Modeling charged-particle spectra of proton-proton collisions in ALICE using deep neural networks

Maria Alejandra
Calmon Behling

Bachelor's Thesis

September 2023

Institut für Kernphysik
Fachbereich Physik
Goethe-Universität Frankfurt am Main

Supervisor: Prof. Dr. Henner Büsching
Second examiner: Prof. Dr. Harald Appelshäuser

Contents

Chapter 1

Motivation

High-energy proton-proton (pp) collisions provide an excellent opportunity to investigate *quantum chromodynamics* (QCD), the theory of the strong interaction, in the laboratory. Experiments like ALICE at the Large Hadron Collider (LHC) measure the particles emerging from these collisions. The measured particle abundancies can be connected to the underlying QCD processes via phenomenological models. Those are implemented in event generators like PYTHIA [1], which intend to model the whole evolution of a high-energy collision and rely on free parameters that are "tuned" to best describe experimental data. The current most prominent and established high-energy tune of PYTHIA was adjusted to best reproduce LHC Run 1 measurements of pp collisions at $\sqrt{s} = 7$ TeV [2].

Recently, ALICE published a comprehensive dataset of inclusive charged-particle multiplicity (N_{ch}) distributions and transverse momentum (p_{T}) spectra for pp collisions at five different center-of-mass energies ranging from $\sqrt{s} = 2.76$ TeV up to 13 TeV [3]. It was shown that PYTHIA's description of these fundamental observables deteriorates for collision energies further away from the tuning energy, which raises the question of how accurately it projects particle production to regimes beyond the LHC energies.

This thesis proposes an alternative approach to predict the previously mentioned fundamental observables (N_{ch} distributions and p_{T} spectra) at unmeasured energies by parameterizing ALICE data with two *deep neural networks* (DNNs). As opposed to PYTHIA, this approach gives no direct access to a physics interpretation of the measurements as it makes no assumptions about the underlying QCD processes leading to particle production. Instead, the neural networks learn a functional representation of the collision-energy dependence and make "data-driven" predictions for the N_{ch} distributions and p_{T} spectra. This allows for interpolating and extrapolating these observables beyond the discrete LHC collision energies. Energy interpolations of p_{T} spectra can be used for constructing a pp reference for the *nuclear modification factor* (R_{AA}). This quantifies how the presence of *quark-gluon plasma* (QGP), an extreme state of

matter created in central heavy-ion collisions, affects particle production. In 2017, Xe–Xe collisions were measured at the LHC with an energy of $\sqrt{s_{\mathrm{NN}}} = 5.44$ TeV, but no corresponding pp reference measurement was recorded. In a previous publication, a power-law interpolation was performed to provide a pp reference for the calculation of the R_{AA} [4]. Training a *deep neural network* on p_{T} spectra at different energies, as proposed in this thesis, offers an alternative, purely data-driven interpolation method with no assumption regarding the functional shape. In this thesis, the interpolation by the DNN model is compared to the power-law interpolation published in [4]. Furthermore, the extrapolation potential of the DNNs can be used to estimate the spectra at energies beyond those currently available at the LHC. Comparisons to measurements at the Relativistic Heavy-Ion Collider ($\sqrt{s}_{\mathrm{max}} = 200$ GeV) or potential future high-energy accelerator facilities like the planned High-Energy LHC ($\sqrt{s}_{\mathrm{max}} = 27$ TeV) or even the Future Circular Collider ($\sqrt{s}_{\mathrm{max}} = 100$ TeV), could help to quantify the differences in particle production at these energies. In addition, the energy dependence of the average number of produced particles predicted by the DNN models is compared to an empirically observed power-law scaling. PYTHIA allows for simulating collisions at any given energy, providing the opportunity to indirectly assess the predictive power of the ALICE-trained DNN models beyond the LHC energies. For this purpose, two further DNN models are trained on PYTHIA-simulated N_{ch} distributions and p_{T} spectra at LHC energies and subsequently tested on other simulated energies. Finally, the predictions from the ALICE-trained DNNs are compared to PYTHIA simulations over a wide range of collision energies (0.5 TeV $\leq \sqrt{s} \leq$ 100 TeV). This enables to quantify how the discrepancy of charged-particle production predicted by the QCD-inspired physics models implemented in PYTHIA to the LHC measurements evolves with collision energy.

The thesis is structured as follows. In Chapter 2, an overview of the theoretical background of high-energy physics and the machine-learning techniques relevant to this thesis is given. Subsequently, Chapter 3 provides a detailed description of the analysis conducted in this thesis. The results are discussed in Chapter 4. Finally, Chapter 5 provides a summary and outlook.

Chapter 2

Theoretical background

2.1 The strong interaction

The universe, at its smallest scale, consists of so-called indivisible particles. The field of high-energy physics studies these particles as well as the fundamental forces that govern them. The dominating fundamental force at the smallest scales is the strong interaction, whose effects are described by the theory of *quantum chromodynamics* (QCD). The strong interaction is mediated by an exchange particle called gluon that couples to the so-called color charge of the strong interaction. The fundamental color-charged particles, sensitive to the strong interaction, are so-called quarks and, notably, gluons themselves. In analogy to the primary colors, a quark carries one of three color charges: red (r), green (g) or blue (b). Their antimatter counterparts, the antiquarks, have a color charge corresponding to antired ($\overline{\mathrm{r}}$), antigreen ($\overline{\mathrm{g}}$) or antiblue ($\overline{\mathrm{b}}$). Gluons exhibit both a color and an anticolor charge. A characteristic of the strong interaction is the so-called *color confinement*. This dictates that only color-neutral particle states can be realized in nature. No free (anti-)quarks or gluons have been observed. Color neutrality can be achieved by combining all three (anti-)colors or a color with its anticolor counterpart. Therefore, to form color-neutral states, (anti-)quarks and gluons bind themselves into new particles, so-called hadrons. A bound state of three quarks (qqq) with color charges rgb is called baryon. Protons and neutrons are examples of baryons. A quark and an antiquark with complementary color and anticolor such as $\mathrm{r}\overline{\mathrm{r}}$, $\mathrm{g}\overline{\mathrm{g}}$ or $\mathrm{b}\overline{\mathrm{b}}$ can also form a bound state ($\mathrm{q}\overline{\mathrm{q}}$). These bound states are called mesons, like pions and kaons. Hadrons stay bound by constantly exchanging gluons. Since gluons are also color-charged, they can also attract each other, forming so-called *strings*. When attempting to separate a bound quark-antiquark pair, the binding *strings* come closer together. As a result, the binding force of the meson becomes increasingly stronger, explaining the phenomenon of *color confinement*. If the energy is increased further until the string breaks, the released potential energy becomes sufficient to

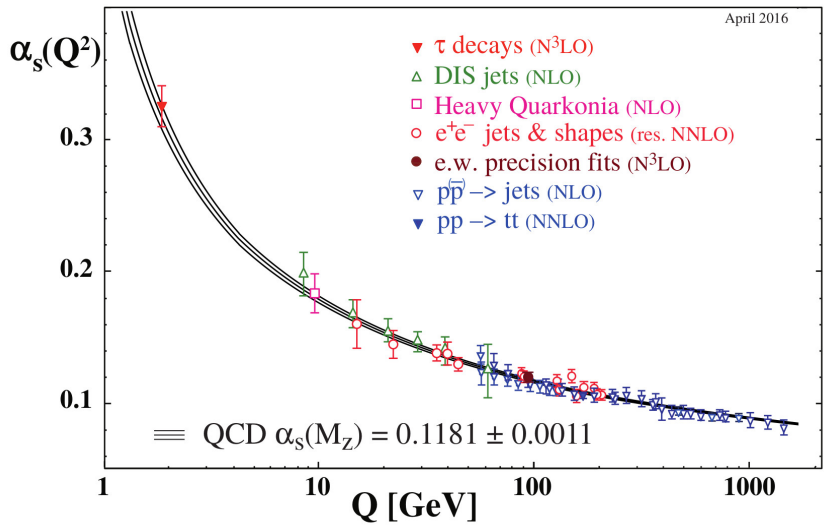
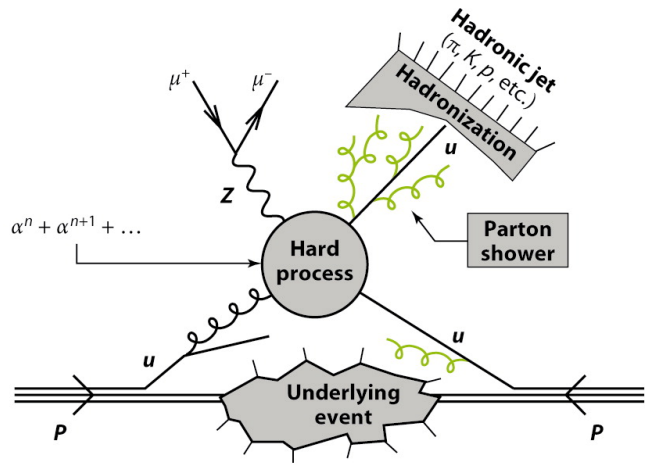


Figure 2.1: Measurements of α_s as a function of Q as published in the *quantum chromodynamics* review of [5].

create a new quark-antiquark pair. The coupling constant of the strong interaction, α_s, determines its strength. A summary of α_s measurements as of 2016 is shown as a function of Q for a temperature of T = 0 K in Figure 2.1. Notably, α_s can vary significantly with the momentum transfer Q between strongly interacting particles. Due to its dependence on Q, α_s is commonly referred to as a *running coupling constant*. Typically, the world average for α_s is given for an energy scale of the invariant mass of the Z^0 boson (M_Z), one of the exchange particles of the weak interaction. The current world average lies at $\alpha_s(M_Z) = 0.1179 \pm 0.0009$ [6]. As shown in Figure 2.1, α_s becomes very large for low momentum transfer and very small for large momentum transfer. A large momentum transfer corresponds to small distances between the interacting particles. Therefore, for large momentum transfer and small distances, the coupling constant converges towards zero. In this scenario, quarks are quasi-free as they are no longer confined in hadrons. This effect is known as *asymptotic freedom*. Notably, α_s is also dependent on the temperature. Therefore, in a thermalized medium α_s becomes small for high temperatures or energy densities, resulting in an exotic state of matter in which partons are quasi-free: a *quark-gluon plasma*. A strongly interacting QGP is formed within instants after a high-energy heavy-ion collision, where the energy density becomes extremely high. For processes with large momentum transfer, where α_s becomes small, the self-interaction of gluons becomes negligible. Therefore, these so-called "hard" processes can be accessed theoretically via perturbative methods. However, for processes with small momentum transfer, α_s diverges. Consequently, the self-interaction of gluons becomes increasingly relevant so that the processes cannot be described analytically and phenomenological models are needed. These processes



Butterworth JM, et al. 2012.
Annu. Rev. Nucl. Part. Sci. 62:387–405

Figure 2.2: Schematic representation of a pp collision with a hard quark-gluon scattering [7].

are called "soft" processes.

2.2 Particle production in pp collisions

Particle accelerators, like the LHC, allow to study the strong interaction by means of high-energy particle collisions, e.g. pp collisions. At these high energies, the probing wavelength of the colliding particles becomes small, resulting in a higher resolution. The interactions occur between the fundamental building blocks of the colliding particles, quarks and gluons, also referred to as partons. These partons undergo various interactions during the collision. The majority of the partons scatter inelastically, resulting in the production of new particles. The momentum fraction of a parton in the proton is described by a so-called *parton-distribution function* (PDF). The accelerated partons emit gluons during the collisions. These gluons can split up into gluons or quark-antiquark pairs which can, in turn, emit further gluons. This results in a so-called *parton shower*. The energy of each showering parton is distributed between the produced partons. Therefore, the *parton shower* keeps propagating until the parton energy is too small for further gluon radiation. The color-charged partons from the shower cannot stay isolated due to *color confinement*, so that new quark-antiquark pairs are formed. Consequently, the created quark-antiquark pairs bind other quarks and antiquarks in the parton shower to form color-neutral hadrons. The hadrons form a cone-like structure, a so-called *jet* that can be measured by the detectors at the accelerator as a cluster of hadrons. Jet production is a dominant process of hard in-

teractions. The process of forming hadrons is called *hadronization*. Because of the decreasing energy of the partons in the shower, *hadronization* involves soft partonic interactions.

The hardest interaction in a collision is called the primary process. However, other numerous interactions occur between the residual initial state partons that did not participate in the primary process of the collisions. All these additional interactions are summarized into the so-called *underlying event* (UE) of the collision. The momentum transfer involved in the UE is significantly lower than in the primary process. Therefore, the UE often yields softer and fewer particles. An illustration of the hard scattering with resulting showering and *hadronization* into final-state particles as well as UE processes in a pp collision is shown in Figure 2.2. Understanding the effect of the UE on the final state can help to separate its contribution from that of the primary process. This can lead to more precise measurements of the primary process of a collision. Soft processes contribute significantly to the final state of a collision. Due to large values of α_s, these processes cannot be accessed analytically. There are various *Monte Carlo generators* that employ such phenomenological models to simulate the underlying processes of particle collisions. One of these models is the *Lund String Model*, which aims to describe the *hadronization* process in high-energy particle collisions. As such, it attempts to describe the process of partons combining to form color-neutral hadrons. The *Lund String Model* assumes that this process involves the previously introduced *string breaking*. In the latter, the exchanged gluons between a quark-antiquark pair attract one another due to their self-interaction and form *strings*. The potential for the strong interaction between the quark and the antiquark is described by two terms:

$$V(r) = -\frac{4}{3}\frac{\alpha_s(Q(r)) \cdot \hbar c}{r} + k \cdot r. \tag{2.1}$$

The first term is Coulomb-like and dominates the potential at small distances r, where it describes the behavior of the quark and the antiquark as quasi-free particles, the *asymptotic freedom*. The second term dominates for larger distances. It describes the linear increase of the potential energy with increasing distance between the quark and antiquark, with k representing the energy density per unit length. The *Lund String Model* is implemented in the PYTHIA event generator [1]. PYTHIA can simulate particle collisions, in the following also called events, from the initial to the final state of the collision. The free parameters in the phenomenological models must be tuned in order to correctly describe the underlying processes of a collision. For that purpose, measured final states of particle collisions are used that constrain the possible values for these parameters. Therefore, highly precise and extensive measurements at collider experiments are essential to tune them. For example, the *Monash* tune of

PYTHIA incorporates constraints provided by measurements of e^+e^- experiments to tune the final-state radiation and hadronization parameters [2]. Furthermore, it utilizes extensive measurements of pp collisions at the LHC to tune further parameters. This tune is designed to describe the measurements from the LHC at a center-of-mass energy of $\sqrt{s} = 7$ TeV. Therefore, it is well suited to simulate high-energy hadron collisions. Another collider experiment, the Relativistic Heavy Ion Collider (RHIC), conducts pp collision at significantly lower $\sqrt{s}$, the highest being $\sqrt{s} = 0.2$ TeV. Previous publications have shown that UE observables of measured pp collisions at RHIC for $\sqrt{s} = 0.2$ TeV are not correctly described by the *Monash* tune [8]. In low-energy collisions, soft processes become more relevant to the final state of a collision. Therefore, it has been argued that the discrepancies between the simulations and the measurements can be traced back to incorrect modeling of the soft QCD processes of the UE by the *Monash* tune. To address this issue, PYTHIA tunes that are optimized to describe measurements of low-energy collisions have been proposed.

As discussed previously, event generators like PYTHIA require precise measurements of the final state of a collision to tune their parameters. This thesis focuses on the final-state charged particles (predominantly consisting of pions, kaons and protons) produced in high-energy pp collisions measured at ALICE.

2.3 Charged particles in ALICE

In the Large Hadron Collider (LHC) at CERN hadrons collide at the currently highest possible center-of-mass energies in the world. It consists of a 27 km-long underground ring equipped with superconducting magnets at a depth of about 100 meters. Within the ring, opposing hadron beams are accelerated close to the speed of light in two separate beam pipes. At four interaction points, these opposing beams are brought to collision. Four large experiments (ATLAS, ALICE, CMS and LHCb) are positioned around these crossing points to measure the particles produced in the collisions. The strength of the magnetic fields from the superconducting magnets as well as the radius of the LHC determine the maximum center-of-mass energy that can be reached, which corresponds to $\sqrt{s} = 14$ TeV. Currently, in LHC Run 3, proton beams have been accelerated to a maximum of $\sqrt{s} = 13.6$ TeV.

ALICE is dedicated to studying an exotic state of matter, the *quark-gluon plasma*. For this purpose, the research program of ALICE focuses on measuring central heavy-ion collisions such as lead-lead (Pb–Pb) to reach energy densities high enough to form a QGP. ALICE consists of a wide range of sub-detectors, which allow highly precise measurements. A schematic representation of the ALICE detectors is shown in Figure 2.3. The ALICE detector provides excellent tracking capabilities for charged

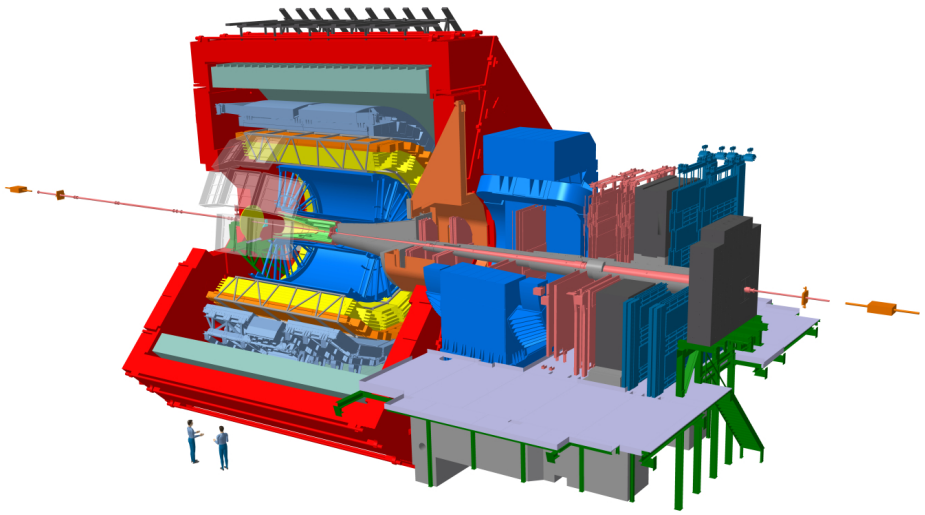


Figure 2.3: Schematic representation of the experimental setup of ALICE [9].

particles. The most relevant detectors for the measurement of charged particles are the *Inner Tracking System* (ITS) and the *Time Projection Chamber* (TPC). The ITS consists of six layers of silicon detectors. It is used to identify the primary vertex of a collision with extremely high precision. The TPC is a large gas-filled chamber that allows for the three-dimensional reconstruction of charged-particle trajectories via their ionization of the gas molecules in the chamber. The trajectories are bent in the 0.5 T magnetic field of a solenoid magnet. By measuring their curvature, the transverse momentum or p_{T} of the charged particles produced in the collisions can be determined. Since the transverse momentum of the colliding protons is zero, the measured transverse momentum of the produced charged particles can only originate from the collision. ALICE is specially designed for measurements down to very low p_{T}, which characterize soft QCD processes and consequently provide important constraints for phenomenological models. The number of charged particles produced in a collision is commonly referred to as multiplicity or N_{ch}. Measuring N_{ch} and p_{T} for many collisions yields charged-particle N_{ch} distributions and p_{T} spectra. Both are basic observables that characterize the charged-particle production mechanisms of high-energy collisions. Multiplicity distributions represent the probability for the production of a given number of charged particles in a collision or $P(N_{\mathrm{ch}})$. Transverse momentum spectra represent the production rate of charged particles with a given p_{T} in a collision, the so-called *yield*.

While ALICE is dedicated to the study of heavy-ion collisions, pp collisions still play a significant role as the charged-particle spectra of pp collisions are useful for the tuning of phenomenological models. Furthermore, reference measurements from pp collisions are required to study QGP properties in heavy-ion collisions via the *nuclear*

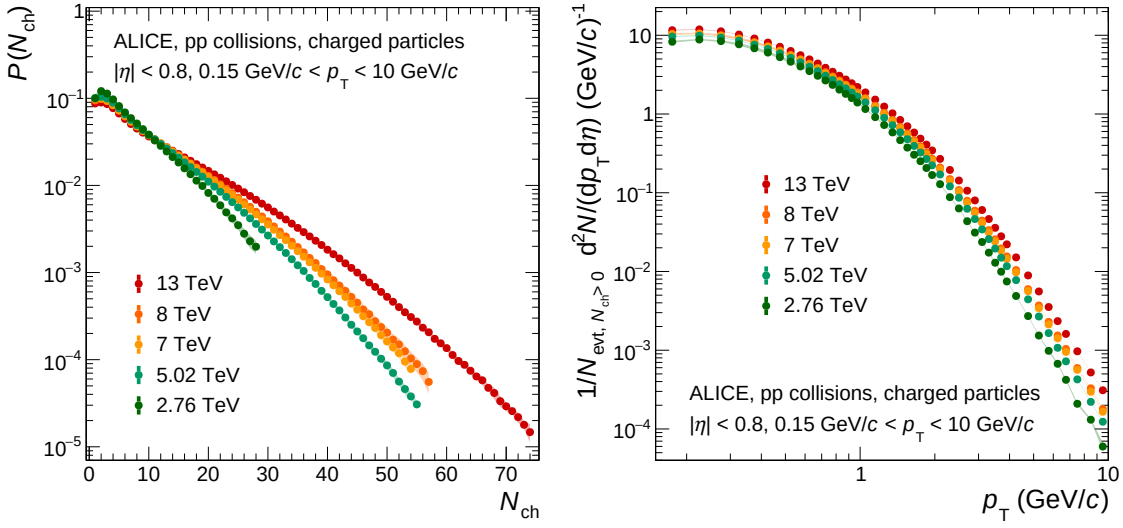


Figure 2.4: Charged-particle multiplicity distributions and p_{T} spectra of pp collisions at different $\sqrt{s}$ as published by the ALICE collaboration in [3].

modification factor.

A recently published paper by the ALICE collaboration presents a comprehensive analysis of charged-particle production measured with ALICE during LHC Run 1 and Run 2 (2009 - 2018) for various collision systems. The measurements contain pp collisions at the following center-of-mass energies: $\sqrt{s} = 2.76, 5.02, 7, 8$, and 13 TeV [3]. The results of the analysis include the N_{ch} distribution and the charged-particle p_{T} spectrum of the collisions within the kinematic range $0.15\,\mathrm{GeV}/c < p_{\mathrm{T}} < 10\,\mathrm{GeV}/c$ and $|\eta| < 0.8$. Only collisions with at least one charged particle in this kinematic region are considered. These measurements lay the foundation for this thesis. The N_{ch} distribution and the charged-particle p_{T} spectrum for pp collisions at the different $\sqrt{s}$ are shown left and right in Figure 2.4, respectively. The N_{ch} distributions show that the number of produced charged particles increases with increasing center-of-mass energies. At low multiplicities, the distributions exhibit a peak at around $N_{\mathrm{ch}} = 2$, marking the most probable number of charged particles produced in a collision. The p_{T} spectra show an exponentially decreasing trend at low p_{T}, where the charged-particle production is dominated by soft processes. At higher p_{T}, where the processes are increasingly harder, the trend follows a power-law-like behavior. These two components of the spectrum point to the two different production mechanisms, hard and soft, and are often implemented for the parametrization of the spectrum [10].

