techniques to implement a full modelling of all elements of the experimental design, achieving end-to-end optimization of the design of instruments via a fully differentiable pipeline. The objective function for this task can for example be defined as the expected precision of a measurement, and may be represented as a combination of performance and cost considerations that are balanced within reasonable limitations. Neural networks are naturally suitable for this task, since they can be used as surrogates for simulators to enable gradient-based optimization in cases where a simulator is non-differentiable. An example for an optimization pipeline for a muon radiography apparatus is shown in Fig. 2.9.

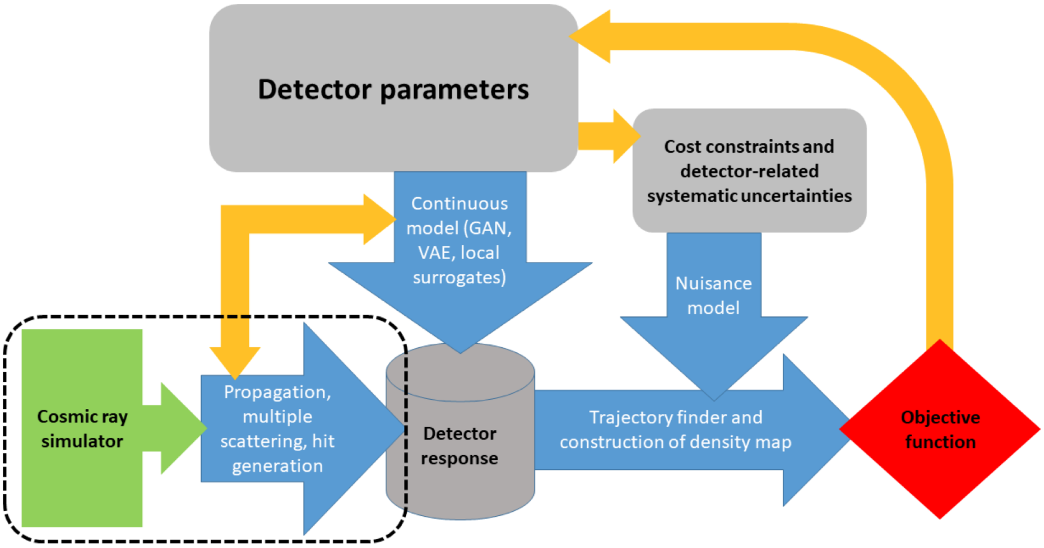

Detector parameters
Cost constraints and detector-related systematic uncertainties
Continuous model (GAN, VAE, local surrogates)
Nuisance model
Cosmic ray simulator
Propagation, multiple scattering, hit generation
Detector response
Trajectory finder and construction of density map
Objective function

Figure 2.9: Conceptual layout of an optimization pipeline for a muon radiography apparatus [6].

Cosmic rays are fed to a fast simulation of detection apparatus and scanned volume. The simulation of multiple scattering, particle propagation, and resulting electronic signals in the detector can be directly produced by a differentiable program. Alternatively, a differentiable module based on deep generative models or local generative surrogates can be used. A generation and validation loop keeps the model appropriate as the layout parameters are modified during the optimization task. After applying a reconstruction step, the output is used to compute a loss function that describes as closely as possible the real goal of the system. Exploratory studies have shown that very large gains in performance are potentially achievable even for very simple apparata [8, 49].

3 Inference Aware Neural Optimization

As discussed in Section 2.4, classification and regression models have become very popular in HEP to construct powerful summary statistics that are used for inference. However, as will be discussed in more detail in the following sections, neither the standard cross-entropy loss, nor the standard measures of performance for the learning task are aligned with the inference goal when the simulated events depend on nuisance parameters. The presence of nuisance parameters then causes a reduction of the statistical power of the summary statistics during inference. In recent years, a novel approach, called INFERNO [11], an acronym that stands for Inference-Aware Neural Optimization, has been developed to construct machine learning based summary statistics that are optimal for the specific analysis goal. In the following, event classification will be reviewed from a statistical perspective. Subsequently, an overview over existing approaches to deal with nuisance parameters in machine learning will be given. In the last section the INFERNO algorithm and its performance on a synthetic example will be reviewed. The following sections are based on the review in [10], the thesis [50], and the Statistics and Machine Learning chapters of the Physics Data Group review [9] and are the main references for the material presented below.

3.1 Event Classification

Event classification in HEP is commonly based on the training of probabilistic classifiers with samples of different processes obtained from MC simulations. In the following, the relation between probabilistic classifiers and probability density ratios will be reviewed, as well as the limitations of using probabilistic classifiers to construct summary statistics.

3.1.1 Classification and Density Ratios

In Chapter 2.1.3 it has been shown that a probabilistic classifier trained with the cross-entropy loss $\mathcal{L}_{\text{xe}}(y, f(x))$ will approximate the optimal classifier f^*. For binary classification with labels $y = \{0, 1\}$ this is equal to the Bayesian posterior probability $p(y = 1|x)$ that the label y equals 1 given the features x:

$$f^*_{\text{BCE}}(x) = \frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0) + p(x|y=1)p(y=1)} \,. \tag{3.1}$$

The prior distributions $p(y)$ on the labels of the classes represent the frequency in the training dataset. For binary classification problems, it is common to use a balanced training dataset with $p(y = 0) = p(y = 1) = \frac{1}{2}$. The true $p'(y = 1)$ in the experimental data can be very small if the true signal is small, or even zero in the case of a hypothetical particle. If $p'(y)$ and $p(y)$ are known, the posterior $p(y|x)$ can be re-calibrated from one prior to another with the Bayes theorem. An example of a re-calibration is the correspondence of binary classification to frequentist hypothesis tests, where according to the Neyman-Pearson lemma the optimal classifier is given by the likelihood-ratio:

$$f^*_{\text{N.P.}}(x) = \frac{p(x|y=1)}{p(x|y=0)} \tag{3.2}$$

which does not depend on the prior probabilities $p'(y = 0)$ and $p'(y = 1)$.
With the Bayes theorem it can be proven that both functions are related by a monotonic transformation:

$$f^*_{\text{N.P.}}(x) = \frac{p(y=0)}{p(y=1)} \frac{f^*_{\text{BCE}}(x)}{1 - f^*_{\text{BCE}}(x)} \tag{3.3}$$

which is also referred to as the likelihood-ratio trick. The tradeoff of type-I and type-II error is not affected by this transformation, therefore the ROC curve for $f_{\text{N.P.}}$ and f_{BCE} are identical and do not depend on the prior probabilities $p(y)$. By training a probabilistic classification model with binary cross-entropy, the likelihood ratio $r(x) = p(x|y = 1)/p(x|y = 0)$ can be approximated by:

$$\frac{f^*_{\text{BCE}}(x)}{1 - f^*_{\text{BCE}}(x)} = \frac{p(x|y=1)}{p(x|y=0)} \frac{p(y=1)}{p(y=0)} = r(x) \frac{p(y=1)}{p(y=0)} \tag{3.4}$$

where the equality is only true for the optimal Bayes classifier. This relation can also be obtained for other approaches that minimize continuous relaxations of the zero-one loss and can be generalised for the multi-class case. Interpreting probabilistic classifiers as approximations for density ratios allows to study the limitations of traditional machine learning approaches.

3.1.2 Sufficient Summary Statistics

A summary statistic for a set of n i.i.d. events $D = \{x_0, \ldots, x_n\}$, where each $x \in \mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional representation of the event information, is a function of the data D that reduces the dimensionality from $n \times d$ to $n \times b$:

$$s(x) : \mathcal{X} \subseteq \mathbb{R}^d \longrightarrow \mathcal{Y} \subseteq \mathbb{R}^b \tag{3.5}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is the original feature space, and $\mathcal{Y} \subseteq \mathbb{R}^b$ the new low-dimensional space. Since the generating probability distribution $p(x|\theta)$ is in general not known analytically and has to be estimated from a finite number of simulated samples, only summary statistics that are as low-dimensional as possible and approximately preserve the information are relevant for inference. In HEP applications, often simple sample-wise statistics, such as histograms, are constructed from $s(x)$, that are then used to construct Poisson-count likelihoods. The effect of nuisance parameters is modeled by producing sets of simulations with different values of the nuisance parameters and making use of interpolation algorithms.

The Fisher–Neyman factorization theorem states that a summary statistic for a set of n i.i.d. observations x is sufficient with respect to a statistical model and a set of parameters θ if the generating probability distribution function of the data $p(x|\theta)$ can be factorized as:

$$p(x|\theta) = q(x)r(s(x)|\theta) \tag{3.6}$$

where $q(x)$ is a non-negative function that does not depend on the parameters θ and $r(x)$ is a non-negative function that depends on θ and depends on the data x only through the summary statistic $s(x)$. A sufficient statistic contains all information about the relevant model parameters θ, and no information can be added by any complementary statistic.

For a two-component mixture model it can be shown analytically that an approximately sufficient summary statistic can be obtained by training a probabilistic classifier. The two-component mixture model forms the basis for both cross-section measurements and new physics searches and thus is of particular importance in HEP. In the absence of nuisance parameters, it can be written as:

$$p(x|\mu) = (1-\mu)p_b(x) + \mu p_s(x) \tag{3.7}$$

where μ is a parameter corresponding to the signal mixture fraction and p_s and p_b are the PDFs for the signal and background distribution. This can be rewritten as:

$$p(x|\mu) = p_b(x)\left(1 - \mu + \mu\frac{p_s(x)}{p_b(x)}\right) \tag{3.8}$$

from which can be proven that the density ratio

$$s_{s/b}(x) = \frac{p_s(x)}{p_b(x)} \tag{3.9}$$

is a sufficient summary statistic for the mixture coefficient μ, according to the Fisher-Neyman factorisation criterion defined in equation 3.6. It can be shown that any bijective function of a sufficient summary statistic is also a sufficient summary statistic, thus the conditional probability

$$s_{s/(s+b)}(x) = \frac{p_s(x)}{p_s(x)+p_b(x)} \tag{3.10}$$

is a sufficient summary statistic as well. This quantity can be approximated by training a balanced probabilistic classifier as defined in equation 3.1:

$$f^*_{\text{BCE}}(x) = \frac{p(x|y=1)}{p(x|y=0)+p(x|y=1)} \tag{3.11}$$

with the additional advantage that the output score of this classifier is bounded between zero and one. The fact that signal-versus-background classification allows to obtain an approximately sufficient summary statistic for the mixture model and mixture fraction μ explains why these techniques are frequently used in HEP.

3.1.3 Why Is Classification Not Enough?

In the previous section, signal-versus-background classification was reviewed from a statistical perspective. It has been shown that training a probabilistic classifier approximates the density ratio $r(x) = p(x|y=1)/p(x|y=0)$ between the signal and background generating distributions. In the case when the generating probability distributions of the data are not fully specified, but depend on additional unknown nuisance parameters θ, a probabilistic classifier trained to distinguish samples from the data-generating distributions $p(x|\theta, y = 1)$ and $p(x|\theta, y = 0)$ will approximate a function of the density ratio

$$r(x; \theta) = \frac{p(x|\theta, y = 1)}{p(x|\theta, y = 0)} \tag{3.12}$$

that will itself depend on the actual value of the parameters θ and the Neyman-Pearson lemma does not hold any more. Thus, in the presence of nuisance parameters, even an optimal probabilistic classifier is not guaranteed to provide a transformation that is optimal for inference. Moreover, if it is assumed that the true value of the parameters is fixed but unknown, as usually done in frequentist inference in HEP, then the optimal classifier is not uniquely defined. Training a classifier with simulated data generated for a specific value of the nuisance parameters θ might not be optimal for the classification of experimental data that correspond to the unknown true parameter values θ_{true}. This is the main issue when probabilistic classifiers are used to construct summary statistics for inference. In practice, classifiers can be trained for the most probable value of the nuisance parameters and their effect can be accounted for during inference. However, with this approach the statistical power for inference can degrade even if the classifier is optimal.

The limitations of classification for statistical inference can also be formulated based on the sufficiency conditions for summary statistics, as defined by the Fisher-Neyman factorisation criterion. If the mixture model depends on additional nuisance parameters θ it can be written as:

$$p(x|\mu, \theta) = p_b(x|\theta)\left(1 - \mu + \mu\frac{p_s(x|\theta)}{p_b(x|\theta)}\right) \tag{3.13}$$

and it can be seen that the Fisher-Neyman factorisation criterion from equation 3.6 is not fulfilled any more, since p_b and p_s depend on the nuisance parameters θ. Thus, even a Bayes optimal probabilistic classifier does not provide a sufficient summary statistics and useful information for inference might be lost if a low-dimensional classification-based summary statistic is used.

3.2 Nuisance Parameters in Machine Learning

There is a growing interest in the development of new techniques to mitigate the effect of nuisance parameters in inference problems. Recent work has shown that some of the innovations in the field of machine learning can be used to deal more closely with the statistical inference objective of HEP analyses and several methods have been developed. In the following, the various existing approaches are summarized based on the review in [10], which is also recommended for a more detailed reading.

3.2.1 Nuisance-Parametrized Models

The most direct way to account for the effect of nuisance parameters in the construction of a summary statistic is to include them in the physical model by parametrizing their effect on the event features. For simple problems it is sometimes possible to develop an analytical solution. An example in HEP is the decorrelation of the variable τ_{21} that is used to study the sub-structure in hadronic jets [51]. The τ_{21} variable has a dependence on the jet p_{T}, thus applying a selection on this variable biases the distribution of the reconstructed jet mass. It is possible to remove this bias almost entirely by parametrizing the dependence of τ_{21} on the jet p_{T} [52].

If experimental data is available that is informative of the value of the nuisance parameters it sometimes can be exploited in the construction of summary statistics with probabilistic classifiers. This was first studied by Neal [53], who addressed the problem of how the construction of a sufficient summary for the signal fraction θ with a binary classifier is affected by unknown parameters α, when these modify the PDF of the signal and background events. The proposed solution is the construction of low-dimensional summary statistics for both the nuisance parameters α and the observable event features x with a probabilistic classifier. If suitable parametric models of the summaries can be constructed, they can be used for inference, exploiting the informative power of the data to constrain the nuisance parameters. Approximate sufficiency can be reached, if the parametrizations do not cause a significant loss of information.

In case no knowledge or constraints on a nuisance parameter is available, sometimes it is possible to parametrize its effect on the observations. An example of this situation is the search for a new particle whose true mass M_{true} is unknown.

A classifier trained with signal events simulated assuming a mass $M_1 = M_{true} + \alpha$, suffers from a progressive degradation in performance as $|\alpha|$ increases. A possible solution [54, 55] is to independently train a set of classifiers using data simulated with different mass values for the unknown signal. However, this approach is still sub-optimal, since each classifier is ignorant about the information processed by the other ones. A way to avoid this limitation is to parametrize the effect of the nuisance parameter in the construction of the classifier [56] by including the unknown value of M_{true} in the set of features of the signal events when training the classifier. The advantage of this procedure is that an interpolated classification score for events with mass values never seen during training can be obtained.

3.2.2 Feature Decorrelation and Penalized Methods

If a direct parametrization of the effect of nuisance parameters is not feasible, several alternative approaches exist. In the context of the search for new physics, the ATLAS and CMS experiments have developed methods to increase the signal purity without modifying the shape of the distribution of the reconstructed mass, M_{rec}. The goal of these methods is to avoid that a selection on the output of a classifier biases the background towards displaying a ”signal-like” mass distribution, which enhances systematic uncertainties on the estimate of the signal fraction and hinders the application of bump-hunting techniques. The technique to reduce the dependence of a classification score on the mass is called *planning* [57, 58]. It is implemented by pre-selecting training samples for signal and background such that they have the same marginal PDF in the variable one aims to decorrelate, $p_S(M_{rec})^{sel} = p_B(M_{rec})^{sel}$. This can be implemented by weighting each event i by a mass-dependent value $w(M_{rec,i})$:

$$w(M_{rec,i}) = \left\{ \begin{array}{l} 1/p_S(M_{rec,i})^{\text{train}} \ , i \in S \\ 1/p_B(M_{rec,i})^{\text{train}} \ , i \in B \end{array} \right\} . \tag{3.14}$$

The weights are applied in the loss function of the classifier in the training stage, but are not used during the validation and testing stage. It has been shown that *planning* can significantly reduce the correlation between the classifier output and the planed variable in specific situations. However, the effectiveness of this approach is limited when the classifier learns the value of the planed variable indirectly from other event features.

In [57] it was also shown that a decorrelation of the output of a neural network classifier from the mass of boosted hadronic jets can be possible by feature pre-processing based on principal component analysis.

If a decorrelation of the classifier output from a variable of interest is not feasible due to other informative event features, a different type of solutions tries to make the classifier score independent from variations in the value of the nuisance parameters by implementing a robust optimization objective for classification. The first algorithm that designed such an objective is *uBoost* [59], which relies on boosted decision trees to improve signal purity. The method is based on the *AdaBoost* algorithm of increasing the weight of training events misclassified by the decision tree in the previous iteration. It augments the algorithm by modifying the weight of signal events depending on the disuniformity of the selection. The uniformity weight is defined as the inverse of the density of signal in the proximity of the event, and is computed with the k-NN algorithm. Applied on a Dalitz analysis [59], the method was shown to achieve uniformity with almost no degradation in classification performance. Several other approaches that try to achieve a uniform selection efficiency of a BDT classifier were introduced in [60] and it was shown that they outperform *uBoost* in specific situations.

The robustness to nuisance parameters has also been addressed by adding suitable regularizer terms to the loss function of neural network classifiers. In [61] a measure is introduced that quantifies to which extend two sets of features x and y are independent. The proposed measure is called *DisCo* ("distance correlation") and is bound between 0 and 1, where a value of 0 indicates that x and y are fully independent. The measure is differentiable and and its value can be added as a penalty term to the loss of the classifier. A hyperparameter λ allows to control the amount of interdependence of x and y when minimizing the penalized loss.
A similar approach is taken in [62] where the authors aim at decorrelating a neural network classifier output from nuisance parameters based on an approximately differentiable histogram. The loss of the classifier is penalized by a term derived from the difference in the smoothed bin counts of the nominal output and its nuisance-varied value, which decorrelates the classifier output from the nuisance parameters.

3.2.3 Adversary Losses

Another possible approach is the use of adversarial techniques to find the best compromise between signal discrimination and impact of nuisances. The main idea of adversarial setups are two independent neural networks that compete with each other in the search for the optimal working point in a constrained classification problem. The global loss function is the combination of a classification loss and a penalization loss that tries to learn information about the nuisance parameter from the output of the classifier. Adversarial neural networks in HEP problems were first studied by Louppe, Kagan and Cranmer [63], who introduced adversarial techniques to make the classification score pivotal, such that its distribution is independent on the value of nuisance parameters. If the nuisance parameters affect the shape of the decision boundary, a hyperparameter λ multiplying the adversary loss can be introduced. Louppe et al. study this approach both with a synthetic example and a HEP use case where the nuisance parameter Z is categorical, describing the absence ($Z = 0$) or presence ($Z = 1$) of pile-up in LHC collision data. In the case of $Z = 1$ they show how a compromise between the classification and the pivotal tasks can be obtained by tuning the parameter λ.

The adversarial technique has been applied to the discrimination of the decay of boosted heavy particles [64] where background systematics affect the inference of the neural network based selection. In [65] the effectiveness of the adversarial training to alternatives based on data augmentation and tangent propagation is studied based on the Higgs Kaggle Challenge [66]. Another study in [67] examines adversarial classification as a preliminary step to reduce the dependence of an autoencoder task on systematic uncertainties. Moreover, the study in [68] addresses the problem of theoretical uncertainties with adversarial networks. While adversarial methods were shown to achieve approximate independence of the classifier output from nuisance parameters, there is no guarantee that the equilibrium point between the two competing tasks is optimal for the inference goal of the analysis.

3.2.4 Semi-Supervised Approaches

An alternative approach to deal with nuisance parameters is exploiting experimental data to complement or substitute simulated samples in the model training. This aims at minimizing the gap between the inference performance of real and simulated data. The approach is based on novel ideas from weak supervision and semi-supervised learning and has the advantage that a classifier could be trained using data from the experiment. Dery et al. [69] proposed a method, referred to as learning from label proportions (LLP), where a neural network is trained only based on class proportions. The method is applied to a quark-versus-gluon tagging problem, and the authors find that the performance is similar to that of a fully supervised classifier, while being more robust to mismodelled input variables. However, this approach requires knowledge of the label proportions in the mixed samples, which might be unknown during the training stage.

To overcome this limitation, a method called classification without labels (*CWoLA*) was developed by Metodiev et al. [70]. The main idea of this approach is to train a probabilistic classifier to distinguish between two mixed samples with different, and possibly unknown, component fractions. This simplifies the *LLP* approach because it is based on a standard classification loss, where the label is not the observation class but an identifier of the mixed sample it belongs to. The authors show that the optimal binary classifier to distinguish samples from each of the mixed samples is a function of the density ratio between the components. Both, *CWoLA* as well as *LLP* are tested with practical examples such as a quark-versus-gluon discrimination problems. Two other studies have applied variations of *CWoLA* to a new-physics search for gluino production [71] and the quark-versus-gluon discrimination problem [72]. However, while weak supervision can in principle be useful to build classifiers that are more robust to certain types of mismodelling, existing practical approaches do not fully address the issue of dealing with nuisance parameters [10].

3.2.5 Bayesian Neural Networks

Recently, there have also been proposals to use Bayesian Neural Networks for the estimation of uncertainties. Bayesian Neural Networks allow to estimate uncertainties of neural networks by treating their weights as distributions, such that the network output is a distribution instead of a fixed value. The uncertainties of a network can then be estimated by combining multiple measurements of the same test data to calculate the mean prediction and its standard deviation. Recently, several attempts have been made to apply Bayesian Neural Networks to HEP problems. In [73] the authors show that Bayesian Neural Networks can be used in the event reconstruction of collider events to improve the prediction accuracy and uncertainties. The authors in [74] demonstrate how to treat systematic uncertainties with the use of Bayesian Networks for a synthetic top tagger example that uses jet images. Building on this work, in [75] a Deep Bayesian Neural Network is used in a regression task to treat systematic uncertainties on the momenta of boosted top quarks forming fat jets.

3.2.6 Simulation Based Inference

The goal of simulation-based inference is to extend statistical procedures to the situation where one does not know the explicit likelihood $p(x|\theta)$, but has access to a simulator that defines the likelihood implicitly [76]. The key idea of the concept is the training of a classifier using supervised learning to discriminate two sets of data, that both come from the simulator and are generated for different parameter points θ_0 and θ_1. The classifier output function can be converted into an approximation of the likelihood ratio $r(x|\theta_0, \theta_1) = p(x|\theta_0)/p(x|\theta_1)$ between θ_0 and θ_1. The approach is amortized, which means that after a simulation and training phase, the surrogates can be evaluated efficiently for arbitrary data and parameter points. The use of neural networks eliminates the requirement of low-dimensional summary statistics, because the model learns the structures in high-dimensional data, which potentially improves the quality of inference. With the drawback of a more complex training procedure, this technique is the first general solution for dealing with nuisance parameters when using machine learning for HEP inference. In the original paper [78], Cranmer et al. introduced a generic framework for inference using calibrated parametrized classifiers referred to as CARL.

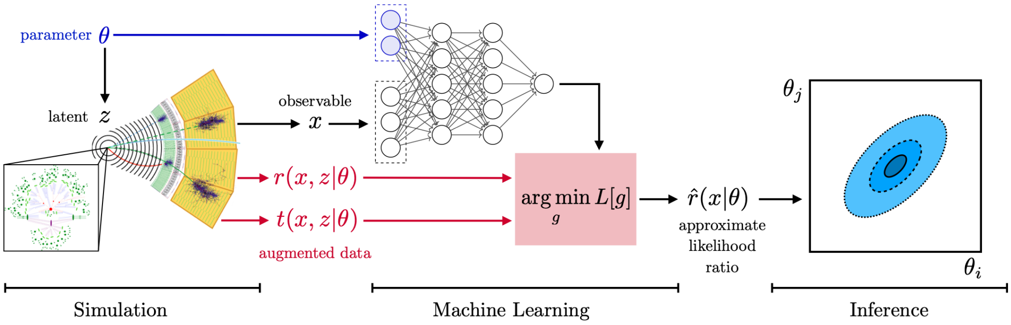


Figure 3.1: Sketch of a typical HEP analysis. Simulation based inference methods propose to obtain additional information from the simulator that can be used to train neural networks to efficiently approximate arbitrary likelihood ratios for inference. The figure is taken from [77].

Brehmer et al. further extended the approach of parametrized classifiers to better exploit the latent-space structure of generative models from complex scientific simulators [77, 79, 80, 81]. The authors show that in some situations, that are common for HEP analysis, it is possible to extract additional quantities from the simulator that characterize the likelihood of the latent process. A sketch of a typical HEP analysis is shown in Fig. 3.1, where the simulator is a Monte Carlo event generator. The additional information can be used to augment the training data for surrogate models. This often allows to solve the supervised learning task more efficiently, which can improve the sample efficiency in the inference task. In some cases, one can augment the training dataset to include the joint likelihood-ratio

$$r(x_i, z_i|\theta_0, \theta_1) := p(x_i, z_i|\theta_0)/p(x_i, z_i|\theta_1) \tag{3.15}$$

where z_i are unobserved latent variables from the simulator. This can be used to reduce the variance of the loss function. While the marginal likelihood $p(x|\theta)$ is intractable due to the high-dimensional integral over the latent space, the joint likelihood is often tractable and therefore it is often possible to augment the training dataset with the joint score

$$t(x_i, z_i|\theta_0) := \nabla_\theta \log p(x_i, z_i|\theta)|_{\theta_0} \ . \tag{3.16}$$

Based on this additional information, Brehmer et al. developed several new methods that extend CARL to more efficiently approximate the parametrized likelihood ratio $r(x|\theta_0, \theta_1)$ and they demonstrate the effectiveness in several example problems. Moreover, they developed a new class of methods referred to as SALLY using the regressed score approximation $\hat{t}(x|\theta_0)$ at a single reference parameter point θ_0 to construct a summary statistic. The score $\hat{t}$ defines an optimal summary statistic in the neighborhood of θ, and thus is useful for inference. To further reduce the dimensionality if the number of parameters is large, the authors propose another technique, referred to as SALLINO.

A challenge for the application of these methods in HEP, particularly for the methods that use augmented data from the simulator, is to approximate or model the effect of all relevant nuisance parameters in the joint likelihood ratio and score. To facilitate this, Brehmer et al. developed a software library MADMINER [82] to simplify the application of these techniques to LHC measurements.

3.3 INFERNO

In the previous sections it has been shown that classification-based summary statistics cannot easily account for the effects of nuisance parameters and their power for statistical tests is reduced when nuisance parameters are taken into account during inference. The simulation-based inference techniques, that propose a general solution for dealing with nuisance parameters in machine learning, do not construct summary statistics, but aim at directly addressing the inference problem. While this approach has great potential, it is quite different from the typical approaches in HEP. Therefore, adapting it in large experimental collaborations may be challenging. In recent years complementary inference-aware techniques have been developed, that aim at constructing machine-learning based summary statistics that are better aligned with the statistical inference goal of HEP analysis and can be used in place of traditional histograms. A generic technique in this category is INFERNO [11]. In this work the authors P. De Castro and T. Dorigo show how non-linear summary statistics can be constructed by minimizing inference-motivated losses via stochastic gradient descent. The algorithm is studied with a synthetic example, inspired by a typical cross-section measurement. The proposed algorithm can be used to directly minimize the approximated variance of the parameter of interest, fully accounting for the effect of relevant nuisance parameters.

3.3.1 Algorithm

The INFERNO algorithm [11] aims at directly minimizing the expected variance of the parameter of interest (POI) obtained via a non-parametric simulation-based synthetic likelihood. The parameters of a neural network are optimized by stochastic gradient descent via automatic differentiation, where the loss function accounts for the details of the statistical model and in particular the effect of nuisance parameters. The original algorithm has been implemented in TensorFlow 1 [83]. A sketch of the INFERNO algorithm is shown in Fig. 3.2. An inference-aware summary statistics

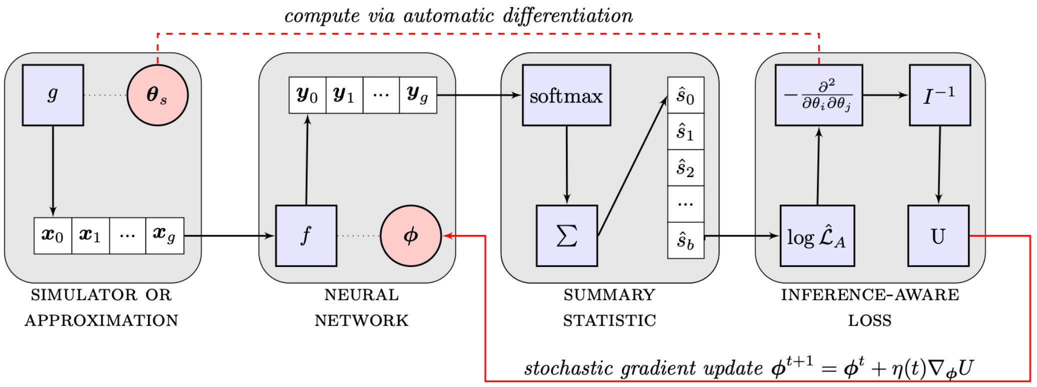

compute via automatic differentiation
g
$\boldsymbol{\theta}_s$
$\boldsymbol{x}_0$ $\boldsymbol{x}_1$ ... $\boldsymbol{x}_g$
SIMULATOR OR APPROXIMATION
$\boldsymbol{y}_0$ $\boldsymbol{y}_1$ ... $\boldsymbol{y}_g$
f
$\boldsymbol{\phi}$
NEURAL NETWORK
softmax
$\sum$
$\hat{s}_0$ $\hat{s}_1$ $\hat{s}_2$ ... $\hat{s}_b$
SUMMARY STATISTIC
$-\frac{\partial^2}{\partial\theta_i\partial\theta_j}$
I^{-1}
$\log\hat{\mathcal{L}}_A$
U
INFERENCE-AWARE LOSS
stochastic gradient update $\boldsymbol{\phi}^{t+1} = \boldsymbol{\phi}^t + \eta(t)\nabla_{\boldsymbol{\phi}}U$

Figure 3.2: Sketch of the INFERNO algorithm. Batches from a simulator are passed through a neural network and a differentiable summary statistics is constructed that allows to calculate the variance of the POI. The parameters of the network are then updated by SGD. The figure is taken from [11].

is learnt by optimizing the parameters $\boldsymbol{\phi}$ of a neural network f in order to reduce the dimensionality d of each input observation $\boldsymbol{x}$:

$$f(\boldsymbol{x};\boldsymbol{\phi}) : \mathbb{R}^d \to \mathbb{R}^b \ . \tag{3.17}$$

The network is trained with batches of simulated samples $G_s = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_g\}$ obtained from a simulator g with parameters $\boldsymbol{\theta}_s$. Here G_s denotes one batch of samples. The number of nodes in the last layer of the network determines the dimension b of the summary statistics. Since histograms are not differentiable, the original algorithm uses a softmax function as a differentiable approximation for the neural network output y:

$$\hat{s}_i(G_s;\boldsymbol{\phi}) = \sum_{\boldsymbol{x}} \frac{e^{f_i(\boldsymbol{x};\phi)/\tau}}{\sum_{j=0}^{b} e^{f_j(\boldsymbol{x};\phi)/\tau}} \tag{3.18}$$

where the temperature hyperparameter τ regulates the softness of the operator. For small temperatures $\tau \to 0^+$, the probability of the largest component will tend to 1 while others to 0. With this approximation it is possible to construct a summary statistic for each batch by computing the Asimov Poisson-count likelihood $\hat{\mathcal{L}}_A$ [84]:

$$\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi}) = \prod_{i=0}^{b} \text{Pois}(\hat{s}_i(G_s;\boldsymbol{\phi})|\hat{s}_i(G_s;\boldsymbol{\phi})) \ . \tag{3.19}$$

The MLE for the Asimov likelihood is the parameter vector $\boldsymbol{\theta}_s$ used to generate the simulated dataset G_s, i.e. $\text{argmax}_{\boldsymbol{\theta}}\left(\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi})\right) = \boldsymbol{\theta}_s$. The concept of Asimov data is also referred to as "saturated models" in statistics. The effect of the parameters of interest and the main nuisance parameters can be included by changing the mixture coefficients of mixture models, translations of a subset of features, or conditional density ratio re-weighting. An example for this will be discussed in the next section, where the application of the algorithm to a synthetic example is described. From the Asimov likelihood the Fisher information matrix is then calculated via automatic differentiation according to:

$$\boldsymbol{I}(\boldsymbol{\theta})_{ij} = \frac{\partial^2}{\partial\theta_i \partial\theta_j}\left(-\log \hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi})\right) \ . \tag{3.20}$$

As has been discussed in Chapter 2.3.1, the covariance matrix can be estimated from the inverse of the Fisher information matrix if $\hat{\boldsymbol{\theta}}$ is an unbiased estimator of the values of $\boldsymbol{\theta}$:

$$\text{cov}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) \geq I(\boldsymbol{\theta})^{-1} \ . \tag{3.21}$$

It is also possible to include auxiliary measurements that constrain the nuisance parameters, characterized by likelihoods $\{\mathcal{L}_C^0(\boldsymbol{\theta}), \ldots, \mathcal{L}_C^c(\boldsymbol{\theta})\}$, by considering the augmented likelihood $\hat{\mathcal{L}}'_A$:

$$\hat{\mathcal{L}}'_A(\boldsymbol{\theta}; \boldsymbol{\phi}) = \hat{\mathcal{L}}_A(\boldsymbol{\theta}; \boldsymbol{\phi}) \prod_{i=0}^{c} \mathcal{L}_C^i(\boldsymbol{\theta}) \ . \tag{3.22}$$

The loss function used to optimize the parameters of the neural network $\boldsymbol{\phi}$ can be any function of the covariance matrix at $\boldsymbol{\theta}_s$, depending on the concrete inference problem. The diagonal elements $I_{ii}^{-1}(\boldsymbol{\theta}_s)$ correspond to the expected variance for the parameter θ_i. Thus, if the aim is optimal inference about one of the parameters $\omega_0 = \theta_k$ a possible loss function is:

$$U = I_{kk}^{-1}(\boldsymbol{\theta}_s) \tag{3.23}$$

which corresponds to the approximated expected width of the confidence interval for ω_0.

3.3.2 Synthetic Example

The performance of INFERNO has been studied with a synthetic three-dimensional mixture model with two components [11] and the authors show that the summary statistics learnt with INFERNO outperform those obtained by using a classifier trained with binary cross-entropy (BCE). The studied model is defined by the following PDFs for the background:

$$f_b(\boldsymbol{x}|r, \lambda) = \mathcal{N}\left((x_0, x_1)|(2+r, 0), \begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}\right) \mathrm{Exp}(x_2|\lambda) \tag{3.24}$$

and the signal:

$$f_s(\boldsymbol{x}) = \mathcal{N}\left((x_0, x_1)|(1, 1), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \mathrm{Exp}(x_2|2) \tag{3.25}$$

such that (x_0, x_1) are distributed according to a multivariate normal distribution while x_2 is given by an exponential distribution. The signal distribution is fully specified while the background distribution depends on the parameter r, that shifts the mean of the background density, and a parameter λ that specifies the rate of the exponential distribution. A common model for physics analyses at the LHC, where the expected number of observations can be inferred from simulations, is given by:

$$p(\boldsymbol{x}|s,r,\lambda,b) = \frac{b}{s+b} f_b(\boldsymbol{x}|r,\lambda) + \frac{s}{s+b} f_s(\boldsymbol{x}) \tag{3.26}$$

where s and b correspond to the expected number of signal and background events. The performance of INFERNO is compared to a neural network trained with binary cross-entropy. Furthermore, the performance is also compared to the optimal inference baseline obtained from the analytical extended likelihood:

$$\mathcal{L}(s,r,\lambda,b) = \text{Pois}(n|s+b) \prod^{n} p(\boldsymbol{x}|s,r,\lambda,b) \tag{3.27}$$

and the performance is also compared to the optimal classifier, given by:

$$s^*(\boldsymbol{x}|r,\lambda) = \frac{f_s(\boldsymbol{x})}{f_s(\boldsymbol{x}) + f_b(\boldsymbol{x}|r,\lambda)} \ . \tag{3.28}$$

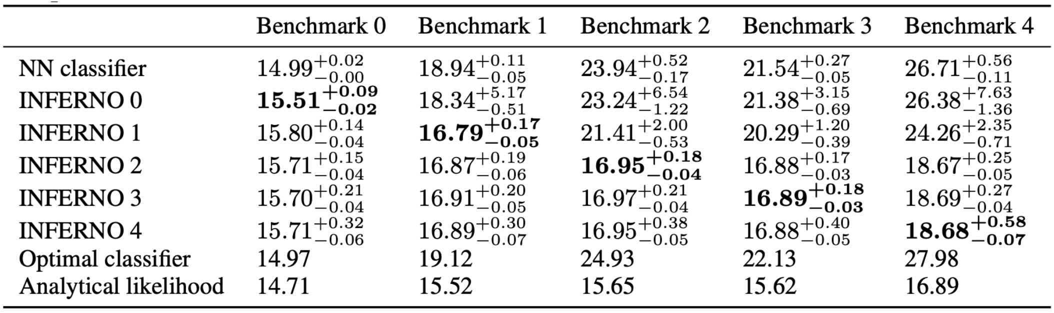
As described in Section 3.1.1, s^* is a sufficient summary statistic for a two-component mixture model if the only unknown parameter is the signal mixture fraction.

In the INFERNO paper the authors consider five inference benchmarks with $s = 50$ signal events and $b = 1000$ background events, that vary in the number of nuisance parameters and their constraints:

- Benchmark 0: no nuisance parameters are considered, both signal and background distributions are taken as fully specified ($r = 0.0, \lambda = 3.0$ and $b = 1000$).
- Benchmark 1: r is considered as an unconstrained nuisance parameter, while $\lambda = 3.0$ and $b = 1000$ are fixed.
- Benchmark 2: r and λ are considered as unconstrained nuisance parameters, while $b = 1000$ is fixed.
- Benchmark 3: r and λ are considered as nuisance parameters but with the following constraints: $\mathcal{N}(r|0.0, 0.4)$ and $\mathcal{N}(\lambda|3.0, 1.0)$, while $b = 1000$ is fixed.
- Benchmark 4: all r, λ and b are all considered as nuisance parameters with the following constraints: $\mathcal{N}(r|0.0, 0.4), \mathcal{N}(\lambda|3.0, 1.0)$ and $\mathcal{N}(b|1000, 100)$.

In the training of INFERNO, the expected number of signal and background events, s and b are included in the computation graph by scaling the Poisson counts of the signal and background observations. The effect of the nuisance parameters r and λ is modelled as an analytical transformation of the input data.

200,000 events have been considered for the training, while 1,000,000 events are used for evaluation. The same network architecture is used both for cross-entropy and inference-aware training: two hidden layers of 100 nodes followed by ReLU activation functions. A temperature of $\tau = 0.1$ has been used for the inference-aware models, which were trained during 200 epochs with SGD using mini-batches of 2000 observations and a learning rate $\gamma = 10^{-6}$. The models based on a cross-entropy loss were trained during 200 epochs using a mini-batch size of 64 and a fixed learning rate of $\gamma = 0.001$. The results of the study are provided in Fig. 3.3. The median and 1-σ percentiles on the expected uncertainty on s are reported for 100 random-initialized instances of each model and the optimal classifier and likelihood-based inference are included for comparison. The conclusion of the study is that confidence intervals obtained using INFERNO-based summary statistics are more precise compared to those obtained with binary classification and tend to be closer to the confidence intervals expected when using the true likelihood for inference. The improvement over binary classification increases when more nuisance parameters are considered. The authors also show for Benchmark 2 that the inference-aware summary statistics learnt for θ_s work well if the value used during the training deviates from the true value of θ_s.



	Benchmark 0	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
NN classifier	$14.99^{+0.02}_{-0.00}$	$18.94^{+0.11}_{-0.05}$	$23.94^{+0.52}_{-0.17}$	$21.54^{+0.27}_{-0.05}$	$26.71^{+0.56}_{-0.11}$
INFERNO 0	$\mathbf{15.51^{+0.09}_{-0.02}}$	$18.34^{+5.17}_{-0.51}$	$23.24^{+6.54}_{-1.22}$	$21.38^{+3.15}_{-0.69}$	$26.38^{+7.63}_{-1.36}$
INFERNO 1	$15.80^{+0.14}_{-0.04}$	$\mathbf{16.79^{+0.17}_{-0.05}}$	$21.41^{+2.00}_{-0.53}$	$20.29^{+1.20}_{-0.39}$	$24.26^{+2.35}_{-0.71}$
INFERNO 2	$15.71^{+0.15}_{-0.04}$	$16.87^{+0.19}_{-0.06}$	$\mathbf{16.95^{+0.18}_{-0.04}}$	$16.88^{+0.17}_{-0.03}$	$18.67^{+0.25}_{-0.05}$
INFERNO 3	$15.70^{+0.21}_{-0.04}$	$16.91^{+0.20}_{-0.05}$	$16.97^{+0.21}_{-0.04}$	$\mathbf{16.89^{+0.18}_{-0.03}}$	$18.69^{+0.27}_{-0.04}$
INFERNO 4	$15.71^{+0.32}_{-0.06}$	$16.89^{+0.30}_{-0.07}$	$16.95^{+0.38}_{-0.05}$	$16.88^{+0.40}_{-0.05}$	$\mathbf{18.68^{+0.58}_{-0.07}}$
Optimal classifier	14.97	19.12	24.93	22.13	27.98
Analytical likelihood	14.71	15.52	15.65	15.62	16.89

Figure 3.3: Summary of the results for the benchmarks defined in the text. Details can be found in [11].

3.3.3 Novel Developments and Related Work

Charnock et al. [85] propose a machine learning technique, called information maximizing neural networks, that aims at finding non-linear functionals of the data that maximize the Fisher information. By design, this approach will find transformations that are minimally affected by nuisance parameters while being maximally sensitive to the parameters of interest. Related to this work, Alsing et al. [86] have developed a transformation that can be applied to marginalize the summary statistics resulting from information maximizing neural networks.

Additionally, alternative approaches exist that may be useful in specific situations. In [87] a variation of boosted decision trees is developed, referred to as QBDT. The loss is based on the statistical significance, and can also include the effect of nuisance parameters in its approximation.
In [88] the expected significance approximation formula for a single-bin counting experiment that can include the effect of a single source of systematic uncertainty is used as a loss function of a neural network. Related to this work, in [89] the authors use an approximation of the Punzi figure of merit as a loss function. The Punzi loss is used in the search for new physics and maximizes the inverse of the minimum detectable cross-section, which defines a sensitivity region for which the experiment will certainly give conclusive results.
More recently, the authors of [90] proposed a custom loss function for the search of new physics where the effect of different nuisance parameters is included with a polynomial approximation.

While INFERNO focuses on uncertainty-aware loss functions, there is a complementary line of research that focuses on the profiling aspect of uncertainty awareness. The authors in [91] train a classifier that is fully aware of uncertainties and their corresponding nuisance parameters. Specifically, the nuisance parameter z is treated as a feature alongside the observed data x, such that the network learns a decision function which varies with the nuisance parameter, allowing for later profiling over the nuisance parameters during inference. The method is studied using a synthetic Gaussian dataset, as well as the Kaggle Higgs Challenge [66]. For both cases the authors show that the uncertainty-aware approach can achieve a better sensitivity than alternative machine learning strategies.

More recently, there has also been some work that is based on the ideas of INFERNO.

Wunsch et al. [92] approximate the bins of a histogram with Gaussian functions and use a neural network with a sigmoid function in the last layer as the basis for the inference-aware loss to construct a Poisson count likelihood. The authors study the approach with a synthetic example and the Higgs ML dataset including nuisance parameters. The authors of NEOS [93, 94] implement a differentiable version of kernel-density estimation to construct a differentiable summary statistic and compute the gradients of the profile likelihood with a technique referred to as fixed-point differentiation that directly minimizes the expected upper CLs limits.
Moreover INFERNO has been rewritten in TensorFlow 2 and a PyTorch version PyTorch_Inferno [95] has been developed by G. Strong, that besides analytically parametrizing the effects of nuisance parameters, allows to use a differentiable interpolation algorithm, which is important for typical HEP analysis. Details of this implementation will be further discussed in Chapter 7, where the INFERNO algorithm will be applied to a realistic CMS measurement of the $\mathrm{t\bar{t}}$ production cross-section in the τ+jets channel. The following two chapters will provide an overview over the Standard Model and the CMS detector that are essential for this analysis.

4 The Standard Model and Top Physics at the LHC

Following the introduction of the INFERNO algorithm, this chapter will provide an overview over the Standard Model of elementary particles (SM) that is studied in detail at the LHC, and is of particular relevance for the $\mathrm{t\bar{t}}$ cross-section measurement discussed in Chapter 6.

The Standard Model is one of the great triumphs of modern physics. It is a Quantum Field Theory that describes three of the four known fundamental interactions in our world. Its development started in the early 1970s with the formulation of a unified theory of the electromagnetic and weak interactions by Glashow, Salam and Weinberg [96]. The theory became widely accepted by the discovery of neutral weak currents at CERN in 1973 [97]. Various precision measurements and discoveries have established the validity of the Standard Model at energies up to the electroweak scale. The discovery of the Higgs boson by the ATLAS and CMS collaborations in 2012 completed the spectrum of particles in the Standard Model [98, 99]. Despite its great success, the Standard Model cannot explain several physical phenomena, therefore it is not a complete theory of fundamental interactions. One of the most promising avenues to discover new physics that can lead to the development of a complete theory of fundamental interactions is the study of top physics at particle colliders, such as the Large Hadron Collider (LHC) in Geneva. This chapter will provide an overview over the mathematical foundations of the Standard Model, as well as its limitations. It will then introduce the physics of proton-proton collisions and give an overview over the top physics program of the CMS experiment at the LHC.

4.1 The Standard Model

Mathematically the Standard Model can be specified as a Quantum Field Theory with gauge group $\mathrm{SU}(3) \times \mathrm{SU}(2) \times \mathrm{U}(1)$ with 15 left-handed spinor fields in three copies of the representation $\left(1, 2, -\frac{1}{2}\right) \oplus (1, 1, +1) \oplus \left(3, 2, +\frac{1}{6}\right) \oplus \left(\overline{3}, 1, -\frac{2}{3}\right) \oplus \left(\overline{3}, 1, +\frac{1}{3}\right)$. The first entry of each triplet refers to the representation of the group SU(3), the second entry to the representation of SU(2) and the last entry corresponds to the value of the U(1) hypercharge. The eight gauge bosons that mediate the strong force are called gluons and are associated to the generators of the group SU(3). The Higgs mechanism induces a mixing of the gauge bosons that couple to the generators of SU(2) and U(1). A linear combination of the bosons $A_\mu^{1,2}$, associated with the group SU(2), form the two physically observable gauge bosons W^+ and W^-. The photon, that couples to the electric charge, is a linear combination of

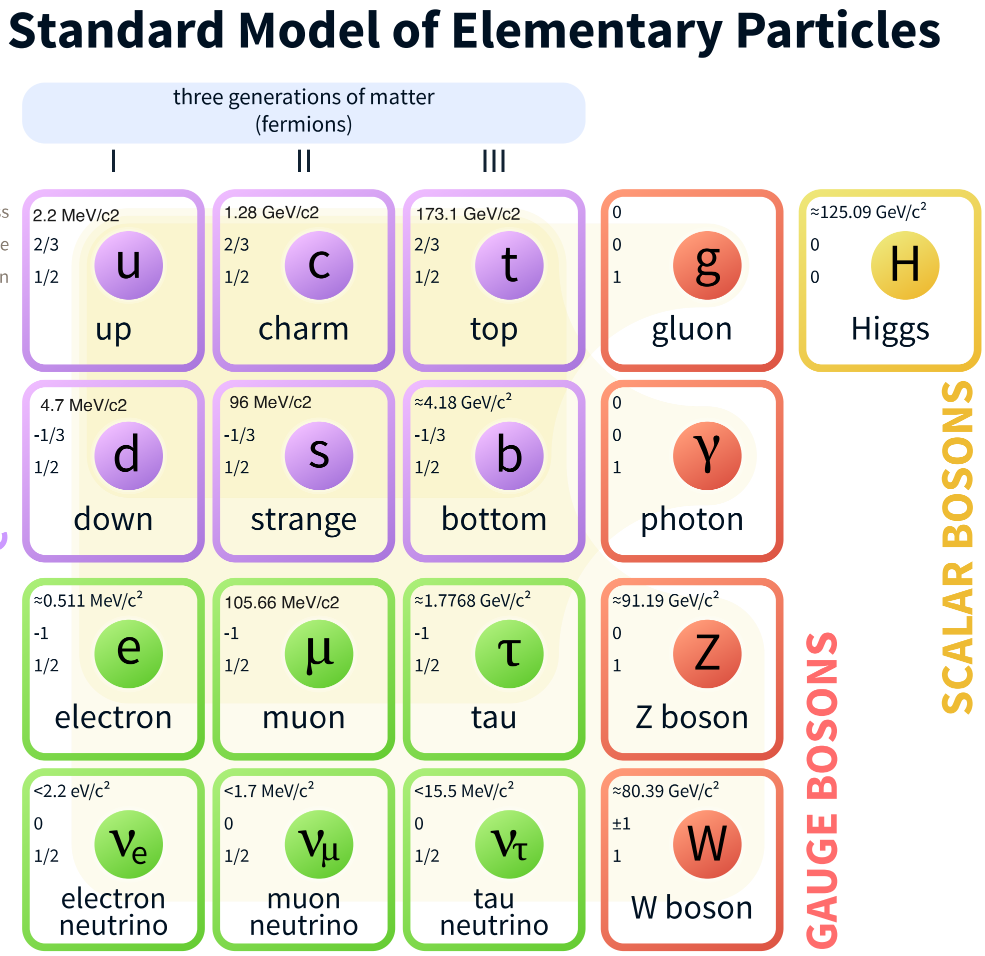


Figure 4.1: Overview of the particle content within the Standard Model of particle physics. Diagram adapted under licence CC BY 4.0.

the third boson of SU(2), A^3_μ and the boson B_μ that is associated with the group U(1). The linear combination orthogonal to the photon corresponds to the Z boson. The spinor fields in the SU(3) × SU(2) × U(1) representation $\left(3, 2, +\frac{1}{6}\right)$ correspond to the up- and down-quark, while their conjugate right-handed partners are in the representation $\left(\overline{3}, 1, -\frac{2}{3}\right)$ and $\left(\overline{3}, 1, +\frac{1}{3}\right)$. The electron and neutrino are in the representation $\left(1, 2, -\frac{1}{2}\right)$, while the conjugate right-handed partner of the electron is in the representation $(1, 1, +1)$ and the neutrino does not have a conjugate right-handed partner. The fact that the up- and down-quarks, as well as the neutrino and electron, transform as a doublet under SU(2), while their conjugate partners do not, implies that the theory is parity violating. The masses of the elementary particles are explained by the Higgs mechanism. The Standard Model Lagrangian $\mathcal{L}$ contains all terms that are allowed by the gauge symmetries and Lorentz invariance and are of mass dimension four or less. The action of a quantum field ψ is defined as:

$$S = \int \mathcal{L}(\psi, \partial_\mu\psi)\ d^4x \tag{4.1}$$

and the equations of motion for the field ψ can be obtained with the action principle, which requires the action integral to be stationary under small perturbations, i.e. $\delta S = 0$, yielding the Euler-Lagrange equations:

$$\partial_\mu \left(\frac{\partial \mathcal{L}}{\partial(\partial_\mu \psi)} \right) - \frac{\partial \mathcal{L}}{\partial \psi} = 0\ . \tag{4.2}$$

The important Noether theorem states that every continuous global symmetry of the action leads to a conserved current J^μ:

$$\partial_\mu J^\mu = 0 \tag{4.3}$$

which implies a conserved charge for solutions of the equations of motion. The description of the Standard Model in this chapter is based on the standard literature [100] and [101], which is also recommended for a more detailed review on Quantum Field Theory and the Standard Model.

