Jet Reconstruction and Graph Neural Networks in Heavy-Ion Collisions

Master Thesis in Theoretical Atomic, Nuclear and Particle Physics

by
Ida-Marie Fauske Johansson

A Preliminary Master's Thesis
Department of Physics and Technology
University of Bergen

May 26, 2023

Abstract

The study of quark-gluon plasma, a unique form of matter resulting from heavy-ion collisions, relies on investigating jet quenching – energy loss and substructure modifications experienced by jets within a hot quark-gluon plasma. To address the challenge of biased observations towards minimally modified jets, this thesis employs a graph neural network to predict the degree of modifications. By transforming jets into graph representations that capture their branching history, the network predicts energy loss caused by interactions with the medium. The hybrid strong/weak approach effectively models these interactions, while the energy loss ratio is determined by comparing the initial and final transverse momentum of the jets.

The primary objective of this research is to train the graph neural network using hybrid jets and accurately predict the energy loss ratio, while examining the network's resilience to background radiation. Results indicate satisfactory performance after training on hybrid jets. Notably, the network's performance significantly deteriorates when soft information is removed from the jets using grooming techniques. Furthermore, embedding jets in a heavy-ion environment leads to a gradual decrease in performance, albeit slower than information removal. Additionally, the choice of clustering algorithm heavily influences the graph representation of the jets. This study contributes to the understanding of energy loss estimation in jets traveling through a quark-gluon plasma medium.

Contents

Chapter 1

Introduction

Jets are collimated sprays of hadrons originating from the fragmentation of high-energy quarks or gluons. In relativistic heavy-ion collisions, jets are modified, or quenched, by interacting with the hot, dense quark-gluon plasma (QGP) created in the collision. Historically, the jet quenching phenomenon has been primarily studied through the suppression of intermediate-p_T hadrons at the Relativistic Heavy-Ion Collider (RHIC) at Brookhaven National Laboratory, and the dijet asymmetry and suppression of high-p_T hadrons and jets at the Large Hadron Collider (LHC) at CERN. Recently, efforts have been made to measure the modifications of internal properties of jets, also known as jet substructure measurements. Jet quenching has become one of the most powerful experimental probes of the properties of the hot, deconfined quark-gluon plasma produced in heavy-ion collisions [1].

It is generally believed that the momentum scales related to high-energy jet production are much larger than the scale of the local medium. This implies that the parton branching takes place independently of the medium. The parton branching plays an important role in the evolution of the jet and the amount of medium modification it can experience. Therefore, it is expected that the modification of measured jet properties correlates with the amount of energy lost in the medium, and one would like to be able to estimate the amount of energy loss a jet has experienced. This is a nontrivial task on a jet-by-jet basis.

In recent years, machine learning techniques have been widely applied to jet physics tasks, such as jet tagging [2], quark/gluon jet discrimination [3], and heavy flavor classifications [4]. Various machine learning techniques have been employed to tackle these problems, depending on how the jet is represented. The most popular jet representation is jet images, often used as input for convolutional neural networks (CNNs). Jet images encode the radiation patterns of the jets, however providing the network with additional substructure information presents a problem. Therefore, one can also propose using graph representations of jets. That can be done several ways, the most useful is perhaps the representation of jets as point clouds of constituent particles where one can use k-nearest neighbors to create a graph-like structure between the particle or as a graph of parton splittings that encodes information about the radiation patterns and substructure of jets.

Previous work has used CNNs and jet images to perform a regression task to predict the amount of energy loss a jet has suffered [5]. The results were quite good, and the network was able to draw correlations between the number of "extra" particles created

by interactions with the medium, and the amount of energy lost by the jet. However, when the jets are embedded into a realistic heavy ion environment, the performance of this model decreases [5]. Improving the performance of the model under the conditions in which jets are experimentally measured should be one of the primary focuses of such an analysis.

In an attempt to improve the predictions of the amount of energy loss suffered while embedded in a heavy ion background, this thesis will use a graph representation of jets and subsequently use a graph neural network (GNN) to make predictions on jet energy loss. This thesis will use a graph representation of the Lund tree of a jet, which essentially is a graph of the parton branching within a jet. The Lund tree contains a rich set of information on the substructure and radiation patterns of a jet, therefore serving as a natural way to represent jets in machine learning. To do such an analysis, we will use the existing framework of LundNet [6], modifying the network to perform a regression task instead of the classification task it was created for. The results of using the modified LundNet is that the overall prediction performance decreased but the model showed great resilience towards the embedding into a heavy-ion environment.

This thesis will focus on the prediction of energy loss of a jet traveling through a medium, while also discussing the effect of embedded heavy-ion background on these jets. We also wish to study the network itself and what features of the graph it deems most important to make accurate predictions.

This thesis is structured as follows: In Chapter 2 we introduce the fundamental concepts of QCD and heavy-ion collisions. Chapter 3 will introduce the concept of jets, both in medium and vacuum. We will focus on the concept of jet quenching and introduce the hybrid strong/weak model of jet quenching. Then we will discuss jet definitions, jet algorithms, and jet substructure, which will be important when we transform the jets into graphs. In Chapter 4 we introduce the basic concepts of machine learning, including the LundNet model. Chapter 5 is dedicated to introducing the machine learning task and summarizing previous results. In Chapter 6 we explore the effect of embedded background on vacuum jets. Chapter 7 discusses the results from our analysis and how well the model performed, as well as the different configurations of the model. Finally, we present our conclusion and suggestions for further work and improvement. All the code developed for this master thesis is publicly available at GitHub [7].

My contributions

I created vacuum jet events using PYTHIA [8] and identified the quark and gluon jets using FastJet [9]. I developed a code for creating a thermal Blast-Wave model for embedded heavy ion background, and to perform constituent subtraction and SoftDrop.Then I extracted the events from the hybrid model data created specifically for this project by Daniel Pablos and matched the vacuum and hybrid jets to calculate the energy loss ratio, and stored the jets and labels for the training of the network. I modified the existing LundNet model by changing the task from classification to regression and setting up the prediction performance analysis. Then I tuned the hyperparameters of the LundNet model to fit the data. Finally, I created a Q-jet algorithm which did not make it into the thesis for more than an appendix.

Part I

Physics Background

Chapter 2

QCD and Colliders

Jet physics is QCD physics. Therefore, a description of jets and their substructure relies on a deep understanding of the dynamics of strong interactions in collider experiments [10].

QCD is the theory describing the interaction of quarks and gluons, which can be found inside hadrons. We probe the underlying structure of hadrons through collider experiments. Depending on what structures we wish to probe, one can collide different types of particles. For example, by colliding electrons and protons in deep-inelastic scattering, one can probe the structure of the proton, which resulted in the discovery of point-like constituents inside the proton, namely quarks. Other collider experiments such as proton-proton (pp) collisions are often used to explore different aspects of QCD interactions, and collisions of heavy nuclei, called relativistic heavy-ion collisions, create dense quark-gluon plasma inside the detectors which affects the QCD interactions.

In this chapter, we will describe the basics of QCD needed to understand jet physics. After that, we will take a quick look at colliders, where we will look at what we should expect to observe in a detector and how to do the simplest theoretical calculations in relation to pp collisions. Finally, we will shortly introduce heavy-ion collisions and the ever so important quark-gluon plasma.

2.1 Quantum Chromodynamics

Quantum Chromodynamics (QCD) is the theory of the strong interactions within the Standard Model (SM) of particle physics. It describes the building blocks of strongly interacting particles and the forces acting on them.

The development of QCD started in the 1960s, with the purpose of understanding and classifying the multitude of new particles produced by the first particle colliders. These new, strongly-interacting particles were collectively referred to as **hadrons**. It was proposed that these hadrons were composed of different combinations of fundamental particles called **quarks**. This idea was the basis of the quark model which successfully applied group theory to describe the quantum numbers of hadrons.

The quark model leads to another important discovery: the introduction of a new quantum number, **color**. A baryon is a hadron composed of 3 quarks, therefore, since the spin and spatial wave functions of baryons are symmetric under the exchange of quarks, the **Pauli principle** demands that a new quantum number is introduced which

discriminates between the three quarks of the baryons. There need to be at least 3 colors to make an antisymmetric wave function. Through experimental measurements of the **R-ratio**, the ratio of hadronic cross-sections to the muon pair cross-section

$$R = \frac{\sigma(e^+e^- \to hadrons)}{\sigma(e^+e^- \to \mu^+\mu^-)}, \tag{2.1}$$

they found the R-ratio to be consistent with exactly 3 colors. All hadrons have to be non-colored. The color charges of the quark bind the quarks within hadrons.

To do precision studies of the internal structure of hadrons, one can do deep inelastic scattering of electrons on protons. The results indicated that the electron did not interact with the proton as a whole, but with point-like constituents inside the proton. To explain these experimental results, **the parton model** was introduced. The parton model assumed that in high energy interactions, hadrons behave as made up of almost free constituents, partons, which carry a fraction of the hadron momentum.

QCD is a successful theory of the strong interaction, which unifies the quark and parton models. The theory describes the strong interaction between fermions, **quarks**, mediated by the gauge bosons, **gluons**. The gluons are massless mediators, but they do carry a color charge. This gives rise to the self-interactions of the gluons.

2.1.1 The QCD Lagrangian

QCD is a quantum field theory that describes the strong interaction between quarks, mediated by gluons. QCD is a non-Abelian theory and can be described by the $SU(N_C = 3)$ symmetry group, where $N_C = 3$ is the number of color charges. The Lagrangian of QCD is defined as

$$\mathcal{L}_{QCD} = \bar{\psi}^f (i\not{D} - m_f)\psi^f - \frac{1}{4} G_{i\mu\nu} G_i^{\mu\nu} \tag{2.2}$$

where f is a flavor index [11]. We follow the notation where the metric is given as $g^{\alpha\beta} = (1,-1,-1,-1)$ and use natural units, with $\hbar = c = 1$. The field strength $G_i^{\mu\nu}$ of the gluon field A is defined as

$$G_i^{\mu\nu} = \partial^\nu A_i^\mu - \partial^\mu A_i^\nu + g_s f_{ijk} A_i^\mu A_k^\nu, \tag{2.3}$$

where $i, j = (1,2,...,8)$ and f_{ijk} is the structure constants.

The covariant derivative is defined as

$$i\not{D} = \gamma_\mu D^\mu, \quad D^\mu \psi^f = (\partial^\mu + i g_s T^a A_j^\mu)\psi^f. \tag{2.4}$$

where T^a is the generators for the $SU(3)$ symmetry group. The covariant derivative couples the quark field with a coupling strength $g_s = \sqrt{4\pi\alpha_s}$ to the gluon field using the $SU(3)$ generators. The fine structure constant α_s is often referred to as the "coupling".

The $SU(3)$ group is a Lie group. The generators of the group are defined from the traceless hermitian Gell-Mann matrices λ_a [11]. Since the symmetry group is a non-Abelian Lie group, then the generators T^a has to satisfy the Lie Algebra. They have to obey the relations

$$[T^a, T^b] = i f^{abc} T^c, \tag{2.5}$$

where f_{abc} is the structure constants. The structure constant is a completely anti-symmetric tensor that we can normalize as $\sum_{c,d} f^{acd} f^{bcd} = N\delta^{ab}$, where N is the number of colors. From this it follows that the generators take the form of

$$T^a = \frac{1}{2}\lambda_a \quad , \quad (a = 1,2,...,8). \tag{2.6}$$

These generators define the fundamental representation of the $SU(3)$ group. A basis-independent way of characterizing different representations of the generators is through the quadratic Casimir $C_2(R)$ defined by

$$T_R^a T_R^a = C_2(R)\mathbb{I}, \tag{2.7}$$

where R indicates the representation. For a given $SU(N)$ group the index T_R and the quadratic casimir C_R of a given representation R is given by

$$T_A = C_A = N \quad , \quad T_F = \frac{1}{2} \quad , \quad C_F = \frac{N^2-1}{2N} \tag{2.8}$$

where F is the fundamental and A is the adjoint representation of the group. For $SU(3)$ this gives $C_A = 3$ and $C_F = 4/3$. These quantities are often referred to as color factors.

The Lagrangian in Eq. (2.2) is not gauge fixed and not suitable for quantization. To be able to perform perturbative calculations on the Lagrangian, we need to introduce a gauge fixing term.This is done using the Faadeev-Popov method [11], which fixes the choice of gauge by adding a gauge-fixing term to the Lagrangian

$$\mathcal{L} = \mathcal{L}_{QCD} - \frac{1}{2\xi}(\partial_\mu A_i^\mu)^2, \tag{2.9}$$

with the gauge parameter ξ. Since QCD is a non-Abelian gauge theory, another gauge-fixing term must be added to the Lagrangian. The additional term is called a ghost Lagrangian. The Lagrangian becomes

$$\mathcal{L} = \mathcal{L}_{QCD} - \frac{1}{2\xi}(\partial_\mu A_i^\mu)^2, + \partial_\mu \eta^{i\dagger}(D_{ij}^\mu \eta^j), \tag{2.10}$$

where η^i is a complex scalar field that obeys Fermi statistics. The ghost field is a result of the Faadeev-Popov method. The ghost field only appears in higher-order calculations and cancels unphysical degrees of freedom, which otherwise would propagate in covariant gauges [11].

The new Lagrangian is sufficient to define the Feynman rules. We can separate the Lagrangian into a free part $\mathcal{L}_0$ and the separate interacting parts of the Lagrangian for the quark, gluon, and ghost fields,

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_{I,quark} + \mathcal{L}_{I,gluon} + \mathcal{L}_{I,ghost}. \tag{2.11}$$

From this, we find the Feynman rules for QCD on page 10 in [11]. Using the Feynman rules, one can make Feynman diagrams, calculate the transition amplitude of most Feynman diagrams and derive the parton splitting functions.

Some of the most noticeable differences between QCD and quantum electrodynamics (QED), the theory of the interactions of charged particles with the electromagnetic

Figure 2.1: Tree level Feynman diagrams of 3-gluon and 4-gluon interaction.

field, is the gluon self-interaction. Unlike the photons that are the mediators of QED interactions, gluons have color charge and are therefore able to interact with other gluons. In Figure 2.1 we see the 3- and 4-gluon vertices. These appear in the last term of the QCD Lagrangian in Eq. (2.2) as a consequence of the gluon carrying color charge. We can ignore the 4-gluon vertex for tree-level calculations, however, they cannot be neglected in higher-order diagrams and loop corrections.

2.1.2 Coupling and Confinement

In field theories, the **coupling constant** describes the strength of a given interaction, relative to free fields. For QED the coupling constant is the fine-structure constant $\alpha \simeq \frac{1}{137}$ which is proportional to the electric charge squared, which we can interpret as how strongly an electrically charged particle interacts with an electric field.

For QCD, the coupling constant g_s only appears in the interaction part of the Lagrangian. Higher order interactions scale with higher powers of g_s. The QCD Lagrangian contains the coupling constant g_s in the first order. This is only valid for tree-level calculations in perturbation theory and leads to divergences in higher-order loop corrections. This needs to be corrected with renormalization. Renormalization allows us to replace the coupling constant with a renormalized coupling constant $g_r(\mu)$, where μ is a scale dependence. The new coupling is defined similarly to the fine-structure constant, so that $\alpha_s(\mu) = \frac{g_r^2}{4\pi}$. The new coupling can be written, in leading order (LO), as

$$\alpha_s(\mu) = \frac{2\pi}{\beta_0 \ln(\mu^2/\Lambda_{QCD}^2)} \tag{2.12}$$

where β_0 is the leading order expansion of the β-function, which decides how the coupling changes with scale. The β-function is responsible for the running of the coupling constant. Λ_{QCD} is the scale at which the coupling constant becomes infinite, and for QCD this scale is typically $\Lambda_{QCD} \sim 0.2GeV$ [12].

Figure 2.2 illustrates the coupling strength of QCD and QED at different momentum transfer scales. The regions of confinement and asymptotic freedom are highlighted. The running coupling decreases at large momentum transfer, i.e., when μ is large, and that phenomenon is known as **asymptotic freedom**. This implies that strong interactions are computable in perturbation theory if the energy scale is sufficiently high. In high energy regimes or at short distances, the quark and gluons would move as quasi-free particles, called partons.

Quarks and gluons interact weakly at high energies, while their interaction grows as the energy decreases until they reach the confinement of particles carrying non-

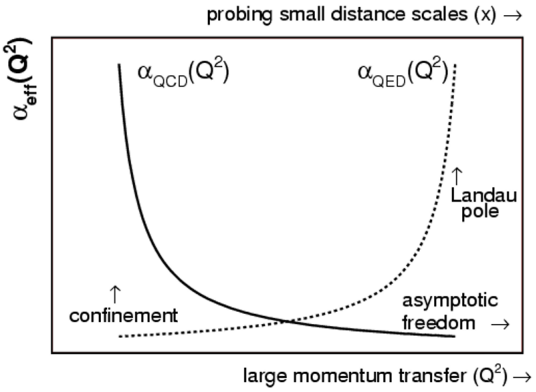


Figure 2.2: QCD and QED coupling strength. Figure from [13]

zero color charges. The running coupling suggests that perturbative techniques break down when the partons are confined [11]. **Confinement**, or color confinement, is the phenomenon that particles with color charge do not exist as isolated particles under normal conditions, i.e., low energies. As a consequence, quarks, and gluons are forced to be bound into color-neutral states called hadrons [11]. An example is the mesons, which are color-neutral hadrons composed by a $q\bar{q}$ pair.

Another effect of confinement is **pair production**. The strong coupling between a pair of color charges is constant regardless of their separation. Therefore, as two color charges are forcefully separated, the potential between them would become so large that it at some point becomes energetically favorable for a new $q\bar{q}$ pair to appear. The new pair will then combine with the existing pair, creating two mesons.

2.2 Colliders

The best way to probe QCD processes is through colliders. There are several ways one can use colliders to explore QCD. One can use e^-e^+ collisions, deep inelastic lepton-hadron (e.i. e^+p) scattering, proton-proton (pp) collisions, or heavy ion collisions. The common idea is that two or more fundamental particles interact with each other and that the resulting particles further interact via the strong nuclear force with the available particles.

Colliders like the LHC have been used to study the internal structure of composite particles, explore interactions between particles, and do precision measurements to experimentally determine observables such as their mass and spin.

Let us use pp collisions as an example and take a short look at the process of colliding two particles. As we know from QCD, partons are composite particles and consist of several partons, each carrying a fraction of the total proton energy. In a pp collision, the partons in the two protons interact with each other through a large momentum transfer, called the **hard process**. The hard process often leads to the creation of new partons. The resulting partons are highly energetic and will continue to interact with the environment and branch out while they lose energy. When the resulting partons have lost enough energy to reach the confinement scale, they are bound together to form color-neutral particles through the process of hadronization. Hadronization is a non-perturbative process. The hadrons produced during hadronization can be measured

in detectors.

In the hard process, large momentum transfers and interaction scales are explored. New heavy particles can be created, and new interactions can be tested. However, it is not only the hard processes that take place during the collisions. There is also measurable radiation from the particles in the protons that did not contribute to the hard process. As protons collide in bunches, multiple pp collisions can occur simultaneously. While one of these many events might be interesting, the other simultaneous events can typically create noise in the detector and make reconstruction of the event difficult. The resulting particles from the simultaneous events are called **pileup**.

In the aftermath of the collision, we look for interesting hard events to study. To study the hard events, we have to compare the experiments to theoretical predictions and look for anomalies or similarities between the two. However, the theoretical description of high-energy collisions of protons is quite complex. In a typical event, hundreds of particles are produced. The high energy part of the process can be computed using perturbation theory. However, the low energy part, the hadronization, is non-perturbative. We still use perturbative calculations on these processes because even though hadronization does modify the final state particles, it occurs so late in the process that it does not modify the original probability for the event to happen.

Let us show the importance of perturbative calculations by taking a look at one of the most important measurable quantities, namely the **cross-section**. The cross-section is a measure of the probability that a specific process will take place when two protons in the particle beams of a collider intersects. The hard scattering of two protons can be described by the hard scattering of quarks and gluons inside the proton. The cross-section to produce a parton k can then be written as

$$\sigma^k = \sum_{i,j} \int dx_1 dx_2 f_i(x_1, Q^2) f_j(x_2, Q^2) \hat{\sigma}^{ij\to k}(\hat{p}_1, \hat{p}_2, \alpha_s(Q), Q), \tag{2.13}$$

where p_1 and p_2 are the four momenta of the colliding protons, x_1 and x_2 are the fractions of the interacting partons defined by $x_1 \equiv \hat{p}_1/p_1$. Q is the characteristic scale of the scattering, which is the transverse momentum exchanged between the two incoming partons. $f_i(x, Q^2)$ are the parton distribution functions (PDFs) defined at a scale Q. $\hat{\sigma}^{ij\to k}$ is the scattering cross-section of two interacting partons to produce a parton k. This parton will then subsequently fragment and form a jet through the processes described in Chapter 3, which can be measured in experiments.

The partonic cross section can be calculated **perturbatively.** The cross-section is usually approximated by a perturbative series in powers of the strong coupling, a fixed-order expansion [10]. Theoretical precision is achieved by computing cross-sections σ including increasingly higher order corrections in the strong coupling α_s

$$\sigma(v) = \sigma_0 + \alpha_s \sigma_1 + \alpha_s^2 \sigma_2 + \mathcal{O}(\alpha_s^3) \tag{2.14}$$

where v is a dimensionless, generic observable. In Eq. (2.14) the leading order (LO) contribution σ_0 is the scattering process that produces one parton in the final state and is a proxy of a simple jet. The other contributions in the perturbative expansion σ_i constitute the next-to^i-leading order (N^{i}LO) corrections. In terms of Feynman diagrams, each power of α_s corresponds to the emission of a QCD parton in the final state or to a virtual correction.

The calculation of higher-order Feynman diagrams suffers from the appearance of **ultraviolet divergences**. However, QCD is a renormalizable theory, which allows for the absorption of these infinities into a redefinition of the Lagrangian's parameters, such as the strong coupling constant α_s.

In fixed-order perturbation calculations, the presence of soft emissions and collinear splittings leads to **infrared divergences**. To ensure the calculations are free from these divergences, it is necessary to consider observables that are **infrared and collinear safe** (IRC-safe). IRC-safety implies that if an event is modified by a collinear splitting or the addition of soft radiation, the hard splittings within the jet should remain unchanged [14]. Collinear splittings can occur due to non-perturbative dynamics, and soft emission can occur through both perturbative and non-perturbative effects.

The goal of IRC-safety is to provide a framework for studying events that is less sensitive to these effects. It is important to note that in practice, non-perturbative effects such as hadronization regulate the soft and collinear divergences [10]. The requirement of IRC safety allows for reliable computations of observables in perturbative QCD, with non-perturbative corrections taken into account. However, there are also interesting observables that are IRC-unsafe, which requires the inclusion of non-perturbative functions to describe their soft and collinear behavior.

2.2.1 Collider Kinematics

Now that we have an idea of a collision works, we wish to define some IRC safe observables related to pp-collision.

When describing a particle in the detector, we need to know the four-momentum of said particle. We typically describe the particle's momentum in the detector by the **azimuthal angle** ϕ and the **rapidity** y. The azimuthal angle is defined in the transverse plane (p_x, p_y), which takes the values between $\{0, 2\pi\}$. The four-momentum can be described by the energy and momentum vector, or by the positions ϕ and y, as well as the **transverse momentum** p_T. This way, we can represent the momentum four-vector of a particle with mass m as

$$p^\mu = (E, p_x, p_y, p_z) = (m_T \cosh y, \quad p_T \cos\phi, \quad p_T \sin\phi, \quad m_T \sinh y), \tag{2.15}$$

where p_T is the transverse momentum, and the transverse mass is defined as $m_T = \sqrt{p_T^2 + m^2}$. From this definition of the four-momentum, we can define the new observables. The transverse momentum p_T is defined as

$$p_T = \sqrt{p_x^2 + p_y^2}, \tag{2.16}$$

and the rapidity y as

$$y = \frac{1}{2}\log\left(\frac{E + p_z}{E - p_z}\right). \tag{2.17}$$

It is more common to use **pseudorapidity** η instead of rapidity, which we define as

$$\eta = -\log\left(\tan\frac{\theta}{2}\right), \tag{2.18}$$

where θ us the polar angle with respect to the beam. For massless particles rapidity and pseudorapidity are identical. One can define a distance between two particles inside the detector, in the (y, ϕ) plane, as

$$\triangle R = \sqrt{\triangle y^2 + \triangle \phi^2}. \tag{2.19}$$

The distance between particles will be very useful when discussing jets in the following chapters.

2.3 Heavy Ion Collisions

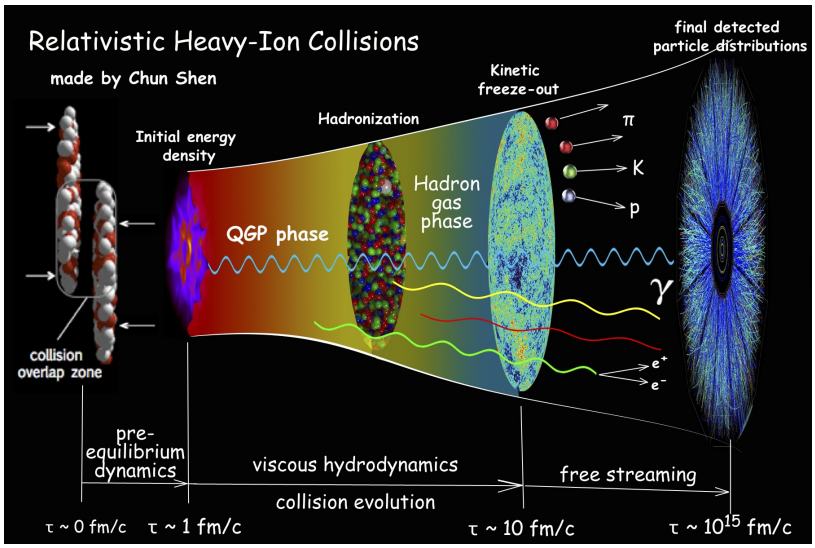


Figure 2.3: Sketch of relativistic heavy ion collisions. Figure from [15].

In **relativistic heavy-ion collisions**, we collide massive nuclei such as lead *Pb* or gold *Au*. This is done at the LHC and RHIC colliders. The goal of such collisions is to create and study quark-gluon plasma (QGP).

Figure 2.3 shows a sketch of a relativistic heavy-ion collision and shows the multiple stages of the collision. First, the heavy ions are accelerated to relativistic velocities, which flatten the nuclei like pancakes. When the pancaked nuclei collide, multiple particles smash together creating a hot, dense plasma of quarks and gluons, a QGP. The dynamics of QGP can be described using hydrodynamics. As the system expands and cools, it will cross over from the QGP phase to a hadron gas phase, through hadronization. As the QGP continues to expand and cool, the collision rates between the hadrons decrease and the system reaches kinetic freeze-out. After the freeze-out, the particles can be detected by the detectors and studied.

When QGP is present, any particles propagating through the plasma will be **modified** by elastic and inelastic processes that take place during their passage through the plasma [5]. By measuring the final state particles after they traverse the medium, one can study both the plasma itself and the QCD processes that occurred due to the collision.

While we will return to how the plasma affects the particles propagating through it in the next chapter, we will now introduce the QGP and one of the models to simulate such a thermal background.

2.3.1 Quark-Gluon Plasma

As discussed in the section about **confinement**, particles with color charge cannot exist as isolated particles under normal temperature and pressure conditions. However, as the energy scale increases, we observe the phenomenon of **asymptotic freedom**. With rising temperature T or baryonic chemical potential μ_B, a phase transition occurs where hadrons cease to exist.

The state of matter where quarks and gluons can move freely is known as **quark-gluon plasma** (QGP). We can expect to find QGP in three places: in the early universe, at the center of compact stars, and in the initial stage of energetic heavy ion collisions [16]. The easiest of these options to precisely explore is heavy ion collisions.

Although QGP is referred to as a **plasma**, it is unclear whether it should be interpreted as a weakly interacting gas or a strongly interacting fluid. At sufficiently high temperatures the QGP must be weakly coupled, however, in the temperature range explored by current colliders, we know from the comparison of precisely measured experimental observables and sophisticated calculations of relativistic hydrodynamics that QGP produced in heavy ion collisions is a **strongly coupled liquid** that expands and flows hydrodynamically [17]. This makes the QGP a very interesting form of matter, but it also complicates the theoretical understanding of its properties and dynamics.

QGP is short-lived and difficult to measure directly. The main method to study QGP is to probe it using jets, or highly energetic particles. Jets lose energy when traveling through a QGP, in a process called **jet quenching**. Jets are "hard probes", meaning they are strongly interacting but moving so fast and with so much energy that they are often not completely absorbed by the surrounding quark-gluon plasma. The degree of jet quenching, the jet properties, and how the energy is transferred to the medium reveal the properties of the QGP. These kinds of experiments are currently done at the particle accelerators RHIC at Brookhaven National Laboratory and LHC at CERN.

The first evidence for jet quenching was seen in 2003 in experiments at RHIC [18]. Recently at the LHC, they have been reaching higher collision energies and doing more precise measurements of the QGP, using jets [18]. However, the theoretical understanding of these measurements is challenging, and we will discuss the theoretical understanding of jet quenching in the next chapter.

2.3.2 Blast Wave Model

Heavy-ion collisions are extremely complex, and there are several models trying to explain one or more features. One of these models is the blast wave model [19]. The blast wave model describes the freeze-out phase of the QGP, which creates a thermal background of hadrons originating from the QGP.