This thesis focuses on the energy dependence of charged-particle production. Figure 2.5 shows the average number of charged particles measured by different experiments as a function of the center-of-mass energy. The ALICE points are highlighted in red. Measurements of the same event class can be parametrized by a power-law func-

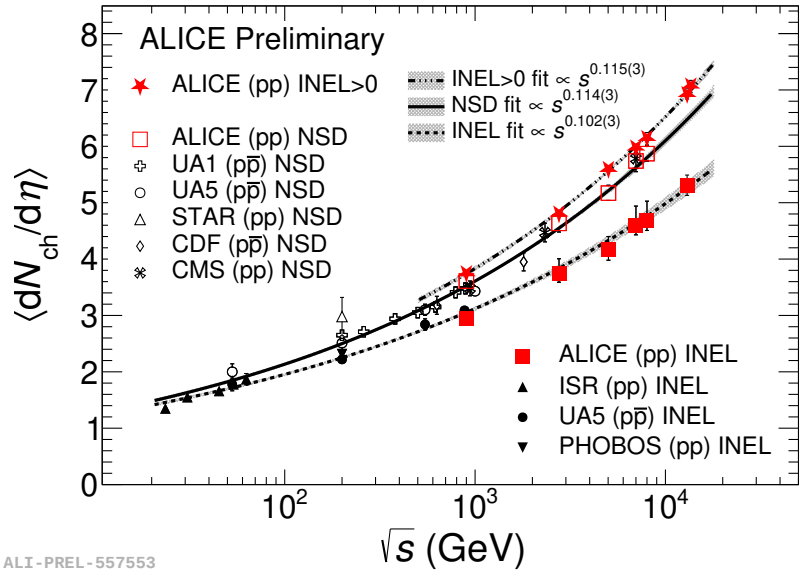


Figure 2.5: Average number of charged particles measured by different experiments in pp and p$\bar{\text{p}}$ collisions between $\sqrt{s}$ = 23.6 GeV and 13.6 TeV [11].

tion (s^n). This thesis aims to model the energy-dependent charged-particle spectra of pp collisions at ALICE based on measurements published by the ALICE Collaboration in [3] over a wide center-of-mass energy range using *deep neural networks*.

2.4 Neural networks

The high performance of artificial neural networks (NNs) in solving complex tasks has contributed significantly to the success of machine learning in recent years. Neural networks are often implemented for two types of tasks: assigning a label to data based on their features, so-called classification tasks, and finding the functional relationship between input and output values, so-called regression tasks. NNs are designed to resemble the inner workings of biological neural networks in the human brain, where incoming information propagates through the connections between so-called neurons and is processed within each of these neurons. In analogy, NNs are comprised of interconnected neurons, as well, and propagate the incoming data similarly. Therefore, they can imitate the brain's learning process and are widely employed to solve complex tasks.

2.4.1 Structure

Artificial neural networks consist of interconnected neurons that are organized into layers. The structure of a NN includes both an input layer and an output layer. Additionally, NNs can have more layers in between, so-called hidden layers. The

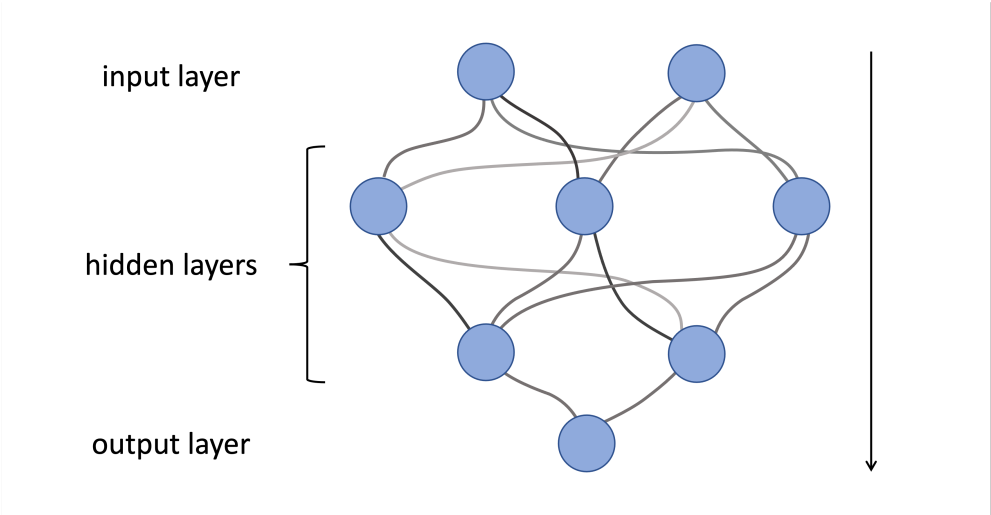


Figure 2.6: Schematic representation of a *feedforward deep neural network*.

structure of a NN is illustrated in Figure 2.6, where the input layer has two neurons and the output layer has a single neuron. The hidden layers consist of three and two neurons, respectively. The number of neurons in the input layer of a neural network corresponds to the number of so-called input features that characterize the data. As shown in Figure 2.6, the input layer receives the input data and distributes it through the connections between neurons to the hidden layers. The hidden layers, also interconnected, are responsible for transforming and transmitting the data to the output layer. The output layer then generates the output of the NN.

NNs with more than one hidden layer are called *deep neural networks* (DNNs). Furthermore, in so-called *feedforward neural networks*, the information flows in a forward direction, propagating from the input layer to the output layer. In this type of NN, only neighboring layers are interconnected. In case the neurons in each hidden layer are connected to all neurons from the neighboring layers, a *feedforward neural network* is considered to be fully connected.

2.4.2 Data propagation

In NNs, the data is transformed in each layer and passed to the next through the connections between neurons, so that the output of one layer becomes the input of the next. This process of transforming and propagating the data toward the output layer is called a forward pass through the network. In the following, the mathematical operations and transformations related to the forward pass are presented for *feedforward deep neural networks*.

The connections between neurons as well as the neurons themselves are assigned numerical values. These so-called weights w and biases b correspond to the NN's

parameters. The weights of the connections between neurons assign a certain degree of importance to the incoming data from the previous layer. NNs aim to make informed predictions by recognizing relationships and patterns within the data. This is done by adjusting the NN's parameters during the training process. The input of a neuron is a linear transformation of the incoming data, corresponding to the weighted sum z of the output values y_n of each neuron n from the previous layer:

$$z = \sum_n w_n \cdot y_n + b \quad . \tag{2.2}$$

Each layer processes the incoming data z from the previous layer, through a non-linear transformation. For this purpose, a so-called activation function f is applied to z. The output values of each neuron in the current layer become $y = f(z)$.

To compute the output values of an entire layer, the data, weights and biases are represented by matrices. In the following example, two neighboring layers $(j-1)$ and j are considered. Here, layer $j-1$ has m neurons and layer j has n neurons. Since all neurons from layer $(j-1)$ are connected to those of layer j, the weights form a matrix $\hat{\mathbf{W}}_{n,m}^{(j,j-1)}$ with dimensions $n \times m$. The bias and the output of layer j can be represented as vectors $\vec{b}^j$ and $\vec{y}^j$ with dimensions of $n \times 1$. The output of layer j can then be calculated by adapting the dimensions in Equation 2.2:

$$\vec{y}^j = f(\vec{z}^j) = f\left(\hat{\mathbf{W}}_{n,m}^{j,j-1} \cdot \vec{y}^{j-1} + \vec{b}^j\right) \quad . \tag{2.3}$$

The functional representation F of the whole network is a recursive form of Equation 2.3 through all its L layers and computes the output $F(\vec{x})$ for a given set of input features $\vec{x}$ as:

$$F(\vec{x}) = f^L\left(\vec{b}^L + \hat{\mathbf{W}}^L \cdot f^{L-1}\left(\vec{b}^{L-1} + \hat{\mathbf{W}}^{L-1} ... f^1\left(\vec{b}^1 + \hat{\mathbf{W}}^1 \cdot \vec{x}\right) ...\right)\right) \quad . \tag{2.4}$$

Thus, a neural network is essentially a multidimensional function with many parameters. The number of input features varies and matches the number of neurons in the input layer; the same applies to the output features. Figure 2.6, introduced in the previous section, depicts an example of a network architecture with two input features and a single output feature. The activation function may also vary between layers.

2.4.3 Learning process

In this thesis, *deep neural networks* are trained with data whose output is known. This approach is called *supervised learning*. In this case, the neural network aims to

approximate an unknown output function $f^\star$, which represents the true functional relationship between the input and the target values, with F. F represents the function of the network shown in Equation 2.4. By adjusting the parameters α (weights and biases) of F during training, the approximation of $f^\star$ becomes more accurate, effectively optimizing the NN's performance. For this, the NN performance must first be quantified by the deviation of the NN's predictions, or values of F, from the target values. This is done using a metric. Metrics are mathematical constructs that allow to define a distance between elements within a set called metric space [12]. In machine learning, metrics come into play by specifying a distance or deviation between the NN predictions and target values. Thus, they generate numerical values that quantify the NN's performance. Different metrics focus on evaluating other characteristics of the NN's performance. Therefore, the choice of metric strongly depends on the task at hand. A common metric for regression tasks is the *Mean Absolute Error* (MAE), which measures the absolute difference of F from the target function $f^\star$ averaged over the number of samples N for a given set of inputs $\{x_i\}$ with $i = 0, ..., N$. This metric is α-dependent and, if evaluated during the NN's learning process, is commonly referred to as the loss function. To optimize the NN's performance, the goal of the learning process is to minimize the loss function:

$$\mathcal{L}_{MAE}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} |F(x_i, \alpha) - f^\star(x_i)| \quad . \tag{2.5}$$

This is done by finding an optimal set of parameters α that result in a minimum of the loss function. This optimal set of parameters is characterized by the gradient of the loss function becoming zero at these values of α, indicating a local or, ideally, a global minimum of the loss function. Therefore, to find the optimal α the gradient of $\mathcal{L}(\alpha)$ must be calculated for each parameter in the network. However, these calculations can be computationally expensive. Therefore, a balance must be found between achieving accurate calculations of the gradient and reducing computational cost. As shown in Equation 2.4, the output of each layer in the network depends on all previous layers. Therefore, the chain rule can be applied when computing the gradient of each layer, avoiding redundant gradient calculations. Consequently, the gradient for each parameter can be computed more efficiently by applying the chain rule going backward through the network layer by layer in a so-called backward pass. This method is known as backpropagation. Here, the backward pass for computing the gradient is done after the forward pass of a batch of training samples. A detailed derivation of the calculations used to compute the gradient for each NN parameter during the backward pass is given in [13].

A widely used optimization method for adjusting the NN's parameters is *Gradient*

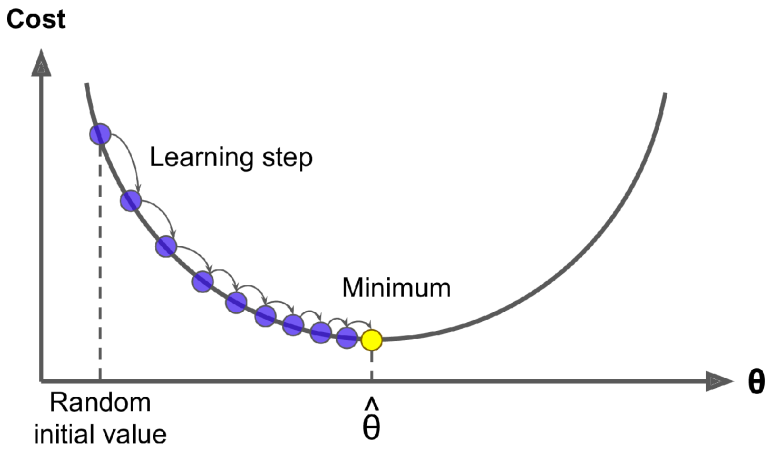


Figure 2.7: Illustration of the stepwise adjustment of the NN's parameters ($\hat{\theta}$) towards the minimum of the loss function (cost) [14].

Descent (GD). As shown in Equation 2.6, the NN's parameters are adjusted in the direction of the falling gradient of $\mathcal{L}(\alpha)$, with a specific step size η:

$$\alpha_j' = \alpha_j - \eta \cdot \frac{\partial \mathcal{L}}{\partial \alpha_j} \quad . \tag{2.6}$$

The adjustment of α gets smaller, the closer the gradient comes to zero. This enables to fine-tune the set of parameters α to their optimum values. This method is an analogy to the physical phenomenon of a body moving towards the minimum of a potential. An example of GD is illustrated in Figure 2.7. Here, the loss function (or cost) is shown as a function of the NN's parameters. Additionally, the updated parameters show larger steps where the slope is steep and then gradually smaller steps once the location of the minimum is approached.

In GD, the gradient is calculated for the entire dataset. A different method estimates the gradient for a small batch of training samples instead of the whole dataset. This helps reduce computational costs as well as accelerate training. This method is commonly referred to as *Stochastic Gradient Descent* (SGD) and becomes increasingly significant for large datasets.

A training iteration consists of a single forward and backward pass over the whole dataset. After a training iteration has been completed, a so-called epoch has passed. The whole training process continues for many epochs, during which the parameters are updated according to Equation 2.6. Ideally, at some point during training, the convergence of the loss function can be observed with an increasing number of epochs and indicates that a minimum value of the loss function is approached.

2.4.4 Regularization techniques

A major challenge when training neural networks is to ensure that they acquire a sufficiently generalized knowledge of the correlations in the training data. This generalized knowledge allows the NN not only to successfully fit the training data but to make accurate predictions on new, unseen data as well. Commonly referred to as generalization power, it helps prevent the NN from memorizing details or fluctuations in the training data too well, performing badly on new data. This effect is known as overfitting. Different techniques help improve generalization, like data splitting, *early stopping*, *dropout* as well as *L1* and *L2* regularization, all of which will be discussed in the following.

The process of data splitting consists of dividing the data into separate datasets, each serving a different purpose for the NN. Typically, the data is separated into three datasets for training, validation and testing. The training dataset is implemented during the training process to compute the loss and update the NN's parameters. Furthermore, the primary role of the validation dataset is to help detect overfitting which can happen if the NN is trained for too long resulting in a bias of the NN's parameters towards the training data. The longer an overfitting NN is trained, the worse it performs on unseen data. Calculating the so-called validation loss, or value of the loss function for the validation dataset, quantifies the performance of the NN on unseen data. Therefore, a gradual increase in the validation loss as a function of the epochs is an indicator of overfitting. Once it ceases to improve after a certain number of epochs, usually referred to as *patience*, the training process is stopped. As discussed earlier, monitoring the validation metric provides insight into whether the NN is experiencing overfitting, since it would gradually worsen. This information can be actively used to decide when to stop the training process, effectively reducing overfitting. This technique is called *early stopping*. Additionally, it is common practice to assign data to a third dataset, entirely independent of the training process. This so-called test dataset has the purpose of evaluating the NN's performance on new, unseen data with a given metric. Since the dataset did not influence the training process or its duration, the evaluation results are unbiased, providing a fair estimate of the NN's performance.

Two further regularization techniques help increase the generalization power of the NN by penalizing large weights when computing the loss function [15]. This is done by adding a penalty term to the loss function. One of these techniques adds a penalty term to the loss function that is proportional to the absolute values of the weights. The second technique adds a the penalty term is proportional to the squared values of the weights. The resulting loss functions are shown in Equation 2.7 and Equation 2.8. The

choice of the constant λ, also known as the regularization factor, is key to achieving a NN that neither underfits, as is the case for large values of λ, nor overfits for small values of λ.

$$\mathcal{L}_{L1}(\alpha) = \mathcal{L}(\alpha) + \lambda \cdot \sum_i |w_i| \tag{2.7}$$

$$\mathcal{L}_{L2}(\alpha) = \mathcal{L}(\alpha) + \lambda \cdot \sum_i w_i^2 \tag{2.8}$$

By strategically adjusting its weights during the training process, a neural network can learn to give more importance to features that best represent the relationships and patterns within the data. This can be done by assigning larger weights to these features and smaller weights to features of little contribution to the desired output. The first regularization technique, so-called *L1* regularization, encourages the NN to drive some of its weights to zero. As a result, the most relevant features are emphasized and the less influential ones are ignored. Therefore, *L1* regularization is useful for improving feature selection.

The second technique, so-called *L2* regularization, promotes a more balanced distribution of the weight values across the NN. Accordingly, it facilitates the processing of highly correlated features by preventing the dominance of a single feature.

An effective regularization technique combines both *L1* and *L2* regularization to leverage their respective strengths, as seen in Equation 2.9. The resulting penalty term of the loss function is a linear combination of the *L1* and *L1* penalty terms. The strength of each term is controlled by separate coefficients λ_1 and λ_2, as shown in Equation 2.9. This allows the so-called *elastic net* to consider both feature selection and weight distribution simultaneously [16].

$$\mathcal{L}_{L1+L2}(\alpha) = \mathcal{L}(\alpha) + \lambda_1 \cdot \sum_i |w_i| + \lambda_2 \cdot \sum_i w_i^2 \tag{2.9}$$

Dropout is a regularization method commonly employed when training neural networks. During this process, a *dropout* rate is assigned to each neuron in a layer, determining the probability of that neuron being deactivated during training. This means that the output of the neuron is set to zero. The *dropout* rate indicates the portion of neurons to be deactivated, while the specific neurons are selected at random. This random selection introduces a form of noise into the training process. As a result, the NN is encouraged to develop more general representations of the underlying relationships in the data. While this regularization technique has proven useful in enhancing the generalization of classification problems, it is less suitable for handling regression problems. It is important to consider that *dropout* is applied exclusively

during training and is deactivated afterward. Therefore, deactivated neurons regain activity after training, producing non-zero outputs. These non-zero outputs of the activated neurons influence the overall output of each layer, subsequently influencing the output of the network. During training, the parameters of the output layer (weights and biases) are adjusted to transform the outputs of the last hidden layer to better describe patterns in the training data. As a result, a discrepancy arises between the outputs generated by the NN with *dropout* active or inactive. This discrepancy can lead to the NN yielding less accurate predictions once the training is concluded, undermining the NN's performance. Therefore, it becomes crucial to choose regularization techniques that are suitable for the specific task at hand. The negative effects of *dropout* on the performance of neural networks for regression were discussed in [17].

2.4.5 NN hyperparameters

The hyperparameters of a neural network determine the NN's architecture and cannot be learned during training. These hyperparameters include the depth, width, activation function(s), optimizer, learning rate, batch size, regularization and initialization of the network. These new concepts are introduced below.

Depth of the network

The depth of the network refers to the number of hidden layers in the network. As seen in Equation 2.4, the order of the data transformation increases with each layer, similar to the order of polynomials. This allows deeper networks to capture more intricate patterns within the data. However, excessive depth can lead to issues such as vanishing or exploding gradients. These can occur when the gradients get very small or very large during backpropagation. As shown in Equation 2.6, "vanishing" gradients (close to zero) lead to a minimal adjustment of the NN's parameters, which causes slow learning. When gradients "explode", the updates to the NN's parameters become very large, which results in the NN failing to converge during the training process. The issue of vanishing and exploding gradients is especially present in deep networks since the gradients are repeatedly multiplied when they propagate backward through the network's layers during backpropagation, following the chain rule.

Width of the network

The width of the network represents the number of neurons in the network's hidden layers. In Figure 2.6, the input data, consisting of two features, gets partitioned into three features when passing from the input layer (with two neurons) to the first hidden layer (with three neurons). As shown by this example, the more neurons are

in the hidden layers, the finer the features that the input data gets partitioned into. Therefore, a wider network has the potential to represent complex functions with a higher number of parameters. However, too wide networks tend to overfit the training data by memorizing it. In recent years, the focus has shifted towards deep learning, where NNs are becoming deeper instead of wider [18].

Activation function

Activation functions are designed to provide the NN with the necessary nonlinearity to tackle problems of high complexity and to imitate the observed threshold behavior of biological neural networks. As discussed in Subsection 2.4.2, the activation function of a layer is applied to the output of each neuron. Using a linear function as an activation function would only deliver linear transformations and thus limit the NN's problem-solving abilities. Besides nonlinearity, there are further conditions that the activation function must fulfill. Equation 2.4 and Equation 2.5 show that the loss function depends on the activation function. Thus, the gradient of the activation function is required to calculate the gradient of the loss function during training. Consequently, the activation function needs to be continuous and differentiable. Furthermore, when employing*deep neural networks*, it is important to choose activation functions that are not inclined to vanishing or exploding gradients, which were discussed earlier. Some commonly used activation functions include $ReLU$ [19], *tanh*, and *sigmoid*. Functions like *softplus* [20], $SELU$ [21] and *swish* [22] belong to the more modern activation functions that effectively address some limitations of the traditional activation functions. Their performance varies widely depending on their application. Some of these examples are depicted in Figure 2.8.

Batch size

During backpropagation, the backward pass for computing the gradient is done after the forward pass of a batch of training samples. The gradient is then calculated for this batch of samples. This speeds up training and decreases the amount of used memory space. For this reason, the choice of batch size (number of training samples in each batch) can be of great importance to the NN's performance. In deep learning, it is common practice to choose a batch size that is a power of two.

Learning rate

The step size η in Equation 2.6, also known as the learning rate of the network, plays a big role in the success of the training process: if a too-small learning rate is chosen, then the loss function could land in a local minimum during training and converge

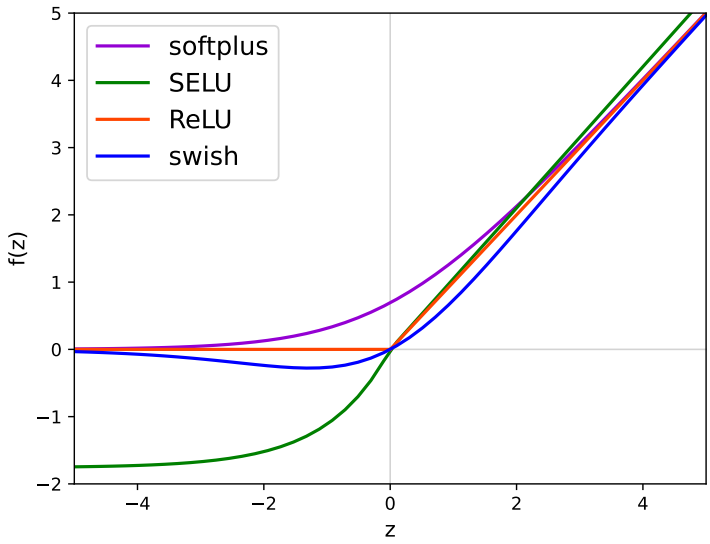


Figure 2.8: Softplus, SELU, ReLU, and swish activation functions.

there. Therefore, it must be large enough to avoid local minima, but small enough for the loss function to achieve convergence.

Optimizer

Optimizers are neural network optimization algorithms that adjust the NN's parameters. They use mathematical functions to help minimize the loss function. One of the most common optimizers, *Gradient Descent* (GD), was introduced in subsection 2.4.3. This optimizer implements a constant learning rate to update the parameters. In contrast, *Adaptive Moment estimation* (*Adam*) is a more advanced optimizer that applies an adaptive learning rate [23]. *Adam* calculates an individual learning rate for each parameter in the network based on the first and second moments of the gradients. The first moment corresponds to the mean of the gradients, while the second moment corresponds to the variance. These moments are computed as exponentially decaying moving averages of past gradients and squared gradients, respectively. The exponential decay gives more weight to recent gradients and gradually reduces the impact of older gradients. This helps capture the overall trend of the gradients and leads to faster convergence during training. Since the moments are initialized as zeros at the start of training, a bias towards zero can be present in the early stages of training. To correct this bias, the averages are scaled using decay rates denoted as β_1 and β_2. These decay rates also help control the contribution of older gradients to the averages. The optimizer can balance considering recent gradients and including information about past gradients by adjusting these decay rates. This approach, commonly

referred to as the momentum method, aims to accelerate learning when encountering high curvature regions. Overall, *Adam* combines the advantages of adaptive learning rates, momentum-induced updates and bias correction to enhance and accelerate the learning process in neural networks. This makes it a popular choice for an optimizer.

Initialization

While the NN's weights and biases are constantly updated during training, they must first be initialized to random numbers. This may be crucial to the learning process because the initial parameter values determine the starting point in the loss function landscape within the parameter space during training. Therefore, depending on these initial values, it can be challenging for the network to achieve convergence. Furthermore, the initial values can result in the network getting trapped on a local minimum. Too small or too large weights can also cause vanishing or exploding gradients, respectively, as previously discussed in this section in terms of the activation function. For this reason, implementing weight initialization strategies can help choose suitable initial parameter values. However, the correct weight initialization strategy often depends on the choice of the activation function. For example, the *He initialization* is specifically designed for networks that employ the ReLU activation function [24]. Another common technique is *random initialization*, which samples weight values from a normal or uniform distribution. A further strategy consists of keeping the output variance constant across the whole network. This helps avoid exploding and vanishing gradients can be avoided. For this purpose, the initial weights are scaled depending on the number of neurons in neighboring layers and the initial values of the biases are set to zero. This is the strategy implemented by *Xavier initialization* [25].

2.4.6 Hyperparameter tuning

The choice of hyperparameters deeply impacts an NN's predictions and, in turn, its performance. Therefore, tuning these hyperparameters during the design of the NN's architecture is of great importance. The following will describe some of the most common methods to achieve this.