4.1.1 Gauge and Higgs Sector

The electroweak part of the gauge group, SU(2) × U(1) with the complex scalar Higgs field φ in the representation $\left(2, -\frac{1}{2}\right)$ gives rise to the electromagnetic and the weak force and provides an explanation for the masses of the massive $W^{\pm}$ and Z vector bosons. The covariant derivative of the Higgs field φ is given by:

$$\left(D_\mu \varphi\right)_i = \partial_\mu \varphi_i - i\Big[g_2 A_\mu^a T^a + g_1 B_\mu Y\Big]_i^j \varphi_j \; . \tag{4.4}$$

with $T^a = \frac{1}{2}\sigma^a$ where σ^a are the Pauli matrices, g_2 and g_1 the coupling constants of SU(2) and U(1), and Y is the hypercharge generator. The form of the Higgs potential $V(\varphi)$, which is displayed in Fig. 4.2, is chosen such that it provides degenerated vacuum states and a local maximum:

$$V(\varphi) = \frac{1}{4}\lambda\left(\varphi^\dagger\varphi - \frac{1}{2}v^2\right)^2 \tag{4.5}$$

where λ is a positive constant. As a result, the Higgs field acquires a nonzero vacuum expectation value v that spontaneously breaks the group SU(2) × U(1) to U(1). It is possible to make a global gauge transformation to bring the vacuum expectation value in the first component and make it real:

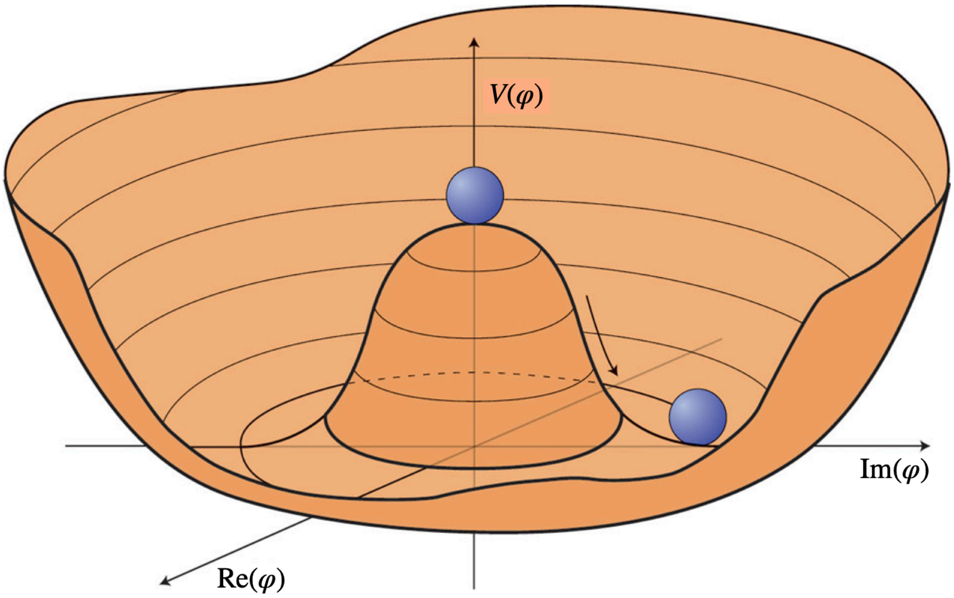


Figure 4.2: Sketch of the Higgs potential $V(\varphi)$. Selecting one of the points at the bottom of the potential breaks spontaneously the rotational U(1) symmetry [102].

$$\langle 0|\varphi(x)|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} v \\ 0 \end{pmatrix} \tag{4.6}$$

and by replacing φ with its vacuum expectation value and inserting this relation in the kinetic term of the Higgs field φ, which is given by $-(D^\mu\varphi)^\dagger D_\mu\varphi$, the mass-squared matrix for the gauge fields can be identified as:

$$\mathcal{L}_{\text{mass}} = -\frac{1}{8}v^2(1,0)\begin{pmatrix} g_2 A_\mu^3 - g_1 B_\mu & g_2\left(A_\mu^1 - iA_\mu^2\right) \\ g_2\left(A_\mu^1 + iA_\mu^2\right) & -g_2 A_\mu^3 - g_1 B_\mu \end{pmatrix}^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{4.7}$$

This matrix can be diagonalized by defining the weak mixing angle:

$$\theta_{\text{W}} \equiv \tan^{-1}(g_1/g_2) \tag{4.8}$$

and the physical gauge boson fields, that correspond to the normalized eigenvectors of the mass-squared matrix, are then defined as:

$$W_\mu^\pm \equiv \frac{1}{\sqrt{2}}\left(A_\mu^1 \mp iA_\mu^2\right) \tag{4.9}$$

$$Z_\mu \equiv c_{\text{W}} A_\mu^3 - s_{\text{W}} B_\mu \tag{4.10}$$

$$A_\mu \equiv s_{\text{W}} A_\mu^3 + c_{\text{W}} B_\mu \tag{4.11}$$

where $s_{\text{W}} \equiv \sin\theta_{\text{W}}$ and $c_{\text{W}} \equiv \cos\theta_{\text{W}}$. The physical fields are mixtures of the massless bosons associated with the U(1) and SU(2) local gauge symmetries. In terms of these fields, equation 4.7 can be written as:

$$\mathcal{L}_{\text{mass}} = -M_{\text{W}}^2 W^{+\mu} W_\mu^- - \frac{1}{2} M_{\text{Z}}^2 Z^\mu Z_\mu \tag{4.12}$$

where $M_{\text{W}} = g_2 v/2$ and $M_{\text{Z}} = M_{\text{W}}/\cos\theta_{\text{W}}$. The mass of the W boson is determined by the coupling constant of the SU(2) gauge interaction g_2 and the vacuum expectation value of the Higgs field. The Z boson, which is associated with the neutral Goldstone boson of the broken symmetry, has acquired mass through the Higgs mechanism, while the field A_μ that corresponds to the photon remains massless. This means that there is an unbroken U(1) subgroup that can be identified with the gauge group of electromagnetism. The two complex components of the Higgs field φ result in four real scalar fields. After the spontaneous breaking of the symmetry, three of the scalar fields give the longitudinal degrees of freedom of the $W^\pm$ and Z

bosons, while the remaining scalar field accounts for shifts in the overall scale of φ. Therefore, the Higgs field in unitary gauge can be written as:

$$\varphi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} v + H(x) \\ 0 \end{pmatrix} \tag{4.13}$$

where H is a real scalar field. The particle corresponding to the field H is the Higgs boson. With this the potential becomes:

$$V(\varphi) = \frac{1}{4}\lambda v^2 H^2 + \frac{1}{4}\lambda v H^3 + \frac{1}{16}\lambda H^4 \tag{4.14}$$

and the mass of the Higgs boson can be identified as $m_{\mathrm{H}}^2 = \frac{1}{2}\lambda v^2$. The kinetic term for H can be written as $-\frac{1}{2}\partial^\mu H \partial_\mu H$, while the kinetic terms for the gauge fields are given by:

$$\mathcal{L} = -\frac{1}{4} F^{a\mu\nu} F^a_{\mu\nu} - \frac{1}{4} B^{\mu\nu} B_{\mu\nu} \tag{4.15}$$

where

$$\begin{aligned} F^i_{\mu\nu} &= \partial_\mu A^i_\nu - \partial_\nu A^i_\mu + g_2 \epsilon^{ijk} A^j_\mu A^k_\nu \\ B_{\mu\nu} &\equiv \partial_\mu B_\nu - \partial_\nu B_\mu \ . \end{aligned} \tag{4.16}$$

By forming the combinations $F^1_{\mu\nu} \pm iF^2_{\mu\nu}$ and using equation 4.8, one can define a covariant derivative that acts on W^+_μ:

$$D_\mu \equiv \partial_\mu - ig_2 A^3_\mu = \partial_\mu - ig_2(s_{\mathrm{W}} A_\mu + c_{\mathrm{W}} Z_\mu) \tag{4.17}$$

and A_μ can be identified as the electromagnetic vector potential. By assigning an electric charge $Q = +1$ to the W^+ boson, then it follows from equation 4.17 that the electromagnetic coupling constant e is identified as:

$$e = g_2 \sin\theta_{\mathrm{W}} \ . \tag{4.18}$$

All of this can be assembled into the complete Lagrangian for the electroweak gauge fields and the Higgs boson in unitary gauge with the electromagnetic field strength $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$, and $Z_{\mu\nu} \equiv \partial_\mu Z_\nu - \partial_\nu Z_\mu$ which finally gives:

$$\begin{aligned}\mathcal{L} = &-\frac{1}{4}F^{\mu\nu}F_{\mu\nu} - \frac{1}{4}Z^{\mu\nu}Z_{\mu\nu} - D^{\dagger\mu}W^{-\nu}D_\mu W^+_\nu + D^{\dagger\mu}W^{-\nu}D_\nu W^+_\mu \\ &+ ie(F^{\mu\nu} + \cot\theta_{\mathrm{W}} Z^{\mu\nu})W^+_\mu W^-_\nu \\ &- \frac{1}{2}\left(e^2/\sin^2\theta_{\mathrm{w}}\right)\left(W^{+\mu}W^-_\mu W^{+\nu}W^-_\nu - W^{+\mu}W^+_\mu W^{-\nu}W^-_\nu\right) \\ &- \left(M^2_{\mathrm{w}}W^{+\mu}W^-_\mu + \frac{1}{2}M^2_{\mathrm{Z}}Z^\mu Z_\mu\right)\left(1 + v^{-1}H\right)^2 \\ &- \frac{1}{2}\partial^\mu H\partial_\mu H - \frac{1}{2}m^2_{\mathrm{H}}H^2 - \frac{1}{2}m^2_{\mathrm{H}}v^{-1}H^3 - \frac{1}{8}m^2_{\mathrm{H}}v^{-2}H^4 \;.\end{aligned} \tag{4.19}$$

Many properties of electroweak phenomena were verified by experiments, including the discovery of the massive $W^\pm$ [103, 104] and Z bosons [105, 106], and the existence of weakly-interacting neutral and charged currents [97]. The observed masses of the $W^\pm$ and Z bosons are $M_{\mathrm{W}} = 80.4$ GeV and $M_{\mathrm{Z}} = 91.2$ GeV [9] which implies $\cos\theta_{\mathrm{W}} = 0.882$. In 2012 the CMS and ATLAS collaborations discovered a particle with a mass of 125 GeV consistent with the expected properties for the Higgs boson [98, 99].

4.1.2 Lepton Sector

Leptons are particles with a spin value of one-half. Since they are singlets of the color group SU(3), they are not affected by the strong force. There are three generations of leptons: e and ν_e, μ and ν_μ, τ and ν_τ. The electron and its neutrino can be described by left-handed spinor fields ℓ and $\bar{e}$ in the representations $\left(2, -\frac{1}{2}\right)$ and $(1, +1)$ of $\mathrm{SU}(2) \times \mathrm{U}(1)$. The SU(2) components of ℓ can be written as:

$$\ell = \begin{pmatrix} \nu \\ e \end{pmatrix} . \tag{4.20}$$

Since only the left-handed spinor fields of ℓ transform as a doublet under SU(2) the gauge theory is parity violating. The covariant derivatives of the lepton fields are given by:

$$\begin{aligned}(D_\mu\ell)_i &= \partial_\mu\ell_i - ig_2A^a_\mu(T^a)_i{}^j\ell_j - ig_1\left(-\frac{1}{2}\right)B_\mu\ell_i \\ D_\mu\bar{e} &= \partial_\mu\bar{e} - ig_1(+1)B_\mu\bar{e}\end{aligned} \tag{4.21}$$

and their kinetic terms are:

$$\mathcal{L}_{\text{kin}} = i\ell^{\dagger i}\bar{\sigma}^{\mu}(D_{\mu}\ell)_i + i\bar{e}^{\dagger}\bar{\sigma}^{\mu}D_{\mu}\bar{e} \ . \tag{4.22}$$

The Higgs mechanism does not only generate the masses of the gauge bosons, but is also responsible for the masses of the leptons. Since the left- and right-handed fields transform differently under SU(2), a mass term involving only ℓ and/or $\bar{e}$ does not respect the gauge symmetry of SU(2) × U(1). However, it is possible to introduce mass terms for fermions with a Yukawa coupling that respects the gauge symmetry and takes the form:

$$\mathcal{L}_{\text{Yuk}} \ = -y\varepsilon^{ij}\varphi_i\ell_j\bar{e} + \text{h.c.} \tag{4.23}$$

where h.c. refers to the hermitian conjugate, φ is the Higgs field in the $\left(2, -\frac{1}{2}\right)$ representation, and y is the Yukawa coupling constant. In unitary gauge this equation becomes:

$$\begin{aligned}\mathcal{L}_{\text{Yuk}} &= -\frac{1}{\sqrt{2}}y(v+H)\left(e\bar{e} + \bar{e}^{\dagger}e^{\dagger}\right)\\ &= -\frac{1}{\sqrt{2}}y(v+H)\overline{\mathcal{E}}\mathcal{E}\end{aligned} \tag{4.24}$$

with the definition of a Dirac field for the electron and a Majorana field $\mathcal{N}_{\text{L}}$ for the neutrino:

$$\mathcal{E} \equiv \begin{pmatrix} e \\ \bar{e}^{\dagger} \end{pmatrix}, \quad \mathcal{N}_{\text{L}} \equiv P_{\text{L}}\mathcal{N} = \begin{pmatrix} \nu \\ 0 \end{pmatrix} \tag{4.25}$$

with $P_{\text{L}} \ = \ \frac{1}{2}(1-\gamma_5)$. From equation 4.24 it can be seen that the electron has acquired the mass $m_e = yv/\sqrt{2}$, while the neutrino is still massless. By expressing the covariant derivatives in equation 4.21 with the fields $W_{\mu}^{\pm}, Z_{\mu}$, and A_{μ}, one can identify the generator of the electric charge:

$$Q = T^3 + Y \tag{4.26}$$

and from the definitions of the generators T^3 and Y the following relations can be obtained:

$$\begin{aligned} T^3\nu = +\frac{1}{2}\nu, \quad T^3 e = -\frac{1}{2}e, \quad T^3\bar{e} = 0\\ Y\nu = -\frac{1}{2}\nu, \quad Ye = -\frac{1}{2}e, \quad Y\bar{e} = +\bar{e}\end{aligned} \tag{4.27}$$

and the values of the charge of the spinor fields are given by:

$$Q\nu = 0, \quad Qe = -e, \quad Q\bar{e} = +\bar{e} \tag{4.28}$$

which are the electric charges that are expected for the electron and the neutrino. In terms of the Dirac fields one finds the following relations for the couplings of the gauge boson fields to the leptons:

$$\mathcal{L}_{\text{int}} = \frac{1}{\sqrt{2}} g_2 W_\mu^+ J^{-\mu} + \frac{1}{\sqrt{2}} g_2 W_\mu^- J^{+\mu} + \frac{e}{s_{\text{W}} c_{\text{W}}} Z_\mu J_{\text{Z}}^\mu + e A_\mu J_{\text{EM}}^\mu \tag{4.29}$$

with the definition of the conserved charged and neutral currents:

$$\begin{aligned} J^{+\mu} &\equiv \overline{\mathcal{E}}_{\text{L}} \gamma^\mu \mathcal{N}_{\text{L}} \\ J^{-\mu} &\equiv \overline{\mathcal{N}}_{\text{L}} \gamma^\mu \mathcal{E}_{\text{L}} \\ J_{\text{Z}}^\mu &\equiv J_3^\mu - s_{\text{w}}^2 J_{\text{EM}}^\mu \\ J_3^\mu &\equiv \frac{1}{2} \overline{\mathcal{N}}_{\text{L}} \gamma^\mu \mathcal{N}_{\text{L}} - \frac{1}{2} \overline{\mathcal{E}}_{\text{L}} \gamma^\mu \mathcal{E}_{\text{L}} \\ J_{\text{EM}}^\mu &\equiv -\overline{\mathcal{E}} \gamma^\mu \mathcal{E} \ . \end{aligned} \tag{4.30}$$

From equation 4.29, equation 4.19, equation 4.22 and equation 4.24 the Lagrangian of Quantum Electrodynamics for the electron Dirac spinor $\mathcal{E}$ and the photon field A_μ can be identified:

$$\mathcal{L}_{\text{QED}} = i \overline{\mathcal{E}} \gamma^\mu \partial_\mu \mathcal{E} - \frac{1}{4} F^{\mu\nu} F_{\mu\nu} - m_e \overline{\mathcal{E}} \mathcal{E} + e A_\mu J_{\text{EM}}^\mu \tag{4.31}$$

where γ^μ are the Dirac matrices. Applying the Euler-Lagrange equations, as defined in equation 4.2, with respect to the field of the photon A_μ yields the Maxwell equations that form the foundation of classical electromagnetism:

$$\partial_\mu F^{\mu\nu} = e J_{\text{EM}}^\mu \tag{4.32}$$

and applying the Euler-Lagrange equations with respect to the field $\mathcal{E}$ yields:

$$(i \gamma^\mu \partial_\mu - m_e) \mathcal{E} = e \gamma^\mu A_\mu \mathcal{E} \tag{4.33}$$

which is the original Dirac equation on the left side and the interaction with the photon field A_μ on the right side. It can further be shown that the exchange of the W boson generates the Fermi weak interaction [107] that was developed to describe the β-decay and can be interpreted as a low-energy Effective Field Theory of the SM. The Fermi constant can be defined as:

$$G_{\text{F}} \equiv \frac{e^2}{4\sqrt{2} \sin^2 \theta_{\text{w}} M_{\text{w}}^2} \tag{4.34}$$

which corresponds to the historical definition of the Fermi coupling.

4.1.3 Quark Sector

Quarks are spin-one-half particles that transform under the color group SU(3). The Quantum Field Theory of the strong interaction associated with the group SU(3) is also referred to as Quantum Chromodynamics(QCD). There are three families of quarks: up u and down d, charm c and strange s, top t and beauty (or bottom) b. A single quark family can be described by left-handed spinor fields q, $\bar{u}$ and $\bar{d}$ in the representation $\left(3, 2, +\frac{1}{6}\right)$, $\left(\overline{3}, 1, -\frac{2}{3}\right)$, and $\left(\overline{3}, 1, +\frac{1}{3}\right)$ of SU(3) × SU(2) × U(1). The SU(2) components of q can be written as:

$$q = \begin{pmatrix} u \\ d \end{pmatrix} . \tag{4.35}$$

As in the case of the leptons, only the left-handed spinor fields of q transform as a doublet under SU(2) implying that the theory is parity violating. Experimentally, parity violation was first discovered in the Wu experiment in 1956 [108]. The covariant derivatives of the quark fields are given by:

$$\begin{aligned} (D_\mu q)_{\alpha i} &= \partial_\mu q_{\alpha i} - i g_3 A^a_\mu (T^a_3)^\beta_\alpha q_{\beta i} - i g_2 A^a_\mu (T^a_2)^j_i q_{\beta j} - i g_1 \left(+\tfrac{1}{6}\right) B_\mu q_{\alpha i} \\ (D_\mu \bar{u})^\alpha &= \partial_\mu \bar{u}^\alpha - i g_3 A^a_\mu (T^a_3)^\alpha{}_\beta \bar{u}^\beta - i g_1 \left(-\tfrac{2}{3}\right) B_\mu \bar{u}^\alpha \\ \left(D_\mu \bar{d}\right)^\alpha &= \partial_\mu \bar{d}^\alpha - i g_3 A^a_\mu (T^a_3)^\alpha{}_\beta \bar{d}^\beta - i g_1 \left(+\tfrac{1}{3}\right) B_\mu \bar{d}^\alpha \end{aligned} \tag{4.36}$$

and the kinetic terms can be expressed as:

$$\mathcal{L}_{\text{kin}} = i q^{\dagger \alpha i} \bar{\sigma}^\mu (D_\mu q)_{\alpha i} + i \bar{u}^\dagger_\alpha \bar{\sigma}^\mu (D_\mu \bar{u})^\alpha + i \bar{d}^\dagger_\alpha \bar{\sigma}^\mu \left(D_\mu \bar{d}\right)^\alpha . \tag{4.37}$$

As in the case of the leptons, a mass term can be generated via Yukawa couplings:

$$\mathcal{L}_{\text{Yuk}} = -y' \varepsilon^{ij} \varphi_i q_{\alpha j} \bar{d}^\alpha - y'' \varphi^{\dagger i} q_{\alpha i} \bar{u}^\alpha + \text{h.c.} \tag{4.38}$$

where φ is the Higgs field in the $\left(1, 2, -\frac{1}{2}\right)$ representation, and y' and y'' are the Yukawa coupling constants. In unitary gauge this becomes:

$$\begin{aligned} \mathcal{L}_{\text{Yuk}} &= -\frac{1}{\sqrt{2}} y'(v + H)\left(d_\alpha \bar{d}^\alpha + \bar{d}^\dagger_\alpha d^{\dagger \alpha}\right) - \frac{1}{\sqrt{2}} y''(v + H)\left(u_\alpha \bar{u}^\alpha + \bar{u}^\dagger_\alpha u^{\dagger \alpha}\right) \\ &= -\frac{1}{\sqrt{2}} y'(v + H) \overline{\mathcal{D}}^\alpha \mathcal{D}_\alpha - \frac{1}{\sqrt{2}} y''(v + H) \overline{\mathcal{U}}^\alpha \mathcal{U}_\alpha \end{aligned} \tag{4.39}$$

with the definition of the Dirac fields for the down- and up-quarks:

$$\mathcal{D}_\alpha \equiv \begin{pmatrix} d_\alpha \\ \bar{d}^\dagger_\alpha \end{pmatrix}, \quad \mathcal{U}_\alpha \equiv \begin{pmatrix} u_\alpha \\ \bar{u}^\dagger_\alpha \end{pmatrix} . \tag{4.40}$$

From equation 4.39 it follows that the up-quarks have acquired the mass $m_u = y''v/\sqrt{2}$ and the down-quarks have acquired the mass $m_d = y'v/\sqrt{2}$. With the definition of the generator of the electric charge $Q = T^3 + Y$ the following charges are assigned to the up- and down-quarks:

$$Qu = +\frac{2}{3}u, \quad Qd = -\frac{1}{3}d, \quad Q\bar{u} = -\frac{2}{3}\bar{u}, \quad Q\bar{d} = +\frac{1}{3}\bar{d} \tag{4.41}$$

which are the charges that are expected for the up- and down-quarks. By expressing the covariant derivatives in equation 4.36 with the fields $W^\pm_\mu, Z_\mu$, and A_μ and using the definitions of the four-component fields, the following couplings of the electroweak gauge fields to the quarks can be found:

$$\mathcal{L}_{\text{int}} = \frac{1}{\sqrt{2}} g_2 W^+_\mu J^{-\mu} + \frac{1}{\sqrt{2}} g_2 W^-_\mu J^{+\mu} + \frac{e}{s_\text{W} c_\text{W}} Z_\mu J^\mu_\text{Z} + eA_\mu J^\mu_\text{EM} \tag{4.42}$$

where the conserved currents have been defined as:

$$\begin{aligned} J^{+\mu} &\equiv \overline{\mathcal{D}}_\text{L}\gamma^\mu\mathcal{U}_\text{L} \\ J^{-\mu} &\equiv \overline{\mathcal{U}}_\text{L}\gamma^\mu\mathcal{D}_\text{L} \\ J^\mu_\text{Z} &\equiv J^\mu_3 - s^2_\text{W} J^\mu_\text{EM} \\ J^\mu_3 &\equiv \frac{1}{2}\overline{\mathcal{U}}_\text{L}\gamma^\mu\mathcal{U}_\text{L} - \frac{1}{2}\overline{\mathcal{D}}_\text{L}\gamma^\mu\mathcal{D}_\text{L} \\ J^\mu_\text{EM} &\equiv +\frac{2}{3}\overline{\mathcal{U}}\gamma^\mu\mathcal{U} - \frac{1}{3}\overline{\mathcal{D}}\gamma^\mu\mathcal{D} \ . \end{aligned} \tag{4.43}$$

The Yukawa couplings can be generalized for multiple quark generations by defining the fields $q_{\alpha I}, \bar{u}_I$, and $\bar{d}_I$, where $I = 1, 2, 3$ denotes the generation index. The kinetic term for these fields is given by:

$$\mathcal{L}_{\text{kin}} = iq^{\dagger\alpha i I}\bar{\sigma}^\mu (D_\mu)^{\beta j}_{\alpha i} q_{\beta j I} + i\bar{u}^\dagger_{\alpha I}\bar{\sigma}^\mu (D_\mu)^\alpha{}_\beta \bar{u}^\beta_I + i\bar{d}^\dagger_{\alpha I}\bar{\sigma}^\mu (D_\mu)^\alpha{}_\beta \bar{d}^\beta_I \tag{4.44}$$

where the sum runs over the generation index I. The most general Yukawa term that is possible in unitary gauge can be written as:

$$\mathcal{L}_{\text{Yuk}} = -\frac{1}{\sqrt{2}}(v+H) d_{\alpha I} y'_{IJ} \bar{d}^\alpha_J - \frac{1}{\sqrt{2}}(v+H) u_{\alpha I} y''_{IJ} \bar{u}^\alpha_J + \text{ h.c.} \tag{4.45}$$

where y'_{IJ} and y''_{IJ} are complex 3×3 matrices, and the generation indices are summed. It is possible to make transformations between the different generations of the fields:

$$\begin{aligned} d_I &\to D_{IJ} d_J, \quad \bar{d}_I \to \bar{D}_{IJ} \bar{d}_J \\ u_I &\to U_{IJ} u_J, \quad \bar{u}_I \to \bar{U}_{IJ} \bar{u}_J \end{aligned} \tag{4.46}$$

where $U, D, \bar{U}$ and $\bar{D}$ are independent unitary matrices. These matrices can be chosen such that $D^{\mathrm{T}} y' \bar{D}$ and $U^{\mathrm{T}} y'' \bar{U}$ are diagonal and contain only positive real entries y'_I and y''_I. The down-quarks $\mathcal{D}_I$ thus acquire masses $m_{d_I} = y'_I v/\sqrt{2}$, and the up-quarks $\mathcal{U}_I$ acquire masses $m_{u_I} = y''_I v/\sqrt{2}$. While in the neutral currents simply a generation index I is added to each field, the charged currents become:

$$\begin{aligned} J^{+\mu} &= \overline{\mathcal{D}}_{\mathrm{L}I} \left(V^{\dagger}\right)_{IJ} \gamma^{\mu} \mathcal{U}_{\mathrm{L}J} \\ J^{-\mu} &= \overline{\mathcal{U}}_{\mathrm{L}I} V_{IJ} \gamma^{\mu} \mathcal{D}_{\mathrm{LK}} \end{aligned} \tag{4.47}$$

where $V \equiv U^{\dagger} D$ is called the Cabibbo–Kobayashi–Maskawa (CKM) matrix. Making use of phase rotations of the fields allows to eliminate 5 out of the 9 parameters of the 3×3 unitary CKM matrix, leaving 4 free parameters: θ_1 (the Cabibbo angle), θ_2, θ_3, and δ, and V can be written as:

$$V = \begin{pmatrix} c_1 & +s_1 c_3 & +s_1 s_3 \\ -s_1 c_2 & c_1 c_2 c_3 - s_2 s_3 e^{i\delta} & c_1 c_2 s_3 + s_2 c_3 e^{i\delta} \\ -s_1 s_2 & c_1 s_2 c_3 + c_2 s_3 e^{i\delta} & c_1 s_2 s_3 - c_2 c_3 e^{i\delta} \end{pmatrix} \tag{4.48}$$

with the coefficients $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. The experimentally determined values of these angles are: $\sin\theta_{12} = 0.226$, $\sin\theta_{13} = 0.003$, $\sin\theta_{23} = 0.040$ and $\delta = 1.196$ [9]. Since the charged currents have some terms with a phase factor $e^{i\delta}$, and some without, it is implied that the weak interactions are violating the CP symmetry. CP-violation was first discovered experimentally in 1964 in the decay of neutral Kaons [109].

Experimentally, there is evidence for the existence of quarks. However, despite many experimental attempts, free quarks have not been observed directly so far. This can be explained by the hypothesis of colour confinement, which states that in the non-perturbative regime only objects with net-zero colour charge can propagate as free particles. The origin of colour confinement is believed to be the gluon–gluon self-interactions that are possible because the gluons themselves carry colour charge.

So far there is no analytic proof of colour confinement, but there is strong evidence for it from the calculations of lattice QCD [9].
Another important feature of QCD is asymptotic freedom which means that the interactions between particles become asymptotically weaker as the energy scale increases. Asymptotic freedom can be derived by calculating the beta-function describing the variation of the coupling constant under the renormalization group. Evaluating the beta-function of QCD for the strong coupling at the momentum scale μ defined as $\alpha_S(\mu) \equiv g(\mu)^2/4\pi$ yields the following relation for α_S at a scale Q:

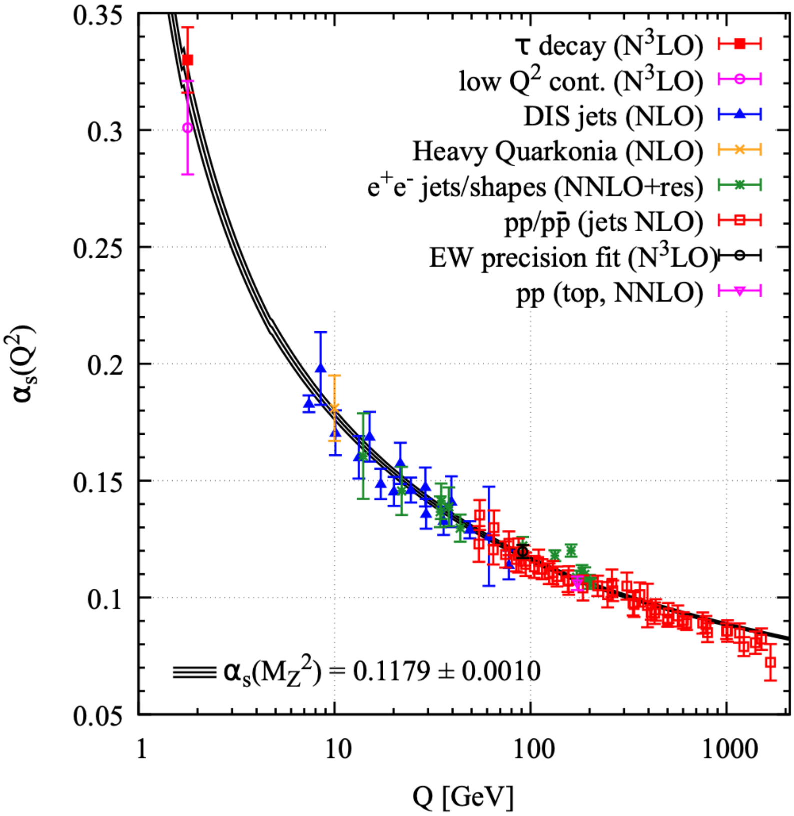


Figure 4.3: Summary of the measurements of α_S as a function of the energy scale Q. The degree of QCD perturbation theory used in the extraction is indicated in brackets [9].

$$\alpha_S(Q) = \frac{\alpha_S(\mu)}{1 + (1/4\pi)\left(11 - \frac{2}{3}n_f\right)\alpha_S(\mu)\log(Q^2/\mu^2)} \tag{4.49}$$

where n_f is the total number of quark flavours. This result shows explicitly that $\alpha_S(Q) \to 0$ logarithmically as $Q \to \infty$. Different ways exist in which α_S can be measured, including studies of the hadronic decays of the tau-lepton, the spectra of bound states of heavy quarks, measurements of deep inelastic scattering, and jet production rates in electron-positron annihilation [101]. Figure 4.3 summarizes the most important measurements of α_S [9]. As predicted, α_S decreases with increasing Q and the data are consistent with the QCD predictions for the running of α_S with a value of α_S at $Q^2 = M_Z^2$ of:

$$\alpha_s\left(M_Z^2\right) = 0.1179 \pm 0.0010 \; . \tag{4.50}$$

At LHC energies α_S is sufficiently small that perturbation theory can be used, however often higher order corrections have to be taken into account.

4.2 Beyond the Standard Model

The Standard Model is the most successful particle physics theory up to date. However, there are several phenomena that the Standard Model cannot explain. Various proposals for physics “Beyond the Standard Model” exist that modify the Standard Model in a way that they are still consistent with the existing experimental data, but are able to predict possible deviations from the Standard Model in new experiments. In this section the limitations of the SM, as well as two important proposals for extensions of the SM are discussed.

4.2.1 Limitations of the Standard Model

The Standard Model is not the ultimate theory of particle physics since there are several fundamental physical phenomena that the Standard Model does not explain. Some of the most important ones are:

- **Gravity**: the addition of a graviton to the Standard Model is not consistent with the existing experimental data without requiring other theoretical modifications that have not been observed yet. In the last decades there have been several theoretical efforts that tried to unify the Standard Model and General Relativity, such as loop quantum gravity [110] or string theory [111], but so far there is no experimentally verifiable theory that fits the Standard Model and Einstein’s General Relativity theory in a single framework. However, at the scale of particle physics at the LHC, the very small effect of gravity can be neglected in most of the calculations.

- **Dark matter**: cosmological observations, such as galaxy rotation curves [112], gravitational lensing [113] and the Cosmic Microwave Background [114], indicate the existence of matter that does not interact with the electromagnetic force. This matter is referred to as dark matter and it is estimated that it makes up about 26% of the total matter-energy budget of the universe. It is also estimated that only 5% of the matter-energy budget of the universe is contained in the observable stars and galaxies. However, the Standard Model does not predict any fundamental particles that are suitable candidates for dark matter.

- **Dark energy:** cosmological observations such as the Cosmic Microwave Background [114] and the redshift of type Ia supernovae [115], indicate that the universe is accelerating. A possible way to describe this in cosmological models is the inclusion of a cosmological constant that corresponds to an intrinsic energy density of the vacuum which drives the observed expansion of the universe. This is referred to as dark energy, and it is estimated that approximately 69% of the universe's energy consists of it. The dark energy cannot be explained with the vacuum energy of the Standard Model.

- **Neutrino masses**: according to the SM, neutrinos are massless particles. However, there is experimental evidence that neutrinos oscillate between different flavour eigenstates, which implies that neutrinos do have a small non-zero mass [116, 117]. It is possible to extend the SM Lagrangian with mass terms for the neutrinos by introducing Yukawa couplings similarly as has been done for quarks and leptons. However, this leads to new theoretical issues, in particular the Yukawa couplings to the Higgs field turn out to be unnaturally small and the existence of weakly interacting right-handed neutrinos is required. An alternative way of explaining the masses of the neutrinos is the inclusion of a Majorana mass term in the SM Lagrangian assuming that neutrinos are their own anti-particles. Several experiments are currently investigating this problem.

- **Matter–antimatter asymmetry**: as discussed in Section 4, in the SM every particle has an anti-particle with opposite quantum numbers, thus the Standard Model predicts that the amount of matter and antimatter should be similar. However, it is observed that our universe consists mostly out of matter. The CP violation that is predicted by the SM seems to be insufficient to explain this observed matter-antimatter asymmetry of the universe [101].

Furthermore, the Standard Model is constructed in an ad-hoc fashion in order to reproduce the experimental data. This is no issue of the model, but some of the features of the SM are considered unnatural by theorists and they imply a lack of understanding. The most important ones are:

- **Hierarchy problem**: as described in Section 4.1.1 the masses of the particles in the SM are generated by spontaneous symmetry breaking caused by the Higgs field. Within the SM theory, virtual particles introduce large quantum corrections to the mass of the Higgs. These corrections require a fine-tuning of the bare mass parameter of the Higgs in the SM, such that the quantum corrections are canceled almost completely. This level of fine-tuning is considered unnatural by many theorists [101].

- **Number of parameters**: if neutrinos are Dirac fermions, the Standard Model has 19 free parameters: the 12 Yukawa couplings of the fermions to the Higgs field, the three coupling constants describing the strength of the gauge interactions, the three coupling constants describing the strengths of the gauge interactions, and the parameters of the CKM matrix. The values of these parameters have to be inferred from experiment and the origin of the values is unknown. Between the different parameters certain patterns are visible and the coupling constants of the three gauge interactions are of a similar order of magnitude, which may be a hint that they are different low-energy manifestations of a Grand Unified Theory [101].

- **Strong CP problem**: in principle the SM can contain a term that leads to CP symmetry breaking in the strong interaction [101]. However, experimentally it was found that this strong CP phase is very close to zero, which is also considered unnatural by many theorists.

To the date of writing this thesis, no experimental result is known to contradict the Standard Model at the 5-σ level [9]. Due to statistical and systematic uncertainties, it is expected that some of the experimental tests of the SM will significantly differ from the predictions of the SM. So far, most of these contradictions turned out to be statistical fluctuations once more data was analyzed. However, any possible new physics beyond the SM will first show up in experimental data as a statistical discrepancy between the prediction of the SM and the measured data. In order to distinguish statistical fluctuations from signs of new physics, a precise modelling of systematic uncertainties is crucial. To the date of the writing of the thesis, the two most notable experimental results that are in tension with the SM prediction are:

- **Anomalous magnetic dipole moment of the muon**: the experimentally measured value of the muon's anomalous magnetic dipole moment differs significantly from the Standard Model prediction. Muons that travel through the strong external magnetic field of a storage ring have the direction of their magnetic moments precess at a rate that depends on its strength g. The g-factor is predicted to be equal to two by the Dirac equation, but higher order loops add an anomalous moment, $a_\mu = (g-2)/2$, which can be calculated with very high precision. In the early 2000s the Brookhaven National Laboratory conducted an ultra-precise experiment and the measured data are disagreeing with the SM at the 3.7 σ level [118]. The experiment was rebuilt at Fermilab and in 2021 the first run of the muon g-2 experiment has reported a disagreement with the SM prediction of 3.3 σ [119], in agreement with the Brookhaven experiment. Combining the results of both experiments, the disagreement with the SM prediction currently stands at 4.2 σ.

- **Anomalies in flavour-changing B decays**: the LHCb collaboration has measured a flavour anomaly in the decay of B mesons into a K^* and a pair of muons [120, 121]. The $\mathrm{B}^0 \to \mathrm{K}^{*0}\mu^+\mu^-$ decay is sensitive to new physics since this decay is forbidden at the lowest perturbative order in the SM and instead occurs via higher-order penguin and box processes, which are sensitive to the presence of new, heavy particles. Such particles could significantly change the decay rate and the angular distribution of the final-state particles. The LHCb collaboration first reported the anomaly in 2013, which was confirmed with more data in 2020 and currently stands at a disagreement with the SM prediction at 3.3 σ.

4.2.2 Possible Extensions

Due to the limitations of the Standard Model discussed in the previous section, alternative theories to describe fundamental interactions are being developed. Most of the proposed theoretical models contain the SM as a low-energy Effective Field Theory. Two important extensions of the SM that are being tested experimentally at the LHC are:

- **Supersymmetry (SUSY)**: which is a possible solution to the hierarchy problem based entirely on symmetries [122]. The basic idea of SUSY is that each Standard Model particle has a super-partner "sparticle" which differs by half a unit of spin. In the minimal supersymmetric Standard Model (MSSM), the particle content of the SM is approximately doubled. The super-partner of each fermion is a spin-0 scalar (sfermion) and the super-partners of the spin-1 gauge fields are spin-half gauginos. For example, for a quark there is a squark, for a lepton there is a slepton, and for the gluon and photon there is a gluino and a photino. The MSSM enlarges the SM Higgs to two Higgs doublets H_u and H_d, which have spin-1/2 superpartners called Higgsinos. Under electroweak symmetry breaking, the Higgsinos and gauginos mix and form the physical chargino and neutralino states. In the formalism of SUSY, the large quantum corrections to the mass of the Higgs in the SM are canceled by corresponding loops of superpartners. Furthermore, in many supersymmetric models, the lightest neutralino is a weakly interacting stable particle, and is a possible WIMP candidate for the dark matter in the universe [101].

- **Composite Higgs models** aim at solving the hierarchy problem through a combination of strong dynamics and symmetry [122]. It is assumed that the Higgs boson is a composite bound state of an additional strongly coupled sector. The compositeness scale is constrained by direct searches and electroweak precision tests to be at least in the multi-TeV range. Modern composite Higgs models generally also introduce an approximate global symmetry in the composite sector to explain why the Higgs boson is a narrow, light state that is separated from the other resonances of the composite sector. When the global symmetry is spontaneously broken, the SM Higgs boson emerges as a pseudo–Nambu–Goldstone boson similar to the pion of QCD. Many composite Higgs models predict towers of electroweak vector resonances, such as the W' and colored fermionic resonances starting at around the compositeness scale.

Despite extensive searches for BSM physics by the ATLAS and CMS experiments, so far no evidence for new physics has been found and stringent limits have been set on many BSM models [122].

4.3 Proton-Proton Collisions at the LHC

In order to analyze the data of the proton-proton collisions recorded by the LHC experiments, it is crucial to understand the hard scattering processes of the proton constituents, called partons. These processes are characterized by a momentum transfer that is large compared to the proton mass and they can be calculated to high accuracy with perturbative Quantum Chromodynamics (pQCD). Recording the final states of the hard scatterings in proton-proton collisions at the LHC allows to confront the predictions of pQCD with the experimental data. The discussion of hard processes in this chapter is based on reference [123].

4.3.1 PDFs

The initial stage of the proton-proton collisions is dominated by the effective densities of the parton distribution functions (PDFs) of the proton. The quarks inside the proton interact with each other through the exchange of gluons. At high energies a sea of virtual quark-antiquark pairs is generated by the constant gluon exchange between the three constituent quarks of the proton. Therefore, in the interaction of two protons, both the constituent quarks and the gluons and sea quarks take part in the hard scattering process [101]. The dynamics of this interacting system results in a distribution of quark momenta within the proton and the distributions are expressed in terms of PDFs. In practice, the functional forms of the PDFs depend on the detailed dynamics of the proton and have to be inferred from experiments. There

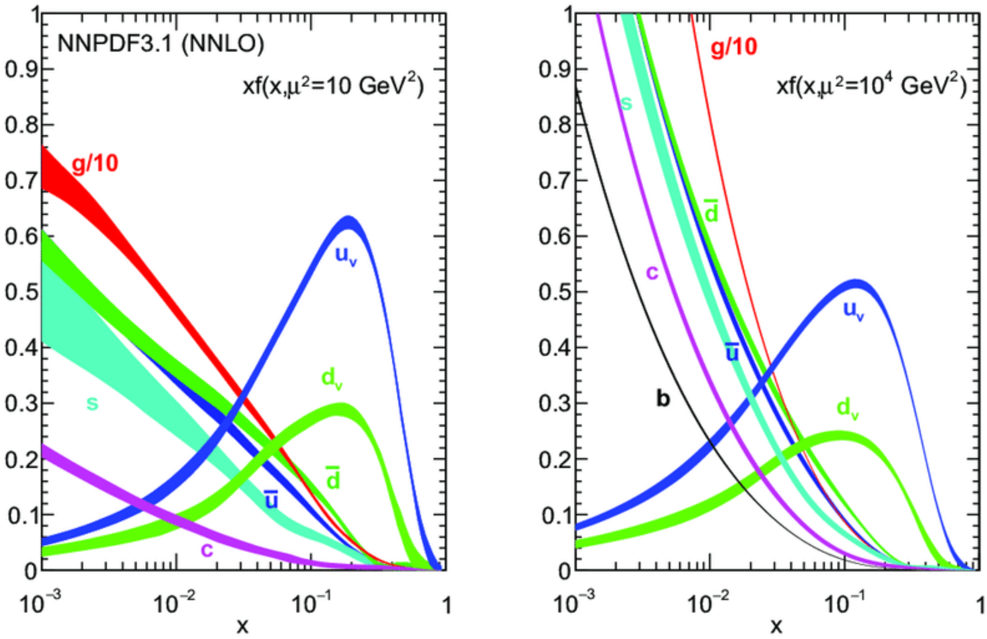


Figure 4.4: Results of the latest global PDF fit by the NNPDF Collaboration. PDFs are shown at factorization scales of $\mu^2 = 10$ GeV^2 (left) and $\mu^2 = 10^4$ GeV^2 (right). The figure is adapted from [124].

are many complementary ways to measure the proton PDFs and the PDFs that are usually used in LHC analysis are extracted from a global fit to the experimental data from various experiments [125]. The theoretical framework of QCD, in particular the DGLAP evolution equations allow to calculate the evolution of the PDFs at different scales once that they have been measured at a fixed scale [126, 127, 128]. Various groups provide parametrizations of PDFs based on fits of the experimental data, the most popular are those from the CTEQ [129], MSTW [130], and NNPDF [131] Collaborations. Figure 4.4 shows the parton distribution functions at two different energy scales at $\mu^2 = 10$ GeV^2 (left) and $\mu^2 = 10^4$ GeV^2 (right) estimated by the NNPDF collaboration [124]. Valence quarks are dominant at lower energies, while at higher energies gluon scattering is the dominant process.

4.3.2 Hard Processes

Hard scattering processes probe distance scales far below the radius of the proton and can be described as collisions between partons. The cross-section can be calculated by factorising the PDFs $f_{i,p}$ of the proton and the cross-section of the parton processes. The full expression for the cross-section can be written as [123]:

$$\sigma(pp \to X) = \sum_{i,j} \int dx_1 dx_2 f_{i,p}\left(x_1, \mu_{\mathrm{F}}^2\right) f_{j,p}\left(x_2, \mu_{\mathrm{F}}^2\right) \hat{\sigma}_{ij\to X}\left(x_1 x_2 s, \mu_{\mathrm{R}}^2, \mu_{\mathrm{F}}^2\right) \tag{4.51}$$

where the sum runs over all possible initial-state partons with longitudinal momentum fractions $x_{1,2}$, that can give rise to a final state X at a center-of-mass energy of $\sqrt{x_1 x_2 s}$, with s denoting the total proton-proton center-of-mass energy squared. The renormalization scale μ_{R}^2 and the factorization scale μ_{F}^2 originate from truncated expansions in the strong coupling constant. The hard cross-section $\hat{\sigma}$ can be calculated at leading order (LO) in the strong coupling, α_S, and can also incorporate next-to-leading-order (NLO) and next-to-next-to-leading-order (NNLO) corrections. In general, fixed-order predictions are associated with final states that have a small number of partons.

4.3.3 Underlying Event, Hadronization and Parton Showers

Every LHC analysis requires a thorough understanding of QCD processes which involves more than studying only the hard scattering [132]. An overview over the process of a hard scattering in a proton-proton collision is shown in Fig. 4.5. The resulting event contains initial and final-state radiation, as well as particles that come

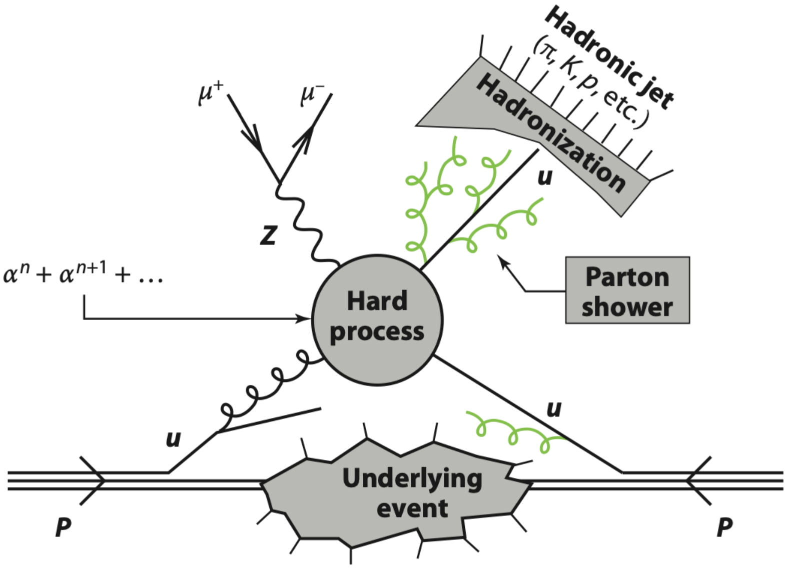


Figure 4.5: Sketch of a proton-proton collision, where a quark-gluon scattering results in a final state consisting of a Z boson and a jet [123].

from the breakup of the proton and antiproton, also called “beam-beam remnants”. The “underlying event” is everything except the two outgoing hard scattered partons and receives contributions from the “beam-beam remnants” plus initial- and final-state radiation. The “hard scattering” component consists of the outgoing two partons plus initial and final-state radiation. Shortly after hard partons are produced, they repeatedly radiate low-energy and collinear gluons in a process called parton shower [123]. After the particle shower has terminated, the partons transition to hadrons, which is called hadronization. In this energy regime non-perturbative effects become important and so far there exists no exact theory that analytically describes hadronization. However, models exist that can describe hadronization and parton showers, forming the basis of popular general-purpose Monte Carlo (MC) simulation programs such as PYTHIA [133], HERWIG [134], and Sherpa [135]. These MC generators allow to obtain realistic descriptions of proton-proton collisions with all final-state particles. Often, parton showers are matched with multileg tree-level matrix elements, e.g. from MadGraph [136] and also with NLO matrix elements, making use of methods such as MC@NLO [137] and POWHEG [138]. Usually the results of the measurements at the LHC are compared either with NLO or NNLO fixed-order predictions or with results from parton-shower programs.

4.3.4 Jets

A high energy parton emitted from a hard scattering in a proton-proton collision cannot be recorded directly in the detector. According to the hypothesis of colour confinement, the reason for this is that the QCD force that becomes stronger the farther the parton gets from the proton. Soft gluons are radiated at small angles relative to the original parton, until the threshold for hadronization is reached and the partons form colour-neutral hadrons. This results in a collimated jet of hadrons where the collective energy and momentum reflect those of the initially scattered parton. Several algorithms exist that allow to combine these hadrons into a jet. Typically, these algorithms have to be collinear and infrared safe, which means that the resulting hard jets are not affected by the collinear and soft splittings that orig-

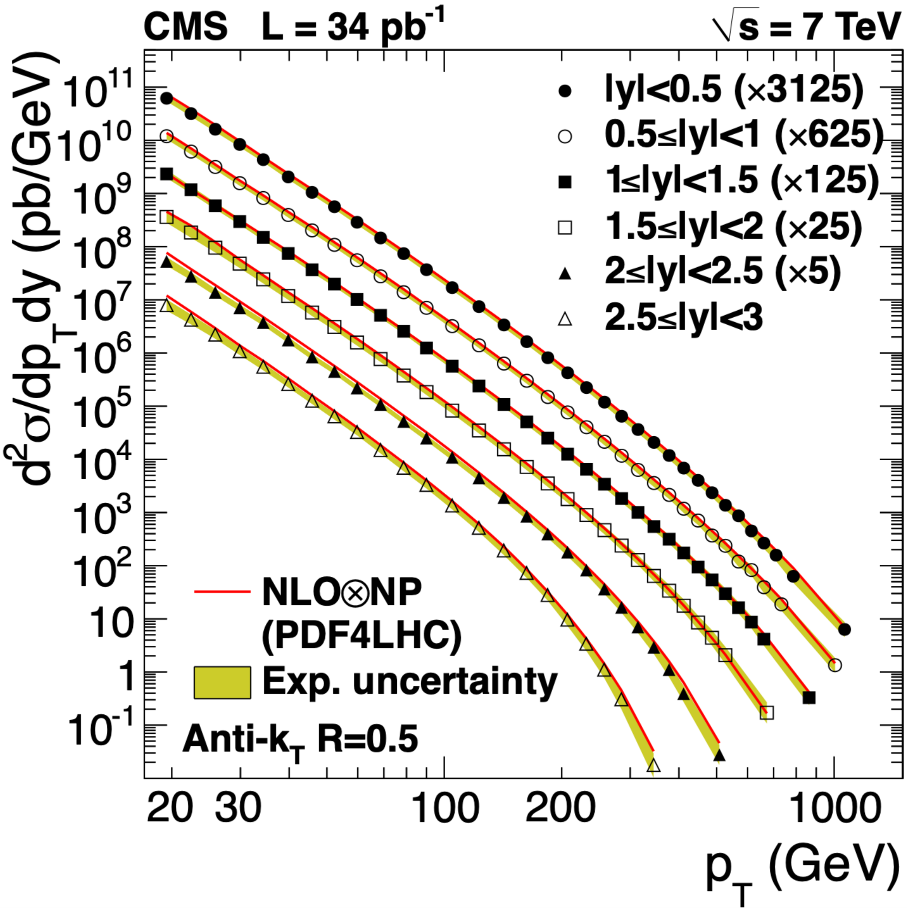


Figure 4.6: Measurement of the double-differential inclusive jet cross-section as a function of the jet transverse momentum and rapidity by the CMS collaboration and comparison with NLO QCD predictions [139].

inate from the parton shower. This feature is important because it ensures that the calculations in pQCD are finite at every order in perturbation theory. Measurements of jet properties at very high energies allows to confront the pQCD with experimental data. An example for a measurement of the inclusive jet production cross-section in intervals of $\Delta y = 0.5$ of rapidity by the CMS collaboration [139] is shown in Fig. 4.6. The NLO QCD calculations are in good agreement with the data over the full considered range. In general, QCD is found to provide an excellent description of jet phenomena in proton–proton collisions.