For an ideal hydrodynamical fluid, the local momentum distribution of hadrons should be a thermal distribution, for example, a Bose-Einstein distribution. The fluid is not at rest, thus the thermal distribution must be boosted to the laboratory frame. One model for such a boosted distribution is the blast wave model where the distribution of momentum is

$$\frac{dN_i}{m_t dm_t dy} = A_i m_t K_1\left(\frac{m_t \cosh\rho}{T_F}\right) I_0\left(\frac{p_t \sinh\rho}{T_F}\right) \tag{2.20}$$

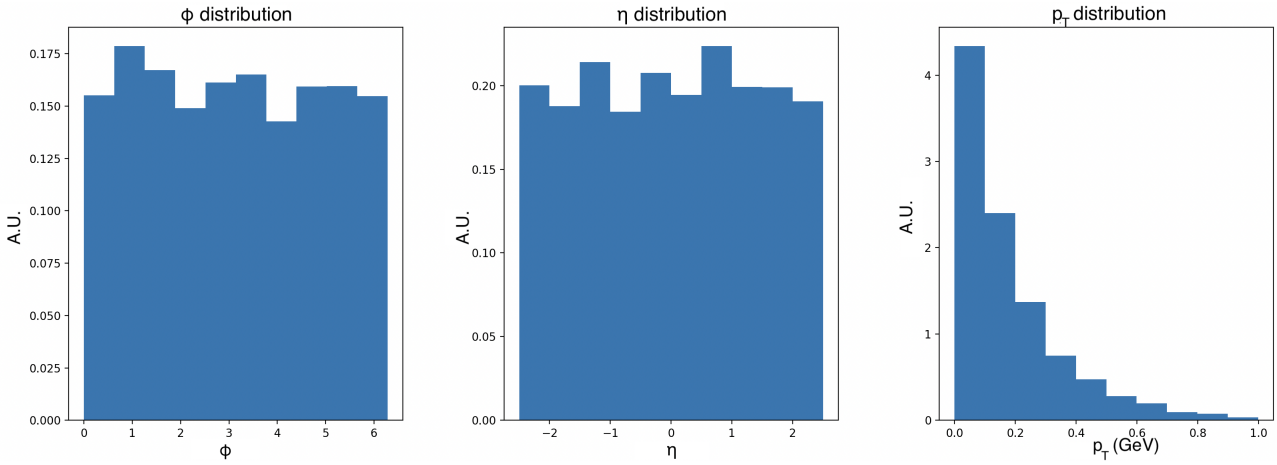


Figure 2.4: Distribution of ϕ, η and p_T, using $\beta_T = 0.63$ and $T_F = 100MeV$ to obtain 1000 background particles in each rapidity slice.

where A_i is a normalization constant, $m_t = \sqrt{p_t^2 + m^2}$, K_1 is the modified Bessel function of second kind, I_0 is the modified Bessel function of the first kind, $\rho = \tanh^{-1}\beta_T$, T_F is the "true temperature" and β_T is the transverse expansion velocity.

Depending on how many background particles we wish to create, we use figures 5 and 6 in [20] to find the value of β_T and T_F. For example, we wish to create 1000 particles per unit rapidity η in the aftermath of a heavy ion collision at the LHC, which collides PbPb. In Figure 5 in [20], we find the marker for PbPb collisions and read $\beta_T \sim 0.62$ and in Figure 6 in [20] we find $T_F \sim 100MeV$.

In our model, we wish to create background particles with zero mass, $m = 0$. Using a uniform distribution for ϕ between $\{0, 2\pi\}$ and a uniform distribution for η between $\{-2.5, 2.5\}$, we find the distribution of p_t of the background particles from Eq. (2.20). We use the Monte Carlo Accept-Reject Method [21] when we sample from the blast wave model to obtain a p_t value.

The Accept-Reject method is quite simple. The probability distribution is normalized so that the largest value is $dN_i/dp_t = 1$. You also sample a number from a uniform distribution between $\{0, 1\}$. First, you sample a p_t value, between $\{0, 1000\}$MeV, and get the probability of getting that number from the blast wave probability distribution. Then we check if this probability is larger than the random number we generated. If that is the case, we keep the sampled value as our p_t. Otherwise, we resample from our distributions until we get an accepted sample.

By sampling from these distributions and getting a ϕ, η and p_T, we create the 4-momentum for the background particles using Eq. (2.15). In Figure 2.4 we have plotted the distribution of the ϕ, η and p_T after generating 1000 background particles per rapidity slice with the blast-wave model. Based on this distribution of background particles the average transverse momentum is about $< p_t >= 0.17$GeV, which is at the scale of the temperature $T = 0.1$GeV. A more realistic average transverse momentum would be $< p_T >= 0.4 - 0.5$GeV.

By using this method to generate thermal background particles, we can embed jets in a heavy-ion environment and simulate real experimental conditions where the removal of background particles is difficult.

The Key Concepts of This Chapter

QCD is the theory of the **strong interaction** between quarks, mediated by gluons. The **coupling constant** describes the strength of the interactions. The coupling strength depends on the energy scale of the interaction. At high energy scales, we reach **asymptotic freedom**, where quarks and gluons can move as free particles, called partons. At small energy scales, the partons are **confined** and bound into color-neutral hadrons. The best way to study QCD processes is through collider experiments. Proton-proton collisions are described using **perturbation theory**.

Relativistic heavy-ion collisions are the collision of heavy nuclei such as lead. When large nuclei are smashed together, they create a hot, dense **quark-gluon plasma**. Quark-gluon plasma is a state of matter where quarks and gluons can move freely. QGP is described by hydrodynamics as a **strongly coupled liquid**. QGP is difficult to measure directly and is often studied using **jets** as hard probes. There are many models to simulate QGP, such as the blast-wave model which simulates the thermal particles created by the hadronization of the quark-gluon plasma.

Chapter 3

Jets

High-energy quarks and gluons are not directly observable but instead undergo a series of successive branchings at small angles, resulting in a cascade of collimated partons known as a **parton shower**. When the parton shower reaches the scale of hadronization, the collimated cluster of hadrons is referred to as a "jet." Jets are directly detected in the detectors, where the hadronic final state appears as collimated structures in the (y, ϕ) space within the detector.

The branching and evolution of the parton shower give rise to the characteristic radiation patterns observed in jets. In the presence of a medium, such as in heavy-ion collisions, each parton in the parton shower interacts with the medium, leading to energy loss in the jet. This phenomenon is known as **jet quenching**. Jet quenching can be described using various modeling approaches. Due to the contrasting nature of the strongly coupled quark-gluon plasma and weakly coupled QCD processes involved in parton shower formation, a hybrid model can be employed to account for the physics at different energy scales.

Partons are not well-defined objects, due to higher-order QCD corrections. Therefore, whether two particles are part of the same jet, or belong to different jets has some degree of arbitrariness, depending on what we mean by “collimated”. The simple concept of what a jet is meant to represent is therefore not sufficient to identify the jets in an event. To do that, one relies on a jet definition, a well-defined procedure that tells us how to reconstruct the jets from the set of hadrons in the final state of the collision.

This chapter begins by discussing jets in vacuum, and how the branching evolves, then moving on to a discussion of jets in a medium where we delve into one hybrid model of jet quenching. Afterward, the jet observables are introduced, focusing on the Lund plane. Finally, the relevant jet definitions are introduced, as well as the jet substructure tools that will be used in this paper.

3.1 Jets in Vacuum

In some regions of phase space, we cannot neglect higher-order terms in our perturbative calculation. Instead of aiming for a precise prediction to some fixed order perturbation theory, we seek an approximate result where terms of all orders are considered [11]. This led to parton shower pictures which can be implemented in computer simulations, such as Monte Carlo event generators.

A parton shower relies on the factorization of a process into a hard scattering, a perturbative shower, describing soft and collinear emissions, and a non-perturbative hadronization process. Parton shower generators can generate fully exclusive events consisting of hadrons, allowing in principle arbitrary questions to be asked about the final state [22]. However, with this generality comes a loss of theoretical precision compared to calculations, as well as a reliance on models.

To understand the creation of parton showers and jets, we need to understand the processes of parton branching and parton evolution.

3.1.1 Parton Branching

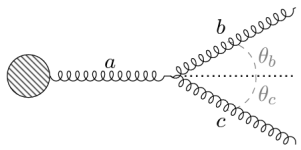


Figure 3.1: Branching of an outgoing gluon a from some initial blob, into two gluons b, c.

Parton branching is the process of an energetic parton splitting into two new partons. This can happen via splitting and radiation, which allows for the number of partons in the jet to increase and lead to a parton shower. When discussing parton branching, we generally mean soft and collinear branching. In soft branching, the emitted parton carries a very small transverse momentum fraction, z, relative to the parent parton. In collinear branching, the emitted parton travels in roughly the same direction as the parent parton, which implies that the opening angle θ is very small. The branching is illustrated in Figure 3.1. The momentum-sharing fraction z can be defined as

$$z = \frac{E_b}{E_a} = 1 - \frac{E_c}{E_a} \tag{3.1}$$

The opening angle in the small angle limit is given as

$$\theta = z\theta_b + (1-z)\theta_c \tag{3.2}$$

Without any medium present, the parton shower will be angular ordered, meaning the angle between each successive splitting decreases. Angular ordering leads to resolvable branchings, meaning we can identify each branch in the shower and where it branched from [11]. This also implies that color charge is conserved along the parton shower. This is useful when we simulate the parton shower using a Monte-Carlo event generator.

3.1.2 Parton evolution

We can introduce the **parton distribution function** (PDF) $f(x,t)$ that we mentioned earlier, which represents the probability to find a parton with a momentum fraction x of the original momentum p, at a given time in the evolution of the jet. The probability that a parton j emits a collinear parton i, with a momentum fraction z, is described by the

Altarelli-Parisi splitting functions $P_{ij}(z)$ [23]. The method of deriving the splitting functions is outlined in section 5 in [11]. The resulting splitting functions is,

$$P_{gg} = C_A[\frac{1-z}{z} + \frac{z}{1-z} + z(1-z] \tag{3.3}$$

$$P_{qg}(z) = n_f T_R[z^2 + (1-z)^2] \tag{3.4}$$

$$P_{qq}(z) = C_F \frac{1+z^2}{1-z} \tag{3.5}$$

$$P_{gq}(z) = P_{qq}(1-z) = C_F \frac{1+(1-z)^2}{z} \tag{3.6}$$

where the color factors are $C_A = 3$, $C_F = 4/3$ and $T_R = 1/2$. The factor $n_f = 5$ is the number of active quark flavors and represents the probability of a gluon emitting a $q\bar{q}$-pair with equal probability for all flavors. The probability of emitting a gg-pair from a quark is the same regardless of its flavor and also, when a quark emits a gluon there is no flavor change. This is all under the assumption that the quarks are massless [23].

This parton distribution $f(x,t)$ is described by the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (**DGLAP**) evolution equations [11]. The DGLAP equations read [24]

$$f(x,t) = \triangle(t) f(x,t_0) + \int \frac{dt'}{t'} \frac{\triangle(t^2)}{\triangle(t')} \int \frac{dz}{z} P(z) f(\frac{x}{z}, t'), \tag{3.7}$$

where $P(z)$ is the Altarelli-Parisi splitting functions where we have dropped the flavor indices, $t = Q^2 \simeq p_T^2$ is the evolution parameter, $f(x,t)$ is the PDF and $\triangle(t)$ is called the **Sudakov form factor**. The Sudakov form factor is defined as

$$\triangle(t) = \exp[-\int_{t_0}^{t} \frac{dt'}{t'} \int dz \frac{\alpha_s}{2\pi} P(z)]. \tag{3.8}$$

The Sudakov is the probability of a parton not decaying between the evolution scales t_0 to t, which in our case is the virtuality. Hence, the DGLAP describes the decreasing virtuality of an initially hard particle. Eq. 3.7 consists of two terms. The first term describes the evolution of a particle if no decay takes place. The second term describes the evolution of a particle if a decay happens between the evolution scale t and Q^2. Formulating the parton evolution in terms of the Sudakov is well suited for computer simulations of jets and parton showers.

As a parton evolves, the time it takes before a decay happens is called the **formation time** t_f. The formation time is defined as

$$t_f = 2\frac{E}{Q^2} \tag{3.9}$$

where the factor $1/Q$ is the time scale of decay for an off-shell particle and the factor E/Q is a boost factor due to time dilation. The formation time of a particle becomes endowed with physical meaning when we consider a particle moving through a medium since the plasma has an extension in space-time.

3.1.3 Monte Carlo Event Generators

Monte Carlo (MC) event generators are used to simulate QCD parton branching. They simulate QCD parton branching by using the DGLAP equations and the Sudakov form factor. In the second term in Eq. 3.7, we have the ratio $\triangle(t)/\triangle(t')$ which tells us the probability of a parton branching between the evolution scales t and t'. This is the basis of the modern MC branching algorithms.

The branching algorithm works as follows: given the evolution parameter t_1 and momentum fraction z_1, after some step in the evolution, generate the values (t_2, z_2) of the next step [11]. This is done by generating a random number $\mathcal{R}$ from a uniform distribution between 0 and 1, and obtaining t_2 by solving

$$\frac{\triangle(t_1)}{\triangle(t_2)} = \mathcal{R}. \tag{3.10}$$

If the value of t_2 is lower than the cut-off value t_0, then there is no further branching [11]. Otherwise, we continue with the next branching and generate a new momentum fraction $z = x_2/x_1$. We can do this by solving the equation

$$\int_{\varepsilon}^{x_2/x_1} dz \frac{\alpha_s}{2\pi} P(z) = \mathcal{R}' \int_{\varepsilon}^{1-\varepsilon} dz \frac{\alpha_s}{2\pi} P(z) \tag{3.11}$$

where $\mathcal{R}'$ is another random number between 0 and 1, $P(z)$ is the appropriate splitting function and ε is the infra-red cut-off for resolvable branching. This applies to timelike parton branching. The values (t_i, x_i) generated define the virtual masses and momentum fractions of the exchanged parton, from which one can compute the momenta of the parton.

Successive timelike branching generates a parton shower. Each branch is the source of a new cascade until the cut-off is reached. At the cut-off scale, the outgoing partons are converted into hadrons via some hadronization model to simulate real-life events. Typical cut-off scales are around $0.2 - 1$ GeV. Hadronization will not be discussed in this thesis.

Numerous parton shower generators exist, each with its own advantages and disadvantages. In this thesis, we use the event generator of PYTHIA8 [8] to generate vacuum events.

3.2 Jets in a Medium

When a particle propagates through a medium, it interacts with the medium, giving rise to complex phenomena such as jet broadening and energy loss. When a jet travels through a medium, such as a QGP, it interacts with the medium and loses energy to the medium, this is known as jet quenching.

3.2.1 Jet Quenching

Jet quenching is the suppression of high-energy jets in heavy-ion collisions. When a jet propagates through a QCD medium, the components of the jet suffer energy loss

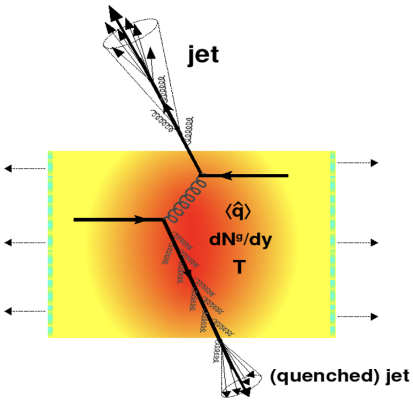


Figure 3.2: Illustration of jet quenching in a head-on nucleus-nucleus collision. Two quarks suffer a hard scattering: one goes out directly to the vacuum, radiates a few gluons and hadronises, and the other goes through the dense plasma created (characterized by transport coefficient $\hat{q}$, gluon density $\frac{dN_g}{dy}$ and temperature T), suffers energy loss due to medium-induced radiation and finally fragments outside into a (quenched) jet. Figure from [25].

while the jet passes through the medium. Figure 3.2 illustrates the quenching of a jet in a heavy-ion collision. The quenching of a jet depends on both the characteristics of the jet propagating through it and on the medium properties.

When measuring jet quenching in experiments, one measures the ratio of the p_T spectrums of a jet from a heavy-ion collision and a pp-collision. This is quantified by the **nuclear modification factor** R_{AA}, defined experimentally as

$$R_{AA} = \frac{1}{n_{AA}} \frac{dN_{AA}}{dp_T d\eta} \Big/ \frac{dN^{pp}}{dp_T d\eta} \tag{3.12}$$

where n_{AA} is the average number of binary nucleon-nucleon collisions, N_{AA} is the average particle multiplicity and N^{pp} is the multiplicity in proton-proton collisions. The presence of jet quenching manifests itself in the **suppression of the p_T spectrum** of high p_T hadrons [26].

High-energy jets are good probes to study jet quenching because they are produced at high energy scales, which guarantees that their production spectrum is under good theoretical control since it is governed by perturbative QCD [27]. Similarly, the properties of jets in a vacuum are also controlled by physics at high energy scales. Therefore, systematic deviations of these properties in a heavy-ion environment must be due to the jet's **interaction with the medium**. In general, interaction with a medium will lead to energy loss in the final jet, but the precise process depends on the nature of the medium.

A **weakly coupled medium** means unrealistic high temperatures at which the coupling constant at the medium scale is so small that the medium can be described as a collection of scattering centers. Then one can describe the interaction between a parton and the scattering centers in the medium using weak coupling techniques. The two techniques for weakly coupled mediums are collisional and radiative energy loss [12], shown in Figure 3.3. The **collisional energy loss** is due to the elastic scattering of an incoming parton with a medium constituent. [12]. Collisional energy loss is the dominant process at low particle momentum. **Radiative energy loss** through inelastic scatterings with the medium dominates in high momentum processes. This process

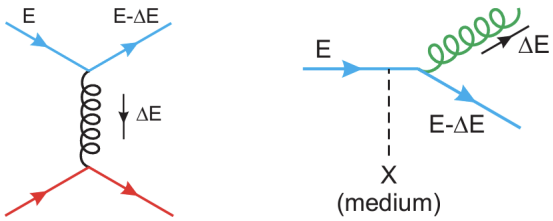


Figure 3.3: Diagrams for collisional (left) and radiative (right) energy losses of a quark of energy E traversing a quark-gluon medium. Figure from [25].

causes the parton to radiate gluons through gluon bremsstrahlung [25].

The production of a hard parton and the fragmentation of that parton is controlled by weakly coupled physics at high momentum scales. However, the physics of the medium produced by heavy ion collisions is not weakly coupled, as previously mentioned the QGP is more like a droplet of **strongly coupled liquid**. To combine the weakly coupled physics of the parton with the strongly coupled physics of the medium, they proposed a new approach called the hybrid strong/weak approach to jet quenching [27].

3.2.2 The Hybrid Strong/Weak Coupling Approach

The hybrid strong/weak coupling approach [27] is a model for the energy loss of a jet traveling through a strongly coupled medium. The model treats physical processes at different energy scales separately. The model treats the weakly coupled dynamics involved with the creation and evolution of jets perturbatively. The dynamics of the strongly coupled medium are obtained via gauge/gravity duality. That allows us to draw quantitative relations between a strongly coupled gauge field theory and a weakly coupled string theory [28].

The weakly coupled processes occur at momentum scales set by the virtuality, $Q \sim p_T$. The strongly coupled process involves momenta at the typical scales that characterize the medium, which is at the order of the temperature T. For high-energy processes involved in heavy-ion collisions at the LHC, these two scales are sufficiently separated so that $Q \gg T$. We use this separation of scales to justify the different treatment of the relevant dynamics at each regime.

The hybrid model is a minimalistic model. It only uses the well-understood weakly coupled and strongly coupled physics, and introduces as few as possible phenomenological parameters. In the hybrid strong/weak model, they focus only on the loss of energy of the partons in the shower which can be modeled with a single free parameter. Since the momentum transfer between the partons and the medium is not large, the physics of energy loss can be described by strong coupling processes.

While the jet evolves inside the medium, its constituents continuously interact with the medium and exchange momentum. In this model, we assume that the exchange of momentum with the medium is so small that it cannot alter the weakly coupled processes of the high-virtuality partons. This assumption holds since the evolution reaches the non-perturbative hadronization scale when almost all the partons in the shower are outside the medium [29]. We also assume that the splitting probabilities are not modified by these soft exchanges.

As a consequence of the soft exchanges of momenta with the medium, the particles in the shower lose energy as they propagate through the strongly coupled medium [30].

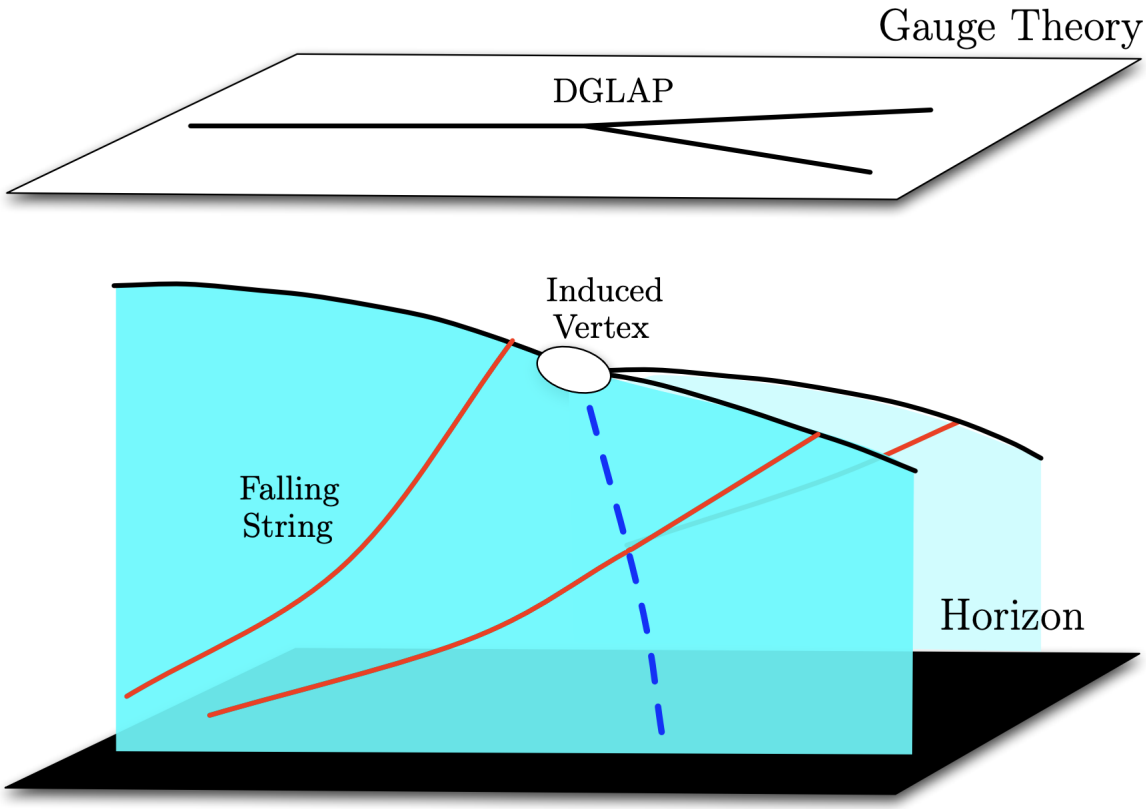


Figure 3.4: Sketch of two views of the interaction of a high energy jet with the strongly coupled plasma. In the gauge theory, represented by the white plane at the top of the figure, an energetic virtual parton propagates through the medium, loses energy, and splits via (vacuum) DGLAP evolution. In the dual gravitational view, represented below, the soft interactions are represented by a string trailing behind each parton, transporting energy from the parton "down" to the horizon which is represented by the black plane at the bottom of the figure. Figure from [30].

In vacuum physics information about when and where anything happens is unimportant since the components of the jet do not interact with anything, they simply fragment. However, in a heavy-ion environment, before the jet emerges from the medium, every parton in the particle shower interacts with the medium, and the medium changes as a function of time and space. Therefore, we need to know when and where each splitting happens. This is where the formation time t_f from Eq. 3.9 comes in. The formation time describes the time it takes for a parton to decay.

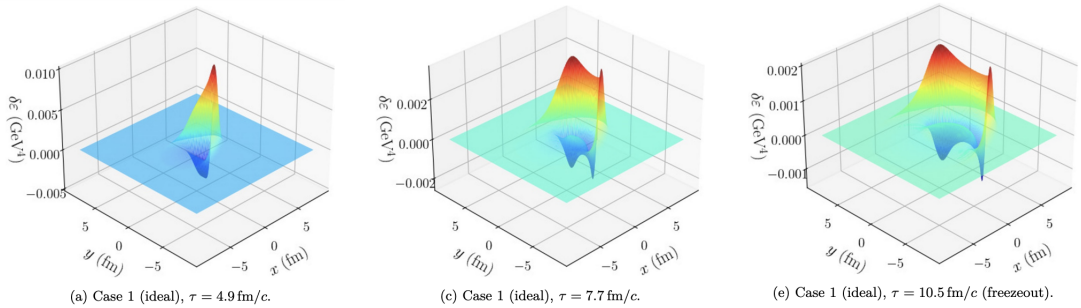
To model this interaction with the medium, we use the gauge-gravity duality. The concept of gauge-gravity duality is illustrated in Figure 3.4. The vacuum interaction and DGLAP evolution happen in 4-dimensional spacetime. If we add a 5th dimension in the y-axis of this plot and say that the bottom plane is the horizon of a black hole, then we can use the **gauge-gravity duality** description. The black hole is a proxy for the medium, where the black hole has the same Hawking temperature T as the medium. The interaction with the medium can be represented by a string trailing behind each

parton, transporting energy from the parton "down" to the horizon. The pull of the string causes the partons to lose energy. The parton itself is represented by the endpoint of the string, and is also pulled "downward" toward the horizon. If a parton gets pulled so far down that it disappears into the horizon, then it is no longer part of the jet but becomes part of the medium.

Due to the gauge-gravity duality, we get this equation for the **energy loss** of a parton per distance x, or time since we are in spacetime after all,

$$\frac{dE}{dx}|_{stronglycoupled} = -\frac{4}{\pi}E_{in}\frac{x^2}{x_{stop}^2}\frac{1}{\sqrt{x_{stop}^2 - x^2}}, \quad x_{stop} = \frac{1}{2\kappa_{SC}}\frac{E_{in}^{1/3}}{T^{4/3}} \tag{3.13}$$

Here, E_{in} is the initial energy of the parton before it enters the plasma, T is the local temperature of the plasma, and x_{stop} is the stopping distance. The stopping distance is the smallest distance traveled through the medium which results in the energetic excitation losing all of its energy. The dimensionless κ_{SC} is taken as a free parameter that is fit to hadron and jet suppression data [31].



(a) Case 1 (ideal), $\tau = 4.9\,\mathrm{fm}/c$. (c) Case 1 (ideal), $\tau = 7.7\,\mathrm{fm}/c$. (e) Case 1 (ideal), $\tau = 10.5\,\mathrm{fm}/c$ (freezeout).

Figure 3.5: Evolution of the wake through time. Figure from [32].

The amount of energy lost by a parton, as described by Eq. 3.13, corresponds to the amount of energy flowing into the QGP [33]. The flow of energy into the QGP generates a **wake** that moves in the same direction as the jet [34]. When energy is deposited into the medium, the energy is spread out through the medium through time as one can see in Figure 3.5. As one can see, the energy is spread out over a larger area of the medium and diluted after some time. As the plasma reaches hadronization, or parton freezeout, the energy of the wake is deposited into thermal particles, using the equation here.

Since the wake dilutes the energy deposited in the medium, then the wake creates many soft particles at wider angles. For a jet with a certain jet radius, this means that a lot of thermal particles are not within that radius after exiting the medium, and that is why not all the energy deposited into the medium is returned to the jet, and the jet in question loses energy.

The contribution from the wake in the hybrid model is estimated by an expansion of the Cooper-Frye formula at the perturbed freeze-out hypersurface, which yields [35]

$$\begin{aligned} E\frac{d\triangle N}{d^3p} = \frac{1}{32\pi}\frac{m_T}{T^5}\cosh(y-y_j)\exp[-\frac{m_T}{T}\cosh(y-y_j)] \\ \{p_T\triangle P_T\cos(\phi-\phi_j) + \frac{1}{3}m_T\triangle M_T\cosh(y-y_j)\}, \end{aligned} \tag{3.14}$$

where p_T, m_T, ϕ and y are the kinematic parameters of the emitted thermal particles, and where $\triangle P_T$ and $\triangle M_T = \triangle E/\cosh y_j$ are the transverse momentum and transverse mass transferred from the jet, with ϕ_j and y_j. The thermal particles are then hadronized using the Lund string model included in PYTHIA [8].

The final state particles after the jet has passed through the medium will contain the hadronized particles from the parton shower after they have been modified and lost energy, as well as the hadronized thermal partons from the wake.

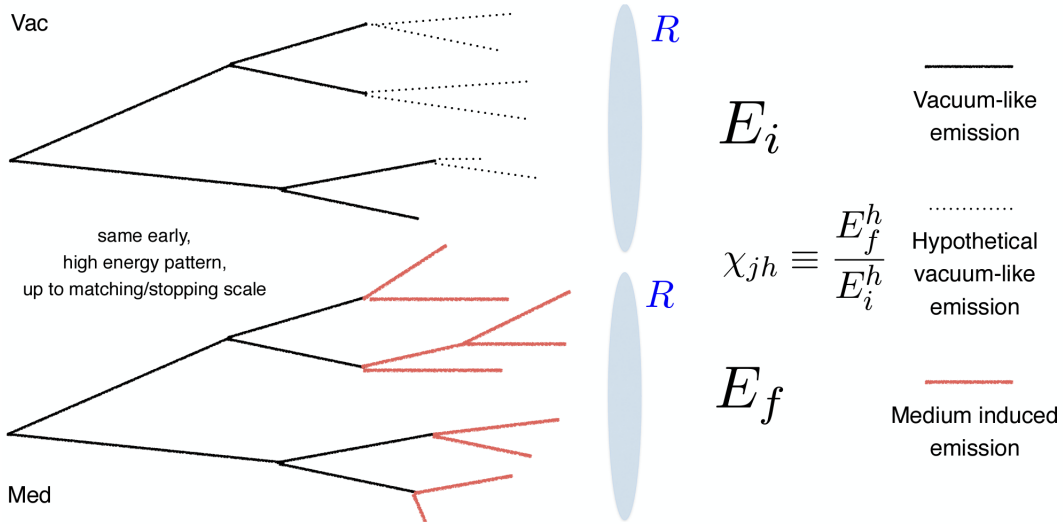


Figure 3.6: Illustration of energy loss in the hybrid model. Figure from [36].