A common method for tuning hyperparameters is to define a discrete subset of the so-called hyperparameter space. This space is multidimensional, with each dimension representing a distinct hyperparameter. In this approach, a NN is trained for each combination of hyperparameters within the subset. These combinations form a so-called grid. Each of the trained NNs is then evaluated using a chosen metric on a test dataset, separate from the training data. This helps to measure the performance of each NN when encountering previously unseen data. Therefore, comparing the

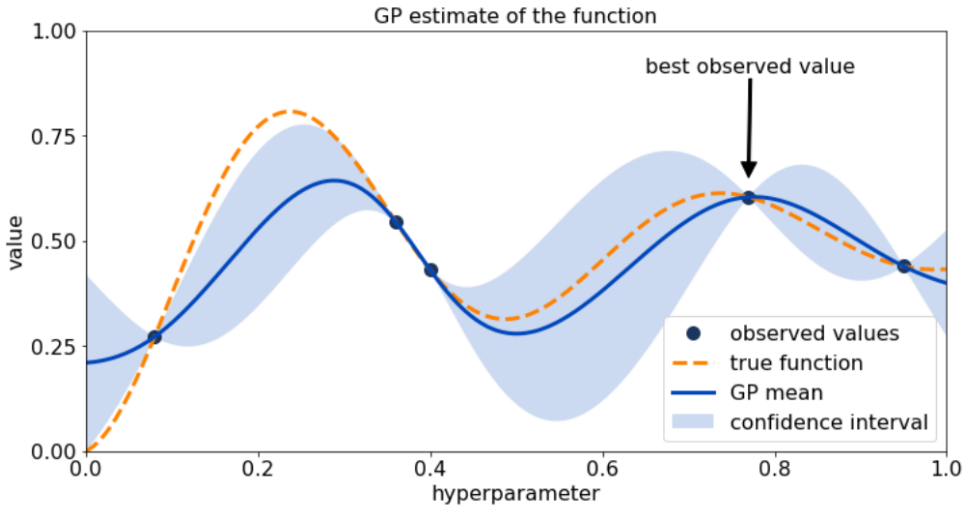


Figure 2.9: GP estimate of a function based on a given set of data points [27].

evaluation results for each NN reveals the best combination of hyperparameters on the grid. This method is straightforward to implement but has the disadvantage that it can be computationally expensive and lacks sensitivity to important hyperparameter values outside of the predetermined grid. This tuning method is called *Grid Search*.

Another technique comes into play when optimizing an objective function of an unknown analytical form that is computationally expensive to evaluate. In the field of machine learning, it is particularly useful for determining the set of hyperparameters that minimizes or maximizes the objective function. The objective function typically represents a performance metric of the NN. This technique involves choosing the hyperparameters to evaluate based on their likelihood of success in optimizing f. This approach leads to a more efficient optimization of the NN's performance compared to methods like *Grid Search*, where each possible combination of hyperparameters is evaluated. This technique is called *Bayesian Optimization* (BO) and will be discussed in the following. A more in-depth overview is provided by [26].

Bayesian Optimization is a decision-making strategy that employs *Bayesian Inference* [28] to optimize an unknown objective function. In the context of machine learning, this objective function represents an evaluation metric. In the case of regression tasks, the goal is to minimize evaluation metrics like MAE, so that the optimization problem takes the form:

$$\min_{x \in A} f(x) \quad . \tag{2.10}$$

Here, x represents a set of hyperparameters within a predefined region A of the hyperparameter space and f represents the evaluation metric.

To simplify the optimization process, BO works with a so-called observation model.

The role of this observation model is to approximate the unknown analytical form of f and thus predict the NN's performance for a given set of hyperparameters. This observation model is based on prior beliefs, which represent initial assumptions about the probabilistic distribution of the objective function f over the different hyperparameters within A. *Gaussian Processes* (GP) are often implemented to define this initial model. As further evaluations are conducted and thus more information about f becomes available, the initial model is updated into a so-called posterior distribution according to Bayes' Theorem. This is a refined approximation of the metric f, or NN's performance, by the observation model. Figure 2.9 provides an example of a GP estimation of an unknown function (true function) with an observation model by indicating the (GP) mean and variance (confidence interval) of the observation model for a given hyperparameter after evaluating the true function at different hyperparameter values (observed values).

With each subsequent evaluation, the observation model becomes increasingly accurate. Nevertheless, the goal is not to precisely reproduce the true function, as this would involve high computational costs. Rather, the aim is to efficiently identify the optimal hyperparameter values that minimize (or maximize) the true function f. The example of a BO process shown in both Figure 2.9 and Figure 2.10 aims to maximize the true function. Unlike the *Grid Search* method, where the sets of hyperparameters to evaluate and their order of evaluation are predetermined, BO takes a different, more efficient approach. The sets of hyperparameters to evaluate are not fixed beforehand. Instead, it is a dynamic process guided by the BO algorithm. Once a set of hyperparameters has been selected, an NN with these hyperparameters is trained. Subsequently, the performance (f) of this NN is evaluated using test data. Therefore, each evaluation of a given set of hyperparameters yields a further value of f, new information about f. As previously discussed, the observation model that approximates f is updated with each new evaluation to include the newly gained information. Therefore, it provides a forecast of which sets of hyperparameters are likely to optimize f (the performance of the NN) based on current observations. The BO strategy focuses on identifying and selecting the most promising set of hyperparameters to evaluate next. For this purpose, a so-called acquisition function measures the probability of improvement when evaluating f for a given set of parameters.

There are two different objectives during the optimization process that the acquisition function aims to balance. On one hand, the selection of hyperparameters that yield a low predicted value of f by the observation model in case of minimization or a high predicted value of f in case of maximization (exploitation). On the other hand, the exploration of the hyperparameter space by choosing hyperparameters that result in high uncertainty by the observation model (exploration). This is important because

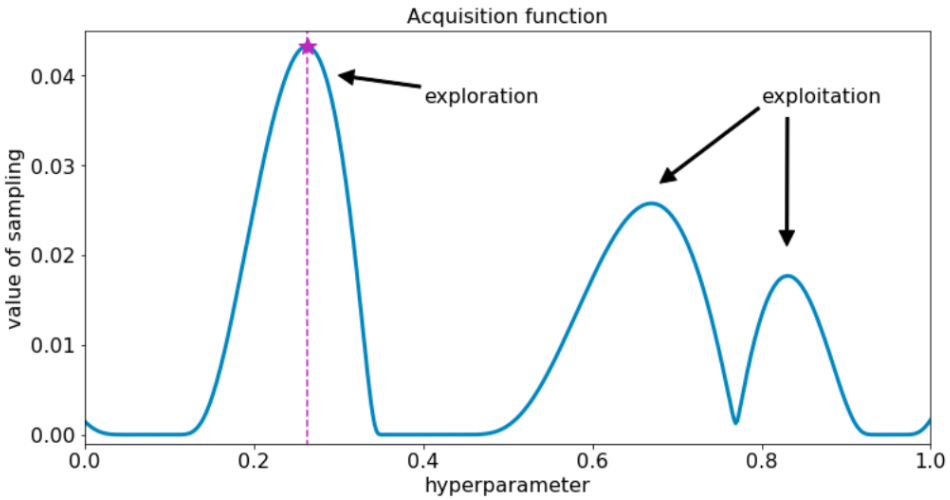


Figure 2.10: Acquisition function of GP estimate based on a given set of data points of a function [27].

evaluating f at regions of high uncertainty can yield valuable information regarding f. To achieve a balance between both objectives, the acquisition function generally consists of two terms, an exploitation and an exploration term, representing these objectives. A popular acquisition function is *Upper Confidence Bound* (UCB), which is defined for minimization problems as:

$$a_{UCB}(x, \beta) = \mu(x) - \beta \cdot \sigma(x). \tag{2.11}$$

In this case, $\mu(x)$ represents the mean of the current observation model for a given set of hyperparameters x and, with that, the expected value of f. As $\sigma(x)$ represents its standard deviation, it describes the prediction uncertainty of the observation model for a given x. β is a predetermined, positive constant that describes the balance between the exploitation and exploration terms. This means that the maximum of a points to the most promising set of hyperparameters to evaluate next according to current observations. Therefore, the iterative process of evaluating the sets of hyperparameters that maximize a can lead to convergence towards the global minimum (or maximum) of f. Figure 2.10 shows the acquisition function corresponding to the observation model in Figure 2.9. In this example, the objective of BO is to maximize the true function. The acquisition function exhibits a global maximum at the highest possible value of the current GP estimation within the confidence interval, indicating the high potential for improvement if the true function is evaluated at that hyperparameter value. The next step involves evaluating the true function at the location of the global maximum of the acquisition function, after which the GP estimation (observation model) is updated accordingly.

Chapter 3

Analysis

This chapter presents a detailed description of the analysis methods implemented in this thesis. The primary objective of the analysis is to develop a robust DNN model able to effectively interpolate the training data obtained from ALICE measurements and also to extrapolate it primarily in the energy dimension beyond the energy range covered by the LHC. A proof of principle is provided by a previous study performed on PYTHIA simulations [29], which demonstrated that particle-production observables can be successfully interpolated and extrapolated using neural networks. Also, in this thesis, PYTHIA is used to evaluate the extrapolation performance of a given model architecture.

In the first section of this chapter, a description of the different datasets utilized in the analysis is provided. Then, the basic characteristics of the implemented DNN model are introduced. Subsequently, data preparation and hyperparameter tuning procedures are outlined. These analysis methods lay the foundation for the estimation of the systematic uncertainties associated with the DNN models.

3.1 Datasets

3.1.1 ALICE data

This thesis utilizes data from a comprehensive analysis of charged-particle production measured with ALICE during LHC Run 1 and 2 (2009 - 2018). The dataset contains pp collisions at the following center-of-mass energies: $\sqrt{s} = 2.76$, 5.02, 7, 8, and 13 TeV [3]. Two different observables are considered: the charged-particle multiplicity distribution and the charged-particle transverse momentum spectrum within the kinematic range $0.15\,\mathrm{GeV}/c < p_{\mathrm{T}} < 10\ \mathrm{GeV}/c$ and $|\eta| < 0.8$. Only collisions with at least one charged particle in this kinematic region are considered. A systematic uncertainty related to the analysis procedures is assigned to the data and should be interpreted such that

the true data point could fall anywhere within the quoted bounds.

Figure 2.4 presented in Section 2.3 shows the N_{ch} distributions (left) and p_{T} spectra (right) measured with ALICE, which are the basis for this thesis. The N_{ch} distributions start at $N_{\mathrm{ch}} = 1$ and have different multiplicity ranges for each center-of-mass energy. These distributions have a data point for each multiplicity within the given range. The transverse momentum spectra consist of 46 p_{T}-intervals of logarithmically increasing width ranging from $p_{\mathrm{T}} = 0.15$ GeV/c up to 10 GeV/c. The corresponding systematic uncertainties, represented by error bands, are barely visible due to their small magnitude.

3.1.2 PYTHIA simulations

The primary goal of the analysis is to develop a model able to extrapolate to unmeasured center-of-mass energies. As these regions have not been explored experimentally, an assessment of the model's extrapolation capabilities is not possible based on ALICE data. However, data from Monte Carlo simulations can be used for testing purposes. These simulations can generate data at any given center-of-mass energy, which allows for training a model on the simulated data at the same LHC energies as the ALICE measurements presented in Subsection 3.1.1 and subsequently testing its interpolation and extrapolation capabilities across unmeasured $\sqrt{s}$ regions.

The simulated data is obtained using the Monte Carlo event generator PYTHIA with the *Monash* tune [2]. Proton-proton (pp) collisions are simulated at the LHC energies for which the ALICE measurements are available ($\sqrt{s}$ = 2.76, 5.02, 7, 8, 13 TeV) as well as other center-of-mass energies ($\sqrt{s}$ = 0.2, 0.5, 0.9, 1.5, 5.36, 13.6, 14, 20, 27, 50, 100 TeV) for testing the model performance. The test energies in the simulation are motivated as follows. $\sqrt{s} = 0.2$ TeV is the highest energy recorded at the Relativistic Heavy Ion Collider (RHIC) [30]. $\sqrt{s} = 0.9$ TeV is the minimum center-of-mass energy for pp collisions at the LHC. Additionally, pp collisions were measured at $\sqrt{s} = 0.9$ TeV in LHC Run 1 and, more recently, LHC Run 3. The measurements from LHC Run 3 include Pb–Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.36$ TeV and pp collisions at $\sqrt{s} = 13.6$ TeV. The LHC is built to reach a maximum center-of-mass energy of $\sqrt{s} = 14$ TeV. A proposed upgrade of the LHC, the High-Energy Large Hadron Collider (HE-LHC), foresees a potential increase in the maximum center-of-mass energy to $\sqrt{s} = 27$ TeV, while the Future Circular Collider (FCC-hh), envisioned as the successor to the LHC, is anticipated to reach center-of-mass energies of up to $\sqrt{s} = 100$ TeV [31]. Including simulations at $\sqrt{s} = 0.5$ and $\sqrt{s} = 1.5$ TeV in the dataset serves the purpose of improving the outcome of the hyperparameter scan as will be discussed further in Section 3.4 of this chapter.

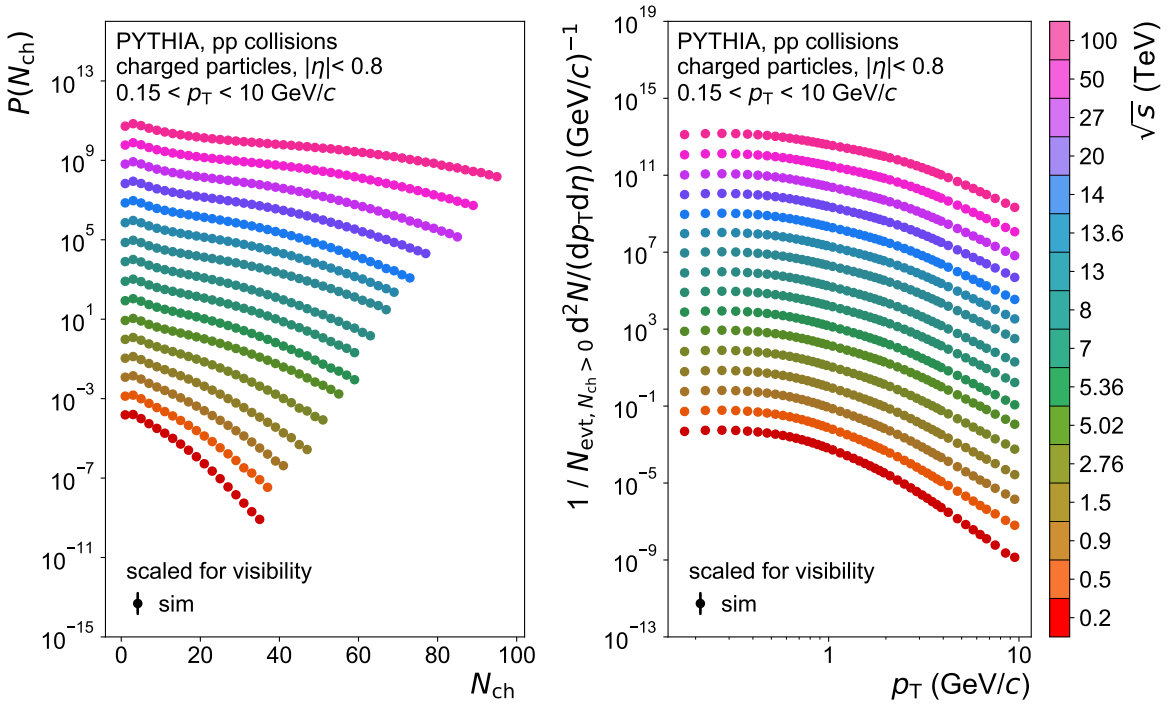


Figure 3.1: Charged-particle multiplicity distributions (left) and p_{T} spectra (right) for pp collisions at various center-of-mass energies simulated with PYTHIA.

The simulated N_{ch} distributions for each of the previously mentioned center-of-mass energies are shown in Figure 3.1 (left), while the p_{T} spectra are shown in Figure 3.1 (right). Each color in the figure represents a different $\sqrt{s}$ value. As seen in Figure 2.4, the N_{ch} distributions and p_{T} spectra at different center-of-mass energies tend to lie very close to each other, making it difficult to discern the individual spectral shapes. Therefore, the simulated data shown in Figure 3.1 is scaled with a $\sqrt{s}$-specific factor for better visibility. This means that N_{ch} distributions and p_{T} spectra with the same $\sqrt{s}$ value are scaled with the same factor. In this thesis, the visual scaling will be implemented in all figures containing N_{ch} distributions and p_{T} spectra. The N_{ch} range of each multiplicity distribution is selected such that data points with large statistical fluctuations are not present in the dataset and thus avoid possible bias during the training of the DNN caused by the fluctuating points. Since the probability for high N_{ch} diminishes with decreasing $\sqrt{s}$, the ranges are usually shorter for lower center-of-mass energies. In contrast, the p_{T}-range is the same for all simulated p_{T} spectra, reaching up to 10 GeV/c. An overview of the number of events, the selected N_{ch} range, the visual scaling factor and motivation for each simulated energy is given in Table 3.1. Since the N_{ch} distributions have a high data point density, only data points with odd N_{ch} values are shown in Figure 3.1 (left) and all further Figures in this thesis depicting N_{ch} distributions.

$\sqrt{s}$ (TeV)	events (M)	max. N_{ch}	visual scaling	motivation
0.2	27.66	35	10^{-3}	RHIC
0.5	11.11	37	10^{-2}	-
0.9	13.29	42	10^{-1}	LHC minimum
1.5	20.42	48	10^{0}	-
2.76	8.72	52	10^{1}	training
5.02	7.16	56	10^{2}	training
5.36	11.82	60	10^{3}	LHC Run 3 Pb–Pb
7	11.89	60	10^{4}	training
8	9.80	63	10^{5}	training
13	12.06	67	10^{6}	training
13.6	14.22	70	10^{7}	LHC Run 3
14	14.26	74	10^{8}	LHC maximum
20	26.94	77	10^{9}	-
27	22.29	85	10^{10}	HE-LHC
50	22.55	90	10^{11}	-
100	4.92	95	10^{12}	FCC-hh

Table 3.1: Number of events, maximum selected N_{ch}, factor for the visual scaling of the data and motivation for each simulated center-of-mass energy.

3.2 Baseline model architecture

This thesis aims to use *deep neural networks* to model charged-particle spectra from ALICE measurements. The choice of model architecture is complex, so it is important to establish some basic characteristics of the model before searching for the optimal model architecture. The modeling in this thesis is performed using the deep learning software Keras [32], which acts as an interface for TensorFlow [33], a machine learning library. Keras simplifies the design and training of artificial neural networks, while the computations are handled by TensorFlow.

Since the dataset comprises both N_{ch} distributions and p_{T} spectra, two distinct DNN models sharing a baseline architecture are employed. However, variations in the architecture are allowed to better take the distinctive spectral shapes of each observable into account. The best model structure for each observable is selected based on the results of a hyperparameter scan, which will be discussed in Section 3.4. The DNN model is constructed through the implementation of ensemble learning. This concept will be explained in Subsection 3.5.1.

The baseline architecture for the DNN models consists of a fully connected, *feedforward deep neural network*. The input layer contains two input neurons corresponding to the two input features of each model, which are the center-of-mass energy and either N_{ch} for multiplicity distributions or p_{T} for transverse momentum spectra, as illustrated in Figure 3.2. The baseline architecture for the DNN models also contains

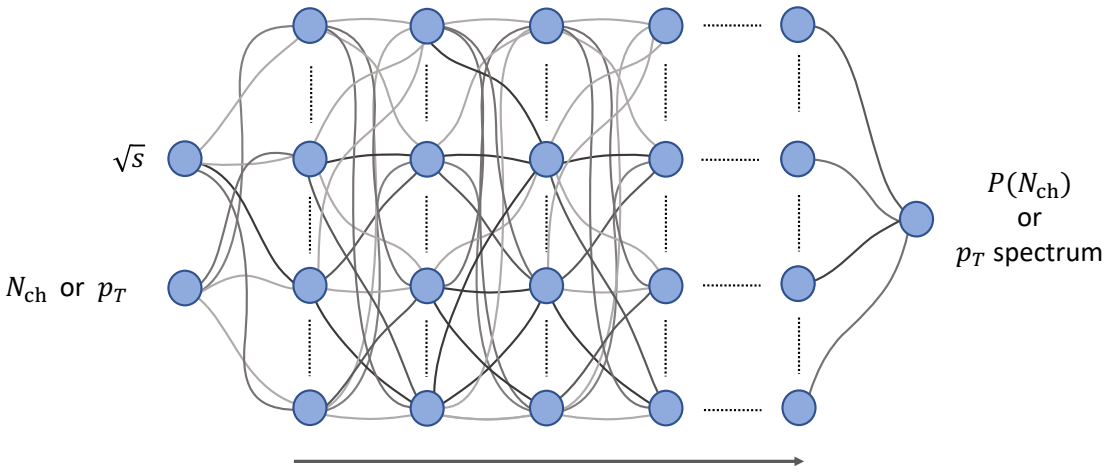


Figure 3.2: Illustration of the baseline architecture for the DNN models.

hidden layers with a nonlinear activation function and a constant number of neurons per layer. The output layer has a linear activation function and consists of a single output neuron. The corresponding output feature is $P(N_{\mathrm{ch}})$ for the multiplicity distribution or, in the case of the transverse momentum spectra, the p_{T}-dependent charged-particle production rate (yield). The weights and biases in each hidden layer, as well as the output layer, are initialized before training. Furthermore, the *elastic net* weight regularizer, introduced in Subsection 2.4.4, is integrated into each hidden layer to limit the model's complexity. The corresponding regularization term to the loss function is a linear combination of both the *L1* and *L2* penalty terms with the regularization factors λ_1 and λ_2, respectively. These factors determine the strength of the regularization.

The adjustment of the model's parameters during training is guided by an optimizer. The chosen loss function is the *Mean Absolute Error* (MAE), which is also the metric selected to evaluate the model performance on the test data in the case of the PYTHIA-trained DNN model. The model's training includes an *early stopping* feature, which monitors the validation loss with a *patience* of 15 epochs. The minimum learning rate that can be reached is set to 10^{-8}. Also, as indicated in Subsection 2.4.5, a batch size corresponding to a power of two (2^x) is selected for the training.

In this thesis, DNN models using both data from ALICE measurements and PYTHIA simulations are trained for N_{ch} distributions and p_{T} spectra. This results in four different DNN models. To maintain clarity, DNN models trained on ALICE or PYTHIA data are referred to as "ALICE DNN models" or "PYTHIA DNN models", respectively. To distinguish the PYTHIA-simulated data from the ALICE measurements,

the former is referred to as "sim" and the latter as "data" in all Figures in this thesis. Furthermore, training, validation and test data are referred to as "train", "val" and "test", respectively. The predictions for the training energies (LHC energies) are called "parametrizations" and those for the chosen test energies are called "extrapolations".

3.3 Data preparation

A crucial step before training the *deep neural network* model is preparing the data to achieve the optimal model performance. This section presents the data preparation methods implemented in the analysis. First, the motivation for scaling the output values ($P(N_{\mathrm{ch}})$ or p_{T}-dependent yield) is discussed, followed by the motivation for scaling the input values ($\sqrt{s}$ and N_{ch} or p_{T}). Additionally, a detailed explanation of the data split into training and validation, the augmentation of the data within the corresponding data uncertainties and the motivation for changing the order of data points before feeding them into the network is provided.