4.4 Top Physics at the LHC

The top-quark is the heaviest elementary particle in the Standard Model and was discovered by the CDF [140] and D0 [141] collaborations at the Tevatron proton–anti-proton collider at Fermilab in 1995. Due to its very short lifetime ($\sim 0.5 \times 10^{-24}$ s), which is shorter than the hadronization timescale, no hadronic top-quark pair bound states can form. This fact offers a unique possibility to study the properties of the particle as a quasi-free quark [9]. The precise knowledge of the elementary particle masses and their couplings is an important element in consistency tests of the SM

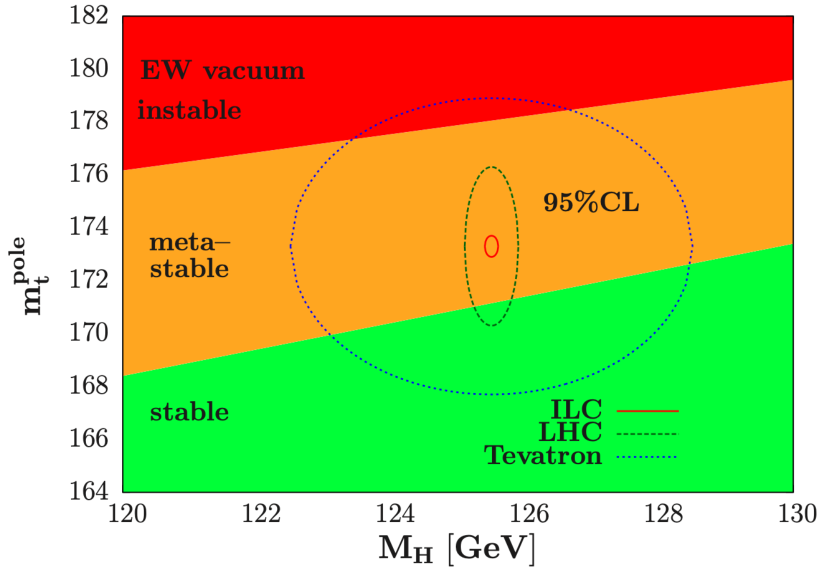


Figure 4.7: Two-σ ellipses in the top and Higgs mass plane obtained from Tevatron and LHC measurements (from 2012) and a possible future ILC collider, confronted with the areas in which the SM vacuum is absolutely stable, metastable and unstable up to the Planck scale [142].

and in indirect searches of physics beyond the SM. So far, the LHC has not discovered new particles that are not predicted by the SM, therefore indirect searches for new physics, which focus on finding deviations between experimental data and SM predictions, are becoming increasingly important. These searches require a high level of precision and an excellent understanding of experimental systematic uncertainties. The precise determination of the top mass is crucial for testing the overall consistency of the Standard Model and to constrain new physics models through precision electroweak fits [122, 143]. Due to its large value of the order of the electroweak scale, the top mass has a direct impact on the Higgs sector of the SM, and on extrapolations of the SM to high-energy scales. Therefore, precision measurements of the top-quark mass are important to understand the stability of the electroweak vacuum, since radiative top-quark corrections can drive the Higgs-boson self-coupling towards negative values, which potentially can result in an unstable vacuum. The precision of the top-quark mass measurement is crucial for the determination of the energy scale where the vacuum might become unstable, which possibly requires new physics at lower or comparable energies [144]. As indicated in Fig. 4.7, the current experimental uncertainties do not allow to determine whether the electroweak vacuum is stable when the SM is extrapolated up to the Planck scale. Thus, the detailed study of top-quark production, decays, couplings and other top properties with high precision is an important part of the physics program of the CMS experiment. In the following, a selective overview on some of the key properties of top-quark measurements will be given. The description is based on a review on top physics [145] which is also recommended for a complete overview of top-quark properties.

4.4.1 Top Pair Production

The measurement of the inclusive top pair production cross-section $\sigma_{\mathrm{t\bar{t}}}$ is an important test of Quantum Chromodynamics. Calculations of the cross-section are performed at next-to-next-to-leading order (NNLO) including the resummation of soft gluon terms (NNLL). Important systematic uncertainties are stemming from scales, PDFs, and the strong coupling constant α_S [145, 146]. The main decay channel of top-quarks is the flavour-changing charged current decay $t \to Wb$. Several $\mathrm{t\bar{t}}$ final states are available for the measurement of $\sigma_{\mathrm{t\bar{t}}}$ that can be grouped in three W boson decay categories: an all hadronic jet channel $W \to q\bar{q}$, a leptonic decay channel $W \to \ell\nu_\ell$ where ℓ stands for electrons and muons, including those from the leptonic decays of tau-leptons, and a channel where the tau-lepton decays hadronically $W \to \tau_{\mathrm{h}}\nu_\tau$. Accordingly, the $\mathrm{t\bar{t}}$ events can be classified as “dilepton”, “single-

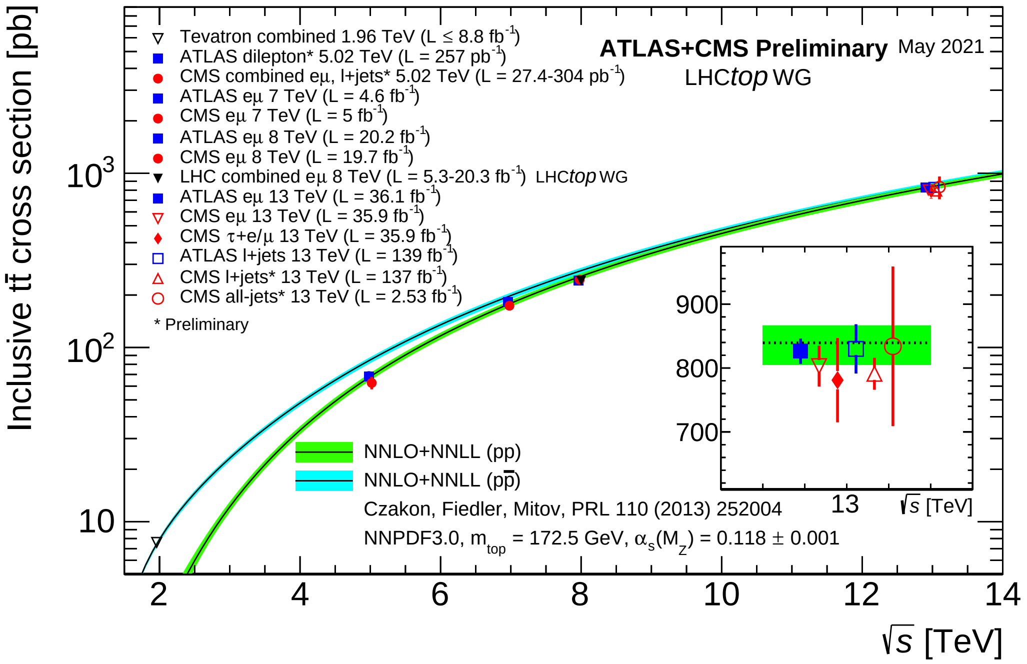


Figure 4.8: Summary of LHC and Tevatron measurements of the top-pair production cross-section as a function of the center-of-mass energy compared to the NNLO QCD calculation complemented with NNLL resummation (top++2.0). The theory band represents uncertainties due to renormalization and factorization scale, parton density functions and the strong coupling. The measurements and the theory calculation are quoted at $m_{top} = 172.5$ GeV. Measurements made at the same center-of-mass energy are slightly offset for clarity. The figure and the description is provided by the LHC Top Working Group and adapted under licence CC BY 4.0.

lepton", "all-hadronic" and categories with tau-leptons. The different channels are complementary and inconsistencies between the measured cross-sections could potentially indicate new physics. The most precise measurements of $\sigma_{\mathrm{t\bar{t}}}$ at the LHC are obtained in the dilepton channel [145], and in particular the $e\mu$ final state, since it is essentially background free. A typical analysis technique is a template fit to multi-differential binned distributions that are characterized by the number of b-jets and the transverse momentum and multiplicity of other jets in order to extract the number of signal and background events. The largest yield of $\mathrm{t\bar{t}}$ events is in the all-hadronic channel, however the large multijet background and the many possible jet combinations make the measurement challenging. CMS and ATLAS have also measured the inclusive cross-section in events with one identified hadronically decaying tau-lepton, both in the channels $\tau_\mathrm{h} + \ell$ and τ_h+jets [145]. Besides testing the consistency of the $\mathrm{t\bar{t}}$ cross-section measurements in different channels, this channel is interesting since hypothetical charged Higgs bosons could be produced in top-quark decays that further decay through $H^+ \to \tau^+\nu_\tau$, which would reflect in a modification of the branching fractions. The CMS analysis of the decay channel τ_h+jets [12] is of particular importance for this thesis, since its replication will be the main subject of Chapter 6. Figure 4.8 shows a summary of LHC and Tevatron measurements of the top-pair production cross-section as a function of the center-of-mass energy compared to the NNLO QCD calculation complemented with NNLL resummation [145]. The measured inclusive cross-sections for $\mathrm{t\bar{t}}$ pair production at proton-proton center-of-mass energies of 7 TeV, 8 TeV and 13 TeV are found to be in very good agreement with the theoretical calculations.

4.4.2 Top Mass

As discussed Section 4.1.3, the top Yukawa coupling is a free parameter in the SM, which implies that the top mass has to be inferred from experimental measurements. It can either be measured through direct measurements, which rely on the kinematic reconstruction of the final-state top-quark decay products, or indirectly through relations with other measured observables, such as the $\mathrm{t\bar{t}}$ cross-section at a given center-of-mass energy [147]. Currently the most precise mass measurements come from direct measurements. The reconstructed final-state particles are compared with predictions of Monte-Carlo event generators to determine the top-quark mass value that best describes the data. The measured top-quark mass then corresponds to the parameter implemented in the MC generator and the direct measurements yield a world average of $m_{top}^{MC} = 173.34$ GeV with a precision up to 0.5 GeV. The result for the top mass obtained from direct measurements m_{top}^{MC} is usually identified with the top-quark pole mass m_{top}^{pole}, which is a popular renormalization scheme used for perturbative QCD computations at next-to-leading order and beyond [143]. However, so far no precise relation between the Monte Carlo mass m_{top}^{MC} and more fundamental and field-theoretic mass definitions is known. As a result, the identification yields an uncertainty of the order of 1 GeV. Indirect determinations of the top mass, based on the comparison of inclusive or differential production cross-sections to the corresponding theory calculations can also be used to extract the top mass in a well-defined renormalization scheme, but currently are less precise than direct measurements.

The direct measurements performed by CMS and ATLAS are based on kinematic

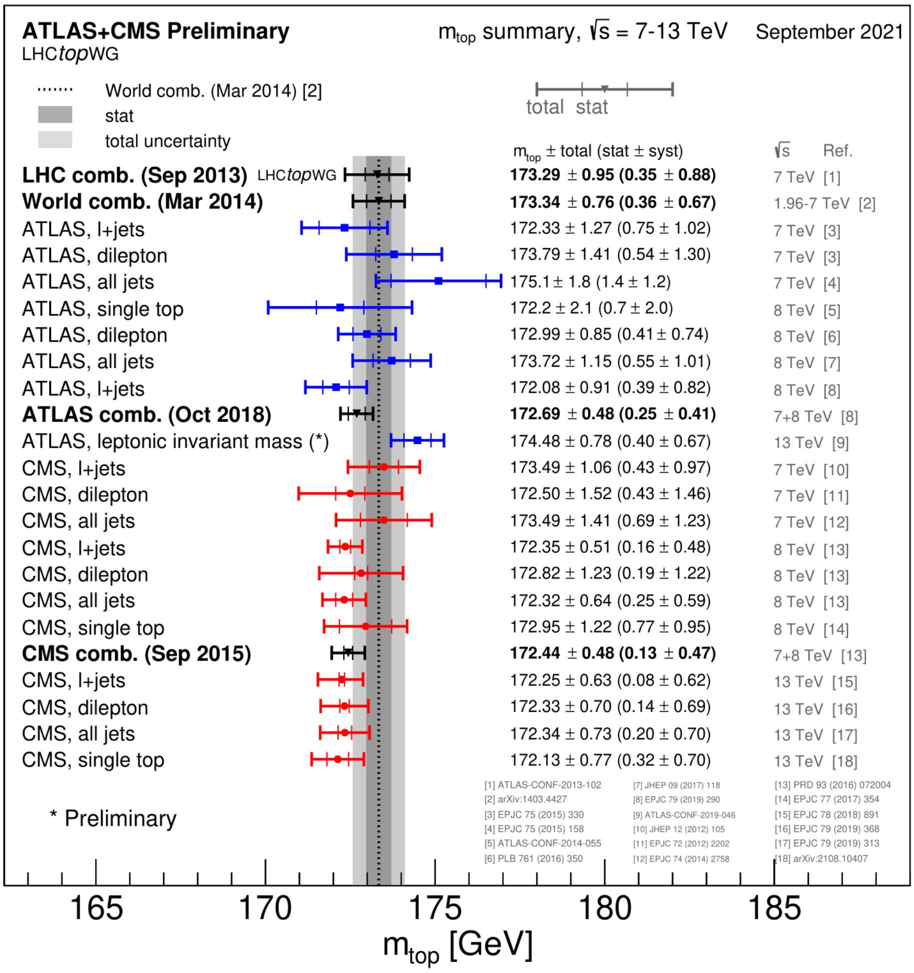


Figure 4.9: Summary of the direct top-quark mass measurements by ATLAS and CMS. The figure is provided by the LHC Top Working Group and adapted under licence CC BY 4.0.

observables reconstructed from top decay products for the different accessible top decay and top production modes. The top-quark mass has been measured in $\mathrm{t\bar{t}}$ events, where the top pair decays in the ℓ+jets channel, in the dilepton channel, and in the all-hadronic channel, as well as in single-top events. Important analysis techniques for the direct mass measurements are ideogram and matrix element methods, where the likelihood of a whole reconstructed final state compatible with a production hypothesis is determined event-by-event [145]. Figure 4.9 provides a summary of all state-of-the-art direct top mass measurements by the ATLAS and CMS collaborations.

4.4.3 CKM Matrix Elements

The CKM matrix element $|V_{tb}|$ can be measured in single-top events that are mainly produced via charged-current electroweak interactions. The W boson virtuality defines three channels: the t-channel, s-channel and the Wt-channel. The square of the magnitude of the CKM matrix element V_{tb} multiplied by a form factor f_{LV} has been determined for different production modes and center-of-mass energies by the ATLAS and CMS collaborations, using the ratio of the measured cross-section to its theoretical prediction. For the extraction, it is assumed that $|V_{tb}| \gg |V_{td}|, |V_{ts}|$ and that the tWb interaction is left-handed as predicted by the Standard Model. The combination of all $|f_{LV}V_{tb}|^2$ determinations at $\sqrt{s} = 7$ and $\sqrt{s} = 8$ TeV, yields $|f_{LV}V_{tb}| = 1.02 \pm 0.04\ (meas.) \pm 0.02\ (theo.)$ [148], consistent with the corresponding Standard Model predictions.

CMS also extracted the CKM matrix elements V_{tb}, V_{td}, and V_{ts} simultaneously and model independent by separating single-top t-channel signal events into multiple categories depending on the quark interactions at both tWb vertices. The signal strength of the individual single-top t-channel modes is determined from a likelihood fit to a multivariate discriminator response. A three-fold interpretation of the measured signal strength parameters is provided, setting limits on the CKM matrix elements under the SM unitary assumption $|V_{tb}| > 0.970$, $|V_{td}|^2 + |V_{ts}|^2 < 0.057$ and allowing for more quark families $|V_{tb}| = 0.988 \pm 0.051$ [3].

4.4.4 Effective Field Theory

Another important approach to interpret SM measurements is the framework of Effective Field Theory (EFT), which extends the SM Lagrangian from Chapter 4 with additional operators:

$$\mathcal{L}_{\text{EFT}} = \mathcal{L}_{\text{SM}} + \sum_i \frac{c_i}{\Lambda^{d_i-4}} \mathcal{O}_i \tag{4.52}$$

where $\mathcal{O}_i$ are the effective operators that characterize the new interactions in the extended theory and d_i is the dimension of the operator. The coefficients c_i are called the EFT or Wilson coefficients that parametrize the strength of the new interactions. The effect of an operator is expected to be suppressed by $(1/\Lambda)^{d_i-4}$, thus high-dimensional operators can be neglected. By probing anomalous couplings of higher order operators and their dimensionless coupling strengths represented by the Wilson coefficients, a model-independent measurement of BSM effects can be obtained.

An example of a global EFT interpretation of multiple relevant processes has been recently published by the CMS collaboration [149]. Multilepton final state events are analyzed and total yields of the top production modes ttZ, $\mathrm{t\bar{t}W}$, $\mathrm{t\bar{t}H}$, tZq, tHq are parametrized in terms of 16 Wilson coefficients associated with Effective Field Theory operators relevant to the dominant processes in the data. A simultaneous fit of the 16 Wilson coefficients to the data is performed and two-standard-deviation confidence intervals for the coefficients are extracted. The results from fitting the Wilson coefficients to the data are consistent with the Standard Model prediction.

5 The CMS experiment at the LHC

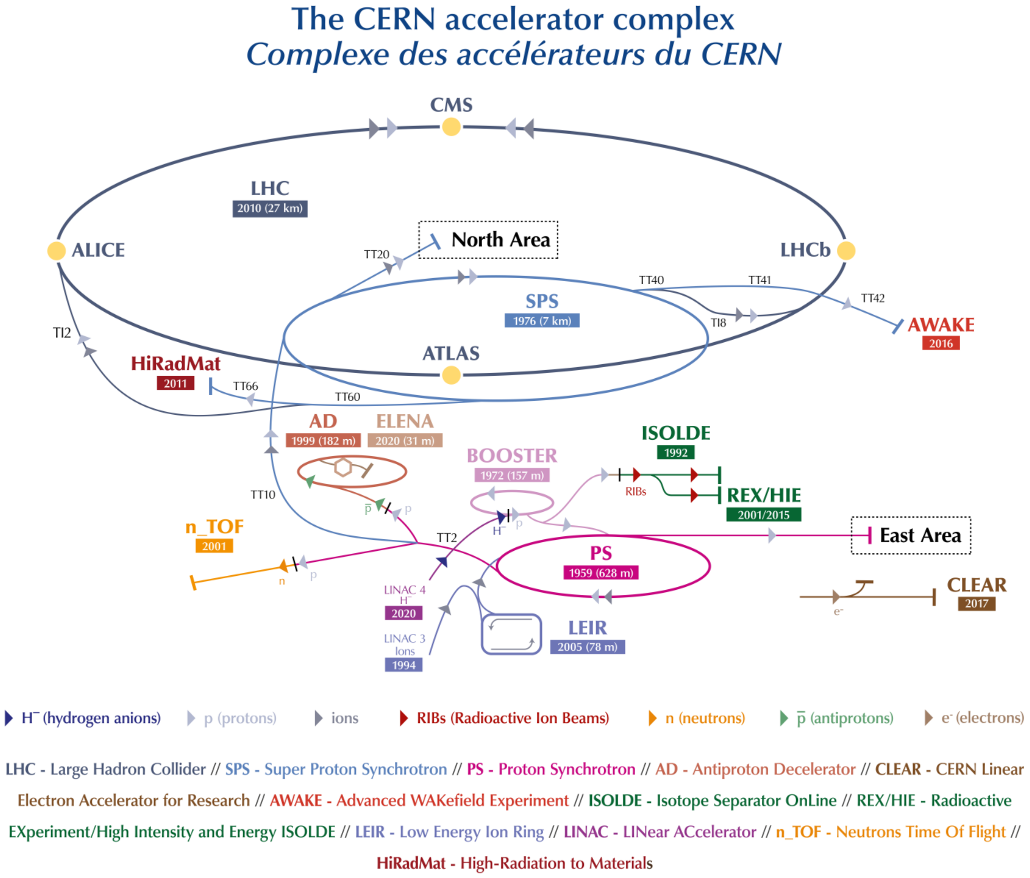
In Chapter 4 the most successful theory of fundamental interactions, called the Standard Model, and its limitations have been discussed. Its predictions, as well as alternative theoretical models, can be tested by confronting the calculations with data obtained from the analysis of proton-proton collisions at very high energies. The Large Hadron Collider (LHC) at CERN and its experiments have been built in order to record proton-proton collisions in a controlled setting. In this chapter, the design of the Compact Muon Solenoid (CMS) detector at the LHC, and its techniques to reconstruct particles produced in proton-proton collisions will be discussed.

5.1 The Large Hadron Collider

The Large Hadron Collider is the world's largest and most powerful particle accelerator up to date. It is designed to achieve center-of-mass energies up to 14 Tev with nominal instantaneous luminosities reaching 1×10^{34} cm^{-1} s^{-1} for proton-proton collisions and luminosities of 10^{27} cm^{-2} s^{-1} for heavy ions (Pb) with an energy of 2.8 TeV per nucleon. The LHC accelerator is located on average 100 meters below the surface in the 26.7 km long accelerator tunnel that previously contained the LEP accelerator. The accelerator collides beams of protons or ions that travel in opposite directions almost at the speed of light in two independent vacuum pipes. The beams are guided around the accelerator ring by a strong magnetic field maintained by superconducting electromagnets. The description of the accelerator in the following sections is based on reference [151].

5.1.1 Injection and Acceleration Chain

The LHC accelerator consists of eight long straight sections and eight arcs where superconducting dipole magnets are installed that deflect the particles [151]. The four main LHC experiments, ALICE, ATLAS, CMS, and LHCb are installed at interac-



LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive EXperiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials

Figure 5.1: Schematic view of the CERN acceleration complex with its main experiments [150]. The protons are first accelerated in LINAC before they are passed on to the PS Booster, the PS, the SPS and finally to the LHC.

tion points (IP) in the middle of four long straight sections. A schematic view of the CERN accelerator complex with the four experiments is shown in Fig. 5.1. In order to bend protons with a momentum of up to 7 TeV per unit charge, a dipole field of 8.3 T is generated by superconducting dipole magnets made of Niobium–Titanium (NbTi) [152]. The two beam pipes are contained in a single cryostat. In addition to the dipole magnets, also quadrupole, sextupole and octupole magnets are installed. The purpose of the quadrupole and octupoles magnets is the stabilization of the beam, while the sextupole magnets allow to correct the energy dependence of the magnetic fields. In total 8000 superconducting magnets are used to control the two beams. A superfluid helium cooling system with eight continuous cryostats is used to cool the superconducting dipole magnets and the quadrupole magnets to a temperature of 1.9 K.

In order to reach beam energies of the order of a few TeV, the protons have to pass through several components of the CERN accelerator complex [152]. After the extraction of a low-energy beam of protons, the protons are first accelerated in a linear accelerator. In the LHC Run I and Run II this was done with the Linear Accelerator 2 (LINAC2), which accelerated the protons up to 50 MeV. Since 2020 the Linear Accelerator 4 (LINAC4) has become the source of proton beams for the CERN accelerator complex [153]. The beams from LINAC are further accelerated in the four Proton Synchrotron Booster (PSB) rings to 1.4 GeV, and in the next step by the Proton Synchrotron (PS) to 26 GeV. Finally, the Super Proton Synchrotron (SPS) at the end of the injection chain accelerates the protons for the LHC to an energy of 450 GeV and injects them in opposite directions in the LHC ring. This first LHC injection phase lasts 20 to 30 minutes. Subsequently the beams are further accelerated with a superconducting radio-frequency system to an energy of up to 7 TeV in about 20 minutes. Once the LHC accelerator has been filled, the proton beams can be used to record collisions at the IPs for several hours. In the case that a problem occurs, the LHC fill can be terminated by forcing the proton bunches to collide against graphite absorbers that are installed tangent to the beam pipes.

5.1.2 Performance

The particles in the beams are stored in bunches that collide at the interaction points where the main experiments are installed. The resulting products from the particle collisions are recorded with the different detectors. A high event rate, characterized by the luminosity $\mathcal{L}$, is crucial since only a small fraction of the collisions are of interest. The event rate of a process with cross-section σ is given by:

$$\frac{dn}{dt} = \mathcal{L}(t) \cdot \sigma \ . \tag{5.1}$$

The instantaneous luminosity depends only on the beam parameters and, assuming a Gaussian beam distribution, can be calculated as:

$$\mathcal{L}_{\text{inst}} = \frac{N_b^2 \gamma_r f_{rev} n_b}{4\pi\epsilon_n \beta^*} F \tag{5.2}$$

where N_b is the number of particles per bunch, n_b the number of bunches per beam, f_{rev} the revolution frequency, γ_r the relativistic gamma factor, ϵ_n the normalised transverse beam emittance, β^* the beta function at the collision point, and F the geometric luminosity reduction factor due to the crossing angle at the interaction points. Since 2010 the LHC has recorded proton-proton data in multiple acquisition periods at different center-of-mass energies. The different runs and recorded luminosity by the CMS experiment are summarized in Table 5.1.

Period	Year	$\sqrt{s}$ [TeV]	LHC delivered [fb^{-1}]	CMS Recorded [fb^{-1}]
Run I	2010	7	40.76×10^{-2}	40.22×10^{-2}
	2011	7	6.13	5.55
	2012	8	23.30	21.79
Run II	2015	13	4.22	3.81
	2016	13	40.82	37.76
	2017	13	49.79	44.98
	2018	13	67.86	63.67

Table 5.1: Summary of the cumulative luminosity delivered by the LHC and recorded by the CMS experiment. The information is accumulated from the CMS Public Luminosity Website.

5.1.3 The Experiments

Eight experiments with diverse research programs are installed at the LHC. The experiments are run by collaborations of scientists from institutes all over the world. The four large particle experiments installed at the LHC interaction points are:

- **ALICE** [154] - an acronym that stands for “A Large Ion Collider Experiment” - is a detector dedicated to heavy-ion physics at the LHC. It is designed to study the physics of strongly interacting matter at extreme energy densities, where a phase of matter called quark-gluon plasma forms. Studying its properties is a key ingredient for a better understanding of Quantum Chromodynamics, in particular the phenomenons of confinement and chiral-symmetry restoration.
- **ATLAS** [155] - an acronym that stands for “A Toroidal LHC ApparatuS” - is one of the two general-purpose detectors at the LHC. It investigates a wide

range of physics, from the search for the Higgs boson to extra dimensions and particles that could make up dark matter. It is the largest experiment at the LHC.

- **CMS** [156] - an acronym that stands for a "Compact Muon Solenoid" - is the other general-purpose detector at the LHC and will be described in more detail in the next sections, since the work of this thesis has been done within this collaboration. It has a broad physics program and although it has similar scientific goals as the ATLAS experiment, it uses different technical solutions and a different magnet system design.
- **LHCb** [157] - an acronym that stands for the "Large Hadron Collider beauty" - focuses on precision measurements of the properties and decays of b-quark and c-quark hadrons as well as the search for indirect evidence of new physics that can explain CP violation.

Besides of the four big experiments, there are four smaller experiments installed at the LHC: TOTEM [158] and LHCf [159], which focus on measuring protons or heavy ions at forward rapidity. TOTEM uses detectors positioned on either side of the CMS interaction point, while LHCf is made up of two detectors located at the LHC beamline, at 140 m either side of the ATLAS collision point. MoEDAL [160] uses detectors deployed near LHCb to search for magnetic monopoles. The newest LHC experiment is FASER [161], located 480 m from the ATLAS collision point with the goal to search for light new particles and study neutrinos.

5.2 The Compact Muon Solenoid

The Compact Muon Solenoid (CMS) [156] detector at the CERN LHC is a general purpose detector designed primarily to search for signatures of new physics in proton-proton and heavy-ion collisions. The CMS detector has a cylindrical geometry that is azimuthally (ϕ) symmetric with respect to the beamline. Particles that are produced in a collision in the interaction region first pass through a tracker, in which the trajectories and vertices of charged particles are reconstructed from signals in the sensitive layers. The next layer is an electromagnetic calorimeter (ECAL), where electrons and photons are absorbed. The electromagnetic showers are detected as clusters of energy in neighbouring cells, from which the energy and direction of the particles are determined. Subsequently, the hadronic showers of charged and neutral hadrons are fully absorbed in a hadron calorimeter (HCAL) and the corresponding

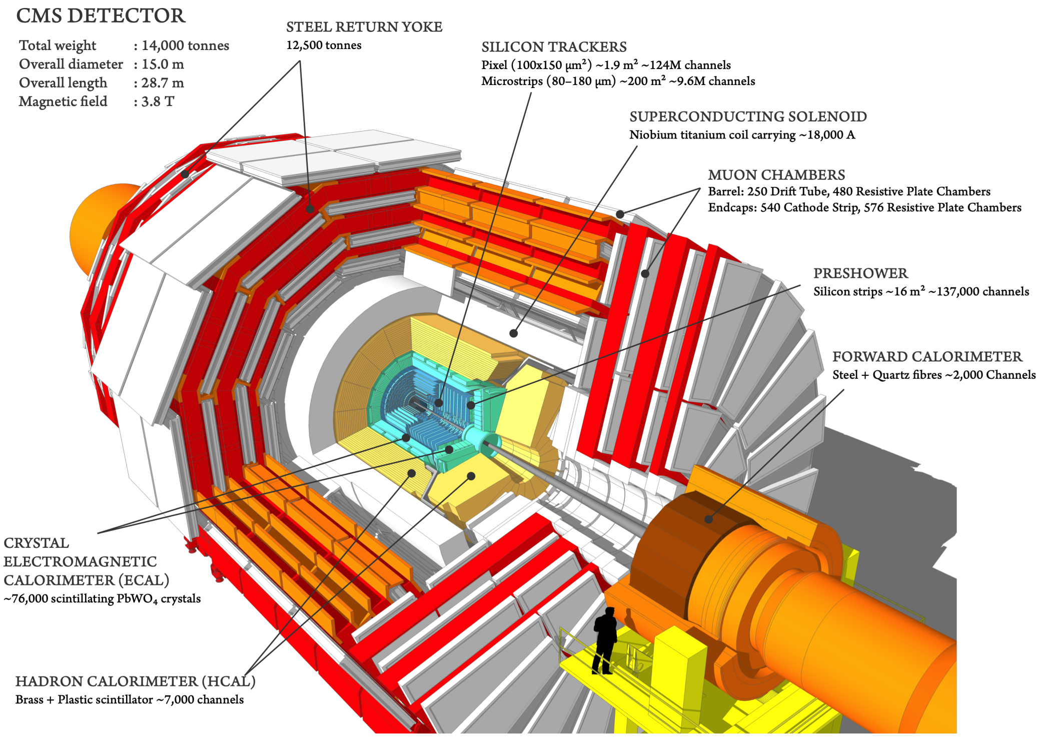

CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm²) ~1.9 m² ~124M channels
Microstrips (80–180 μm) ~200 m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000 A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 540 Cathode Strip, 576 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16 m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO4 crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

Figure 5.2: Cutaway view of the CMS detector with the main detecting systems and characteristics [162].

clusters are used to estimate their energies. The tracker, as well as the calorimeters, is contained in a superconducting magnet, which provides a 3.8 T solenoidal field that bends the trajectories and allows to measure the electric charge and momentum of the particles. Outside of the solenoid, additional tracking layers are installed that allow to measure muons which traverse the calorimeters without being stopped in the dense material. Figure 5.2 displays a 3-dimensional view of the CMS detector with its main detector sub-components.

The coordinate system used to describe the CMS detector has its center inside the detector at the nominal interaction point. The x-axis points inwards towards the LHC ring center, while the y-axis points vertically upwards and the z-axis points tangent to the beam line. Due to the cylindrical symmetry of the detector, the coordinate system can be expressed in spherical coordinates, where ϕ is the angle with respect to the x-axis in the transverse $x-y$ plane and θ is the polar angle with respect to the LHC plane.
Since the colliding partons carry different longitudinal momentum fractions, the

center-of-mass frames of the parton-parton collisions have different longitudinal boosts. It is therefore useful to define the rapidity y for a particle with energy E and momentum p_z as:

$$y = \frac{1}{2} \ln\left(\frac{E + p_z}{E - p_z}\right) . \tag{5.3}$$

This variable has the advantage that the difference Δy in rapidity between two particles is Lorentz invariant under boosts along the longitudinal z-axis. The rapidity can be approximated by the pseudorapidity η in the relativistic limit, where $E \gg m$:

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) . \tag{5.4}$$

With these definitions, the angular distance between two particles can be defined as:

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2} \tag{5.5}$$

which is invariant under boosts along the z direction of the particle in the highly relativistic limit.

5.2.1 Magnet

The magnetic field at the CMS experiment is generated by a superconducting solenoid that is 13 m long, has a diameter of 5.9 m and weighs 12000 tons [163]. The radius of the magnet is sufficiently large to contain the inner tracker and the calorimeters, while outside of it is the flux return system and the muon detector. An advantage of this setup is the avoidance of energy losses of the particles before the calorimeters by interactions with the coil material. An axial and uniform magnetic field of 3.8 T is generated with currents up to 19 kA in NbTi wires that are kept at a temperature of 4.5 K by a liquid helium cooling system. The magnetic field is crucial for the momentum measurement of charged particles, since the measurement is based on the bending of their trajectories in the magnetic field. The combination of a high magnetic field and high precision on the spatial resolution of the tracker ensures a high momentum resolution. The direction of the curvature is also used to determine the electric charge of the particle.

5.2.2 Tracker System

The CMS tracker [164, 165] is a silicon detector with a sensitive area of over 200 m^2. The sensors are located inside the 3.8 T magnetic field provided by the magnet and are arranged in concentric cylinders and disks surrounding the interaction point of the LHC beams. The tracker system provides hits along the curved trajectories of charged particles with very high precision up to pseudorapidities $|\eta| < 2.5$. The tracker consists of two sub-detectors with independent cooling, powering, and read-out schemes. An overview over the two sub-detectors is shown in Fig. 5.3. A challenge for the operation and data reconstruction of the tracker is the high particle flux that induces radiation damage in the inner layers.

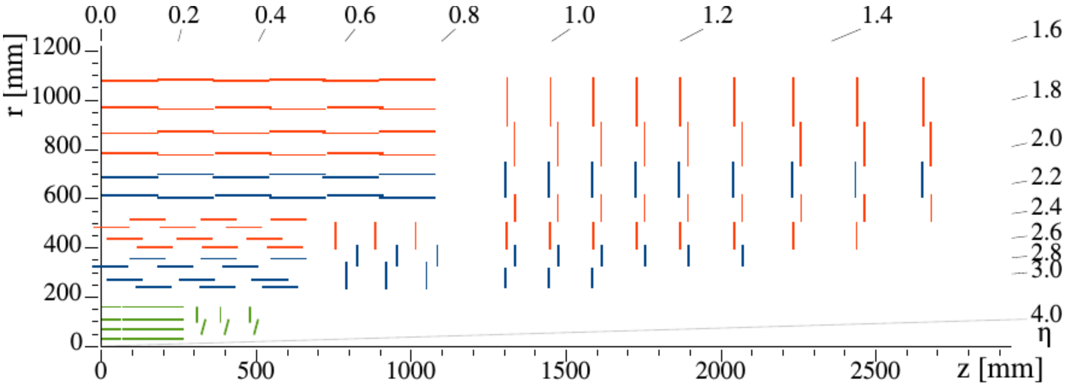


Figure 5.3: Sketch of one quarter of the Phase-1 CMS tracking system in the $r - z$ view. The pixel detector is shown in green, while single-sided and double-sided strip modules are depicted as red and blue segments, respectively [164].

The inner sub-detector is the pixel detector which has a surface area of 1.1 m^2. It is segmented into 66 million pixels of size 100 μm by 150 μm and thickness 285 μm. The detector is arranged in three layers in the barrel region at radii of 4.3 cm, 7.2 cm, and 11 cm, respectively, and two disks on each side of the barrel and the endcap regions at 34.5 cm and 46.5 cm from the interaction point.

The second sub-detector is the strip detector which surrounds the pixel detector. It is segmented into 9.6 million strips with thickness between 320 μm and 500 μm, and distances between the strips varying from 80 μm to 205 μm. The silicon strip lengths range from 10 cm to 20 cm. The detector is arranged in 4 layers in the inner barrel (TIB) and 6 in the outer barrel (TOB) at radii from 25 cm to 110 cm and up to 120 cm in the z direction. Moreover, it also includes 12 disks in the endcap region with radii up to 110 cm and in z up to 280 cm.

The p_T resolution of the tracker for charged hadrons with transverse momentum $p_\mathrm{T} < 20$ GeV is approximately 1%. With increasing p_T the resolution degrades approximately linearly. The charged particle tracks are used to reconstruct the positions of the primary and secondary vertices and the fine granularity allows the separation of closely-spaced particle trajectories in jets.

5.2.3 Electromagnetic Calorimeter

The electromagnetic calorimeter (ECAL) [166] allows to identify and reconstruct photons and electrons, and is valuable for the measurement of jets and missing transverse momentum. The total energy of electrons, positrons and photons is measured by detecting the light of low energy photons from electromagnetic showers. The light is produced in lead tungstate ($PbWO_4$) scintillating crystals and is detected and amplified by photodetectors located at the end of each lead tungstate crystal. The crystals are transparent to their entire scintillation emission spectrum and the choice of the crystals was made to obtain a high energy resolution by minimizing sampling fluctuations. In order to measure the energy with high precision, an accurate calibration of the calorimeter is necessary.

The CMS ECAL is a homogeneous calorimeter made of 75848 crystals and is located inside the CMS superconducting solenoid magnet. It is made up of a barrel section (EB) covering the central rapidity region (up to $|\eta| = 1.479$) and of two disks called endcaps (EE), which detect incident particles up to $|\eta| = 3$. The cylindrical barrel consists of 61200 crystals assembled into 36 supermodules, each weighing around three tonnes and containing 1700 crystals. The flat ECAL endcaps close the barrel at either end and consist of almost 15000 crystals. The front face of the barrel crystals has an area of 2.2×2.2 cm^2, and the endcaps have a front-face area of 2.9×2.9 cm^2. The material choice of $PbWO_4$ with a radiation length $X_0 = 0.89$ cm and a Molière radius $R_0 = 2.19$ cm results in a fine granularity calorimeter and ensures the radiation hardness necessary to handle the high particle flux at the LHC. The barrel and endcap crystal length of 23 cm and 22 cm corresponds to 25.8 and 24.7 radiation lengths respectively, which is sufficient to contain more than 98% of the energy of electrons and photons up to 1 TeV. Electrons and photons can be reconstructed up to $|\eta| < 2.5$, while jets are reconstructed up to $|\eta| = 3.0$. The relative energy resolution of the ECAL barrel as a function of the electron energy has been measured to [167]:

$$\frac{\sigma_E}{E} = \frac{2.8\%}{\sqrt{E/\mathrm{GeV}}} \oplus \frac{12\%}{E/\mathrm{GeV}} \oplus 0.3\% \; . \tag{5.6}$$

In addition to the EE and EB, a pre-shower sampling detector, consisting of two layers of lead absorber and two layers of silicon strip detectors, is placed in front of each endcap disk. The detector provides a very high granularity in the forward region and allows to measure the position of electromagnetic showers with high accuracy. The main purpose of the pre-shower detector is to indicate the presence of a photon or an electron in the front ECAL, and resolving photons from neutral pions to discriminate them from photons stemming from the primary interaction.

5.2.4 Hadronic Calorimeter

The purpose of the CMS hadronic calorimeter (HCAL) [168] is to measure the energy and position of hadrons, such as pions, kaons, protons, and neutrons. It is also a crucial component for the measurement of non-interacting, uncharged particles such as neutrinos via missing transverse energy. The hadronic calorimeter is located behind the tracker and the electromagnetic calorimeter and consists of four sections. A schematic overview of one quarter is shown in Fig. 5.4. The HCAL barrel (HB) and endcap (HE) calorimeters cover regions of $|\eta| < 1.3$ and $1.3 < |\eta| < 3.0$, respectively. They are both sampling calorimeters made of alternating layers of brass absorber and plastic scintillator tiles, with a hybrid photodetector (HPD) readout. The brass absorbers cause the deposition of the energy of secondary particles due

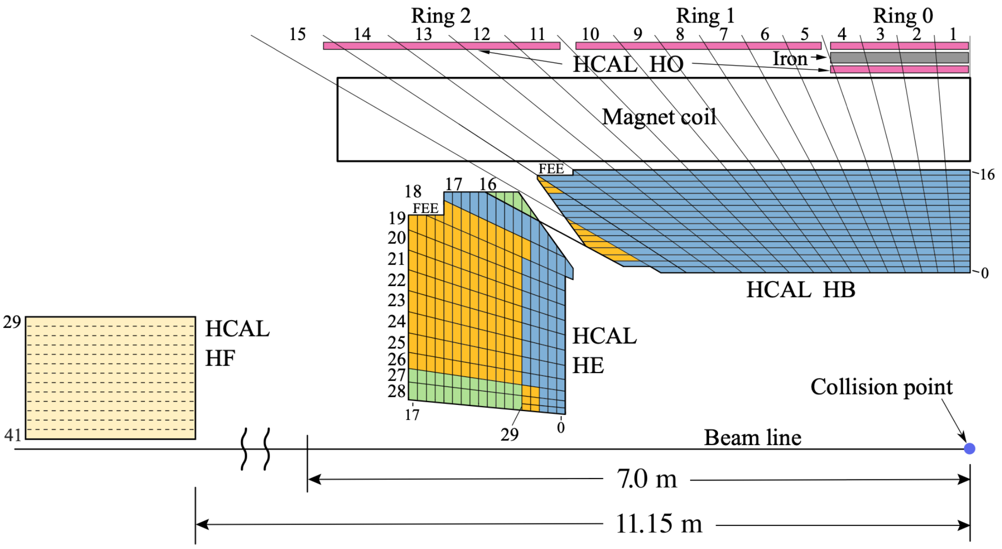


Figure 5.4: A schematic view of one quarter of the CMS HCAL, showing the positions of its four major components: the hadron barrel (HB), the hadron endcap (HE), the hadron outer (HO), and the hadron forward (HF) calorimeters [169].

to interactions with the nuclei of the material and the scintillators convert a part of this energy to visible light. The light from each tile produces an electric signal in the photodetector, which makes it possible to measure the total amount of deposited energy. In general, hadrons deposit energy in both ECAL and HCAL, thus the ECAL and HCAL need to be accurately calibrated to estimate the true hadron energy. The HCAL absorber thickness in the barrel is about six interaction lengths at normal incidence and about ten interaction lengths at larger pseudorapidities. Since the space inside the solenoid is limited and about eleven radiation lengths are required to absorb about 99% of the total energy of the hadrons, an outer detector (HO) made of plastic scintillator layers is placed outside of the solenoid, also referred to as tailcatcher. Including the ECAL, the total depth of the calorimeter system amounts to at least 12 interaction lengths in the barrel, and about 10 interaction lengths in the endcaps.

The combined ECAL and HCAL calorimeter energy resolution has been measured in a pion test beam [170]:

$$\frac{\sigma_E}{E} = \frac{110\%}{\sqrt{E}} \oplus 9\% \tag{5.7}$$

where E is expressed in GeV.

The HCAL Forward (HF) is located at $|z| = 11$ m covering the range $2.9 < |\eta| < 5$. It is a Cherenkov calorimeter that collects light with scintillating quartz fibres inserted in a steel absorber, and is read out with photomultiplier tubes. Due to the HF, nearly the full pseudorapidity is covered, and thus by measuring the energy of charged and neutral particles, an accurate estimate of the missing energy in the event is possible.

5.2.5 Muon System

The muon system [171] is located outside the solenoid and covers the range $|\eta| < 2.4$. Many of the signatures of the physics that CMS measures include muons, which are the only type of charged particles that can traverse through all other detector systems without a significant energy loss. The muon system consists of gaseous detectors sandwiched between the layers of the steel flux-return yoke that allow a traversing muon to be detected at multiple points along the track path. A schematic overview of the muon system is shown in Fig. 5.5. Three different types of gaseous detectors have been chosen depending on the uniformity and strength of the magnetic field, expected radiation fluxes and signal readout times: drift tube chambers (DTs), cathode strip chambers (CSCs), and resistive plate chambers (RPCs). Muons ionize the gas in the chambers, which causes electric signals that are detected at the wires

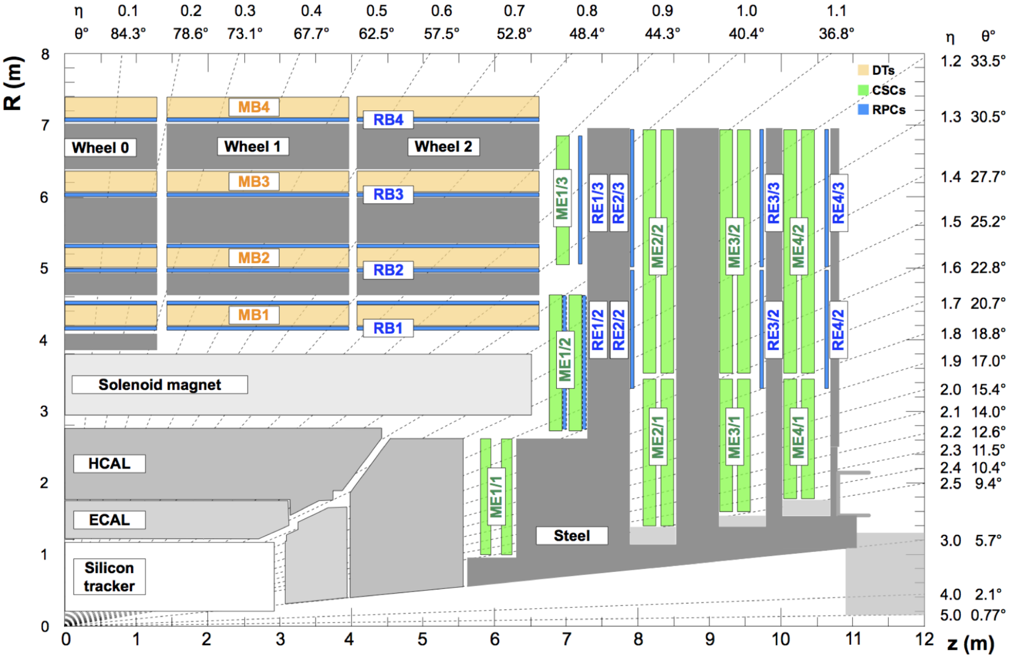


Figure 5.5: Quadrant of the CMS detector that shows the muon system, including resistive plate chambers, drift tube chambers and cathode strip chambers [171].

and strips. The DT and CSC chambers are located in the regions $|\eta| < 1.2$ and $0.9 < |\eta| < 2.4$, respectively, and are complemented by RPCs in the range $|\eta| < 1.9$. The chambers are arranged to maximize the coverage and to provide some overlap where possible. The DTs are segmented into drift cells and the position of the muon is determined by measuring the drift time to an anode wire of a cell with a uniform electric field, such that the drift velocity is approximately constant. The CSCs are multi-wire proportional counters with a finely segmented cathode strip readout. The position of the traversing muon is determined by combining information from the cathode strips and anode wires, which yields an accurate measurement of the $(R-\phi)$ coordinate at which the muon crosses the gas volume. The RPCs are double-gap chambers operated in avalanche mode, designed to provide timing information for the muon trigger. The inner tracker resolution dominates the muon momentum measurement up to a p_{T} of about 200 GeV, since the material of the calorimeters induces multiple scattering. For muons with p_{T} up to 100 GeV, matched to tracks in the silicon tracker, the relative p_{T} resolution is of 1% in the barrel and 3% in the endcaps. For muons with p_{T} up to 1 TeV the p_{T} resolution in the barrel is better than 7% [171, 172]. The muon system is the key element for high momentum muon measurements.

5.2.6 Trigger and DAQ

Only a small fraction of the proton-proton collisions contain events of interest to the CMS physics program, and due to the limited computing budget only a small fraction of the produced events can be stored for the offline analysis. The trigger system [173] selects the interesting events for offline storage from the bulk of the proton-proton collisions. To achieve this, the CMS trigger system utilizes two different levels. The first level exploits information from the calorimeters and muon detectors in custom hardware processors to select the most interesting events in a fixed time interval of less than 4 μs. The second level consists of the high level trigger (HLT), which further decreases the event rate by using the full event information, including the information from the silicon tracker. The thresholds of the first trigger level are adjusted during data taking depending on the value of the LHC instantaneous luminosity in order to keep the output rate below 100 kHz, which is the upper limit imposed by the CMS readout electronics. The HLT further reduces the output rate to 400 Hz for offline storage and improves the purity of the selected physics objects. The overall output rate of the L1 trigger and HLT can be adjusted by prescaling the number of events that pass the selection criteria of specific algorithms. Moreover, the trigger and data acquisition systems also provide information for the monitoring of the detector. The CMS experiment has also developed two new strategies at the high level trigger to search for new physics [174]. The first strategy, called Data Scouting, is based on event-size reduction rather than event filtering and can for example be used to search for low mass resonances. The second strategy, called Data Parking, refers to the technique of selecting events at the HLT and immediately moving them to tape storage in order to skip the prompt reconstruction, such that events can be stored on tape until there are sufficient computing resources to reconstruct them. In 2018, a large amount of data containing B hadrons was collected by CMS and parked for a delayed offline reconstruction.

5.3 Event Reconstruction

The offline event reconstruction for the CMS experiment is based on the Particle Flow (PF) algorithm [175]. The description of the event reconstruction will focus on the aspects relevant for the analysis of Run I CMS Open Data. The concept of the algorithm is a global event reconstruction by correlating basic elements (i.e. tracks and clusters) obtained from all sub-detector systems, in order to reconstruct all particles in the event and measure their properties. The particles are locally reconstructed in the different sub-detectors which provides the building blocks for the subsequent overall event description. In order to reconstruct the final state objects, possible superpositions of the signals are taken into account. The final objects that are then used in the CMS analyses are Particle Flow candidates such as jets, photons, electrons, charged and neutral hadrons, muons, taus, and missing transverse energy. The full detector information is also exploited for particle identification. Figure 5.6 depicts the interactions of several particle types with the sub-detector systems of the CMS detector.

A great challenge in the reconstruction of the physics objects is to mitigate pile-up effects. Pile-up refers to additional proton-proton interactions in the same bunch crossing. The particles produced in these interactions are also recorded in the detector and the pile-up interactions result in additional charged hadrons, photons, and neutral hadrons, which affects jets, missing transverse energy, the isolation of leptons and the identification of hadronic tau-lepton decays. Therefore, dedicated pile-up mitigation techniques have been implemented, such as the CHS and PUPPI algorithms [176]. The description of the algorithm is based on reference [175] which is also recommended for further reading.

5.3.1 Particle Flow Algorithm

The initial step of the CMS event reconstruction is the reconstruction of the trajectories of charged particles in the inner tracker. A combinatorial track finder based on Kalman Filtering (KF) is used to reconstruct these tracks in three stages. The first stage consists of the initial seed generation with a few hits that are compatible with a charged-particle trajectory. In the second step a trajectory is built by gathering hits from all tracker layers along this charged-particle trajectory and in the third step a final fit is performed to determine the origin, transverse momentum, and direction of the charged-particle. An additional operation is then applied, that removes candidate tracks that do not pass a quality threshold and removes possible duplicates. To increase the tracking efficiency while keeping the mis-reconstructed track rate low, the combinatorial track finder is applied in several iterations, each with moderate efficiency but with a high purity.