To determine the amount of energy loss suffered by the jet due to the propagation through a QCD medium, we calculate, on a jet-by-jet basis, the energy loss ratio

$$\chi \equiv \frac{E_f}{E_i}, \tag{3.15}$$

where E_f is the final state energy after interactions with a medium p_T and E_i is initial energy. We identified as the initial jet energy as the energy of the corresponding vacuum jet after the matching procedure, with $E_i = p_{T,pp}$. The p_T of the vacuum jet will not be exactly the same as the initial energy, because even here the vacuum jet loses some energy in the branching. An example of a vacuum jet and a corresponding medium-modified jet can be found in Figure 3.6. To ensure we find the corresponding jets, we perform the matching procedure explained here:

Given a medium-modified jet of energy, $E_f = p_{T,PbPb}$, we find its vacuum partner with the following **matching procedure**:

1. Create the vacuum jets by clustering the list of vacuum hadrons, without medium modifications.
2. Create the medium jets by clustering the list of medium-modified hadrons, including the hadrons from the hadronized parton shower and the wake.
3. For each medium jet, get its vacuum partner by selecting the highest p_T vacuum jet whose axis is within $\triangle R < 0.4$ from the medium jet axis.

The procedure described above is for hadronic jets, but the matching procedure is analogous for partonic jets.

3.3 Jet Definitions

A jet definition is composed of two building blocks: the **jet algorithm** and a **recombination scheme**. The jet algorithm is the recipe and a set of parameters associated with the algorithm. A typical parameter is the **jet radius**, R, which provides a distance in the (η, ϕ) plane where particles outside the radius are not considered part of the jet. A recombination scheme specifies how the kinematic properties of the jet are obtained from its constituents. Most jet definitions today use the **"E-scheme"**, which simply sums the components of the four-vector.

Some general properties a jet definition should have were proposed by the *Snowmass accord*, which was published in 1990 [37] by a group of theorists and experimentalists. The Snowmass accords contain the fundamental criteria that any jet definition should satisfy, and reads as follows,

1. Simple to implement in experimental analysis.
2. Simple to implement in the theoretical calculation.
3. Defined at any order of perturbation theory.
4. Yields finite cross sections at any order of perturbation theory.
5. Yields a cross-section that is relatively insensitive to hadronization.

There is no one standard jet definition that suits all scenarios. Multiple jet algorithms exist, each serving different purposes. There is no single source of information about all the different algorithms, and it is not always evident how well the algorithms follow the Snowmass accords. The sensitivity of various jet definitions to non-perturbative effects, such as pileup and detector effects, has been extensively debated in the context of selecting a jet definition at the LHC [10].

Jet algorithms can be categorized into two main groups: **Cone algorithms** and **sequential recombination algorithms**.

Cone algorithms can be imagined as "top-down" algorithms, relying on the notion that jet energy flows into a cone-shaped region. In the past, experimentalists favored cone algorithms, but they were not as popular among theorists due to their lack of IRC-safety, potentially leading to unphysical outcomes.

Sequential recombination algorithms are "bottom-up" techniques based on the fact that jets result from successive parton branchings. These algorithms try to invert this process by iteratively combining pairs of particles into a single object. This recombination is determined by a **distance measure**, which is minimized when the desired kinematic property is most favorable. Jets are formed by iteratively clustering particles that minimize the distance measure, mimicking the progression of a parton shower.

The most popular sequential recombination algorithms are the **generalized k_T algorithms** [9]. These algorithms take a list of the particles in the event and cluster them with the following algorithm:

1. From the list of objects, create two distance measure metrics: An inter-particle distance

$$d_{ij} = \min\{p_{T,i}^{2p}, p_{T,j}^{2p}\}\frac{\triangle R_{ij}^2}{R^2} \tag{3.16}$$

where p is a free parameter that we will return to shortly, $\triangle R_{ij}$ is the angular distance between the particles in the (η, ϕ) plane and R is the jet radius, and a beam distance

$$d_{iB} = p_{T,i}^{2p}. \tag{3.17}$$

2. Iteratively find the smallest distance among the d_{ij} and d_{iB}

   - If the smallest distance is a d_{ij} then object i and j are recombined into a new object k using the recombination scheme. The objects i and j are removed from the list of objects and k is added.
   - If the smallest distance is a d_{iB} then object i is called a jet and removed from the list of objects.

3. Repeat until left with only one object.

We see that two objects close in the (η, ϕ) plane have small distances d_{ij} and are more likely to recombine. This aligns with our understanding of collinear splittings in parton showers, where recently split particles exhibit small inter-particle distances. The choice of the free parameter p allows the distance measure to emphasize various kinematic properties. The most popular algorithm is the k_T algorithm when $p = 1$, the Cambridge/Aachen algorithm, when $p = 0$, and the anti-k_T algorithm when $p = -1$.

The k_T **algorithm** is historically the best-known algorithm in the generalized-k_T family of algorithms. In that case, soft emission will be associated with a small distance and therefore recombine early in the clustering process. The sensitivity to soft emission is desirable from a perturbative QCD standpoint, however, has the disadvantage that jets become sensitive to background radiation and pileup.

Another important algorithm is the **Cambridge/Aachen algorithm** (C/A). In this case, the distance measure becomes purely geometrical, and the algorithm is less sensitive to background radiation than the k_T algorithm.

At the LHC, jets are almost exclusively identified with the **anti-k_T algorithm**, corresponding to $p = -1$. This algorithm favors hard particles, which will cluster first. A hard jet will successively combine soft particles around it until it has reached a distance R away from the jet axis. This means that the hard jets will be insensitive to soft radiation. Since the algorithm is less sensitive to soft radiation, the anti-k_T algorithm allows calibration in experimental contexts, and that is the main reason it is the default jet algorithm at the LHC experiments.

In jet substructure studies, several jet algorithms are used. Typically, the jets are initially reconstructed using the anti-k_T algorithm with a large radius. Then, many of the jet substructure tools require reclustering of the constituents contained in a jet. The reclustering is then often done with other algorithms, such as C/A, which is easier to decluster into subjets.

3.4 Kinematics of Parton Branching

Jets can be used to explore different processes by studying different jet observables. Jet observables are experimentally accessible properties of the kinematic of the jet. One would like to work with observables that are IRC safe, as little sensitive as possible to

model-dependent non-perturbative effects such as hadronization, and as little sensitive as possible to pileup and soft background radiation.

When a jet is measured we typically have access to the transverse momentum p_T, the mass m, and the detector coordinates in the (ϕ, η) space. Each constituent of the jet has a four-momentum $p = (E, \vec{p})$. From these kinematic variables, one can make a lot of different observables.

Let us define some observables for an arbitrary $1 \to 2$ splitting. In Eq. 3.1 we have defined the **momentum-sharing fraction**, which can be rewritten in terms of the transverse momentum as

$$z = \frac{\min(p_{T,1}, p_{T,2)}}{p_{T,1} + p_{T,2}}, \tag{3.18}$$

where the indices $1, 2$ represent the two subjets of the splitting. We can also rewrite the **opening angle**, or splitting angle, θ from Eq. 3.2 as

$$\theta = \arccos \frac{\vec{p_1} \cdot \vec{p_2}}{|\vec{p_1}| + |\vec{p_2}|} = \triangle R_{12}. \tag{3.19}$$

where $\triangle R$ is the distance between the particles, or subjects, in the (ϕ, η) space. Finally, one can also rewrite the formation time from Eq. 3.9 as

$$t_f = \frac{2p_T}{M} = \frac{2}{k_T \theta}. \tag{3.20}$$

In addition to the observables above, we also define the **invariant jet mass** M, using the energy sharing fraction and splitting angle, by the relation

$$M^2 = z(1-z)p_T^2\theta^2, \tag{3.21}$$

and the **relative transverse momentum** k_T

$$k_T = z(1-z)p_T\theta. \tag{3.22}$$

These are some relevant examples of observables that encodes a lot of information about each splitting in the jet and the jet substructure.

3.4.1 Lund Plane

Lund diagrams [39] are theoretical representations of the internal structure of a jet. Lund diagrams represent the radiation patterns inside the jet and are often plotted as a triangle in a $\ln R/\theta$ and $\ln k_T/p_T \approx \ln z\theta/R$ plane, also called **the Lund plane**, as shown in the left plot in Figure 3.7. Depending on the characteristics of the jet, certain areas of the Lund diagram are populated. The area right below $z = 1$ is typically populated by jets containing hard, collinear radiation with large z values. Jets with soft, large-angled radiation typically resides at low values of $1/\theta$.

The Lund diagram is obtained by reclustering a jet's constituents with the Cambridge-Aachen algorithm, and then decluster the jet and going through each step in the clustering sequence from the full jet down to the constituents creating a tree-like structure. **The Lund tree** is created by the following process:

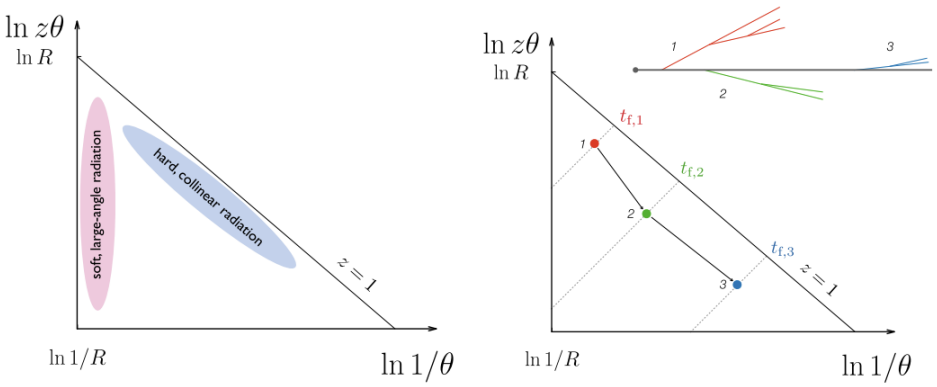


Figure 3.7: Left: The Lund plane for jets with opening angle R, highlighting the areas typically populated by soft, large-angled radiation and hard, collinear radiation. Right: Example of how to construct the primary Lund Plane. Figures from [38].

1. Declustering the current jet into two subjets j_1, j_2.
2. Calculate a number of kinematic variables associated with the declustering, which we denote as the tuple $\mathcal{T}^{(i)}$

$$\mathcal{T}^{(i)} = \{k_T, \theta, z, m, \psi\} \tag{3.23}$$

   where $\psi = \tan^{-1}(\frac{y_b - y_a}{\phi_b - \phi_a})$ is the azimuthal angle around the subjet j_1's axis.
3. Repeat the procedure for subjets j_1 and j_2 if they contain more than one particle.

This procedure produces a binary Lund tree with a tuple of observables $\mathcal{T}^{(i)}$ for each splitting i of the Lund tree. The first two elements of the tuple provide the coordinates in the Lund plane of the splitting, and the remaining variables provide complementary kinematic information.

A Lund diagram is then built by mapping every splitting along one branch of the jet to a point on the Lund plane. One branch is defined as following the hardest splitting in each vertex all the way through the tree. The primary emissions off the main branch build up the **primary Lund plane** and are illustrated in the right plot in Figure 3.7. Emission from each of these primary emissions generates new, orthogonal Lund planes, and so on. The primary Lund plane can be used as a two-dimensional visual representation of the radiation patterns in a jet [40].

Corrections to the Lund plane originating from non-perturbative hadronization effects affect the low k_T region of the plane [6]. One can therefore limit the dependence on non-perturbative effects by removing emissions that fall below a certain transverse momentum k_T threshold.

3.5 Jet Substructure Tools

Jet substructure studies aim to investigate the inherent kinematic characteristics of jets. However, the presence of background radiation and pileup can obscure the internal structure of jets. To disentangle the signal from noise, various jet structure tools are employed. Two widely used tools are background subtraction and jet grooming. Although

these tools share similarities, background subtraction primarily focuses on eliminating soft radiation within the jet, while grooming techniques are designed to target specific regions on the Lund plane.

3.5.1 Background Subtraction

Constituent subtraction [41] is the local subtraction of soft background radiation and pileup at the level of individual constituents. Constituent-based subtraction is performed particle-by-particle, simultaneously correcting the 4-momentum of the jet and its substructure. This is done by combining the kinematics of particles within a specific jet with the kinematics of soft "negative" particles that are added to balance the pileup contribution.

The basic ingredient of constituent subtraction is pileup energy density estimation. The contamination due to pileup is described in terms of the transverse momentum density ρ and mass density ρ_m. All particles in the event are grouped into *patches* in order to estimate the densities. The patches are defined by jets reconstructed using the k_T algorithm. The $p_{T,patch}$ and $m_{\delta,patch}$ of each patch are determined by summing over all particles within the patch, so that

$$p_{T,patch} = \sum_{i \in patch} p_{T,i}, \quad m_{\delta,patch} = \sum_{i \in patch} (\sqrt{m_i^2 + p_{T,i}^2} - p_{T,i}), \tag{3.24}$$

where $p_{T,i}$ and m_i are the transverse momentum and mass of particle i. Each patch covers a certain area A_{patch} in the (η, ϕ) plane. The overall p_T and mass densities are

$$\rho = median_{patches}\{\frac{p_{T,patch}}{A_{patch}}\}, \quad \rho_m = median_{patches}\{\frac{m_{\delta,patch}}{A_{patch}}\}. \tag{3.25}$$

The estimation of the background densities is followed by a scheme by which to do the subtraction. In this approach, massless particles with very low momentum are incorporated into the event so that they uniformly cover the (η, ϕ) plane with high density. These soft particles are referred to as ghosts and are most commonly used to define the area of a jet [42]. Each ghost covers a fixed area, A_g, in the (η, ϕ) plane. The 4-momentum of each particle or ghost is expressed by

$$p^\mu = [E, p_x, p_y, p_z] = [(p_T + m_\delta)\cosh y, p_T \cos\phi, p_T \sin\phi, (p_T + m_\delta)\sinh y] \tag{3.26}$$

where $m_\delta = \sqrt{m^2 + p_T^2} - p_T$. After adding ghosts into the event, the jet clustering algorithm runs over all particles and ghosts, giving back the same jets as in the case without the ghosts. Now the jets contain the real particles as well as ghosts, where the ghosts can be used to correct for the pileup in each jet.

We pick the jet we are interested in and separate the ghost particles from the real particles. We then create the 4-momentum for each ghost by identifying the transverse momentum p_T^g and mass m_δ^g for each ghost with area A_g as

$$p_T^g = A_g \cdot \rho, \quad m_\delta^g = A_g \cdot \rho_m. \tag{3.27}$$

For each pair of particle i and ghost k, the distance measure $\triangle R_{i,k}$ is defined as

$$\triangle R_{i,k} = p_{T,i}^{\alpha} \cdot \sqrt{(y_i - y_k)^2 + (\phi_i + \phi_k)^2} \tag{3.28}$$

Generally, α can be any real number but is taken to be zero here. The list of all distance measures is sorted from the lowest to the highest value. Then we start from the particle-ghost pair with lowest $\triangle R_{i,k}$. For each pair, the transverse momentum p_T and mass m_δ for each particle i and ghost k are modified as follows

$$p_{T,i} \leq p_{T,k} : \begin{cases} p_{T,i} \to p_{T,i} - p_{T,k} \\ p_{T,k} \to 0 \end{cases} \quad otherwise : \begin{cases} p_{T,i} \to 0 \\ p_{T,k} \to p_{T,i} - p_{T,k} \end{cases} \tag{3.29}$$

$$m_{\delta,i} \leq m_{\delta,k} : \begin{cases} m_{\delta,i} \to m_{\delta,i} - m_{\delta,k} \\ m_{\delta,k} \to 0 \end{cases} \quad otherwise : \begin{cases} m_{\delta,i} \to 0 \\ m_{\delta,k} \to m_{\delta,i} - m_{\delta,k} \end{cases} \tag{3.30}$$

The azimuthal angle ϕ and rapidity η of the particles and ghosts remain unchanged. The iterative procedure ends when the end of the sorted list is reached. One can also introduce a threshold $\triangle R^{max}$ to stop the iterations, this can guarantee that only a ghost neighboring to a given particle is used to correct it. Then, particles with zero transverse momentum are discarded, and we can recombine the jet of the remaining particles.

The above-described subtraction procedure corrects the 4-momentum of a jet without ruining its internal substructure.

3.5.2 Jet Grooming

Jet grooming techniques have been developed to mitigate the influence of soft background radiation on jets. These methods typically involve the removal of soft radiation originating from the background rather than the QCD radiation within the jet. By targeting regions in the Lund plane that are predominantly populated by background particles, groomers effectively reduce the impact of soft radiation. Additionally, groomers can be applied in a tagger mode, allowing them to identify the most important $1 \to 2$ splittings in the Lund tree. In the following section, we will outline several well-known grooming techniques.

The **Mass-drop tagger** was originally proposed as a tool to isolate boosted Higgs bosons from the QCD background [43]. In this method, one first reclusters the jet constituents of the jet with the C/A algorithm, and follow the recipe as follows:

1. Break the jet j into subjets j_1 and j_2 by undoing the last step of the clustering
2. If the splitting follows these two conditions, then we keep j as the final jet:

$$\max(m_{j_1}, m_{j_2}) < \mu_{cut} m_j \tag{3.31}$$

$$\min(p_{T,j_1}^2, p_{T,j_2}^2) \triangle R_{j_1 j_2}^2 > y_{cut} m_j^2 \tag{3.32}$$

3. Otherwise, we redefine j to be the most massive of the subjets, and iterate the procedure.

4. If j can no longer be declustered, keep j as the final jet

The procedure has two parameters, the mass-drop parameter μ_{cut} and the symmetry cut y_{cut}. The mass-drop tagger is a technique used to identify two-pronged boosted objects by utilizing the principles of symmetric cut and mass-drop condition. The symmetric cut requires the identification of two distinct and energetic prongs within the jet, while the mass-drop condition ensures the transition from a jet originating from a massive boson to two jets arising from massless QCD radiation. Originally designed as a tagger, the mass-drop tagger also serves as a grooming tool as it progressively removes soft radiation located in the outer regions of the jet.

The **modified mass-drop tagger** (mMDT) was proposed as an adaptation of the Mass-drop tagger to prioritize the hardest branch, rather than the most massive branch, during the iterative declustering process [44]. This modification simplifies and enhances the analytical calculations involved in the tagger. Additionally, the study introduced two minor modifications to further refine the technique. First, the symmetry condition is replaced by

$$\min(p_{T,j_1}, p_{T,j_2}) > z_{cut}(p_{T,j_1} + p_{T,j_2}) \tag{3.33}$$

which slightly reduces sensitivity to non-perturbative effects. Second, the mass-drop condition would only enter as a secondary correction in the strong coupling constant, compared to the symmetry condition, and can therefore usually be ignored.

SoftDrop[45] can be seen as a generalization of mMDT. It replaces the symmetry condition for the declustering with

$$\frac{\min(p_{T,j_1}, p_{T,j_2})}{p_{T,j_1} + p_{T,j_2}} > z_{cut}(\frac{\triangle R_{j_1 j_2}}{R})^{\beta} \tag{3.34}$$

where R is the jet radius, z_{cut} is the symmetry cut and the β parameter control how aggressive the groomer is. z_{cut} is the same as in mMDT, keeping the hard structure and excluding the soft emission, starting from large angles. The β parameter controls the strength of the groomer, wherein the limit $\beta \to 0$, SoftDrop reduces to mMDT. Increasing β leads to less aggressive grooming, with $\beta \to \infty$ returning an ungroomed jet.

For $\beta > 0$, SoftDrop declustering removes soft radiation while maintaining a fraction of the IRC radiation. The consequence of this is that the SoftDrop procedure gives IRC safe results even on a jet with one constituent. In this regime, SoftDrop acts as a groomer. For $\beta < 0$, SoftDrop declustering can remove both soft and collinear radiation. Thus, in this regime, SoftDrop acts as a tagger since it vetoes jets that do not have two hard prongs. We typically consider $z_{cut} \simeq 0.1$ and $\beta \simeq 1$.

The Key Concepts of This Chapter

Jets are collimated sprays of hadrons that are created from the branching and evolution of a parton produced in particle collisions. **Parton branching** refers to the process of a single parton splitting into two, while **parton evolution** equations describe the evolution of the partons from one evolution scale to another.

When a jet travels through a medium, its constituent particles interact with the medium, resulting in energy loss through a phenomenon known as **jet quenching**.

Various models exist to describe jet quenching, depending on the properties of the medium. In the case of QGP, a strongly coupled liquid, **a hybrid model** is employed, treating physical processes at different energy scales separately. First, all constituent particles interact with the medium and **experience energy loss**. This energy is transferred to the medium and **generates an energy wake** that propagates through the medium. Finally, during hadronization, this energy wake converts into thermal particles. Since the wake has been distributed through the medium, the thermal particles from the wake are deposited over a larger region, causing the jet to lose energy.

Jets are not well-defined objects. To create jets, one has to rely on a **clustering algorithm**. The **generalized k_T algorithms** are commonly employed clustering algorithms, merging pairs of particles based on a chosen distance measure. The choice of **distance measure** can slightly influence the kinematic properties of the resulting jet, which, in turn, can impact the construction of graphs derived from the jets.

Part II

Methods

Chapter 4

Introduction to Machine Learning

Machine learning (ML) is a group of algorithms where the general goal is to recognize patterns in data. ML algorithms identify and classify structures within a dataset. In recent times, the improvement in the field of ML has led to a variety of applications for such algorithms. In physics, the goal is also to learn about the unknown patterns in nature, and analyze data to create models that are able to predict the behavior of complex systems [46]. ML tools are able to make predictions based on patterns in data, however, the resulting models are often referred to as black box models. A black box model refers to the lack of insight into the underlying workings of the parameters of the model. This means that our understanding of the physics behind the patterns found in the data, is limited by our understanding of the data patterns themselves.

The type of ML tools we are interested in is the different approaches to optimize jet substructure analyses. A jet is defined by a clustering algorithm, which is an example of an unsupervised machine learning technique [22]. Unlike the output of most clustering procedures, jets have a physical meaning.

Another ML tool used in jet physics is supervised learning, which includes all forms of jet tagging, or classification. In high-energy physics, it is possible to generate large datasets of jets with a known origin, or label. This makes ML especially well suited to tackle jet physics problems, such as tagging.

In this chapter, we will first give a short introduction to machine learning in general. Then we will discuss jet representations, such as jet images and graphs, that can be used in ML. After that, we will take a look at the most relevant machine learning algorithms for this thesis, namely neural networks (NNs), convolutional neural networks (CNNs), and graph neural networks (GNNs).

4.1 Basic Concepts of Machine Learning

Machine learning is a subfield of artificial intelligence that enables machines to learn patterns from data without being explicitly programmed. It involves developing algorithms that can improve their performance at a specific task with experience. There are various types of machine learning algorithms, including supervised learning, unsupervised learning, and reinforcement learning.

Unsupervised learning is used to find structure in unlabeled data. The goal of unsupervised learning is often to cluster or group data points based on similarities in

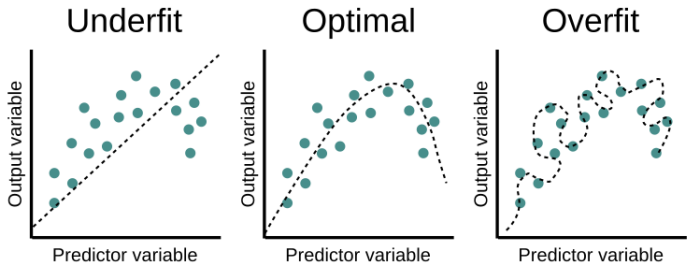


Figure 4.1: Overfitting vs underfitting. Figure from [47].

the input data. This approach can help identify underlying patterns or relationships in data sets, even when the categories or labels for the data are unknown.

Supervised learning, on the other hand, relies on labeled datasets. In this approach, the machine learning model is given input data along with corresponding target outputs or labels. The model then makes predictions based on the input data and compares these predictions to the true outputs or labels. The parameters of the model are then updated based on the differences between the predicted and true outputs, with the aim of minimizing the error.

Reinforcement learning is a bit different, as it involves the algorithm learning through trial and error. The algorithm is provided with rules or constraints, and then tries to learn optimal strategies by maximizing rewards and minimizing punishments. A classic example of reinforcement learning is training an algorithm to play a video game, where the reward might be a higher score.

The most commonly used machine learning approach is supervised learning, which can be used for both classification and regression tasks. In **classification** tasks, the algorithm is trained to identify which category or class a data point belongs to. For example, it might be used to distinguish between images of cats and dogs. In **regression** tasks, the algorithm is trained to estimate a continuous relationship between the input and output variables. For example, it might be used to predict the energy loss of a jet as it moves through a medium.

Regardless of whether we are dealing with classification or regression, supervised learning is about learning a function that maps inputs to outputs. The basic operation of a machine learning algorithm involves a forward pass and a backward pass. During the forward pass, the input data is processed layer by layer through the model until a prediction is made. During the backward pass, the algorithm calculates the error in the prediction and adjusts the weights of the model to improve the performance.

When it comes to choosing the right algorithm for a particular task, it's important to keep in mind the **"no free lunch"** (NFL) theorem [47]. This theorem states that there is no one algorithm that works best for every problem, and that the performance of an algorithm is highly dependent on the specific problem being solved. Therefore, it's important to evaluate and compare different algorithms based on their performance on specific tasks.

Another important concept in machine learning is the **bias-variance tradeoff**, which explores the relationship between the two sources of error for any model. Bias refers to how closely the model's predictions match the true values, while variance refers to how much the predictions vary based on changes in the input data. This trade-

off involves balancing the bias of the model with the variance of the model. A high-bias model is one that is too simple and fails to capture the complexity of the data, while a high-variance model is one that is too complex and overfits the data. A visualization of the underfitting of a high-bias model and overfitting of a high-variance model is shown in Figure 4.1. The optimal model has enough bias to avoid simply memorizing the training data, and enough variance to actually fit the patterns of the data[47]. Achieving the right balance between bias and variance is important for building a model that performs well on new data.

It's important to be aware of the issue of **overfitting** when training machine learning algorithms. Overfitting occurs when the algorithm becomes too good at recognizing patterns in the training data, to the point where it starts to perform poorly on new, unseen data. This can happen if the algorithm is too complex, or if there is not enough data available to train it effectively. To avoid overfitting, it's important to test the algorithm on a separate dataset from the one used for training.

One way to test for overfitting is to split the dataset into **training and testing sets**. The training data is used to train the algorithm, while the testing data is used to evaluate how well it generalizes to new data. A common choice is to split the dataset in 80% training data and 20% test data. Another way to avoid overfitting is to use regularization techniques to discourage overly complex models.

Another important consideration when training machine learning algorithms is choosing the right hyperparameters. Hyperparameters are settings that determine how the algorithm is trained, such as the learning rate or the number of layers in a neural network. Choosing the right hyperparameters can have a big impact on the performance of the algorithm, and often requires trial and error to tune.

In summary, machine learning involves creating algorithms that can learn patterns from data and make predictions or decisions based on that data. When selecting an algorithm, it is important to choose one that is well-suited for the specific problem being solved. Overfitting is a common challenge in machine learning that can be addressed through regularization techniques.

4.2 Jet Representation in Machine Learning

Machine learning algorithms require a proper representation of the data, so the first step is to decide how to represent particle jet data. Jet physics has complex structures, and there is no one unique way to encode information about the radiation pattern. In Figure 4.2, we provide an overview of popular jet representations and the corresponding algorithms.

One of the most commonly used representations of jets for machine learning purposes is **jet images** [2], as shown in the middle section of Figure 4.2. Jet images are snapshots of the energy distribution of the final state particles of a jet, in the azimuthal angle and rapidity plane, (ϕ, η) space. Jet images are great at visualizing the energy distribution in a jet, and by using pre-processing methods based on jet substructure, we can extract important information about the jet. When working with images, **convolutional neural networks** (CNNs) are a natural choice. CNNs are widely used for image classification tasks, and their use in jet physics has also gained popularity in recent years.

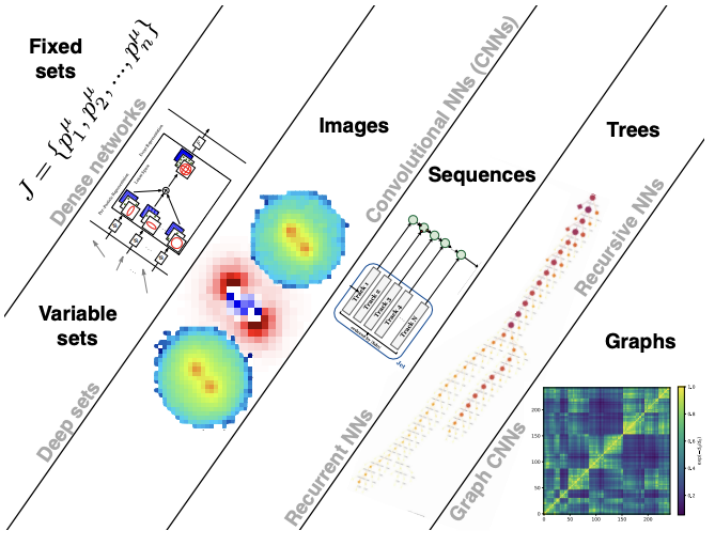


Figure 4.2: A schematic overview of the ways to represent jets and the NN that goes with the representations. Figure from [22].

Despite the promising performance, the jet image representation has two main shortcomings. While it can include all information about the energy distribution within the jet, how to incorporate additional information of the particles is unclear. Moreover, treating jets as images also leads to a very sparse representation. A jet typically needs atleast a 32×32 image in order to fully contain the jet, however, more than 90% of the pixels are blank. This makes the CNNs computationally inefficient on jet images. However, jet images are not the only representation of jets, as shown in Figure 4.2. We can also use unordered sets of four-vectors, variable sets of observables, sequences of constituent particles, binary trees originating from the clustering history, or graphs to represent jets.