A model architecture is chosen as an example to show the effects that certain data preparation techniques have on the model's performance. The importance of applying these techniques is emphasized by comparing the performance of the model in two scenarios: first, where all methods are applied before training, and second, where one of these methods is omitted in the data preparation process. The comparison is made for two of the preparation methods, showing their influence on the model. In each case, the model is trained on the PYTHIA-simulated p_{T} spectra at $\sqrt{s} = 5.02$ TeV and $\sqrt{s} = 7$ TeV and subsequently tested at $\sqrt{s} = 13.6$ TeV. This model consists of a *deep neural network* with four hidden layers of 64 neurons, the *swish* activation function and a regularization factor of $\lambda_1 = \lambda_2 = 5 \cdot 10^{-5}$. Furthermore, all random seeds are set to a constant value and the batch size is set to $2^6 = 64$. The *Adam* optimizer is used with a maximum learning rate of 0.05. Additionally, random uniform initialization is utilized. In the upper part of Figure 3.3, the p_{T} spectra predicted by the PYTHIA DNN model when all data preparation methods are applied are shown together with the corresponding PYTHIA-simulated data. The simulated p_{T} spectra are depicted by data points with each color representing a different $\sqrt{s}$. The energies used for training are $\sqrt{s} = 5.02$ TeV and 7 TeV. Out of the corresponding spectra, 70% of randomly chosen data points are used for training and 30% for validation. The spectrum at $\sqrt{s} = 13.6$ TeV is used for testing the model's extrapolation performance. The circle markers represent the data used for training, while the diamond markers represent the validation data and the cross markers the test data. The predictions of the PYTHIA DNN model are represented by lines. Solid lines are used for the training energies ($\sqrt{s} = 5.02$ TeV and 7 TeV) and a dashed one is used for the test

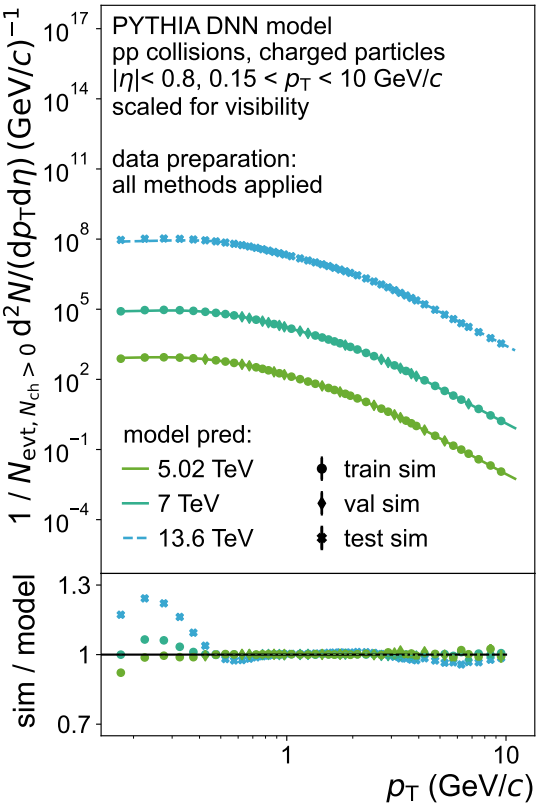


Figure 3.3: p_{T} spectra predicted by the PYTHIA DNN model at $\sqrt{s}$ = 5.02, 7 TeV (training) and 13.6 TeV (test) when all data preparation methods are applied.

energy ($\sqrt{s}$ = 13.6 TeV). As mentioned in Subsection 3.1.2, all spectra are scaled with an $\sqrt{s}$-dependent factor for better visibility. The factor for visual scaling of each $\sqrt{s}$ value is listed in Table 3.1. The lower part of Figure 3.3 shows the ratio between the simulated and the predicted spectra for a given $\sqrt{s}$. Here, it illustrates good agreement between the predicted and simulated spectra at the training energies. Some deviations of up to 30% are seen in the low p_{T} region in the case of the energy extrapolation ($\sqrt{s}$ = 13.6 TeV). However, the remaining p_{T} spectrum shows a similar agreement with the test data as seen for the training energies ($\sqrt{s}$ = 5.02 TeV and 7 TeV).

Wide ranges of values within the training data can, among other issues, lead to vanishing or exploding gradients of the loss function during backpropagation. These effects were discussed in Subsection 2.4.5. Large values are often given a more significant role during training than smaller values, which can make it difficult for the model to learn correlations within the data without being misguided by the differences in scale. Therefore, data scaling prior to training is helpful to achieve a more balanced representation of the data during the learning process. In the following, the scaling of the output values and, later, the input values is discussed.

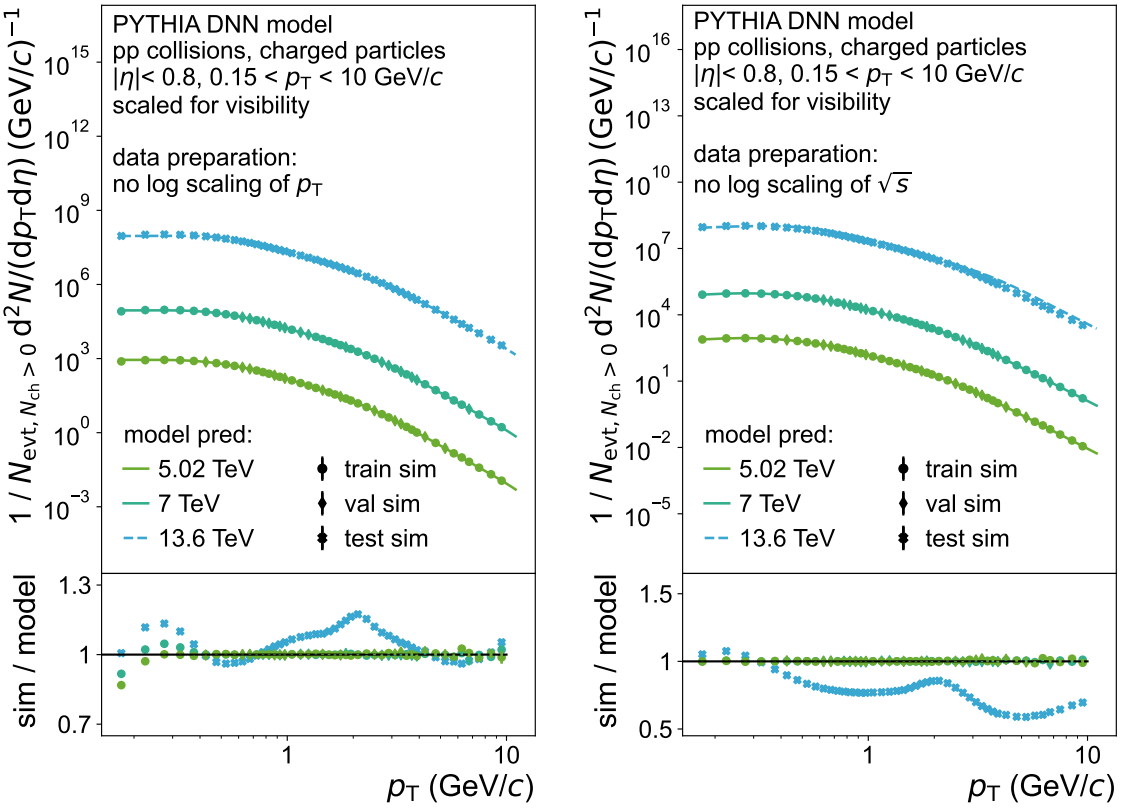


Figure 3.4: p_T spectra predicted by the PYTHIA DNN model at $\sqrt{s}$ = 5.02, 7 TeV (training) and 13.6 TeV (test) in case p_T (left) or $\sqrt{s}$ (right) is not scaled logarithmically before training.

The output values for both the N_{ch} distributions and the p_T spectra span many orders of magnitude. When minimizing the loss function during the training process, reducing large absolute deviations between the model predictions and training data in low N_{ch} (or p_T) regions will be more rewarding than addressing the much smaller deviations in high N_{ch} (or p_T) regions. Therefore, the low N_{ch} (or p_T) regions with the high values dominate the training process. This results in a deteriorating model performance towards high N_{ch} (or p_T). To avoid this effect, the values are scaled logarithmically prior to training. The output of the model becomes $\log[P(N_{ch})]$ for the N_{ch} distributions and $\log[(1/N_{evt,N_{ch}>0}) \cdot d^2N/(dp_T d\eta)]$ for the p_T spectra.

The second aspect to consider is the range of the input values fed into the DNN. Wide ranges of input values can, like in the case of the output values, cause vanishing and exploding gradients of the loss function, which destabilizes the training process. Since N_{ch} as well as $\sqrt{s}$ span ranges of $1 \leq N_{ch} \leq 100$ and 0.2 TeV $\leq \sqrt{s} \leq 100$ TeV, these input features are also scaled logarithmically. The p_T intervals in the p_T spectra have logarithmic widths. The resulting higher data point density towards low p_T values poses an additional bias for the model. This can be rectified by the logarithmic

scaling of the p_{T} values, as well. The effects of logarithmic scaling can be seen by comparing the performance of the PYTHIA DNN model introduced in the previous section when all data preparation methods are applied with the performance when no scaling of the input values is performed before training. In Figure 3.4, the p_{T} spectra predicted by the PYTHIA DNN model are shown together with the corresponding PYTHIA-simulated spectra for the case when p_{T} (left) or $\sqrt{s}$ (right) are not scaled logarithmically prior to training. The extrapolation performance of the DNN model is represented by the ratio between the simulated data and the predictions, which is worse than the one shown in Figure 3.3 where all data preparation methods are applied. When the p_{T} values are not scaled before training, the extrapolation performance in the mid-p_{T} region deteriorates significantly. In the case that the $\sqrt{s}$ values are not scaled prior to training, the extrapolated spectrum shows large deviations from the simulated spectrum at $\sqrt{s} = 13.6$ TeV of up to 50%. These observations highlight the importance of scaling the input values as a data preparation method before training a DNN model. In summary, the input of the model for each dataset becomes $(\log[p_{\mathrm{T}}], \log[\sqrt{s}])$ and $(\log[P(N_{\mathrm{ch}})], \log[\sqrt{s}])$, respectively.

A common method discussed in Subsection 2.4.4 to improve the model consists of splitting the data into subsets of the original dataset for training and validation, respectively. In this thesis, the original dataset is randomly split into 70% training and 30% validation data. In order to correctly describe the edges of the spectra, the data points corresponding to the lowest and highest N_{ch} (or p_{T}) values are always included in the training set.

The uncertainties in the data imply that the respective data points have a certain freedom to exist within the error bounds. If the model was trained only with the nominal values, these uncertainties would not be taken into account. Therefore, numerous artificial data points are sampled within these error bounds. Since the uncertainties of the ALICE data are systematic, the artificial data points are sampled out of a uniform distribution within the uncertainty of the original data point. The uncertainties of the PYTHIA-simulated data are statistical. Therefore they are sampled from a Gaussian distribution with a width corresponding to the error bounds.

To avoid any potential bias in the neural network caused by the ordering of the data during the training process, the data points are shuffled prior to training. This introduces randomness into the training process and therefore improves generalization, as the model does not memorize patterns specific to the original order of the data.

sampling: discrete values				
optimizer	layers	neurons	activation	initializer
Adam (A)	2	32	*ReLU* (RE)	*RandomNormal* (RN)
Nadam (N)	3	64	*SELU* (SE)	*RandomUniform* (RU)
	4	128	*swish* (SW)	*GlorotNormal* (GN)
		256	*softplus* (SP)	*TruncatedNormal* (TN)
				VarianceScaling (VS)

sampling: intervals (logarithmic)			
	λ_1	λ_2	learning rate
min	$5 \cdot 10^{-8}$	$5 \cdot 10^{-8}$	$1 \cdot 10^{-4}$
max	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$1 \cdot 10^{-2}$

Table 3.2: Hyperparameter space for the search.

3.4 Hyperparameter scan

The power of DNN models lies in their ability to recognize underlying patterns in the training data. Therefore, models with a sufficient degree of complexity are expected to generate similar predictions for the same training set, independent of their specific architectures. However, when the models venture beyond the value range of the training set they lack the necessary constraints to provide consistent predictions. In this extrapolation regime, the models have a greater freedom to generate predictions, which strongly depend on the chosen model architecture. One of the primary objectives of this thesis is to develop a robust DNN model capable of producing accurate extrapolations to unexplored center-of-mass energy regimes. Therefore, it is important to evaluate the extrapolation performance for a given model architecture at these $\sqrt{s}$ regimes far away from the training data. As the ALICE measurements used in this thesis are only available at five LHC energies, data from PYTHIA simulations is utilized for this purpose. The simulated data, described in Subsection 3.1.2, spans a wide energy range of up to $\sqrt{s} = 100$ TeV. The extrapolation performance of a given model architecture is assessed through its consistency with the simulated test data. The extrapolation performance is expected to deteriorate with increasing distance between the extrapolation energy and those used for training. Therefore, the search for the best set of hyperparameters focuses on finding a model architecture with good extrapolation capability. To accomplish this, a hyperparameter optimization framework for Keras, called KerasTuner [32], is used. This framework offers a systematic approach to identifying the optimal hyperparameters for a neural network model by exploring the hyperparameter space. The selected strategy for the hyperparameter scan is Bayesian Optimization. In this thesis, an increased β value of 3.6 (default: 2.6) is used to ex-

obs.	lay.	nod.	opt.	lr	act.	init.	λ_1	λ_2
$\mathrm{P}(N_{\mathrm{ch}})$	2	32	A	$1.16 \cdot 10^{-3}$	SW	RU	$4.26 \cdot 10^{-4}$	$2.08 \cdot 10^{-6}$
p_{T} spectrum	4	64	A	$4.61 \cdot 10^{-3}$	SP	TN	$9.05 \cdot 10^{-7}$	$6.55 \cdot 10^{-6}$

Table 3.3: Top-performing model architectures from the hyperparameter scans. More well-performing architectures of the scans are listed in Table A.1 of the Appendix.

plore the hyperparameter space more thoroughly. The possible values to explore for a given hyperparameter can be either discrete or sampled from a specific range of values. Logarithmic sampling is typically used for hyperparameters with possible values that span many orders of magnitude. Therefore, it is implemented for the learning rate as well as the regularization factors λ_1 and λ_2. Predefined sets of values are chosen for the optimizer, the number of layers, the number of neurons, the activation function and weight initialization. The search space explored during the hyperparameter scan is presented in Table 3.2.

During the scan, multiple models are trained and evaluated using different hyperparameter combinations. To evaluate the performance of each model, the MAE is calculated for the test dataset. Most of the datasets simulated for this thesis were motivated by center-of-mass energies from (possible) collider experiments, as summarized in Table 3.1. However, this set of energies is biased towards high energies, so that the MAE values from the scan are dominated by them. Therefore, PYTHIA-simulated data at $\sqrt{s} = 0.5$ TeV and 1.5 TeV is added to the test dataset to provide a more equal representation of low and high center-of-mass energies in the scan results.

The scan consists of two hundred trials, representing the number of hyperparameter sets or combinations to be evaluated. Each trial is executed three times and the resulting MAE value is calculated as the average of the MAE values of all three iterations. This is done to obtain a more reliable performance estimation for each hyperparameter configuration since the weights and biases of the network are initialized randomly at the beginning of training. As a direct consequence, models that generate stable predictions despite random variations within the network receive lower MAE values compared to unstable models. This aspect plays a crucial role in estimating the systematic model uncertainties caused by inherent randomness within the network, as will be discussed in the next section. Since the hyperparameter scan involves training two hundred model architectures with three different executions each, performing the hyperparameter scan is computationally expensive, especially if each of the models is trained with the full number of epochs. To address this issue, an *early stopping* feature is implemented not only to prevent overfitting but also to avoid spending computing resources on non-converging models.

Two independent hyperparameter scans are conducted for the N_{ch} distributions and

p_{T} spectra to identify the optimal hyperparameter configurations for each observable. Table A.1 in the Appendix provides an overview of the ten top-performing model configurations out of the two hundred trials conducted for each observable. These configurations represent the hyperparameter values that resulted in the lowest MAE values, indicating superior predictive capabilities for the specific observable. From the evaluated models, the hyperparameter configurations with the lowest MAE values are adopted as the primary DNN models and are listed in Table 3.3.

3.5 Systematic uncertainties

While the DNN model hyperparameters are tuned to provide predictions, especially extrapolations, that align well with the PYTHIA-simulated test data, it is important to identify the potential sources of uncertainty. These uncertainties quantify the reliability and limitations of the models.

In this section, the systematic uncertainties associated with the DNN models are discussed. Two potential sources of uncertainty are presented. The corresponding systematic uncertainties are referred to as ensemble uncertainties and hyperparameter uncertainties, respectively. They will be discussed in detail in the following subsections.

3.5.1 Ensemble uncertainties

This section focuses on the systematic uncertainties caused by the intrinsic randomness of the DNN model, its training process and the training data selection. These uncertainties are collectively referred to as ensemble uncertainties.

A random seed determines the starting point for the generation of pseudorandom numbers during the training process. When no specific random seed is set, each training instance is initialized with different starting conditions, leading to different model parameters and, with that, different predictions. The corresponding uncertainties are estimated by systematically varying the random seed within a so-called ensemble. This ensemble is a collection of many models with the same architecture but trained with different random seeds. The ensemble uncertainties can then be inferred from the spread of the individual predictions of each model compared to the mean ensemble predictions.

One source of ensemble uncertainty is the random initialization of the model's parameters defining the initial conditions of the model before being trained on the data. As the loss function depends on these parameters, their random initialization also defines the starting point of the loss function in the parameter space. This could result in finding a different minimum of the loss function during training. Furthermore,

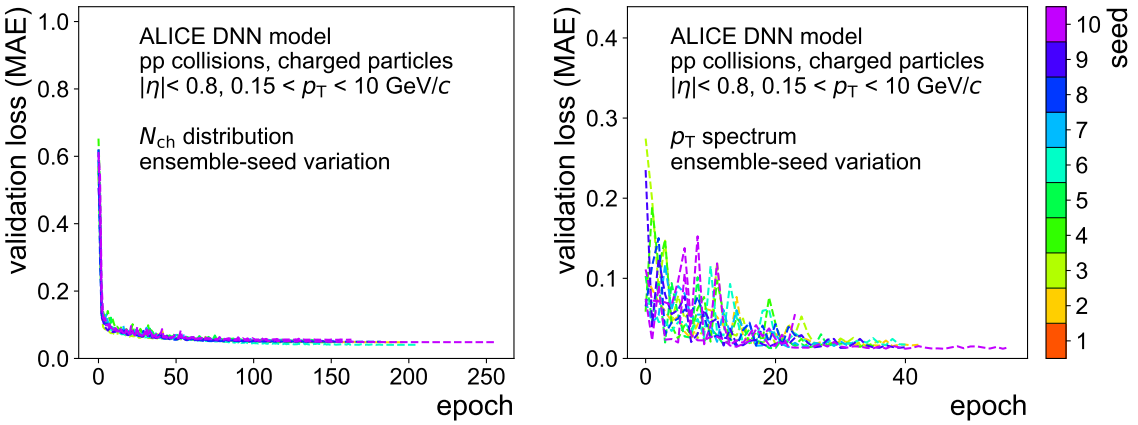


Figure 3.5: Evolution of the validation loss functions of the ALICE DNN models with different ensemble seeds for the N_{ch} distributions (left) and p_{T} spectra (right).

the random splitting of the data into training and validation sets affects the training outcome.

In this thesis, an ensemble consisting of ten models is trained with different random seeds. The validation loss functions of the ALICE DNN models for different ensemble seeds are shown in Figure 3.5 as a function of the number of training epochs for the N_{ch} distributions (left) and p_{T} spectra (right). The validation loss functions of the PYTHIA DNN models are shown in Figure A.1 in the Appendix. Each curve in the Figures represents the loss function of a model with a specific ensemble seed. It is important to mention that the validation data only plays a role in the decision of when to stop the training process and avoid overfitting. The model parameters are solely adjusted based on the training data. After the initial fluctuations of the validation loss functions, they converge to similar values. This shows that the predictions of the DNN models for the validation data are stable when confronted with variations of the ensemble seed. Furthermore, the ensemble models for the N_{ch} distributions are trained for a larger number of epochs than the models for the p_{T} spectra. This is a result of the more pronounced changes in the shape of the N_{ch} distributions as a function of the center-of-mass energy compared to the p_{T} spectra.

The training of the ensemble results in ten different predictions for each data point. In this thesis, the nominal predictions of each DNN model represent the mean predictions of the corresponding ensemble. The ensemble uncertainty ($\sigma_{\mathrm{ensemble}}$) of the DNN models is calculated as the standard deviation of the predictions of all models in the ensemble. The relative deviation between the individual predictions and the mean ensemble prediction for the N_{ch} distributions (left) and p_{T} spectra (right) of the ALICE DNN models is illustrated in Figure 3.6. The corresponding results for the PYTHIA DNN models are shown in Figure A.3 in the Appendix. The predictions

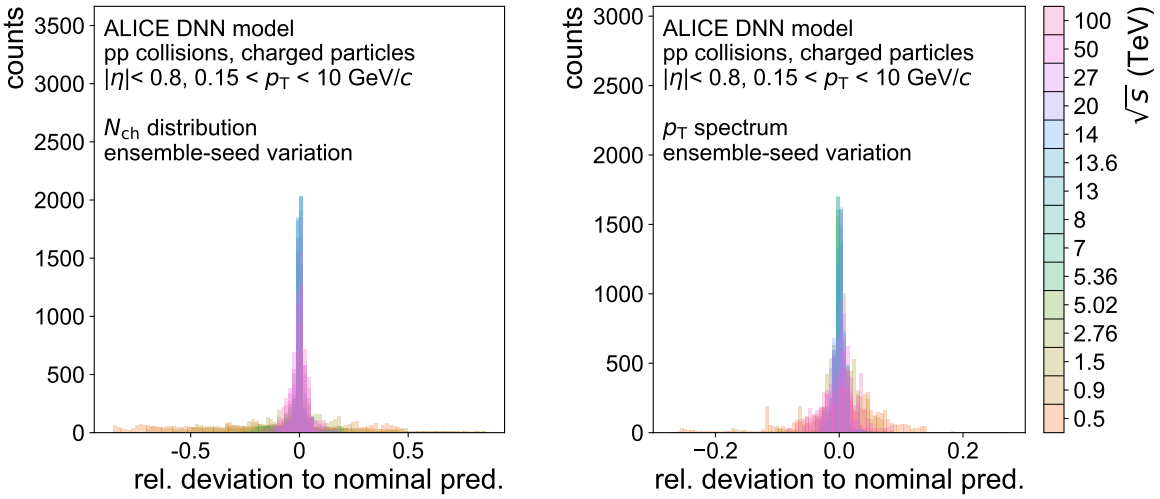


Figure 3.6: Relative deviation between ALICE DNN models trained with different random seeds and the ALICE DNN nominal prediction for the N_{ch} distributions (left) and p_{T} spectra (right).

are made for 500 uniformly distributed data points per energy with a multiplicity range of $1 \leq N_{\mathrm{ch}} \leq 100$ for the N_{ch} distributions and a transverse momentum range of 0.15 GeV/c $< p_{\mathrm{T}} <$ 10 GeV/c for the p_{T} spectra. Figure 3.6 shows that the majority of the predictions are clustered closely around the nominal prediction, demonstrating that the models are very stable against randomness. The N_{ch} distributions exhibit some outliers with a relative deviation of up to 75%, in particular at low energies, which is a direct consequence of the shorter N_{ch} ranges present at lower center-of-mass energies. This results in the models having more freedom in their predictions for the high N_{ch} regions of lower $\sqrt{s}$, where no test data constrains the choice of the model hyperparameters. For the p_{T} spectra, the predictions by the ensemble models are more consistent than those for the N_{ch} distributions and lie typically within 10%.