The reconstructed charged particle trajectories can be used to identify primary and secondary vertices by applying a custom algorithm for adaptive vertex fitting in

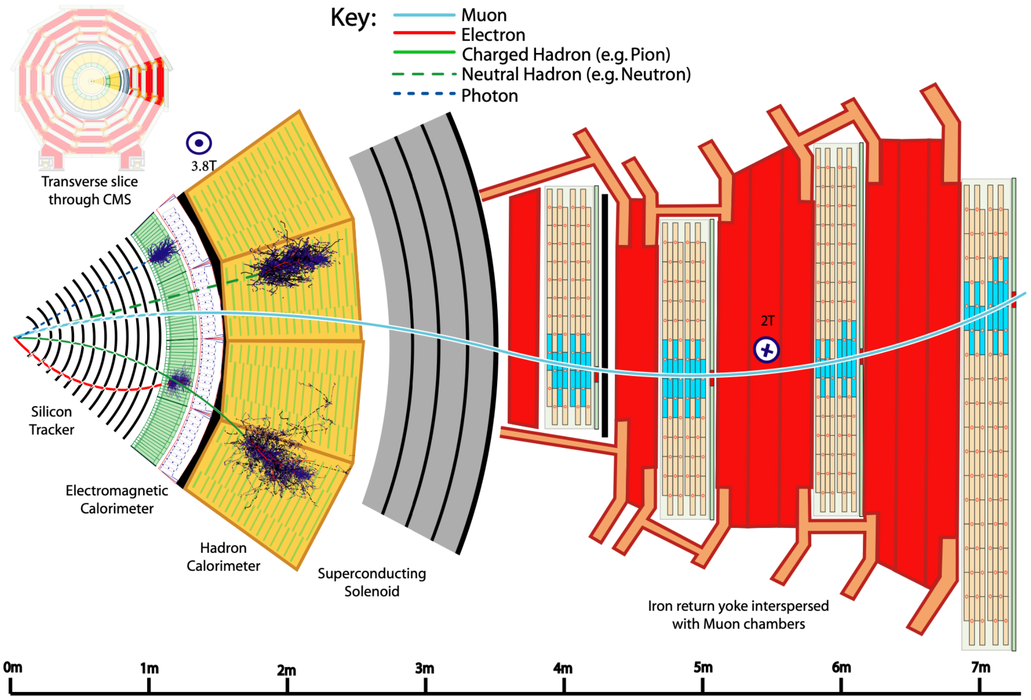

Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
Transverse slice through CMS
3.8T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0m
1m
2m
3m
4m
5m
6m
7m

Figure 5.6: Sketch of the particle interactions in a transverse slice of the CMS detector in each of the sub-detector systems [175].

combination with deterministic annealing [177]. The primary vertex has to satisfy several quality criteria. If an event contains multiple possible primary vertices, the one with the largest transverse momenta squared sum of objects associated with the vertex is chosen. The selection of a main primary vertex mitigates the effect of pile-up interactions by removing the contributions from particles linked to pile-up vertices. In recent years, novel algorithms based on modern machine learning techniques have been developed that potentially will improve the performance of tracking [178, 179].

For the clustering of the calorimeter detector readouts, a specific clustering algorithm was developed for the PF event reconstruction, that aims at a high detection efficiency also for low-energy particles and at separating close energy deposits. The clustering is performed separately in each sub-detector. In the first step cluster seeds are identified as cells with an energy greater than a given seed threshold, as well as greater than the energy of the neighbouring cells. The second step consists of building topological clusters from the seeds by aggregating cells with at least a corner in common with a cell already in the cluster and with an energy greater than a threshold. Finally, the clusters are reconstructed with an expectation-maximization algorithm based on a Gaussian-mixture model. The purpose of the clustering algorithm in the calorimeters is fourfold: it detects and measures the energy and direction of stable neutral particles such as photons and neutral hadrons, it separates these neutral particles from charged hadron energy deposits, it reconstructs and identifies electrons and all accompanying bremsstrahlung photons and additionally improves the energy measurement of charged hadrons for which the track parameters were not determined accurately, which often affects low-quality and high-p_{T} tracks.

A particle gives rise to several PF elements in the various CMS sub-detectors. Therefore, the particle is reconstructed with a link algorithm that connects the PF elements from different sub-detectors. The link algorithm is computationally expensive and is thus restricted to the nearest neighbours objects. The quality of the link is quantified by the distance between two linked elements. The link algorithm then constructs a PF block of elements associated either by a direct link or by an indirect link through common elements. First, the muon candidates are reconstructed in each PF block, and the corresponding PF elements are removed. Second, the electrons are reconstructed and isolated photons are identified in the same step. The remaining elements in the block are then identified as charged hadrons, neutral hadrons, and photons. In addition, secondary particles originating from nuclear interactions are reconstructed. The reconstruction process is rather conservative and usually additional selection criteria are specified during a CMS data analysis, depending on the concrete objective of the analysis. Recently, CMS has also started to use modern machine learning techniques for particle flow reconstruction that aim at a scalable, flexible full-event reconstruction [180].

5.3.2 Muons

The reconstruction of muons is not PF specific since the muon spectrometer allows to identify and reconstruct muon tracks with very high efficiency over all the detector acceptance. This is mainly due to the fact that the calorimeters absorb almost all particles except for muons and neutrinos. One of the main challenges of muon reconstruction includes the dismissal of cosmic muons, as well as the rejection of signals from highly energetic hadrons that produce a response in the muon detectors. Three different types of muon candidates can be defined depending on how they are reconstructed:

- **Stand alone muons**: are tracks that are built from segments reconstructed from local hits in the CMS muon spectrometer.
- **Global muons**: are identified by merging stand-alone tracks and inner tracks that are reconstructed in the inner tracker system. A combined fit is then performed and the transverse momentum is reevaluated.
- **Tracker muons**: are obtained by extrapolating inner tracks to track segments locally reconstructed in the muon detectors.

The PF algorithm applies an additional set of selection criteria to muon candidates reconstructed with the standalone, global, or tracker muon algorithms. Additionally, the PF algorithm also exploits information from muon energy deposits in ECAL and HCAL which further improves the identification performance. Several selection criteria are defined to balance the desired efficiency and purity during the subsequent analysis depending on the concrete requirements. The variables are based on track properties, such as the χ^2 or the number of hits per track, and on global properties, such as compatibility with the primary vertex. Based on these variables, a loose, medium and tight working point is defined for the muon identification that is commonly used in CMS physics analyses.
A particle flow isolation variable is calculated by evaluating the p_{T} of all charged hadrons $\mathrm{h}^{\pm}$, photons γ, and neutral hadrons h^0 within a ΔR cone between 0.3 and 0.5 around the direction of the muon. The PF isolation relative to the muon p_{T} is defined as:

$$I_{\mathrm{PF}} = \frac{1}{p_{\mathrm{T}}}\left(\sum_{\mathrm{h}^{\pm}} p_{\mathrm{T}}^{\mathrm{h}^{\pm}} + \sum_{\gamma} p_{\mathrm{T}}^{\gamma} + \sum_{\mathrm{h}^0} p_{\mathrm{T}}^{\mathrm{h}^0}\right) . \tag{5.8}$$

This isolation variable allows to select prompt muons produced in the electroweak decay of massive particles such as Z or W bosons, which for example can be used to identify leptons produced in jets through the decay of heavy-flavour hadrons or charged pions and kaons .

5.3.3 Electrons and Photons

The reconstruction of electrons is based on the combined information from the inner tracker and the calorimeters. Since both electrons and photons cause electromagnetic showers in the ECAL, the energy deposition patterns of electrons and photons are similar and therefore they are reconstructed together. In a given PF block, an electron candidate is seeded from a track if the corresponding ECAL cluster is not linked to three or more additional tracks. Photon candidates are seeded from an ECAL supercluster with E_{T} larger than 10 GeV if they are isolated from other tracks and calorimeter clusters in the event. The total energy of the ECAL clusters is corrected with analytical functions. The energy of the electrons is estimated by combining the information from the track and the corrected ECAL energy and the direction is chosen to be that of the track. In the case of photons the assigned energy is the corrected ECAL energy and the photon direction is taken to be that of the cluster and the primary vertex. Electron identification is done by applying selections to multiple variables based on track and ECAL cluster properties. Since 2012, BDT discriminators are trained separately in the ECAL barrel and endcaps acceptance with up to 14 of the variables, which improves the identification. During reconstruction, electron candidates are only required to satisfy loose identification criteria in order to ensure a high identification efficiency, with the drawback of a large mis-identification probability for charged hadrons. Typically, the electron identification is tightened in the physics analysis depending on the specific requirements of each analysis. Similar to muons, for electrons a PF isolation variable is calculated as described in equation 5.8 that can be used to identify electrons stemming from the primary interaction vertex.

5.3.4 Jets and b-Tagging

Jets are reconstructed with the anti-k$_{\mathrm{T}}$ algorithm [181], which is a collinear safe and infrared safe algorithm that allows to compare jet properties with theoretical QCD calculations. The algorithm clusters all particles reconstructed by the PF algorithm and the final state objects are referred to as “PF Jets”. The algorithm successively merges particles into clusters according to the distance d_{ij} between two particles i and j and the distance d_{iB} between the particle i and the beam B:

$$d_{ij} = \min\left(p_{Ti}^{2a}, p_{Tj}^{2a}\right)\frac{\Delta R_{ij}^2}{R^2} \quad \text{and} \quad d_{iB} = p_{Ti}^{2a} \tag{5.9}$$

where $p_{\mathrm{T}i}$ is the transverse momentum of the particle i, ΔR_{ij} is the distance as defined in equation 5.5, R is the radius parameter and $a = -1$ in the anti-k$_{\mathrm{T}}$ algorithm. The clustering algorithm proceeds by identifying the smallest of the distances between d_{ij} and d_{iB}. If it is d_{ij} then the physics objects i and j are combined together, else if it is d_{iB} then the object i is called “Jet” and is removed from the list of objects and the algorithm is repeated until no PF particles are left. The choice of the parameter R has to provide a balance between including all radiation from the initial parton and including noise from the underlying event. The radius of the parameter R can vary, and in CMS values of 0.4, 0.5 and 0.8 are chosen, where radii of 0.8 are used for final states with highly boosted particles to detect hadronic decays.

After the primary vertex has been reconstructed, all charged hadrons whose tracks are associated to a pile-up vertex are removed from the list of particles to be used in the jet reconstruction for the event. This procedure is called pile-up charged-hadron subtraction and denoted as CHS. The modern PUPPI algorithm [176] aims to use information related to local particle distribution, event pile-up properties, and tracking information to mitigate the effect of pile-up on observables of clustered hadrons. The jets need to be calibrated in order to have the correct energy scale which is achieved by the application of jet energy corrections (JEC) [182]. The energy scale and the momentum resolution of the jets is often one of the main sources of the systematic uncertainty, thus a detailed understanding of the corrections is crucial for many analysis. The jet energy corrections are calculated from MC simulations and are then tuned to data by combining several channels and data-driven methods. The JEC successively correct for the offset energy stemming from pile-up, the detector response to hadrons, and residual differences between data and MC simulation.

A key ingredient of many high precision measurements and searches for new physics is the identification of jets originating from heavy flavor quarks. The identification is based on the distinctive properties of heavy flavour hadrons. Typically, they have relatively large masses, long lifetimes, and decay particles with hard momentum spectra. If the hadrons are boosted, the long lifetime with a $c\tau$ of $\sim$ 0.5 mm for b-hadrons and $\sim$ 0.3 mm for c-hadrons results in displaced tracks with respect to the primary vertex that can be used to reconstruct secondary decay vertices. A sketch of this is depicted in Fig. 5.7 for a b-jet. The higher mass results in decay products with a larger transverse momentum relative to the jet axis compared to jets originating from light partons. Additionally, heavy hadrons possess a large branching ratio for semileptonic decays, hence the presence of soft leptons in the produced jets constitutes another feature that can be exploited. Flavour tagging techniques for bottom and charm hadrons, referred to as b-tagging or c-tagging, combine the information related to these properties to identify the flavour of the parton of a considered jet. Several different algorithms are used in CMS, that have been constantly improved in the last decade. The best performing algorithms are based on multivariate combinations of the available information [184]. The mis-identification versus efficiency curve

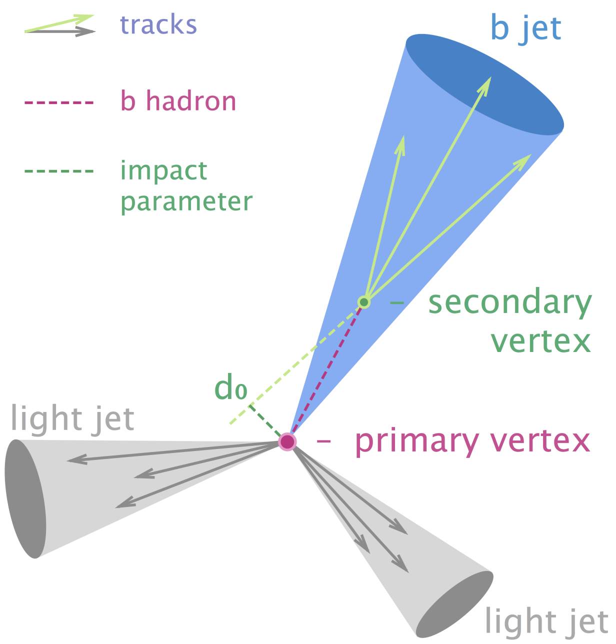


Figure 5.7: Schematic representation of the features of a b-jet that can be exploited for b-tagging. Diagram adapted under licence CC BY 4.0.

of the main b-tagging algorithms in Run I and Run II is shown in Fig. 5.8 [183]. The Jet probability algorithm (JP) is based on a calibrated estimation of the displaced track probabilities. The Combined Secondary Vertex (CSV) algorithm combines secondary vertex and track-based lifetime in a likelihood discriminator [184]. The improvement between different versions of the CSV b-tagging algorithm is due to the application of multivariate techniques and the inclusion of additional discriminating variables. The CMVA tagger combines the discriminator values of various taggers, which further improves the identification. The newest generation of b-tagging algorithms in CMS are based on modern deep-learning techniques, which improves the performance significantly. The DeepCSV approach starts from the same jet features as CSVv2, but extends the number of considered tracks per jet and exploits a deep neural network architecture. As discussed in Chapter 2.4.2, the DeepJet [185] algorithm uses a network architecture that does not only use a subset of the jet constituents but exploits the full information of all jet constituents, charged and neutral particles, secondary vertices, and global event variables simultaneously. The most sophisticated algorithms are based on GNNs, such as the ParticleNet algorithm [47].

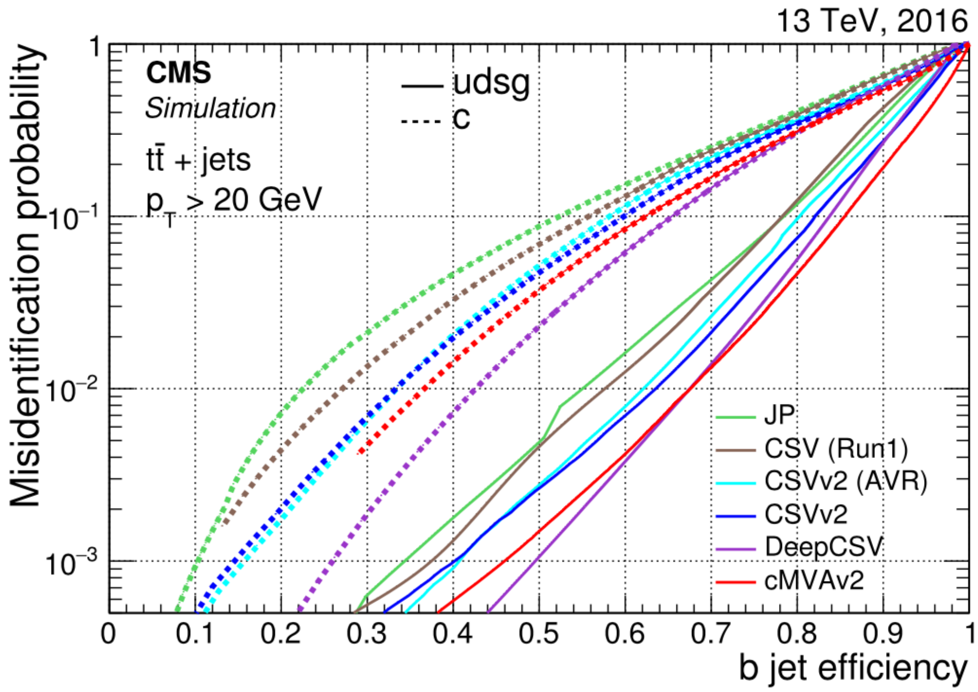


Figure 5.8: Mis-identification probability for c- and light-flavour jets versus b-jet identification efficiency for various CMS b-tagging algorithms measured in $t\bar{t}$ events [183].

5.3.5 Taus

The tau-lepton decay produces either a charged lepton and two neutrinos, or several hadrons and one neutrino, with the branching fractions given in Table 5.2. Hadronic tau decays, also denoted as τ_h, can be distinguished from quark and gluon jets by exploiting differences in the multiplicity, the collimation, and the isolation of the decay products. The PF algorithm is able to resolve particles produced in the decay of a tau-lepton and to reconstruct the surrounding particles to determine its isolation. In order to reconstruct and identify the hadronic decay products of tau leptons, the hadrons-plus-strips (HPS) algorithm [186] is applied to the reconstructed PF particles. The algorithm is seeded by jets with $p_T > 14$ GeV and $|\eta| < 2.5$. The jet constituents are combined into τ_h candidates corresponding to one of the main tau-lepton decay modes given in Table 5.2. To reconstruct the ρ and a_1 resonances, the HPS algorithm relies not only on photons but on so-called strips in order to include possible electrons and positrons from photon conversions. Strips are a collection of electrons or photons, that are collected around the most energetic electromagnetic particle found within the PF jet associated to the hadronic tau candidate in a win-

Decay mode	Meson resonance	$\mathcal{B}$[%]
$\tau^- \rightarrow \mathrm{e}^-\bar{\upsilon}_\mathrm{e}\upsilon_\tau$		17.8
$\tau^- \rightarrow \mu^-\bar{\upsilon}_\mu\upsilon_\tau$		17.4
$\tau^- \rightarrow \mathrm{h}^-\upsilon_\tau$		11.5
$\tau^- \rightarrow \mathrm{h}^-\pi^0\upsilon_\tau$	$\rho(770)$	26.0
$\tau^- \rightarrow \mathrm{h}^-\pi^0\pi^0\upsilon_\tau$	$\mathrm{a}_1(1260)$	10.8
$\tau^- \rightarrow \mathrm{h}^-\mathrm{h}^+\mathrm{h}^-\upsilon_\tau$	$\mathrm{a}_1(1260)$	9.8
$\tau^- \rightarrow \mathrm{h}^-\mathrm{h}^+\mathrm{h}^-\pi^0\upsilon_\tau$		4.8
Other modes with hadrons		1.8
All modes containing hadrons		64.8

Table 5.2: Branching fractions of the main τ decay modes [9].

dow with size 0.05×0.20 in the (η, ϕ) plane. The most energetic electromagnetic particle is associated to the strip and the four momentum is recalculated. The association is iterated until no further electromagnetic particle is found.

Each τ_h candidate is required to have a mass compatible with its decay mode. Collimated τ_h candidates are selected by requiring all charged hadrons and neutral pions to be contained in a cone of radius $\Delta R = (3.0\ \mathrm{GeV})/p_\mathrm{T}$ in the (η, ϕ) plane which is called the signal cone. The four-momentum of the τ_h candidate is determined by summing the four-momenta of its constituent particles. The following decay signatures are defined:

- **Single Hadron** which corresponds to $h^-\upsilon_\tau$ or $h^-\pi^0\upsilon_\tau$ with a low p_T π^0;
- **Hadron + Strip** which corresponds to $h^-\pi^0\upsilon_\tau$ with the two photons stemming from the π^0 being close on the calorimeter surface;
- **Hadron + 2 Strips** which corresponds to $h^-\pi^0\upsilon_\tau$ where the two photons stemming from the π^0 are well separated;
- **3 Hadrons** which is equivalent to h^-h^+h.

The loose, medium, and tight working points of the HPS algorithm refer to the level of isolation of the tau candidate. The isolation is defined by considering the additional charged hadrons or photons reconstructed in an isolation cone ($\Delta R = 0.5$) around the tau candidate. The probability for hadronic jets to be mis-tagged as tau jets can be reduced by the application of isolation discriminators. The newest generation of tau-identification algorithms exploits recent DNN multi-classification methods [187].

5.3.6 Missing Transverse Energy

Particles that do not interact with the detector material, e.g. neutrinos, are measured indirectly by missing transverse momentum. In the Particle-Flow algorithm the raw missing transverse momentum vector is computed as the sum of the transverse momenta of all PF reconstructed objects:

$$\vec{p}_{\mathrm{T,PF}}^{\,\mathrm{miss}}(\mathrm{raw}) = -\sum_{i=1}^{N_{\mathrm{particles}}} \vec{p}_{\mathrm{T},i} \; . \tag{5.10}$$

The jet-energy-corrected missing transverse momentum includes a term that replaces the raw momentum of each PF jet with momentum above 10 GeV by its corrected value:

$$\vec{p}_{\mathrm{T,PF}}^{\,\mathrm{miss}} = -\sum_{i=1}^{N_{\mathrm{particles}}} \vec{p}_{\mathrm{T},i} - \sum_{j=1}^{N_{\mathrm{PF}jets}} \left(\vec{p}_{\mathrm{T},j}^{\,\mathrm{corr}} - \vec{p}_{\mathrm{T},j} \right) \tag{5.11}$$

which improves the resolution of the missing transverse energy.

5.4 Event Simulation

Event simulations are crucial for the design and upgrade of the CMS detector, as well as to analyze and interpret the recorded data. Most importantly, simulations indicate which signature a new particle would leave in the detector if it existed. Typically, detector simulation consists of four steps: the first step is the generation of primary physics processes with dedicated MC generators, such as PYTHIA [133] and MadGraph [136]. The second step consists of the simulation of the interactions of a traversing particle with the detector and the resulting energy depositions in the detector, followed by the third step which is the digitization, where the energy deposits in the detector are converted into digital signals. The final step is the reconstruction, where the objects for physics analysis are reconstructed from the digital signals [188]. The CMS experiment uses a software package called CMSSW to produce Monte Carlo simulation events [189, 190]. Primary physics processes are generated by Monte-Carlo event generators, as discussed in Chapter 4.3. CMS has developed a full detector simulation framework (FullSim) that produces detailed simulations of the particle interactions within the CMS detector. It is based on the "GE-ometry ANd Tracking" package (GEANT4), which is a toolkit capable of describing complex geometry and the propagation of particles through materials and fields. It allows detailed modeling of detector geometry and particle interactions.

FullSim is dominantly used for SM processes in precision measurements or searches for new physics that require a very high accuracy. However, FullSim needs several minutes of simulation time per event. In particular, the detector simulation step is the most expensive in terms of CPU usage, consuming 40% of the total computing budget of CMS [191].

To mitigate this issue, CMS has developed a parametric fast simulation framework (FastSim) [192], which reduces the simulation time by approximately two orders of magnitude and the simulation and reconstruction time by a factor of ∼20, yet reproduces FullSim with ∼10% accuracy [193]. The main difference between FullSim and FastSim lies in the simulation step. While FullSim uses the exact detector geometry and detailed models for material interactions, FastSim makes use of a simplified geometry with infinitely thin material layers and simple parametrized interaction models. In the reconstruction step, FullSim makes use of the PF algorithm that is used to reconstruct the real CMS data. FastSim uses the standard reconstruction for calorimetry and muon systems, but a simplified reconstruction for tracking in order to reduce CPU time. The increasing LHC luminosity will have significant implications for the CMS computing budget [191] and pile-up will require the simulation of a higher number of events, thus the importance of fast simulations is expected to increase. The use of unsupervised machine learning techniques, such as Generative Adversarial Networks or Variational Autoencoders, may be a possibility to provide a reliable fast simulation alternative without relying on simplified parametrizations and could bring orders of magnitude of improvement [194].

6 Measurement of the $\mathrm{t\bar{t}} \rightarrow \tau_{\mathrm{h}} + \mathrm{jets}$ Cross-Section with CMS Open Data

In Chapter 3 several novel approaches have been discussed that address the problem of constructing summary statistics with machine learning in the presence of nuisance parameters. In order to convince large collaborations consisting of several thousand members of the usefulness and correctness of novel analysis methods, it is beneficial to test and benchmark the algorithms on datasets that are as realistic as possible. This motivates the development of a dataset that reproduces a realistic CMS analysis and is accessible by the public in order to facilitate comparisons between novel approaches to inference. In this chapter, the reproduction of a full historic CMS analysis with systematic uncertainty based on real Run I legacy data of the CMS experiment, available in the CERN Open Data portal, is described. The reproduced analysis has been published by the CMS experiment in 2013 and measures the $\mathrm{t\bar{t}}$ production cross-section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV [12]. In the context of testing and benchmarking the INFERNO algorithm, this analysis is of interest because it is dominated by systematic uncertainties and uses a neural network classifier to construct a summary statistic as input for the inference. Thus it constitutes a use case in which the INFERNO technique can possibly improve the precision of the measurement. The reproduction of the analysis is based on the research paper of the CMS Collaboration [12], who is the original author of this analysis, and the thesis in [195] that contains additional details. In the following this analysis will also be referred to as the "original analysis".

6.1 Motivation

The CMS measurement of the $\mathrm{t\bar{t}}$ production cross-section in the τ+jets channel uses events that contain one jet identified as a hadronically-decaying tau-lepton and at least four jets, where at least one is identified as a b-jet. The Feynman diagram of this process is shown in the left panel of Fig. 6.1. The branching fraction of a top-quark decay to a W boson and a b-quark is close to 100% in the Standard Model. In total, 9.8% of the produced $\mathrm{t\bar{t}}$ pairs are expected to lead to this final state. The $\mathrm{t\bar{t}}$ production cross-section at $\sqrt{s} = 7$ TeV measured in this analysis is:

$$\sigma_{tt} = 152 \pm 12 \text{ (stat.)} \pm 32 \text{ (syst.)} \pm 3 \text{ (lum.) pb} \tag{6.1}$$

which is consistent with the Standard Model prediction of $\sigma_{tt}^{SM} = 164 \pm 10$ pb. The data-simulation agreement of the neural network classifier used in the analysis is shown in the right panel of Fig. 6.1. In general, the exact reproduction of a Run I analysis is difficult due to changes in the software and different processing of simulation and data. For example, the b-tagging algorithms used by the CMS collaborations have been continuously improved and the algorithms recommended for reprocessed legacy data are different from the algorithms used during the first Run I measurements. Therefore, this work does not aim at an exact numerical reproduction of this $\mathrm{t\bar{t}}$ cross-section measurement with CMS Open Data, but uses the same methodology and shows that the measured cross-section is in agreement with the original analysis. While the motivation for this work is not measuring new physics, but rather testing a novel algorithm, it still is of interest to understand the historical physics motivation for this measurement. As discussed in Chapter 4.4, the measurement of the $\mathrm{t\bar{t}}$ production cross-section is an important test of the Standard Model,

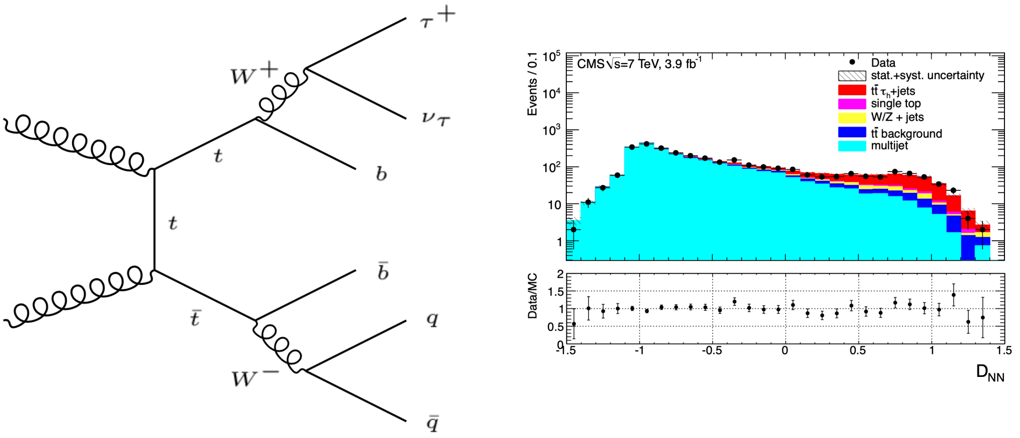


Figure 6.1: Left panel: Feynman diagram for the decay of a top pair into τ+jets. Right panel: data-simulation agreement for the neural network classifier used in the original analysis [12].

since the mass of the top-quark is of particular importance in many extensions of the Standard Model and the direct measurement of the $\mathrm{t\bar{t}}$ cross-section in the τ+jets final state offers the opportunity to investigate possible mass- or flavour-dependent couplings of the top-quark. Moreover, a hypothetical charged Higgs boson where the top-quark decays via $\mathrm{t} \rightarrow \mathrm{H^-b}$ and the charged Higgs boson subsequently via $\mathrm{H}^- \rightarrow \tau^-\bar{\nu}_\tau$, would result in an enhanced cross-section. A possible deviation from the SM expectation might thus occur in the $\mathrm{t\bar{t}} \rightarrow \tau + \mathrm{jets}$ decay channel [145].

6.2 CMS Open Data and Software

To make the analysis and the study of the INFERNO algorithm reproducible, the analysis is performed with CMS Open Data. The CMS experiment at CERN has released data from recorded proton-proton collisions at the LHC since 2014 that can be used for research purposes. Most of the data from the first LHC Run In 2010–2012 with the corresponding simulated samples are available in the CERN Open Data portal and tools to analyze them are provided [196]. The data have been published in the format and with the same data quality requirements that are used in official analyses of the CMS collaboration. The main format used in CMS for Run I data analysis is the Analysis Object Data (AOD) format, based on the ROOT framework [31] and processed with the CMS software CMSSW [197], which is also used for data taking, event reprocessing, and analysis, as well as for the production of MC simulations. The recorded proton-proton collisions are stored in “primary datasets” depending on the CMS trigger selections. An event typically contains

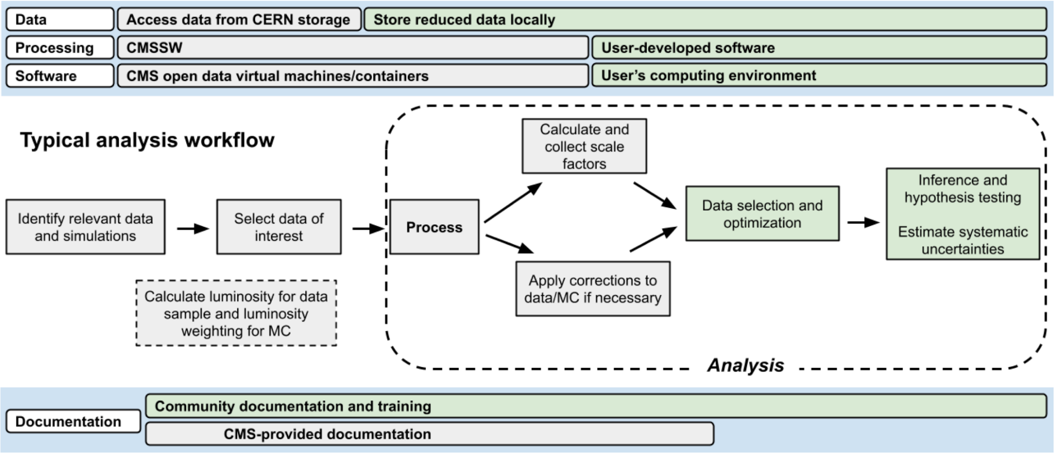

Data
Access data from CERN storage
Store reduced data locally
Processing
CMSSW
User-developed software
Software
CMS open data virtual machines/containers
User's computing environment
Typical analysis workflow
Identify relevant data and simulations
Select data of interest
Calculate luminosity for data sample and luminosity weighting for MC
Process
Calculate and collect scale factors
Apply corrections to data/MC if necessary
Data selection and optimization
Inference and hypothesis testing
Estimate systematic uncertainties
Analysis
Documentation
Community documentation and training
CMS-provided documentation

Figure 6.2: Typical workflow of a CMS analysis [196].

the data of one hard-scattering event and several pile-up events in the same beam crossing. The simulated datasets are generated by Monte Carlo generators, and the interactions of the produced particles with the CMS detector are simulated with CMSSW. Subsequently, additional events are added on top of the simulated process to emulate pile-up effects and the simulated events are processed into the same format as the collision data. In the analysis of the AOD format, values such as jet energy corrections or trigger information are accessible from a condition database. Moreover, a tool is provided that allows the evaluation of luminosity for specific event selections.

The workflow for the reproduction of the $\mathrm{t\bar{t}}$ cross-section measurement in the τ+jets channel follows the typical workflow of a CMS analysis as depicted in Fig. 6.2. The High Performance Computing system HTCondor at CERN is used to process the data and store the relevant information in a lightweight ROOT format, called NanoAOD [198] with the 5.3.32 version of the CMS software CMSSW. Processing all selected samples takes of the order of $\mathcal{O}(48h)$. The object and event selection is done with the awkward array package [199], which is a library for nested, variable-sized data using numpy-like idioms. The awkward array package allows to do columnar analysis that is significantly faster than using plain ROOT event loops. The coffea package [200], which is compatible with awkward array, is used to calculate several corrections, such as jet energy corrections, for the selected MC samples. Processing all preselected NanoAODs with systematic uncertainties takes of the order of $\mathcal{O}(1h)$. Further data pre-processing for the machine learning is done with the pandas package [201], while the machine learning models are implemented with PyTorch [202]. For visualization matplotlib [203] and ROOT are used. The final inference is done with the cabinetry [204] package based on the inference tool pyhf [33]. The CMS combine tool [205], which is based on the RooFit/RooStats package, is used to cross-check the results obtained with cabinetry. Most of the python packages for fast columnar analysis and inference have been developed in the last years and will be crucial for the development of HEP analysis in the future.

6.3 Data and Simulation

The choice of the data and simulation samples follows the choices made in the original analysis [12]. The used triggers, called QuadJet40_IsoPFTau40 and QuadJet45_IsoPFTau45, are part of the MultiJet primary dataset [206, 207] which is

Dataset	run range	trigger	$\mathcal{L}\left(\mathrm{pb}^{-1}\right)$
Run2011A [206]	160431 – 165969	QuadJet40_IsoPFTau40	357.5
Run2011A [206]	165970 – 166782	QuadJet45_IsoPFTau45	363.5
Run2011A [206]	166783 – 171049	QuadJet40_IsoPFTau40	514.7
Run2011A [206] Run2011B [207]	171050 – 178420	QuadJet45_IsoPFTau45	2930.2
Total Luminosity			4165.9

Table 6.1: Datasets with the chosen trigger, corresponding run numbers and luminosity. The version of the datasets is 12Oct2013-v1.

available in the CERN Open Data portal. The design of the triggers will be discussed in detail in the next section. Table 6.1 summarizes the chosen trigger and the corresponding integrated luminosity for the 2011 Run-A and Run-B data taking period. According to the original analysis [12], the QuadJet40_IsoPFTau40 trigger was prescaled by mistake for the runs 165970-166782, for this reason the QuadJet45_IsoPFTau45 trigger is used instead. Only part of the 2011 Run-B is used in the analysis since the QuadJet45_IsoPFTau45 trigger is prescaled from Run-178421 on. In total, about 80 % of the data have been recorded with the QuadJet45_IsoPFTau45 trigger. All selected runs have passed a set of data qualification criteria and are contained in a file of validated runs that is provided by the CMS collaboration [196]:
Cert_160404-180252_7TeV_ReRecoNov08_Collisions11_JSON.txt.
The luminosity is calculated with the official CMS tool brilcalc and the total integrated luminosity of the analyzed datasets sums up to $L = 4.16\ \mathrm{fb}^{-1}$. It yields a slightly higher luminosity compared to the luminosity quoted by the original analysis ($L = 3.9\ \mathrm{fb}^{-1}$).

The legacy Open Data Summer11 simulation provided by the CMS experiment [196] is used to estimate the signal efficiency and the efficiencies of the electroweak and $\mathrm{t\bar{t}}$ background processes, which include contributions from the full hadronic, lepton+jets, τ_{h}+lepton and $\tau_{\mathrm{h}}\tau_{\mathrm{h}}$ channels. Table 6.2 summarizes the considered simulated datasets that are available in the CERN Open Data portal, as well as the corresponding theoretical cross-sections. The acronym “st” denotes simulated single-

Process	σ(pb)	Dataset	Events
$t\bar{t}$	164 ± 10	TTJets-TuneZ2-TTeV-madgraph-tauola [208]	54,990,752
W + jets	31314 ± 1558	WJetsToLNu_TuneZ2_7TeV-madgraph-tauola [209]	78,347,691
Z + jets	3048 ± 132	DYJetsToLL_TuneZ2_M-50_7TeV-madgraph-tauola [210]	36,408,225
st (s)	2.76 ± 0.1	T_TuneZ2_s-channel_7TeV-powheg-tauola [211]	229,786
st (tW)	5.3 ± 0.6	T_TuneZ2_tW-channel-DR_7TeV-powheg-tauola [212]	744,859
st (t)	42.6 ± 2.4	T_TuneZ2_t-channel_7TeV-powheg-tauola [213]	3,249,552
s$\bar{t}$ (s)	1.52 ± 0.09	Tbar_TuneZ2_s-channel_7TeV-powheg-tauola [214]	139,258
s$\bar{t}$ (tW)	5.3 ± 0.6	Tbar_TuneZ2_tW-channel-DR_7TeV-powheg-tauola [215]	801,626
s$\bar{t}$ (t)	22.0 ± 0.9	Tbar_TuneZ2_t-channel_7TeV-powheg-tauola [216]	1,943,163

Table 6.2: Simulated datasets for the signal and background samples used in the CMS Open Data analysis. The values of the theoretical cross-section and the corresponding uncertainties are taken from [217, 218].

top events in the s-, $t-$ and tW-channel. The $\mathrm{t\bar{t}}$ signal and background events and the W/Z + jets events are simulated with the MADGRAPH [136] generator using the parton distribution function set CTEQ66 [129]. The parton showering, fragmentation, hadronization and decays of short lived particles, except tau-leptons, is simulated with PYTHIA [133]. The tau-leptons are decayed using TAUOLA [219]. Single-top events are simulated with POWHEG [220] interfaced to PYTHIA and TAUOLA. The used top-quark mass value is 172.5 GeV and the Next-to-Next-Leading-Log (NNLL) $\mathrm{t\bar{t}}$ cross-section is assumed to be 164 ± 10 pb [217].

6.4 Trigger

According to the original analysis [12], a dedicated multijet trigger was developed to record $\mathrm{pp} \to \mathrm{t\bar{t}} \to \tau_\mathrm{h}$+jets events. The trigger requires the presence of four calorimeter jets, one of them identified as a tau-lepton. It is based on two consecutively applied filters, referred to as jet and tau filters. Since the event rate increased with the rising instantaneous luminosity, two versions of the trigger have been developed: QuadJet40_IsoPFTau40 and QuadJet45_IsoPFTau45, where in the latter the p_T thresholds for the jets and the tau have been raised. The Level-1 decision of the trigger is based on the identification of four L1 jets with $p_\mathrm{T} > 20$ GeV ($p_\mathrm{T} > 28$ GeV starting from the beginning of Run2011B). The HLT decision consists of two steps:

1. the HLT-jet-filter requires the presence of four corrected calorimeter jets with $p_\mathrm{T} > 40$ GeV (respectively $p_\mathrm{T} > 45$ GeV for the more stringent trigger) and $|\eta| < 2.5$;
2. the HLT tau filter requires the presence of one isolated particle-flow HLT tau with $p_\mathrm{T} > 40$ GeV (respectively $p_\mathrm{T} > 45$ GeV for the more stringent trigger), with $|\eta| < 2.5$ and with leading track $p_\mathrm{T} > 5$ GeV. The tau candidate has to be matched to one of the four trigger jets within $\Delta R = 0.4$.

The efficiencies have been reevaluated with datasets available in the CERN Open Data portal. The direct measurement of the trigger efficiency is impossible due to the large QCD background which makes it difficult to select a pure $\tau_\mathrm{h}+3$ jets sample in data. The efficiency of the trigger is thus evaluated by measuring the efficiencies for the jets and the tau-lepton separately. The single-jet efficiency has been measured with the 2011 SingleMu primary dataset of the CMS Open Data [221, 222] in events selected with a single muon trigger (HLT_mu15, HLT_mu20, HLT_mu24,

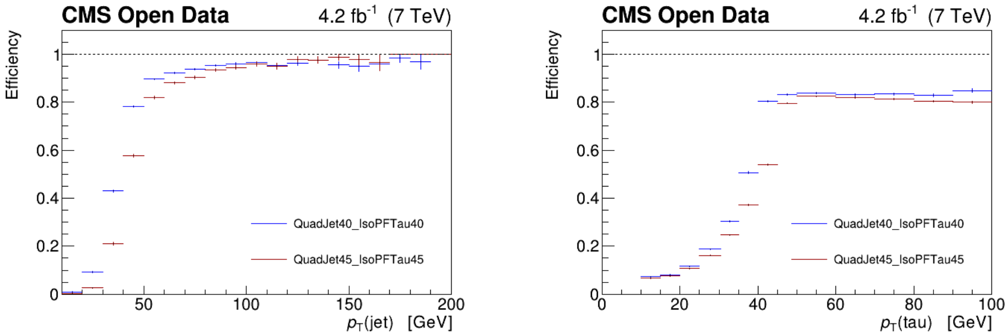


Figure 6.3: Left panel: measured single-jet efficiency for both multijet trigger versions. Right panel: measured tau-lepton efficiency for both multijet trigger versions.

HLT_mu30). The presence of four central particle-flow jets with $|\eta| < 2.5$ is required that are matched to jet objects used in the HLT-jet-filter within $\Delta R = 0.4$ to satisfy the HLT-jet-filter requirement. Three of the central jets are required to have $p_\mathrm{T} > 70$ GeV and the fourth jet is used as a probe jet to be matched to one of the HLT jets in the HLT filter within $\Delta R = 0.4$. The efficiency ϵ is then calculated from the probe jets passing and failing the requirement as:

$$\varepsilon = \frac{N_\text{passing}}{N_\text{passing} \ + N_\text{failing}} \ . \tag{6.2}$$

The left panel of Fig. 6.3 shows the obtained efficiency per single-jet for the two trigger versions QuadJet40_IsoPFTau40 and QuadJet45_IsoPFTau45. The jet trigger plateau is reached above 120 GeV due to the different energy scale of particle-flow jets and calorimeter jets.

The tau trigger efficiency is measured in events of the 2011 MultiJet primary dataset available in the CERN Open Data portal [206, 207]. The events are required to contain four particle-flow jets matched to the HLT-jet objects in the HLT-jet-filter within $\Delta R = 0.4$ in order to ensure that the HLT-jet-filter has fired. In addition, the events are required to contain exactly one HPS tau-lepton with medium isolation matched to one of the four selected particle-flow jets within $\Delta R = 0.4$. The selected tau-lepton is then used as a probe to be matched to the HLT-tau used in the HLT-tau-filter within $\Delta R = 0.4$ and the efficiency is calculated with equation 6.2. The right panel of Fig. 6.3 shows the tau trigger efficiency for the two multijet trigger versions. The trigger plateau is reached for $p_\mathrm{T} > 45$GeV (respectively $p_\mathrm{T} > 50$GeV for the more stringent trigger) yielding an efficiency of $\approx 85\%$.

For the simulated samples the trigger efficiency is calculated by multiplying the trigger efficiencies of the three most energetic central jets and the trigger efficiency of the tau-lepton candidate. The simulated events are weighted randomly by the QuadJet40_IsoPFTau40 and QuadJet45_IsoPFTau45 trigger efficiency according to the integrated luminosity fraction of the two trigger versions. For recorded data the offline reconstructed jets and tau-lepton are required to be matched within $\Delta R < 0.4$ to the jet and tau objects in the HLT in order to have similar conditions in data and simulation.

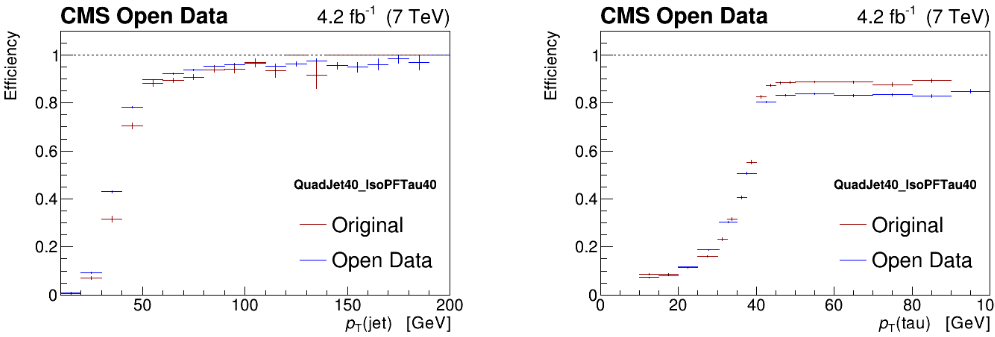


Figure 6.4: Left panel: comparison of the jet p_T spectrum after applying similar selection criteria between the original analysis [12] and the CMS Open Data analysis. Right panel: comparison of the tau-lepton p_T spectrum after applying similar selection criteria between the original analysis [12] and the CMS Open Data analysis.

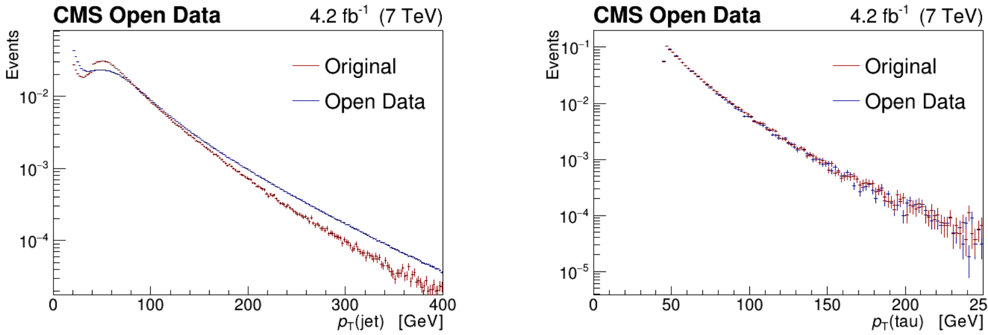


Figure 6.5: Left panel: jet trigger efficiency comparison between the reproduced analysis with CMS Open Data and the original analysis [12]. Right panel: tau-lepton trigger efficiency comparison between the reproduced analysis with CMS Open Data and the original analysis [12].

A comparison of basic kinematic distributions of the selected jets and the selected tau-lepton has been performed between the reproduced analysis with CMS Open

Data and the original analysis, where the same selection criteria have been applied. The comparison of the p_T spectra is shown for the selected jets in the left panel of Fig. 6.4 and for the tau-lepton in the right panel. While the tau-lepton p_T spectra are very similar, the distribution of the selected jets shows some differences. A possible explanation are differences in the CMS software versions used to reconstruct the data for the original analysis (CMSSW with version 4.2) and the CMS Open Data analysis (CMSSW with version 5.3), as well as differences in the applied jet energy corrections.

A comparison of the jet trigger efficiencies and the tau trigger efficiencies between the original analysis and the CMS Open data analysis indicates that the different p_T spectra result in slightly different trigger efficiencies. A comparison of the jet and tau trigger efficiencies for the QuadJet40_IsoPFTau40 trigger is shown in the left and right panel of Fig. 6.5. The jet trigger efficiencies are slightly higher in the CMS Open Data analysis, while the tau trigger efficiencies are slightly lower. A similar comparison for the more stringent trigger QuadJet45_IsoPFTau45 is included in Appendix A.1, where a similar trend is observed.

6.5 Event Selection

The event selection follows closely the original analysis [12] and requires the presence of at least four particle-flow jets, and the presence of one particle-flow tau-lepton candidate reconstructed with the HPS algorithm. One of the selected particle-flow jets is required to be b-tagged. The object reconstruction and the particle-flow algorithm [175] are discussed in detail in Chapter 5.3.

6.5.1 Vertex Selection

The events are required to contain at least one primary vertex fulfilling several quality criteria. The number of degrees of freedom of the vertex fit is required to be at least 4, $n_{dof} > 3$ and the z coordinate of the vertex has to be located inside the detector center, i.e. $z(PV) < 24$ cm. Moreover, the radial coordinate of the primary vertex w.r.t. the beam line is required to be smaller than 2 cm, $\rho(PV) < 2$ cm and the vertex may not be identified as a fake vertex. A primary vertex is identified as a fake vertex if it only consists of the beamspot and does not include any tracks in the fit.

6.5.2 Tau Selection

The hadronically decaying tau-lepton candidate is reconstructed with the HPS algorithm [186] that is described in Chapter 5.3. The tau candidates are required to be isolated: the sum of the transverse energies of the charged hadrons and photons reconstructed in an isolation cone of $\Delta R = 0.5$ around the tau candidate is required to be less than 1 GeV. This is referred to as the "Loose Isolation" working point. The leading track of the tau candidate is vetoed if it is identified as a muon in order to suppress the muon contamination. In addition, the charged tau candidate may not be identified as a minimum ionising particle, therefore the ratio of the sum of the energy deposits in the ECAL and HCAL calorimeters associated to the tau candidate over the leading track momentum is required to be larger than 0.2. To be consistent with the trigger conditions, the transverse momentum of the tau candidate is required to fulfill $p_{\mathrm{T}} > 45$ GeV and the tau candidate is required to be matched within $\Delta R < 0.4$ to the tau object used in the HLT. The pseudorapidity of the tau candidate is required to be in the range $|\eta(\tau_{\mathrm{h}})| < 2.3$ and $|\eta(\tau_{\mathrm{h}})| \notin [1.444, 1.566]$ in order to exclude the barrel-endcap transition region of the electromagnetic calorimeter. The transverse momentum of the leading track of the tau is required to fulfill $p_{\mathrm{T}} > 10$ GeV. To ensure that the tau candidate originates from the collision vertex, the z coordinate of the tau has to fulfill $|z_{\mathrm{vtx}}(\tau_{\mathrm{h}}) - z_{\mathrm{PV}}| < 1$ cm and the impact parameter d_0 of the leading track w.r.t to the beam spot has to fulfill $|d_0| < 0.04$ cm.

6.5.3 Lepton Veto

In order to suppress the misidentification of electrons and muons as tau candidates, a veto on the presence of loosely isolated electrons and muons is applied. The isolation requirement for the leptons, defined in equation 5.8, is $I/p_{\mathrm{T}} < 0.15$, where I is the sum of the transverse energy deposits in the ECAL and HCAL calorimeters and p_{T} is the scalar value of the track momenta within a cone of $\Delta R = 0.3$. For muons, it is further required that the transverse momentum fulfills $p_{\mathrm{T}} > 10$ GeV and the pseudorapidity $|\eta(\mu)| < 2.4$. It is also required that the muon is identified as a global muon. Moreover, it is required that the z coordinate of the muon fulfills $|z_{\mathrm{vtx}}(\mu) - z_{\mathrm{PV}}| < 1$ cm. For electrons it is required that the transverse momentum fulfills $p_{\mathrm{T}} > 15$ GeV and the pseudorapidity $|\eta(e)| < 2.5$. As for the muons, the z coordinate has to fulfill $|z_{\mathrm{vtx}}(e) - z_{\mathrm{PV}}| < 1$ cm. If any lepton is identified as such, the event is discarded.