Let us now consider the **graph representation of jets**, which is an extension of the sequence and binary tree representation. One can represent jets as graphs in various ways. One can represent each particle in the jet as a node, creating a particle cloud where the particles can be combined in different ways. Another graph representation of jets is using the Lund declusterings, or the clustering history of the jet. As we hav discussed previously, one can build a Lund tree by working through the reclustered jet. One can transforming the Lund tree into a graph, where each node corresponds to a Lund declustering and carries the tuple of kinematic variables $\mathcal{T}$ from Eq. 3.23. This graph representation models the branching history of the jet, with each node representing a branching, or declustering, at a point in the jet's evolution and being connected to other nodes through edges. The graph structures allow for the inclusion of any kind of features for each particle and therefore is significantly more flexible representation of jets, and can be utilized by **graph neural networks** (GNNs).

Although GNNs are becoming more popular, they are not yet widely used in jet physics. However, access to information about the structure and evolution of a jet should allow the network to learn more about the data and hopefully make better predictions.

To summarize, jets can be represented in a number of different ways, but the most relevant for us is jets represented as images and graphs. The use of jet images is very

popular, and CNNs are widely used for all kinds of image classification tasks. The graph representation of jets is gaining popularity due to the vast amount of information available in the clustering history of the jet and GNNs are becoming state of the art within the machine learning community. In the next sections, we will dive deeper into CNNs and GNNs, however as both CNNs and GNNs are neural networks (NNs), then we should first introduce the more general NNs.

4.3 Neural Network

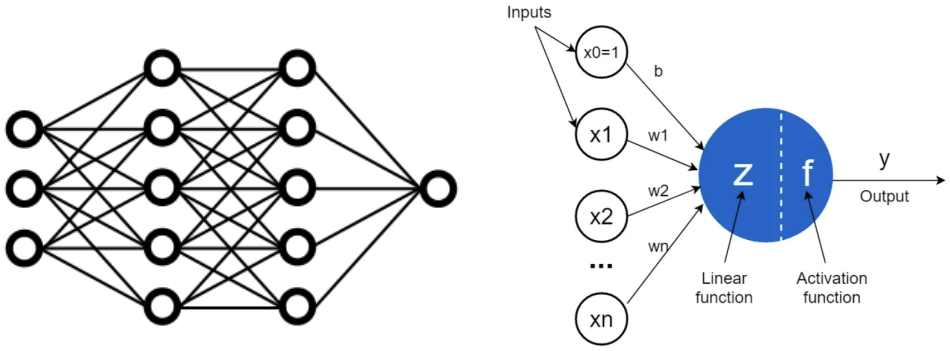


Figure 4.3: Left: An example of a fully connected neural network with the input layer, two hidden layers and one output node. Figure from [22]. Right: Image of one artificial neuron. Figure from [48].

Neural networks (NNs) are algorithms inspired by the way biological neurons in the brain work. They are composed of connected nodes that exchange and process information. There are various types of NNs, including fully connected neural networks (FCNNs), convolutional neural networks (CNNs), and graph neural networks (GNNs), which we'll explore in this section.

Typically, an NN consists of an input layer, one or more hidden layers, and an output layer. The input layer receives data, and the output layer produces the network's final output. In between, the hidden layers perform computations on the data. The left image in Figure 4.3 shows a simple neural network.

A fully connected neural network, also known as a multilayer perceptron (MLP), is a simple example of an NN. In an MLP, each node is a **perceptron**, or artificial neuron. The typical structure of a perceptron is shown in right image in Figure 4.3. Each perceptron receives inputs, calculates the weighted sum of the inputs and a bias term, and applies an activation function to produce an output. The weights and biases of the perceptrons are the trainable parameters of the network, often called Θ, that get adjusted during training to optimize the network's performance. The calculation of the weighted sum of the inputs is called a linear operation and looks like this:

$$L(x,\Theta) = \sum_{i=0}^{n} \Theta_i x_i + b \tag{4.1}$$

where n is the number of inputs.

Activation functions are essential components of neural networks. They introduce non-linearity to the output, which is crucial for learning complex patterns in the data.

All activation functions need to be differentiable and quickly converging with respect to the weights, for optimization purposes. The most common activation functions are sigmoid, ReLU [49], and softmax.

To improve the performance of an MLP, various tools can be used. **Batch normalization** [50] is a technique that normalizes the inputs to a layer for each mini-batch, which speeds up training and reduces the number of epochs needed for convergence. **Dropout** [51] is a regularization method that randomly drops nodes in the hidden layers to reduce overfitting and make the network more robust.

To summarize, simple neural network, such as MLPs, typically consists of an input layer, one or more hidden layers, and an output layer. The neurons in the network are interconnected and each neuron performs a computation on its inputs and passes the result to the next layer of neurons. Neural networks use activation functions to introduce non-linearity to the model which is crucial to learn complex patterns in data. Various methods can be used to improve the performance of MLPs, but MLPs are not suitable for all types of data, such as images or graphs. Convolutional neural networks (CNNs) are designed to handle image data, while graph neural networks (GNNs) are tailored to work with graph data. We'll discuss these specialized types of neural networks in the next sections.

4.4 Convolutional Neural Network

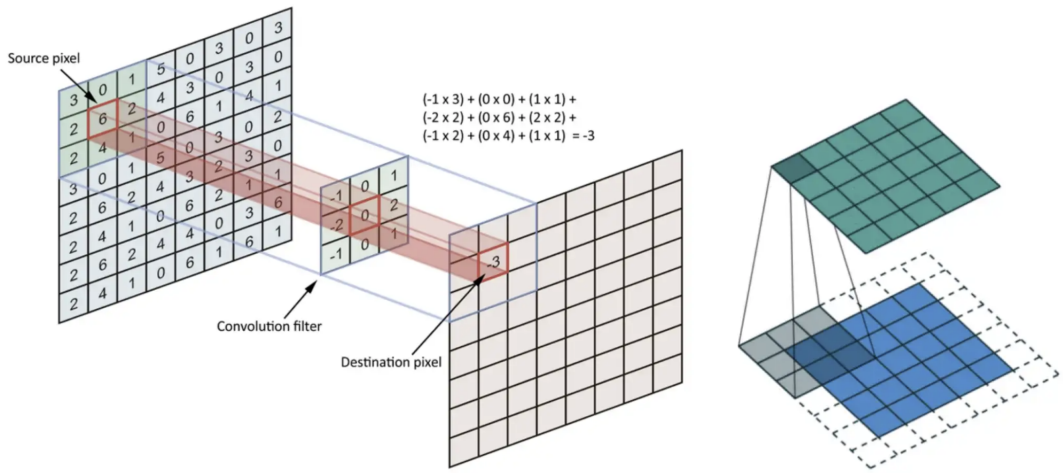


Figure 4.4: An example of a convolution operator, and padding. Figures from [52].

A convolutional neural network (CNN) is a specialized neural network commonly used for image processing. Unlike regular MLPs, CNNs are designed to preserve the spatial information of an image by extracting local features through one or more convolutional layers.

CNNs contain one or more **convolutional layers**, which apply a set of filters to the input image to extract features. Each filter is applied to a specific area of the image, and a dot product is computed between the input pixels and the filter. By moving the filter across the image, a series of dot products are generated, resulting in a feature map that represents the presence of a certain local feature in the image. This is illustrated in the left image in Figure 4.4. After the feature map is created one can apply an activation

function to introduce non-linearity to the model and determine whether a certain feature is present at a given location in the image. The most common activation function for CNNs is ReLU. Depending on what filter you apply, different features may be detected. The filters are updated as the network is trained.

One challenge of using convolutional layers is that the filters do not run along the edges. As we can see in the left image in Figure 4.4, the filter is placed at the first possible position, but the output of that filter describes the middle pixel of the filter. To consider the edge pixels of the image one can apply **padding**, which adds zeros around the image before applying the filters as shown in the right image in Figure 4.4. This ensures we do not lose information about the edge of the image and that the resulting feature map has the same dimensions as the original image.

After each convolutional layer, a **pooling** layer is often added to reduce the size of the feature map and extract the most important features. There are different types of pooling, such as max pooling and average pooling, which compute the maximum or average value of a local region of the feature map. Pooling helps to reduce the number of parameters in the network and prevent overfitting.

After several convolutional and pooling layers, the network typically ends with one or more **fully connected layers**, which flatten the output of the previous layer and apply a set of weights to produce the final output. The fully connected layers can be used to classify the input image into different categories or perform regression tasks.

To summarize, convolutional neural networks extract local features from images using convolutional layers. Convolutional layers apply a set of filters to the image and compute the dot product between the filter and the input pixels. One can use padding to extract information also from the edges of the image, while one can use pooling to reduce the dimension of the feature maps and reduce the number of trainable parameters in the network. CNNs usually end with fully connected layers leading to a single output node.

4.5 Graph Neural Network

A Graph Neural Network (GNN) is a type of neural network that is specifically designed to operate on graph-structured data [53]. A graph is a collection of objects, referred to as **nodes**, and the connections between them, called **edges**. Graphs can represent various types of data and their relationships, making them a flexible and powerful data structure. The information associated with each node or edge can be used as input to a GNN, and the entire graph can also contain global information. By utilizing the information of the neighboring nodes, information is accumulated through the layers in a GNN, as visualized in Figure 4.5.

An **adjacency matrix** is a way to represent the connectivity of a graph in matrix form. It is a square matrix where each entry corresponds to a connection between two nodes. In Figure 4.6 we find an example of the adjacency matrix for a reclustered jet.

The simplest GNNs apply separate MLPs on each component of the graph - nodes, edges, and the global graph - to learn representations of each component. The output graph has the same number of nodes and edges as the input graph, and its connectivity remains unchanged.

Pooling is a technique commonly used in CNNs to reduce the spatial dimensions of

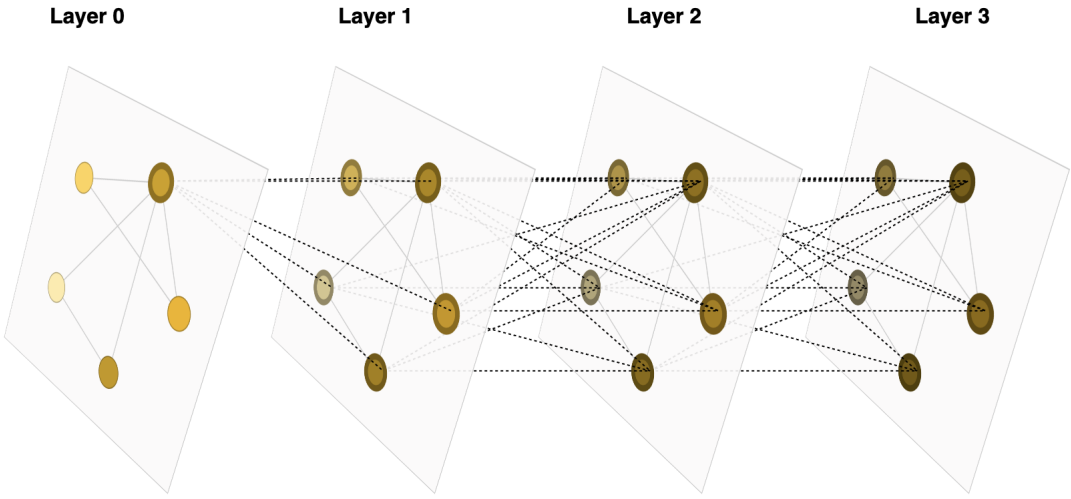


Figure 4.5: An example of how information from one node accumulates through layers in a GNN. Figure from [53].

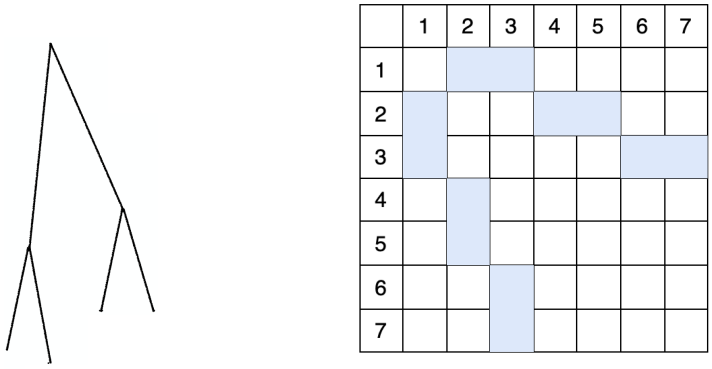


Figure 4.6: An examples of an adjacency matrix for a simple jet with two subjets.

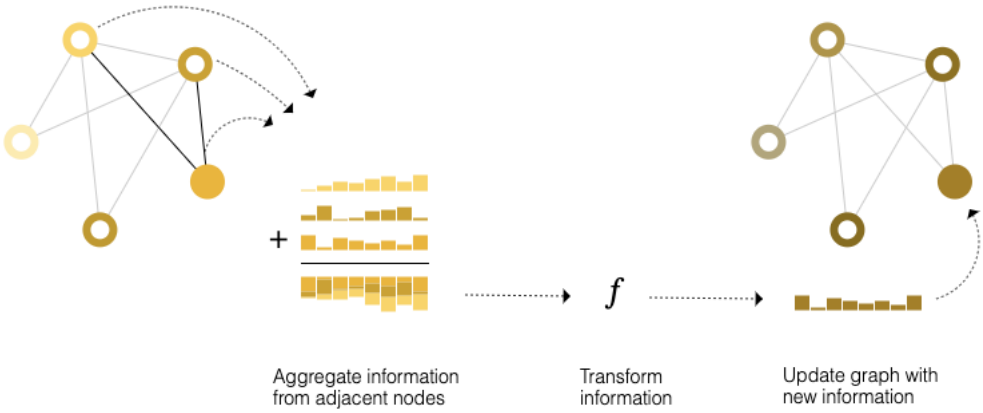


Figure 4.7: Example of message passing. Figure from [53].

the input, but it can also be used in GNNs to aggregate node features or edge features. One way to perform pooling in GNNs is by using message passing [54], where neighboring nodes or edges exchange information. Message passing works in three steps: gathering messages from neighbors, aggregating the messages, and updating the node or edge embeddings using an update function. The concept of message passing is illus-

trated in Figure 4.7. The message-passing steps are key to utilizing the connectivity of graphs, and by stacking multiple message-passing GNN layers, a node can incorporate information from across the entire graph.

Different GNN architectures may focus on node features or edge features, depending on the problem at hand, such as node classification, graph classification, and link prediction. In the context of jet physics, there are several popular GNN architectures that emphasize different types of features and employ different message-passing strategies.

To summarize, graph neural networks are specialized for operating on graphs. Information can be stored in both the nodes and edges of a graph, making them very flexible and powerful data structures. The basis of GNNs is the message-passing between neighboring nodes, which allows information from the whole graph to accumulate through various layers of the GNN.

4.5.1 ParticleNet and EdgeConv

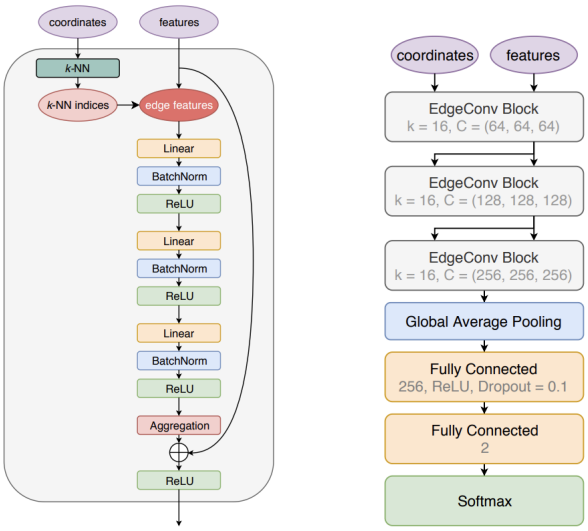


Figure 4.8: The architecture of the EdgeConv block and ParticleNet. Figure from [55].

One of the most popular GNNs for jet physics is ParticleNet [55]. With ParticleNet they propose an approach that considers a jet as a point cloud of particles. ParticleNet is a CNN-like graph neural network for jet tagging with particle cloud data. Motivated by the success of CNNs, ParticleNet adopts a similar approach to learning on particle cloud data.

However, regular convolution operation cannot be applied on point clouds, as the points can be distributed irregularly, rather than following some uniform grids as the pixels in an image. Therefore they have to redefine the definition of a "local patch" for point clouds.

Edge convolution [56] (EdgeConv) is proposed as a convolution-like operation on point clouds. EdgeConv starts by representing a point cloud as a graph, whose nodes are the points them self and the edges are constructed as connections between each point and its k nearest neighboring points. The EdgeConv operation for each point x_i

has the form

$$\vec{x}_i' = \square_{j=1}^{k} h_\Theta(\vec{x}_i, \vec{x}_{i_j}), \tag{4.2}$$

where $\vec{x}_i$ denotes the feature vector of the point x_i and $\{i_1, ..., i_k\}$ are the indices of the k nearest neighboring point of the point x_i. The **edge function** h_Θ is some function with learnable parameters Θ, and $\square$ is a symmetric **aggregation operation**, such as max, sum, or mean. The learnable parameters Θ are shared for all nodes. Generally, given a graph with n nodes and an F dimensional feature vector belonging to each node, EdgeConv produces a graph with the same number of nodes and an F' dimensional feature vector for each node. The choice of edge function h_Θ and aggregation operator influence the properties of the EdgeConv. In [56] they use the edge function

$$h_\Theta(\vec{x}_i, \vec{x}_{i_j}) = \bar{h}_\Theta(\vec{x}_i, \vec{x}_{i_j} - \vec{x}_i), \tag{4.3}$$

which encodes both the global shape captured by the features at the central node $\vec{x}_i$ and the local information captured by $\vec{x}_{i_j} - \vec{x}_i$ [56]. The edge function can be implemented as an MLP, whose parameters are shared among all edges. For the aggregation operation, they use an element-wise average of the edge features of all the neighboring edge features, as it shows the best performance [55]. A shortcut connection [57] is also added to allow the input features to pass through directly.

Putting this all together, one creates an EdgeConv block, as shown in the left image in Figure 4.8. The input to the EdgeConv blocks is the coordinates of the particle, to find the k-nearest neighbors and the feature vector for the node. An EdgeConv block is characterized by the number of neighbors k and the number of channels C corresponding to the number of units in each linear layer. One important feature of the EdgeConv is that it can easily be stacked, similar to regular convolutions, as it only changes the dimensions of the feature vector while the graph itself stays the same.

The architecture of ParticleNet is shown in the right image in Figure 4.8. It consists of three EdgeConv blocks, then a global average pooling operation is applied to aggregate the learned features over all particles in the cloud. This is followed by two fully connected layers, where the second fully connected layer had two units, and is followed by a softmax activation function, as to generate the output for the binary classification task.

To summarize, ParticleNet is a CNN-like graph neural network that does jet tagging on a particle cloud. To do this it uses a similar approach as CNNs, by implementing an edge convolution operation. The edge convolution operator considers the k nearest neighbor of each particle when creating the feature map. This allows information from essentially the whole graph to be accumulated into the later layers of the model. ParticleNet then ends with a couple of fully connected layers and finish with a softmax function to generate a binary classification output.

4.5.2 LundNet

Jets can also be represented as graphs by considering the Lund tree. The Lund tree contains a rich set of information on the substructure and radiation patterns of a jet, therefore serving as a natural way to represent jets in machine learning. As mentioned, one can transform the Lund tree into a graph with each declustering as a node with the feature vector $\mathcal{T}$ from Eq. 3.23.

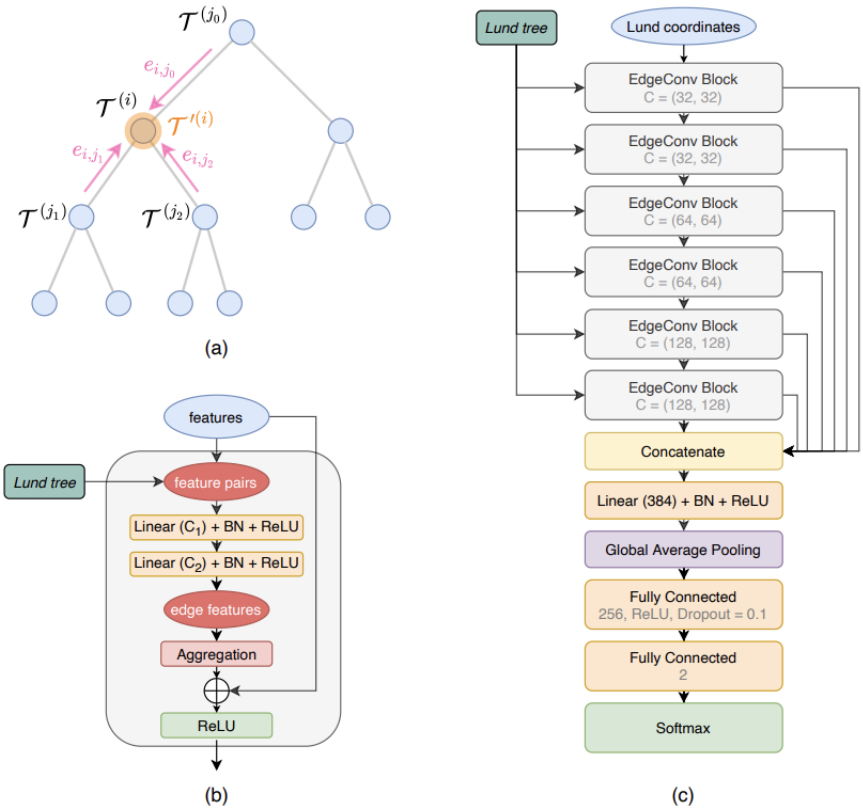


Figure 4.9: (a) Illustration of the EdgeConv operation on a node in the Lund Tree. (b) The architecture of the EdgeConv block. (c) The architecture of the LundNet model. Figure from [6].

To utilize this graph representation one can use LundNet [6], which is an adaptation of ParticleNet, with edge convolutions. In these graphs, each node corresponds to a Lund declustering and carries feature vector $\mathcal{T}$. Figure 4.9(a) is an illustration of how the EdgeConv operation updates a node in the graph. While the EdgeConv in ParticleNet updates the feature vectors based on the k nearest neighbors, the graphs in LundNet are predetermined and each node can only have up to three edges. This means that LundNet does not require the k-nearest neighbor search, which reduces the computational cost drastically.

Much like ParticleNet, LundNet consists of several EdgeConv blocks. As the operation is simplified to at most three known connections, the EdgeConv block is simplified, as one can see in Figure 4.9(b).

The architecture of the LundNet model is shown in Figure 4.9(c). They stack six EdgeConv blocks to form a deep graph network. The model can now use more EdgeConv blocks than as the computational cost is reduced without the k-nearest neighbor search. The outputs from these EdgeConv blocks are concatenated per node and further processed by another MLP to better aggregate features learned at different stages. Then, a global average pooling is applied to read out information from all nodes in the graph. This is followed by a fully connected layer with dropout, before the final output.

The LundNet model is implemented using Deep Graph Library [58] and PyTorch [59].

To summarize, LundNet is an adaption of ParticleNet which considers the Lund tree. The Lund tree is a binary tree, meaning it has only up to three connections for each

node. This means that the k nearest neighbor search from ParticleNet is not necessary and the computational cost is drastically reduced.

The Key Concepts of This Chapter

The utilization of **jet images** and **convolutional neural networks** (CNNs) has become increasingly popular in the field of jet physics. However, due to the inherent limitations of jet images in capturing additional information beyond radiation patterns, alternative jet representations for machine learning applications have gained attention.

One such representation is the graph-based approach, which allows for the extraction of information from the internal structure of jets and enables the inclusion of plenty of node-specific information. Jets can be represented as graphs using various methods, such as point clouds with k nearest neighbors or **the Lund tree representation**.

To exploit the graph-like nature of jets, we can use **graph neural networks**. In particular, we are interested in using the framework of **LundNet**, a powerful tool designed to analyze the Lund tree. The Lund tree contains a rich set of information regarding a jet's substructure and radiation patterns. LundNet employs **edge convolutions**, inspired by the convolutional operators in CNNs, to effectively gather information from the entire tree structure through the graph neural network.

Chapter 5

Task Description and Network Architecture

This chapter delves into the specifics of our task, which revolves around predicting the energy loss experienced by a jet as it traverses a medium.

During the jet's journey through the medium, interactions with the environment result in energy loss for its constituent particles. However, the presence of a steeply falling production spectrum introduces a bias towards jets that have undergone minimal energy loss. This bias poses a challenge to the interpretation of jet quenching, as it neglects the most heavily modified jets. To overcome this limitation, it is useful to be able to estimate the magnitude of energy loss suffered by jets due to interactions with the medium. In this regard, various machine learning techniques can be employed.

The primary objective of this chapter is to provide a comprehensive overview of our task, examine previous work on the subject, and describe the network architecture we will employ in this thesis. Furthermore, we introduce the dataset, introduce all relevant pre-processing steps, and outline the key aspects we aim to investigate in the analysis. Through these efforts, we establish the groundwork for a detailed exploration of our approach to predicting jet energy loss.

5.1 Task Description

When studying modifications of jets in experiments, it is common to select jets within a specific p_T range. However, this introduces a selection bias due to the steeply declining jet production spectrum. Jets that have lost a significant amount of energy are unlikely to be observed above the p_T cut, as they would have been produced at higher p_T where the spectrum is suppressed. Consequently, the actual impact of medium effects on final measurements becomes obscured, posing challenges in extracting information about the properties of the medium.

To mitigate the selection bias and gain a better understanding of the medium effects, it is essential to estimate the energy loss experienced by individual jets, a nontrivial task on a per-jet basis. Machine learning techniques provide a means to predict energy loss, where regression models learn the relationship between input features and the true output values. However, the choice of machine learning model depends on the type of input representation employed.

In this section, we delve into the most relevant architectures, inspired by previous work, as we explore alternative jet representations. These explorations aim to enhance our approach to predicting energy loss.

Before proceeding, it is worth mentioning our decision to predict the energy loss ratio, denoted as χ, and the rationale behind this choice. Firstly, χ serves as a suitable metric for assessing energy shifts at the level of observable particles, aiding in the mitigation of event generator bias. Moreover, neural networks demonstrate effective approximation capabilities for χ, outperforming alternative quantities [5]. Lastly, the calculation of χ from hybrid events is straightforward, further supporting its selection as the target variable for our predictions.

5.2 Previous work

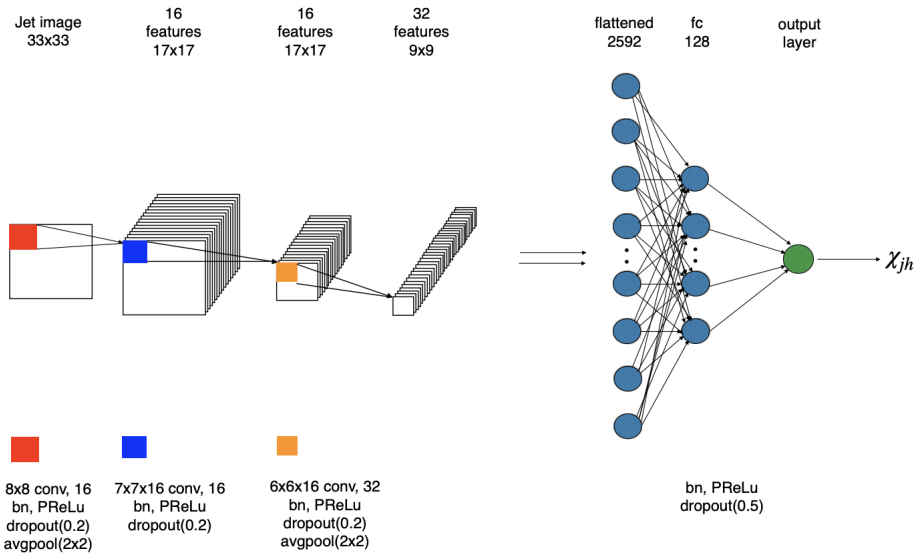


Figure 5.1: Architecture of the CNN for predicting the energy loss ratio χ_{jh} from jet images. Figure from [5].

A previous study employed jet images and a convolutional neural network (CNN) to predict energy loss in a hybrid model [5]. The goal of the study was to predict the energy loss and explore the modification of observables depending on the amount of energy loss.

The jet images were created as 33×33 pixel representations, with each pixel's color corresponding to the deposited energy amount. To preprocess the images, the jet was first shifted so that the hardest subjet was centered. Next, the image was rotated such that the second hardest subjet aligned with $-\pi/2$ radians. Lastly, a parity flip ensured that the right side of the jet consistently had a higher sum of pixel intensities. These preprocessed images served as the input to the network.

The CNN architecture, illustrated in Figure 5.1, comprised three convolutional layers and one fully connected layer. Batch normalization, PReLU activation [60], dropout, and average pooling were applied after each convolutional and fully connected layer. The weight and bias matrices of the filters and dense layers were initialized using the "He normal" initializer [60] and constrained with L2 regularization [61] in the

loss function. The output layer consisted of a single neuron for predicting the energy loss ratio χ.

To assess the difference between the true and predicted χ, the regression task utilized a Log-Cosh loss function. The CNN's trainable parameters were updated using the AdaMax optimizer [62], and the network was constructed using Tensorflow Keras [63].

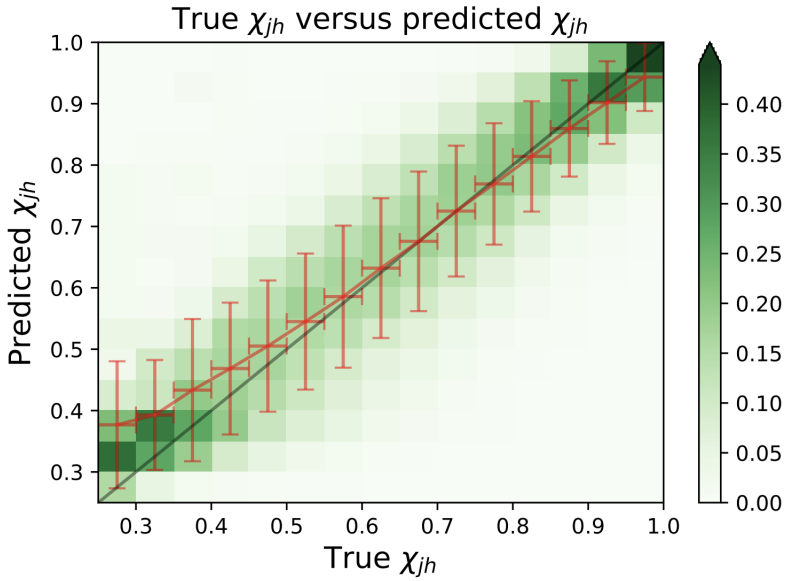


Figure 5.2: Prediction Performance. Figure from [5].