3.5.2 Hyperparameter uncertainties

The exploration of different hyperparameter combinations during the hyperparameter scan represents an important source of systematic uncertainty. The primary goal of the scan is to identify sets of hyperparameters resulting in strong extrapolation capabilities of the model, evaluated on the PYTHIA-simulated data, as discussed in Section 3.4. The MAE values of the top-performing model architectures evaluated on the test data are shown in Table A.1 in the Appendix. Despite being trained on the same dataset, the top-performing models from the scan still demonstrate variations in their MAE values. The variations of the ten top-performing models from the scan are considered to estimate the uncertainties from the choice of the model architec-

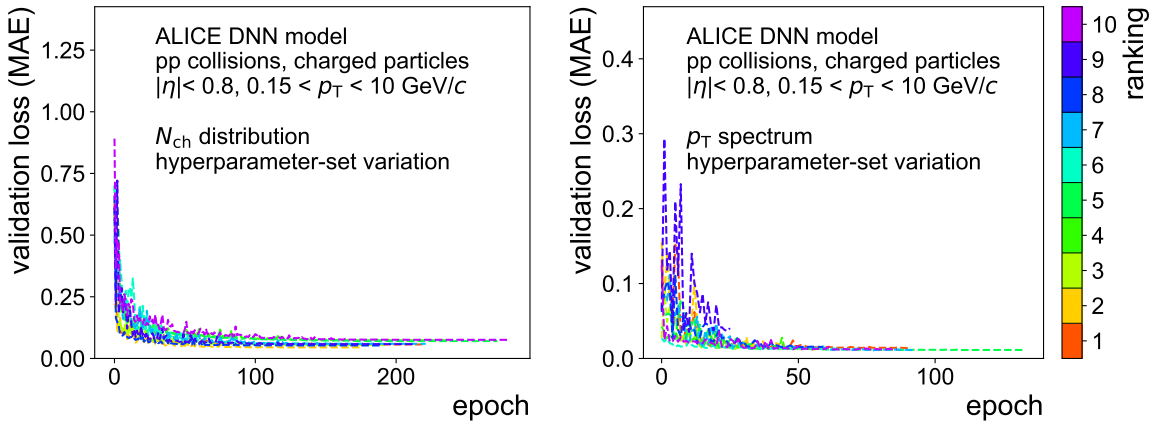


Figure 3.7: Evolution of the validation loss functions of the ALICE DNN models with different hyperparameter sets for the N_{ch} distributions (left) and p_{T} spectra (right).

ture. Each of these models is trained with a fixed random seed to avoid the inherent randomness of the model that is already considered in the ensemble uncertainties. The resulting validation loss functions for each of the ALICE-trained models with the different hyperparameter sets of the top-performing architectures from the scan are shown in Figure 3.7 for N_{ch} distributions (left) and p_{T} spectra (right). The loss functions corresponding to the PYTHIA-trained models are illustrated in Figure A.2 in the Appendix. The evolution of the validation loss as a function of the training epochs shows a similar trend as the validation loss functions from the ensemble in Figure 3.5.

The hyperparameter uncertainty ($\sigma_{\mathrm{hparams}}$) of the DNN models is calculated as the mean absolute deviation between the individual predictions by each of the considered models with different architectures according to Table A.1 and the model with the top-performing architecture from the hyperparameter scan. In Figure 3.8, the relative deviation between the predictions of each model architecture and the best-performing model architecture is depicted for the N_{ch} distributions (left) and p_{T} spectra (right) of the ALICE DNN models. The relative deviation of the PYTHIA DNN models is illustrated in Figure A.4 in the Appendix. The Figures are analogous to the ones in Figure 3.6 and Figure A.3 for the ensemble deviations. However, in this case, the deviations are not calculated relative to the nominal predictions, but rather relative to the predictions by the model with the best-performing model architecture trained with a fixed random seed. All models describe the training data very well. The relative deviations reveal larger discrepancies between the predictions for energies with a large distance from the training energy range, like $\sqrt{s}$ = 0.5 and 100 TeV. In the case of the N_{ch} distributions, the predictions show discrepancies of up to 50% and, for the p_{T} spectra, over 20%. This demonstrates that extrapolations to both lower and higher energies are more sensitive to changes in the model architecture than those close to

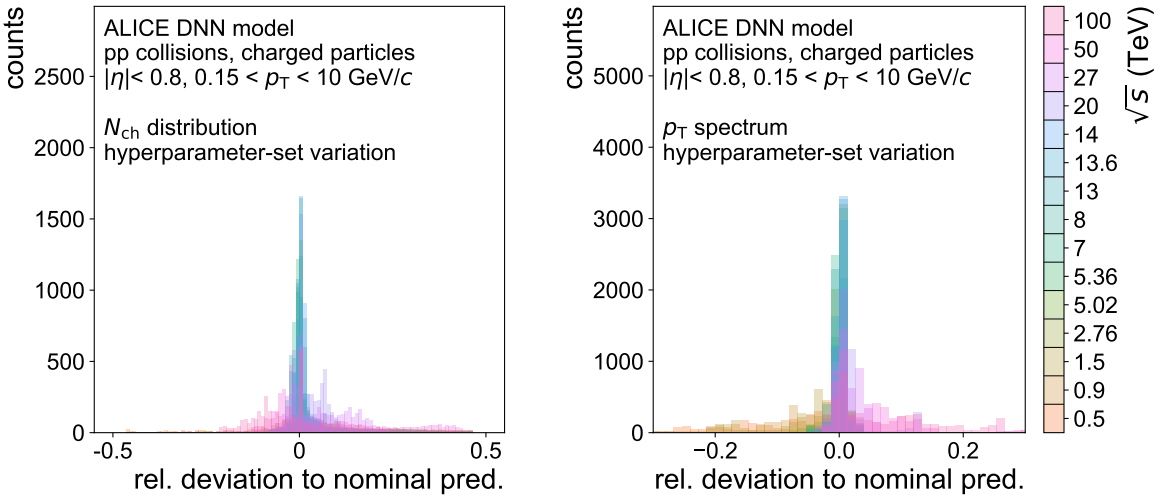


Figure 3.8: Relative deviation between ALICE DNN models with different hyperparameter sets and the ALICE DNN nominal prediction for the N_{ch} distributions (left) and p_{T} spectra (right).

the training energy range. Furthermore, the relative deviations from different hyperparameter sets are wider than those from the different ensemble models. Therefore, the choice of the model architecture is the dominating source of uncertainty.

3.5.3 Total uncertainties

Both the ensemble and the hyperparameter uncertainties contribute to the total systematic uncertainty of the DNN models. For each predicted point it is calculated as the root of the square sum over the ensemble and hyperparameter uncertainties:

$$\sigma_{\mathrm{total}} = \sqrt{\sigma^2_{\mathrm{ensemble}} + \sigma^2_{\mathrm{hparams}}} \quad . \tag{3.1}$$

The relative model uncertainty of the ALICE DNN models is illustrated in Figure 3.9 for the N_{ch} distributions (left) and p_{T} spectra (right) within an energy range of $\sqrt{s}$ = 0.5 TeV to 100 TeV. The relative model uncertainty of the PYTHIA DNN models is depicted in Figure A.5 in the Appendix. The solid lines represent the relative uncertainties of the parametrizations at LHC energies, while the dashed lines represent the relative uncertainties of the $\sqrt{s}$ extrapolations. Furthermore, each center-of-mass energy is indicated by a different color. In Figure 3.9, the relative model uncertainties exhibit a clear $\sqrt{s}$ dependence across the entire N_{ch} and p_{T} ranges. This dependence is generally characterized by a larger uncertainty for extrapolation energies that lie further away from the training energy range (LHC energies). In the case of the multiplicity distributions, the N_{ch} range of the available data varies with different $\sqrt{s}$ values. Since lower collision energies have a smaller reach in multiplicity, the model

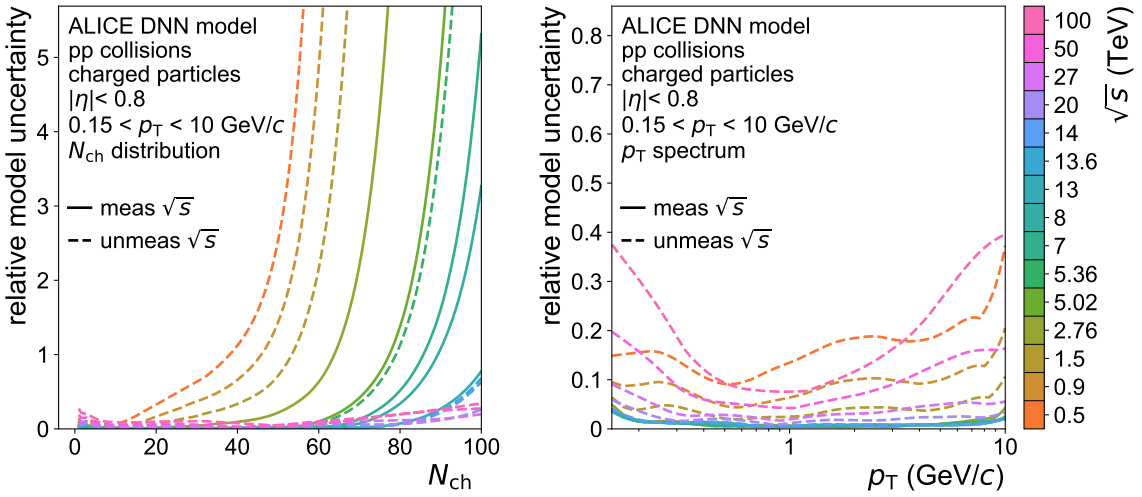


Figure 3.9: Relative total systematic model uncertainty for N_{ch} distributions (left) and p_{T} spectra (right) predicted by the ALICE DNN models.

predictions are less constrained for low-energy extrapolations toward high multiplicity. As a result, the relative model uncertainty gets significantly larger. It is worth mentioning that $\mathrm{P}(N_{\mathrm{ch}})$ approaches zero in high-N_{ch} regions, so their large uncertainty is less relevant. High-energy extrapolations, where the larger N_{ch} ranges offer more constraints to the model predictions, have the smallest uncertainties of all energies in the high-N_{ch} region. In the case of the p_{T} spectra, the p_{T} range stays constant over all the center-of-mass energies. Here, the relative model uncertainties are very small for the energies used in the training. The uncertainties become gradually larger for the $\sqrt{s}$ extrapolations, with the highest values at $\sqrt{s} = 0.5$ and 100 TeV. Overall, the relative model uncertainties of the p_{T} spectra predicted by the ALICE DNN model are lowest in the mid-p_{T} region, since the predictions are constrained by data from both low and high-p_{T} regions. However, the uncertainties increase in these two regions, where the models lose their constraints, as no more data points are available outside the p_{T} range of the measurement. Both for the N_{ch} distributions and p_{T} spectra, the ALICE DNN model uncertainties are slightly larger than those of the PYTHIA DNN model, which is expected since the model architectures were chosen based on their performance on PYTHIA simulations.

Chapter 4

Results

This chapter presents and discusses the outcomes of the analysis described in this thesis. First, the parametrizations and extrapolations of the N_{ch} distributions and p_{T} spectra by the PYTHIA and ALICE DNN models are presented for an energy range of $\sqrt{s}$ = 0.5 TeV to 100 TeV. The predictions by the PYTHIA DNN models serve as a crucial quality assurance for the ALICE DNN models since they demonstrate the extrapolation capabilities of the chosen model architectures. Subsequently, the DNN interpolation of the p_{T} spectrum is compared to a published power-law interpolation. Additionally, the limits of the low-energy extrapolation for the PYTHIA DNN models are explored. Then, the mean multiplicity ($\langle N_{\mathrm{ch}} \rangle$) and mean transverse momentum ($\langle p_{\mathrm{T}} \rangle$) obtained from the model predictions are presented as a function of $\sqrt{s}$ and compared to those directly derived from the PYTHIA simulation and ALICE data. Furthermore, the $\sqrt{s}$ dependence of the predicted $\langle N_{\mathrm{ch}} \rangle$ is compared to a power-law parametrization of the ALICE data similar to the ones shown in Figure 2.5. Finally, the parametrizations and extrapolations by the ALICE DNN model are compared to the PYTHIA-simulated data.

4.1 N_{ch} distributions

The upper panel of Figure 4.1 (left) shows N_{ch} distributions from PYTHIA simulations and those predicted by the PYTHIA DNN model. Each color in the figure represents a different center-of-mass energy. The circles represent the training data corresponding to the LHC energies and the crosses indicate the test data corresponding to the chosen test energies ranging from $\sqrt{s}$ = 0.5 TeV to 100 TeV. In contrast to the Figures in Section 3.3, there are no dedicated validation data points because each of the ensemble models used to calculate the nominal predictions of the PYTHIA DNN model is trained with different training-validation partitions of the original dataset. As a result, most data points are used as validation data at least once during the training of the ensemble.

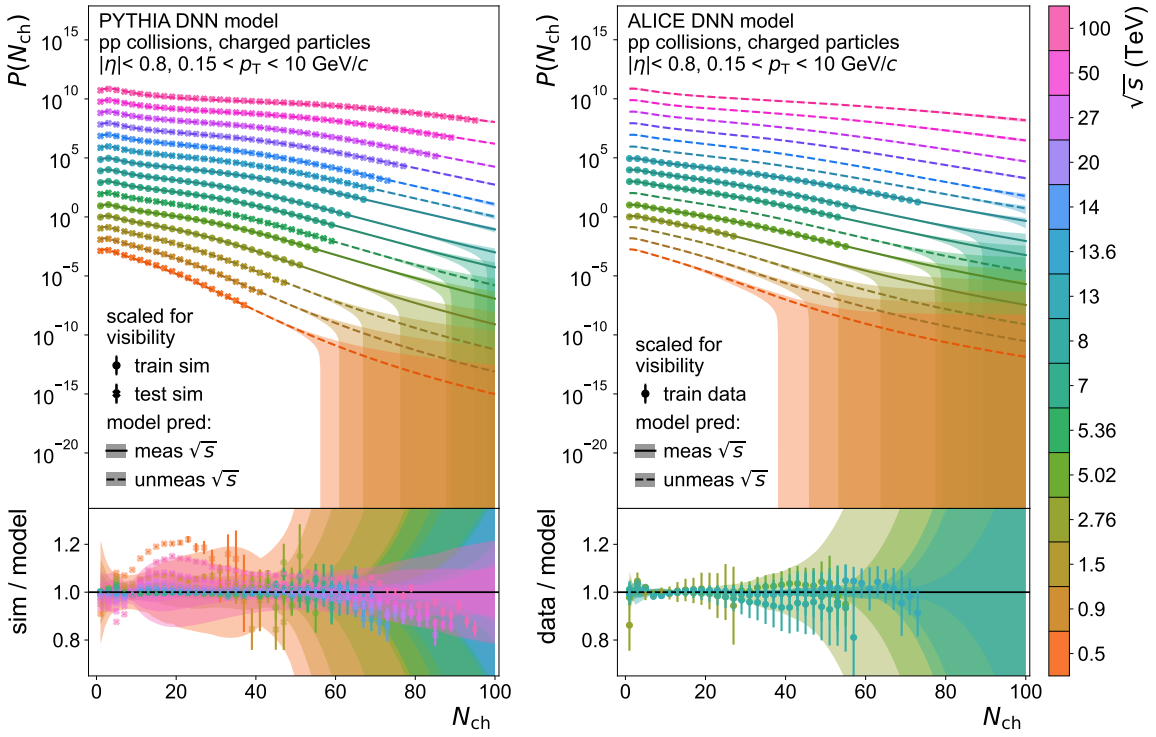


Figure 4.1: Parametrizations and extrapolations of N_{ch} distributions by the PYTHIA DNN model (left) and ALICE DNN model (right) at LHC energies ("meas $\sqrt{s}$ ") and chosen test energies ("unmeas $\sqrt{s}$ ").

The PYTHIA DNN model predictions are depicted as lines. Solid lines represent the parametrizations of the training data at LHC energies, while dashed lines represent the extrapolations to the chosen test energies. The predicted distributions for each center-of-mass energy encompass a multiplicity range of $1 \leq N_{\mathrm{ch}} \leq 100$. The systematic uncertainties of the DNN model are depicted as error bands and the uncertainties related to the data are presented as vertical error bars. The lower panel of Figure 4.1 (left) shows the ratio between the PYTHIA-simulated data at a given $\sqrt{s}$ and the corresponding model prediction. As mentioned in Subsection 3.1.2, all data points and predictions are scaled for visibility according to the visual scaling factors listed in Table 3.1. Furthermore, only simulated data points with odd N_{ch} values are shown for better visualization.

By simulating pp collisions at the chosen test energies using PYTHIA, a test dataset is provided to compare to the predictions by the DNN model at these unmeasured energies. Quantifying the deviation between the test data and the corresponding extrapolations allows for evaluating the extrapolation quality of the DNN model. The prediction accuracy of the model for different energies can be observed in the ratio between the simulated data and model predictions. A clearer overview of the individual ratios for each energy is shown in Figure 4.2. The ratio between the N_{ch}

distributions predicted by the PYTHIA DNN model and those simulated by PYTHIA demonstrates excellent parametrizations of the LHC energies and extrapolations to the test energies. At LHC energies, the majority of the predictions display deviations of less than 2% from the training data. Notably, the model extrapolations to the energies $\sqrt{s}$ = 13.6, 14 and 20 TeV are just as accurate as the interpolation to $\sqrt{s}$ = 5.36 TeV. The extrapolations to $\sqrt{s}$ = 0.5 and 100 TeV exhibit deviations of 25% and 15%, respectively. These are remarkable results considering that the DNN model was trained on five LHC energies within a range of $\sqrt{s}$ = 2.76 TeV to 13 TeV. The systematic uncertainties of the model are larger for high multiplicities, especially for low $\sqrt{s}$. As discussed in Subsection 3.5.3, this effect is caused by the different N_{ch} ranges of each distribution. Furthermore, the individual ratios in Figure 4.2 show that the deviations between data and predictions are mostly covered by the model uncertainties across the whole energy range. This demonstrates that the methods employed to estimate the model uncertainties successfully describe possible deviations from actual data for the PYTHIA DNN model. Therefore, it is assumed that the corresponding model uncertainties for the ALICE DNN model also describe possible discrepancies between the predictions and reality beyond the LHC energies used for training. The N_{ch} distributions predicted by the ALICE DNN model and those measured by ALICE are shown in the upper panel of Figure 4.1 (right). The representation of colors, data points, parametrizations, extrapolations and uncertainties is analogous to that of Figure 4.1 (left). Data points are shown for the five LHC energies available from ALICE measurements. These correspond to the training data for the ALICE DNN model, indicated by circles. The predictions are shown within a range of $\sqrt{s}$ = 0.5 TeV to 100 TeV. The parametrizations of the ALICE data are found to be excellent. The ratio in the lower panel indicates that they are fully consistent with unity within the systematic uncertainties of the data and the model. As in the case of the PYTHIA DNN model, the systematic model uncertainties are larger for high multiplicities due to the different N_{ch} ranges. However, the predictions by the ALICE DNN model show larger uncertainties compared to the PYTHIA DNN model. Here, it must be considered that the model hyperparameters are tuned to PYTHIA. Furthermore, the N_{ch} distribution at $\sqrt{s}$ = 2.76 TeV spans a much smaller N_{ch} range compared to PYTHIA ($N_{\mathrm{ch}} \leq 28$ vs. $N_{\mathrm{ch}} \leq 52$). Therefore, the ALICE DNN model lacks some constraints in the N_{ch} dimension. Nonetheless, the extrapolations exhibit a high degree of stability.

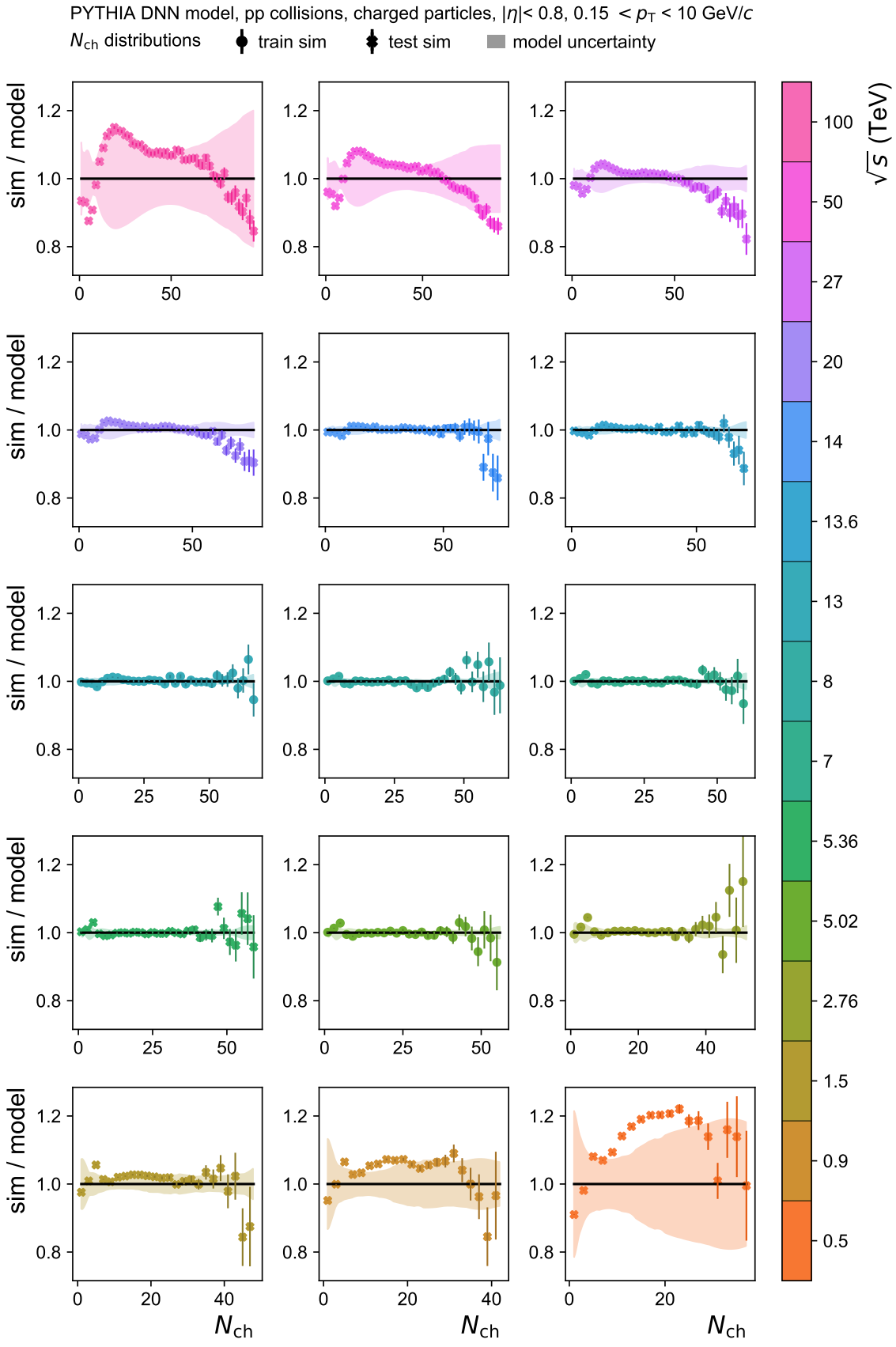


Figure 4.2: Ratio of the N_{ch} distributions predicted by the PYTHIA DNN model and those from PYTHIA simulations for all considered center-of-mass energies.

The individual ratios in Figure 4.2 show that the performance of the PYTHIA DNN model deteriorates for center-of-mass energies far away from the training energies. To study the center-of-mass energy dependence of the model performance, the evaluation metric MAE between the predictions by the PYTHIA DNN model and the PYTHIA-simulated data is calculated separately for each of the considered center-of-mass energies. A small MAE value represents a better model performance. It is expected that the model predictions are less accurate with increasing distance from the training energies. Therefore, the dependence of the MAE values from the distance to the training energies is studied. For this purpose, the center value within the set of five LHC energies, $\sqrt{s} = 7$ TeV, is chosen as a reference. Since the center-of-mass energy values are scaled logarithmically for training the model, the logarithmic distance between them is considered: $\log(\sqrt{s}) - \log(7\text{ TeV})$. In Figure 4.3 (left), the resulting MAE values for the predicted N_{ch} distributions by the PYTHIA DNN model based on the simulated test data are shown as a function of this logarithmic distance to $\sqrt{s} = 7$ TeV. As before, circles represent training energies and crosses indicate test energies. Furthermore, the data points are color-coded depending on $\sqrt{s}$. A clear correlation between the logarithmic distance of a $\sqrt{s}$ value to the training energy range and the predictive power of the model can be observed. However, some MAE values seem to fluctuate so that the energy dependence deviates from a smooth trend. Since a different N_{ch} range is selected for each of the simulated N_{ch} distributions, the number of statistical fluctuations can vary between the distributions of neighboring energies, slightly affecting the MAE values. To test this hypothesis, the MAE of the different $\sqrt{s}$ can be calculated for a constant N_{ch} range of $1 \leq N_{\text{ch}} \leq 37$, within which all considered N_{ch} distributions exhibit almost no fluctuations. The resulting MAE values are shown in Figure 4.3 (right) and demonstrate a smooth, largely symmetrical trend in their energy dependence. The PYTHIA and ALICE DNN models provide the most accurate parametrization of the training data at $\sqrt{s} = 7$ TeV. As the center value within the set of five LHC energies, the N_{ch} distribution at $\sqrt{s} = 7$ TeV is constrained by two distributions below ($\sqrt{s} = 2.76$, 5.02 TeV) and two distributions above this energy ($\sqrt{s} = 8$, 13 TeV), respectively. Therefore, the models receive the most constraints at this energy.