6.5.4 Jet Selection and Transverse Missing Energy

The jets are reconstructed with the particle-flow algorithm and the anti-$\mathrm{k_T}$ clustering algorithm [181] with a distance parameter $R = 0.5$. Selected events are required to have at least four particle-flow jets with pseudorapidity $|\eta| < 2.4$. Jets overlapping with leptons within $\Delta R(\mathrm{jet}, \mathrm{lepton}) > 0.4$ are excluded. To be consistent with the trigger design, three jets are required to have $p_\mathrm{T} > 45$ GeV and the fourth jet is required to have $p_\mathrm{T} > 20$ GeV. The jet candidates are required to be matched to jet objects used in the HLT within $\Delta R < 0.4$. The jet energies are corrected for the $L1FastJets$, $L2Relative$, $L3Absolute$ prescriptions [223]. To account for differences in the jet energy resolution of the order of 10% between simulation and data, also the $L2L3Residuals$ correction is applied. Additionally, for simulated events the jet energy resolution is corrected and smeared following the recommendations of the CMS Collaboration [196]. The transverse missing energy (MET) is obtained with the particle-flow algorithm and the jet energy scale corrections are propagated. A selection on the transverse missing energy, MET > 20 GeV, is applied to reject QCD background.

6.5.5 B-Jet Identification

The selected events are required to contain at least one b-tagged jet. For the 2011 legacy simulations, the recommended Combined Secondary Vertex algorithm (CSV) is used at its medium working point [184]. The CSV algorithm combines information from secondary vertices and track impact parameters in a tagging variable to discriminate between jets originating from b-quarks and those from other sources. The b-tagging efficiency is measured in simulated $\mathrm{t\bar{t}}$ events obtained from the CERN Open Data portal and shown in Fig. 6.6 for the medium working point. To mitigate differences between simulation and data, b-tagging scale factors (SF) have to be

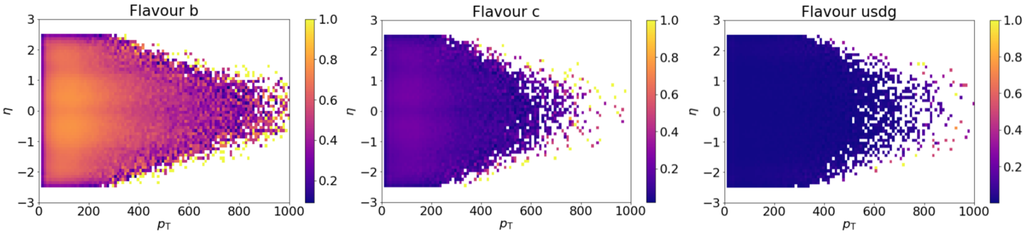


Figure 6.6: B-tagging efficiency for the CSV algorithm, measured in $\mathrm{t\bar{t}}$ simulated events, as a function of η and p_T for bottom (left), charm (middle) and light (right) quarks.

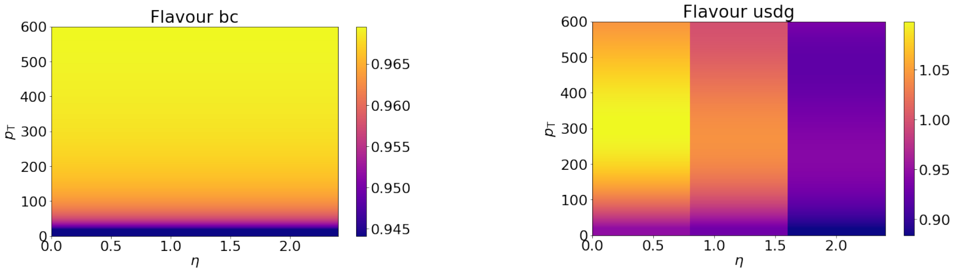


Figure 6.7: B-tagging scale factors for the CSV algorithm as a function of η and p_T for b- and c-quarks (left) and light quarks (right).

applied to the simulated samples. The scale factors, displayed in Fig. 6.7, are published by the CMS Collaboration [196] and depend on the transverse momentum and the pseudorapidity. SF_b, SF_c, SF_l denote the scale factor for b-jets, c-jets and light-jets respectively. The scale factors for b- and c-quarks, SF_b and SF_c are assumed to be equal. The calculation of the b-tagging event weight follows a probabilistic approach instead of directly selecting the simulated events depending on the value of the b-tagging discriminant. Following the recommended prescription provided by the CMS Collaboration [196], the probability that a jet i is selected depending on the b-tagging discriminant is given by:

$$P_i = \mathrm{SF}_i \cdot \mathrm{Eff}_i^{\mathrm{MC}} \tag{6.3}$$

where $\mathrm{Eff}_i^{\mathrm{MC}}$ is the b-tagging efficiency measured in simulated $t\bar{t}$ events and SF_i the scale factor associated to the jet. SF_i depends on the p_T and η of the jet as well as on its flavour. The probability that the event does not contain a b-tagged jet is then defined by:

$$P(0\ tag) = \Pi_i(1 - P_i) \tag{6.4}$$

where the product includes all selected jets in the event. The probability that the event contains at least one b-tagged jet is then given by:

$$P(\geq 1\ tag) = 1 - P(0\ tag)\ . \tag{6.5}$$

Therefore in order to select events with at least one b-tagged jet, the simulated events are weighted by $P(\geq 1\ tag)$.

6.5.6 QCD Background

The dominating background in this analysis are high multiplicity multijet events where one of the jets is misidentified as a hadronic tau-lepton. The smaller contributions from the electroweak processes are estimated from simulated events and are normalized to the theoretical cross-section and the total integrated luminosity. Since accurate simulations of the multijet background at the LHC are difficult, a data-driven approach is used. The multijet background is estimated by applying the same selection criteria to the data sample as described above and inverting the b-tagging requirement, i.e. vetoing the presence of a b-tagged jet selected with the CSV algorithm. To account for the b-tagging efficiency, the multijet events in data are weighted by the misidentification probability $P(\geq 1\ mistag)$ to select at least one b-jet in the event. This probability is computed with the probabilistic approach described in the previous section, where the mis-tagging efficiency for light jets is assumed for P_i:

$$P(\geq 1\ mistag) = 1 - P(0\ tag)\ . \tag{6.6}$$

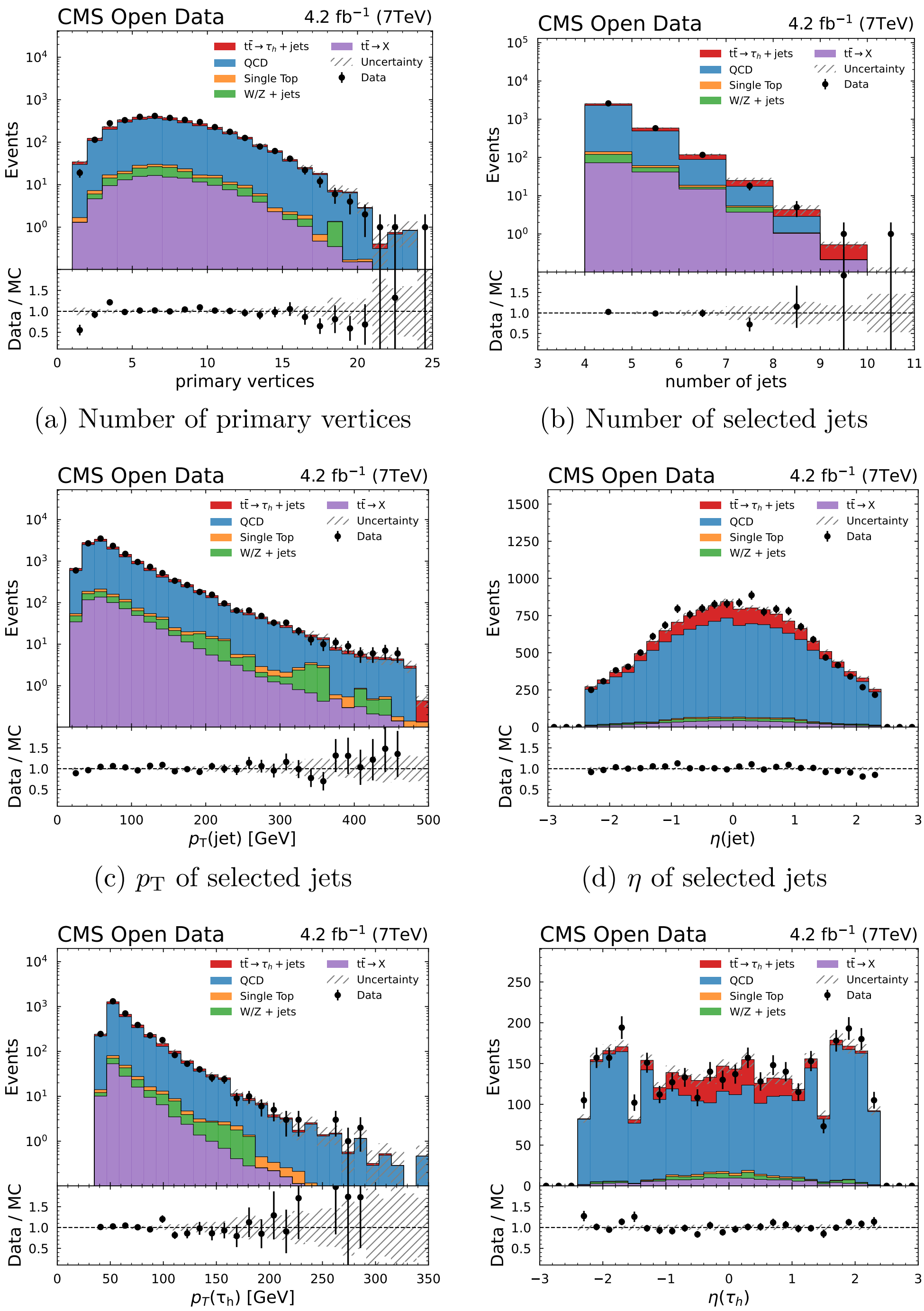
The original analysis has verified with simulations that the resulting sample contains less than 0.6% of $\mathrm{t\bar{t}}$ signal events, less than 0.3% $\mathrm{t\bar{t}}$ background events and less than 2% W/Z + jets events [12]. This indicates that the sample provides a good representation of the multijet background and is suitable for the training of a multivariate classifier.

6.6 Event Yields

Selection	Run A	Run B	$\mathrm{t\bar{t}} \to \tau_\mathrm{h}$+jets	$\mathrm{t\bar{t}} \to$ X
4 jets + τ_h	14408	10394	461.8 ± 2.0	339.4 ± 1.7
e, μ veto	14126	10210	452.7 ± 2.0	239.6 ± 1.4
MET	7977	6522	414.1 ± 1.9	164.1 ± 1.2
$\geq$ 1 b-tag	1846	1477	348.5 ± 1.7	134.0 ± 1.1
Selection	W + jets	Z + jets	st (tW)	$\mathrm{s\bar{t}}$ (tW)
4 jets + τ_h	286.7 ± 18.3	163.9 ± 6.2	18.2 ± 0.7	19.0 ± 0.7
e, μ veto	260.6 ± 17.4	122.0 ± 5.3	16.2 ± 0.7	17.0 ± 0.7
MET	242.9 ± 16.8	95.8 ± 4.7	13.8 ± 0.6	14.6 ± 0.6
$\geq$ 1 b-tag	44.1 ± 7.1	19.5 ± 2.1	10.5 ± 0.6	11.3 ± 0.6
Selection	st (t)	$\mathrm{s\bar{t}}$ (t)	st (s)	$\mathrm{s\bar{t}}$ (s)
4 jets + τ_h	8.6 ± 0.6	4.5 ± 0.4	1.0 ± 0.2	0.4 ± 0.1
e, μ veto	8.3 ± 0.5	4.3 ± 0.4	1.0 ± 0.2	0.4 ± 0.1
MET	5.6 ± 0.4	3.2 ± 0.3	0.8 ± 0.2	0.3 ± 0.1
$\geq$ 1 b-tag	4.8 ± 0.4	2.7 ± 0.3	0.7 ± 0.2	0.2 ± 0.1

Table 6.3: Expected number of events for the simulated signal and background samples for an integrated luminosity of $L = 4.16\ \mathrm{fb}^{-1}$ and the number of selected events in data. The uncertainties are statistical only.

In Table 6.3 the event yields for the various samples are shown at different steps of the event selection, taking into account trigger weights and b-tagging weights. The simulated events are normalized to the theoretical cross-sections and an integrated luminosity of $L = 4.16\ \mathrm{fb}^{-1}$. The data-simulation agreement is shown in Fig. 6.8 for the number of primary vertices and for several kinematic distributions of the selected jets and the hadronic tau-lepton . The normalization of the data-driven QCD sample is set to the best fit value after performing the log-likelihood fit described later in Section 6.9, which will be the case for all data-simulation plots shown in this thesis. In general, a good data-simulation agreement is observed. For the further analysis,



(a) Number of primary vertices

(b) Number of selected jets

(c) p_{T} of selected jets

(d) η of selected jets

(e) p_{T} of selected tau-lepton

(f) η of selected tau-lepton

Figure 6.8: Data-simulation agreement for the number of primary vertices and kinematic variables of the selected jets and the tau-lepton candidate.

the W + jets and Z + jets samples are combined into one W/Z + jets sample and the different single-top channels are combined into one single-top sample. Table 6.4 shows a comparison of the pre-fit event yields between the analysis reproduced with CMS Open Data and the events selected by the original analysis. Comparing the event yields between both analysis shows that there are slight differences in the number of events for the relevant processes. Possible reasons for the discrepancies of the numerical values are the slightly different trigger efficiencies discussed in Section 6.4, the different CMSSW versions used to reconstruct the data, different jet energy corrections, and also the b-tagging algorithm that has been changed from the Jet Probability (JP) algorithm for the original analysis to the CSV algorithm in the CMS Open Data analysis, since the JP algorithm is not supported for the 2011 legacy data. Moreover, it is also possible that slightly different event selections have been applied that were not obvious from the documentation.

Source	Original	CMS Open Data
Signal $\mathrm{t\bar{t}} \to \tau_{\mathrm{h}}$+ jets	383	348
Background $\mathrm{t\bar{t}} \to \mathrm{X}$	151	134
WZ+Jets	83	64
Single-Top	41	30
QCD	2392	2690
Total backgrounds	2667	2918
Data	3050	3323

Table 6.4: Pre-fit event yield comparison between the original analysis [12] and the CMS Open Data analysis.

6.7 Systematic Uncertainties

The calculation of the systematic uncertainties follows the list of relevant systematic sources considered in the original analysis [12] that is shown in Table 6.5. However, in some cases the updated procedures recommended for the use of the CMS Open Data datasets are applied [196]. The main sources of systematic uncertainties in the CMS Open Data analysis are due to the PDFs, the jet energy scale (JES), the tau energy correction and the tau identification. The original analysis considers several alternative samples for the $t\bar{t}$+jets sample. However, these samples are not yet present in the CERN Open Data portal; therefore, the values from the original analysis are quoted. According to the original analysis these variations only affect the normalization, but not the shape of the distributions [12]. The following table describes the systematic uncertainties considered in the CMS Open Data analysis.

Source	Rel. uncert. [%]	Shape
Cross-section uncertainty	±3	X
Top-quark mass	±3	X
Renormalization / factorization scale	±2	X
Parton shower matching	±3	X
PDF	±5	✓
τ_h trigger efficiency	±7	X
Pile-up	$+5-1$	X
τ_h energy scale	±7.	✓
τ_h identification	±9	X
Jet energy scale	±11	✓
Jet energy resolution	±2	✓
Unclustered MET	±7	X
B-tagging	±3	✓
Multijet background reweighting	±5	X
Syst. uncert.	±21	X
Stat. uncert. from fit and MC samples	±8	X
Stat. uncert. from trigger	±0.4	✓
Total stat. uncert.	±8	X

Table 6.5: Relative systematic uncertainties in the cross-section measurement of the original analysis [12].

Source	Description
Cross-section	The uncertainty on the theoretical cross-sections of the different simulated processes is taken from [217, 218], following the choices of the original analysis.
PDF	The uncertainty due to the CTEQ66 proton PDF [224] on all of the simulated signal and background processes has been estimated by combining the uncertainties of the twenty-two alternative PDF variations by summing the varied 1σ-up and 1σ-down PDF weights in quadrature. This procedure yields alternative 1σ-up and 1σ-down event weights that allow to reweight the nominal distributions.
Luminosity	The uncertainty on the luminosity measurement is estimated to 2% [225].
Tau trigger	Following the original analysis, a ± 5% uncertainty is accounted for the tau-leg trigger efficiency measurement, in order to take into account the fact that the used reference sample to estimate the tau trigger efficiency consists mainly of jets misidentified as tau-lepton candidates. This uncertainty is derived from tau-lepton candidates in $Z \rightarrow \tau^+\tau^-$ events with similar trigger conditions [12].
Tau ID	The uncertainty corresponding to the tau identification efficiency has been measured to 6% [226].
Statistical trigger efficiency	The estimation of the statistical uncertainty corresponding to the trigger efficiency for the particle-flow jets and the particle-flow tau is done by recalculating the trigger weight with a $\pm 1\sigma$ statistical variation of the efficiencies of the jets and the tau for all signal and background processes. This yields an alternative 1σ-up and 1σ-down event weight.

JES /JER	The uncertainty due to the jet energy scale JES and jet energy resolution JER are estimated for the simulated background and signal processes according to the prescription described in [223]. The uncertainty corresponding to the JES is estimated by shifting the jet energy up and down by the uncertainties corresponding to one standard deviation. For the jet energy resolution the distribution of the jet energy has been smeared by one standard deviation. The corrections are propagated to the missing transverse energy measurement. The event selection is repeated for each of the variations which yields alternative up and down datasets that correspond to the $\pm 1\sigma$ JES and JER uncertainty.
Tau energy scale	The uncertainty of the tau energy correction is estimated for the simulated background and signal processes by shifting the value of the tau energy up and down by ± 3% [227]. The corrections are propagated to the missing transverse energy. As for the JES, the event selection is repeated with the varied energy and an alternative 1σ-up and 1σ-down dataset is obtained.
B-tagging	The uncertainty due to the application of the b-tagging scale factors for b-, c- and light-jets to the simulated events is estimated by shifting the value of the applied scale factors by the uncertainty corresponding to one standard deviation ($\pm 1\sigma$) [184]. This yields alternative 1σ-up and 1σ-down event weights.
B-mistagging	For the b-mistagging reweighting method on the multijet data sample the uncertainty is estimated to 5%, following the choice of the original analysis [12].

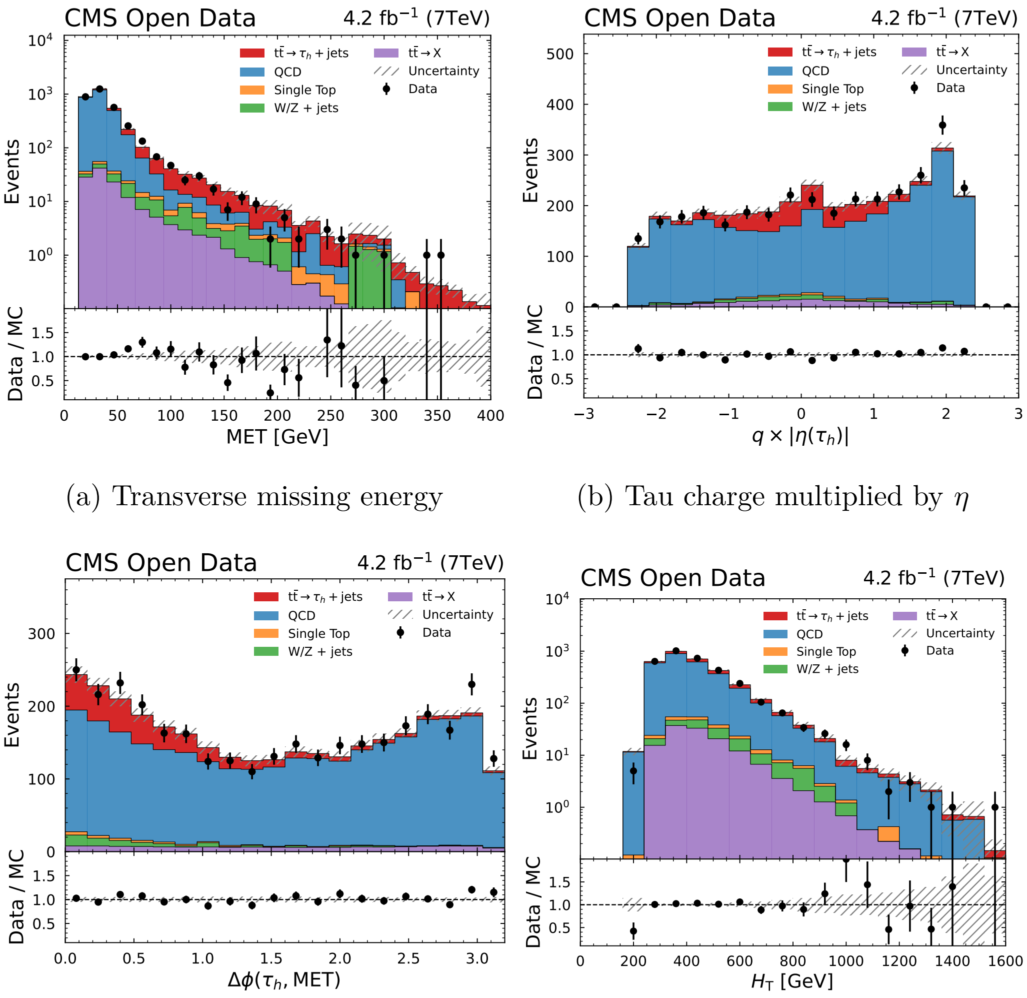
top-quark mass	The uncertainty due to the top-quark mass is taken from the original analysis [12], since the required samples are not available in the CERN Open Data portal. The uncertainty of the top-quark mass is evaluated considering two simulated samples where the nominal top mass of 172.5 GeV has been shifted by ±6 GeV. The estimated uncertainty for the $\mathrm{t\bar{t}}$ signal and background processes is 3%.
Renormalization and factorization scale	The uncertainty due to the renormalization and factorization scale is taken from the original analysis [12], since the required samples are not available in the CERN Open Data portal. The dependency of the selection on the renormalization and factorization scale Q, $Q^2 = m_{top}^2 + \sum p_T^2$, has been estimated using dedicated samples for the $\mathrm{t\bar{t}}$ processes, where the scales have been varied by a factor of 0.5 and of 2.0. The relative uncertainty for the $\mathrm{t\bar{t}}$ signal and background processes processes is estimated to be 2%.
Parton shower matching	The uncertainty due to the parton shower matching is taken from the original analysis [12], since the required samples are not available in the CERN Open Data portal. The influence of the matching thresholds used to associate the matrix elements to the parton showers has been varied from 20 GeV to respectively 10 GeV and 40 GeV. The relative uncertainty for the $\mathrm{t\bar{t}}$ signal and background processes processes is estimated to be 3%.

A main difference to the original analysis is the pile-up correction, that according to the CMS Open Data description does not require any reweighting [228], since the distribution used to generate the Monte Carlo events was matched directly to that observed in the data for the 2011 running. Another important difference is the treatment of the PDF uncertainties, that sums the alternative PDF variations in quadrature instead of taking a single alternative PDF that leads to the maximal 1σ-up and 1σ-down variation, as has been done in the original analysis. Moreover, following the recommendations for the 2011 legacy simulations, no separate uncertainty for the missing transverse energy is considered.

6.8 Neural Network with BCE

A neural network classifier is trained to discriminate $\mathrm{t\bar{t}}$ signal events from QCD background events, yielding the final discriminant variable used as input to the statistical analysis. Feature engineering based on kinematic variables of the selected jets, tau-lepton and missing transverse energy is applied in order to construct high-level features that allow to discriminate between signal and background. The following variables have been calculated and form the input for the multivariate classifier:

Variable	Description
H_T	scalar sum of the transverse momenta of all the selected jets and hadronic tau-lepton candidate
aplanarity	$A = \frac{3}{2}\lambda_1$ with λ_1 being the smallest eigenvalue of the momentum tensor $M^{\alpha\beta} = \sum_i p_i^\alpha p_i^\beta / \sum_i \lvert\vec{p}_i\rvert^2$, where i runs over the number of jets and the τ_h candidate and $\alpha, \beta = 1, 2, 3$ specify the three spatial components of the momentum.
sphericity	$A = \frac{3}{2}(\lambda_1 + \lambda_2)$ with λ_1, λ_2 being the smallest eigenvalue of the momentum tensor $M^{\alpha\beta} = \sum_i p_i^\alpha p_i^\beta / \sum_i \lvert\vec{p}_i\rvert^2$, where i runs over the number of jets and the τ_h candidate.
$q \times \lvert\eta(\tau_\mathrm{h})\rvert$	charge of the tau-lepton candidate multiplied by the absolute value of the pseudorapidity
MET	transverse missing energy
$\Delta\phi(\tau_\mathrm{h}, \mathrm{MET})$	azimuthal angle between the hadronic tau-lepton candidate and the transverse missing energy direction
$M(\tau_\mathrm{h}, \mathrm{jets})$	invariant mass of the selected jets and the hadronic tau-lepton candidate
$M_T(\tau_\mathrm{h}, \mathrm{MET})$	transverse mass of the hadronic tau-lepton candidate and transverse missing energy



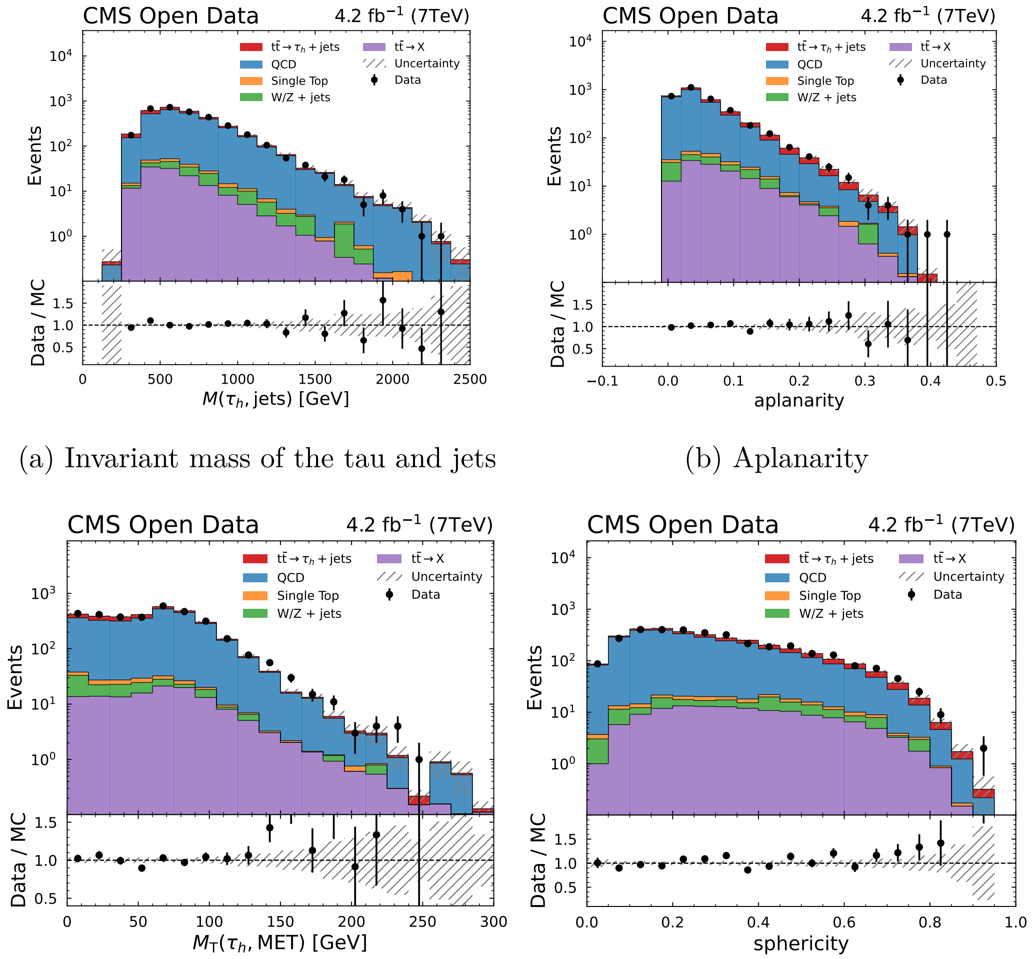
(a) Transverse missing energy

(b) Tau charge multiplied by η

(c) Angle between tau and MET

(d) Scalar p_T sum of the jets and tau

Figure 6.9: Data-simulation agreement for the high-level features that form the input for the machine learning.



(a) Invariant mass of the tau and jets

(b) Aplanarity

(c) Transverse mass of the tau and MET

(d) Sphericity

Figure 6.10: Data-simulation agreement for the high-level features that form the input for the machine learning.

The aplanarity and sphericity account for the spherical topology of the top-quark decay products and the $q \times |\eta(\tau_{\mathrm{h}})|$ variable exploits the charge-symmetry of $\mathrm{t\bar{t}}$ events in contrast to W + jets events. In Fig. 6.9 and Fig. 6.10 the data-simulation agreement for the eight high-level features is shown. As in the previous section, the normalization of the data-driven QCD sample is set to the best fit value after performing the log-likelihood fit discussed in Section 6.9. In general a good agreement between data and simulation is observed.

The total number of $\mathrm{t\bar{t}}$ signal events amounts to 43570 and the number of QCD background events amounts to 11176. The datasamples are split into a training set of 20000 $\mathrm{t\bar{t}}$ signal events and 5000 QCD background events, the remaining events are used for validation. The event weights of the signal and the background samples have been rescaled such that the mean value of the signal event weights equals one and the class weight of the background relative to the signal is set to four accounting for the imbalance of the datasets. For the training of the neural network classifier the PyTorch package [202] is used. A feed-forward neural network architecture with two hidden layers, ReLU activations and a sigmoid function in the last layer has been chosen. The standard binary cross-entropy (BCE) loss function $\mathcal{L}_{\mathrm{xe}}$ as defined in equation 2.10 is used with a batch size of 256. The input to the training are the eight high-level features described above. The features have been rescaled to have a mean of zero and a standard deviation of one. A hyperparmater scan for the learning rate in the range $[10e^{-4}, 10e^{-1}]$ and the number of neurons per layer in the range $[20, 100]$ has been performed, with the performance measured by the lowest BCE loss on the validation set. It was found that a wide range of hyperpa-

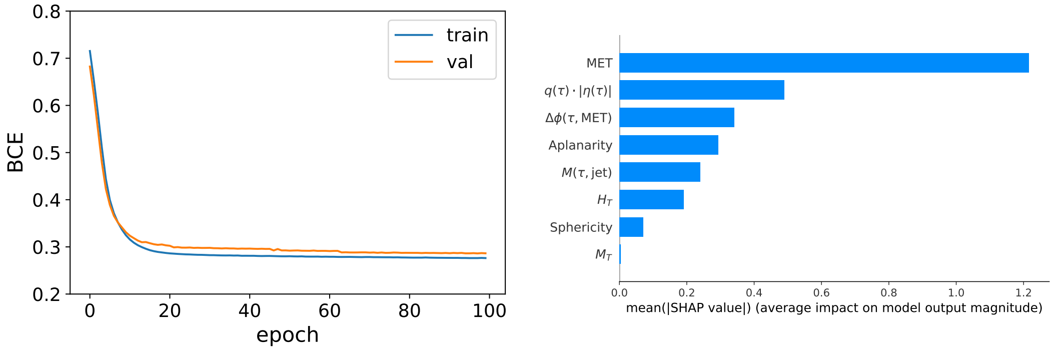


Figure 6.11: Left panel: training and validation BCE loss as a function of the number of epochs. Right panel: estimated variable importance using the SHAP package [229].

rameters give very similar results. This indicates that the performance is limited by the amount of training data. The chosen hyperparameters for the final BCE model are 20 neurons per layer and a learning rate of 0.001. The model is trained for 100 epochs with the ADAM optimizer and the weights that give the lowest BCE loss on the validation set are stored. In the left panel of Fig. 6.11 the training and validation loss of the model are shown, indicating a similar performance on both datasets. The SHAP package [229] which is a game theoretic approach to explain the output of any machine learning model, has been used with a simple tree-based model to provide some intuition about the importance of the input features. A summary plot is depicted in the right panel of Fig. 6.11 that shows the mean average SHAP value which indicates the average impact on the model output magnitude. According to this analysis, the most important variable is the missing transverse energy, which is consistent with the visual impression of Fig. 6.9 and Fig. 6.10. The binned classifier score of the neural network model is shown in the left panel of Fig. 6.12 for the validation set and the data-simulation agreement is shown in the right panel. A good agreement between data and simulation is observed.

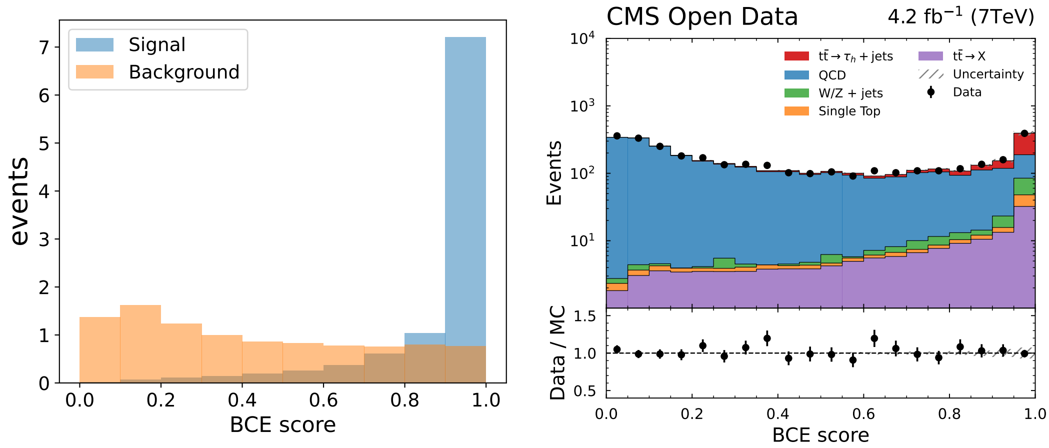


Figure 6.12: Left panel: classifier score of a neural network trained with binary cross-entropy. Right panel: data-simulation agreement for the neural network classifier score.

The original analysis used a Multi-Layer-Perceptron (MLP) of the TMVA package [230], which was state-of-the-art when this analysis was carried out in 2011. The same input variables are used, except of a variable based on the χ^2 returned by a kinematic fit constraining the hadronic W boson and top-quark masses by solving an event-by-event least square problem together with the application of Langrange

Multipliers. This variable has not been considered in the analysis with the CMS Open Data analysis, since it is difficult to reproduce and in the original analysis it proved to be of little importance [12].

6.9 Cross-Section Measurement

The original analysis estimates the QCD multijet background and the $\mathrm{t\bar{t}}$ signal fraction with a two-component binned maximum likelihood fit based on the summary statistic obtained from the neural network classifier output. The normalization of the minor W/Z + jets, $\mathrm{t\bar{t}}$ and single-top backgrounds is fixed to the theory prediction and subtracted from the data prior to the fit. The systematic uncertainties are calculated by repeating the fit with templates varied by $\pm 1\sigma$ of the respective systematic source and summing the differences with respect to the nominal template in quadrature. This procedure is also referred to as "cut variation" and has been a standard approach in early LHC analyses. However, this procedure is considered problematic from a statistical point of view, since it ignores correlations between the systematic uncertainties and the parameter of interest. In addition, important statistical quantities, such as confidence intervals, cannot be defined in a meaningful way. Thus the profile likelihood method is used, which incorporates all relevant parameters in the likelihood and allows to estimate correct confidence intervals, as discussed in detail in Chapter 2.3. This will also be of importance in order to compare the inference with the BCE model with the summary statistic that is obtained with the INFERNO algorithm, since the main principle of INFERNO is to take all relevant uncertainties into account during the training of a neural network classifier. The inference is thus based on the profile likelihood ratio [84], which for a hypothesized value of the signal strength μ is given by:

$$t_{\mu} = -2 \ln \frac{L(\mu, \hat{\hat{\boldsymbol{\theta}}}(\mu))}{L(\hat{\mu}, \hat{\boldsymbol{\theta}})} \tag{6.7}$$

where $\hat{\hat{\boldsymbol{\theta}}}(\mu)$ refers to the the conditional ML estimators of $\boldsymbol{\theta}$ given a strength parameter μ, which means that it maximizes L for a given value of μ. The denominator is the maximized likelihood function, i.e. $\hat{\mu}$, and $\hat{\boldsymbol{\theta}}$ are their ML estimators.

The model trained with a BCE loss as described in the previous section is used to construct a summary statistic by histogramming the predictions of the classifier output. As discussed in the next chapter, it has been found that a value of

20 bins is optimal for the inference. The events of the training set are discarded when building the templates for the fit. The fit is performed with CABINETRY [204] based on PYHF [33]. It has been verified that similar results are obtained with the CMS COMBINE package. The uncertainties and correlation coefficients are calculated with the MINUIT package [231, 232]. The standard error estimate is done with HESSE that calculates the full second-derivative matrix by finite differences and inverts it, which yields the parameter errors and parameter correlations. However, since HESSE assumes a parabolic shape at the minimum of the profile likelihood, the MINOS algorithm [233] is used to calculate the correct positive and negative errors of μ. MINOS uses the profile likelihood method to compute asymmetric confidence intervals by scanning the negative log-likelihood around the minimum. As listed in Table 6.5, the

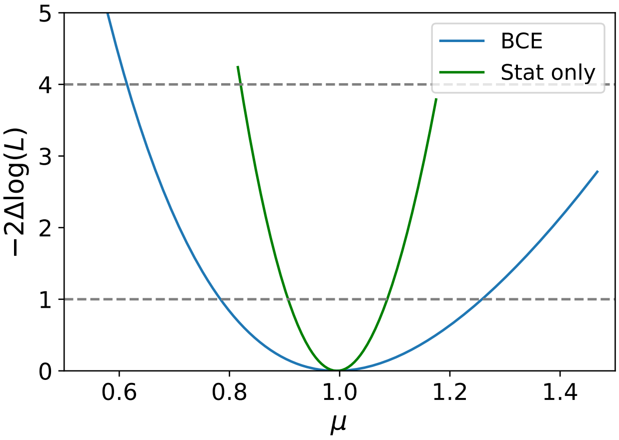


Figure 6.13: Profile likelihood scan for the signal strength μ with the summary statistic obtained by training a neural network classifier with binary cross-entropy.

systematic uncertainties that are affecting the shape of the templates are the JES and JER variations, the b-tagging variation, the statistical trigger variation, the τ energy scale variation and the PDF variation. The shape variations of the templates are taken into account using morphing techniques. Following the original analysis, a rate parameter that multiplies the normalization has been added for the QCD template, which allows this process to float freely in the fit. Therefore the normalization uncertainty corresponding to the mis-tagging of the QCD background has been pruned. The observed signal strength μ with systematic uncertainties evaluates to:

$$\mu = 0.99^{+0.25}_{-0.19}\ (\text{syst.}) \pm 0.09\ (\text{stat.}) \tag{6.8}$$

where the statistical only uncertainty is evaluated by fixing the systematic uncertainties at their best fit value and repeating the fit. An explicit scan of the profile likelihood, both for the fit with and without systematic uncertainties, is shown in Fig. 6.13. The presence of the nuisance parameters broadens the profile likelihood as a function of μ, which reflects the loss of information about μ due to the systematic uncertainties.

To further analyze the systematic uncertainties, the effects that the systematic uncertainties have on the signal strength are evaluated, which is referred to as calculating the "impact" of each uncertainty. The impact is calculated by determining the shift in the signal strength, with respect to the best-fit value, that is induced if a given nuisance parameter is shifted by its ±1σ post-fit uncertainty values. A large shift in the signal strength indicates, that it has a strong dependency on this systematic uncertainty. This is strongly related to the correlation coefficient between the signal strength and the nuisance parameter. The pulls of the nuisance parameters quantify the changes in the nuisance parameter values and uncertainties, relative to their initial pre-fit values. The central value of the pull is defined as:

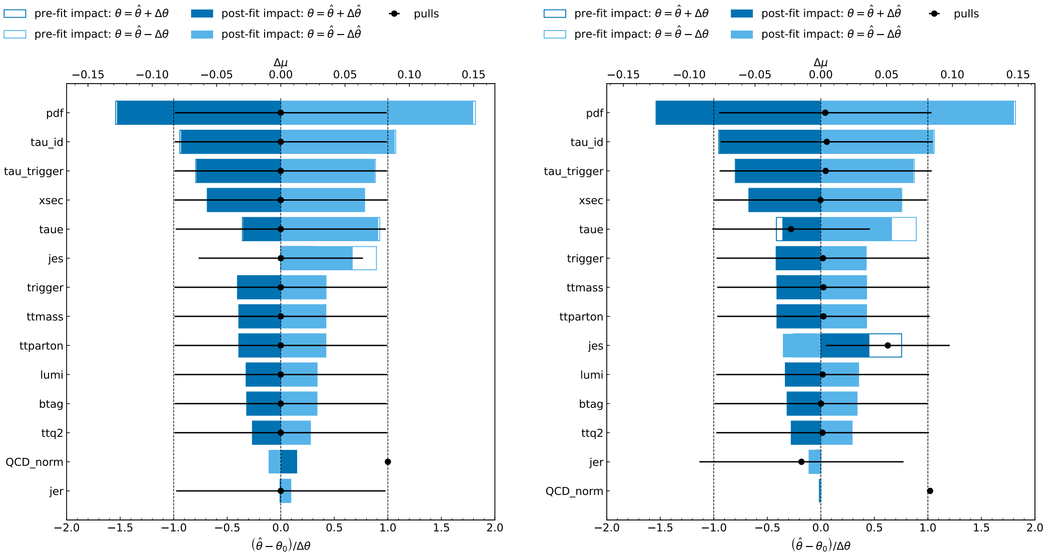


Figure 6.14: Left panel: impacts and pulls for a profile likelihood fit to Asimov data. Right panel: impacts and pulls for a profile likelihood fit to the measured data.

$$\frac{(\theta - \theta_I)}{\sigma_I} \tag{6.9}$$

where θ is the post-fit value, θ_I the pre-fit value and σ_I the pre-fit uncertainty. The pull uncertainty is calculated as the ratio of the post-fit to the pre-fit uncertainty:

$$\frac{\sigma}{\sigma_I} \ . \tag{6.10}$$

In case that the nuisance parameter has no pre-fit uncertainty, such as the rate parameter for the QCD background, a value of 1 is reported. The impact evaluation and the pulls are shown in Fig. 6.14 for the profile likelihood fit to the optimal Asimov data in the left panel and to the measured data in the right panel. In the Asimov dataset all observed quantities are set equal to their expected values. The impacts will be compared to the results for a fit with an INFERNO summary statistic in the next section. It is evident that the largest impact stems from the PDF variation. The correlation matrices obtained in the profile likelihood fit are included in Appendix A.1.

In Fig. 6.15 the result for the signal strength μ of the original analysis and the result of the reproduced CMS Open Data analysis are shown and are in fair agreement. The order of magnitude of the statistical uncertainty and the order of magnitude of the total systematic uncertainty are similar for the original analysis and the CMS Open Data analysis. However, due to the different inference procedures and different procedures to calculate the systematic variations, the relative importance

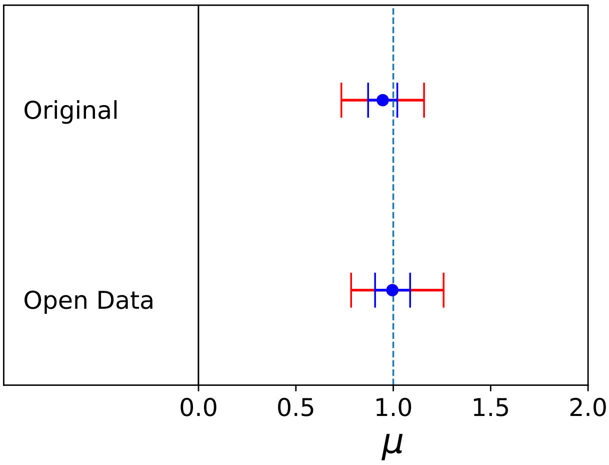


Figure 6.15: Comparison of the signal strength μ between the original analysis and the reproduced analysis with CMS Open Data. The blue bars indicate the statistical uncertainty.

of the single systematic contributions in the original analysis (Table 6.5) and the CMS Open Data analysis (Fig 6.14) are different. In conclusion, by using the same methodology, the CMS Open Data analysis is a realistic reproduction of the original CMS analysis that measures the $\mathrm{t\bar{t}}$ production cross-section in the τ+jets channel at $\sqrt{s} = 7$ TeV and can be used to study the INFERNO algorithm.

7 Application of INFERNO to a $\mathrm{t\bar{t}}$ cross-section measurement

In this chapter the adaption of the INFERNO algorithm to a typical HEP problem will be described and its performance will be evaluated based on the measurement of the $\mathrm{t\bar{t}}$ production cross-section in the τ+jets channel in proton-proton collisions at $\sqrt{s} = 7$ TeV reproduced with CMS Open Data. In order to quantify the behaviour of the algorithm, a study has been performed that tests the performance of the algorithm in a controlled setup by introducing artificial nuisance parameters. In the final section, the performance with the full systematic uncertainties of the CMS Open Data analysis will be evaluated and compared to the results obtained in Chapter 6.

7.1 Extension of INFERNO to HEP Data

The original code for the INFERNO algorithm described in Ref. [11] has been developed for the structure of a synthetic problem. However, the systematic uncertainties typically encountered in HEP have a special structure and thus the algorithm needs to be adapted to HEP-like systematics. There are two types of systematic uncertainties that one commonly encounters in HEP analysis. The first type are simple multiplicative uncertainties that affect the normalization of a process; the second type of systematics affect both the shape of the spectra and the normalization of a process, which is referred to as "*ShapeNorm*" systematics. They are often given as alternative $\pm 1\sigma$ MC samples and morphing algorithms are used to interpolate between the alternative shapes. This has been first implemented in the PyTorch version of INFERNO [95], which also reproduces the results of the synthetic problem. In the following, the two types of systematic uncertainties and an alternative version of an interpolation algorithm that is used in the standard fitting tool of CMS, called COMBINE [234], will be discussed in more detail based on Ref. [235], and a description of the INFERNO algorithm that runs with an arbitrary number of HEP-like systematics for one signal and one background process will be given. Moreover, an alternative approach for a differentiable summary statistic will be discussed.

7.1.1 Normalization Uncertainties

The simplest type of systematic uncertainties are multiplicative uncertainties that affect the normalization of a process [235]. They can be represented by nuisance parameters in profile likelihoods. For example, the integrated luminosity $\tilde{L} \pm \sigma_L$ measured in an auxiliary study can be incorporated into a likelihood as follows:

$$\mathcal{L} = \prod_{i=1}^{N} \mathcal{P}(n_i \mid \mu_i) \mathcal{G}\Big(L \mid \tilde{L}, \sigma_L\Big) \tag{7.1}$$

where n_i are the number of events in each bin i with expected events μ_i and $\mathcal{G}$ is a normalized Gaussian of mean $\tilde{L}$ and width σ_L that constrains the value of the nuisance parameter to its measured value L. The expected number of events μ_i can typically be expressed as:

$$\mu_i = \sum_{j=1}^{n_{\text{source}}} L \sigma_j \epsilon_{ji} \tag{7.2}$$

with cross section σ_j for signal and background sources j, and efficiency ϵ_{ji} for source j in bin i, which is often obtained from MC simulations. The negative log likelihood is then given by

$$-\ln \mathcal{L} = \sum_i [-n_i \ln \mu_i + \mu_i] + \frac{(L - \tilde{L})^2}{2\sigma_L^2} \tag{7.3}$$

and the Gaussian term corresponding to the nuisance parameter L can be interpreted as a penalty on the negative log likelihood. In practical applications it is often useful to constrain each nuisance parameter θ with a Unit Gaussian, that has a mean value of zero and a standard deviation of one, and introduce a normalization function that depends on the nuisance parameter and the measured uncertainty σ_θ:

$$\mathcal{L} = \prod_{i=1}^{N} \mathcal{P}(n_i \mid \mu_i) \mathcal{G}(\theta \mid 0, 1) f^{Norm}(\theta, \sigma_\theta) \ . \tag{7.4}$$

Many different functions are possible, depending on the concrete application. A common normalization function is given by:

$$f^{Norm}(\theta, \sigma_\theta) = 1 + \theta \sigma_\theta \tag{7.5}$$

such that a variation of the nuisance parameter by $\pm 1\sigma$ corresponds to a multiplication by $1 \pm \sigma_\theta$. In Bayesian terms the constraint functions can be considered as the prior probability densities of the nuisance parameters. In this way, any multiplicative uncertainty can be included in the likelihood. It is also possible to use normalization functions with different properties to constrain the values of the nuisance parameters. Usually the physical requirement for a multiplicative nuisance parameter is that it remains positive. In this case the parameter can be constrained with a log-normal distribution that does not allow the parameter to become negative. A log-normal normalization function that depends on a nuisance parameter θ, constrained with a Unit Gaussian, can be written as:

$$f^{LogNorm}(\theta, \kappa) = \exp\left(\theta \ln(\kappa)\right) \tag{7.6}$$

where $\kappa = 1 + \sigma_\theta$. Moreover, it is possible that a process can have asymmetric uncertainties, such that a variation of $+1\sigma$ is different from a variation of -1σ. Following the approach in the CMS COMBINE tool [234], an asymmetric log-normal distribution can be defined that interpolates between $\kappa^{up} = 1 + 1\sigma$ and $\kappa^{down} = 1 - 1\sigma$. This is accomplished by defining a heuristic smoothing function, as implemented in the `AsymPow` class of the CMS COMBINE tool:

$$f^s(x) = 0.25x\ \left(4x^2\ \left(12x^2 - 10\right) + 15\right) \tag{7.7}$$

which can be used to define a function that interpolates between κ^{up} and κ^{down}:

$$f^I(\theta, \kappa^{up}, \kappa^{down}) = \begin{cases} 0.5(\kappa^{up} - \kappa^{down}) + (0.5(\kappa^{up} + \kappa^{down})f^s(\theta)) & \text{if } |\theta| < 0.5 \\ \kappa^{up} \text{ if } \theta > 0 \text{ otherwise } \ -\kappa^{down} & \text{otherwise} \end{cases} \tag{7.8}$$

and the asymmetric log-normal function can then be defined as:

$$f^{AsymmLN}(\theta, \kappa^{low}, \kappa^{hi}) = \exp\left(\theta f^I(\theta, \ln \kappa^{up}, \ln \kappa^{down})\right)\ . \tag{7.9}$$

Within the context of this thesis, the different normalization functions have been implemented in PyTorch [1].