Figure 5.2 demonstrated the CNN's prediction performance of χ based on the preprocessed jet images. The green column represents the normalized joint distribution, indicating the probability of predicted χ given the true χ bin. The red line with error bars quantifies the average and standard deviation of the predicted χ for each true χ bin. The error bar decreases as χ increases. Notably, the red line deviates slightly from the diagonal line in low and high χ values.

This study explored various input configurations, and the overall conclusion was that the network performed well. The analysis revealed that the soft particles within the jet carry highly relevant information about the energy loss magnitude. The total number of activated pixels in the jet image, predominantly contributed by the production of thermal particles from the wake, exhibited a strong correlation with the energy loss ratio.

However, this approach has one shortcoming. There is a decrease in prediction performance once jets are embedded into a heavy-ion environment. To investigate the robustness of the results, the jets were embedded in a thermal heavy-ion background, and constituent subtraction was performed. The performance declined due to noise from the thermal particles with p_T similar to the soft particles originating from the wake. Enhancing performance under experimental conditions where jets are actually measured represents the next step in advancing this type of analysis.

5.3 Our Network Architecture: A Modified LundNet

In this thesis, we adopt a graph representation of the Lund tree to encode the radiation pattern and jet substructure, utilizing the LundNet framework to predict the energy loss ratio. The utilization of a graph structure enables us to input a feature vector of observables for each Lund declustering, incorporating the feature vector $\mathcal{T}$ from Eq. 3.23 to provide the network with comprehensive jet information.

However, LundNet is originally designed as a jet-tagging model, which implies that the graph neural network (GNN) functions as a classification algorithm. As our task involves regression to predict the energy loss ratio, several modifications are necessary to transition from a classification task to a regression task.

The first modification involves reducing the number of output neurons to one, aligning with the regression setup where a single output predicts a continuous variable representing the target value.

Next, we employ a sigmoid activation function on the network's output. The sigmoid function confines the output within the range of 0 to 1, which is suitable given that the energy loss ratio cannot exceed 1 in this case.

Furthermore, we adapt the loss function for regression tasks. Mean squared error (MSE) loss is a simple and effective choice for regression, in contrast to the Log-Cosh loss used in the previous work. We also implement L2-regularization and the "He normal" initializer, following the approach of the previous work.

The final step involves establishing the prediction performance measure. Drawing inspiration from Ref. [5], we adopt the visualization of prediction performance, which effectively demonstrates performance across different energy loss ratio values. Additionally, to provide a concrete performance metric, we compute the average MSE loss over the test data.

Aside from these modifications, the LundNet architecture remains the same as in Figure 4.9.

5.4 Dataset and Pre-Processing

The first dataset comprises approximately 100,000 hybrid model events generated from a parton gun. In this setup, a parton with a transverse momentum of $\hat{p_T} = 1000$ GeV is fired into a medium.

The second dataset consists of approximately 100,000 hybrid model events created by a full dijet spectrum at $\hat{p_T} = 50$ GeV and $\sqrt{s} = 5.02$ ATeV, with an oversampling factor of the hard cross-section of p_T^4 to ensure sufficient statistics at high p_T. These events correspond to PbPb collisions in the $0-5\%$ centrality bin, with an average temperature of $T \simeq 250$ MeV.

Both datasets include lists of particles belonging to the vacuum event and medium event at both partonic and hadronic levels. To construct the jets, we extract the relevant particles and perform clustering using the anti-k_T algorithm with a jet radius of $R = 0.4$. Subsequently, we apply the matching procedure outlined in Chapter 3.2.2 to identify matching vacuum and medium jets. The energy loss ratio χ is then calculated, and the medium-modified jets and corresponding energy loss ratio are stored for generating the graphs.

Since the dataset includes particles at both the hadronic and partonic levels, we can consider the partonic and hadronic jets separately. Partonic-level jets do not contain wake particles, as the thermal wake particles are not generated before hadronization. Therefore, parton jets can be regarded as more perturbative or "clean" objects, while hadron-level jets capture a broader range of physics, including non-perturbative effects. Consequently, we now have four datasets that we aim to train our network on.

To train the network using the jet data in our datasets, we need to transform them into graphs. Graphs are generated by reclustering the jets using the C/A algorithm and declustering each jet to construct the Lund tree. In this representation, each declustering corresponds to a node in the graph, and the splitting of particles results in two new particles, which can be envisioned as the edges connecting the nodes. Therefore, each node represents a splitting, and the connecting edges represent the particles generated by the first splitting and subsequently splitting again. Thus, the graph structure serves as input data for our network.

However, it is important to note that the realistic jet spectrum data exhibits a steeply falling distribution. Consequently, there is an imbalance in the dataset, with fewer samples having a large energy loss ratio. This imbalance poses challenges for machine learning algorithms. To address this, a pre-processing step is implemented to re-weight the samples, as done in Ref. [5].

The re-weighting procedure involves assigning each sample a weight in the loss function during training and validation. Since the data is unbalanced both in terms of χ and p_T, a two-dimensional re-weighting approach is proposed. The procedure begins by creating a two-dimensional histogram with the energy loss ratio χ along the y-axis and the transverse momentum p_T along the x-axis. Next, the effective sample number N_{eff} [64] is calculated for a specific (p_T, χ) bin using the formula:

$$N_{eff} = \frac{1-\beta^N}{1-\beta}, \tag{5.1}$$

where N is the total sample number in that bin and β represents the probability that a new sample in that bin is independent of the previous ones.The weight assigned to each sample is inversely proportional to the effective sample number in the corresponding bin. To balance the dataset, we choose $\beta = 0.9998$, which corresponds to a maximum of 5000 effective samples in each bin. The choice of β was inspired by Ref. [5]. Bins with a low number of samples receive higher weights, while bins with a large number of samples are assigned smaller weights, ensuring that the effective number of samples in each bin is capped at the maximum limit.

This re-weighting procedure is particularly necessary for jets from the realistic dijet spectrum, characterized by a steeply falling p_T spectrum. In contrast, jets from the parton gun dataset do not exhibit the same steeply falling spectrum, as they follow a Gaussian distribution of transverse momentum centered around the peak at $\hat{p_T} = 1000$ GeV. Nonetheless, the distribution of the energy loss ratio remains significantly imbalanced. Therefore, for parton gun jets, the same re-weighting procedure is applied, but only for the energy loss ratio, resulting in a one-dimensional re-weighting. This adjustment helps balance the parton gun datasets.

By employing these re-weighting techniques, we obtain more balanced datasets, which are then used for training the network.

5.5 Hopes and Dreams

In this thesis, we delve into our hopes and dreams for this thesis and bring up several intriguing problems concerning jet quenching and machine learning.

Our primary focus will be on tuning our network to improve the prediction performance, and hopefully match the previous work. The previous work done in Ref. [5] was very successful, and while we have no ambition of surpassing their results, we do wish to match their preliminary results for hadron-level jets originating from the realistic jet spectrum.

Another critical aspect of our thesis involves understanding the network's behavior and its perception of different jet features. We aim to determine whether the network favors more or less information and assess the impact of increasing or decreasing the number of input features. Additionally, we seek to explore the effects of grooming techniques, such as removing soft radiation, on the network's performance. Given the importance of hard radiation in the IRC-safe definition of jets, we believe that removing soft radiation should not hinder the prediction of energy loss.

Furthermore, we wish to study the robustness of the model, which posed a challenge in the previous work. We plan to introduce additional soft radiation, effectively injecting noise into the network, to examine its response. This study will shed light on whether the network prefers the removal of soft radiation using grooming tools or if such removal impedes accurate energy loss prediction. Understanding the causes of performance degradation and exploring the effects of information reduction techniques, including grooming, are key aspects of our research.

Finally, we recognize the dependence of the clustering algorithm on the construction of our graphs. Therefore, we wish to investigate the significance of the choice of clustering algorithm within our framework.

Through this thesis, we hope to be able to make accurate predictions on energy loss and explore what makes the performance decrease.

The Key Concepts of This Chapter

The aim of this thesis is to enhance the prediction of energy loss experienced by jets traversing through a medium. Previous studies utilized **jet images and a CNN** to achieve promising results, establishing a correlation between the activated pixels in the image and energy loss (Ref. [5]). However, the performance of the network declined when the jet was embedded in a thermal background, which is an essential consideration in realistic experimental scenarios.

To address this limitation, we propose a novel approach by employing a **graph representation of the jet and a GNN**. Specifically, we transform the Lund tree into a graph and adapt the **LundNet architecture** accordingly to conduct the analysis. Notable modifications include transforming LundNet from a classification network into a regression model and adjusting the loss function.

Through this investigation, we aim to explore the inherent graph structure of jets and develop a network that effectively handles the challenges posed by realistic scenarios involving an embedded background.

Part III

Analysis and Results

Chapter 6

Vacuum Jets Embedded in a Heavy-Ion Environment

In the process of reconstructing jets from the final-state hadrons in collisions, no background subtraction procedure can remove all the particles coming from the hadronization of the medium. Therefore, measurements of jets in experiments are often obscured by the presence of thermal particles emitted by the medium.

To gain a deeper comprehension of the implications of a jet being embedded in a heavy-ion environment, we will conduct a brief investigation focusing on vacuum jets embedded within a thermal background. This chapter aims to examine the influence of embedded background particles on various jet observables and utilize grooming techniques to minimize the impact of the embedded background.

Through this analysis, our objective is to distinguish the signal from the background and comprehend how the presence of an embedded background influences the characteristics of a jet in a vacuum. This analysis will serve as a foundation for the subsequent exploration of medium-modified jets embedded in a heavy-ion environment.

6.1 Generating the Vacuum Events

To initiate this study, we begin by generating vacuum events. We generate 100 000 events using PYTHIA8 [8] with an average energy $\hat{p_T}_{min} = 200GeV$. The clustering of jets is performed utilizing the anti-k_T algorithm with a jet radius $R = 0.7$, using FastJet [9]. A relatively large jet radius is used to ensure that the jets capture a large number of background particles.

The background is generated using the Blast Wave model outlined in Chapter 2.3.2. For this analysis, we create two different background configurations. We establish two distinct background configurations for this analysis. The first configuration consist of 375 thermal particles per rapidity, corresponding to the blast-wave parameter values $\beta_T = 0.52$ and $T_F = 120MeV$. The second configuration involves 1000 thermal particles per rapidity, which correspond to the parameter values $\beta_T = 0.63$ and $T_F = 100MeV$.

When embedding the jet within a heavy-ion environment, we introduce hundreds of particles to the jet. The desired signal becomes obscured by the noise caused by the thermal background particles. Therefore, our aim is to eliminate a significant portion of the background while preserving the signal. To achieve this, we apply constituent

subtraction as described in Chapter 3.5.1. The resulting jets following constituent subtraction are referred to as the ungroomed jets in this chapter.

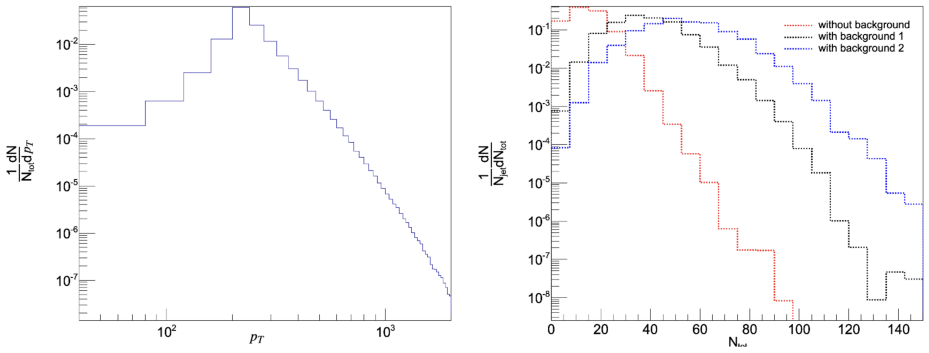


Figure 6.1: Overview plots of p_T and the number of particles in the jet. *Left*: p_T distribution of the hardest jets in each event, with $\hat{p_T}_{min} = 200$GeV. *Right*: N_{tot} distribution, the number of constituents in the jet.

After creating the datasets, we examine the distribution of transverse momentum p_T. Jets with a minimum transverse momentum of 200 GeV are generated, and the left plot in Figure 6.1 showcases a peak at 200 GeV with a decreasing spectrum. The right plot in Figure 6.1 illustrates the number of constituents in the jets, differentiating between jets with and without background particles. The dataset associated with 375 background particles per rapidity is denoted as background 1, while the dataset with 1000 background particles per rapidity is referred to as background 2. It is evident from the plot that the number of constituents in each jet is significantly greater in background 2 compared to the other datasets. The distribution peak has shifted from approximately 10 constituents in jets without background to roughly 50 constituents in the jets featuring background configuration 2.

6.2 Without Grooming

To establish a baseline for our study, we examine the distribution of observables without the application of any groomer, relying solely on constituent subtraction. The key observables of interest are the mass m, the splitting angle θ, momentum-sharing fraction z, and relative transverse momentum k_T. Additionally, we investigate IRC-unsafe observables associated with the number of constituents in the jet (N_{tot}) and the distribution of those constituents in the first pair of subjets derived from the initial declustering. We study the distribution of constituents in the subjets through the ratio $\frac{N_1 - N_2}{N_{tot}}$, where N_1 and N_2 are the numbers of constituents in each of the two primary subjet.

The analysis is performed on jets reclustering with the C/A algorithms, which are commonly employed in grooming techniques and will be utilized in constructing the Lund tree in the subsequent chapter.

We begin by examining the observables of the ungroomed jets. The upper left plot in Figure 6.2 depicts the distribution of the mass m)scaled by the transverse momentum p_T of the jet. It is observed that jets without additional background particles exhibit slightly lower mass compared to jets with background particles.

Next, the distribution of the relative transverse momentum k_T scaled by the transverse momentum p_T is presented in the upper right plot of Figure 6.2. The distributions

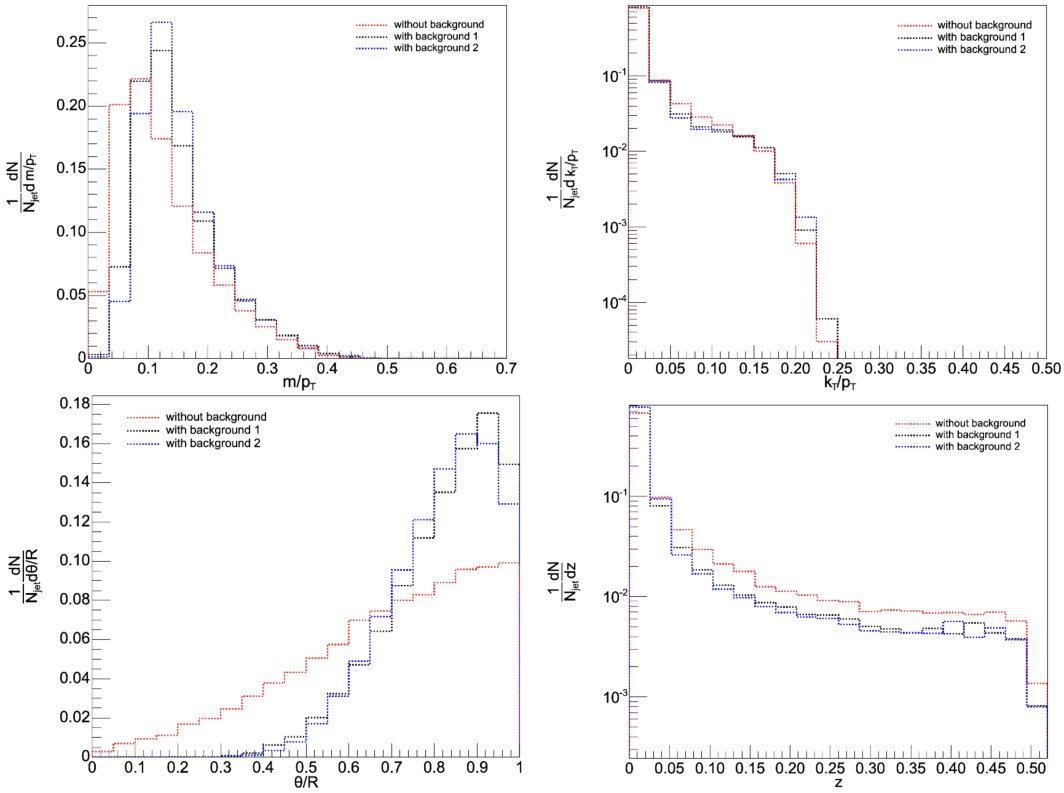


Figure 6.2: Plots of observable distribution for $\hat{p}_{t\,min} = 200\text{GeV}$ when only applying constituent subtraction. *Upper Left*: Mass distribution. *Upper Right*: k_T distribution. *Lower Left*: θ distribution. *Lower Right*: z distribution.

with and without background particles are very similar, with only minor differences in the tail region.

Moving on to the lower left plot in Figure 6.2, we observe the distribution of the splitting angle θ scaled by the jet radius. Jets without any background display a broader distribution than those containing background particles. Additionally, a slight shift in the distribution peak is evident, whereby an increased number of particles leads to a narrower peak at a lower θ value. Figure 5 in Ref. [38] reveals that upon introducing thermal background particles, the radiation in the low $\ln R/\theta$ region of the Lund plane intensifies, resulting in a shift of the θ spectrum towards larger θ/R values.

The lower right plot in Figure 6.2 portrays the distribution of the energy sharing fraction (z). Jets without background exhibit a higher distribution at high z values. Given the logarithmic scaling of the y-axis, the differences between the distributions are smaller than they look.

Furthermore, we examine observables associated with the distribution of constituents within the jets. Figure 6.3 presents the distribution of the ratio of constituents between the two primary subjets. It is evident that the second subjet predominantly contains the greater number of constituents. Jets without background particles exhibit a more pronounced peak, indicating a strong preference towards the majority of particles populating the second subjet, while jets with background particles display flatter distributions. Considering that jets clustered with C/A algorithms should ideally exhibit an even spread of constituents throughout the jet due to the angle-based clustering, the figure agrees with this expectation.

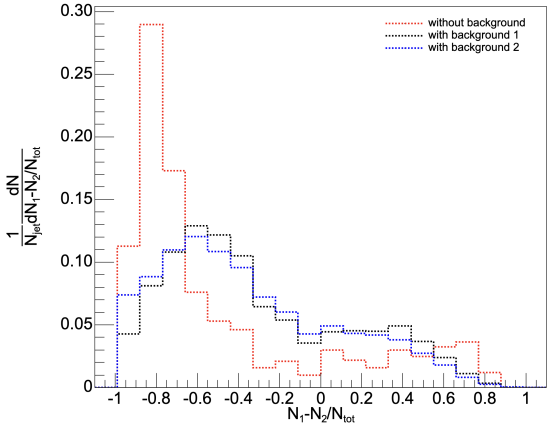


Figure 6.3: Plot of the distribution of the ratio $\frac{N_1-N_2}{N_{tot}}$ for vacuum jets with $\hat{p_T} = 200GeV$ and different degrees of background radiation. The ratio signifies the distribution of constituents within the two primary subjects of each jet.

6.3 Dependence on Reclustering Algorithm

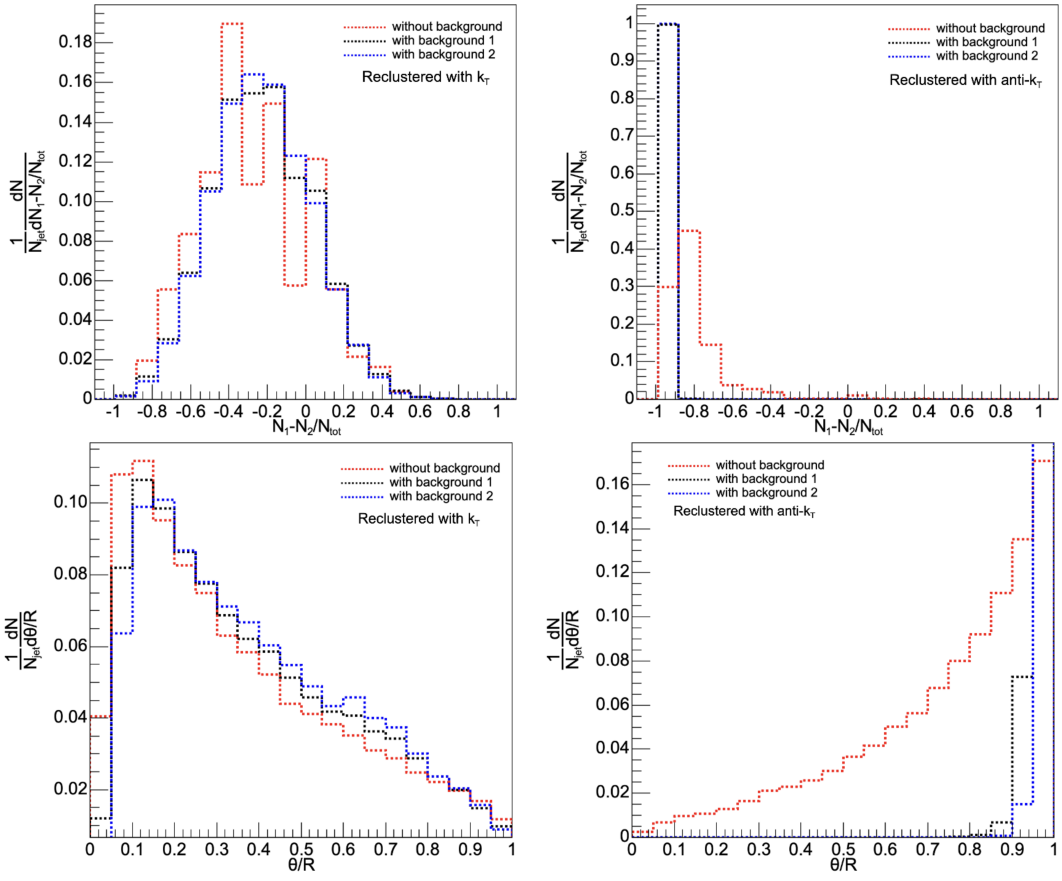


Figure 6.4: The ratio $\frac{N_1-N_2}{N_{tot}}$ (upper) and the opening angle θ (lower). *Left:* Jets reclustered with the k_T algorithm. *Right:* Jets reclustered with the anti-k_T algorithm.

The distribution of the jet observables are to some degree dependent on the choice of reclustering algorithm. Typically, one uses the anti-k_T to identify the initial jet, which is then reclustered using another algorithm before analyzing the observables.

The most commonly used reclustering algorithm for jet substructure analyses is the C/A algorithm. However, certain observables, such as the ratio of constituents in each subjet and the opening angle θ, are highly dependent on the clustering algorithm. Therefore, we wish to explore other reclustering algorithms, such as k_T and anti-k_T.

For jets clustered with the k_T algorithm, the upper left plot in Figure 6.4 shows that the distribution of constituents between the two subjets is more evenly distributed, with the peak close to 0. The differences between the jets with and without background particles are minor, whereas the distribution for jets without background radiation is slightly shifted towards more constituents in the second subject. The lower left plot in Figure 6.4 shows the θ distribution, which diverges significantly from the θ distribution for jets clustered with C/A. In k_T clustered jets, most of opening angles are very small and there are only minor differences between jets with and without background particles. The k_T algorithm clusters soft radiation first before incorporating the hardest particles towards the end, likely explaining the small opening angles observed..

Moving on to the upper right plot in Figure 6.4, we examine the distribution of constituents in the primary subjets of jets reclustered with anti-k_T. The distributions are vastly different from the previous algorithms, where the jets almost exclusively have the majority of the constituents in one subject. The jets without background particles have a slightly wider distribution but still has the majority of particles in one subjet. The lower right plot in Figure 6.4 displays the distribution of the opening angle θ for anti-k_T jets. Here, the majority of the jets have a large opening angle. In contrast to the other clustering algorithms, anti-k_T combines the hardest particles first, and subsequently merges them with softer particles. As a result, the final step involves combining the hard subjet with a soft one at the largest opening angle. Furthermore, the distribution of the opening angle for jets without background particles is wide, meaning that the jets can have large opening angles, while the jets with background particles consistently have small opening angles.

The choice of clustering algorithm does not only impact the distribution of jet observables but also has implications for jet grooming methods. In this thesis, we employ the SoftDrop grooming technique which typically utilizes C/A declustering. However, changing the clustering algorithm significantly affects the outcome of grooming [38]. To illustrate the effect of clustering algorithms on SoftDrop grooming, we examine the distribution of the number of grooming steps n_g.

The upper left plot in Figure 6.5 presents the distribution of n_g for jets without background particles, showcasing the distributions for the different clustering algorithms.

We begin by examining the distribution of C/A-clustered jets, which is typical for SoftDrop grooming. The number of grooming steps is generally quite small and is a middle ground between the other two distributions. A small number of grooming steps means that the effect of grooming is small, and the risk of losing crucial information in the jet is small. C/A jets typically have a small number of grooming steps because the jet is clustered based on geometrical distance, and that closely resembles the partonic branching history. We expect the Lund plane to be uniformly filled with radiation and so grooming should only remove a few branches.

For k_T clustering, grooming has a minimal effect, and the average number of grooming steps is even smaller than for C/A clustering. This result is reasonable considering that the k_T algorithm clusters soft partons with small angles early in the process, resulting in a fragmented jet. Therefore, the initial declustering produces two wide jets

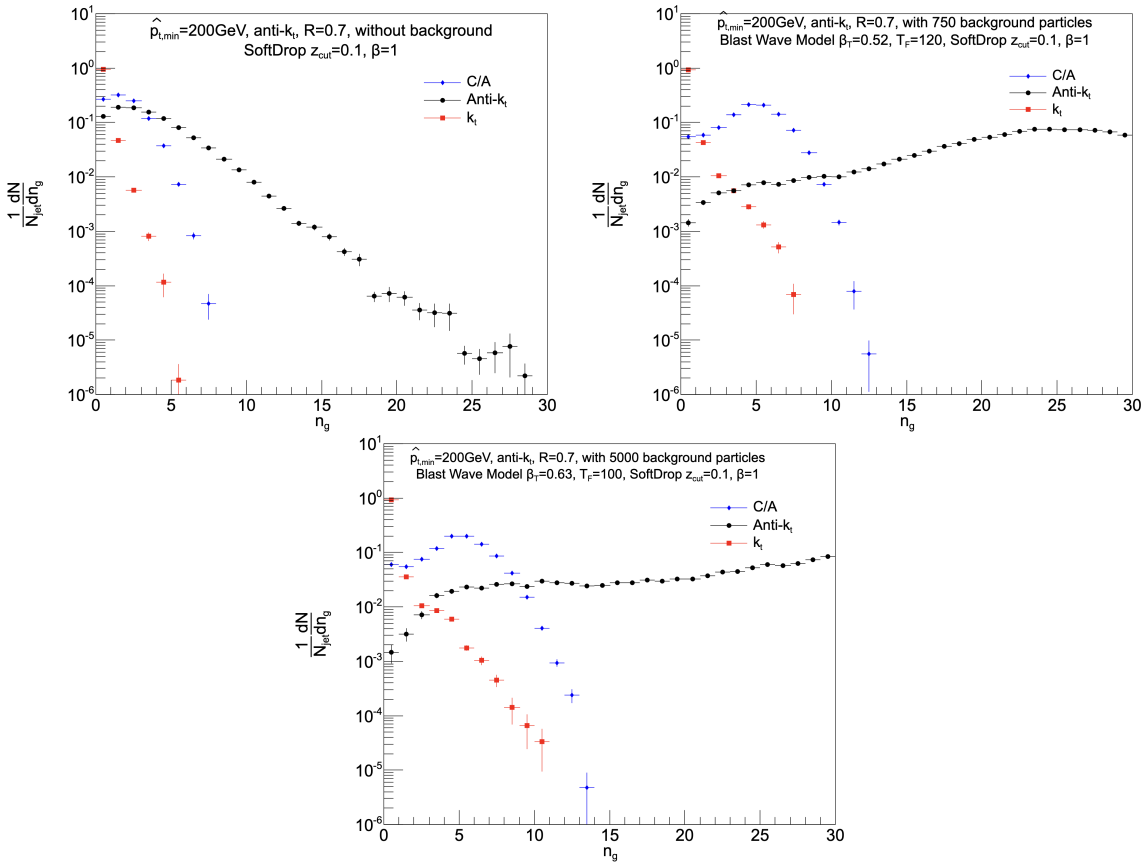


Figure 6.5: Effects of grooming on trees that are built up using different reclustering algorithms. *Upper Left*: For jets without additional background. *Upper Right*: For jets with additional 750 background particles. *Lower*: For jets with additionally 5000 background particles.

comprising both soft and hard particles

In contrast, when anti-k_T clustering is employed, the number of grooming steps can be significantly larger, often exceeding three times the number observed with other algorithms. Given that the anti-k_T algorithm initially combines the hardest partons, numerous soft partons are frequently clustered onto the hardest branch, making them susceptible to removal by SoftDrop. Removing such a substantial number of soft branches can lead to a less energetic jet and strongly affects the observables.

The upper right and lower plots in Figure 6.5 show the distribution of the number of grooming steps for jets with the two configurations of background. It is evident that the number of grooming steps increases for all clustering algorithms when background particles are present. However, a significant distinction remains between using anti-k_T and the other algorithms. The utilization of anti-k_T results in the removal of a considerable number of soft particles from the jets, leading to substantial impacts on the observables.