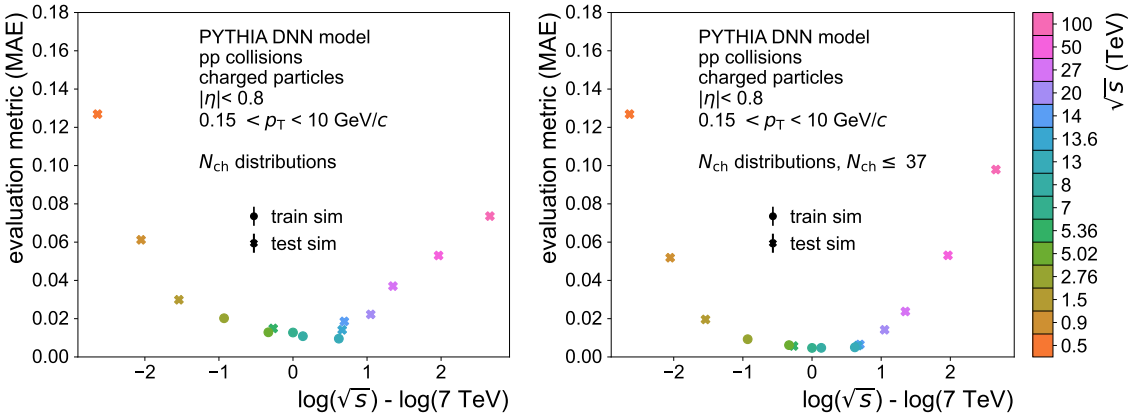


Figure 4.3: MAE values of the comparison between N_{ch} distributions predicted by the PYTHIA DNN model and those from PYTHIA simulations within the N_{ch} ranges in Table 3.1 (left) and within a range of $1 \leq N_{\mathrm{ch}} \leq 37$ for all distributions (right) as a function of the logarithmic distance between the considered energies and $\sqrt{s} = 7$ TeV.

4.2 p_{T} spectra

The resulting parametrizations and extrapolations of the p_{T} spectra by the PYTHIA DNN model (left) and ALICE DNN model (right) are presented in the upper panels of Figure 4.4. In the lower panels, the ratio between data and model predictions is depicted. The predicted p_{T} spectra enclose a range of $0.15 < p_{\mathrm{T}} < 10$ GeV/c and an $\sqrt{s}$ range of 0.5 to 100 TeV. The representation of the predictions, data and uncertainties is the same as in previous Figures. The individual ratios between the p_{T} spectra predicted by the PYTHIA DNN model and those simulated by PYTHIA shown in Figure 4.5 demonstrate excellent parametrizations and extrapolations of the training data. The predictions ranging from $\sqrt{s} = 2.76$ up to 20 TeV, are in strong agreement with the corresponding training and test data, with deviations largely within 2%. This shows that the energy interpolation and extrapolation close to the training energy range demonstrate a similar quality as the parametrizations of the training data. The extrapolation to $\sqrt{s} = 0.5$ and 100 TeV show deviations of 18% and 10%, respectively, demonstrating to be more accurate than the energy extrapolation of the N_{ch} distributions. The deviations in the ratio between the data and the predictions are mostly covered by the model uncertainties. This proves the effectiveness of the employed uncertainty estimation methods. Therefore, it is reasonable to assume that the uncertainties of the ALICE DNN model for the predicted p_{T} spectra span the possible deviations between predictions and real data at unmeasured center-of-mass energies, which builds confidence regarding the interpolations and extrapolations performed by the DNN model. The resulting parametrizations and extrapolations by the ALICE DNN model are presented in Figure 4.4 (right). For all available LHC

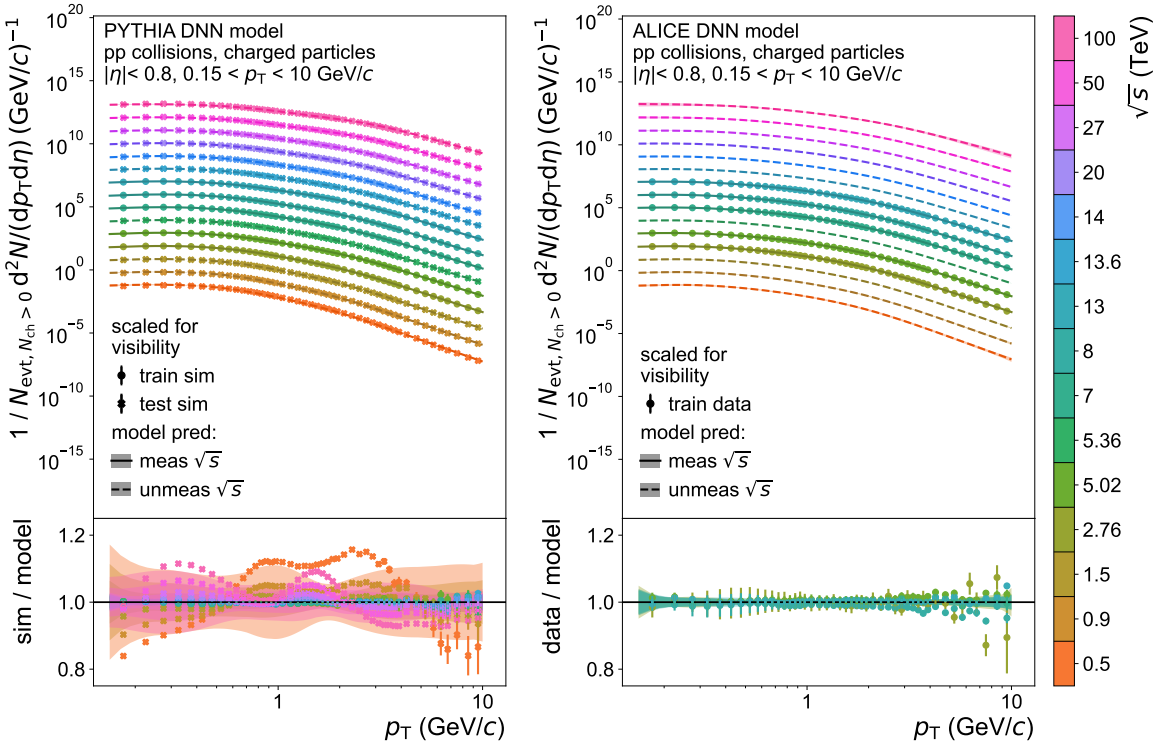


Figure 4.4: Parametrizations and extrapolations of p_{T} spectra predicted by the PYTHIA DNN model (left) and ALICE DNN model (right) at LHC energies ("meas $\sqrt{s}$ ") and chosen test energies ("unmeas $\sqrt{s}$ ").

energies, the ratio between the data and the model predictions is shown to be largely consistent with unity within the systematic uncertainties. These results demonstrate a high consistency of the parametrization with the training data.

The energy dependence of the MAE values of the PYTHIA DNN model can also be compared for the predicted p_{T} spectra when evaluated on the PYTHIA-simulated data. These MAE values are shown in Figure A.6 in the Appendix. As observed for the N_{ch} distributions in Figure 4.3, the model performance clearly depends on the logarithmic distance of $\sqrt{s}$ to 7 TeV. The model performance is best at $\sqrt{s} = 7$ TeV and gradually worsens with increasing logarithmic distance to this energy.

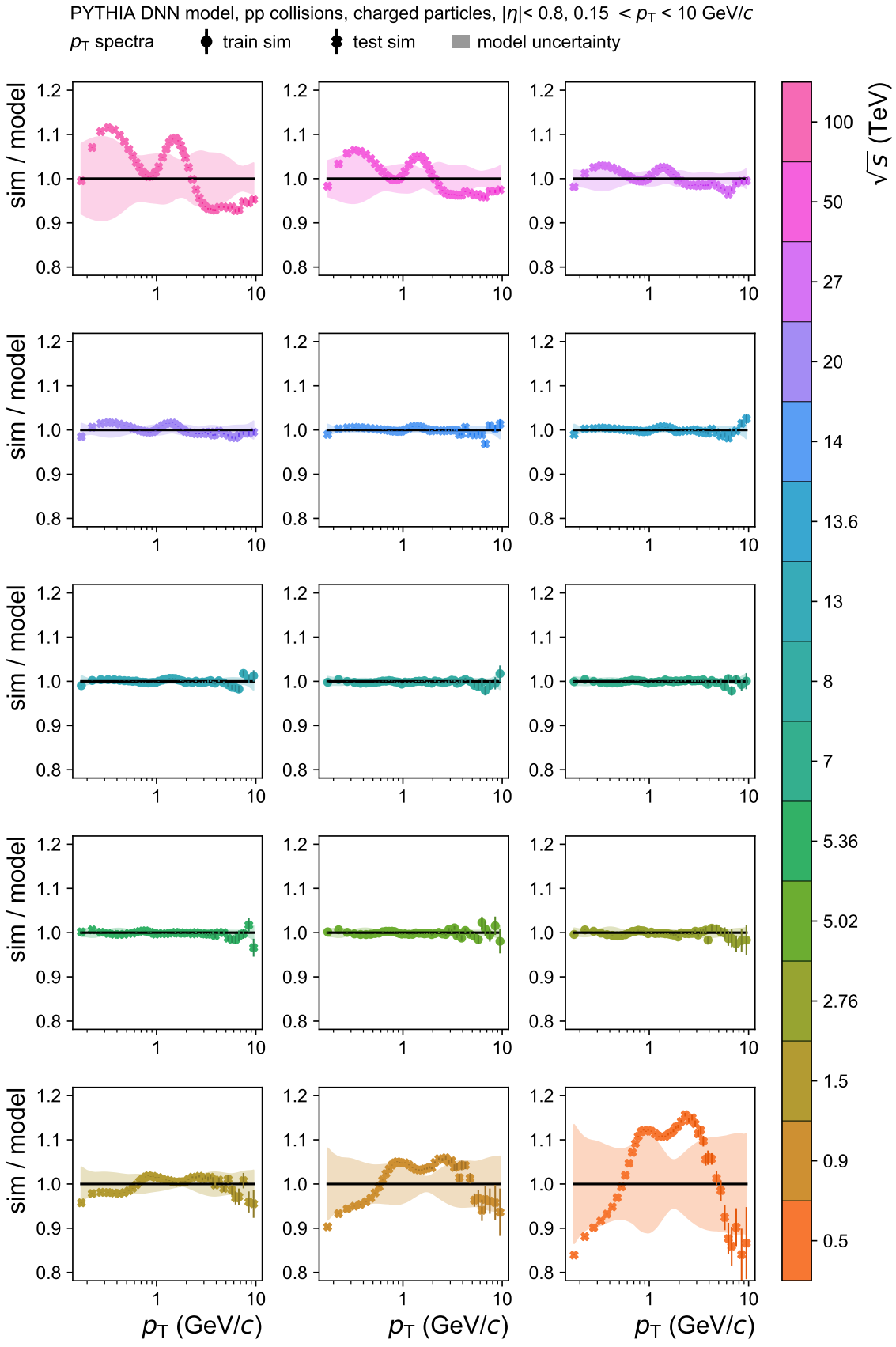


Figure 4.5: Ratio of the p_T spectra predicted by the PYTHIA DNN model and those from PYTHIA simulations for all considered center-of-mass energies.

4.3 Interpolated pp reference

The particle production in central heavy-ion collisions is suppressed due to the presence of *quark-gluon plasma* where the partons lose energy in the medium. The Glauber model describes nucleus-nucleus (AA) collisions as geometric superpositions of multiple binary pp collisions. Comparing the measured particle production in AA collisions with the expectation from the Glauber model offers the opportunity to study the properties of the QGP by means of the *nuclear modification factor* (R_{AA}).

In 2017, ALICE recorded Xe–Xe collisions at a center-of-mass energy per nucleon pair of $\sqrt{s_{\mathrm{NN}}} = 5.44$ TeV. Since there is no pp reference measurement at the same energy, in a previous publication an interpolation of spectra at the closest collision energies ($\sqrt{s} = 5.02$ TeV and $\sqrt{s} = 7$ TeV) was performed in order to calculate an R_{AA} for Xe–Xe [4]. Specifically, the p_{T}-differential cross sections for pp collisions measured at $\sqrt{s} = 5.02$ TeV and $\sqrt{s} = 7$ TeV were parametrized with a power-law function separately for each p_{T} interval. In this thesis, the DNN interpolations are based not on p_{T}-differential cross sections but on p_{T} spectra with a different normalization ($N_{\mathrm{ch}} > 0$ events). Therefore, a direct comparison between the published power-law interpolation and this thesis is not feasible. However, the publication includes a dedicated figure (Figure 2) where the ratio of the interpolated p_{T}-differential cross section for $\sqrt{s} = 5.44$ TeV to the measured one at $\sqrt{s} = 5.02$ TeV is shown. This ratio quantifies the difference in the spectral shape independent of the normalization. Therefore, the DNN interpolation presented in this thesis can be compared to the power-law interpolation in [4]. Furthermore, the ratio is shown between the PYTHIA-simulated p_{T}-differential cross sections at $\sqrt{s} = 5.44$ TeV and $\sqrt{s} = 5.02$ TeV in the publication. Figure 4.6 depicts these ratios from the publication together with those of the ALICE DNN and PYTHIA DNN interpolations presented in this thesis with corresponding uncertainties. It is important to mention that the DNN models are based on five different LHC energies and are purely data-driven, making no previous assumptions regarding the functional relationship between the spectral shape and $\sqrt{s}$. In contrast, the published interpolation only considers two pp energies and assumes a spectral shape scaling with $(\sqrt{s})^n$. As expected, the interpolation by the PYTHIA-trained DNN model aligns perfectly with the PYTHIA-simulated data. Within their uncertainties, the published power-law interpolation and the interpolation by the ALICE-trained DNN model are consistent. For transverse momenta above approximately $p_{\mathrm{T}} = 2$ GeV/c, they align perfectly. Interestingly, at low p_{T} the power-law interpolation yields larger values than the DNN interpolation. Here, the ALICE-DNN interpolation shows a perfect alignment with the PYTHIA simulations. These collective observations are a strong validation of the interpolation capabilities of the DNN models. Furthermore, they point

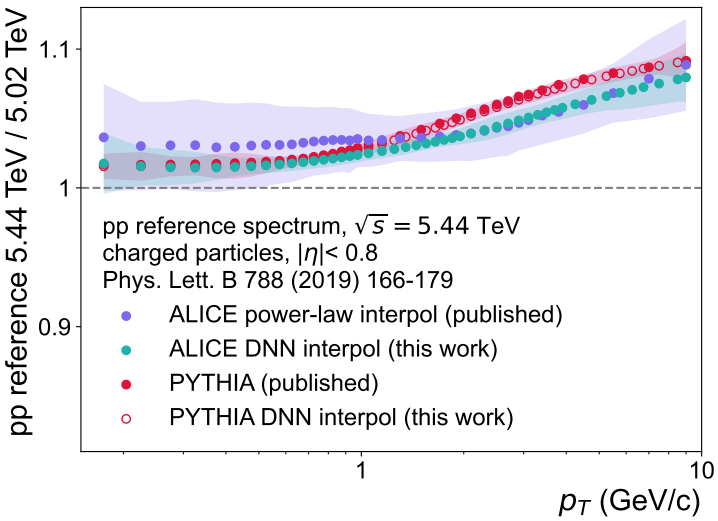


Figure 4.6: Ratio of pp reference spectrum at $\sqrt{s} = 5.44$ TeV obtained via power-law interpolation [4] versus DNN interpolation (this work) to $\sqrt{s} = 5.02$ TeV for both PYTHIA-simulated data and ALICE measurements.

to the power-law interpolation being more accurate at higher p_{T} while deteriorating towards low p_{T}. Overall, the ratio between $\sqrt{s} = 5.44$ TeV and $\sqrt{s} = 5.02$ TeV is shown to have a smoother trend for the ALICE-DNN interpolation than the power-law interpolation.

4.4 Extrapolation to RHIC energies

Figure 4.7 extends the PYTHIA DNN of extrapolations the N_{ch} distributions (left) and p_{T} spectra (right) shown in Figure 4.1 and Figure 4.4 to a lower energy of $\sqrt{s} = 0.2$ TeV. This is done to study the model's extrapolation performance at RHIC energies. The ratio between the prediction and the simulated data at $\sqrt{s} = 0.2$ TeV demonstrates large deviations of up to 80% for the N_{ch} distributions and 50% for the p_{T} spectra. These discrepancies are not covered by the model uncertainties. This presents a prominent contrast to the other simulated energies, where predictions are largely consistent with the data within their combined systematic uncertainties. Apparently, the DNN model fails to correctly describe the PYTHIA data in this low-energy regime, indicating that the energy dependence learned by the model from the simulated data at LHC energies is not transferable to the simulated data at RHIC energies. As mentioned in Section 2.2, the PYTHIA tune employed in this thesis to simulate the data is optimized to best describe $\sqrt{s} = 7$ TeV LHC measurements. Previous research [8] has concluded that this *Monash* tune of PYTHIA does not accurately describe RHIC measurements of pp collisions at $\sqrt{s} = 0.2$ TeV. It is argued that the discrepancy be-

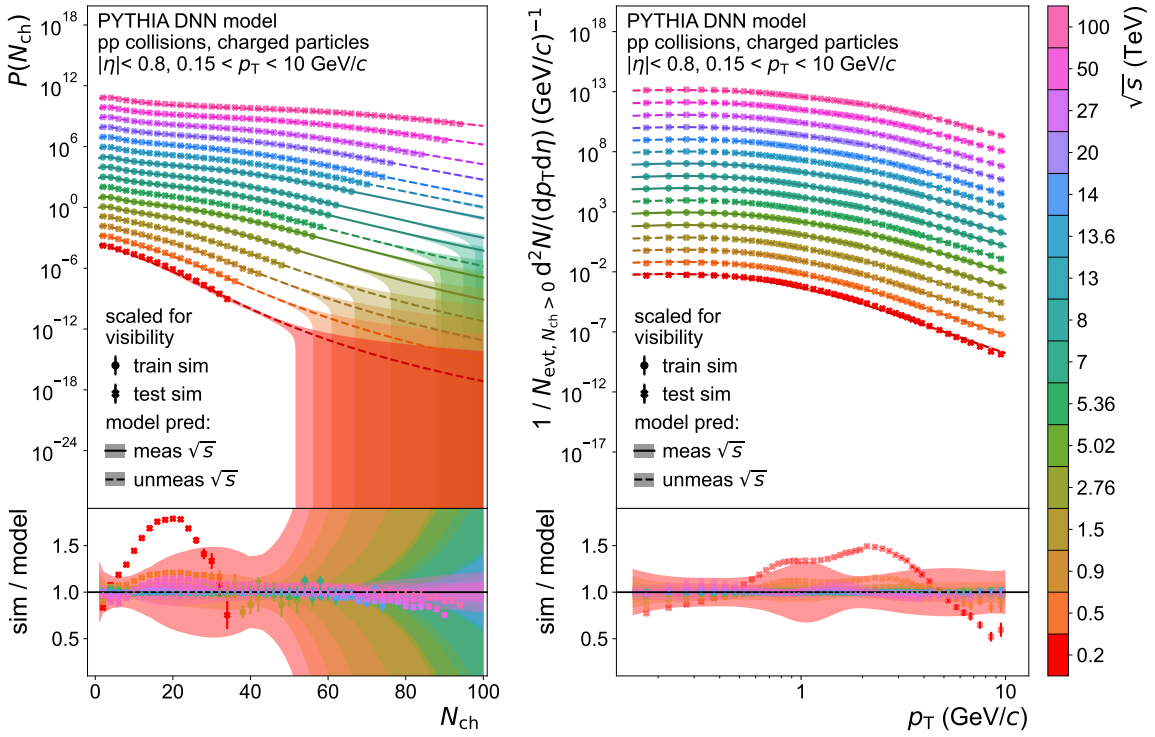


Figure 4.7: PYTHIA DNN parametrizations and extrapolations of N_{ch} distributions (left) and p_{T} spectra (right) including simulated data at $\sqrt{s} = 0.2$ TeV.

tween simulated and experimental data arises due to incorrect modeling of the energy dependence of the underlying event. The energy-dependent spectral shapes predicted by the DNN models at this energy regime diverge from the ones that the *Monash* tune simulates. Given that these simulations also deviate from the actual distributions, a different approach is needed to study the quality of the model's extrapolation to RHIC energies. Therefore, in a future extension of the analysis presented in this thesis, a different PYTHIA tune optimized to describe the RHIC energies [8] could be compared to the DNN model predictions. Another study worth exploring in the future could involve comparing the $\sqrt{s} = 0.2$ TeV extrapolation of the ALICE DNN model with corresponding experimental data from RHIC.

4.5 $\langle N_{\mathrm{ch}} \rangle$ and $\langle p_{\mathrm{T}} \rangle$

Figure 4.8 shows the $\langle N_{\mathrm{ch}} \rangle$ derived from N_{ch} distributions as a function of the center-of-mass energy over an energy range of $\sqrt{s} = 0.5$ TeV to 100 TeV. For the PYTHIA simulations (left) and ALICE data (right), the circles represent training energies and the crosses the test energies. The solid line represents the model predictions for a continuous range of energies between $\sqrt{s} = 0.5$ TeV and 100 TeV, with an error band

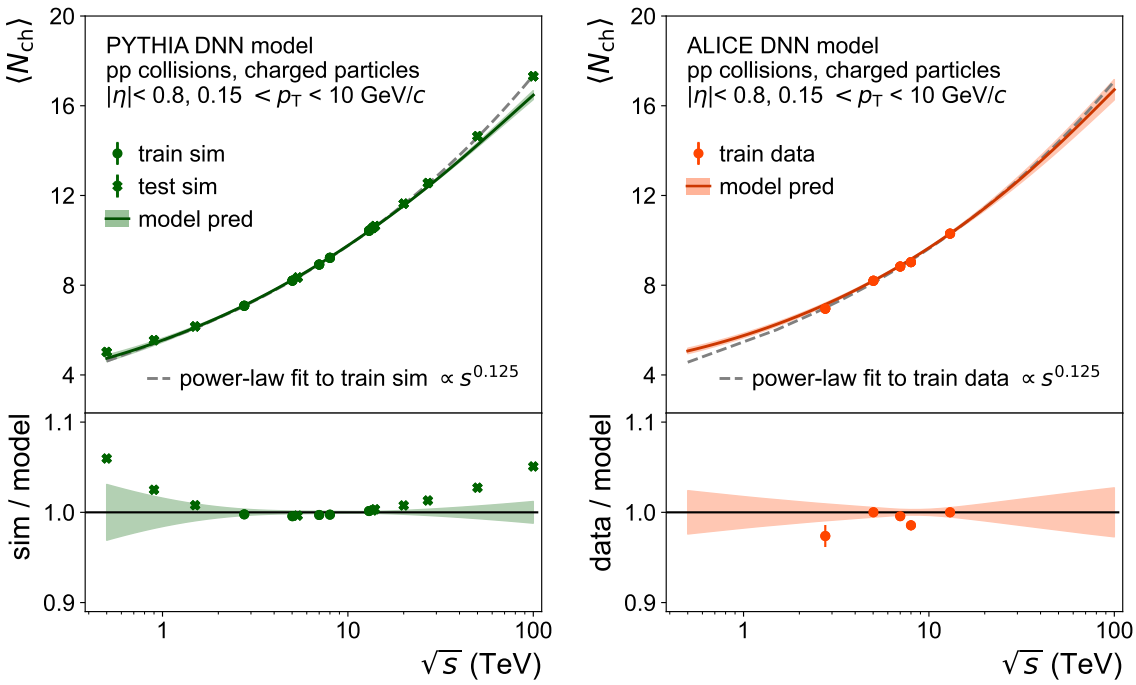


Figure 4.8: Mean N_{ch} derived from the N_{ch} distributions predicted by the PYTHIA DNN model (left) and the ALICE DNN model (right) with the $\langle N_{\mathrm{ch}} \rangle$ from corresponding PYTHIA simulations and ALICE data.

indicating the model uncertainty. The lower panel of Figure 4.8 (left) depicts the ratio between the $\langle N_{\mathrm{ch}} \rangle$ values from the PYTHIA simulations and the $\langle N_{\mathrm{ch}} \rangle$ values predicted by the PYTHIA DNN model. In the lower panel of the Figure 4.8 (right), the same is shown for the $\langle N_{\mathrm{ch}} \rangle$ values from ALICE measurements and those predicted by the ALICE DNN model. As illustrated in Figure 2.5, measurements of multiple experiments indicate that the average number of charged particles produced in a pp collision follows a power-law trend as a function of $\sqrt{s}$. Therefore, a power-law fit was performed to the $\langle N_{\mathrm{ch}} \rangle$ values from the PYTHIA-simulated data and the ALICE data, respectively, to compare it to the energy dependence predicted by the DNN models. In Figure 4.8, the power-law fit to the training data of each model at the five LHC energies is depicted as a dashed gray line. Notably, the fits both to the PYTHIA simulations and the ALICE data yield the same exponent ($s^{0.125}$), which is comparable to the exponents quoted in Figure 2.5.