7.1.2 Shape Uncertainties and Interpolation

Many systematic uncertainties also cause an overall distortion in the shape of the observed event features [235]. An example is the jet energy scale uncertainty (JES) that shifts all jet energies in an event in the same direction. If a selection on the energy has been applied during analysis, also the overall normalization of the process changes. The spectral distortions can be modeled by changing parameters in the MC simulation and recalculating the modified distributions. For example, raising and lowering the jet energy scale by one standard deviation, and recalculating the event features, yields three measures of the shape and normalization of the distributions, which are denoted as s^{nom}, s^{up}, and s^{down}. Another common situation are systematic uncertainties that are given as event weights corresponding to a $+1\sigma$ and -1σ variation. In this case the nominal distribution can be weighted event-by-event to obtain three measures of the shape and normalization of the distributions. An example is a systematic uncertainty stemming from the b-tagging scale factors. The three measures of the spectral shape can be converted into a continuous estimate in each bin by introducing a "morphing" parameter θ, which is constrained by a Unit Gaussian distribution. In the technique implemented in the CMS COMBINE tool this is usually referred to as "vertical morphing" and the spectral distortions are modelled by interpolating quadratically for $|\theta < 1|$ and extrapolating linearly beyond that range. The idea is to treat the difference in the shifted values in the bin as if they represent a measurement of the first-order Taylor expansion around the nominal value [235]. Technically this can be achieved by introducing a heuristic smoothing function, as implemented in the `VerticalInterpHistPdf` class of the CMS COMBINE tool:

$$g^s(x) = \begin{cases} 0.125x\ (x^2\ (3x^2 - 10) + 15) & \text{if } |x| < 1 \\ 1 \text{ if } x > 0 \text{ otherwise } -1 & \text{otherwise} \end{cases} \tag{7.10}$$

and defining an interpolation function based on the method of Lagrange interpolation for the i-th bin of the summary statistic:

$$g_i^I(\theta, s^{nom}, s^{up}, s^{down}) = 0.5\theta\Big[\Big(s_i^{up} - s_i^{down}\Big) + \Big((s_i^{up} + s_i^{down} - 2s_i^{nom})g^s(\theta)\Big)\Big] \tag{7.11}$$

which has the property that $g_i^I(\theta = 1) = s_i^{up} - s_i^{nom}$ and $g_i^I(\theta = -1) = s_i^{nom} - s_i^{down}$ and $g_i^I(\theta = 0) = 0$. This morphing method can be extended to several morphing parameters for different systematic effects by adding linearly the deviations from the nominal summary statistic due to each effect. For a set of of $1 + 2k$ summary statistics $\boldsymbol{s} = \{s^{nom}, \boldsymbol{s}^{up}, \boldsymbol{s}^{down}\}$, where the nominal summary statistic is denoted as s^{nom}, the k 1σ-up variations as $\boldsymbol{s}^{up}$ and the k 1σ-down variations as $\boldsymbol{s}^{down}$, the interpolated summary statistic S with k morphing parameters $\boldsymbol{\theta}$, can be written for the i-th bin as:

$$S_i(\boldsymbol{s}, \boldsymbol{\theta}) = s_i^{nom} + \sum_{j=0}^{k} g_i^I(\theta_j, s^{nom}, s_j^{up}, s_j^{down}) \ . \tag{7.12}$$

More accurate representations of the morphing can be obtained by computing additional shifted spectra and interpolating with a higher order polynomial, which however is computationally expensive. Within the context of this thesis, the interpolation algorithm has been implemented in PyTorch [1]

7.1.3 Differentiable Summary Statistics

The INFERNO algorithm requires a differentiable summary statistic, such that the gradients can be calculated with automatic differentiation in order to optimize the parameters of a neural network. A possible approach is to modify the summary statistic that is used in the original INFERNO paper such that it takes into account event weights, which is required by systematic uncertainties that are given by variations of the event weights, such as b-tagging variations of the scale factor. Thus given a neural network f with parameters $\boldsymbol{\phi}$ and n samples $\boldsymbol{x}$ with weights $\boldsymbol{w}$, a possible summary statistic s^{sm} based on a softmax function can be written for the i-th component as:

$$s_i^{sm}(\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\phi}) = \sum_{j=0}^{n} \frac{e^{f_i(x_j;\phi)/\tau}}{\sum_{k=0}^{b} e^{f_k(x_j;\phi)/\tau}} \cdot w_j \tag{7.13}$$

where b is number of output nodes in the last layer of the neural network and τ is the temperature hyperparameter. The number b of the output nodes of the neural network defines the number of bins of the summary statistic. After the neural network has been trained, the *argmax* operator can be applied to the predictions of the model for unseen data to assign each event to a unique bin.

An alternative approach is the implementation of an approximately differentiable histogram [236], that approximates the non-differentiable bin edges of a histogram with sigmoid functions, where a sigmoid function is given by:

$$\sigma(x) = \frac{1}{exp(-\kappa x)} \tag{7.14}$$

and κ is a hyperparameter that regulates the smoothness. A sketch of this approach is shown in Fig. 7.1. A differentiable bin can be defined as:

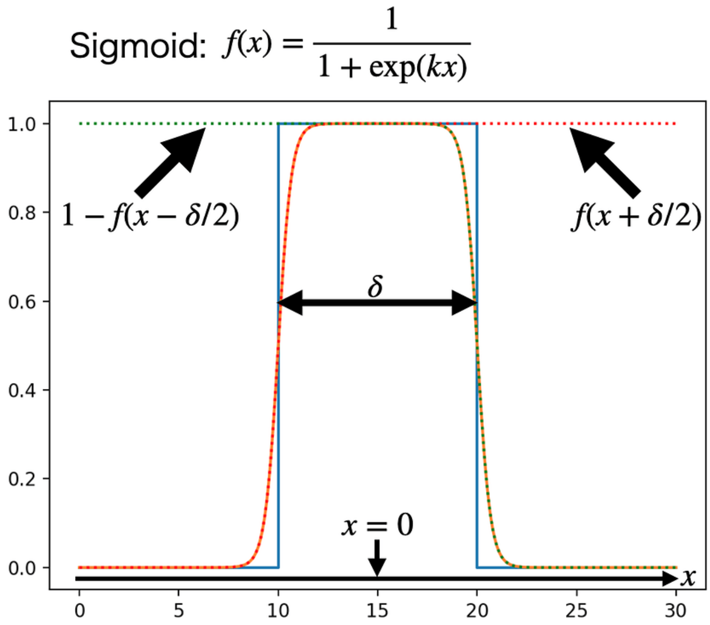


Figure 7.1: Sketch of a differentiable histogram with bin edges approximated by sigmoid functions [236].

$$B(x; c_i, \delta, \kappa) = \sigma(x - c_i + \delta/2) - \sigma(x - c_i - \delta/2) \tag{7.15}$$

where the bin center is denoted by c_i and δ corresponds to the bin width. Given a neural network f with parameters $\boldsymbol{\phi}$ and n samples $\boldsymbol{x}$ with weights $\boldsymbol{w}$, the i-th component of a summary statistics s^{dh} based on an approximately differentiable histogram can be written as:

$$s_i^{dh}(\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{\phi}) = \sum_j B(f(x_j; \boldsymbol{\phi});\ c_i, \delta, \kappa) \cdot w_j\ . \tag{7.16}$$

The summary statistic s^{dh} can be used to bin the predictions of a neural network that has a single sigmoid function in the last layer and thus the predictions are bound between zero and one. An advantage of this approach is that the output is a continuous variable and, as will be shown in Section 7.2, its distribution shows similarities to a model trained with a cross-entropy loss.

7.1.4 Algorithm

Based on the discussed treatment of typical HEP systematics and the introduced summary statistics, the INFERNO algorithm has been extended to run with an arbitrary number of HEP-like systematics for one signal and one background process. This approach could be generalized for multiple signal and background processes. In the following the main aspects of the algorithm, in particular the construction of the Asimov likelihood, will be described.

Input

The input to the algorithm is a nominal dataset x^{nom} with weights w^{nom} consisting of n^{sig} signal samples and n^{bkg} background samples. As discussed in the previous section, there are two types of systematic uncertainties that affect the shape of the classifier. The first class results in m alternative $+1\sigma$ (up) and -1σ (down) datasets $\boldsymbol{x}^{up}$ and $\boldsymbol{x}^{down}$ with corresponding weights $\boldsymbol{w}^{up}$ and $\boldsymbol{w}^{down}$ that originate from a repetition of the analysis with altered parameters in the MC simulation. An example for these types of systematic uncertainties are jet energy scale uncertainties. If a selection criterion is applied, also the normalization of the systematic variations are different with respect to the nominal dataset. The second type of systematic uncertainties are event weights, such as variations due to the b-tagging scale factors, that provide k alternative $+1\sigma$ and -1σ sets of weights $\boldsymbol{w}^{up}$ and $\boldsymbol{w}^{down}$ for the nominal data x^{nom}. Thus in total the input to the algorithm consists of $1+2m$ datasets with $1+2m+2k$ sets of weights. This results in $m+k$ *ShapeNorm* nuisance parameters $\boldsymbol{\theta}^{SN}$ corresponding to the $m+k$ systematic uncertainties. To facilitate the training procedure, it is required that each nominal sample has a corresponding 1σ-up and 1σ-down variation for each considered systematic uncertainty. This requires a suitable pre-processing of the data, in order to exclude samples from the training set that have only one up or down variation.

The algorithm is implemented in PyTorch [202] based on the PyTorch_Inferno implementation [95]. The dimension of the batches $\boldsymbol{X}$ with weights $\boldsymbol{w}$ in a batch with batchsize b is given by:

$$dim(\boldsymbol{X}) = [b,\ dim(\text{features}),\ 1+2m] \tag{7.17}$$

and the dimension of the weights is:

$$dim(\boldsymbol{w}) = [b,\ 1 + 2m + 2k]\ . \tag{7.18}$$

An ordering scheme of the datasets and weights has been defined in order to identify the $1 + 2m$ datasets in a batch $\boldsymbol{X}$ with the corresponding weights $\boldsymbol{w}$. Furthermore, for each *ShapeNorm* nuisance parameter $\boldsymbol{\theta}^{SN}$, an asymmetric uncertainty $\boldsymbol{\sigma}^{SN}$ in percent is provided that corresponds to the $+1\sigma$ and -1σ normalization uncertainty. Besides *ShapeNorm* uncertainties also multiplicative normalization uncertainties, such as the luminosity, can be considered in the training process which requires specifying the multiplicative uncertainties $\boldsymbol{\sigma}^{N}$ in percent. The total set of nuisance parameters $\boldsymbol{\theta}$ consists of the $m + k$ nuisance parameters that correspond to the *ShapeNorm* variations $\boldsymbol{\theta}^{SN}$, the p nuisance parameters corresponding to the normalization uncertainties $\boldsymbol{\theta}^{N}$ and a possible rate parameter θ^{rate} that multiplies the normalization of the background:

$$\boldsymbol{\theta} = \left\{\boldsymbol{\theta}^{SN}, \boldsymbol{\theta}^{N}, \theta^{rate}\right\}\ . \tag{7.19}$$

Training Loop

Given a neural network f with parameters $\boldsymbol{\phi}$, for each batch of data a set of summary statistics is calculated according to the prescription in equation 7.13 or equation 7.16. The nominal summary statistic s^{nom} is calculated with the nominal dataset x^{nom} and the nominal set of weights w^{nom}. The m 1σ-up and 1σ-down summary statistics that correspond to the systematic uncertainties from the repetition of the analysis are calculated from the m 1σ-up and 1σ-down variations of the datasets $\boldsymbol{x}^{up}$ and $\boldsymbol{x}^{down}$ with weights $\boldsymbol{w}^{up}$ and $\boldsymbol{w}^{down}$. The k 1σ-up and 1σ-down summary statistics that correspond to the event weight systematics are calculated by reweighting the nominal dataset x^{nom} with the k weights $\boldsymbol{w}^{up}$ and k weights $\boldsymbol{w}^{down}$. Thus in each batch a set of $1{+}2m{+}2k$ summary statistics $\boldsymbol{s} = \{s^{nom}, \boldsymbol{s}^{up}, \boldsymbol{s}^{down}\}$ are obtained both for the signal and the background samples. All summary statistics are normalized to the total integral. The set of summary statistics that correspond to the signal samples are denoted as $\boldsymbol{s}^{S}$ and the set of summary statistics for the background samples are denoted as $\boldsymbol{s}^B$. The i-th component of the full model $\mathcal{M}_i$ can then be written as:

$$\begin{aligned}\mathcal{M}_i(s,\boldsymbol{\theta},\boldsymbol{\phi}) = s \cdot S_i(\boldsymbol{s}^S,\boldsymbol{\theta}^{SN}) \prod_j^{m+k} f^N(\theta_j^{SN})\mathcal{G}_u(\theta_j^{SN}) \prod_l^p f^N(\theta_l^N)\mathcal{G}_u(\theta_l^N) \; + \\ b \cdot S_i(\boldsymbol{s}^B,\boldsymbol{\theta}^{SN}) \prod_j^{m+k} f^N(\theta_j^{SN})\mathcal{G}_u(\theta_j^{SN}) \prod_l^p f^R(\theta_l^N)\mathcal{G}_u(\theta_l^N) \cdot \theta^R\end{aligned} \tag{7.20}$$

where s is the expected number of signal events, b is the expected number of background events and S_i is the i-th component of the interpolated summary statistic defined in equation 7.12. The function f^N is one of the normalization functions defined in Section 7.1.1 that take into account the provided uncertainties for the *ShapeNorm* nuisance parameters $\boldsymbol{\sigma}^{SN}$ and the normalization nuisance parameters $\boldsymbol{\sigma}^N$. The function $\mathcal{G}_u$ is a Unit Gaussian that constrains each nuisance parameter. For the *ShapeNorm* nuisance parameters $\boldsymbol{\theta}^{SN}$ the interpolation and normalization of a single systematic uncertainty are assumed to be fully correlated, thus only one nuisance parameter is used for the interpolation of the summary statistics and the normalization function. The full Asimov Poisson Likelihood can then be written as:

$$\hat{\mathcal{L}}_A(s,\boldsymbol{\theta};\boldsymbol{\phi}) \; = \prod_i^b \mathcal{P}(\mathcal{M}_i(s,\boldsymbol{\theta},\boldsymbol{\phi}) \mid \mathcal{M}_i(s,\boldsymbol{\theta},\boldsymbol{\phi})) \; . \tag{7.21}$$

From the Asimov likelihood the Fisher information matrix is calculated via automatic differentiation according to:

$$\boldsymbol{I}(s,\boldsymbol{\theta})_{ij} = \frac{\partial^2}{\partial\eta_i\partial\eta_j}\Big(-\log\hat{\mathcal{L}}_A(s,\boldsymbol{\theta};\boldsymbol{\phi})\Big) \tag{7.22}$$

where $\boldsymbol{\eta} = \{s,\boldsymbol{\theta}\}$ is the complete set of parameters used in the model training. As described in Chapter 3.3.1, the covariance matrix can then be estimated from the inverse of the Fisher information matrix if $\hat{s}$ and $\hat{\boldsymbol{\theta}}$ are unbiased estimators of the values of s and $\boldsymbol{\theta}$:

$$\mathrm{cov}(\hat{s},\hat{\boldsymbol{\theta}}) \geq I(s,\boldsymbol{\theta})^{-1} \tag{7.23}$$

and the diagonal elements $I_{ii}^{-1}(s,\boldsymbol{\theta})$ correspond to the expected variance for the parameters s and θ_i. The loss value used to optimize the neural network parameters $\boldsymbol{\phi}$ is chosen to be the variance of the expected number of signal events s:

$$U = I_{00}^{-1}(s,\boldsymbol{\theta}) \tag{7.24}$$

which corresponds to the expected width of the confidence interval for s accounting also for the effect of the nuisance parameters $\boldsymbol{\theta}$. It should be noted that the loss value can be adapted to the particular problem and is not restricted to using the approximate variance of the POI. For example, it is also possible to optimize the likelihood ratio between the signal-plus-background model and the background model. The output of the algorithm is the optimized neural network f that can be used to construct an optimal summary statistic.

7.2 Performance Study

In order to study the performance of the INFERNO algorithm with the $t\bar{t} \rightarrow \tau_h + \text{jets}$ analysis reproduced with CMS Open Data in Chapter 6, the performance of the algorithm is evaluated in several simplified setups. As a consistency check, the inference without systematic uncertainties based on a summary statistic obtained with INFERNO is evaluated and compared to a summary statistic obtained by training a model with BCE. In a further study, artificial *ShapeNorm* systematic uncertainties are introduced in order to quantify possible improvements with INFERNO if nuisance parameters are present. Moreover, the effect of normalization uncertainties is studied. The knowledge gained in these studies is then applied in Section 7.3 to train a more complex model and perform the full inference with all relevant systematic uncertainties for the $t\bar{t} \rightarrow \tau_h + \text{jets}$ analysis.

7.2.1 Training and Inference Setup

For all the studies performed, the chosen number of training events consists of 5000 QCD background and 20000 $t\bar{t}$ signal events, while the validation set consists of 5600 background and 23000 signal events. For the INFERNO training, a feed-forward neural network with two hidden layers and ReLU activations is implemented in PyTorch with 10 output nodes in the final layer. The summary statistic based on the softmax function defined in equation 7.13 is used for all INFERNO models unless otherwise stated.

A large batch size of 1000 is used in order to reduce fluctuations when building the batch-wise summary statistics within the INFERNO algorithm. The input to the INFERNO algorithm are the same eight normalized high-level features that have been used for the training of a model with the BCE loss in Chapter 6. The most important model parameters have been optimized with a hyperparameter scan. The learning rate is optimized in the range $[10e^{-4}, 10e^{-1}]$, the number of neurons per layer in the range $[20, 100]$ and the temperature in the range $[0.01, 0.99]$. The models are compared according to the best validation loss. In general, it has been found that a wide range of hyperparameters give similar results. Thus for the INFERNO model a learning rate of 0.001, 60 neurons per layer, and a temperature τ of 0.1 is chosen. The training is performed for 100 epochs with the ADAM optimizer. The initial expected number of events for the signal is set to 348, as predicted by the simulation normalized to the theoretical cross-section and luminosity (Table 6.4). The number of background events is set to 2690, which was shown in Section 6.6 to be a realistic estimation. Both for the signal and background events the event weights are taken into account during training. The INFERNO model is compared to an optimized BCE model with 20 neurons per layer and a learning rate of 0.001 trained for 100 epochs with the ADAM optimizer and a batch size of 256. The choice of different architectures for the INFERNO and BCE model ensures that each model is optimal for the respective loss.

The profile likelihood fit is performed with CABINETRY based on PYHF. It has been verified that similar results are obtained with the CMS COMBINE package. The parameter uncertainties and correlation coefficients are estimated with the HESSE algorithm based on the MINUIT package. The MINOS algorithm is used to calculate the correct positive and negative errors of the signal strength μ. The events of the training set are discarded when building the templates for the fit. For the summary statistic based on the softmax function, the *argmax* operator is applied to the predictions of the neural network to obtain a one dimensional summary statistic. It is sometimes possible that INFERNO predicts zero for all processes in a bin. In this case the bin is excluded to avoid numerical instabilities. As will be discussed in the next section, for INFERNO it has been observed that the performance of the inference has little dependence on the number of bins, thus in the following the number of the output nodes for INFERNO models have been set to 10. The number of bins for the histograms of the BCE model predictions is set to 20.

7.2.2 INFERNO without Nuisance Parameters

To quantify the behaviour of the INFERNO algorithm, a neural network is trained with INFERNO without including any nuisance parameters. Therefore the only relevant parameter in the algorithm is the number of expected signal events s. According to the Neyman-Pearson Lemma it is expected that a classifier trained with BCE should be optimal for inference and give similar results as a classifier trained with INFERNO if no nuisance parameters are present. The class predictions of the INFERNO model for the signal and background components of the validation set is shown in the left panel in Fig. 7.2. It should be noted that, unlike in a training with BCE, the bins are not ordered by default and the order of the bins can change randomly with the initialization of the network. However, it is possible to sort the bins post-training e.g. according to the signal-background ratio, to emulate a BCE-like behaviour. In the right panel of Fig. 7.2 the loss of the INFERNO training, that corresponds to the approximated variance $\sigma^2(s)$ of the expected number of signal events s is shown. The training converges to a summary statistic that provides low variance for the number of expected signal events s. The value of the variance of s is also shown for the BCE model. This value is obtained by histogramming the BCE output predictions after each epoch and using these histograms as input to the INFERNO algorithm to calculate the approximate covariance matrix and thus obtain the approximate variance of s. Technically this is possible because the INFERNO algorithm has been implemented as a callback. Comparing the INFERNO loss curve of the variance $\sigma^2(s)$ with the evolution of $\sigma^2(s)$ during the BCE training shows that both models converge to a similar value.

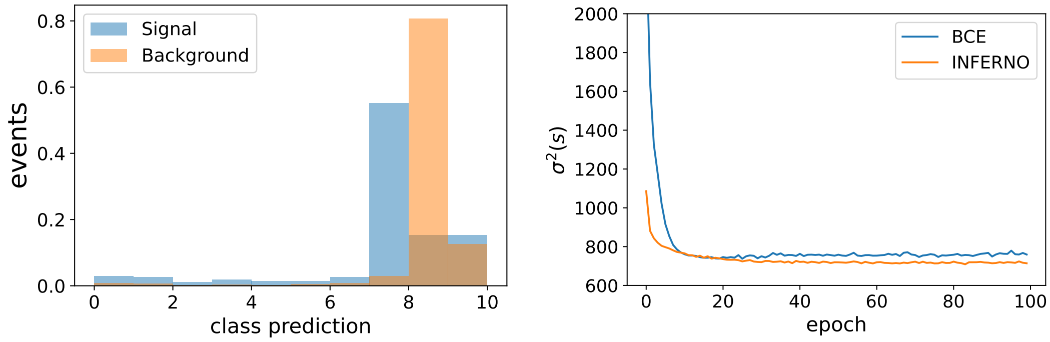


Figure 7.2: Left panel: class prediction of the INFERNO algorithm without nuisance parameters. Right panel: approximated variance $\sigma^2(s)$ for the number of expected signal events s for the INFERNO model and BCE model.

The summary statistic obtained from the predictions of the INFERNO model and the histogrammed predictions of the BCE model are then used to build summary statistics for the profile likelihood fit for all the relevant signal and background processes of the reproduced $\mathrm{t\bar{t}} \rightarrow \tau_{\mathrm{h}} + \mathrm{jets}$ analysis. In order to compare the inference with the model training, no nuisance parameters are included in the profile likelihood fit. The agreement between data and simulation for the INFERNO predictions is

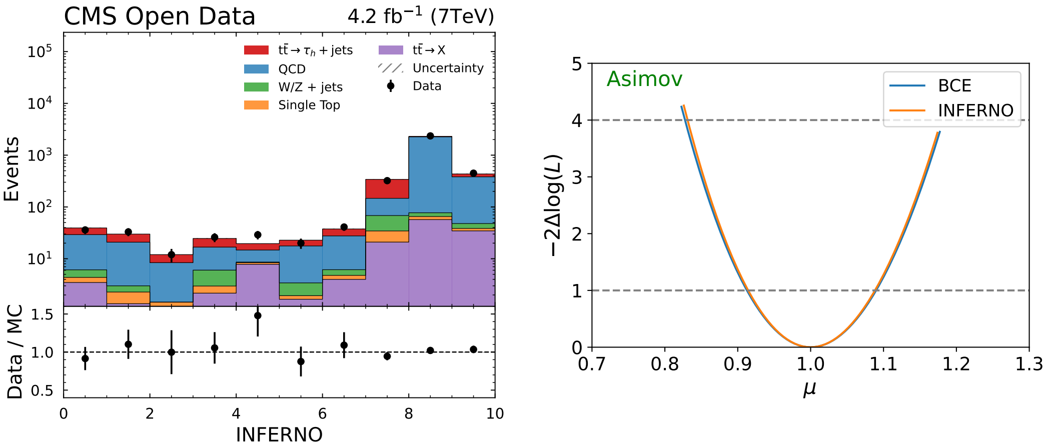


Figure 7.3: Left panel: data-simulation agreement for the INFERNO training without nuisance parameters. Right panel: comparison of profile likelihood scan of the fit with BCE summary statistics and INFERNO summary statistics with Asimov data.

shown in the left panel of Fig. 7.3. A good agreement between data and simulation is observed. The results for the profile likelihood scan with Asimov data, also referred to as a saturated model in statistics, is shown in the right panel of Fig. 7.3. The measured signal strength without systematic uncertainties on the Asimov dataset based on the INFERNO and the BCE summary statistics are:

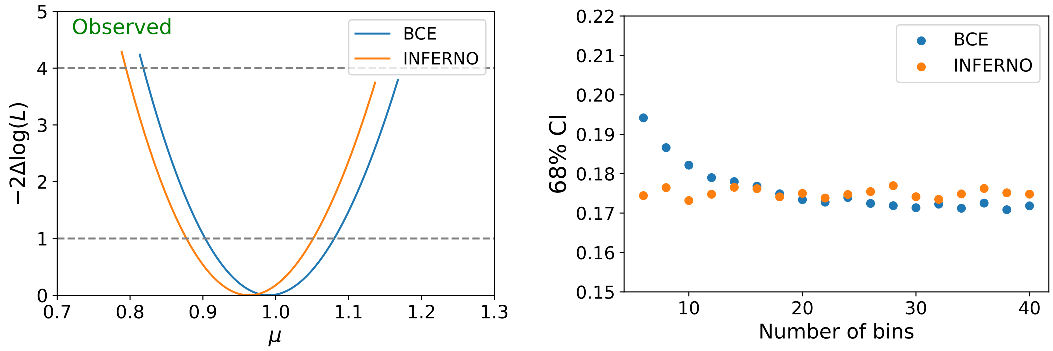


Figure 7.4: Left panel: comparison of the profile likelihood scan of the fit with BCE summary statistics and INFERNO summary statistics with the measured data. Right panel: measured confidence interval on Asimov data as a function of the number of bins.

$$\begin{aligned} \mu^{A}_{\text{BCE}} &= 1.00^{+0.089}_{-0.087} \\ \mu^{A}_{\text{INF}} &= 1.00^{+0.088}_{-0.086} \end{aligned} \tag{7.25}$$

which confirms that the obtained confidence intervals for the INFERNO and BCE models are very similar. It can also be seen that the square root of the converged approximated variance of the expected signal events $\sigma(s) \approx 28$ (Fig 7.2), is a good approximation of the measured uncertainty of μ, which corresponds to ~ 30 signal events. In the fit to the measured data the normalization of the QCD template can float freely, which however has little effect on the uncertainty of μ. The measured signal strength evaluated in a profile likelihood fit to the measured data with INFERNO and BCE summary statistics is:

$$\begin{aligned} \mu_{\mathrm{BCE}} &= 0.99^{+0.089}_{-0.087} \\ \mu_{\mathrm{INF}} &= 0.96^{+0.089}_{-0.085} \ . \end{aligned} \tag{7.26}$$

The scan of the profile likelihoods is shown in the left panel of Fig. 7.4. The minima of the profile likelihoods are slightly different, which can be explained by statistical fluctuations. The magnitude of the obtained confidence intervals for the BCE and INFERNO models is similar. This is expected, since a classifier trained with BCE should be optimal for inference if no nuisance parameters are present and INFERNO should be optimal for inference by construction.

Moreover, a study has been performed to find the optimal number of bins for inference. The results are shown in the right panel of Fig. 7.4, where the magnitude of the 68% confidence interval evaluated on Asimov data has been plotted as a function of the number of bins. For each bin number, a separate INFERNO model has been trained with the number of output nodes in the last layer set to the corresponding bin value. For the BCE model it has been found that a suitable number of bins should be of 20 or more, which motivates the choice of 20 bins for the binning of the BCE predictions in Chapter 6.

7.2.3 INFERNO with ShapeNorm Nuisance Parameters

In order to quantify the performance of the INFERNO algorithm in situations where nuisance parameters are present that affect the shape and normalization of the classifier, a study has been performed with artificial *ShapeNorm* nuisance parameters. An artificial *ShapeNorm* systematic uncertainty is introduced by shifting the mean value of one of the input variables of the analyzed $\mathrm{t\bar{t}} \rightarrow \tau_{\mathrm{h}} + \mathrm{jets}$ data. Out of the eight input variables, the aplanarity variable has been chosen and the mean value of the variable is shifted up and down at 5 points between 0.005 and 0.02 in order to obtain artificial 1σ-up and 1σ-down variations. The normalization is increased by 5% for the 1σ-up variations and reduced by 5% for the 1σ-down variations. This allows to study the confidence interval of the signal strength and the associated covariance matrix obtained in profile likelihood fits as as a function of the shift. The shift can be applied both to the signal and background samples. An example for a shift of 0.0125 is illustrated in Fig. 7.5. The left panel shows the distribution of the artificial $\pm 1\sigma$ variation for the aplanarity variable of the signal process and the right panel shows the distribution of the $\pm 1\sigma$ variations for the aplanarity variable of the background process. This setup, although artificial, allows to study the effect of the *ShapeNorm* uncertainties in a controlled setup and relates to the studies performed with the synthetic example described in Section 3.3.2. As will be shown in Section 7.3, the realistic jet energy scale variation shows a similar behaviour as the systematic uncertainty introduced by this artificial shift.

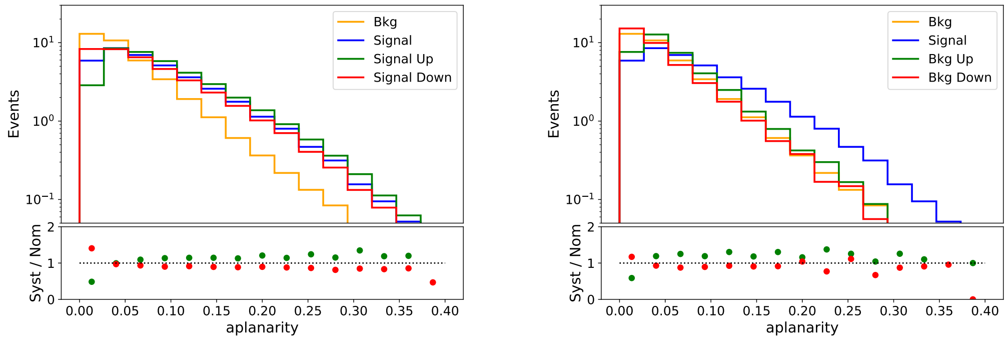


Figure 7.5: Left panel: artificial $\pm 1\sigma$ variation for the aplanarity variable of the signal process. Right panel: artificial $\pm 1\sigma$ variation for the aplanarity variable of the background process.

ShapeNorm Nuisance Parameter for the Signal Process

First, the effect of one artificial *ShapeNorm* nuisance parameter that affects the signal process is studied. The 1σ-up and 1σ-down shape variation and the correlated normalization uncertainty are included in the INFERNO training according to the description in Section 7.1. Thus, the total number of parameters in the algorithm is two and the approximate variance of s is chosen as the loss value. The nuisance parameter θ is constrained with a Unit Gaussian distribution. The INFERNO model is compared to an optimized BCE model trained on the same dataset. The approx-

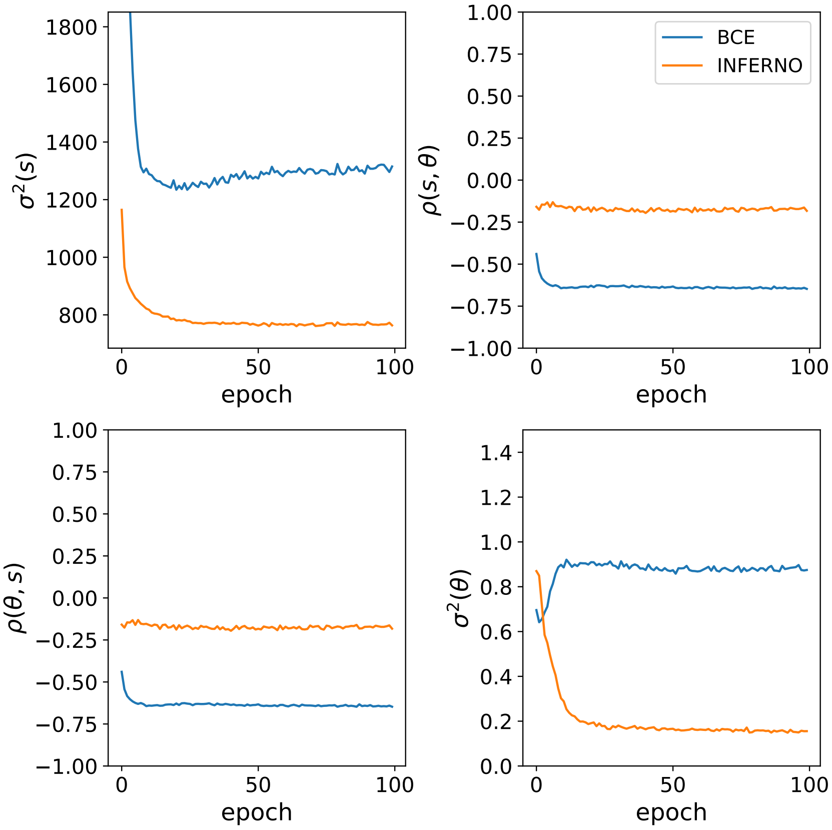


Figure 7.6: Evolution of the covariance matrix evaluated on the validation set during the training. The diagonal elements show the variance of the expected number of signal events s and the variance of the nuisance parameter θ. The off-diagonal elements show the correlation coefficient $\rho(s, \theta)$.

imated covariance matrix is monitored after each epoch on the validation set. As in the previous section, the approximate variance of the expected number of signal events s can also be calculated during the training of a BCE model by binning the model predictions to create summary statistics that can be used in the INFERNO algorithm. The evolution of the 2×2 covariance matrix for a nuisance parameter corresponding to a shift of 0.0125 is shown in Fig. 7.6 as a function of the number of epochs for the INFERNO and BCE model. The diagonal elements of the figure display the variance of the expected number of signal events $\sigma^2(s)$ and the variance of the nuisance parameter θ, denoted by $\sigma^2(\theta)$. The off-diagonal elements show the correlation coefficient ρ between s and θ which is defined as the covariance of the variables divided by the product of their standard deviations:

$$\rho(s,\theta) = \frac{\mathrm{Cov}(s,\theta)}{\sigma(s)\sigma(\theta)} \ . \tag{7.27}$$

The correlation coefficient ρ is bound between -1 and 1. Evaluating the evolution of the covariance matrix shows that the variance of the parameter of interest s converges to a lower value with the INFERNO model compared to classifier trained with BCE. Comparing the evolution of the variance $\sigma^2(s)$ to the one obtained in the previous section without nuisance parameters, illustrates that in the presence of nuisance parameters the BCE classifier is not optimal any more. It is further observed that the correlation coefficient ρ converges to a value closer to zero during the INFERNO model training, and the variance of the nuisance parameter θ converges to a lower value with INFERNO compared to the BCE classifier. This indicates that the INFERNO algorithm makes optimal use of the data in order to decorrelate the parameter of interest s from the nuisance parameter θ, which results in a lower variance for s compared to a model trained with BCE.

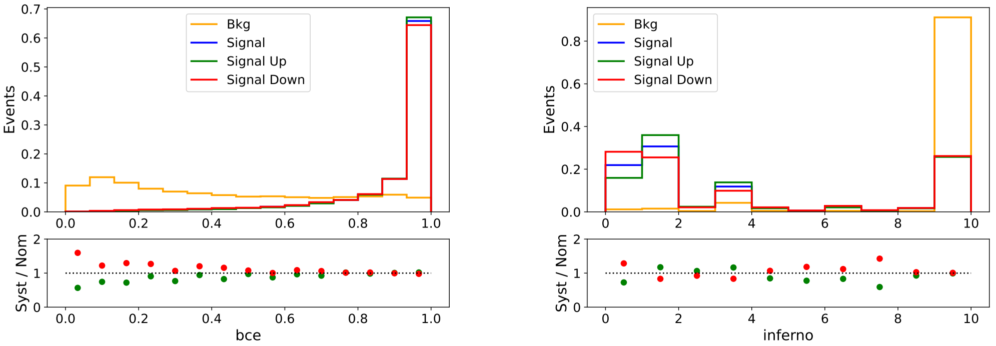


Figure 7.7: Left panel: comparison of the normalized shapes of the BCE predictions for the signal, background and the 1σ-up and 1σ-down variations. Right panel: comparison of the normalized shapes of the INFERNO predictions for the signal, background and the 1σ-up and 1σ-down variations.

A comparison of the normalized shapes of the predictions for the signal, background and the 1σ-up and 1σ-down variations corresponding to the artificial systematic uncertainty is shown in Fig. 7.7 for the BCE model (left panel) and the INFERNO model (right panel). The ratio between the nominal shape and the systematic variations shows that the INFERNO predictions are arranged in a more balanced way compared to the BCE predictions.

The summary statistics for all the relevant signal and background processes of the reproduced $t\bar{t} \to \tau_h +$jets analysis are obtained from the predictions of the INFERNO model and the histogrammed predictions of the BCE model. The summaries are then used as input for the profile likelihood fit. In order to compare the inference with the model training, only the *ShapeNorm* nuisance parameter corresponding to the artificial shift is included in the fit. The measured confidence interval for the signal strength μ evaluated on Asimov data based on the INFERNO and BCE summary statistics is:

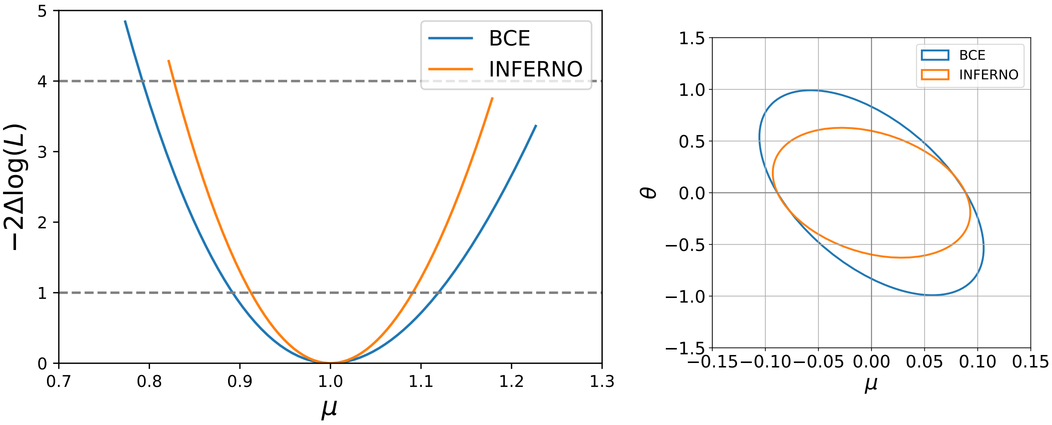


Figure 7.8: Left panel: profile likelihood scan for the signal strength μ and the *ShapeNorm* nuisance parameter θ for the INFERNO and BCE summary statistic. Right panel: error ellipse for the parameters μ and the *ShapeNorm* nuisance parameter θ for the INFERNO and BCE summary statistic.

$$\begin{aligned} \mu^A_{\text{BCE}} &= 1.00^{+0.119}_{-0.108} \\ \mu^A_{\text{INF}} &= 1.00^{+0.091}_{-0.088} \end{aligned} \tag{7.28}$$

and the post-fit uncertainty of the nuisance parameter θ obtained from the HESSE estimate is:

$$\begin{aligned} \theta^A_{\text{BCE}} &= 0.00 \pm 0.975 \\ \theta^A_{\text{INF}} &= 0.00 \pm 0.442 \end{aligned} \tag{7.29}$$

and the correlation coefficient ρ between μ and θ has been evaluated to:

$$\begin{aligned} \rho^A_{\text{BCE}} &= -0.17 \\ \rho^A_{\text{INF}} &= -0.62 \ . \end{aligned} \tag{7.30}$$

A scan of the profile likelihood for the parameter μ is shown in the left panel of Fig. 7.8, both for the INFERNO and BCE model. The result illustrates that the INFERNO summary statistic yields a narrower confidence interval for the signal strength μ compared to the BCE model if a *ShapeNorm* nuisance parameter is present. It further shows that the estimates of the covariance matrix in the INFERNO training (Fig. 7.6) are good estimates of the values obtained from the fit of the Asimov data. The error ellipse for μ and θ is displayed in the right panel of Fig. 7.8. It visualizes the correlation and uncertainties for both parameters. As has been observed during the training, the INFERNO algorithm reduces the correlations between μ and θ which results in reduced uncertainties for both parameters compared to the BCE classifier.

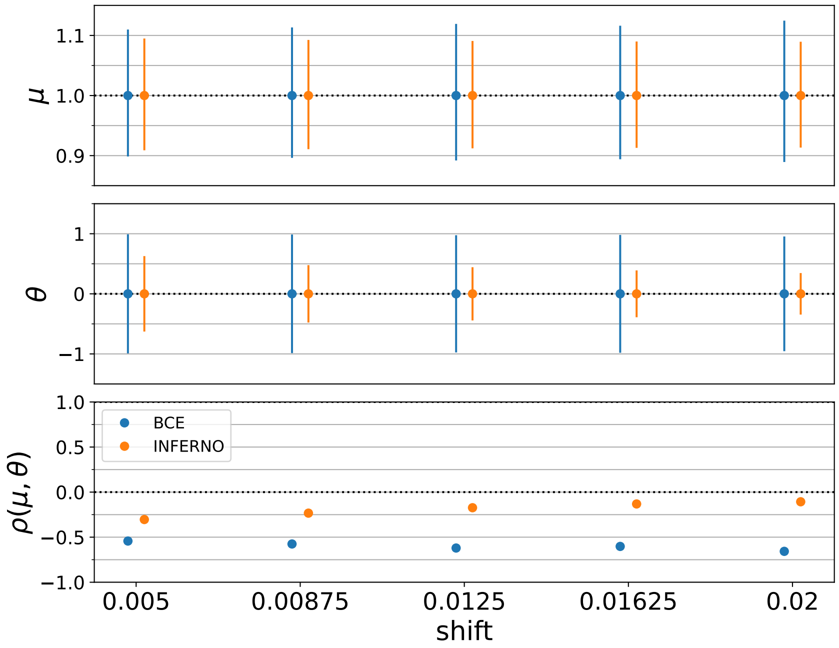


Figure 7.9: Comparison of the confidence intervals for μ (top panel), uncertainty of θ (middle panel) and the correlation coefficient ρ (bottom panel) obtained in a profile likelihood fit.

The study described above has been repeated for five artificial shifts of the aplanarity variable of the signal distribution with values between 0.005 and 0.02. For each shift a separate INFERNO model has been trained and has been compared to a BCE model. Figure 7.9 shows the results of the profile likelihood fits. The top panel displays the MINOS uncertainty of the signal strength μ, the middle panel shows the standard deviation of the nuisance parameter θ and the last panel displays the corre-

lation coefficient ρ between μ and θ. Increasing the magnitude of the shift causes the confidence interval of μ to increase with the BCE summary statistic, while it stays approximately constant with the INFERNO summary statistic. It further is evident that the standard deviation of θ is reduced with INFERNO. Since the parameter θ has been constrained with a Unit Gaussian, this implies that the post-fit uncertainty is reduced with respect to the pre-fit uncertainty. The correlations between μ and θ are closer to zero with the INFERNO summary statistics compared to the BCE summary statistic. This illustrates again that INFERNO makes optimal use of the data to decorrelate the POI μ from the nuisance parameter θ, and also uses the data to constrain the uncertainty of θ.
This study is repeated with a shift of the aplanarity variable of the background distribution. The results obtained in this study are similar to the one described for the signal distribution and a plot with the results of the study has been included in Appendix A.2.

ShapeNorm for Signal and Background

To further quantify the performance of the algorithm, a setup is considered where both the signal and the background process depend on an independent *ShapeNorm* nuisance parameter. This ensures that the INFERNO training works correctly if multiple *ShapeNorm* parameters are included that affect different processes. Therefore, for both the signal and the background process the aplanarity variable is shifted by a value between 0.005 and 0.02. The training of the INFERNO algorithm then includes three parameters: the expected number of signal events s, a nuisance parameter θ^0 corresponding to the shift in the signal distribution and a nuisance parameter θ^1 corresponding to the shift in the background distribution. Both nuisance parameters are constrained with a Unit Gaussian distribution.

The approximated 3×3 covariance matrix is monitored after each epoch on the validation set during the INFERNO and BCE model training. Its evolution with two nuisance parameters θ^0 and θ^1 corresponding to a shift of 0.0125 of the aplanarity variable, is shown in Fig. 7.10 as a function of the number of epochs. The diagonal elements of the figure display the variance of the expected number of signal events $\sigma^2(s)$ and the variance of the nuisance parameters $\sigma^2(\theta^0)$, and $\sigma^2(\theta^1)$. The off-diagonal elements show the correlation coefficients ρ. As has been observed in the previous studies, the variance of the POI s converges to a lower value with an INFERNO model compared to a classifier trained with BCE. The correlation coeffi-

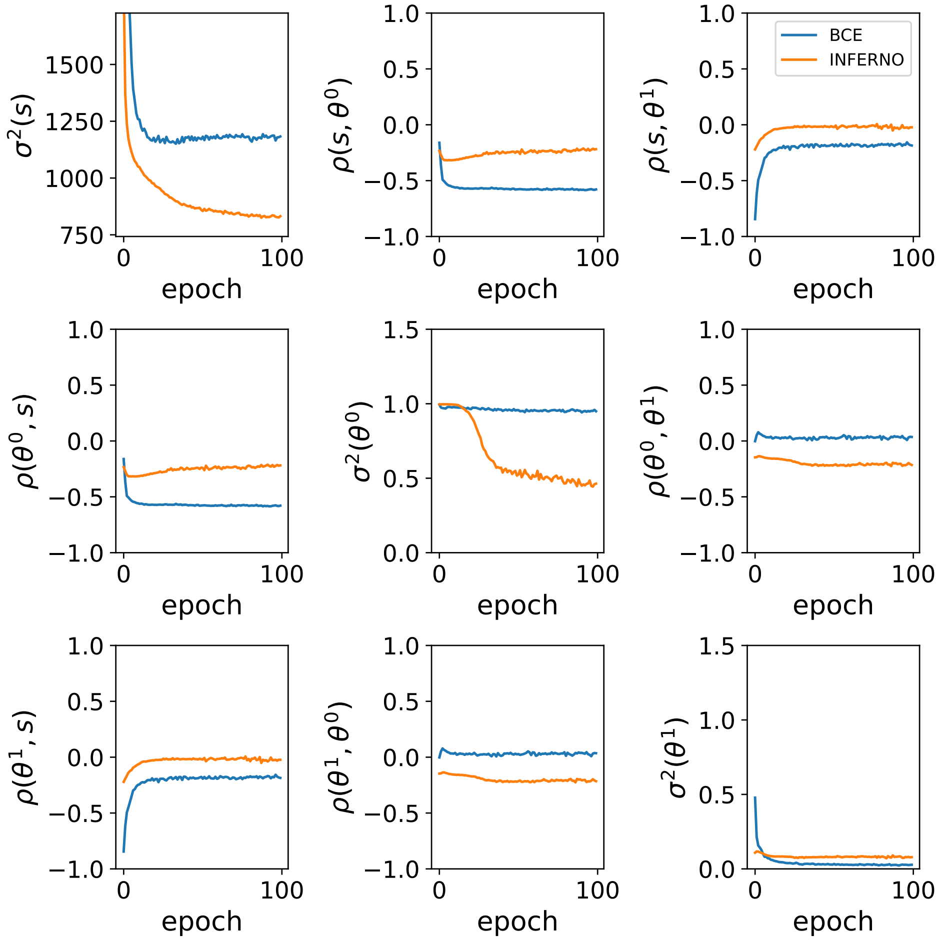


Figure 7.10: Evolution of the 3×3 covariance matrix for the number of expected signal events s and two artificial nuisance parameters corresponding to a shift of 0.0125 in the aplanarity variable of the signal (θ^0) and background (θ^1) process.

cients between the POI s and the nuisance parameters $\rho(s, \theta^0)$ and $\rho(s, \theta^1)$ converge to values closer to zero during the INFERNO model training and the the approximate variance of θ^0 is reduced compared to the BCE model.

Figure 7.11 shows the results of a profile likelihood fit with CABINETRY for artificial shifts of the aplanarity variable between 0.005 and 0.02 in the signal and background distributions. As in the previous study, a separate INFERNO model has been trained for each shift and only the nuisance parameters that have been used during the training are included in the fit. The top panel shows the MINOS uncertainty of the signal strength μ and the following two panels show the uncertainty of the nuisance

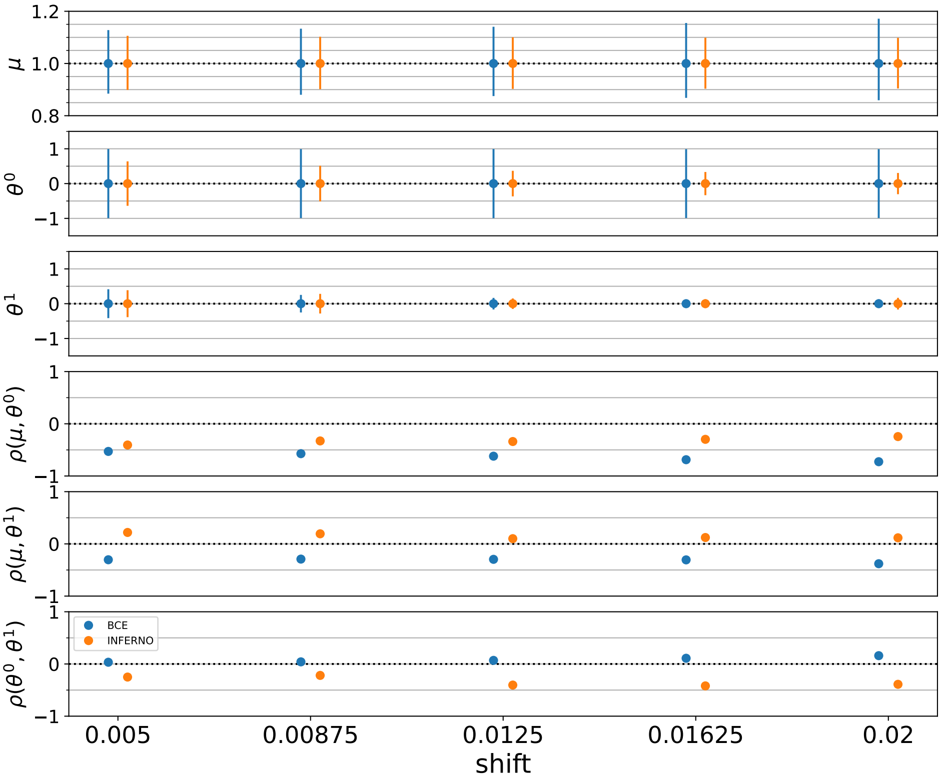


Figure 7.11: Comparison of the confidence intervals for μ (top panel), uncertainty of θ^0 and θ^1 (second and third panel) and the correlation coefficients ρ (bottom three panels) obtained in a profile likelihood fit for the INFERNO and BCE model.

parameters θ^0 and θ^1. The bottom three panels display the three correlation coefficients ρ. The correlations between the signal strength μ and the nuisance parameters θ^0 and θ^1 are closer to zero with the INFERNO summary statistic compared to the BCE summary statistic. Thus the confidence intervals obtained with INFERNO are significantly narrower and the uncertainty of the nuisance parameters θ^0 and θ^1 are reduced. This is consistent with the observations made during the model training, and further strengthens the hypothesis that the INFERNO algorithm obtains a lower variance for the POI s by decorrelating this parameter from the relevant nuisance parameters.

7.2.4 INFERNO with Normalization Nuisance Parameters

In order to evaluate if INFERNO can also outperform a classifier trained with BCE if only normalization uncertainties are considered, a normalization uncertainty has been included in the INFERNO training for both the signal and background process.