The distribution of the number of grooming steps presented in this section aligns with the findings in Ref. [38], reinforcing the choice of utilizing the C/A algorithm when applying SoftDrop grooming.

6.4 With SoftDrop Grooming

To minimize the presence of soft background particles, we will employ SoftDrop grooming. SoftDrop, as detailed in Chapter 3.5.2, will be utilized with specific parameters for this analysis, namely $z_{cut} = 0.1$ and $\beta = 1$. These SoftDrop parameter settings result in an aggressive grooming approach, targeting particles at larger angles. Now, we will examine the impact of SoftDrop grooming on the observables and investigate how it reduces the differences between jets with and without background radiation.

6.4.1 SoftDrop Observables

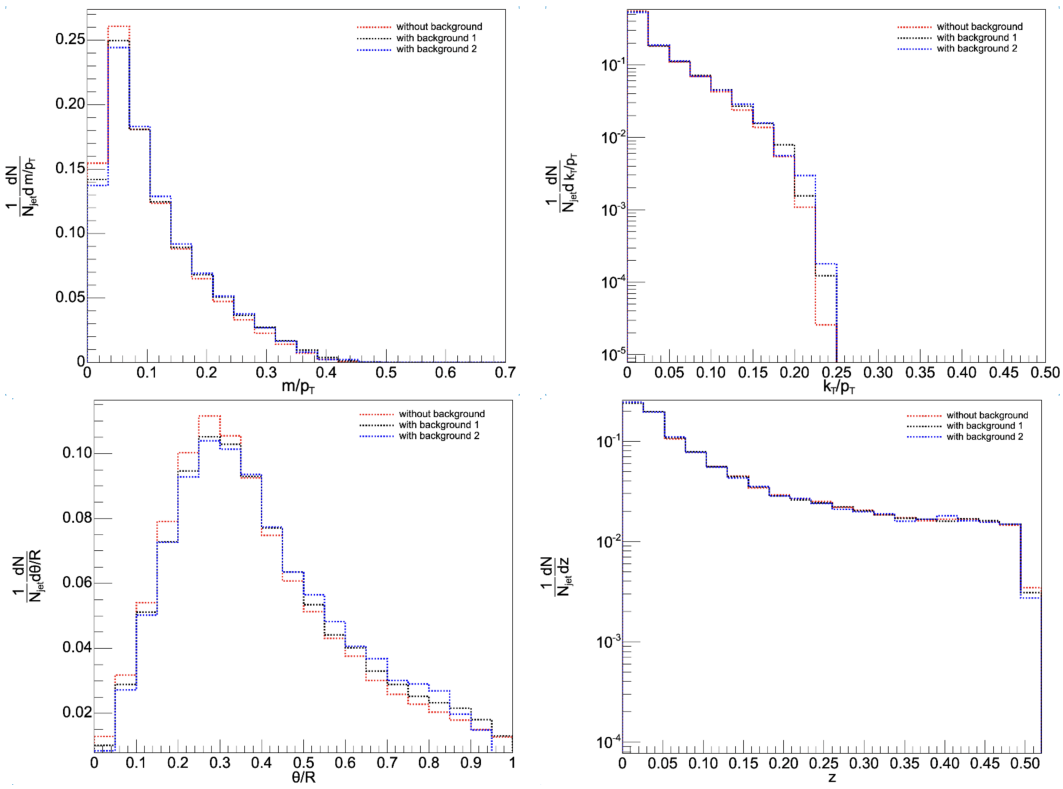


Figure 6.6: Plots of observable distribution for $\hat{p}_{t_{min}} = 200$GeV when applying SoftDrop grooming. *Upper Left*: Mass distribution. *Upper Right*: k_T distribution. *Lower Left*: θ distribution. *Lower Right*: z distribution.

In Figure 6.6 we observe an overall reduction of the difference between the distributions of the jets with and without background particles for all the observables.

Let us begin by examining the distribution of the jet mass in the upper left plot in Figure 6.6. Compared to the corresponding plot in Figure 6.2, both the jets with and without background radiation exhibit much more coherent mass distributions. Although there is still a slight tail in the distribution of jets with background radiation, the difference has significantly diminished.

Moving to the upper right plot in Figure 6.6, we observe that the k_T distributions of jets with background radiation still display a wider tail compared to the distribution of jets without background radiation.

In the lower left plot of Figure 6.6, the opening angle θ distributions exhibit notable differences from those in Figure 6.2. Prior to grooming, the majority of jets have a large opening angle θ, whereas after applying SoftDrop, the distribution shifts towards smaller opening angles. This indicates that grooming effectively removes soft particles in the first splitting. It is also interesting to note that the distributions of θ for jets with and without background radiation are very similar after grooming, whereas they were quite different in the ungroomed distributions.

Finally, in the lower right plot in Figure 6.6, the momentum-sharing fraction z distribution looks very similar to the distribution without grooming. However, the distribution without grooming had more of a dip from small z values to intermediate ones, while the distributions after grooming are flatter. For the z distributions after applying SoftDrop, then the distributions mostly overlap for the whole z scale.

Overall, the differences between the observables before and after applying SoftDrop grooming are mainly that the distributions for jets with and without background radiation are more coherent. Only the distribution of the opening angle θ has been significantly changed after applying SoftDrop, but the other distributions have also experienced some modifications.

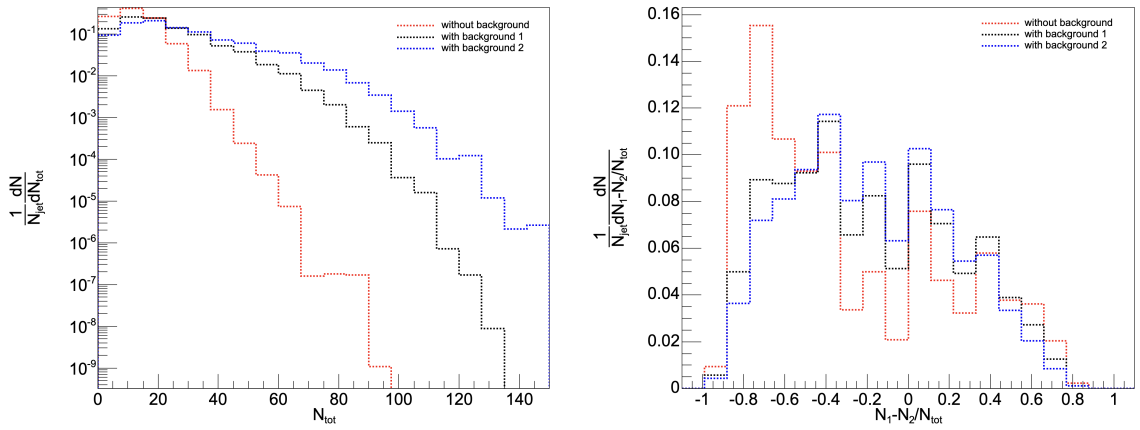


Figure 6.7: Plot distributions of the number of constituents in the jets with $\hat{p}_{t\,min} = 200$GeV, when applying SoftDrop grooming. *Left*: Distribution of the total number of constituents in the jet. *Right*: The ratio $\frac{N_1-N_2}{N_{tot}}$.

Now, let us examine the distribution of jet constituents among the first two subjets, considering our previous section's discussion. Applying SoftDrop grooming using the C/A algorithm removes some soft particles within the jet, resulting in a slight decrease in the overall number of constituents and potential shifts in the distribution of constituents between the leading subjets.

The left plot in Figure 6.7 displays the distribution of the total number of constituents in the jet N_{tot}. We observe that the distributions have shifted towards lower values compared to the corresponding ungroomed plot. While the distributions appear to peak around similar N_{tot} values, there still exists a significant gap in the tail between the number of constituents in jets with and without background radiation, as expected due to the additional particles introduced in the jets with background radiation. However, it is worth noting that the gap between the different distributions has reduced compared to before applying SoftDrop grooming.

If we examine the plot which shows the distribution of particles in the two subjects,

in the right plot in Figure 6.7, the distribution looks quite different. Before applying grooming, the majority of particles were concentrated in the second subjet. However, after applying SoftDrop, the distribution appears to have flattened out. There is no longer a pronounced majority in the second subjet, particularly in the distributions of jets with background radiation. This is intriguing, as the SoftDrop algorithm removes the first soft subjets, leaving the jet with two subjets that are fairly evenly populated.

In summary, applying SoftDrop grooming to jets with and without background radiation leads to more coherent distributions of observables. The gaps between the observables of jets with and without background radiation have diminished, and we observe a shift in the distribution of particles within the jets from a dominant presence in one subjet to a more evenly distributed configuration.

Alternative grooming techniques can be utilized to address this issue, or a more direct approach of implementing cuts on specific observables can be attempted. Appendix A presents a cautionary tale regarding the impact of placing cuts on observables. The main discovery of that detour is that when a cut is imposed on an observable, it leads to a very biased set of jets that no longer represent the true jet distribution.

Conclusion to the Background Study

When jets are embedded in a heavy-ion environment, they tend to contain significantly more particles. This increased number of constituent particles leads to **several noticeable effects in the jet observables**. Firstly, the mass of the jets and the opening angle between subjets increase, and the momentum-sharing fraction decreases. Secondly, the distribution of particles in the two subjets resulting from the first splitting becomes more uneven. In jets without background radiation, the majority of particles tend to be concentrated in one subjet, while in jets embedded in a thermal background, the distribution becomes wider, although the majority of constituents still remain in one subjet.

These studies are dependent on the **choice of clustering algorithm**. Typically, the anti-k_T algorithm is used to locate the jet, while a different clustering algorithm is applied to analyze the jet substructure. The choice of clustering algorithm also **determines the effect of SoftDrop grooming**. SoftDrop is typically performed with the use of the C/A algorithm, as anti-k_T clustering tends to remove too much of the jet, while k_T clustering removes almost nothing from the jet. After applying SoftDrop grooming to both the jets with and without background radiation, the resulting distributions become more similar. However, some differences still persist, particularly in the distribution of constituents among the two leading subjets.

Overall, SoftDrop has the effect of **removing a significant portion of the background radiation**, making the distributions of observables more similar between jets with and without background radiation. However, it is important to note that SoftDrop does not eliminate all background radiation and may inadvertently lead to the loss of vital information through the removal of soft particles.

Chapter 7

Training on Medium-Modified Jets

This chapter presents the main results of the thesis, which focuses on training a graph neural network to predict the energy loss experienced by a jet traversing a medium. As discussed earlier, the interaction between jet constituents and the medium leads to energy dissipation and the generation of soft thermal particles from the wake traveling through the medium. By comparing medium-modified jets with their vacuum counterparts, we can calculate the energy loss ratio χ using Equation 3.15.

To extract information about jet radiation patterns and substructure, the jets are transformed into a graph-like representation known as the Lund tree. A graph neural network is then employed to train a regression model for predicting the energy loss ratio χ. The GNN model used in this study is a modified version of LundNet, as detailed in Chapter 5.3.

The performance of the LundNet model is evaluated on two distinct datasets discussed in Chapter 5.4. The first dataset is generated using a parton gun, while the second dataset represents a more realistic dijet spectrum. The latter dataset is of particular interest as it resembles the conditions of a heavy-ion collision, except for the lack of a thermal background created by the QGP.

To gain a better understanding of energy loss in medium-traversing jets, we begin by analyzing the differences between the medium-modified jets and the vacuum jets. The most significant distinction lies in the transverse momentum p_T spectra, where the medium-modified jets exhibit a shift towards lower energies due to the energy loss. Additionally, we investigate the variation in the number of constituents within each jet. In particular, we expect an increase in the number of jet constituents, especially for hadron-level jets, as they contain numerous thermal particles from the wake generated by the jets' interaction with the quark-gluon plasma.

Consequently, our focus shifts to examining the GNN model's internal workings, particularly in relation to the realistic spectrum. We aim to gain insights into the inner mechanisms of the network.

Finally, we delve into the effects of embedding the jet in a heavy-ion environment. We assess the model's robustness and discuss whether the utilization of a GNN improves the prediction performance under conditions resembling real experimental data, in comparison to a previous study outlined in Ref. [5].

7.1 Parton Gun

In Chapter 5.4, we outlined the creation of the parton gun dataset, which involves tracing the trajectory of a parton through a thermal medium with a transverse momentum $p_T = 1000\text{GeV}$ and an average medium temperature of $T \approx 250\text{MeV}$. The jets are reconstructed using the anti-k_T algorithm with a radius parameter of $R = 0.4$. Subsequently, the medium-modified jets are paired with their corresponding vacuum jets, allowing us to extract the energy loss ratio χ.

In this analysis, we approach the study from both partonic and hadronic perspectives, thereby dividing it into two sections. Firstly, we examine the parton-level jets, followed by an investigation of the hadron-level jets. Subsequently, we compare the results obtained from these two levels and discuss the observed differences.

7.1.1 Parton Level

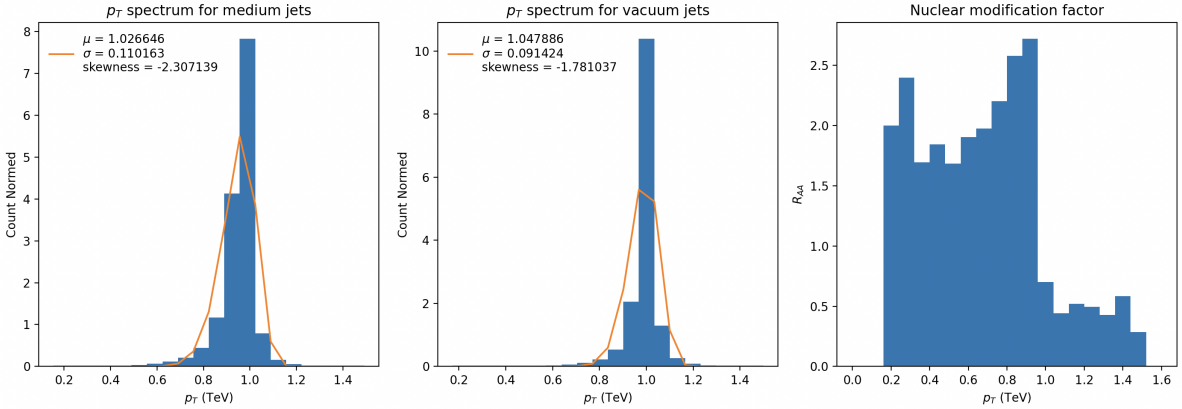


Figure 7.1: Jet p_T spectra for parton-level jets originating from a parton gun, with a skewed Gaussian fit and one plot containing the ratio of the two others. *Left:* p_T spectrum for the medium-modified jet. *Middle:* p_T spectrum for vacuum jets. *Right:* Distribution of the ratio between these spectra, also known as the nuclear modification factor R_{AA}.

We expect the p_T spectra of the parton gun datasets to be a Gaussian-like distribution around $\hat{p_T} = 1000GeV$. The p_T spectra of the medium-modified and vacuum jets are depicted in Figure 7.1. The medium-modified spectrum is characterized by a shift towards lower energies and a broader peak compared to the vacuum spectrum. Fitting a skewed Gaussian distribution to the data reveals that the medium-modified spectrum has a lower mean and a greater negative skewness, indicating a significant alteration in the distribution shape. Furthermore, the ratio between the spectra demonstrates an increase in the abundance of low-energy medium-modified jets, with up to three times more jets than the vacuum spectrum, while the number of high-energy jets decreases by approximately half. These findings highlight the substantial impact of medium modification on the transverse momentum distribution of jets.

Another important aspect of jets is the number of constituent particles they contain. The distribution of the number of constituents in both vacuum and medium-modified jets is illustrated in Figure 7.2. Interestingly, the medium-modified jets tend to have fewer constituents on average than the vacuum jets. Since the energy loss experienced

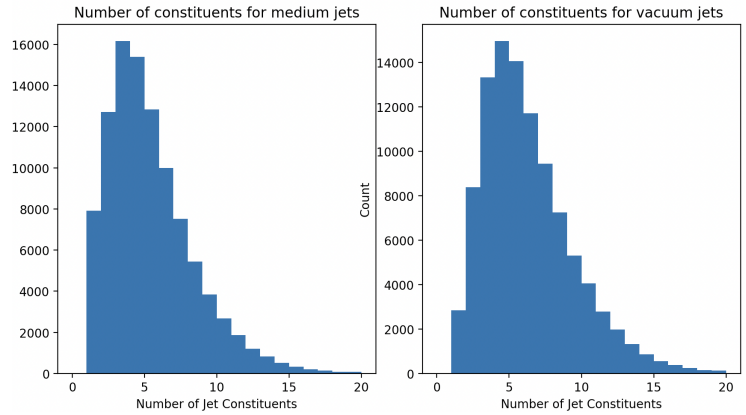


Figure 7.2: Distributions of the number of constituents for parton-level jet, originating from the parton gun. *Left:* Distribution for medium-modified jets. *Right:* Distribution for vacuum jets.

by a jet is proportional to the number of partons passing through the medium, jets with fewer constituents are expected to lose less energy. This observation allows us to utilize the distribution of constituent numbers as a predictive indicator for the χ distribution in medium-modified jets. Consequently, a significant peak at high χ values is expected in the χ distribution for medium-modified jets.

The low number of constituents in jets not only corresponds to low χ values but also results in a highly sparse graph for the Graph Neural Network (GNN). During training, the network relies on nodes representing parton splittings and their associated feature vectors. When there are few nodes, the network has limited information to work with.

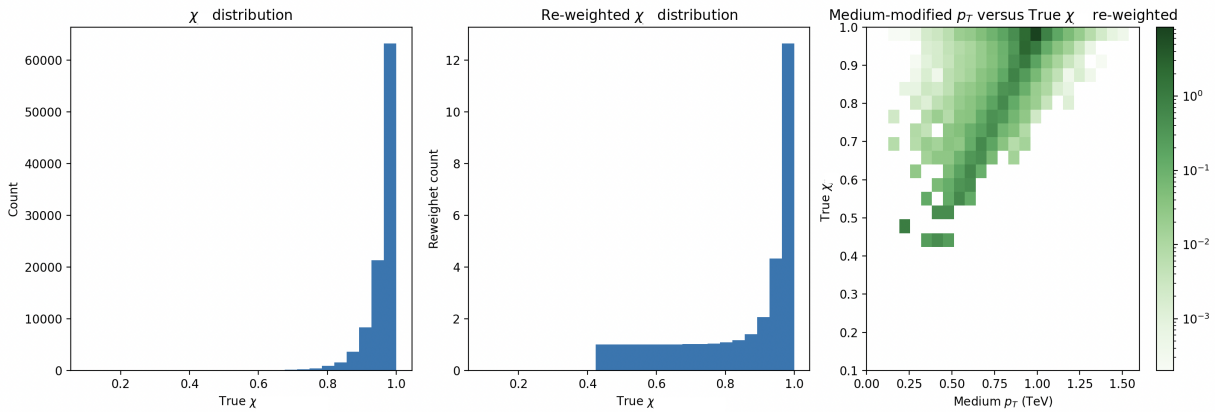


Figure 7.3: The distribution of χ for the jets on parton-level from the parton-gun dataset illustrates how much energy a jet has lost after traveling through a medium. *Left:* Distribution of χ without applying re-weighting. *Middle:* Distribution of χ after applying one-dimensional re-weighting. Note the change of scale on the y-axis. *Right:* The joint histogram of χ vs medium-modified p_T after the one-dimensional re-weighting.

Figure 7.3 presents three plots related to the χ distribution of parton gun jets. The left plot displays the true χ distribution, which predominantly consists of jets with high χ values. However, this imbalanced distribution poses challenges during network training. To address this issue, we assign weights to each sample in the loss function during training and validation, aiming to achieve a relatively flat one-dimensional χ

distribution. This re-weighting process, described in Chapter 5.4, results in a more balanced distribution of χ values, as shown in the middle plot of Figure 7.3. Note the change of scale on the y-axis, indicating that each bin contains fewer samples. By re-weighting the dataset, we obtain a more suitable distribution of χ values for training the network.

The joint histogram of χ and p_T of the medium-modified jet in the right plot of Figure 7.3 offer insights into how χ affects the p_T distribution. This plot provides a visual representation of the impact of χ on the p_T distribution and allows us to assess the influence of re-weighting on this distribution. By applying the re-weighting procedure, we achieve a more balanced distribution of χ values, which can enhance the performance of the network when applied to the dataset.

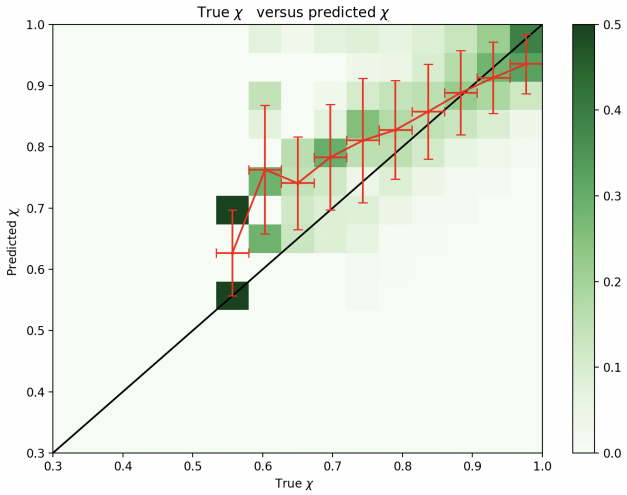


Figure 7.4: Plot illustrating the prediction performance for the parton-level jets from the parton-gun data. The green color represents the probability of predicted χ along the y-axis given true χ in the joint histogram. Each column is normalized here. The red line with error bars quantifies the average and standard deviation of the predicted χ within the given true χ bin.

Finally, the GNN is trained on parton-level jets from the parton gun dataset, utilizing the one-dimensional re-weighting of each sample to improve training. The training involves 50 epochs with a starting learning rate of $lr = 0.01$. The performance of the trained network is evaluated by visualizing the results in Figure 7.4. The plot shows a joint distribution of the predicted χ values normalized by the total number of events in each true χ bin, which is represented by the green columns. The red line with error bars represents the average and standard deviation of the predicted χ values within each true χ bin. We observed that the network struggles to accurately predict low χ values due to the limited number of training samples available. Furthermore, the dataset's χ values cover only a small range, approximately half of the full χ range, resulting in challenges for the network in accurately predicting these low χ values. Decent predictions are achieved for high χ bins; however, all true χ bins exhibit wide distributions of predicted χ values and large error bars, therefore, all results should be interpreted with caution.

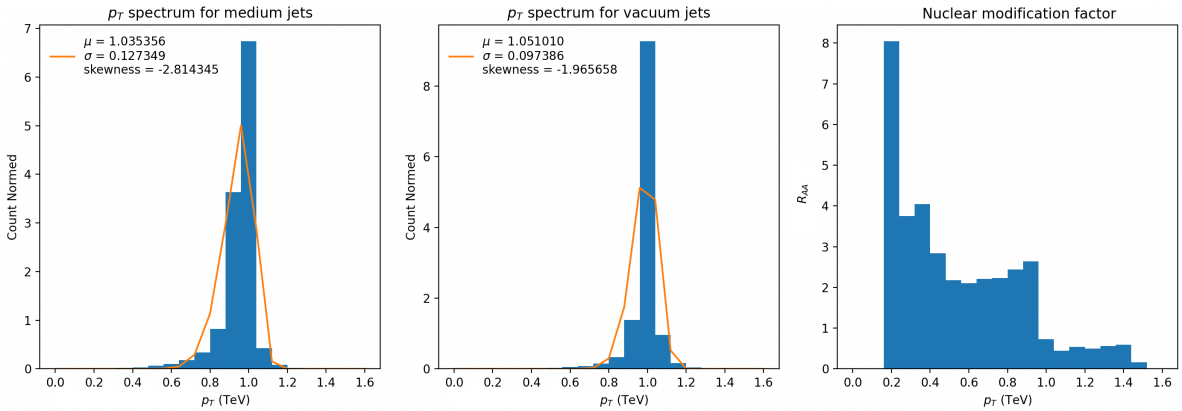


Figure 7.5: Jet p_T spectra for hadron-level jets originating from a parton gun, with a skewed Gaussian fit and one plot containing the ratio of the two others. *Left:* p_T spectrum for the medium-modified jet. *Middle:* p_T spectrum for vacuum jets. *Right:* Distribution of the ratio between these spectra, also known as the nuclear modification factor R_{AA}.

7.1.2 Hadron Level

The p_T spectra of medium-modified and vacuum hadron-level jets are depicted in Figure 7.5. As observed, the medium-modified spectrum exhibits a shift towards lower energies and a broader peak compared to the vacuum spectrum. Fitting a skewed Gaussian distribution to the data reveals that the medium-modified spectrum displays a lower mean and greater negative skewness than the vacuum spectrum, indicating a notable alteration in the distribution shape. This shift in shape is consistent with the observations made for parton-level jets, albeit the distribution of hadron-level jets is slightly wider and more skewed. Analyzing the ratio between the spectra, the nuclear modification factor R_AA, we find that the medium-modified spectra contain up to six times more jets in the low p_T bins compared to the vacuum spectrum, while the vacuum spectrum has a higher number of jets in the high p_T bins. These findings indicate that the medium modification has significantly shifted the p_T spectrum, mirroring the modifications observed in parton-level jets.

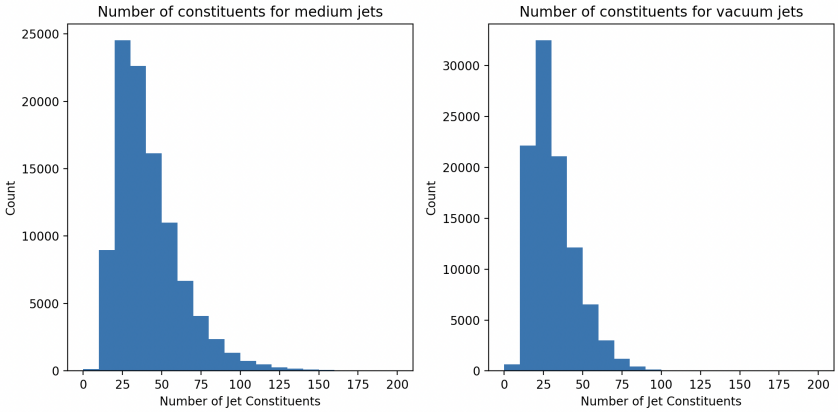


Figure 7.6: Distributions of the number of constituents for hadron-level jet, originating from the parton gun. *Left:* Distribution for medium-modified jets. *Right:* Distribution for vacuum jets.

Let us now turn our attention to the distribution of constituent particles in the jets. Figure 7.6 presents the distributions of the number of constituents for both the medium-

modified and vacuum cases. Surprisingly, we observe an opposite trend compared to the parton-level jets. In the case of hadron-level jets, there is a general increase in the number of constituent particles, and the medium-modified jets exhibit an even greater average number of constituents compared to the vacuum jets. As the energy loss of a jet is believed to be proportional the number of partons interacting with the medium, we can expect a wider distribution of χ for the hadron-level jets.

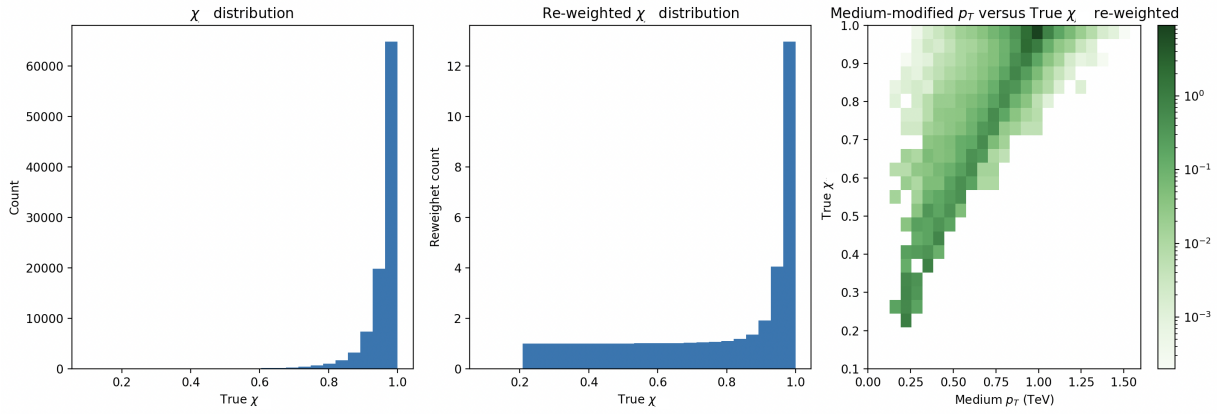


Figure 7.7: The distribution of χ for the jets on hadron-level from the parton-gun dataset illustrates how much energy a jet has lost after traveling through a medium. *Left:* Distribution of χ without applying re-weighting. *Middle:* Distribution of χ after applying one-dimensional re-weighting. Note the change of scale on the y-axis. *Right:* The joint histogram of χ vs medium-modified p_T after the one-dimensional re-weighting.

The χ distribution in the left plot in Figure 7.7 bears a resemblance to the parton-level distribution. However, it is noteworthy that the hadron-level distribution exhibits a longer tail. Similar to the parton-level case, the unbalanced nature of the χ distribution poses a challenge during network training. To address this, we employ the same one-dimensional re-weighting technique along the χ axis. The resulting re-weighted χ distribution is depicted in the middle plot of Figure 7.7. By applying this re-weighting procedure, we achieve a more balanced dataset where the low χ bins are now populated as well. Consequently, the dataset encompasses a broader range of χ values. Finally, in the right plot of Figure 7.7, we present the joint histogram of χ and p_T for the re-weighted dataset, allowing us to visualize the joint distribution and assess the impact of the re-weighting procedure.