The $\langle N_{\mathrm{ch}} \rangle$ values derived from N_{ch} distributions predicted by the PYTHIA DNN model show a smooth energy dependence over the whole considered $\sqrt{s}$ range. They show an excellent agreement with those from the PYTHIA-simulated N_{ch} distributions at LHC energies, where the ratio is consistent with unity. Interestingly, the power-law fit to the five LHC energies perfectly describes the energy dependence of the PYTHIA-simulated charged-particle production. The predicted energy dependence

of the $\langle N_{\mathrm{ch}} \rangle$ values by the PYTHIA DNN model demonstrates a perfect alignment to the power-law fit within the model uncertainties and starts to deviate slightly for center-of-mass energies above $\sqrt{s} = 20$ TeV. At $\sqrt{s} = 100$ TeV, the power-law fit and PYTHIA predict a slightly larger average number of produced charged particles than the DNN model. These results for the DNN model are remarkable considering that it is trained on N_{ch} distributions at five LHC energies and is still able to extrapolate to a wide energy range, largely following an empirically observed power-law scaling of the charged-particle production as a function of $\sqrt{s}$.

The $\langle N_{\mathrm{ch}} \rangle$ values derived from the N_{ch} distributions predicted by the ALICE DNN model are shown in Figure 4.8 (right) together with those derived from ALICE-measured N_{ch} distributions. These derived $\langle N_{\mathrm{ch}} \rangle$ values from the ALICE DNN model are largely consistent with the ALICE data. The ratio shows that the highest deviation at $\sqrt{s} = 2.76$ TeV amounts to about 2%. It is important to mention that the $\langle N_{\mathrm{ch}} \rangle$ predicted by the ALICE DNN model was calculated for a multiplicity range of $1 \leq N_{\mathrm{ch}} \leq 100$. In contrast, the N_{ch} distribution from the ALICE measurement at this energy only reaches a multiplicity of $N_{\mathrm{ch}} = 28$. Since a wider N_{ch} range results in a higher value for $\langle N_{\mathrm{ch}} \rangle$, this explains the observed deviation between the prediction and the data at this energy. Remarkably, the energy dependence of the $\langle N_{\mathrm{ch}} \rangle$ values derived from the ALICE DNN model aligns perfectly with the power-law fit to the $\langle N_{\mathrm{ch}} \rangle$ values of ALICE measurements within its uncertainties over an energy range of $\sqrt{s} = 5.02$ TeV up to 100 TeV. Below $\sqrt{s} = 5.02$ TeV, the model predicts a slightly higher average number of produced charged particles than the power-law fit. This discrepancy is explained by the much shorter N_{ch} range at $\sqrt{s} = 2.76$ TeV since the $\langle N_{\mathrm{ch}} \rangle$ value for this energy is included in the fit. The fact that the DNN model trained on N_{ch} distributions is able to correctly describe the empirically observed energy dependence of a derived observable, $\langle N_{\mathrm{ch}} \rangle$, over a wide $\sqrt{s}$ range highlights the excellent predictive power of the model.

Figure 4.9 shows the mean transverse momentum derived from p_{T} spectra as a function of the center-of-mass energy over an energy range of 0.5 TeV $\leq \sqrt{s} \leq$ 100 TeV derived from the PYTHIA DNN model (left) and from the ALICE DNN model (right). The representation of data points, model predictions and uncertainties is analogous to Figure 4.8. The $\langle p_{\mathrm{T}} \rangle$ values derived from the PYTHIA DNN model show a very good agreement with the $\langle p_{\mathrm{T}} \rangle$ values derived from the PYTHIA-simulated p_{T} distributions. PYTHIA predicts a higher average transverse momentum for the produced charged particles at lower energies and a lower $\langle p_{\mathrm{T}} \rangle$ at higher energies compared to the PYTHIA DNN model. The $\langle p_{\mathrm{T}} \rangle$ values derived from the ALICE DNN model shown in Figure 4.9 (right) are in perfect agreement with those derived from the p_{T} spectra from ALICE measurements. The deviations are well below 1%. Overall, the ALICE DNN

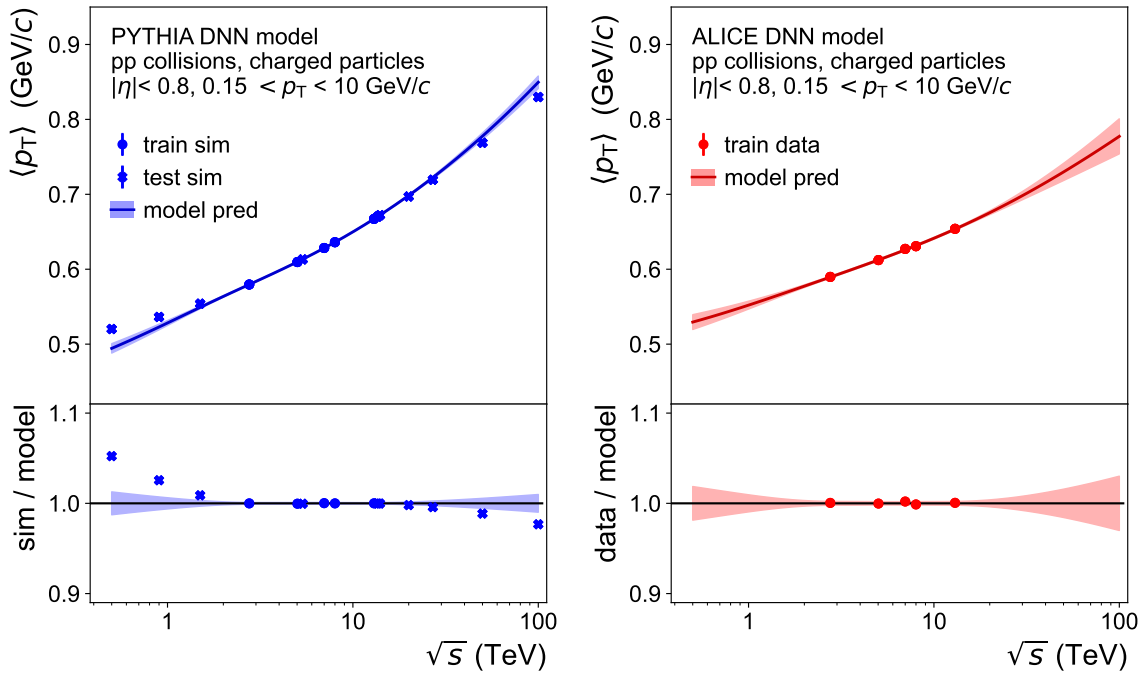


Figure 4.9: Mean p_T derived from the p_T distributions predicted by the PYTHIA DNN model (left) and the ALICE DNN model (right) with the $\langle p_T \rangle$ from corresponding PYTHIA simulations and ALICE data.

model predicts a smooth energy dependence of $\langle p_T \rangle$. The larger model uncertainties of low and high-energy extrapolations indicate the model's caution in these regions. Therefore, it is likely that the actual $\langle p_T \rangle$ values fall within the model uncertainties. The lowest and highest $\sqrt{s}$ extrapolations ($\sqrt{s}$ = 0.5 TeV and 100 TeV) demonstrate a confidence of 2% and 3%, respectively.

The DNN models used in this thesis to predict N_{ch} distributions and p_T spectra are not only trained on data from different observables but also have entirely different architectures. Since both of these charged-particle spectra describe the same collisions, a common variable derived from them must be consistent with each other. The mean charged-particle multiplicity at a given center-of-mass energy can not only be derived from the N_{ch} distribution but also from the corresponding p_T spectrum. This is shown in Figure 4.10 as a function of the center-of-mass energy for the PYTHIA-simulated data and the PYTHIA DNN model (left), as well as the ALICE data and the ALICE DNN model (right). Figure 4.10 (left) shows that the $\langle N_{ch} \rangle$ values derived from both types of PYTHIA-simulated charged-particle spectra (green and pink data points) align perfectly. The predictions by the PYTHIA DNN models also show an excellent degree of consistency within their systematic uncertainties, even for $\sqrt{s}$ extrapolations as far as 100 TeV. In the low-energy region, the predicted spectra show a negligible deviation with the predicted N_{ch} distribution yielding $\langle N_{ch} \rangle \approx 4.2$ and the predicted

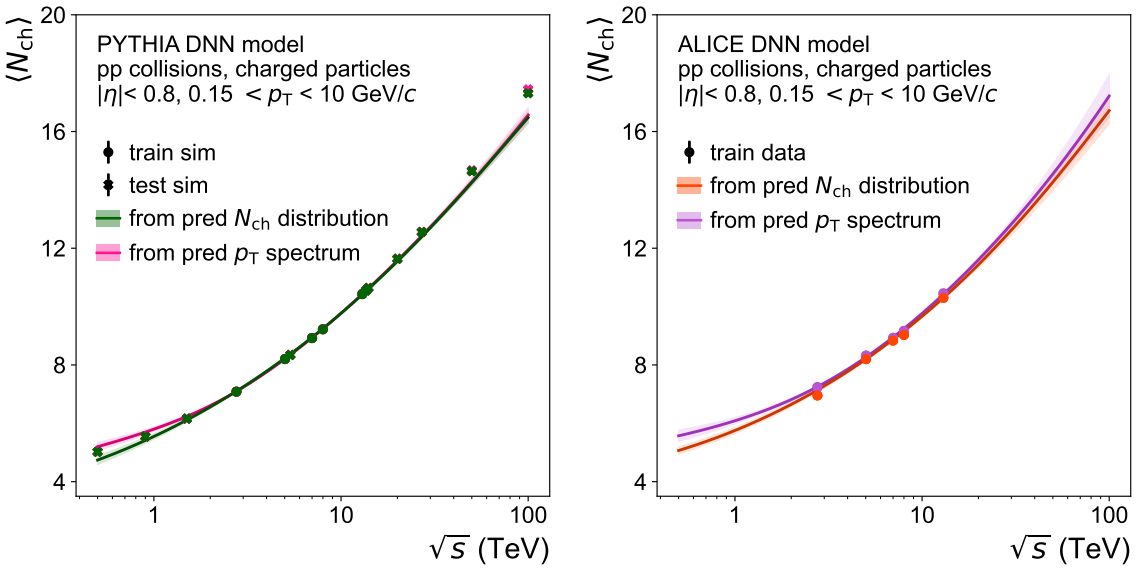


Figure 4.10: Mean N_{ch} derived from N_{ch} and p_{T} distributions predicted by the PYTHIA DNN model (left) and the ALICE DNN model (right) with the $\langle N_{\mathrm{ch}} \rangle$ from corresponding PYTHIA simulations and ALICE data.

p_{T} spectrum yielding $\langle N_{\mathrm{ch}} \rangle \approx 4.3$. In Figure 4.10 (right) the $\langle N_{\mathrm{ch}} \rangle$ values derived from the ALICE-measured p_{T} spectra are slightly higher than those from the measured N_{ch} distributions, which might be a result of the limited N_{ch} range considered for the multiplicity distributions. This same tendency towards slightly higher values of $\langle N_{\mathrm{ch}} \rangle$ is observed for the predicted p_{T} spectra by the ALICE DNN model. Despite this, the $\langle N_{\mathrm{ch}} \rangle$ values derived from these spectra are largely consistent with one another within the systematic uncertainties of the models across the whole $\sqrt{s}$ range, including the extrapolations. This can be interpreted as the DNN models understanding the same evolution of the charged-particle production as a function of $\sqrt{s}$, despite being trained on different datasets and having a completely different architecture. This is a strong validation of the predictive power and reliability of the ALICE DNN models.

4.6 ALICE DNN predictions vs. PYTHIA

The ALICE DNN models presented in this thesis allow for predicting the charged-particle spectra beyond LHC energies. As opposed to PYTHIA, they make no assumptions regarding particle-production mechanisms but predict the charged-particle spectra solely based on correlations learned from being trained on the ALICE measurements at five different LHC energies. In contrast, the PYTHIA *Monash* tune is optimized to best describe experimental measurements at $\sqrt{s} = 7$ TeV. Both PYTHIA and the ALICE DNN model can predict charged-particle spectra for pp collisions at

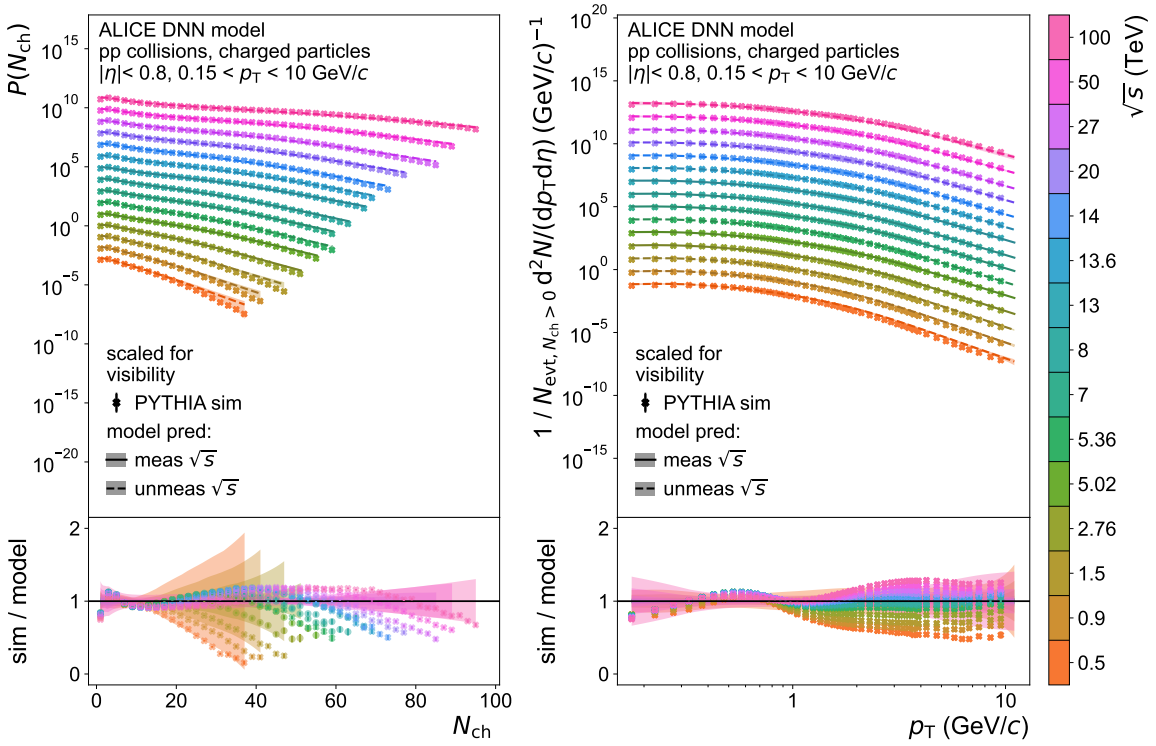


Figure 4.11: PYTHIA-simulated N_{ch} distributions (left) and p_{T} spectra (right) at different $\sqrt{s}$ with corresponding predictions by the ALICE DNN model.

any given center-of-mass energy. The following study quantifies the differences between the charged-particle spectra predicted by these independent approaches.

Figure 4.11 depicts the N_{ch} distributions (left) and p_{T} spectra (right) predicted both by PYTHIA and the ALICE DNN model over a center-of-mass energy range of $0.5\ \mathrm{TeV} \leq \sqrt{s} \leq 100\ \mathrm{TeV}$. The markers represent the PYTHIA-simulated data, while the lines represent the predictions by the ALICE DNN model which was trained on ALICE measurements. As in previous Figures, the colors represent different energies. Solid lines indicate the DNN parametrizations of the ALICE data and dashed lines the extrapolations to unmeasured energies. The uncertainties of the ALICE DNN model are depicted as error bands, while the statistical uncertainties of the PYTHIA-simulated data are depicted as vertical error bars. The lower panels of the Figure show the ratio between the PYTHIA-simulated data and the predictions by the ALICE DNN model. The N_{ch} distributions predicted by the ALICE DNN model and those predicted by PYTHIA are consistent within the multiplicity range $N_{\mathrm{ch}} \lesssim 15$. Above this region, the predictions begin to diverge gradually with increasing N_{ch}. At high multiplicities, PYTHIA predicts a lower probability for pp collisions than the ALICE DNN model. The p_{T} spectra predicted by PYTHIA show an almost energy-independent deviation from those predicted by the ALICE DNN model up to a transverse momentum of approximately $p_{\mathrm{T}} = 1\ \mathrm{GeV}/c$. Above this region, the deviations are more pronounced

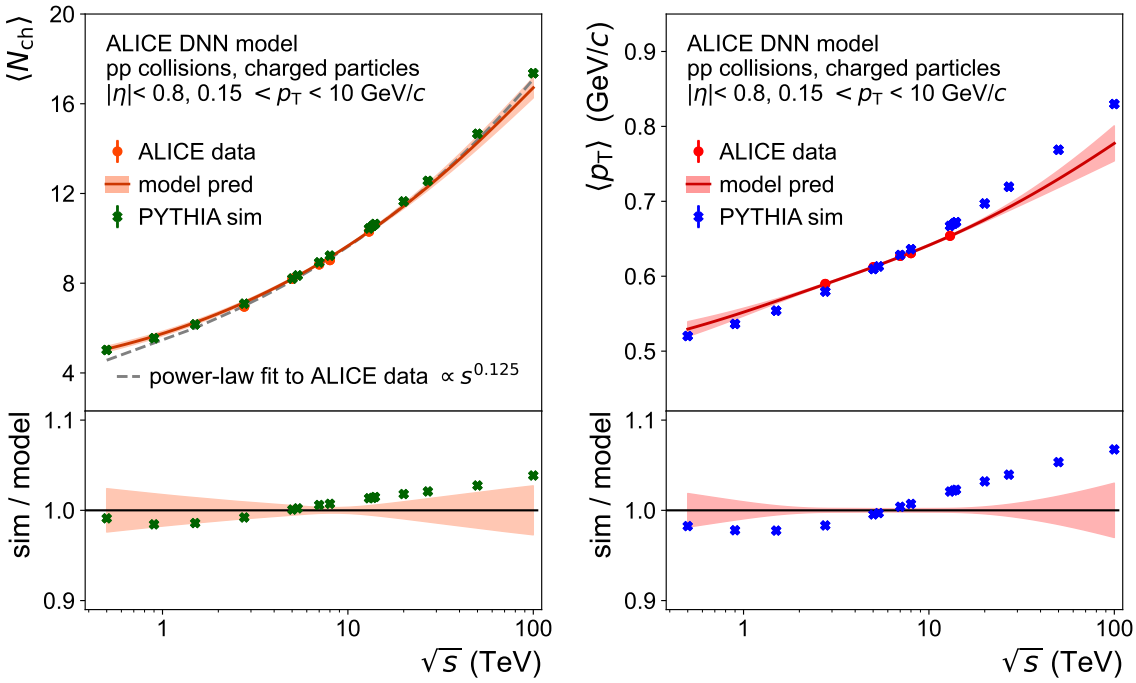


Figure 4.12: Mean N_{ch} derived from N_{ch} distributions (left) and mean p_{T} derived from p_{T} spectra (right) simulated by PYTHIA and those predicted by the ALICE DNN model.

and show a clear energy ordering. Interestingly, the predictions seem to be the most consistent at $\sqrt{s} = 7$ TeV, which is the tuning energy implemented in the PYTHIA *Monash* tune. At collision energies below $\sqrt{s} = 7$ TeV, PYTHIA predicts softer p_{T} spectra than the ALICE DNN model. Above the tuning energy, it predicts harder spectra than the ALICE DNN model.

Figure 4.12 shows $\langle N_{\mathrm{ch}} \rangle$ (left) and $\langle p_{\mathrm{T}} \rangle$ (right) from the PYTHIA simulation and the ALICE DNN model as a function of the center-of-mass energy. Furthermore, they include the $\langle N_{\mathrm{ch}} \rangle$ and $\langle p_{\mathrm{T}} \rangle$ from the ALICE measurements as well as the power-law fit to this data. The lower panels of the Figure show the ratio between the $\langle N_{\mathrm{ch}} \rangle$ (left) and $\langle p_{\mathrm{T}} \rangle$ (right) values derived from PYTHIA-simulated spectra and those derived from the predictions by the ALICE DNN model. The $\langle N_{\mathrm{ch}} \rangle$ values derived from the N_{ch} distributions predicted by the ALICE DNN model almost perfectly describe those derived from the PYTHIA-simulated data within the model uncertainties with deviations of less than 5% over the whole energy range. Furthermore, the power-law fit to the ALICE data also describes the $\langle N_{\mathrm{ch}} \rangle$ values from PYTHIA with high accuracy. At low energies, PYTHIA is most consistent with the ALICE DNN model. At high energies, its values are best described by the power-law fit to the ALICE data. Remarkably, the highest consistency between the ALICE DNN model and PYTHIA is found at $\sqrt{s} = 7$ TeV, which is the tuning energy for the PYTHIA *Monash* tune used in

this thesis. For the $\langle p_{\mathrm{T}} \rangle$ values derived from the p_{T} spectra shown in Figure 4.12 (right), the deviations between the ALICE DNN model and PYTHIA are more pronounced. As observed for $\langle N_{\mathrm{ch}} \rangle$, the values of $\langle p_{\mathrm{T}} \rangle$ from the ALICE DNN model and PYTHIA are the most consistent at the tuning energy of the PYTHIA *Monash* tune. For higher energies, PYTHIA predicts a higher mean transverse momentum for the produced charged particles than the ALICE DNN model (up to 8% at $\sqrt{s} = 100$ TeV). In contrast, it expects a smaller $\langle p_{\mathrm{T}} \rangle$ for energies below $\sqrt{s} = 7$ TeV (up to 2% at $\sqrt{s} = 0.5$ TeV).

Chapter 5

Summary and outlook

In this thesis, two independent DNN models are trained with the published measurements of charged-particle N_{ch} distributions and p_{T} spectra by ALICE at the LHC energies: $\sqrt{s}$ = 2.76, 5.02, 7, 8 and 13 TeV [3]. The modeled spectra are extended to a wide energy range from $\sqrt{s}$ = 0.5 TeV up to 100 TeV. Furthermore, the N_{ch} distributions are extrapolated to a multiplicity of up to 100 charged particles, a region often not accessible experimentally due to missing statistics. In this region, the model uncertainties are significantly larger due to a lack of experimental constraints. The predictions of the ALICE DNN models at the LHC energies used for training show an excellent agreement with the training data. The interpolation and extrapolation capabilities are assessed on two PYTHIA DNN models, with predictions being compared to the PYTHIA-simulated data across the considered energy range of 0.5 TeV $\leq \sqrt{s} \leq$ 100 TeV. The energy interpolation performance is shown to be excellent. The energy extrapolations are widely consistent with the test data. Extrapolations to higher energies up to $\sqrt{s}$ = 20 TeV show the same accuracy as the energy interpolation. With increasing logarithmic distance to the training energies, the extrapolation performance deteriorates gradually, but the systematic model uncertainties increase as well. This thesis has proven that it is possible to parametrize the measured spectra with DNNs. The DNN models in this thesis have yielded predictions for various unmeasured energies that will be accessible with future experiments. As these predictions are purely data-driven, they could serve as the LHC baseline expectation regarding the charged-particle production of high-energy pp collisions.

A previous analysis performed a power-law interpolation of p_{T}-differential cross sections from pp collisions in ALICE to provide a pp reference for Xe–Xe collisions at $\sqrt{s_{\mathrm{NN}}}$ = 5.44 TeV [4]. This allows for the comparison of the DNN model and the power-law interpolation. The ratio between the proton-proton p_{T} spectra predicted by the PYTHIA DNN model at $\sqrt{s}$ = 5.44 TeV and $\sqrt{s}$ = 5.02 TeV shows perfect agreement with that of the PYTHIA-simulated p_{T} spectra at the same energies. The

ratio of the predictions by the ALICE DNN model is highly consistent with that of the PYTHIA simulations at low p_{T} and with the power-law interpolation at high p_{T}. Furthermore, the DNN shows significantly smaller systematic uncertainties than the power-law interpolation. These observations point to the DNN approach being probably more accurate than the power-law interpolation. Therefore, the DNN model in this thesis is shown to be a powerful state-of-the-art tool for providing a pp reference for heavy-ion collisions.