An example for a normalization uncertainty is the luminosity. In the studied setup, the INFERNO training consists of two parameters: the number of expected signal events s and the nuisance parameter θ corresponding to a constrained normalization uncertainty. The magnitude of the normalization uncertainty is increased in five steps from 2% to 10% and a separate INFERNO model is trained for each value. The training is performed with the setup described in Section 7.2.1 and in the profile likelihood fit only one normalization nuisance parameter is taken into account, in order to have the same conditions as during the model training. The results of the profile likelihood fit is shown in Fig. 7.12. The left panel shows the MINOS uncertainty of the signal strength μ, the uncertainty of the nuisance parameter θ and the correlation coefficient ρ for a normalization uncertainty affecting the signal distribution. The right panel shows the same study for a normalization uncertainty affecting the background distribution. It is observed that in both studies the fit results obtained with the BCE summary statistic and the INFERNO summary statistic are similar. In particular, for the confidence intervals and the correlations very similar values have been obtained. None of the performed studies have indicated a mitigation of the effect of normalization nuisance parameters by including them in the INFERNO algorithm compared to a BCE classifier. This indicates that for the studied $t\bar{t} \to \tau_h + \text{jets}$ analysis, an improvement with the INFERNO algorithm over the BCE classifier is mainly possible by reducing correlations between the signal strength parameter μ and *ShapeNorm* nuisance parameters that have a strong effect on the shape of the classifier. However, it should be noted that so far INFERNO has

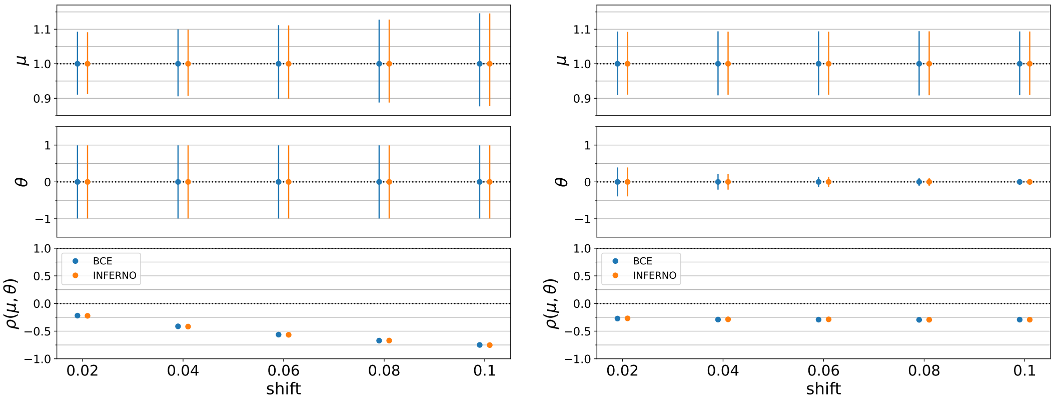


Figure 7.12: Comparison of the confidence intervals for μ (top panel), uncertainty of θ (middle panel) and the correlation coefficient ρ obtained in a profile likelihood fit. The left panel shows a normalization uncertainty affecting the signal and the right panel shows a normalization uncertainty affecting the background.

only be studied with one signal and one background process. It is possible that the inclusion of the normalization uncertainties in INFERNO may be beneficial in cases where multiple background or signal processes are considered in the training, where each process has its own normalization uncertainties, since in this case different background processes could be assigned to different bins.

7.2.5 Alternative Summary Statistic

As described in Section 7.1.3, an alternative differentiable summary statistic has been implemented based on an approximately differentiable histogram. The studies described in the previous sections have been repeated with this summary statistic and similar results have been obtained as for the original INFERNO summary statistic based on a softmax function. To use the approximately differentiable histogram, a feed-forward neural network with two hidden layers, ReLU actications and a sigmoid function in the last layer has been implemented. A value of 200 has been used for the hyperparameter κ that regulates the smoothness of the histogram bin edges. An example for the training with one artificial *ShapeNorm* nuisance parameter corresponding to a shift of 0.0125 in the aplanarity variable of the signal distribution, as studied in Section 7.2.3, is shown in the left panel of Fig 7.13. The

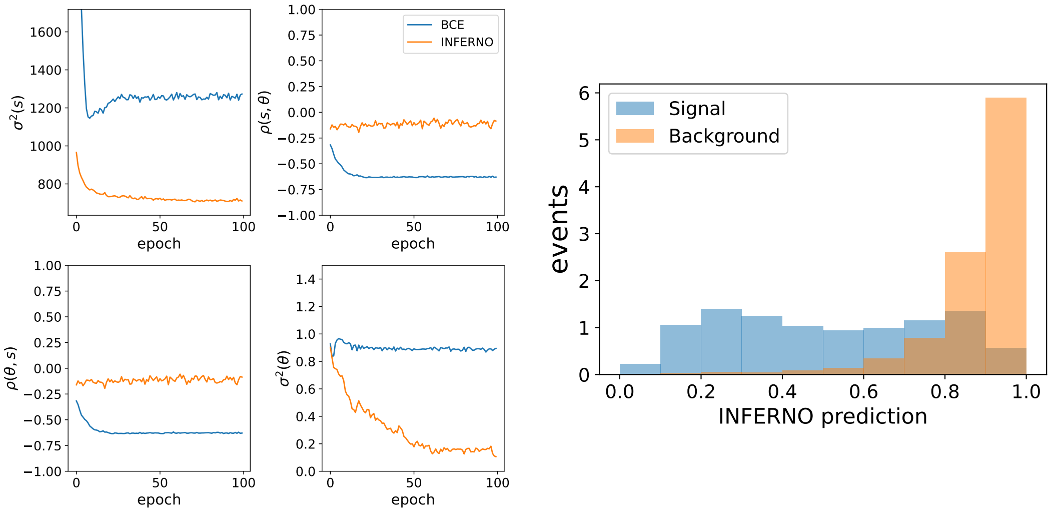


Figure 7.13: Left panel: Evolution of the covariance matrix evaluated on the validation set during the training for an artificial *ShapeNorm* nuisance parameter corresponding to a shift of 0.0125 in the aplanarity variable of the signal distribution. Right panel: predictions of the INFERNO model trained with an approximately differentiable histogram.

approximated covariance matrix shows a similar behaviour as the training with the original INFERNO summary statistic. A lower variance is obtained for the POI with the INFERNO model compared to the BCE model and the correlation between the POI and the nuisance parameter is reduced, which is consistent with the findings in the previous sections.

The predictions of the INFERNO model for the validation set is shown in the right panel of Fig. 7.13. The output of this summary statistic is a continuous variable that is bound between zero and one which has the advantage that it can be easily rebinned. The shape of the predictions shows similarities to the predictions of a model trained with BCE, where the signal and background are pushed towards zero and one. A good data-simulation agreement is observed, as displayed in the left panel of Fig. 7.14. The right panel displays a profile likelihood scan evaluated on Asimov data that has been performed with CABINETRY. A similar confidence interval has been obtained as for the original INFERNO summary statistic, shown in Fig. 7.8. This indicates that both summary statistics described in Section 7.1.3 can be used to train a model with the INFERNO algorithm depending on the concrete application.

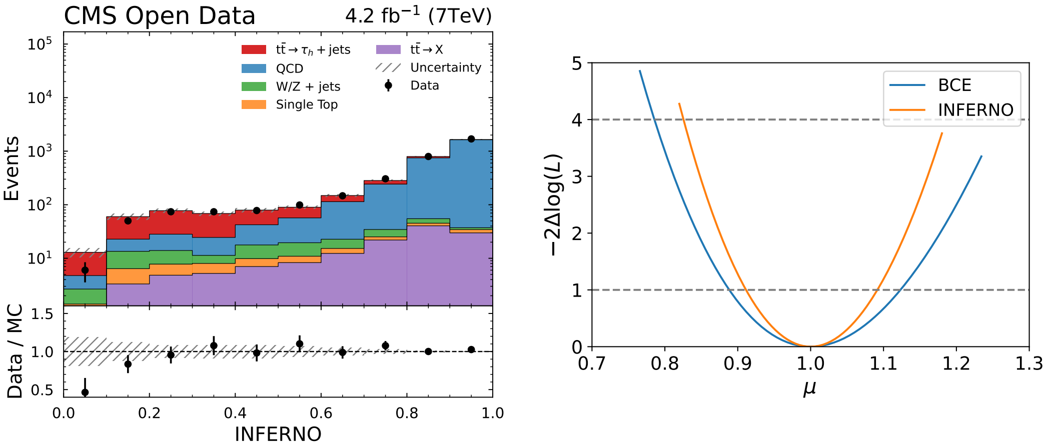


Figure 7.14: Left panel: data-simulation agreement of the predictions obtained from the INFERNO model trained with an approximately differentiable histogram. Right panel: profile likelihood scan for the INFERNO and BCE model on Asimov data.

7.3 Measurement of the $t\bar{t} \rightarrow \tau_h + \text{jets}$ Cross-Section with INFERNO

In this section the INFERNO algorithm will be applied to the $t\bar{t} \rightarrow \tau_h + \text{jets}$ analysis reproduced with CMS Open Data. The studies of the *ShapeNorm* and normalization nuisance parameters in the previous section indicate, that INFERNO has the potential to mitigate the effect of systematic uncertainties that affect the shape of the classifier, whereas in the studied setup INFERNO does not improve normalization uncertainties. Therefore, first a single INFERNO model is trained for each of the considered *ShapeNorm* uncertainties in the $t\bar{t} \rightarrow \tau_h + \text{jets}$ analysis in order to obtain an estimation which uncertainties INFERNO can potentially improve. Then a model is trained that takes all relevant *ShapeNorm* uncertainties and their correlations into account and a profile likelihood fit with all relevant systematic uncertainties is performed to compare the cross-section obtained with INFERNO to the results in Chapter 6.

7.3.1 Evaluation of ShapeNorm Uncertainties

As discussed in Section 6.7, there are six systematic uncertainties in the $t\bar{t} \rightarrow \tau_h + \text{jets}$ analysis that affect the shape and normalization of the classifier (Table 6.5). The systematic uncertainties due to the jet energy scale (JES), the jet energy resolution (JER), and the tau energy scale (TauE) are obtained by repeating the analysis with varied parameters in the MC simulation. The remaining three systematics are given by event weights: the systematic uncertainty of the PDF weights, the statistical uncertainty originating from the trigger efficiency measurement, and the variation of the b-tagging scale factor. To understand the impact of each *ShapeNorm* systematic uncertainty a separate INFERNO model is trained for each systematic variation and the results of a profile likelihood fit based on the summary statistic obtained with INFERNO is compared to a summary statistic obtained from an optimized classifier trained with BCE. The training and inference setup is the same as described in Section 7.2.1: a feed-forward neural network with two hidden layers with 60 neurons per layer, 10 output nodes, a temperature τ of 0.1 and a learning rate of 0.001 is trained for 100 epochs with the ADAM optimizer. Since the 1σ-up and 1σ-down variations have different normalizations, the asymmetric log-normal function is used in the INFERNO algorithm. The profile likelihood fit is performed with CABINETRY.

Figure 7.15 shows the evolution of the 2×2 covariance matrix during the train-

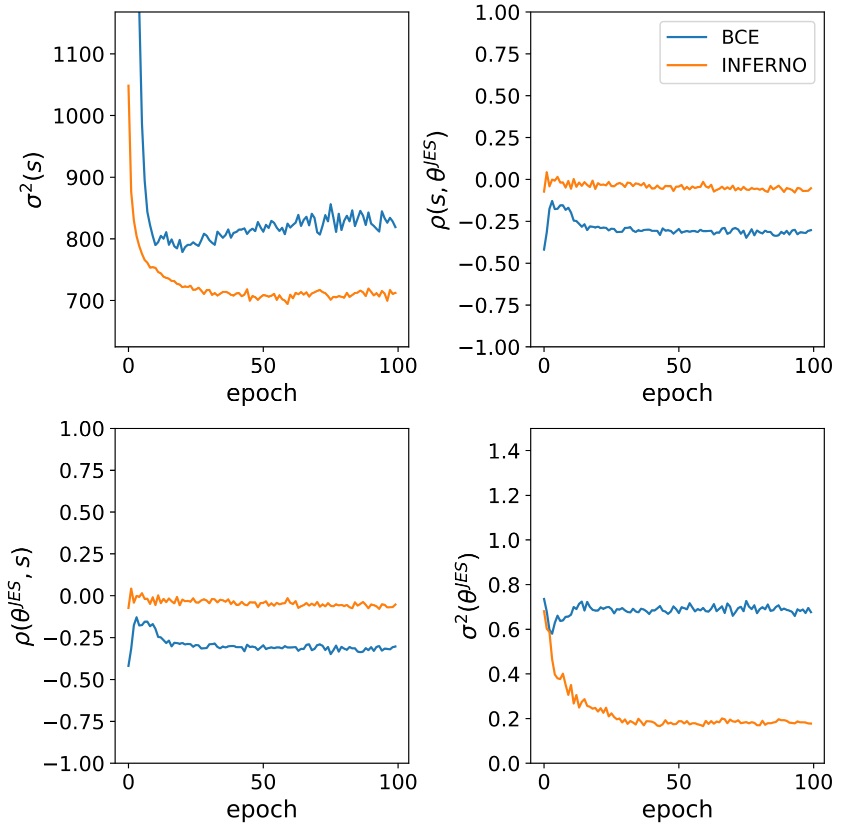


Figure 7.15: Evolution of the covariance matrix evaluated on the validation set during the training. The diagonal elements show the variance of the expected number of signal events s and the variance of the JES nuisance parameter θ^{JES}. The off diagonal elements show the correlation coefficient $\rho(s, \theta^{\mathrm{JES}})$.

ing with the JES systematic uncertainty for an INFERNO and BCE model. The evolution of the variances of the expected number of signal events s and the nuisance parameter θ^{JES} are qualitatively similar to the study with artificial nuisance parameters in Section 7.2.3. The INFERNO model converges to a lower variance $\sigma^2(s)$ compared to the BCE model and the variance of the nuisance parameter θ^{JES} is reduced. The correlation coefficient ρ between s and θ^{JES} converges to a value closer to zero with the INFERNO model, which is consistent with the previous studies. The obtained summary statistics with the INFERNO and BCE model are used as input to the profile likelihood fit. A scan of the profile likelihood on Asimov data for the signal strength μ is shown in Fig. 7.16 and the measured values for the MINOS error of μ, the uncertainty of the nuisance parameter θ^{JES} and the correlation coefficient ρ is displayed in Fig. 7.17. The fit has been performed with the same conditions

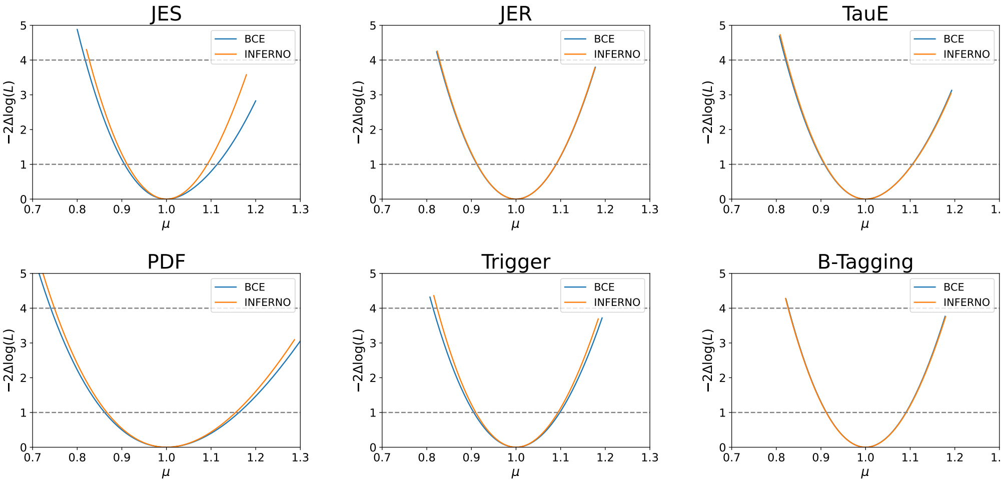


Figure 7.16: Profile likelihood scans of the signal strength μ on Asimov data for the considered *ShapeNorm* nuisance parameters for the BCE and INFERNO model.

that were used during the training of INFERNO and the results confirm the trends that have been seen during the training: a more precise confidence interval has been obtained by including the JES systematic in the training with INFERNO and the correlation between μ and θ^{JES} has been reduced.

For each of the remaining *ShapeNorm* systematic uncertainties a similar model has been trained and the profile likelihood fit has been performed with the obtained summary statistics. In Fig. 7.16 the likelihood scans for the signal strength parameter μ is shown for the six relevant *ShapeNorm* systematic uncertainties and in Fig. 7.17 the MINOS uncertainty for μ, the uncertainty of the corresponding nuisance parameter θ and the correlation coefficient ρ is shown. Comparing the results between the INFERNO and BCE model shows that an improvement is mainly possible for the JES nuisance parameter, where INFERNO manages to decorrelate it from the signal strength μ. For the other nuisance parameters the obtained confidence intervals and correlations are very similar and little improvement is obtained with INFERNO. A comparison of the signal and background shapes and the shapes of the systematic variations is included in Appendix A.2 for the INFERNO model (Fig. A.5) and the BCE model (Fig. A.6). This comparison shows that, except for the JES variation, the other nuisance parameters have only a small influence on the shape of the classifier. Thus INFERNO cannot decorrelate these parameters from the POI, and hardly any improvement with INFERNO over BCE is obtained.

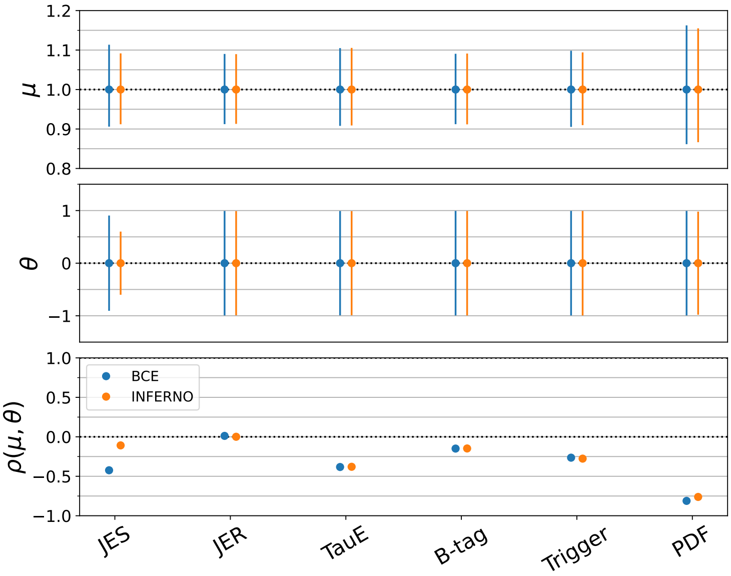


Figure 7.17: Comparison of the confidence intervals for μ (top panel), uncertainty of the respective *ShapeNorm* nuisance parameter θ (middle panel) and the correlation coefficient ρ obtained in a profile likelihood fit for each of the considered *ShapeNorm* nuisance parameters.

7.3.2 Model Training

For the complete cross-section measurement, a model is trained that takes all relevant *ShapeNorm* nuisance parameters discussed in the previous section into account. Thus the model consists of seven parameters: the signal strength μ, and six nuisance parameters corresponding to the JES variation, the JER variation, the tau energy scale variation, the PDF variation, the b-tagging variation and the variation corresponding to the trigger efficiency. As in the previous studies, the chosen number of training events consists of 5000 QCD background and 20000 $t\bar{t}$ signal events, while the validation set consists of 5600 background and 23000 signal events. A feed-forward neural network with two hidden layers, ReLU activations and 10 output nodes in the final layer is used for the INFERNO training with the summary statistic based on the softmax function defined in equation 7.13. A hyperparamter scan has been performed to optimize the learning rate in the range $[10e^{-4}, 10e^{-1}]$, the number of neurons in the range $[20, 100]$ and the temperature in the range $[0.01, 0.99]$. The best loss on the validation set has been used as figure of merit. For each set of hyperparamters three randomly initialized models have been trained and it has been found that slightly higher temperatures are favourable. Figure 7.18 shows the

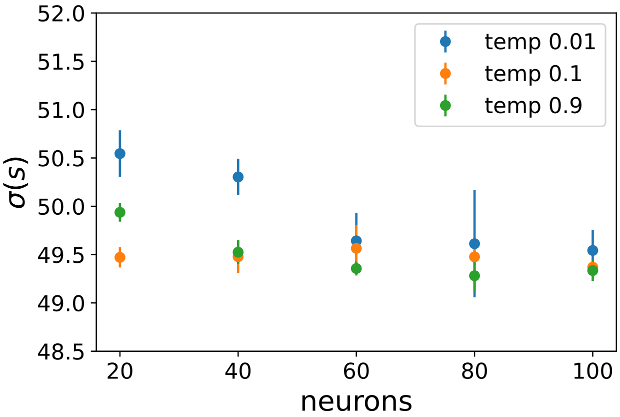


Figure 7.18: Square root of the best INFERNO validation loss for different numbers of neurons per layer and three values of the temperature for a fixed learning rate of 0.001.

square root of the best validation loss $\sigma^2(s)$ for different numbers of neurons and three values of the temperature with a fixed learning rate of 0.001. Finally, an INFERNO model with 60 neurons per layer, a learning rate of 0.001 and a temperature τ of 0.9 has been trained with a batch size of 1000 for 100 epochs with the ADAM optimizer. The INFERNO model is compared to the optimized BCE model described in Section 6.8.

The evolution of a 4×4 subset of the covariance matrix as a function of the number of epochs is shown in Fig. 7.19 for the number of expected signal events s and the three most important *ShapeNorm* nuisance parameters θ^{JES}, θ^{TauE} and θ^{PDF}. The variances and correlations have also been calculated for a model trained with BCE by histogramming the BCE predictions to create a summary statistic and running the INFERNO algorithm to obtain the approximated covariance matrix. Comparing the evolution of the variances and correlations between the BCE and INFERNO model indicates that there is a moderate improvement in the variance of the POI s and INFERNO manages to reduce the correlation between some of the nuisance parameters, particularly of the nuisance parameter corresponding to the JES variation. The correlations and variances for the tau energy scale θ^{TauE} and the PDF θ^{PDF} converge to similar values during the INFERNO and BCE model training. This behaviour is expected according to the evaluation of the individual nuisance parameters in the previous section. The data-simulation agreement for the INFERNO class predictions is shown in the left panel of Fig. 7.20 and a good agreement between data and simulation is obtained.

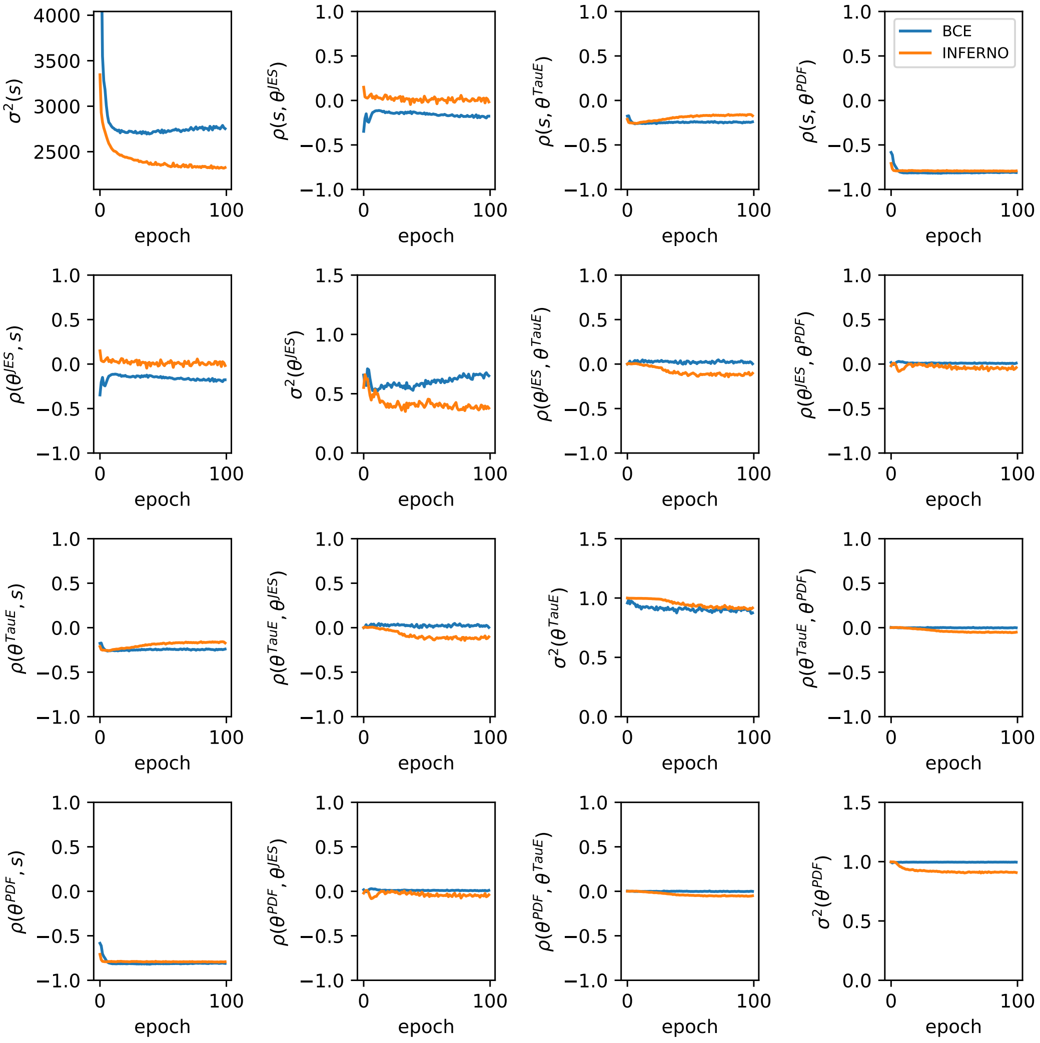


Figure 7.19: Evolution of a 4 × 4 subset of the variances (diagonal) and correlations (off-diagonal) evaluated on the validation set for the INFERNO and BCE model. The number of expected signal events s, and the three most important *ShapeNorm* nuisance parameters for the JES θ^{JES}, the tau energy scale θ^{TauE} and the PDF θ^{PDF} are shown.

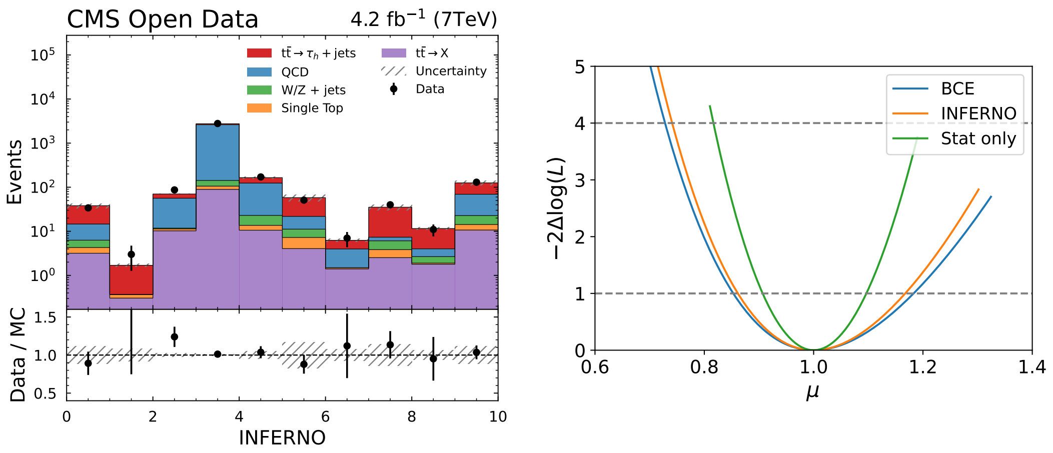


Figure 7.20: Left panel: data-simulation agreement for the INFERNO model taking into account all relevant *ShapeNorm* nuisance parameters. Right panel: profile likelihood scan on Asimov data for the INFERNO and BCE model.

In order to verify the approximations made during the training, the resulting summary statistics are used as input for a profile likelihood fit to the Asimov data where the same nuisance parameters as used during the training are included, i.e. the six shape norm nuisance parameters that affect the $t\bar{t}$ signal. The model training and

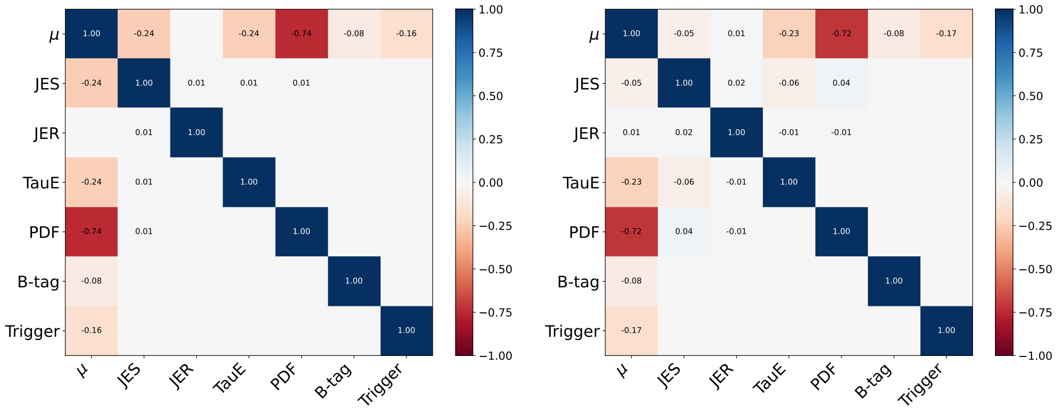


Figure 7.21: Left panel: correlation matrix for the BCE model. Right panel: correlation matrix for the INFERNO model.

profile likelihood fit has been repeated with ten randomly initialized models, both for the INFERNO and BCE summary statistics. The measured values are

$$\begin{aligned}\mu^{A}_{\mathrm{BCE}} &= 1.00^{+0.1864\ (\pm 0.0013)}_{-0.1479\ (\pm 0.0005)} \\ \mu^{A}_{\mathrm{INF}} &= 1.00^{+0.1735\ (\pm 0.0021)}_{-0.1401\ (\pm 0.0007)}\end{aligned} \tag{7.31}$$

where the upper and lower MINOS bound of the confidence interval has been averaged and the standard deviation of the bounds obtained from the ten fits is quoted. The obtained confidence interval is slightly narrower for INFERNO, as expected from the evaluation of the model training. The profile likelihood scan for the signal strength μ evaluated on the Asimov data is shown in the right panel of Fig. 7.20. The correlation matrix obtained from one profile likelihood fit is shown in Fig. 7.21 for the BCE model (left panel) and the INFERNO model (right panel). The correlation between μ and θ^{JES} is reduced with the INFERNO model, as was observed during the model training. Comparing the correlation matrices to the approximated correlations with the INFERNO algorithm in Fig. 7.19 shows, that they are in good agreement. In general, the improvement with the INFERNO algorithm is moderate since, as discussed in Section 7.3.1, except of the JES systematic most of the *ShapeNorm* nuisance parameters only have a small effect on the shape of the classifier in which case the training with BCE is a very good approximation.

Source	$t\bar{t} \to \tau_h + \text{jets}$	$t\bar{t} \to X$	W/Z + jets	Single-top	QCD
JES	✓	✓	✓	✓	
JER	✓	✓	✓	✓	
τ_h scale	✓	✓	✓	✓	
PDF	✓	✓	✓	✓	
Stat. trigger	✓	✓	✓	✓	
b-tagging	✓	✓	✓	✓	
Cross-section	✓	✓	✓	✓	
Top-quark mass	✓	✓			
Renorm. scale	✓	✓			
PS matching	✓	✓			
τ_h trigger	✓	✓	✓	✓	
τ_h identification	✓	✓	✓	✓	
Multijet norm					✓

Table 7.1: Relevant systematic uncertainties for the signal and background processes in the $t\bar{t} \to \tau_h$+jets analysis. The first block are the systematic uncertainties that affect the shape and the normalization. The second block lists the uncertainties that only affect the normalization.

7.3.3 Cross-Section Measurement with inferno

Based on the model trained in the previous section, the inference is performed including all relevant nuisance parameters for the various processes, as done in the measurement of the cross-section in Section 6.9. Table 7.1 summarizes the considered systematic uncertainties and lists the signal and background processes that are affected by the different systematic sources. During the training and fit in the previous studies, the systematic uncertainties have only be taken into account for the signal and background process that were used in the inferno training. In order to compare with the result obtained for the signal strength μ in Chapter 6, the full profile likelihood fit that includes all relevant nuisance parameters for all relevant processes is performed with cabinetry. This makes the assumption that the effect of the minor backgrounds were negligible for the training of inferno and the obtained summary statistic is still optimal.

The resulting confidence interval for μ based on the inferno summary statistic evaluated on Asimov data is measured to:

$$\mu^{A}_{\text{INF}} = 1.00^{+0.22}_{-0.17}\,(\text{syst.}) \pm 0.09\,(\text{stat.}) \tag{7.32}$$

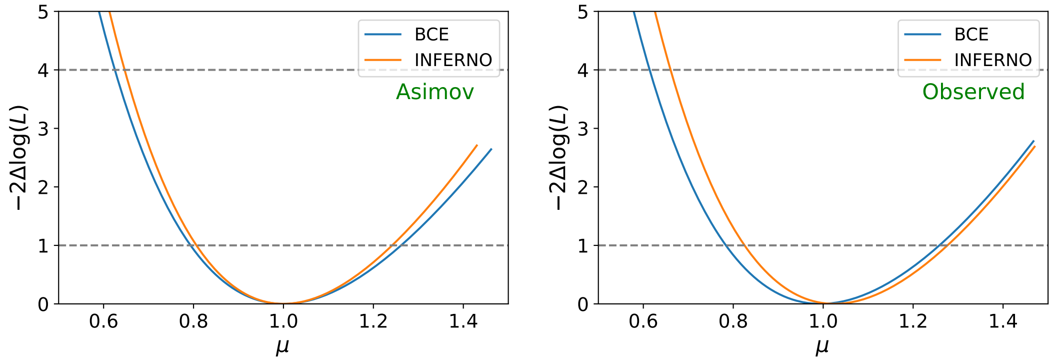


Figure 7.22: Left panel: comparison of the profile likelihood scan for a fit to Asimov data for the INFERNO and BCE model with all relevant nuisance parameters included. Right panel: comparison of the profile likelihood scan for a fit to the observed data for the INFERNO and BCE model with all relevant nuisance parameters included.

and the resulting confidence interval for μ evaluated on the observed data is measured to:

$$\mu_{\mathrm{INF}} = 1.02^{+0.23}_{-0.18}\ (\mathrm{syst.}) \pm 0.09\ (\mathrm{stat.})\ . \tag{7.33}$$

The profile likelihood scan of μ evaluated on Asimov data is shown in the left panel of Fig. 7.22 and the profile likelihood scan evaluated on the observed data is displayed in the right panel. The likelihood scans obtained with the classifier trained with BCE in Chapter 6.9 are included for comparison. The central values measured for the signal strength μ are in good agreement. A moderate improvement in the precision of the confidence interval is obtained with the INFERNO summary statistic compared to the BCE summary statistic. In Fig. 7.23 the impacts and the pulls for the profile likelihood fit with the INFERNO summary statistic is shown for Asimov data in the left panel and for the observed data in the right panel. Comparing the impacts for the INFERNO model with the impacts for the BCE model obtained in Section 6.9, shows that the INFERNO model reduces the impact and uncertainty of the JES nuisance parameter, which is consistent with the studies in the previous sections. Also the impact of the most important uncertainty that stems from the PDF variation is slightly reduced compared to the BCE model. Thus, the moderate improvement in the confidence interval of μ can be explained by the mitigation of the effect of the *ShapeNorm* nuisance parameters that have been included in the INFERNO training. The correlation matrices obtained in the profile likelihood fit have been included in Appendix A.2.

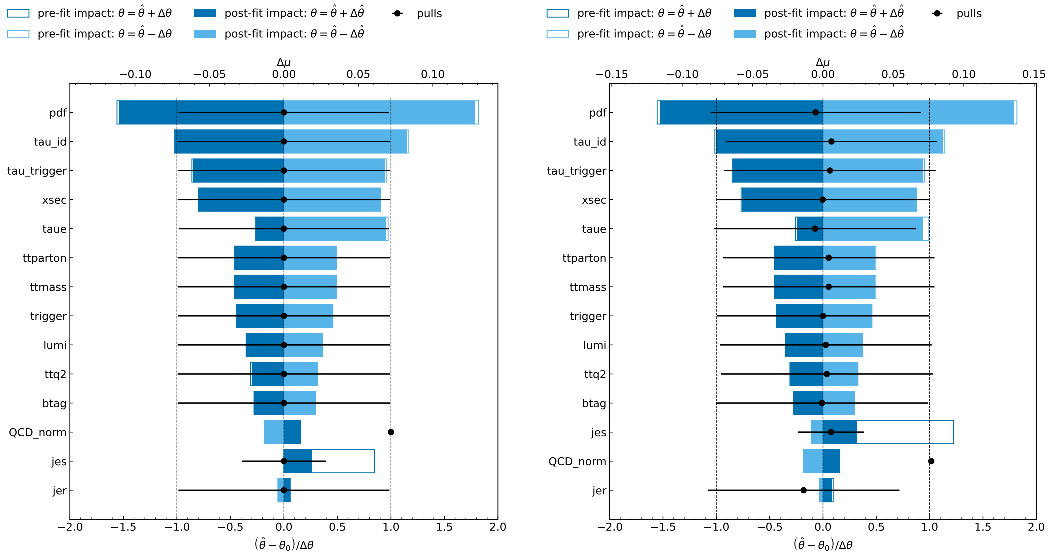


Figure 7.23: Left panel: impacts for the INFERNO model evaluated on Asimov data. Right panel: impacts for the INFERNO model evaluated on the observed data

This result shows that INFERNO can be used in a realistic LHC analysis with HEP-like systematic uncertainties. INFERNO has the potential to improve the precision of confidence intervals if nuisance parameters are present that affect the shape of the classifier by reducing the correlations between the parameter of interest and the relevant nuisance parameters. The calculation of the covariance matrix in the INFERNO algorithm is a good approximation to the values obtained during inference. A possible improvement would be the extension of the INFERNO algorithm to take all relevant background processes into account, as well as potentially multiple signal processes, in order to ensure that the obtained summary statistic is fully optimal for the inference problem.

8 Conclusions and Prospects

A crucial ingredient in precision measurements and searches for new physics at the LHC is the construction of powerful low-dimensional summary statistics by training neural network classifiers with a cross-entropy loss function to distinguish small signals from large backgrounds in a multi-dimensional space of observed event features. The summary statistics are used as input for the statistical inference and the use of neural network classifiers has significantly improved the precision of the LHC measurements in the last decade. However, the cross-entropy loss and the standard measures of performance for the learning task are not aligned with the inference goal when the simulations depend on nuisance parameters that represent systematic uncertainties. The presence of nuisance parameters then causes a reduction of the statistical power of the summary statistics during inference.

To address this problem, the novel INFERNO technique, that constructs powerful summary statistics that account directly for the final inference objective, has been extended to deal with systematic uncertainties that are common for HEP problems. The main idea of the INFERNO technique is the minimization of a loss function of a neural network that approximates the covariance matrix of the parameter of interest and the nuisance parameters with automatic differentiation. The expected variance of the the parameter of interest can then be used to optimize the parameters of the neural network and the output of the model is the optimal summary statistic that accounts for the effect of the nuisance parameters. The strength of the INFERNO technique is that it allows to formally describe the objective of the problem one tries to solve and does not rely on a standard loss-function such as the binary cross-entropy that is not aligned with the inference problem in the presence of nuisance parameters. In particular, not only approximate variances but also likelihood ratios can be optimized. The extension of the INFERNO algorithm to HEP problems is based on a differentiable morphing algorithm that allows to interpolate between the nominal summary statistic and 1σ-up and 1σ-down variations corresponding to systematic uncertainties, inspired by techniques used in the CMS COMBINE tool.

Two differentiable summary statistics have been implemented that allow to take event weights into account. The INFERNO procedure is inspired by the standard machine learning and inference procedures used in HEP and does not require a large paradigm change, such as simulation based inference, which makes it easier to be put in practice.

In order to test and benchmark the INFERNO algorithm, a systematics-dominated analysis of the CMS experiment, “Measurement of the $\mathrm{t\bar{t}}$ production cross section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV” has been reproduced with CMS Open Data. The code is released to the public [1] and the analysis can serve as a benchmark for future studies. The observed signal strength measured with a profile likelihood fit based on a summary statistic obtained by training a neural network with a binary cross-entropy loss is:

$$\mu_{\mathrm{BCE}} = 0.99^{+0.25}_{-0.19}\ (\mathrm{syst.}) \pm 0.09\ (\mathrm{stat.}) \tag{8.1}$$

which is in agreement with the original analysis and the SM prediction. Based on the reproduced analysis, several studies have been performed that compare the inference with summary statistics obtained with INFERNO to summary statistics obtained by training a model with binary cross-entropy. In simplified setups it has been shown that INFERNO has the potential to mitigate the effect of nuisance parameters that affect the shape of the classifier by decorrelating the expected number of signal events from the nuisance parameters which improves the precision of the confidence intervals during inference. For the reproduced top pair cross-section measurement it has been shown that the impact of the jet energy scale systematic can be reduced and a moderate improvement in the resulting confidence interval has been obtained. The observed signal strength measured with an INFERNO summary statistic is:

$$\mu_{\mathrm{INF}} = 1.02^{+0.23}_{-0.18}\ (\mathrm{syst.}) \pm 0.09\ (\mathrm{stat.})\ . \tag{8.2}$$

The analysis demonstrates that the construction of summary statistics with INFERNO has the potential to improve the precision of LHC analysis that are dominated by systematic uncertainties that affect the shape of event features used to train probabilistic classifiers.

A vast physics program is expected at the CMS experiment in the next decades. Since so far the LHC has not observed any new physics yet, model independent searches will become increasingly important and the understanding and mitigation of systematic uncertainties will be crucial for precise measurements. A potential next step in the development of INFERNO is the extension of the algorithm to take multiple background processes and channels into account. In the near future, it is planned to apply the INFERNO algorithm in a novel CMS physics analysis. In the larger context, INFERNO is part of an ambitious new program developed by the MODE collaboration [6] that aims at optimizing complete analysis workflows, such as the end-to-end optimization of detectors. Making optimal use of the data recorded by the LHC in the next decades and optimal use of the financial resources to build new detectors will be crucial for the progress of high energy physics. The INFERNO technique is an important step to understand and unlock the full potential to optimize complex workflows by using modern machine learning techniques with custom loss functions that are aligned with the exact objective of the problem.

A Supplementary Material

A.1 Additional Figures for Chapter 6

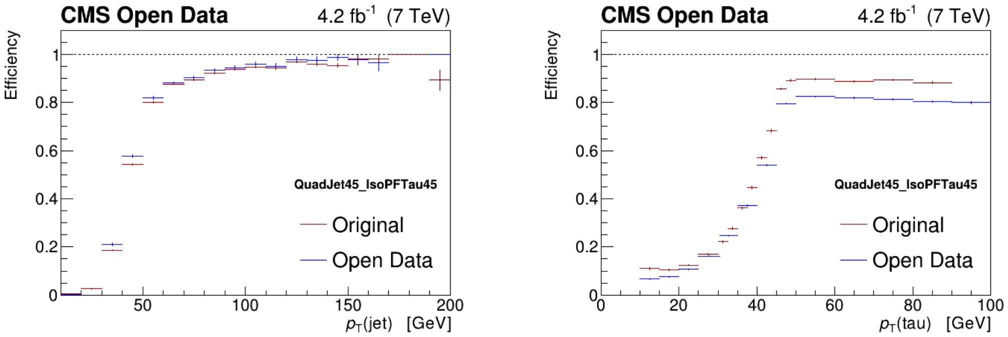


Figure A.1: Left panel: jet trigger efficiency comparison between the reproduced analysis with CMS Open Data and the original analysis [12] for the QuadJet45_IsoPFTau45 trigger. Right panel: tau-lepton trigger efficiency comparison between the reproduced analysis with CMS Open Data and the original analysis [12] for the QuadJet45_IsoPFTau45 trigger.

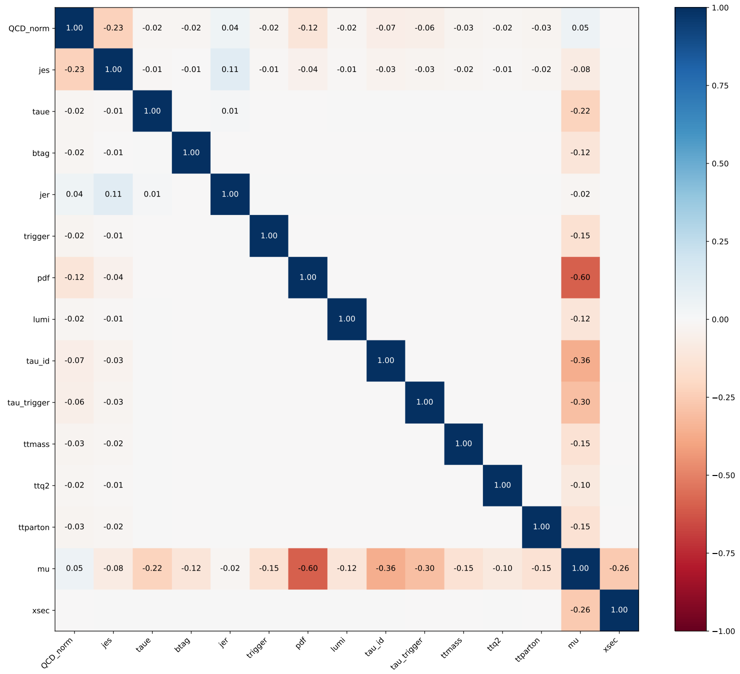


Figure A.2: Correlation matrix for the profile likelihood fit with the BCE model to Asimov data.

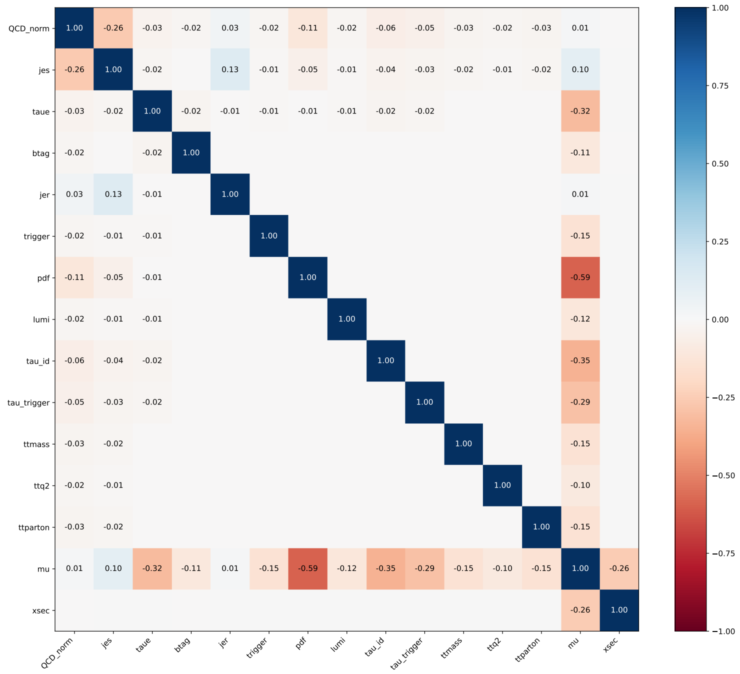


Figure A.3: Correlation matrix for the profile likelihood fit with the BCE model to measured data.

A.2 Additional Figures for Chapter 7

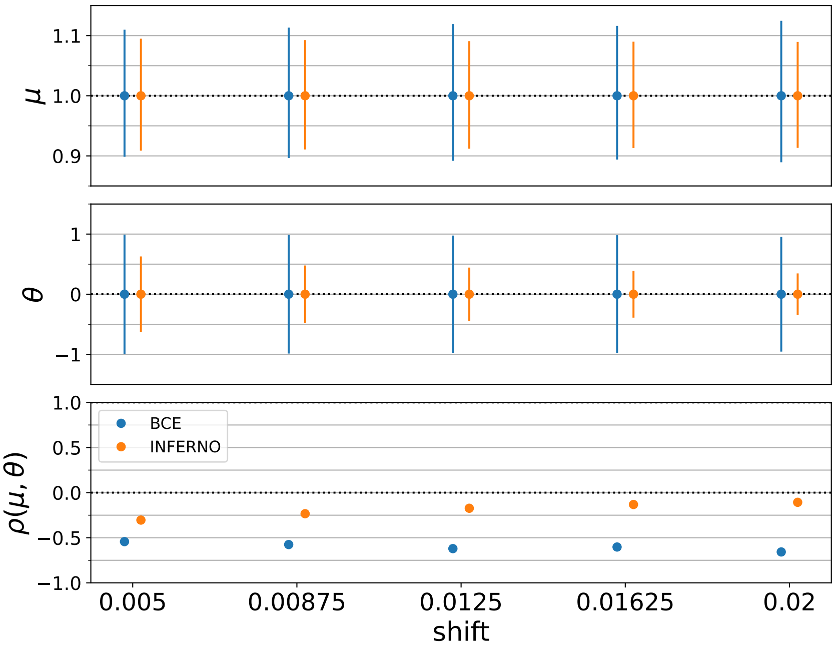
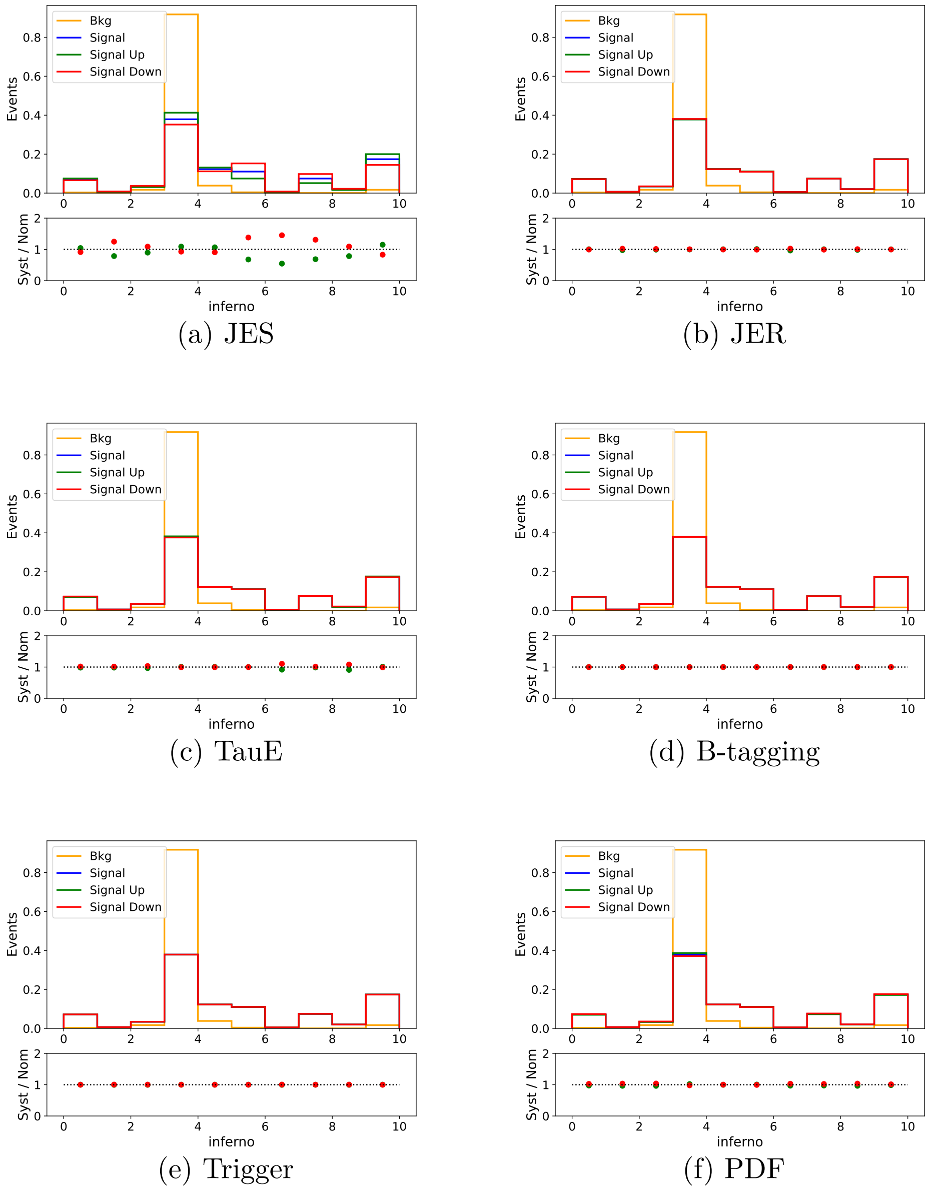


Figure A.4: Comparison of the confidence intervals for μ (top panel), uncertainty of θ (middle panel) and the correlation coefficient ρ obtained in a profile likelihood fit for an artificial shift affecting the background.