Subsequently, we proceed to train the GNN using the parton-level jets obtained from the parton gun dataset, while incorporating the one-dimensional re-weighting of each sample to enhance the training process. The GNN was trained for 50 epochs, employing a initial learning rate of $lr = 0.01$. To assess the performance of the trained network, we present the visualized outcomes in Figure 7.8. The plot illustrates a joint distribution of the predicted χ values, which have been normalized by the total number of events in each true χ bin, represented by the green columns. The red line accompanied by error bars represents the average and standard deviation of the predicted χ values within each true χ bin. We observed that the network encountered challenges when making accurate predictions for the lower range of χ values due to the limited number of training samples available. However, in the high true χ bins, the average

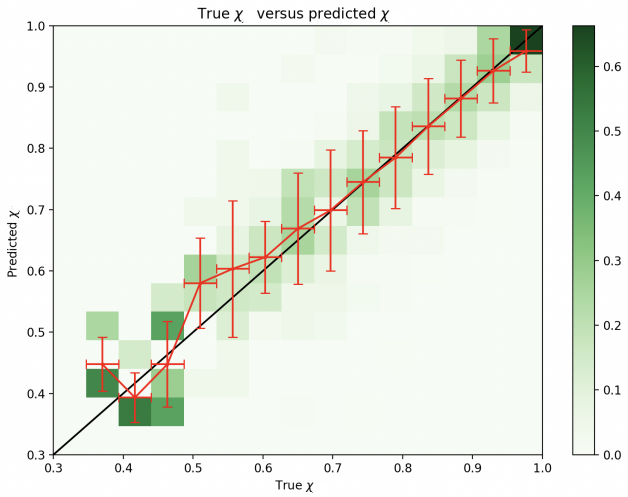


Figure 7.8: Plot illustrating the prediction performance for the hadron-level jets from the parton-gun data. The green color represents the probability of predicted χ along the y-axis given true χ in the joint histogram. Each column is normalized here. The red line with error bars quantifies the average and standard deviation of the predicted χ within the given true χ bin.

predicted values demonstrated a reasonably accurate performance. It is important to note that the light color of the distribution indicates a significant spread of the predicted values. Overall, the prediction performance exhibited the anticipated trend of being reliable for high χ bins and less accurate for low χ bins.

7.1.3 Comparison

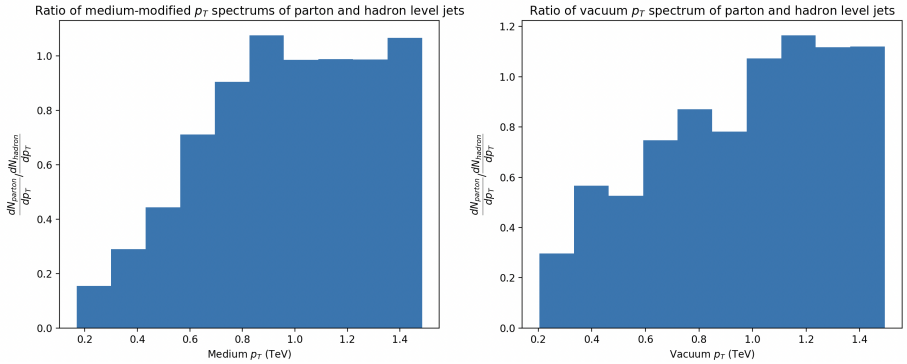


Figure 7.9: Ratio between parton- and hadron-level p_T spectrums originating from the parton-gun data. *Left:* Ratio of medium-modified jets p_T spectrum. *Right:* Ratio of vacuum jets p_T spectrum.

Having examined both the parton-level jets and the hadron-level jets, we can now compare their p_T spectra by analyzing the ratios between the two distributions. The left plot in Figure 7.9 displays the ratio of the p_T spectrum for medium-modified jets at the parton and hadron levels. It reveals that the hadron-level jets have a greater abundance of jets in the low p_T bins compared to the parton-level jets. Conversely, in the high p_T bins, the two distributions are quite similar, while the parton-level spectrum demonstrates a scarcity of jets in the low p_T region. This discrepancy arises due to the fact that only a few parton-level jets possess small χ values, indicating a limited loss

	Medium-modified		Vacuum	
	Parton	Hadron	Parton	Hadron
μ	1.026	1.035	1.048	1.051
σ	0.110	0.127	0.091	0.097
skewness	-2.307	-2.814	-1.781	-1.965

Table 7.1: Table of parameter values for the p_T spectrum fits for the jets originating from the parton-gun.

of energy. Consequently, a smaller number of jets have low medium-modified p_T. In the right plot of Figure 7.9, we observe a similar trend in the ratio for vacuum jets, with fewer parton-level jets in the low p_T bins. However, in the high p_T bins, there is an increase in the number of parton jets. The effect of medium modifications has shifted the surplus parton level jets towards lower energies, creating a more evenly steeply falling ratio.

Furthermore, we can compare the parton-level and hadron-level p_T spectra through spectrum fits. By fitting the p_T spectra with skewed Gaussians, we obtain the parameters summarized in Table 7.1. The μ and σ values of the parton-level distributions are shifted towards smaller values compared to the hadron-level distributions, while the skewness is greater. However, the differences between the p_T distributions are relatively minor.

Input	Average MSE Loss
Parton level jets	0.00289
Hadron level jets	0.00134

Table 7.2: Table of average MSE loss for parton- and hadron-level jets from the parton-gun dataset.

To quantify the model's performance on the two datasets, we utilize the average MSE loss. The corresponding results are presented in Table 7.2. It is evident that the network performs better on the hadron-level jets, which aligns with our observations from the prediction performance plots in Figure 7.4 and 7.8.

7.2 Realistic Jet Spectrum

In accordance with Chapter 5.4, the realistic jet spectrum dataset was generated based on a complete dijet spectrum with $\hat{p_T} = 50GeV$. The jets are reconstructed using anti-k_T algorithm with a radius parameter $R = 0.4$, and a minimum transverse momentum requirement of $p_T^{jet} > 100GeV$ is imposed.

Similar to the approach taken in analyzing the parton gun data, this analysis is also divided into two distinct parts. First, we examine the jets at the parton level, followed by an investigation of the jets at the hadron level. Subsequently, we compare the results obtained from both levels and discuss the observed differences.

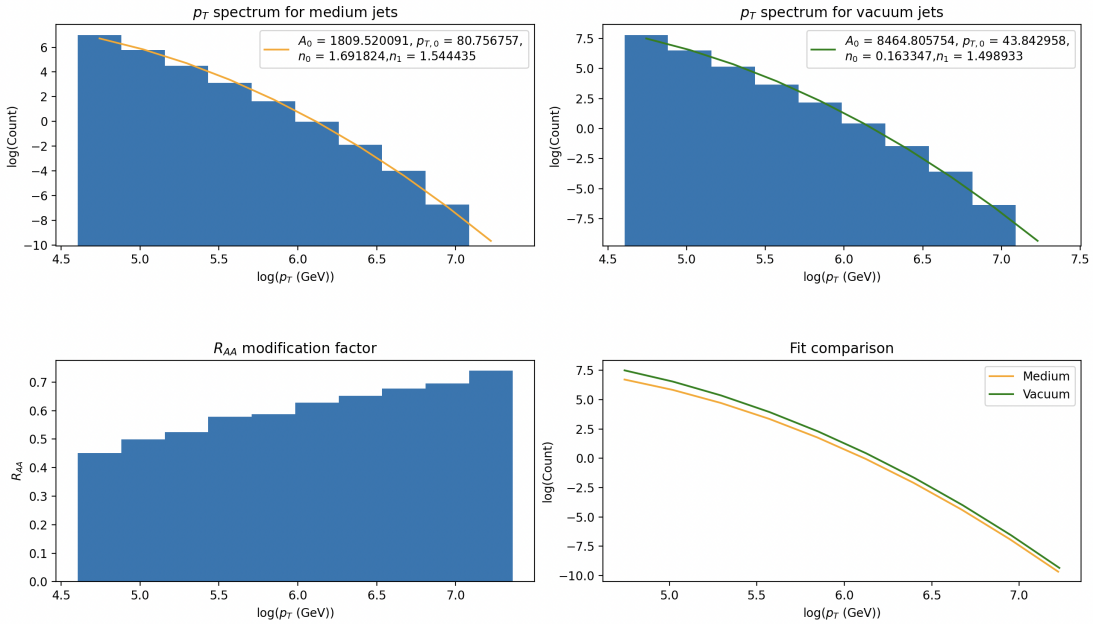


Figure 7.10: Plots relating to the jet p_T spectra for parton-level jets originating from the realistic dijet spectrum. *Upper Left:* p_T spectrum for the medium-modified jet with $p_T > 100$GeV and fitted to the curve in Eq. 7.1. *Upper Right:* p_T spectrum for vacuum jets with $p_T > 100$GeV and fitted to the curve in Eq. 7.1. *Lower Left:* Distribution of the ratio between these spectra, also known as the nuclear modification factor R_{AA}. *Lower Right:* Comparing the fits for medium-modified and vacuum jet spectra.

7.2.1 Parton Level

The p_T spectra of jets originating from the realistic dijet spectrum exhibit a steeply falling distribution. In contrast to the Gaussian fitting used for the parton gun p_T spectra, we now aim to fit a curve to these rapidly declining spectra for a better representation. By plotting the distributions with logarithmic axes, we seek to find a suitable curve to describe the spectra. Specifically, we fit the spectra to the logarithm of the following curve:

$$\frac{d\sigma}{dp_T} = A_0\left(\frac{p_{T,0}}{p_T}\right)^{n(p_T)}, \quad n(p_T) = n_0 + n_1 \log\left(\frac{p_T}{p_{T,0}}\right) + \ldots \tag{7.1}$$

The shape of the curve is motivated by perturbative QCD, where $\hat{\sigma} \sim 1/Q^4$ from Eq. 2.13, and the parameter $n(p_T)$ accounts for scaling violations in the parton evolution functions.

The upper plots in Figure 7.10 display the p_T spectra of medium-modified and vacuum jets, respectively, fitted to the curve in Eq. 7.1. The fits accurately capture the curvature of the distributions.

Upon comparing the fits for the medium-modified and vacuum spectra, as depicted in the lower right plot of Figure 7.10, we observe that the medium-modified spectrum is shifted towards lower values. The difference between the vacuum and medium-modified fits is most prominent in the low p_T bins and diminishes in the high p_T bins, although the medium-modified jets consistently have a smaller number of particles in each bin.

A similar trend is also noticeable in the R_{AA} plot presented in the lower left plot of Figure 7.10. Generally, the medium-modified spectrum has fewer samples in each p_T bin compared to their vacuum counterparts. This indicates that several medium-modified jets are absent from these plots. This absence is a result of imposing a $p_T > 100GeV$ cut on both the medium-modified and vacuum jets. The next section will address the selection bias effect in detail, but it essentially implies that all vacuum jets whose modified energy is less than $100GeV$ due to energy loss are not accounted for in these plots. Consequently, the p_T spectrum for medium jets will exhibit a low p_T tail, occupied by the missing jets. We will revisit these missing jets shortly. However, the network is trained on all the jets, even the ones not shown in these plots due to the p_T cut on the medium-modified jets.

However, the crucial findings from these plots are that the vacuum jets have experienced substantial energy loss, shifting the medium-modified p_T spectrum towards lower values.

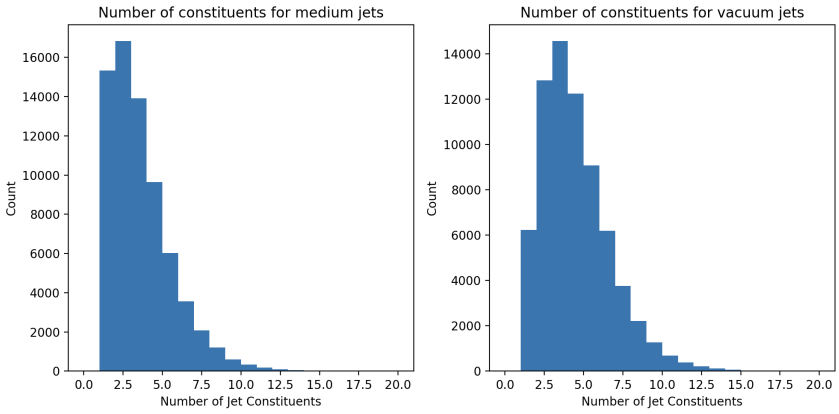


Figure 7.11: Distributions of the number of constituents for parton-level jet, originating from the realistic dijet spectrum. *Left:* Distribution for medium-modified jets. *Right:* Distribution for vacuum jets.

The plots displayed in Figure 7.11 illustrate the distribution of the number of constituents in both the vacuum and medium-modified jets. Similar to the distributions observed in parton-level jets from the parton gun, the medium-modified jets generally have a lower average number of constituents compared to the vacuum jets. This trend suggests that we should anticipate a narrow distribution of χ values, akin to what was observed previously.

However, it is worth noting that the average number of constituents for parton-level medium-modified jets from the parton gun is approximately five, whereas the left plot in Figure 7.11 shows an average of three constituents. A jet with three constituents implies that there are at most two splittings in the clustered tree. Consequently, a graph with only two nodes does not contain a substantial amount of information about the jet. As a result of these sparsely populated graphs, we can expect the performance of the network to be severely impacted.

The observations from the plots in Figure 7.12 deviate from our initial expectations. In the left plot, the true distribution of χ appears to be significantly different from the narrow distribution observed in parton-level jets from the parton gun. Instead, the distribution appears more like an exponential graph, spanning across most of the χ range.

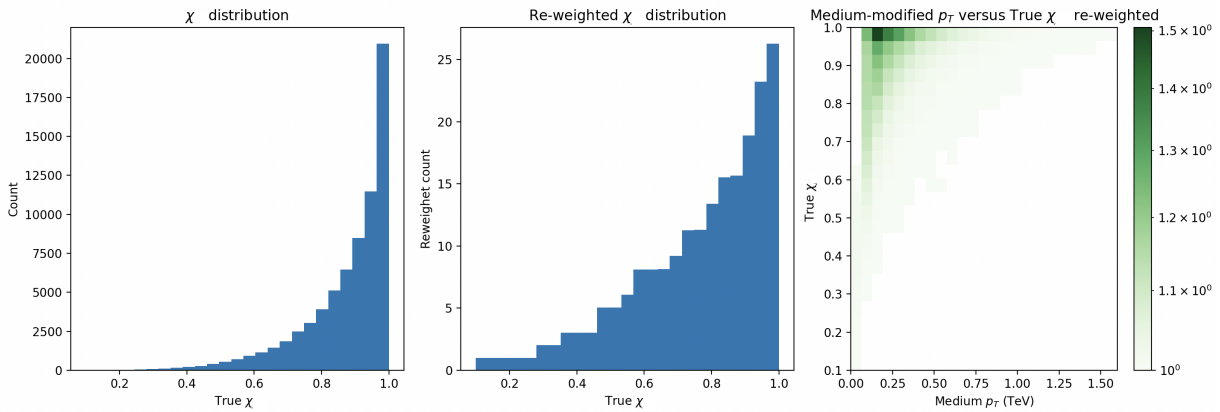


Figure 7.12: The distribution of χ for the jets on parton-level from the realistic dijet spectrum dataset illustrates how much energy a jet has lost after traveling through a medium. *Left:* Distribution of χ without applying re-weighting. *Middle:* Distribution of χ after applying two-dimensional re-weighting. Note the change of scale on the y-axis. *Right:* The joint histogram of χ vs medium-modified p_T after the two-dimensional re-weighting.

Given the unbalanced nature of the dataset, it becomes necessary to perform re-weighting before proceeding with network training. In this case, we employ a two-dimensional re-weighting approach within the p_T versus χ joint histogram, following the procedure outlined in Chapter 5.4, but applied to each (p_T, χ) bin individually. The impact of the re-weighting process becomes more apparent when examining the joint histogram in the right plot of Figure 7.12. In this visualization, we observe a more evenly distributed pixelated weight pattern across the (p_T, χ) space. Furthermore, the middle figure displays the re-weighted distribution of χ, which demonstrates a more balanced distribution with a greater emphasis on the low χ bins.

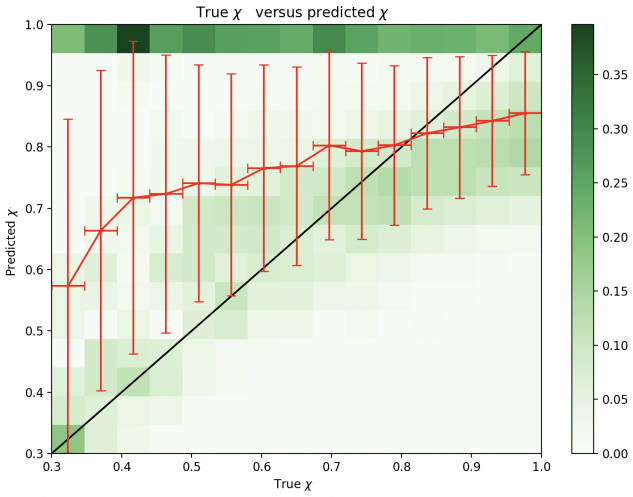


Figure 7.13: Plot illustrating the prediction performance for the parton-level jets from the realistic dijet spectrum data. The green color represents the probability of predicted χ along the y-axis given true χ in the joint histogram. Each column is normalized here. The red line with error bars quantifies the average and standard deviation of the predicted χ within the given true χ bin.

We can now proceed with training the GNN on the given dataset. The prediction performance is depicted in Figure 7.13, and unfortunately, the results are far from sat-

isfactory. The network consistently produces high values for most of the samples, indicating a lack of accuracy. However, we do observe some instances where the predicted values align more closely with the true χ values, as indicated by the lighter green areas along the diagonal.

The poor performance of the network poses a challenge, and we are currently investigating the underlying cause. Our primary hypothesis is that the low number of constituents in the jets plays a significant role. In cases where there is only one or two nodes present in the graph, the network receives minimal information to distinguish between different graphs. Consequently, if the feature values among the nodes are similar, the network struggles to discern the variations between graphs with large or small χ values. For jets with a minimal number of constituents, it appears that the network resorts to "guessing" rather than making informed predictions

Comparatively, the parton-level jets from the parton gun dataset have slightly more constituents per jet. This increased information content may enable the network to better capture the distinctions. Aside from that, identifying the factors contributing to the difference in performance between the parton-level jets from the parton-gun dataset and the realistic jet spectrum dataset is challenging.

7.2.2 Hadron Level

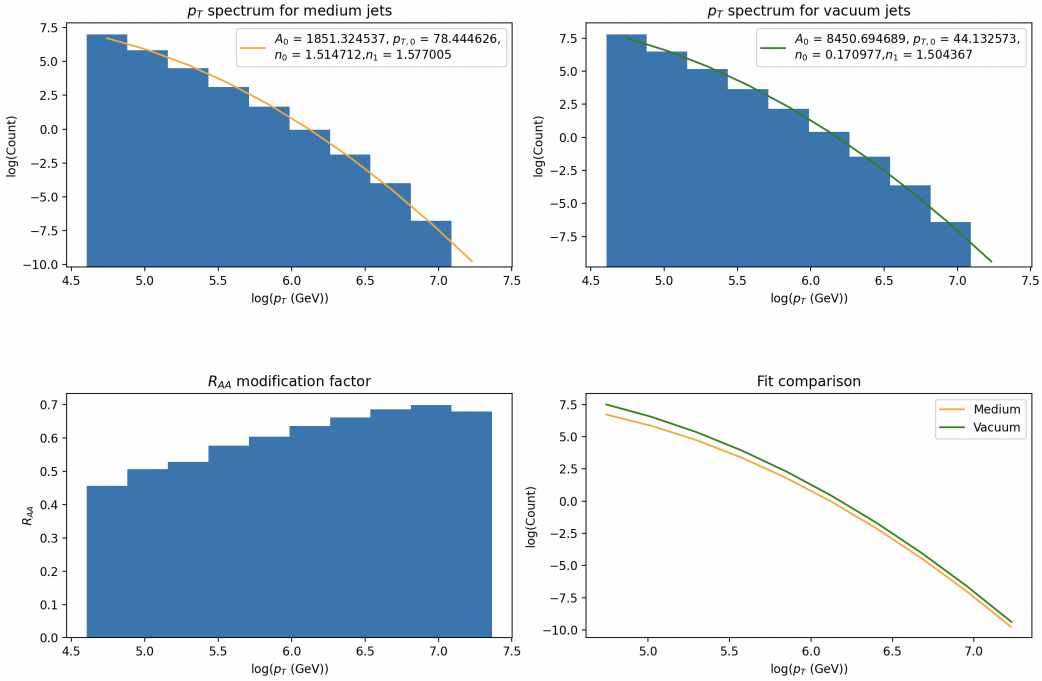


Figure 7.14: Plots relating to the jet p_T spectra for parton-level jets originating from the realistic dijet spectrum. *Upper Left:* p_T spectrum for the medium-modified jets with $p_T > 100$GeV and fitted to the curve in Eq. 7.1. *Upper Right:* p_T spectrum for vacuum jets with $p_T > 100$GeV and fitted to the curve in Eq. 7.1. *Lower Left:* Distribution of the ratio between these spectra, also known as the nuclear modification factor R_{AA}. *Lower Right:* Comparing the fits for medium-modified and vacuum jet spectra.

The upper plots in Figure 7.14 depict the p_T spectrum of medium-modified and vacuum jets. The plots adopt logarithmic axes and have been fitted with the logarithm of

the curve described in Eq. 7.1. It is evident that both spectra are steeply falling, with the fitted curve appearing almost linear. Comparing the fits, we observe a noticeable shift towards lower p_T values for the medium-modified spectrum, as shown in the bottom right plot of Figure 7.14. This difference between the fitted curves is particularly pronounced at low p_T values.

Similar trends are also evident in the R_{AA} plot presented in the lower left plot of Figure 7.14. In this plot, we observe a consistent reduction in the number of samples for the medium-modified spectrum across all p_T bins, in comparison to their vacuum counterparts. The R_{AA} plot bears resemblance to that of the parton-level jets, with the exception of a flatter plateau observed in the high p_T bins. Once again, it is important to note that the medium-modified spectrum consistently contains fewer jets in each bin compared to the vacuum spectrum. This discrepancy arises from the $p_T > 100GeV$ threshold applied to both the medium-modified and vacuum jets, making it easier to compare the two distributions. Consequently, the p_T spectrum for the medium-modified jets displays an extended tail at lower p_T values.

Nonetheless, it is crucial to emphasize the significant shift towards lower p_T values induced by the medium modification of the jets.

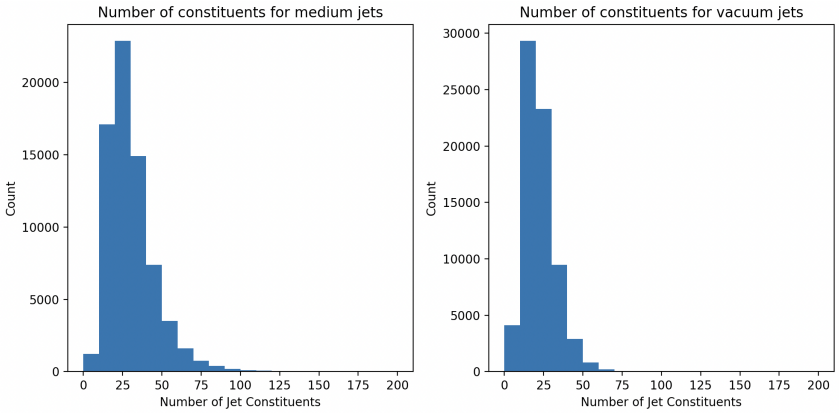


Figure 7.15: Distributions of the number of constituents for hadron-level jets, originating from the realistic dijet spectrum. *Left:* Distribution for medium-modified jets. *Right:* Distribution for vacuum jets.

The plots in Figure 7.15 illustrate the distribution of constituent particles in jets. Notably, the medium-modified jets tend to possess a greater number of constituents compared to the vacuum jets. Although the difference between the two distributions is not substantial, we observe a slightly broader distribution with a marginally higher peak for the medium-modified jets. Given the significant number of constituent particles involved, it is reasonable to anticipate a wider distribution of χ values. This expectation arises from the fact that energy loss scales with the number of partons traversing the medium.

The distribution of χ values depicted in the left plot of Figure 7.16 exhibits an exponential curve with a slightly wider spread compared to other datasets. To address the dataset imbalance, we apply two-dimensional re-weighting to the joint p_T vs χ histogram, following the procedure outlined in Chapter 5.4. The resulting joint distribution is presented in the right plot of Figure 7.16, while the re-weighted χ distribution is displayed in the middle plot. Notably, the re-weighted χ distribution achieves better balance, and we notice the altered scale on the y-axis. Following the re-weighting

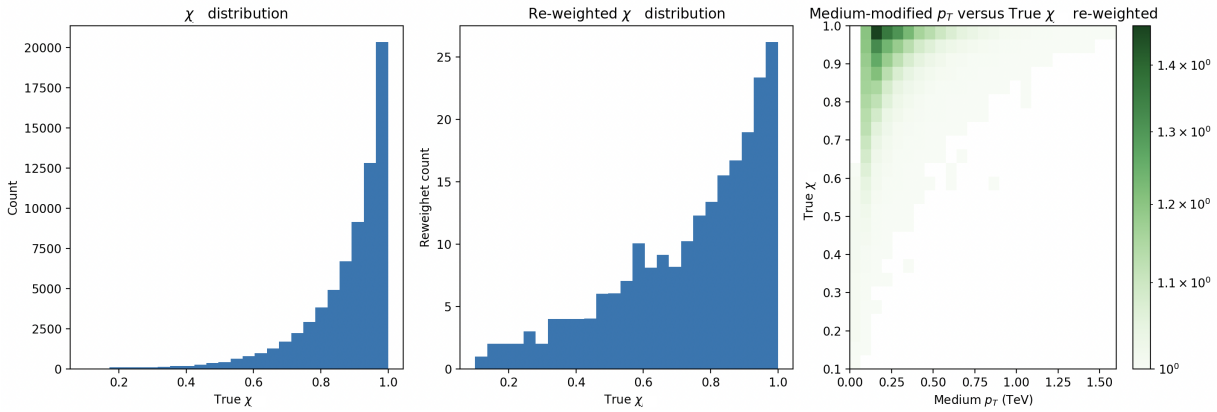


Figure 7.16: The distribution of χ for the jets on hadron-level from the realistic dijet spectrum dataset illustrates how much energy a jet has lost after traveling through a medium. *Left:* Distribution of χ without applying re-weighting. *Middle:* Distribution of χ after applying two-dimensional re-weighting. Note the change of scale on the y-axis. *Right:* The joint histogram of χ vs medium-modified p_T after the two-dimensional re-weighting.

process, the samples in the joint histogram exhibit a more even distribution across the (p_T, χ) space.

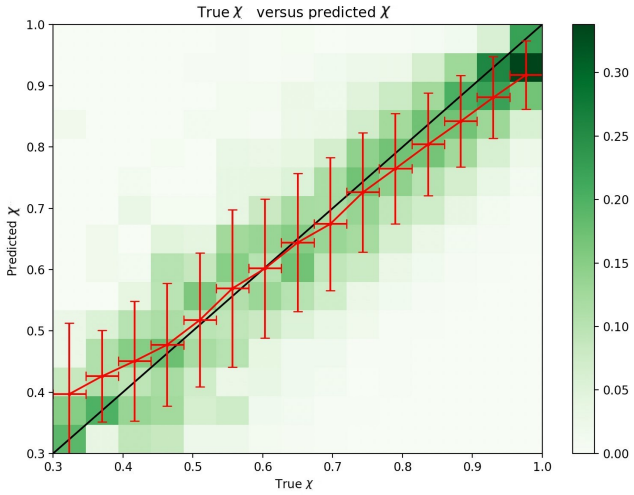


Figure 7.17: Plot illustrating the prediction performance for the hadron-level jets from the realistic dijet spectrum data. The green color represents the probability of predicted χ along the y-axis given true χ in the joint histogram. Each column is normalized here. The red line with error bars quantifies the average and standard deviation of the predicted χ within the given true χ bin.

Once the dataset is balanced, we proceed with training using the re-weighted dataset. The training process comprises 50 epochs with an initial learning rate of $lr = 0.01$. The performance of the network on the re-weighted dataset is depicted in Figure 7.17.

The network demonstrates its best performance on this dataset. The predictions for χ cover the entire range successfully. Across the entire χ range, the average of each true χ bin aligns closely with the diagonal. For high χ values, the average tends to be slightly lower than the true value, while for low χ values, the average tends to be

slightly higher. The predicted χ values exhibit a relatively narrow distribution around the average, and the error bars decrease as χ increases.

This result can be compared to the findings in [5], as both studies utilize machine learning methods and focus on hadron-level jets derived from similar jet spectra. However, the results obtained using a GNN are less accurate compared to those using a CNN. The predictions generated by the GNN exhibit wider distributions for each true χ value and larger error bars. One noticeable distinction when employing the GNN is that the predicted values for high true χ tend to be lower than expected. The red line representing the average predicted value for each true χ bin lies slightly below the diagonal for the CNN, and even lower for the GNN, particularly for high true χ values. The GNN predictions for low true χ bins display substantial error bars, thereby requiring cautious interpretation. Overall, similar trends emerge in the results obtained using both the GNN and CNN.

It is worth noting that the CNN underwent longer training and employed slightly different hyperparameters. Nevertheless, we can still infer that the CNN outperforms the GNN in predicting χ. We do not know why the CNN outperforms our GNN model, but perhaps the predictions of the energy loss are more accurate when the network obtains less information for each jet.

7.2.3 Comparison

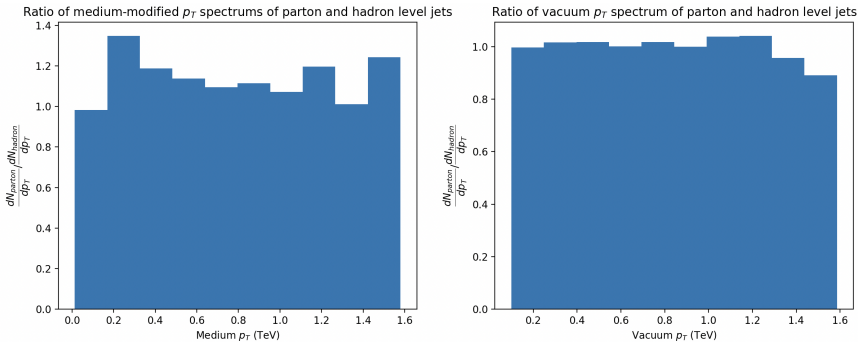


Figure 7.18: Ratio between parton- and hadron-level p_T spectrums originating from the realistic dijet spectrum data. *Left:* Ratio of medium-modified jets p_T spectrum. *Right:* Ratio of vacuum jets p_T spectrum.