The DNN models in this thesis allow for making continuous predictions regarding the mean number of charged particles expected to be produced in a pp collision as well as their mean p_{T} over a wide range of center-of-mass energies. In the case of the ALICE DNN models, the $\langle N_{\mathrm{ch}} \rangle$ and $\langle p_{\mathrm{T}} \rangle$ extracted from the predictions describe those from the ALICE data perfectly, with deviations below 0.5% for $\langle p_{\mathrm{T}} \rangle$. In the case of $\langle N_{\mathrm{ch}} \rangle$, a deviation of 5% is observed at $\sqrt{s} = 2.76$ TeV, which can be caused due to the different N_{ch} ranges between the measurement and the prediction (max. $N_{\mathrm{ch}} = 27$ vs. $N_{\mathrm{ch}} = 100$). The $\langle N_{\mathrm{ch}} \rangle$ and $\langle p_{\mathrm{T}} \rangle$ extracted from the PYTHIA DNN model predictions not only describe those from the training data perfectly but also capture the overall trend shown by the simulated data over the whole energy range. More importantly, the predictions of both DNN models show excellent consistency with the empirically observed power-law scaling of $\langle N_{\mathrm{ch}} \rangle$ as a function of $\sqrt{s}$. The ALICE DNN models provide an estimation of how many charged particles are expected to be produced and what their average transverse momentum will be in future accelerators like HE-LHC ($\sqrt{s} = 27$ TeV) or FCC-hh ($\sqrt{s} = 100$ TeV). This could be helpful input for their planning and experimental design.

The robust ALICE DNN models allow for a comparison between the expected evolution of the spectral shapes as a function of the center-of-mass energy with that of the PYTHIA-simulated data. This comparison yield that the PYTHIA simulations are the most consistent with the measured spectra at $\sqrt{s} = 7$ TeV, precisely the energy that PYTHIA was tuned to best describe. In general, the ALICE DNN models predict harder spectra than those simulated by PYTHIA. This means that they predict a larger production of charged particles. The ratio between the simulated data and the ALICE DNN model shows a clear energy ordering with the deviations becoming larger at the tail of the spectra and, especially visible in the case of the p_{T} spectra, with increasing distance from the training energy range.

A possible extension of the analysis described in this thesis could include a comparison of the predictions by the ALICE DNN models to measurements of pp collisions recorded in LHC Run 3 at $\sqrt{s} = 0.9$ and 13.6 TeV for a direct assessment of their extrapolation capabilities. A further possibility to exploit the predictive power of DNNs would be to model the charged-particle production of other collision systems

at various energies available from ALICE measurements, which include p–Pb, Xe–Xe and Pb–Pb. It could even be feasible to include the collision system itself as an input variable to the model together with N_{ch} (or p_{T}) and $\sqrt{s}$, so that a possible correlation between the collision system and the resulting charged-particle spectra can be modeled. Most importantly, the published ALICE measurements include an even more fundamental observable: the multiplicity-dependent charged-particle p_{T} spectra over all previously mentioned collision systems and LHC energies. A further aim beyond the scope of this thesis will be to model these two-dimensional spectra using a DNN. This study could provide a more complete picture of the charged-particle-production mechanisms at play during the collisions over a wide range of center-of-mass energies beyond the LHC. This approach would parametrize a three-dimensional ($\sqrt{s}$, N_{ch}, p_{T}) phase space, exploiting the ability of DNNs to capture intricate, multidimensional patterns within data.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Prof. Dr. Henner Büsching for granting me the opportunity to write this thesis and for his kind guidance during my academic journey. I am furthermore very grateful to Prof. Dr. Harald Appelshäuser for agreeing to examine this thesis.

I am profoundly thankful to Mario Krüger for his mentoring and unwavering support throughout the development of my analysis and the writing of this thesis, extending far beyond regular office hours. I am also extremely grateful to Jerome Jung for his invaluable guidance and advice during this entire process. Special thanks go to Hannah Bossi for lending her expertise to this analysis and the pleasant conversations. I am also grateful to Stefanie Mrozinski for her encouragement and for proofreading this thesis and to Nicolas Strangmann for the engaging discussions regarding the analysis.

Furthermore, I want to thank the Frankfurt working group I am proud to be a part of for welcoming me so warmly and for the wonderful moments we have shared. I am also thankful to my current and former office colleagues Mario Krüger, Patrick Huhn, Ernst Hellbär and Muaz Al Halabi for contributing to a friendly working atmosphere.

Finally, I want to thank my family and loved ones for their unconditional support throughout my studies. This journey would not have been possible without them. I am also thankful to Prof. Dr. Bruno Deiss and Merve Yüksektepe for guiding me during my initial years in a new country.

References

[1] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 physics and manual", *Journal of High Energy Physics* **2006** (May, 2006) 026–026.

[2] P. Skands, S. Carrazza, and J. Rojo, "Tuning PYTHIA 8.1: the Monash 2013 tune", *The European Physical Journal C* **74** (Aug, 2014) .

[3] **ALICE** Collaboration, S. Acharya *et al.*, "Multiplicity dependence of charged-particle production in pp, p–Pb, Xe–Xe and Pb–Pb collisions at the LHC", *Physics Letters B* **845** (2023) 138110.

[4] **ALICE** Collaboration, S. Acharya *et al.*, "Transverse momentum spectra and nuclear modification factors of charged particles in Xe–Xe collisions at $\sqrt{s_{\mathrm{NN}}} = 5.44$ TeV", *Physics Letters B* **788** (2019) 166–179.

[5] P. D. Group, "Review of Particle Physics", *Phys. Rev. D* **98** (Aug, 2018) 030001.

[6] P. D. Group, "Review of Particle Physics", *Progress of Theoretical and Experimental Physics* **2022** (08, 2022) 083C01.

[7] 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** (Nov, 2012) 387–405.

[8] M. R. Aguilar, Z. Chang, R. K. Elayavalli, R. Fatemi, Y. He, Y. Ji, D. Kalinkin, M. Kelsey, I. Mooney, and V. Verkest, "PYTHIA 8 underlying event tune For RHIC energies", *Physical Review D* **105** (2022) .

[9] A. Tauro, "ALICE Schematics", 2017. `https://cds.cern.ch/record/2263642`. General Photo.

[10] A. A. Rostovtsev and A. A. Bylinkin, "Systematic studies of hadron production spectra in collider experiments", 2010.

[11] **ALICE** Collaboration, B. Kim, "Center of mass energy vs average $\mathrm{d}N\mathrm{d}\eta$ in pp collisions with a new data point at $\sqrt{s} = 13.6$ TeV (ALICE internal preliminary Figure)." `https://alice-figure.web.cern.ch/node/27753`.

[12] S. G. Krantz, *Guide to Topology*. American Mathematical Society, 2009.

[13] F. Marquardt, "Machine Learning for Physicists", 2021. `https://pad.gwdg.de/s/Machine_Learning_For_Physicists_2021`, accessed 2023-08-01.

[14] A. Geron, *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2 ed., 2019.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society. Series B (Methodological)* **58** (1996) 267–288.

[16] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net", *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67** (2005) 301–320.

[17] S. Rasmussen, "Pitfalls with Dropout and BatchNorm in regression problems." `https://towardsdatascience.com/pitfalls-with-dropout-and-batchnorm-in-regression-problems-39e02ce08e4d`, 2020.

[18] R. Eldan and O. Shamir, "The Power of Depth for Feedforward Neural Networks", 2016.

[19] K. Fukushima, "Cognitron: A self-organizing multilayered neural network", *Biological Cybernetics* (1975) .

[20] C. Dugas, Y. Bengio, F. Elisle, and C. Nadeau, "Incorporating Second-Order Functional Knowledge for Better Option Pricing", *Cirano Working Papers* (02, 2001) .

[21] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks", *CoRR* **abs/1706.02515** (2017) , `1706.02515`.

[22] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions", *CoRR* **abs/1710.05941** (2017) , `1710.05941`.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", 2015.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, eds., vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy, 13–15 may, 2010.

[26] R. Garnett, *Bayesian Optimization*. Cambridge University Press, 2023.

[27] Oriel Kiss, "Bayesian Optimization for machine learning algorithms in the context of Higgs searches at the CMS experiment",.

[28] H. L. Harney, *Bayes' Theorem*, pp. 8–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[29] E. Shokr, A. De Roeck, and M. A. Mahmoud, "Modeling of charged-particle multiplicity and transverse-momentum distributions in pp collisions using a DNN", *Scientific Reports* **12** (May, 2022) 8449.

[30] **STAR** Collaboration, J. Adams *et al.*, "The Multiplicity dependence of inclusive p_{T} spectra from pp collisions at $\sqrt{s} = 200$ GeV", *Phys. Rev. D* **74** (2006) 032006, `arXiv:nucl-ex/0606028`.

[31] M. Benedikt, *et al.*, "Future Circular Hadron Collider FCC-hh: Overview and Status", 2022.

[32] F. Chollet, "keras." `https://github.com/fchollet/keras`, 2015.

[33] M. A. et al., "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. `https://www.tensorflow.org/`. Software available from tensorflow.org.

Appendix A

Supplementary material

obs.	lay.	nod.	opt.	lr	act.	init.	λ_1	λ_2	MAE
	2	**32**	**A**	$\mathbf{1.16 \cdot 10^{-3}}$	**SW**	**RU**	$\mathbf{4.26 \cdot 10^{-4}}$	$\mathbf{2.08 \cdot 10^{-6}}$	**0.053**
	2	64	N	$1.00 \cdot 10^{-2}$	SW	RU	$5.00 \cdot 10^{-8}$	$5.00 \cdot 10^{-8}$	0.067
	2	64	A	$1.00 \cdot 10^{-2}$	SP	VS	$9.03 \cdot 10^{-6}$	$1.34 \cdot 10^{-4}$	0.074
	2	128	N	$1.52 \cdot 10^{-3}$	SP	RN	$2.10 \cdot 10^{-6}$	$5.00 \cdot 10^{-8}$	0.078
	2	128	A	$4.85 \cdot 10^{-4}$	SW	TN	$5.00 \cdot 10^{-8}$	$3.98 \cdot 10^{-4}$	0.078
$P(N_{\mathrm{ch}})$	3	32	N	$1.00 \cdot 10^{-2}$	SE	RN	$5.69 \cdot 10^{-5}$	$3.60 \cdot 10^{-5}$	0.079
	2	64	N	$4.07 \cdot 10^{-3}$	SP	RU	$4.05 \cdot 10^{-5}$	$1.47 \cdot 10^{-6}$	0.079
	2	64	A	$2.61 \cdot 10^{-3}$	SW	TN	$1.68 \cdot 10^{-6}$	$6.83 \cdot 10^{-7}$	0.080
	2	128	A	$1.00 \cdot 10^{-2}$	SP	TN	$1.45 \cdot 10^{-5}$	$5.00 \cdot 10^{-8}$	0.081
	2	256	A	$1.48 \cdot 10^{-3}$	SP	GN	$8.82 \cdot 10^{-5}$	$1.29 \cdot 10^{-6}$	0.082
	4	**64**	**A**	$\mathbf{4.61 \cdot 10^{-3}}$	**SP**	**TN**	$\mathbf{9.05 \cdot 10^{-7}}$	$\mathbf{6.55 \cdot 10^{-6}}$	**0.027**
	2	64	A	$7.17 \cdot 10^{-3}$	SP	RN	$2.41 \cdot 10^{-6}$	$4.77 \cdot 10^{-7}$	0.029
	2	256	N	$3.44 \cdot 10^{-3}$	SW	RU	$5.00 \cdot 10^{-8}$	$5.00 \cdot 10^{-8}$	0.032
	2	128	A	$2.01 \cdot 10^{-3}$	SW	RU	$5.00 \cdot 10^{-8}$	$3.13 \cdot 10^{-6}$	0.035
	2	64	N	$1.00 \cdot 10^{-2}$	SW	RU	$5.00 \cdot 10^{-8}$	$5.00 \cdot 10^{-8}$	0.037
p_{T} spect.	4	32	A	$9.26 \cdot 10^{-4}$	SW	RU	$5.00 \cdot 10^{-8}$	$3.89 \cdot 10^{-7}$	0.038
	4	256	N	$1.25 \cdot 10^{-3}$	SW	TN	$7.36 \cdot 10^{-8}$	$3.67 \cdot 10^{-6}$	0.038
	3	64	N	$8.82 \cdot 10^{-3}$	SP	TN	$3.11 \cdot 10^{-6}$	$2.34 \cdot 10^{-5}$	0.040
	3	256	A	$1.00 \cdot 10^{-2}$	SP	RN	$5.00 \cdot 10^{-8}$	$8.90 \cdot 10^{-5}$	0.041
	2	32	A	$5.84 \cdot 10^{-3}$	SW	RU	$2.56 \cdot 10^{-6}$	$1.23 \cdot 10^{-6}$	0.041

Table A.1: Ten top-performing model architectures from the hyperparameter scan with PYTHIA-simulated test data.

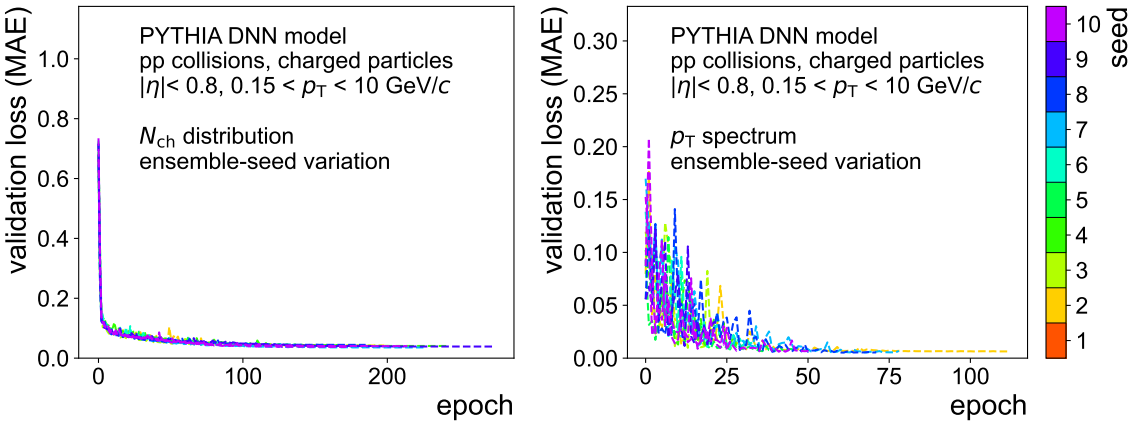


Figure A.1: Evolution of the validation loss functions of the PYTHIA DNN models with different ensemble seeds for the N_{ch} distributions (left) and p_{T} spectra (right).

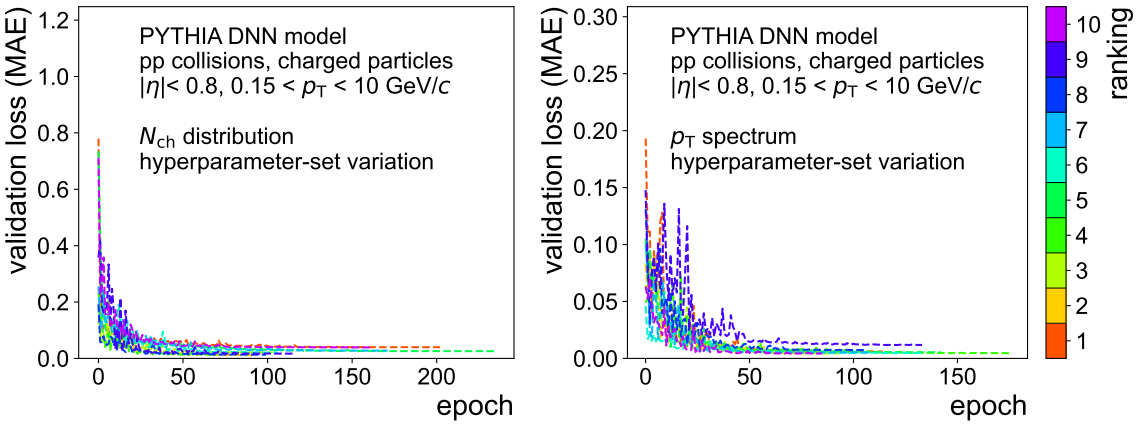


Figure A.2: Evolution of the validation loss functions of the PYTHIA DNN models with different hyperparameter sets for the N_{ch} distributions (left) and p_{T} spectra (right).

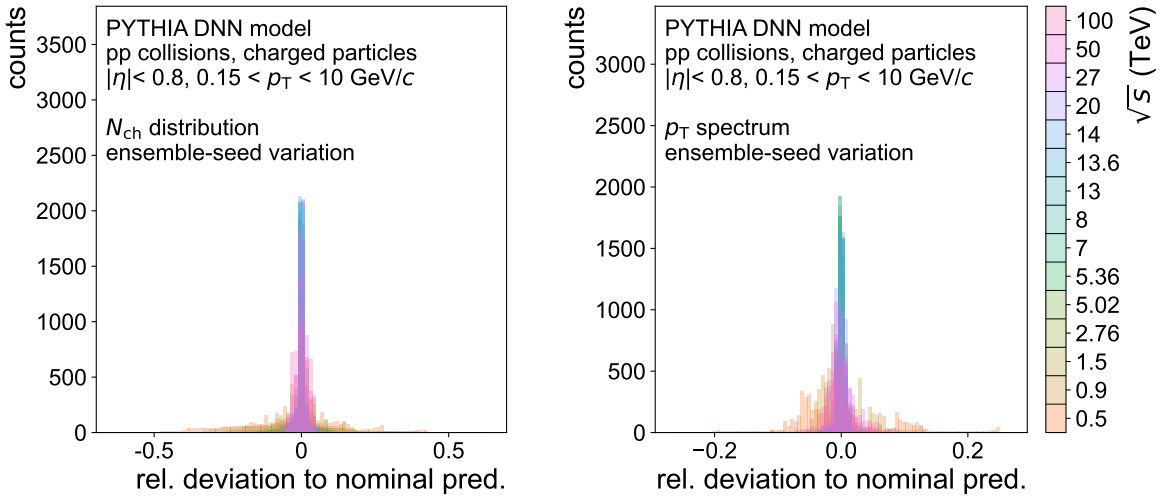


Figure A.3: Relative deviation for the N_{ch} distributions (left) and p_T spectra (right) from PYTHIA between ensemble models with different random seeds and the nominal predictions.

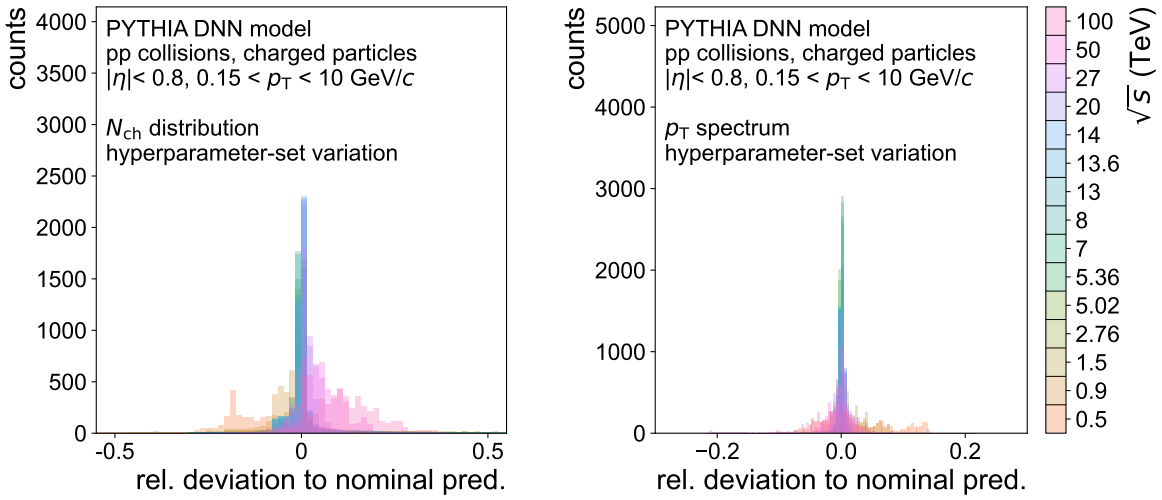


Figure A.4: Relative deviation for the N_{ch} distributions (left) and p_T spectra (right) from PYTHIA between models with different architectures and the nominal predictions.

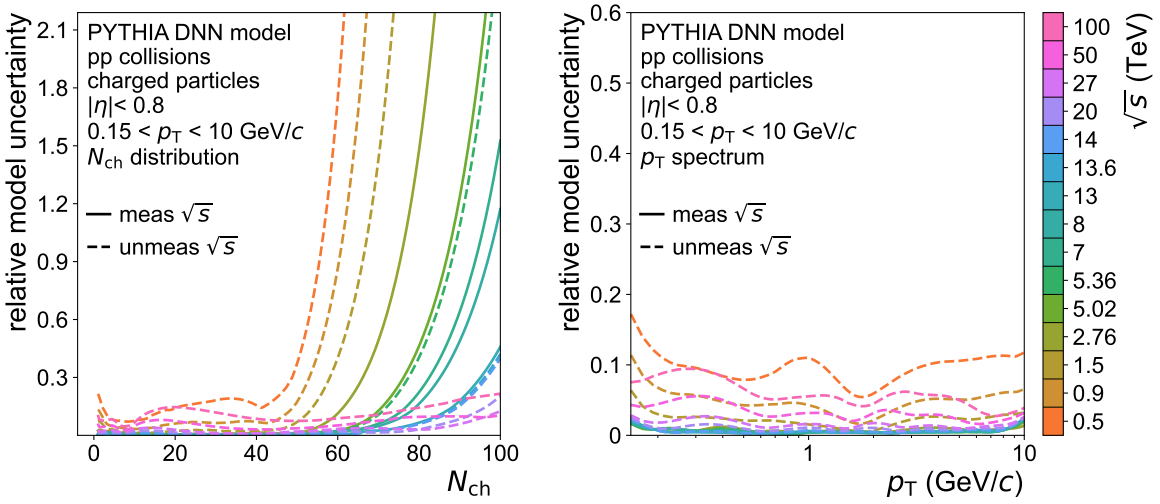


Figure A.5: Relative total systematic model uncertainty for the N_{ch} distributions (left) and p_T spectra (right) of PYTHIA.

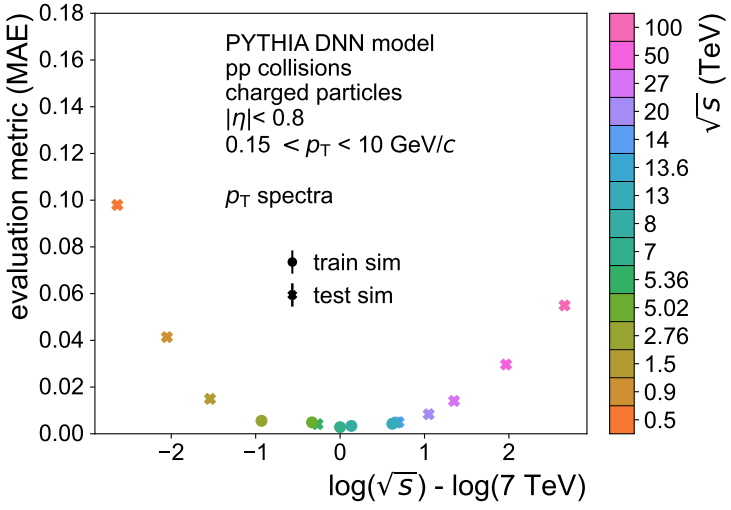


Figure A.6: MAE values of the comparison between p_T spectra predicted by the PYTHIA DNN model and those from PYTHIA simulations as a function of the logarithmic distance between the considered energies and $\sqrt{s}$ = 7 TeV .

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und sämtliche Stellen, die benutzten Werken im Wortlaut oder dem Sinne nach entnommen sind, mit Quellen- bzw. Herkunftsangaben kenntlich gemacht habe.

Frankfurt, den 14. September 2023

Maria Alejandra Calmon Behling