(a) JES

(b) JER

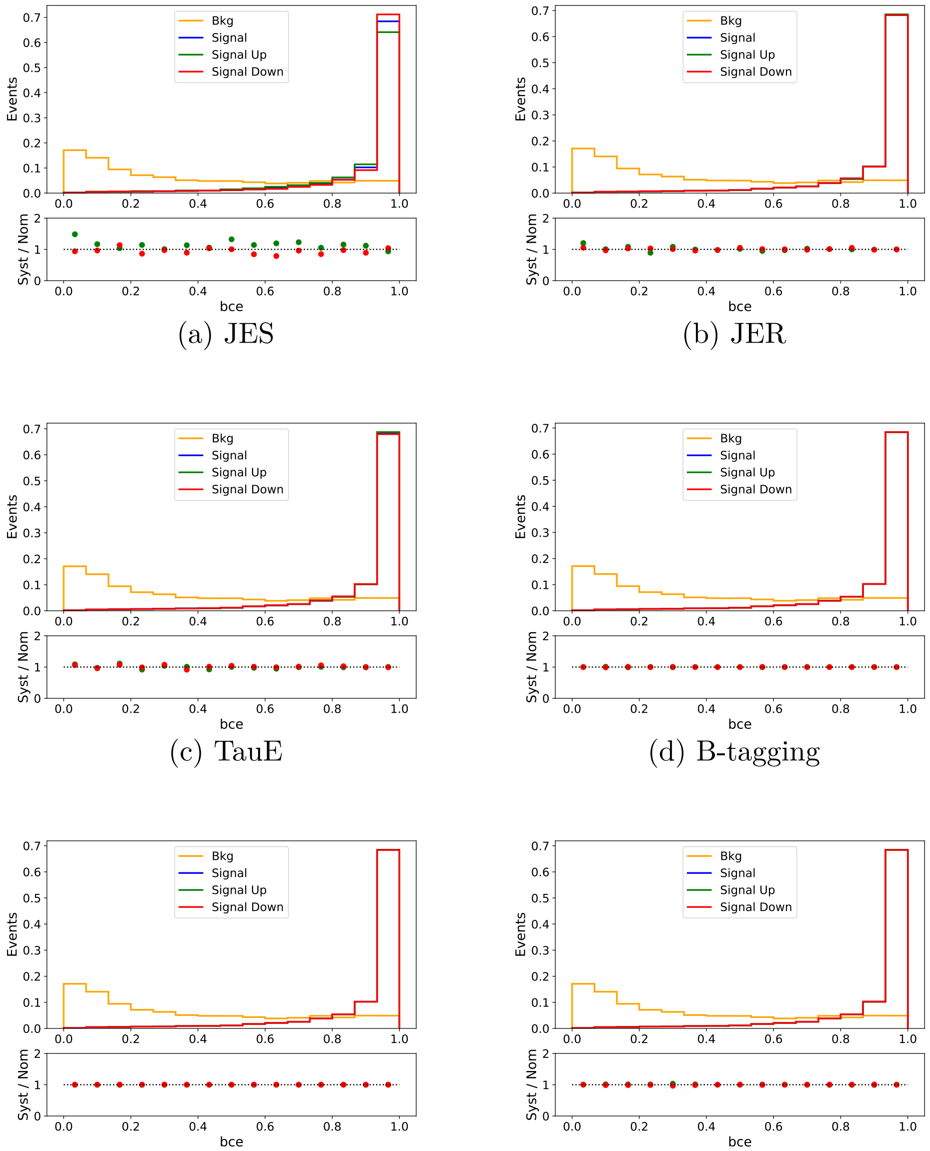
(c) TauE

(d) B-tagging

(e) Trigger

(f) PDF

Figure A.5: Shapes of the signal, background and systematic variations for a classifier trained with INFERNO for the relevant ShapeNorm parameters.



(a) JES

(b) JER

(c) TauE

(d) B-tagging

(e) Trigger

(f) PDF

Figure A.6: Shapes of the signal, background and systematic variations for a classifier trained with BCE for the relevant ShapeNorm parameters.

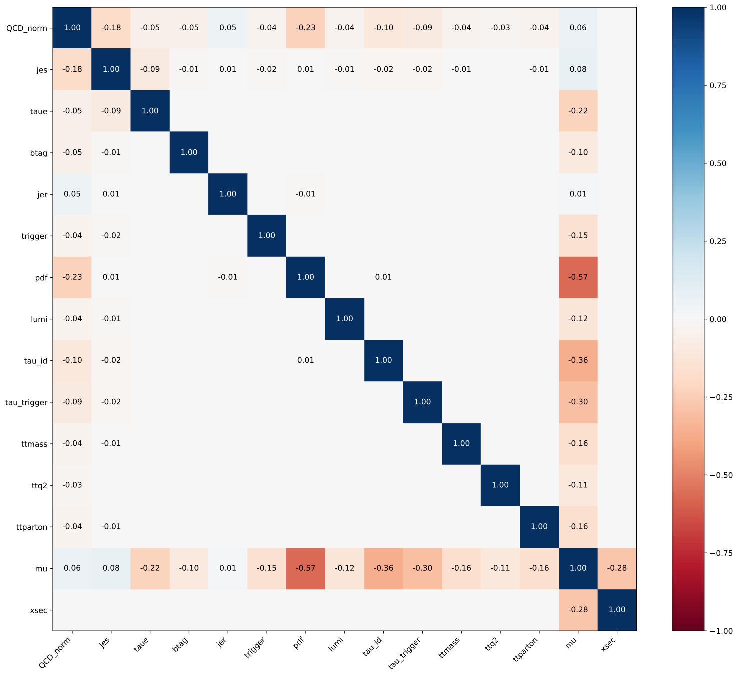


Figure A.7: Correlation matrix for the profile likelihood fit with the INFERNO model to Asimov data.

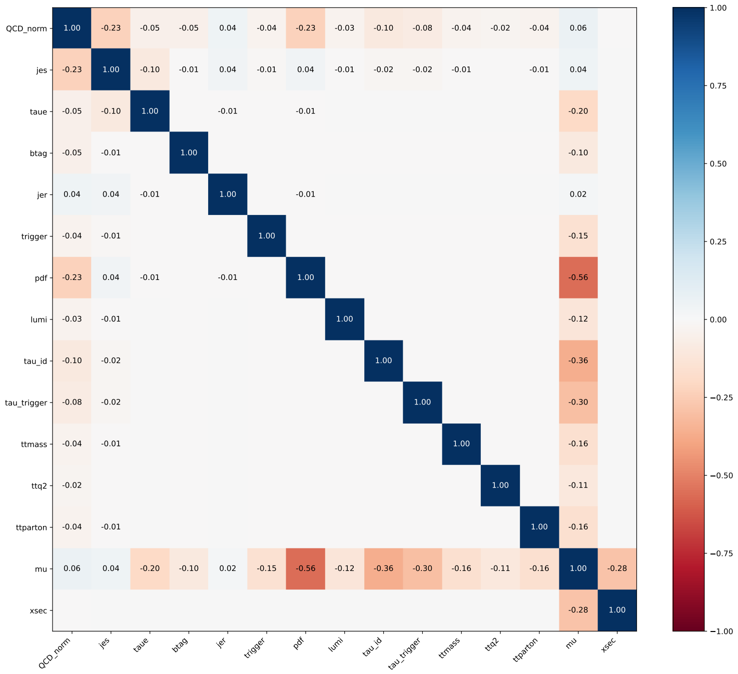


Figure A.8: Correlation matrix for the profile likelihood fit with the INFERNO model to measured data.

Bibliography

1. L. Layer. *llayer/cmsopen: INFERNO for CMS Open Data.* Version v0.0.1. 2022. DOI: 10.5281/zenodo.6080791.
2. L. Layer, D. R. Abercrombie, H. Bakhshiansohi, J. Adelman-McCarthy, S. Agarwal, A. V. Hernandez, W. Si, and J.-R. Vlimant. “Automatic log analysis with NLP for the CMS workflow handling”. *EPJ Web Conf.* 245, 2020, p. 03006. DOI: 10.1051/epjconf/202024503006.
3. CMS Collaboration. “Measurement of CKM matrix elements in single top quark t-channel production in proton-proton collisions at s=13 TeV”. *Physics Letters B* 808, 2020, p. 135609. DOI: 10.1016/j.physletb.2020.135609.
4. F. Giudicepietro, A. M. Esposito, L. Spina, A. Cannata, D. Morgavi, L. Layer, and G. Macedonio. “Clustering of Experimental Seismo-Acoustic Events Using Self-Organizing Map (SOM)”. *Frontiers in Earth Science* 8, 2021. DOI: 10.3389/feart.2020.581742.
5. F. Giudicepietro et al. “Changes in the Eruptive Style of Stromboli Volcano before the 2019 Paroxysmal Phase Discovered through SOM Clustering of Seismo-Acoustic Features Compared with Camera Images and GBInSAR Data”. *Remote Sensing* 14:5, 2022. DOI: 10.3390/rs14051287.
6. A. G. Baydin et al. “Toward Machine Learning Optimization of Experimental Design”. *Nucl. Phys. News* 31:1, 2021, pp. 25–28. DOI: 10.1080/10619127.2021.1881364.
7. T. Dorigo, A. Giammanco, P. Vischia, et al. *Toward the End-to-End Optimization of Particle Physics Instruments with Differentiable Programming: a White Paper.* 2022. DOI: 10.48550/ARXIV.2203.13818.
8. J. Kieseler, G. C. Strong, F. Chiandotto, T. Dorigo, and L. Layer. “Calorimetric Measurement of Multi-TeV Muons via Deep Regression”. *The European Physical Journal C* 82:1, 2022. DOI: 10.1140/epjc/s10052-022-09993-5.
9. P. Zyla et al. “Review of Particle Physics”. *PTEP* 2020:8, 2020, p. 083C01. DOI: 10.1093/ptep/ptaa104.

10. T. Dorigo and P. de Castro. *Dealing with Nuisance Parameters using Machine Learning in High Energy Physics: a Review*. 2021. arXiv: 2007.09121 [stat.ML].

11. P. de Castro and T. Dorigo. “INFERNO: Inference-Aware Neural Optimisation”. *Computer Physics Communications* 244, 2019, pp. 170–179. DOI: https://doi.org/10.1016/j.cpc.2019.06.007.

12. CMS Collaboration. “Measurement of the $\mathrm{t\bar{t}}$ production cross section in the τ+jets channel in pp collisions at $\sqrt{s} = 7$ TeV”. *The European Physical Journal C* 73:4, 2013. DOI: 10.1140/epjc/s10052-013-2386-x.

13. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

14. Y. Dar, V. Muthukumar, and R. G. Baraniuk. *A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning*. 2021. arXiv: 2109.02355 [stat.ML].

15. D. Wolpert. “The Lack of A Priori Distinctions Between Learning Algorithms”. *Neural Computation* 8, 1996. DOI: 10.1162/neco.1996.8.7.1341.

16. P. J. Huber. “Robust Estimation of a Location Parameter”. *The Annals of Mathematical Statistics* 35:1, 1964, pp. 73–101.

17. K. Pearson. “On lines and planes of closest fit to systems of points in space”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2:11, 1901, pp. 559–572. DOI: 10.1080/14786440109462720. eprint: https://doi.org/10.1080/14786440109462720.

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

19. R. E. Schapire. “Explaining adaboost”. In: *Empirical inference*. Springer, 2013, pp. 37–52.

20. J. H. Friedman. “Greedy function approximation: a gradient boosting machine”. *Annals of statistics*, 2001, pp. 1189–1232.

21. T. Chen and C. Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: KDD ’16. ACM, New York, NY, USA, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

22. K. Hornik, M. Stinchcombe, and H. White. “Multilayer feedforward networks are universal approximators”. *Neural Networks* 2:5, 1989, pp. 359–366. DOI: https://doi.org/10.1016/0893-6080(89)90020-8.

23. D. Guest, K. Cranmer, and D. Whiteson. “Deep Learning and Its Application to LHC Physics”. *Annual Review of Nuclear and Particle Science* 68:1, 2018, pp. 161–181. DOI: 10.1146/annurev-nucl-101917-021019.

24. S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. 9:8, 1997. DOI: 10.1162/neco.1997.9.8.1735.

25. K. Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

26. M. Schuster and K. Paliwal. “Bidirectional recurrent neural networks”. *Signal Processing, IEEE Transactions on* 45, 1997, pp. 2673–2681. DOI: 10.1109/78.650093.

27. M. M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. 2021. arXiv: 2104.13478 [cs.LG].

28. B. Sanchez-Lengeling et al. “A Gentle Introduction to Graph Neural Networks”. *Distill*, 2021. https://distill.pub/2021/gnn-intro. DOI: 10.23915/distill.00033.

29. P. W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. 2018. arXiv: 1806.01261 [cs.LG].

30. A. G. Baydin et al. *Automatic differentiation in machine learning: a survey*. 2018. arXiv: 1502.05767 [cs.SC].

31. R. Brun and F. Rademakers. “ROOT — An object oriented data analysis framework”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389:1, 1997. New Computing Techniques in Physics Research V, pp. 81–86. DOI: https://doi.org/10.1016/S0168-9002(97)00048-X.

32. L. Moneta et al. “The RooStats Project”. *PoS* ACAT2010, 2010, p. 057. DOI: 10.22323/1.093.0057. arXiv: 1009.1003 [physics.data-an].

33. L. Heinrich, M. Feickert, G. Stark, and K. Cranmer. “pyhf: pure-Python implementation of HistFactory statistical models”. *Journal of Open Source Software* 6:58, 2021, p. 2823. DOI: 10.21105/joss.02823.

34. N. Reid and D. A. S. Fraser. *Likelihood inference in the presence of nuisance parameters*. 2003. DOI: 10.48550/ARXIV.PHYSICS/0312079.

35. J. Neyman. "Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability". *Philosophical Transactions of the Royal Society A* 236, 1937, pp. 333–380.

36. G. J. Feldman and R. D. Cousins. "Unified approach to the classical statistical analysis of small signals". *Physical Review D* 57:7, 1998, pp. 3873–3889. DOI: 10.1103/physrevd.57.3873.

37. S. S. Wilks. "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". *The Annals of Mathematical Statistics* 9:1, 1938, pp. 60–62. DOI: 10.1214/aoms/1177732360.

38. JetLag. *Type I Error in research: What is the alpha of a study especially when there are multiple comparisons?* Cross Validated. eprint: https://stats.stackexchange.com/q/307568.

39. A. Ly et al. *The Bayesian Methodology of Sir Harold Jeffreys as a Practical Alternative to the P-value Hypothesis Test*. 2019. DOI: 10.31234/osf.io/dhb7x.

40. K. Albertsson et al. *Machine Learning in High Energy Physics Community White Paper*. 2019. arXiv: 1807.02876 [physics.comp-ph].

41. A. Radovic et al. "Machine learning at the energy and intensity frontiers of particle physics". *Nature* 560, 2018, pp. 41–48. DOI: DOI:10.1038/s41586-018-0361-2.

42. HEP ML Community. *A Living Review of Machine Learning for Particle Physics*.

43. P. Baldi, P. Sadowski, and D. Whiteson. "Searching for exotic particles in high-energy physics with deep learning". *Nature Communications* 5:1, 2014. DOI: 10.1038/ncomms5308.

44. ATLAS Collaboration. "Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector". *Journal of High Energy Physics* 2015:4, 2015. DOI: 10.1007/jhep04(2015)117.

45. E. Bols, J. Kieseler, M. Verzetti, M. Stoye, and A. Stakia. "Jet flavour classification using DeepJet". *Journal of Instrumentation* 15:12, 2020, P12012–P12012. DOI: 10.1088/1748-0221/15/12/p12012.

46. A. J. Larkoski, I. Moult, and B. Nachman. “Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning”. *Physics Reports* 841, 2020, pp. 1–63. DOI: 10.1016/j.physrep.2019.11.001.

47. H. Qu and L. Gouskos. “Jet tagging via particle clouds”. *Physical Review D* 101:5, 2020. DOI: 10.1103/physrevd.101.056019.

48. E. Moreno, O. Cerri, J. Duarte, H. Newman, T. Nguyen, A. Periwal, M. Pierini, A. Serikova, M. Spiropulu, and J.-R. Vlimant. “JEDI-net: a jet identification algorithm based on interaction networks”, 2019.

49. T. Dorigo. *Geometry Optimization of a Muon-Electron Scattering Experiment.* 2020. arXiv: 2002.09973 [physics.ins-det].

50. P. De Castro Manzano. *Statistical Learning and Inference at Particle Collider Experiments.* http://paduaresearch.cab.unipd.it/11977/. 2019.

51. J. Thaler and K. Van Tilburg. “Maximizing boosted top identification by minimizing N-subjettiness”. *Journal of High Energy Physics* 2012:2, 2012. DOI: 10.1007/jhep02(2012)093.

52. J. Dolen et al. “Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure”. *Journal of High Energy Physics* 2016:5, 2016. DOI: 10.1007/jhep05(2016)156.

53. M. Neal Radford. “Computing Likelihood Functions for High-Energy Physics Experiments when Distributions are Defined by Simulators with Nuisance Parameters”, 2008. DOI: 10.5170/CERN-2008-001.111.

54. CDF Collaboration. “Evidence for a Particle Produced in Association with Weak Bosons and Decaying to a Bottom-Antibottom Quark Pair in Higgs Boson Searches at the Tevatron”. *Physical Review Letters* 109:7, 2012. DOI: 10.1103/physrevlett.109.071804.

55. CMS Collaboration. “Combined results of searches for the standard model Higgs boson in pp collisions at $\sqrt{s} = 7$ TeV”. *Phys. Lett. B* 710, 2012, pp. 26–48. DOI: 10.1016/j.physletb.2012.02.064. arXiv: 1202.1488 [hep-ex].

56. P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson. “Parameterized neural networks for high-energy physics”. *The European Physical Journal C* 76:5, 2016. DOI: 10.1140/epjc/s10052-016-4099-4.

57. J. A. Aguilar-Saavedra, J. Collins, and R. K. Mishra. “A generic anti-QCD jet tagger”. *Journal of High Energy Physics* 2017:11, 2017. DOI: 10.1007/jhep11(2017)163.

58. S. Chang, T. Cohen, and B. Ostdiek. "What is the machine learning?" *Phys. Rev. D* 97, 5 2018, p. 056009. DOI: 10.1103/PhysRevD.97.056009.

59. J. Stevens and M. Williams. "uBoost: a boosting method for producing uniform selection efficiencies from multivariate classifiers". *Journal of Instrumentation* 8:12, 2013, P12013–P12013. DOI: 10.1088/1748-0221/8/12/p12013.

60. A. Rogozhnikov, A. Bukva, V. Gligorov, A. Ustyuzhanin, and M. Williams. "New approaches for boosting to uniformity". *Journal of Instrumentation* 10:03, 2015, T03002–T03002. DOI: 10.1088/1748-0221/10/03/t03002.

61. G. Kasieczka and D. Shih. "Robust Jet Classifiers through Distance Correlation". *Physical Review Letters* 125:12, 2020. DOI: 10.1103/physrevlett.125.122001.

62. S. Wunsch, S. Jörger, R. Wolf, and G. Quast. "Reducing the Dependence of the Neural Network Function to Systematic Uncertainties in the Input Space". *Computing and Software for Big Science* 4:1, 2020. DOI: 10.1007/s41781-020-00037-9.

63. G. Louppe, M. Kagan, and K. Cranmer. *Learning to Pivot with Adversarial Networks*. 2017. arXiv: 1611.01046 [stat.ML].

64. C. Shimmin et al. "Decorrelated jet substructure tagging using adversarial neural networks". *Physical Review D* 96:7, 2017. DOI: 10.1103/physrevd.96.074034.

65. V. Estrade, C. Germain, I. Guyon, and D. Rousseau. "Adversarial learning to eliminate systematic errors: a case study in High Energy Physics". In: *NIPS 2017*. 2017.

66. C. Adam-Bourdarios, G. Cowan, et al. "The Higgs Machine Learning Challenge". *Journal of Physics: Conference Series* 664:7, 2015, p. 072015. DOI: 10.1088/1742-6596/664/7/072015.

67. A. Blance, M. Spannowsky, and P. Waite. "Adversarially-trained autoencoders for robust unsupervised new physics searches". *Journal of High Energy Physics* 2019:10, 2019. DOI: 10.1007/jhep10(2019)047.

68. C. Englert, P. Galler, P. Harris, and M. Spannowsky. "Machine learning uncertainties with adversarial neural networks". *The European Physical Journal C* 79:1, 2019. DOI: 10.1140/epjc/s10052-018-6511-8.

69. L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman. “Weakly supervised classification in high energy physics”. *Journal of High Energy Physics* 2017:5, 2017. DOI: 10.1007/jhep05(2017)145.

70. E. M. Metodiev, B. Nachman, and J. Thaler. “Classification without labels: learning from mixed samples in high energy physics”. *Journal of High Energy Physics* 2017:10, 2017. DOI: 10.1007/jhep10(2017)174.

71. T. Cohen, M. Freytsis, and B. Ostdiek. “(Machine) learning to do more with less”. *Journal of High Energy Physics* 2018:2, 2018. DOI: 10.1007/jhep02(2018)034.

72. P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz. “Learning to classify from impure samples with high-dimensional data”. *Phys. Rev. D* 98, 1 2018, p. 011502. DOI: 10.1103/PhysRevD.98.011502.

73. J. Y. Araz and M. Spannowsky. “Combine and conquer: event reconstruction with Bayesian Ensemble Neural Networks”. *Journal of High Energy Physics* 2021:4, 2021. DOI: 10.1007/jhep04(2021)296.

74. S. Bollweg, M. Haussmann, G. Kasieczka, M. Luchmann, T. Plehn, and J. Thompson. “Deep-learning jets with uncertainties and more”. *SciPost Physics* 8:1, 2020. DOI: 10.21468/scipostphys.8.1.006.

75. G. Kasieczka, M. Luchmann, F. Otterpohl, and T. Plehn. “Per-object systematics using deep-learned calibration”. *SciPost Physics* 9:6, 2020. DOI: 10.21468/scipostphys.9.6.089.

76. K. Cranmer, J. Brehmer, and G. Louppe. *The frontier of simulation-based inference*. 2020. arXiv: 1911.01429 [stat.ML].

77. J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez. “Constraining Effective Field Theories with Machine Learning”. *Physical Review Letters* 121:11, 2018. DOI: 10.1103/physrevlett.121.111801.

78. K. Cranmer, J. Pavez, and G. Louppe. *Approximating Likelihood Ratios with Calibrated Discriminative Classifiers*. 2016. arXiv: 1506.02169 [stat.AP].

79. J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer. “Mining gold from implicit models to improve likelihood-free inference”. *Proceedings of the National Academy of Sciences* 117:10, 2020, pp. 5242–5249. DOI: 10.1073/pnas.1915980117.

80. J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez. "A guide to constraining effective field theories with machine learning". *Physical Review D* 98:5, 2018. DOI: 10.1103/physrevd.98.052004.

81. M. Stoye, J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer. *Likelihood-free inference with an improved cross-entropy estimator*. 2018. arXiv: 1808.00973 [stat.ML].

82. J. Brehmer, F. Kling, I. Espejo, and K. Cranmer. *MadMiner: Machine learning-based inference for particle physics*. 2020. arXiv: 1907.10621 [hep-ph].

83. M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

84. G. Cowan, K. Cranmer, E. Gross, and O. Vitells. "Asymptotic formulae for likelihood-based tests of new physics". *The European Physical Journal C* 71:2, 2011. DOI: 10.1140/epjc/s10052-011-1554-0.

85. T. Charnock, G. Lavaux, and B. D. Wandelt. "Automatic physical inference with information maximizing neural networks". *Phys. Rev. D* 97, 8 2018, p. 083004. DOI: 10.1103/PhysRevD.97.083004.

86. J. Alsing and B. Wandelt. "Nuisance hardened data compression for fast likelihood-free inference". *Monthly Notices of the Royal Astronomical Society* 488, 2019, pp. 5093–5103. DOI: 10.1093/mnras/stz1900.

87. L.-G. Xia. "QBDT, a new boosting decision tree method with systematical uncertainties into training for High Energy Physics". *Nuclear Instruments and Methods in Physics Research Section A* 930, 2019, pp. 15–26. DOI: 10.1016/j.nima.2019.03.088.

88. A. Elwood and D. Krücker. *Direct optimisation of the discovery significance when training neural networks to search for new physics in particle colliders*. 2018. arXiv: 1806.00322 [hep-ex].

89. P. Feichtinger et al. *Punzi-loss: A non-differentiable metric approximation for sensitivity optimisation in the search for new particles*. 2021. arXiv: 2110.00810 [hep-ex].

90. G. Grosso et al. *An Imperfect machine to search for New Physics: systematic uncertainties in a machine-learning based signal extraction*. NeurIPS 2021.

91. A. Ghosh, B. Nachman, and D. Whiteson. "Uncertainty-aware machine learning for high energy physics". *Physical Review D* 104:5, 2021. DOI: 10.1103/physrevd.104.056026.

92. S. Wunsch, S. Jörger, R. Wolf, and G. Quast. “Optimal Statistical Inference in the Presence of Systematic Uncertainties Using Neural Network Optimization Based on Binned Poisson Likelihoods with Nuisance Parameters”. *Computing and Software for Big Science* 5:1, 2021. DOI: 10.1007/s41781-020-00049-5.

93. N. Simpson and L. Heinrich. “neos: End-to-End-Optimised Summary Statistics for High Energy Physics”, 2022. DOI: 10.48550/arXiv.2203.05570. eprint: arXiv:2203.05570.

94. N. Simpson and L. Heinrich. *neos: version 0.2.0.* Version v0.2.0. 2021. DOI: 10.5281/zenodo.6351423.

95. G. Strong. *GilesStrong/pytorch_inferno: v0.2.2.* Version v0.2.2. 2021. DOI: 10.5281/zenodo.5040810.

96. S. Weinberg. “The Making of the standard model”. *The European Physical Journal C-Particles and Fields* 34:1, 2004, pp. 5–13.

97. F. J. Hasert et al. “Observation of Neutrino Like Interactions without Muon or Electron in the Gargamelle Neutrino Experiment”. *Nucl. Phys. B* 73, 1974, pp. 1–22. DOI: 10.1016/0550-3213(74)90038-8.

98. ATLAS Collaboration. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. *Physics Letters B* 716:1, 2012, pp. 1–29.

99. CMS Collaboration. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”. *Physics Letters B* 716:1, 2012, pp. 30–61.

100. M. Srednicki. *Quantum field theory.* Cambridge University Press, 2007.

101. M. Thomson. *Modern particle physics.* Cambridge University Press, New York, 2013.

102. J. Ellis. “Higgs Physics”, 2013, 117–168. 52 p. DOI: 10.5170/CERN-2015-004.117. arXiv: 1312.5672.

103. UA1 Collaboration. “Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at s**(1/2) = 540-GeV”. *Phys. Lett.* B122, 1983. [,611(1983)], pp. 103–116. DOI: 10.1016/0370-2693(83)91177-2.

104. UA2 Collaboration. “Observation of Single Isolated Electrons of High Transverse Momentum in Events with Missing Transverse Energy at the CERN anti-p p Collider”. *Phys. Lett.* B122, 1983. [,7.45(1983)], pp. 476–485. DOI: 10.1016/0370-2693(83)91605-2.

105. UA1 Collaboration. “Experimental Observation of Lepton Pairs of Invariant Mass Around 95-GeV/c**2 at the CERN SPS Collider”. *Phys. Lett.* B126, 1983. [,7.55(1983)], pp. 398–410. DOI: 10.1016/0370-2693(83)90188-0.

106. UA2 Collaboration. “Evidence for $Z^0 \to e^+e^-$ at the CERN $\bar{p}p$ Collider”. *Phys. Lett. B* 129, 1983, pp. 130–140. DOI: 10.1016/0370-2693(83)90744-X.

107. E. Fermi. “Tentativo di una Teoria Dei Raggi β”. *Il Nuovo Cimento* 11:1, 1934, pp. 1–19. DOI: 10.1007/BF02959820.

108. C. S. Wu, E. Ambler, R. W. Hayward, D. D. Hoppes, and R. P. Hudson. “Experimental Test of Parity Conservation in Beta Decay”. *Phys. Rev.* 105, 1957, pp. 1413–1414. DOI: 10.1103/PhysRev.105.1413.

109. J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. “Evidence for the 2π Decay of the K_2^0 Meson”. *Phys. Rev. Lett.* 13, 4 1964, pp. 138–140. DOI: 10.1103/PhysRevLett.13.138.

110. C. Rovelli. “Loop quantum gravity”. *Living reviews in relativity* 11:1, 2008, p. 5.

111. J. Polchinski. *String Theory*. Cambridge monographs on mathematical physics. Cambridge Univ. Press, Cambridge, 1998.

112. E. Corbelli and P. Salucci. “The extended rotation curve and the dark matter halo of M33”. *Monthly Notices of the Royal Astronomical Society* 311:2, 2000, pp. 441–447.

113. V. Trimble. “Existence and nature of dark matter in the universe”. *Annual review of astronomy and astrophysics* 25:1, 1987, pp. 425–472.

114. P. A. R. Ade et al. “Planck 2015 results. XIII. Cosmological parameters”. *Astron. Astrophys.* 594, 2016, A13. DOI: 10.1051/0004-6361/201525830. arXiv: 1502.01589 [astro-ph.CO].

115. A. G. Riess et al. “Type Ia supernova discoveries at $z > 1$ from the Hubble Space Telescope: Evidence for past deceleration and constraints on dark energy evolution”. *The Astrophysical Journal* 607:2, 2004, p. 665.

116. Y. Fukuda et al. “Evidence for oscillation of atmospheric neutrinos”. *Physical Review Letters* 81:8, 1998, p. 1562.

117. Q. R. Ahmad et al. “Measurement of the rate of $\nu_e + d \to p + p + e^-$ interactions produced by ^{8}B solar neutrinos at the Sudbury Neutrino Observatory”. *Phys. Rev. Lett.* 87, 2001, p. 071301. DOI: 10.1103/PhysRevLett.87.071301. arXiv: nucl-ex/0106015.

118. O. Rind et al. “Precision measurement of muon g-2 at BNL”. *eConf* C010430, 2001. Ed. by D. Bettoni, p. M05. arXiv: hep-ex/0106101.

119. B. Abi et al. “Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm”. *Phys. Rev. Lett.* 126, 14 2021, p. 141801. DOI: 10.1103/PhysRevLett.126.141801.

120. LHCb Collaboration. “Measurement of Form-Factor-Independent Observables in the Decay $B^0 \to K^{*0}\mu^+\mu^-$”. *Phys. Review Letters* 111:19, 2013. DOI: 10.1103/physrevlett.111.191801.

121. LHCb Collaboration. “Measurement of CP-Averaged Observables in the $B^0 \to K^{*0}\mu^+\mu^-$ Decay”. *Phys. Rev. Lett.* 125:1, 2020. DOI: 10.1103/physrevlett.125.011802.

122. E. Halkiadakis, G. Redlinger, and D. Shih. “Status and Implications of Beyond-the-Standard-Model Searches at the LHC”. *Annual Review of Nuclear and Particle Science* 64:1, 2014, pp. 319–342. DOI: 10.1146/annurev-nucl-102313-025632. eprint: https://doi.org/10.1146/annurev-nucl-102313-025632.

123. J. M. Butterworth, G. Dissertori, and G. P. Salam. “Hard Processes in Proton-Proton Collisions at the Large Hadron Collider”. *Annual Review of Nuclear and Particle Science* 62:1, 2012, pp. 387–405. DOI: 10.1146/annurev-nucl-102711-094913.

124. R. D. Ball et al. “Parton distributions from high-precision collider data”. *Eur. Phys. J.* C77:10, 2017, p. 663. DOI: 10.1140/epjc/s10052-017-5199-5. arXiv: 1706.00428 [hep-ph].

125. S. Alekhin et al. *The PDF4LHC Working Group Interim Report.* 2011. arXiv: 1101.0536 [hep-ph].

126. G. Altarelli and G. Parisi. “Asymptotic Freedom in Parton Language”. *Nucl. Phys.* B126, 1977, pp. 298–318. DOI: 10.1016/0550-3213(77)90384-4.

127. Y. L. Dokshitzer. “Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics.” *Sov. Phys. JETP* 46, 1977. [Zh. Eksp. Teor. Fiz.73,1216(1977)], pp. 641–653.

128. V. N. Gribov and L. N. Lipatov. “Deep inelastic e p scattering in perturbation theory”. *Sov. J. Nucl. Phys.* 15, 1972. [Yad. Fiz.15,781(1972)], pp. 438–450.

129. H.-L. Lai et al. “New parton distributions for collider physics”. *Physical Review D* 82:7, 2010. DOI: 10.1103/physrevd.82.074024.

130. A. D. Martin, W. J. Stirling, R. S. Thorne, and G. Watt. “Parton distributions for the LHC”. *The European Physical Journal C* 63:2, 2009, pp. 189–285. DOI: 10.1140/epjc/s10052-009-1072-5.

131. NNPDF Collaboration, R. D. Ball, V. Bertone, F. Cerutti, L. D. Debbio, S. Forte, A. Guffanti, J. I. Latorre, J. Rojo, and M. Ubiali. *Unbiased global determination of parton distributions and their uncertainties at NNLO and at LO*. 2011. arXiv: 1107.2652 [hep-ph].

132. F. Bechtel. “The Underlying Event in Proton-Proton Collisions”. 2009.

133. T. Sjöstrand et al. “An Introduction to PYTHIA 8.2”. *Comput. Phys. Commun.* 191, 2015, pp. 159–177. DOI: 10.1016/j.cpc.2015.01.024. arXiv: 1410.3012 [hep-ph].

134. M. Bähr et al. “Herwig++ physics and manual”. *The European Physical Journal C* 58:4, 2008, pp. 639–707. DOI: 10.1140/epjc/s10052-008-0798-9.

135. T. Gleisberg et al. “Event generation with SHERPA 1.1”. *Journal of High Energy Physics* 2009:02, 2009, pp. 007–007. DOI: 10.1088/1126-6708/2009/02/007.

136. J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer. “MadGraph 5: going beyond”. *Journal of High Energy Physics* 2011:6, 2011. DOI: 10.1007/jhep06(2011)128.

137. S. Frixione and B. R. Webber. “Matching NLO QCD computations and parton shower simulations”. *Journal of High Energy Physics* 2002:06, 2002, pp. 029–029. DOI: 10.1088/1126-6708/2002/06/029.

138. P. Nason. “A New Method for Combining NLO QCD with Shower Monte Carlo Algorithms”. *Journal of High Energy Physics* 2004:11, 2004, pp. 040–040. DOI: 10.1088/1126-6708/2004/11/040.

139. CMS Collaboration. "Measurement of the Inclusive Jet Cross Section in pp Collisions at $\sqrt{s} = 7$ TeV". *Physical Review Letters* 107:13, 2011. DOI: 10.1103/physrevlett.107.132001.

140. CDF Collaboration. "Observation of Top Quark Production in $p\bar{p}$ Collisions with the Collider Detector at Fermilab". *Physical Review Letters* 74:14, 1995, pp. 2626–2631. DOI: 10.1103/physrevlett.74.2626.

141. D0 Collaboration. "Observation of the Top Quark". *Physical Review Letters* 74:14, 1995, pp. 2632–2637. DOI: 10.1103/physrevlett.74.2632.

142. S. Alekhin, A. Djouadi, and S. Moch. "The top quark and Higgs boson masses and the stability of the electroweak vacuum". *Physics Letters B* 716:1, 2012, pp. 214–219. DOI: 10.1016/j.physletb.2012.08.024.

143. G. Cortiana. "Top-quark mass measurements: Review and perspectives". *Reviews in Physics* 1, 2016, pp. 60–76. DOI: https://doi.org/10.1016/j.revip.2016.04.001.

144. G. Degrassi et al. "Higgs mass and vacuum stability in the Standard Model at NNLO". *Journal of High Energy Physics* 2012:8, 2012. DOI: 10.1007/jhep08(2012)098.

145. M. Cristinziani and M. Mulders. "Top-quark physics at the Large Hadron Collider". *Journal of Physics G: Nuclear and Particle Physics* 44:6, 2017, p. 063001. DOI: 10.1088/1361-6471/44/6/063001.

146. CMS Collaboration. *Measurement of differential cross sections for the production of top quark pairs and of additional jets in pp collisions at $\sqrt{s} = 13$ TeV.* Technical report. Geneva: CERN, 2022.

147. A. H. Hoang. "What Is the Top Quark Mass?" *Annual Review of Nuclear and Particle Science* 70:1, 2020, pp. 225–255. DOI: 10.1146/annurev-nucl-101918-023530.

148. ATLAS and CMS Collaboration. "Combinations of single-top-quark production cross-section measurements and $|f_L V_{tb}|$ determinations at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS experiments". *Journal of High Energy Physics* 2019:5, 2019. DOI: 10.1007/jhep05(2019)088.

149. CMS Collaboration. "Search for new physics in top quark production with additional leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV using effective field theory". *Journal of High Energy Physics* 2021:3, 2021. DOI: 10.1007/jhep03(2021)095.

150. E. Mobs. “The CERN accelerator complex - 2019. Complexe des accélérateurs du CERN - 2019”, 2019. General Photo.

151. O. Bruning et al. “LHC Design Report Vol.1: The LHC Main Ring”, 2004. DOI: 10.5170/CERN-2004-003-V-1.

152. J. Wenninger. “The LHC collider”. *Comptes Rendus Physique* 16, 2015, pp. 347–355. DOI: 10.1016/j.crhy.2015.03.005.

153. J. Vollaire et al. *Linac4 design report*. Ed. by M. Vretenar. Vol. 6/2020. CERN Yellow Reports: Monographs. CERN, Geneva, 2020. DOI: 10.23731/CYRM-2020-006.

154. ALICE Collaboration. “The ALICE experiment at the CERN LHC”. *JINST* 3, 2008, S08002. DOI: 10.1088/1748-0221/3/08/S08002.

155. ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. *JINST* 3, 2008, S08003. DOI: 10.1088/1748-0221/3/08/S08003.

156. CMS Collaboration. “The CMS Experiment at the CERN LHC”. *JINST* 3, 2008, S08004. DOI: 10.1088/1748-0221/3/08/S08004.

157. LHCb Collaboration. “The LHCb Detector at the LHC”. *JINST* 3, 2008, S08005. DOI: 10.1088/1748-0221/3/08/S08005.

158. TOTEM Collaboration. “The TOTEM experiment at the CERN Large Hadron Collider”. *JINST* 3, 2008, S08007. DOI: 10.1088/1748-0221/3/08/S08007.

159. LHCf Collaboration. “The LHCf detector at the CERN Large Hadron Collider”. *JINST* 3, 2008, S08006. DOI: 10.1088/1748-0221/3/08/S08006.

160. MoEDAL Collaboration. “The Physics Programme Of The MoEDAL Experiment At The LHC”. *Int. J. Mod. Phys.* A29, 2014, p. 1430050. DOI: 10.1142/S0217751X14300506. arXiv: 1405.7662 [hep-ph].

161. FASER Collaboration. *Technical Proposal for FASER: ForwArd Search ExpeRiment at the LHC*. 2018. arXiv: 1812.09139 [physics.ins-det].

162. T. Sakuma et al. “Detector and Event Visualization with SketchUp at the CMS Experiment”. 513:2, 2014, p. 022032. DOI: 10.1088/1742-6596/513/2/022032.

163. CMS Collaboration. *The CMS magnet project: Technical Design Report*. Technical design report. CMS. CERN, Geneva, 1997.

164. CMS Collaboration. “Description and performance of track and primary-vertex reconstruction with the CMS tracker”, 2014. DOI: 10.5167/uzh-102519.

165. V. Veszpremi. “Operation and performance of the CMS tracker”. *Journal of Instrumentation* 9:03, 2014, pp. C03005–C03005. DOI: 10.1088/1748-0221/9/03/c03005.

166. CMS Collaboration. *The CMS electromagnetic calorimeter project: Technical Design Report.* Technical design report. CMS. CERN, Geneva, 1997.

167. CMS Collaboration. “Energy Calibration and Resolution of the CMS Electromagnetic Calorimeter in pp Collisions at $\sqrt{s} = 7$ TeV”. *JINST* 8, 2013, P09009. DOI: 10.1088/1748-0221/8/09/P09009. arXiv: 1306.2016 [hep-ex].

168. CMS Collaboration. *The CMS hadron calorimeter project: Technical Design Report.* Technical design report. CMS. CERN, Geneva, 1997.

169. CMS Collaboration. “Performance of the CMS hadron calorimeter with cosmic ray muons and LHC beam data”. *Journal of Instrumentation* 5:03, 2010, T03012–T03012. DOI: 10.1088/1748-0221/5/03/t03012.

170. CMS Collaboration. *The CMS Barrel Calorimeter Response to Particle Beams from 2 to 350 GeV/c.* Technical report. Geneva: CERN, 2008.

171. CMS Collaboration. “Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV”. *JINST* 13:06, 2018, P06015. DOI: 10.1088/1748-0221/13/06/P06015. arXiv: 1804.04528 [physics.ins-det].

172. CMS Collaboration. “Performance of the reconstruction and identification of high-momentum muons in proton-proton collisions at $\sqrt{s} = 13$ TeV”. *Journal of Instrumentation* 15:02, 2020, P02027–P02027. DOI: 10.1088/1748-0221/15/02/p02027.

173. CMS Collaboration. “The CMS trigger system”. *JINST* 12:01, 2017, P01020. DOI: 10.1088/1748-0221/12/01/P01020. arXiv: 1609.02366.

174. S. Mukherjee. “Data Scouting and Data Parking with the CMS High level Trigger”. *PoS* EPS-HEP2019, 2020, 139. 6 p. DOI: 10.22323/1.364.0139.

175. CMS Collaboration. “Particle-flow reconstruction and global event description with the CMS detector”. *Journal of Instrumentation* 12:10, 2017, P10003–P10003. DOI: 10.1088/1748-0221/12/10/p10003.

176. CMS Collaboration. “Pileup mitigation at CMS in 13 TeV data”. *Journal of Instrumentation* 15:09, 2020, P09018–P09018. DOI: 10.1088/1748-0221/15/09/p09018.

177. R. Fruhwirth, W. Waltenberger, and P. Vanlaer. “Adaptive vertex fitting”. *J. Phys.* G34, 2007, N343. DOI: 10.1088/0954-3899/34/12/N01.

178. S. Amrouche et al. “The Tracking Machine Learning Challenge: Accuracy Phase”. In: *The NeurIPS ’18 Competition*. Springer International Publishing, 2019, pp. 231–264. DOI: 10.1007/978-3-030-29135-8_9.

179. S. Amrouche et al. *The Tracking Machine Learning challenge : Throughput phase*. 2021. DOI: 10.48550/ARXIV.2105.01160.

180. J. Pata et al. *Machine Learning for Particle Flow Reconstruction at CMS*. Technical report. Presented at the ACAT 2021. 2022. arXiv: 2203.00330.

181. M. Cacciari, G. P. Salam, and G. Soyez. “The anti-k_t jet clustering algorithm”. *JHEP* 04, 2008, p. 063. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189 [hep-ph].

182. CMS Collaboration. “Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV”. *Journal of Instrumentation* 12:02, 2017, P02014–P02014. DOI: 10.1088/1748-0221/12/02/p02014.

183. CMS Collaboration. “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV”. *Journal of Instrumentation* 13:05, 2018, P05011–P05011. DOI: 10.1088/1748-0221/13/05/p05011.

184. CMS Collaboration. “Identification of b-quark jets with the CMS experiment”. 8:04, 2013, P04013–P04013. DOI: 10.1088/1748-0221/8/04/p04013.

185. E. Bols, J. Kieseler, M. Verzetti, M. Stoye, and A. Stakia. “Jet flavour classification using DeepJet”. *Journal of Instrumentation* 15:12, 2020, P12012–P12012. DOI: 10.1088/1748-0221/15/12/p12012.

186. CMS Collaboration. “Reconstruction and identification of τ lepton decays to hadrons and ν_τ at CMS”. 11:01, 2016, P01019–P01019. DOI: 10.1088/1748-0221/11/01/p01019.

187. CMS Collaboration. “Performance of the DeepTau algorithm for the discrimination of taus against jets, electron, and muons”, 2019.

188. V. Daniel Elvira. “Impact of detector simulation in particle physics collider experiments”. *Physics Reports* 695, 2017, pp. 1–54. DOI: 10.1016/j.physrep.2017.06.002.

189. D. Lange, M. Hildreth, V. Ivantchenko, and I. Osborne. “Upgrades for the CMS simulation”. *Journal of Physics: Conference Series* 608, 2015, p. 012056. DOI: 10.1088/1742-6596/608/1/012056.

190. M. Hildreth, V. Ivanchenko, and D. Lange. “Upgrades for the CMS simulation”. *Journal of Physics: Conference Series* 898, 2017, p. 042040. DOI: 10.1088/1742-6596/898/4/042040.

191. K. Pedro. “Current and Future Performance of the CMS Simulation”. *EPJ Web of Conferences* 214, 2019. DOI: 10.1051/epjconf/201921402036.

192. CMS Collaboration. *Fast Simulation of the CMS Detector at the LHC*. Technical report. Geneva: CERN, 2010.

193. S. Sekmen. *Recent Developments in CMS Fast Simulation*. 2017. arXiv: 1701.03850 [physics.ins-det].

194. A. Butter and T. Plehn. *Generative Networks for LHC events*. 2020. DOI: 10.48550/ARXIV.2008.08558.

195. C. Ferro. *Measurement of the tt production cross section in the tau+jets channel in pp collisions at $\sqrt{s} = 7$ TeV*. https://tel.archives-ouvertes.fr/tel-00862736. 2012.

196. K. Lassila-Perini, C. Lange, E. Carrera Jarrin, and M. Bellis. “Using CMS Open Data in research – challenges and directions”. *EPJ Web of Conferences* 251, 2021, p. 01004. DOI: 10.1051/epjconf/202125101004.

197. CMS Collaboration. *CMSSW Software*. http://cms-sw.github.io/(2021). Accessed: 2022-01-30.

198. K. Ehatäht. “NANOAOD: a new compact event data format in CMS”. *EPJ Web Conf.* 245, 2020, p. 06002. DOI: 10.1051/epjconf/202024506002.

199. J. Pivarski, P. Elmer, and D. Lange. “Awkward Arrays in Python, C++, and Numba”. *EPJ Web of Conferences* 245, 2020, p. 05023. DOI: 10.1051/epjconf/202024505023.

200. N. Smith, L. Gray, et al. “Coffea Columnar Object Framework For Effective Analysis”. *EPJ Web of Conferences* 245, 2020, p. 06012. DOI: 10.1051/epjconf/202024506012.

201. W. McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

202. P. Adam et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

203. J. D. Hunter. “Matplotlib: A 2D graphics environment”. *Computing in Science & Engineering* 9:3, 2007, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

204. K. Cranmer and A. Held. “Building and steering binned template fits with cabinetry”. *EPJ Web Conf.* 251, 2021, p. 03067. DOI: 10.1051/epjconf/202125103067.

205. The ATLAS Collaboration, The CMS Collaboration, The LHC Higgs Combination Group. *Procedure for the LHC Higgs boson search combination in Summer 2011*. Technical report. Geneva: CERN, 2011.

206. CMS Collaboration. *MultiJet primary dataset in AOD format from RunA of 2011 (/MultiJet/Run2011A-12Oct2013-v1/AOD). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.8N95.GCTN.

207. CMS Collaboration. *MultiJet primary dataset in AOD format from RunB of 2011 (/MultiJet/Run2011B-12Oct2013-v1/AOD). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.DHUA.BHL2.

208. CMS Collaboration. *Sim. dataset TTJets_TuneZ2_7TeV-madgraph-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.ZBGF.H543.

209. CMS Collaboration. *Simul. dataset WJetsToLNu_TuneZ2_7TeV-madgraph-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.U7P6.CKVB.

210. CMS Collaboration. *Simulated dataset DYJetsToLL_TuneZ2_M-50_7TeV-madgraph-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.TXT4.4RRP.

211. CMS Collaboration. *Simulated dataset T_TuneZ2_s-channel_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.CYDJ.SRGR.

212. CMS Collaboration. *Sim. dataset T_TuneZ2_tW-channel-DR_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.AYNJ.DVM3.

213. CMS Collaboration. *Simulated dataset T_TuneZ2_t-channel_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.TZH7.EFE7.

214. CMS Collaboration. *Simul. dataset Tbar_TuneZ2_s-channel_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.HB9Y.8B4H.

215. CMS Collaboration. *Sim. data Tbar_TuneZ2_tW-channel-DR_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.BN7K.8N4A.

216. CMS Collaboration. *Simul. dataset Tbar_TuneZ2_t-channel_7TeV-powheg-tauola in AODSIM format for 2011 collision data (SM Inclusive). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.ZCNM.27A3.

217. N. Kidonakis. “Next-to-next-to-leading soft-gluon corrections for the top quark cross section and transverse momentum distribution”. *Physical Review D* 82:11, 2010. DOI: 10.1103/physrevd.82.114030.

218. R. Gavin, Y. Li, F. Petriello, and S. Quackenbush. “FEWZ 2.0: A code for hadronic Z production at next-to-next-to-leading order”. *Computer Physics Communications* 182:11, 2011, pp. 2388–2403. DOI: 10.1016/j.cpc.2011.06.008.

219. N. Davidson et al. “Universal interface of TAUOLA: Technical and physics documentation”. *Computer Physics Communications* 183:3, 2012, pp. 821–843. DOI: 10.1016/j.cpc.2011.12.009.

220. S. Alioli, P. Nason, C. Oleari, and E. Re. “A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX”. *Journal of High Energy Physics* 2010:6, 2010. DOI: 10.1007/jhep06(2010)043.

221. CMS Collaboration. *SingleMu primary dataset in AOD format from RunA of 2011 (/SingleMu/Run2011A-12Oct2013-v1/AOD). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.UY8U.9XJ3.

222. CMS Collaboration. *SingleMu primary dataset in AOD format from RunB of 2011 (/SingleMu/Run2011B-12Oct2013-v1/AOD). CERN Open Data Portal.* DOI: 10.7483/OPENDATA.CMS.XBTD.NKD3.

223. CMS Collaboration. “Jet Energy Corrections determination at 7 TeV”, 2010.

224. P. M. Nadolsky et al. “Implications of CTEQ global analysis for collider observables”. *Phys. Rev. D* 78, 2008, p. 013004. DOI: 10.1103/PhysRevD.78.013004. arXiv: 0802.0007 [hep-ph].

225. CMS Collaboration. *Absolute Calibration of the Luminosity Measurement at CMS: Winter 2012 Update*. Geneva, 2012.

226. CMS Collaboration. “Performance of tau-lepton reconstruction and identification in CMS”. *Journal of Instrumentation* 7:01, 2012, P01001–P01001. DOI: 10.1088/1748-0221/7/01/p01001.

227. CMS Collaboration. *Tau identification in CMS*. CMS-PAS-TAU-11-001. 2011.

228. CMS Collaboration. *CMS Pile-up simulation*. http://opendata.cern.ch/docs/cms-guide-pileup-simulation. Accessed: 2022-01-30.

229. S. M. Lundberg and S.-I. Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4765–4774.

230. J. Therhaag. “TMVA Toolkit for multivariate data analysis in ROOT”. *PoS* ICHEP2010, 2010, p. 510. DOI: 10.22323/1.120.0510.

231. F. James and M. Roos. “Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations”. *Comput. Phys. Commun.* 10, 1975, pp. 343–367. DOI: 10.1016/0010-4655(75)90039-9.

232. H. Dembinski and P. O. et al. “scikit-hep/iminuit”, 2020. DOI: 10.5281/zenodo.3949207.

233. B. A. Murtagh and M. A. Saunders. “Large-scale linearly constrained optimization”. *Mathematical Programming* 14:1, 1978, pp. 41–72. DOI: 10.1007/BF01588950.

234. CMS Collaboration. *HiggsAnalysis-CombinedLimit*. https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit. Accessed: 2022-04-07.

235. J. S. Conway. *Incorporating Nuisance Parameters in Likelihoods for Multisource Spectra*. 2011. arXiv: 1103.0354 [physics.data-an].

236. User Tony-Y. *Differentiable torch.histc*. https://discuss.pytorch.org/t/differentiable-torch-histc/25865. Accessed: 2022-01-30.