	Medium-modified		Vacuum	
	Parton	Hadron	Parton	Hadron
A_0	1809.52	1851.32	8464.80	8450.69
$p_{T,0}$	80.76	78.44	43.84	44.13
n_0	1.692	1.515	0.163	0.171
n_1	1.544	1.577	1.499	1.504

Table 7.3: Table of parameter values for the p_T spectrum fits for jets originating from the realistic dijet spectrum dataset.

Having examined the parton-level and hadron-level jets, we now proceed to compare their p_T spectra. Firstly, we analyze the ratio of the p_T spectrum between medium-modified parton-level jets and hadron-level jets, as illustrated in the left figure of Figure

7.18. Notably, the ratio remains consistently close to one, indicating a remarkable similarity between the two spectra. There is a slightly larger number of samples in the high p_T bins for parton jets.

Next, we explore the ratio of the p_T spectrum between vacuum jets at the parton level and the hadron level, depicted in the right figure. Here, the ratio appears to be relatively flat, suggesting minimal differences between the two spectra. The only discernible distinction is a slightly higher number of samples in the high p_T range for hadron-level jets.

Additionally, we can compare the fitted curves of the p_T spectra for medium-modified and vacuum jets at the parton and hadron levels by examining the parameter values in Equation 7.1. Table 7.3 presents the parameter values for the various cases. Upon comparing the fits for parton-level and hadron-level jets, we observe minimal differences in the parameter values between medium-modified jets and vacuum jets. Therefore, we can conclude that there are negligible differences between the p_T spectra at the hadron level and the parton level. This is expected from perturbative QCD, as hadronization and non-perturbative physics should only mildly change high-p_T observables. Especially the p_T spectrum should experience very little change as p_T is the dominant scale, while substructure observables are more affected as they span over larger energy scales.

Input	Average MSE Loss
Parton level jets	0.03103
Hadron level jets	0.00777
Ref. [5] Hadron level jets	0.0028

Table 7.4: Table of average MSE loss for parton- and hadron-level jets from the realistic jet spectrum dataset.

Finally, we proceed to compare the model's performance on the two datasets derived from the realistic jet spectrum data. Table 7.4 displays the average MSE loss for the two datasets. It is evident that the parton-level jets perform poorly compared to the hadron-level jets. As discussed earlier, the parton level jets exhibit a limited number of constituents and sparse graphs, which likely contribute to the performance challenges.

However, the results obtained for the hadron-level jets are of utmost importance, as this dataset is most realistic of the datasets we have considered so far, as it is a more realistic scenario than the parton-gun dataset and what we measure in the detector are hadron jets. It is worth noting that this dataset aligns with the one utilized in Reference [5], enabling a meaningful comparison between the CNN and GNN models. Unfortunately, our results for the hadron level jets do not match the performance achieved in Reference [5], where an average loss of 0.0028 was accomplished.

7.3 Selection Bias Effect

In Chapter 5, we briefly discussed the concept of selection bias when outlining the objective of this analysis. Selection bias arises when applying a p_T cut to a steeply falling spectrum, making it challenging to capture jets that have undergone significant

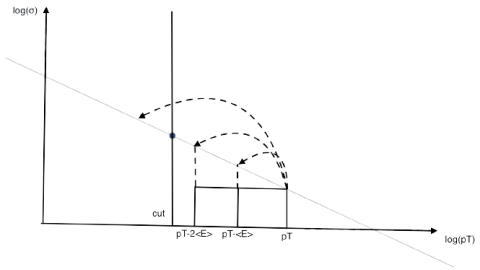


Figure 7.19: Sketch of the bias effect.

energy loss. This effect is evident in the observed shift in the spectrum, as depicted in the lower right plot of Figure 7.14.

When dealing with a rapidly decreasing p_T spectrum, selecting jets based on their medium-modified p_T introduces a bias towards jets that have experienced minimal energy loss [65]. Jets with substantial energy loss would have had higher energies if not quenched, but high-energy jets are less frequently produced in collisions. Hence, selecting jets above a certain p_T threshold implies a lower likelihood of capturing jets with significant energy loss. This bias effect is illustrated in Figure 7.19, highlighting the potential loss of jets with substantial energy loss due to the applied cut.

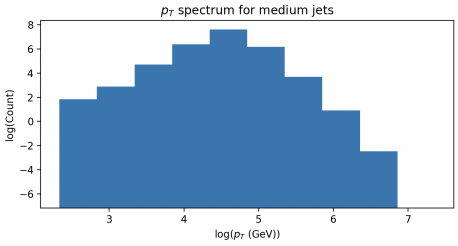


Figure 7.20: The full p_T spectrum for the jet spectrum parton-level jets.

In the left plot of Figure 7.10, we observe the medium-modified p_T spectrum after applying a $p_T < 100GeV$ cut. Notably, it consistently exhibits fewer jets in each p_T bin compared to the vacuum spectrum. The missing jets predominantly populate the medium-modified p_T bins below $100GeV$. The complete p_T spectrum can be found in Figure 7.20. Except for the p_T plots, no p_T cuts will be imposed on the medium-modified jets to include all jets originating from vacuum jets with $p_T > 100GeV$. However, for the purpose of comparing the vacuum and medium-modified jet p_T within the range of $100GeV - 1500GeV$, it is more convenient to utilize the p_T cut.

7.4 Looking inside the GNN

To gain a deeper understanding of our analysis results, it is crucial to delve into the inner workings of the Graph Neural Network (GNN). While Convolutional Neural Networks (CNNs) offer sophisticated tools to examine different layers and their representations, similar tools are not yet standard for GNNs.

Therefore, in our attempt to comprehend the internal dynamics of the GNN, we will focus on several key aspects of the network. Firstly, we will examine the structure of the input graphs themselves and how they are presented to the network. This exploration

will provide insights into the initial information fed into the GNN. Secondly, we will investigate the impact of the number of input features supplied to each node on the model's performance. By varying this aspect, we can explore whether the additional information provided to the network is beneficial or simply add noise to the network.

Lastly, we will explore the effects of removing specific information related to the jets. This will be achieved by imposing a k_T cut on the jets, without adding any background radiation at this stage of the analysis. By applying this cut, we effectively remove constituents that have undergone energy loss or soft particles associated with the jet wake. This investigation will shed light on the importance of such information for the model's performance.

Another idea to examine the input to the network would be to examine how certain jet observables are influenced by the amount of energy loss experienced by the jet. This analysis would help us understand the relationship between the amount of energy loss and various observable properties of the jets, providing valuable insights into the underlying physics. This will not be part of the main analysis, but a short study on this can be found in Appendix B.

By conducting these investigations, we aim to improve our understanding of the GNN and its behavior, even in the absence of comprehensive visualization tools.

7.4.1 Visualizing the Graphs

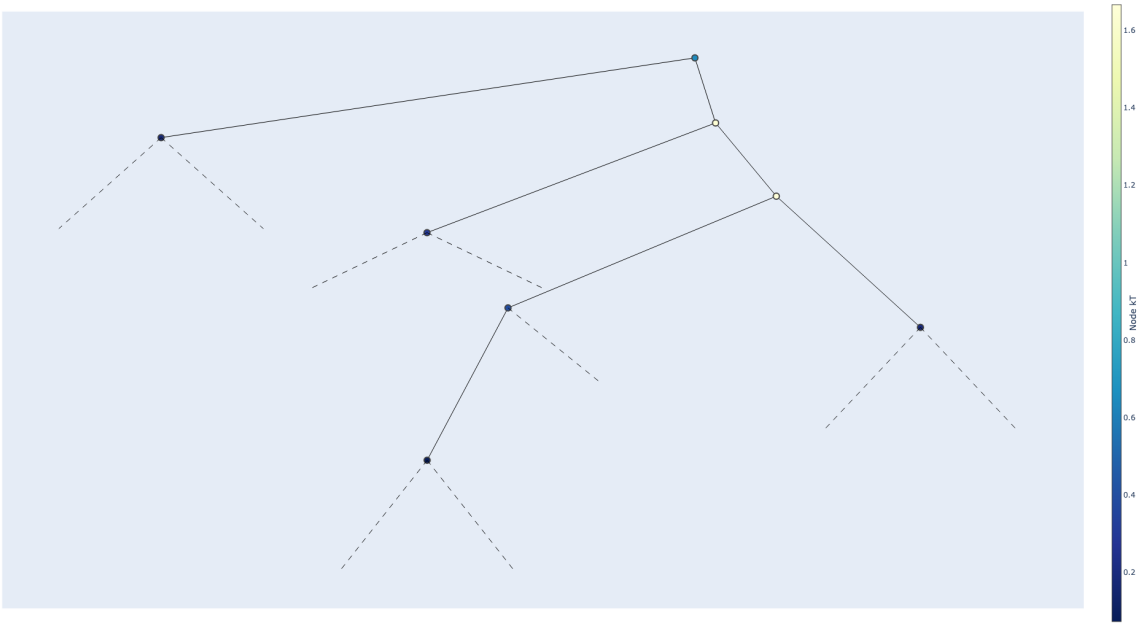


Figure 7.21: The visualized graph of a hadron level vacuum jet, clustered with C/A algorithm. The dotted lines are the final state particles, which are not considered to be part of the graph inside the GNN. The color scale represents the k_T of each node.

To gain a clearer understanding of the input provided to our graph neural network, let us visualize some of the graphs representing the Lund plane. As we have discussed previously, these graphs consist of nodes representing splittings in the jet's Lund plane.

It is important to note that the graphs we present to the network do not include the final state particles, as they do not undergo splittings and therefore lack nodes at the end of the line. Additionally, we have information about the formation time, denoted as t_f, and the opening angle associated with each splitting. To visualize the graph and the underlying physics effectively, we choose to plot the graph with the true opening angle at each splitting, while adjusting the edge lengths to be proportional to the formation time of the particles. Moreover, we aim to highlight the most energetic particles within the jets, so we assign a color to each node based on the k_T of the corresponding splitting. The color scale is determined by the color bar in the plots.

In Figure 7.21, we provide an example of such a graph visualization. This particular graph corresponds to a vacuum jet clustered using the C/A algorithm. The final state particles, which are not part of the actual graph used as input to the network, are represented by dotted lines in the visualization. we notice that the opening angle of each splitting increases as we move from the bottom to the top. Additionally, we observe that the most energetic subjets do not necessarily appear at the top of the graph. It is worth mentioning that if the jet were to undergo grooming procedures such as SoftDrop, the less energetic branch of the graph would be pruned or removed.

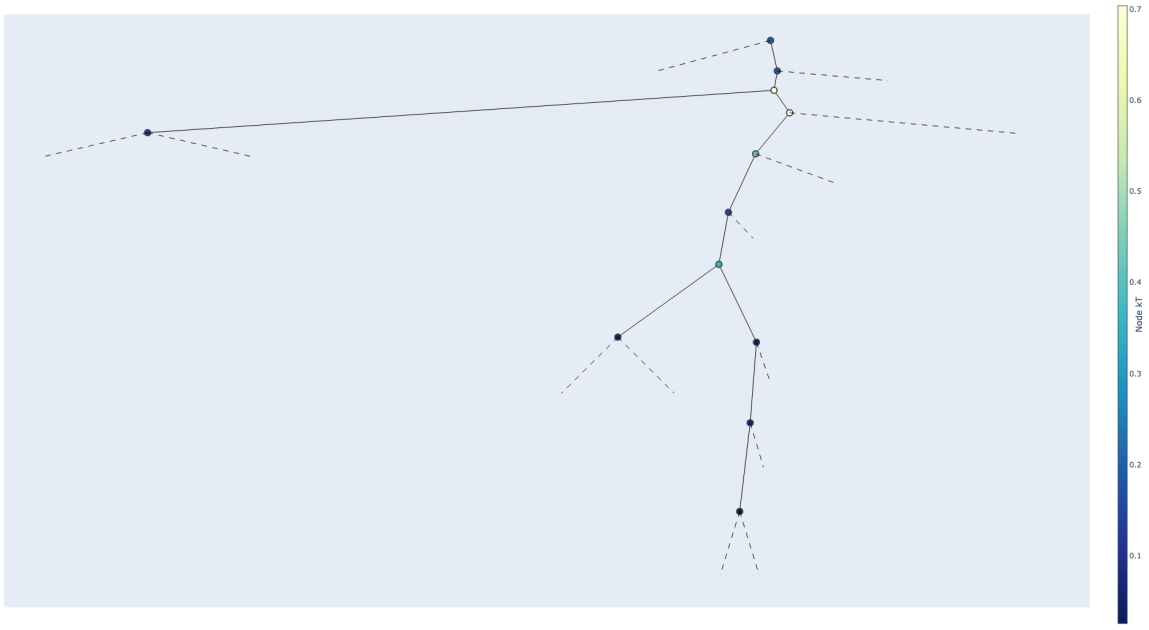


Figure 7.22: The visualized graph of the same hadron level jet after medium modifications, clustered with C/A-algorithm. The dotted lines are the final state particles, which are not considered to be part of the graph inside the GNN. The color scale represents the k_T of each node.

In the case of hadron-level jets, the graph representation for vacuum jets will typically contain fewer constituent particles compared to the medium-modified jets, which include additional particles from the wake. Due to this difference in the number of particles, one can expect the graph structure of vacuum jets to be vastly different from the medium-modified jets. In Figure 7.22, we present the corresponding medium-modified jet graph that matches the one shown in Figure 7.21. This particular jet exhibits an interesting feature with a final state particle appearing at the first splitting, a result of the plenty of soft splittings included in the jet. However, due to the soft- and collinear-safe

nature of the jet algorithms used, we anticipate that the hardest nodes should remain consistent. Although the addition of soft particles may alter the graph's overall structure, the essential hard features are expected to persist.

These graph representations heavily depend on the choice of clustering algorithm used to deconstruct the jet and construct the Lund tree. The most common clustering algorithm for these kinds of jet substructure tasks is C/A clustering, but another option could be k_T clustering. Depending on the jet clustering algorithm used, the graphs will look different. To explore this, we can generate the corresponding graphs using the k_T clustering algorithm.

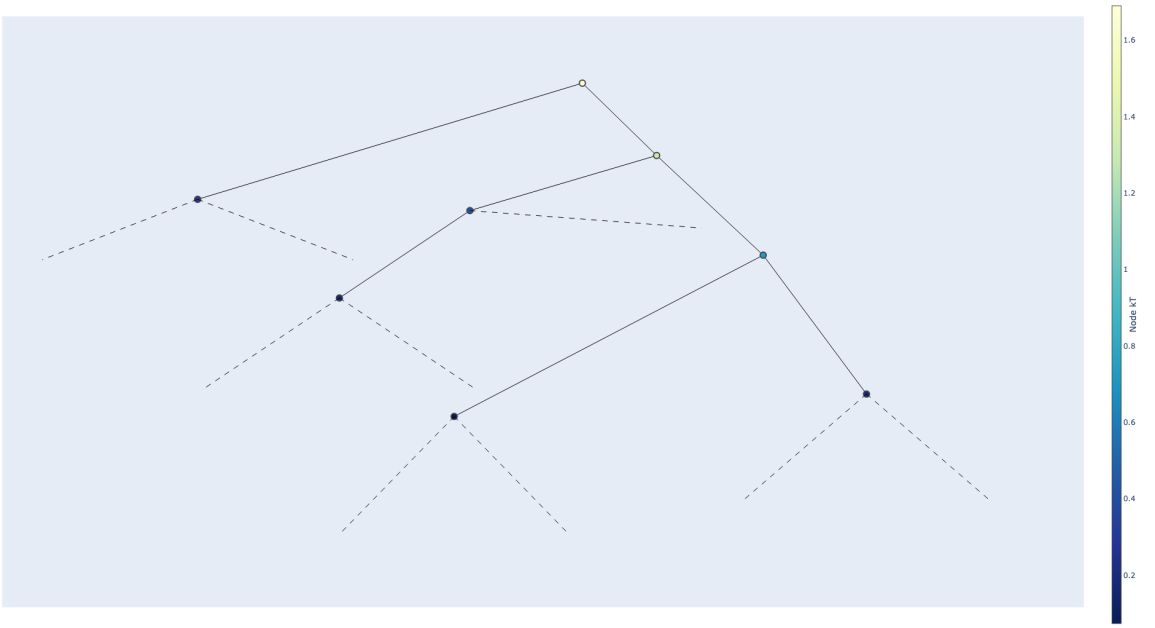


Figure 7.23: The visualized graph of a hadron level vacuum jet, clustered with k_T-algorithm. The dotted lines are the final state particles, which are not considered to be part of the graph inside the GNN. The color scale represents the k_T of each node.

First, we have the vacuum jet clustered with the k_T algorithm, shown in Figure 7.23. The k_T algorithm clusters jets after their k_T. This means that the most energetic particles are combined last. In this graph, we see that the most energetic nodes are at the top of the graph, with the color of the nodes showing the k_T value of the node. This k_T clustered jet is quite different from the one clustered with C/A. We can see the biggest difference in the opening angles and in the location of the hardest nodes. For the k_T clustered jets, the opening angles are quite similar throughout the whole tree, while for C/A clustered jets, the opening angle increases as we move up the tree. In k_T clustered jets, we see the most energetic splittings at the top of the tree, while for C/A clustered jets they appear somewhere else in the graph. This difference in the location of the hardest node is quite important when it comes to grooming techniques, as discussed in Chapter 3.3, C/A clustered jets where the hardest splittings happen later are better for grooming, as they leave room for softer branches early in the jet to be removed.

Figure 7.24 displays the medium-modified jets clustered using the k_T algorithm. Once again, the graph exhibits notable differences compared to both the vacuum jet clustered with the same algorithm and the medium-modified jet clustered with C/A.

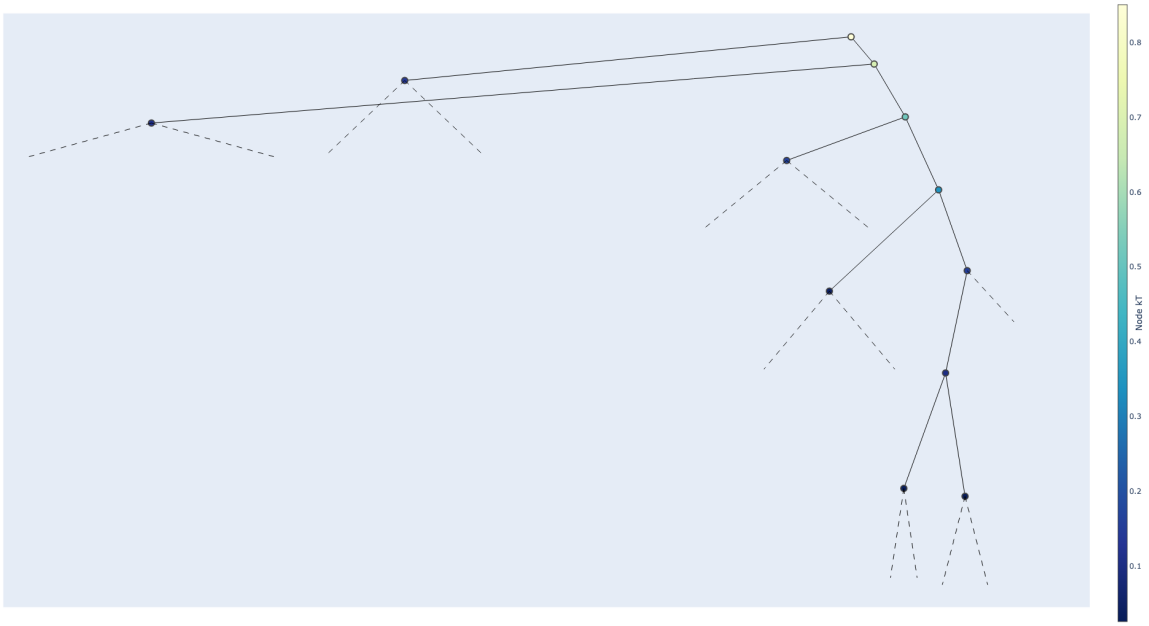


Figure 7.24: The visualized graph of the same hadron level jet after medium modifications, clustered with k_T-algorithm. The dotted lines are the final state particles, which are not considered to be part of the graph inside the GNN. The color scale represents the k_T of each node.

Despite these variations, the presence of the hardest splittings at the top of the graph remains consistent, indicating that the softest particles are combined first during the clustering process.

It is evident that the choice of clustering algorithm has a significant impact on the resulting graph structure. Further investigations can delve into exploring different clustering algorithms and their corresponding prediction performance. Alternatively, researchers could consider utilizing non-deterministic clustering algorithms to alleviate the network's dependence on a specific clustering approach. This approach would enable the network to be more robust and adaptable to different clustering scenarios.

7.4.2 Number of Features

Next, let us examine how the number of features assigned to each node impacts the network's performance. The LundNet model allows for flexibility in choosing the desired observables or features to provide the network. Typically, the model is trained with up to five features represented by the feature vector $\mathcal{T}$, as described in Equation 3.23.

The network can be trained in different modes depending on the desired number of features. The most commonly used mode is lundnet3, which includes three features $\mathcal{T} = \{\log(z), \log(\theta), \log(k_T)\}$. Other modes include lundnet4, which adds the mass m as an additional feature, and lundnet5, which includes both the mass and the azimuthal angle ψ around the subjet's axis. Alternatively, one can opt for the lundnet2 mode, which reduces the number of features to only $\{log(\theta), \log(k_T)\}$.

Throughout this thesis, lundnet3, the most popular configuration, has been used exclusively. This choice is a balance between providing the network with sufficient information and maintaining a reasonable performance level. The inclusion of additional

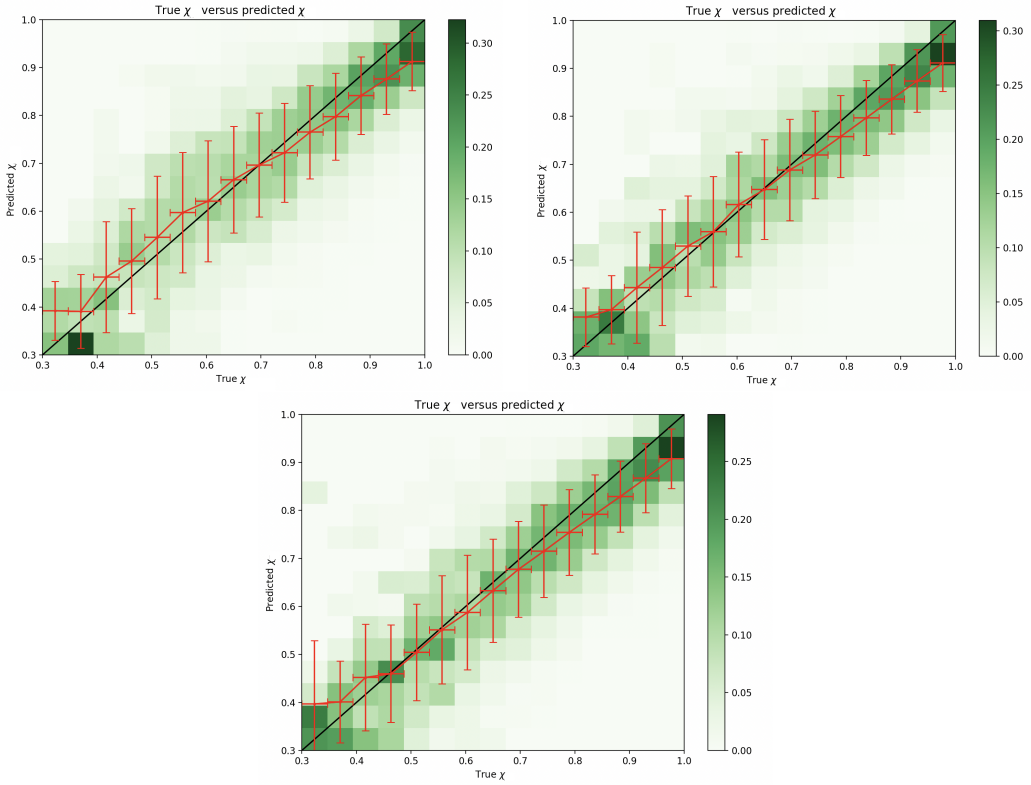


Figure 7.25: Prediction performance for the hadron level jets from the realistic jet spectrum with a different number of input features. *Upper Left:* Trained with lundnet2. *Upper Right:* Trained with lundnet4. *Lower Middle:* Trained with lundnet5.

Input	Average MSE Loss
lundnet2	0.00946
lundnet3	0.00777
lundnet4	0.00834
lundnet5	0.00940

Table 7.5: Table of average MSE loss for hadron-level jets from the jet spectrum dataset, with a different number of features. Including the lundnet3 result from Table 7.4.

features enhances the amount of information available to the network, but it comes at the cost of reduced performance and increased training time. On the other hand, reducing the number of features can expedite the training process, but it leads to sparser information input, resulting in a decline in performance. Thus, lundnet3 strikes a suitable middle ground.

These assumptions are motivated by training the network with the different modes and observing the results. Figure 7.25 showcases the performance of lundnet2, lundnet4 and lundnet5 models trained on the hadron-level jets from the realistic jet spectrum. We observe a broader distribution of predicted values for each true χ, indicating that the inclusion of additional features introduces more noise into the network. This effect becomes more pronounced when examining the average MSE loss presented in Table 7.5. The table also includes the average loss for lundnet2. As we can see, the loss increases with the addition of extra features and even when one feature is removed from lundnet3. Overall, while additional input features provide the network with more

information, they result in decreased performance and longer training times. Moreover, reducing the number of features leads to an increase in average loss roughly equivalent to the loss incurred by adding two additional features, proving to be the worst option so far.

7.4.3 k_T Cut

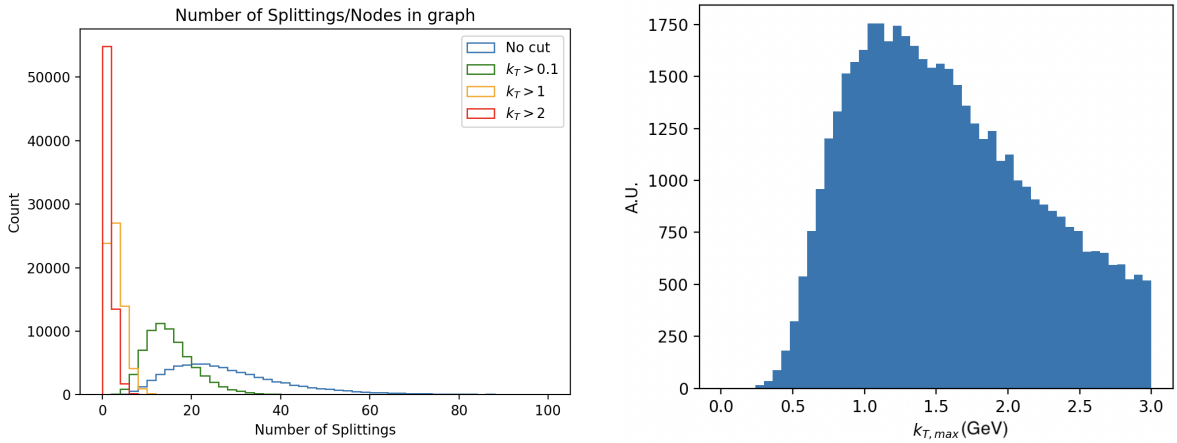


Figure 7.26: For hadron-level jets from the realistic jet spectrum. *Left:* Number of nodes or splitting in a jet, when applying a k_T cut. *Right:* $k_{T,max}$ distribution, which we will use to determine where to place a k_T cut

Having explored the network's response to the inclusion of additional information, we will now examine the consequences of removing information from the jets by implementing cuts on the k_T values of the nodes.

Since the jets are IRC-safe by definition, as discussed earlier, then the hardest splittings should be the most important ones while the soft particles can be safely removed. Therefore, one might expect that placing a k_T cut on the jets would eliminate the softer particles that contribute mainly to noise, but we observe a decrease in performance as information is removed.

It is important to note that no additional background radiation has been added to the jets at this stage. All particles inside the jet are either constituent particles that have lost energy to the medium or particles from the wake. All particles within the jet represent either constituent particles that have experienced energy loss in the medium or particles from the wake.

To determine where to place the cut, we examine the distribution of the maximum k_T for all the jets. We analyze the maximum k_T rather than just the k_T of the first splitting because the C/A algorithm does not always have the hardest splitting first. The right plot in Figure 7.26 displays the distribution of $k_{T,\text{max}}$, peaking at approximately 1.3 GeV. To remove as much soft radiation as possible without discarding the most important hard splittings, we choose to apply a cut just below the peak, at $k_T = 1.0$ GeV. While this represents the ideal cut, we also select two "extreme" cuts: a small cut at $k_T = 0.1$ GeV and a large cut at $k_T = 2.0$ GeV. The small cut should eliminate only the softest splittings, while the large cut risks removing significant information contained in the hard splittings.

To demonstrate the impact of each cut, we explore the distribution of the number of splittings in each jet, depicted in the left plot of Figure 7.26. Without any cut, the graphs typically contain a lot of nodes, or splittings. However, as we apply increasingly larger cuts, more soft particles are progressively removed, leading to a gradual decrease in the number of splittings within the jets. This results in narrower distributions with peaks shifting towards lower values, indicating a decrease in the number of splittings.

When employing the largest cut of $k_T > 2.0$ GeV, the distribution of the number of nodes becomes significantly sharper, centered around a very small number of splittings. This suggests that an excessive amount of information may have been lost in the process, raising concerns about the potential impact on the analysis. In particular, if our hypothesis regarding the small number of nodes being the cause for the poor performance of parton-level jets is accurate, this could also result in quite unsatisfactory outcomes.

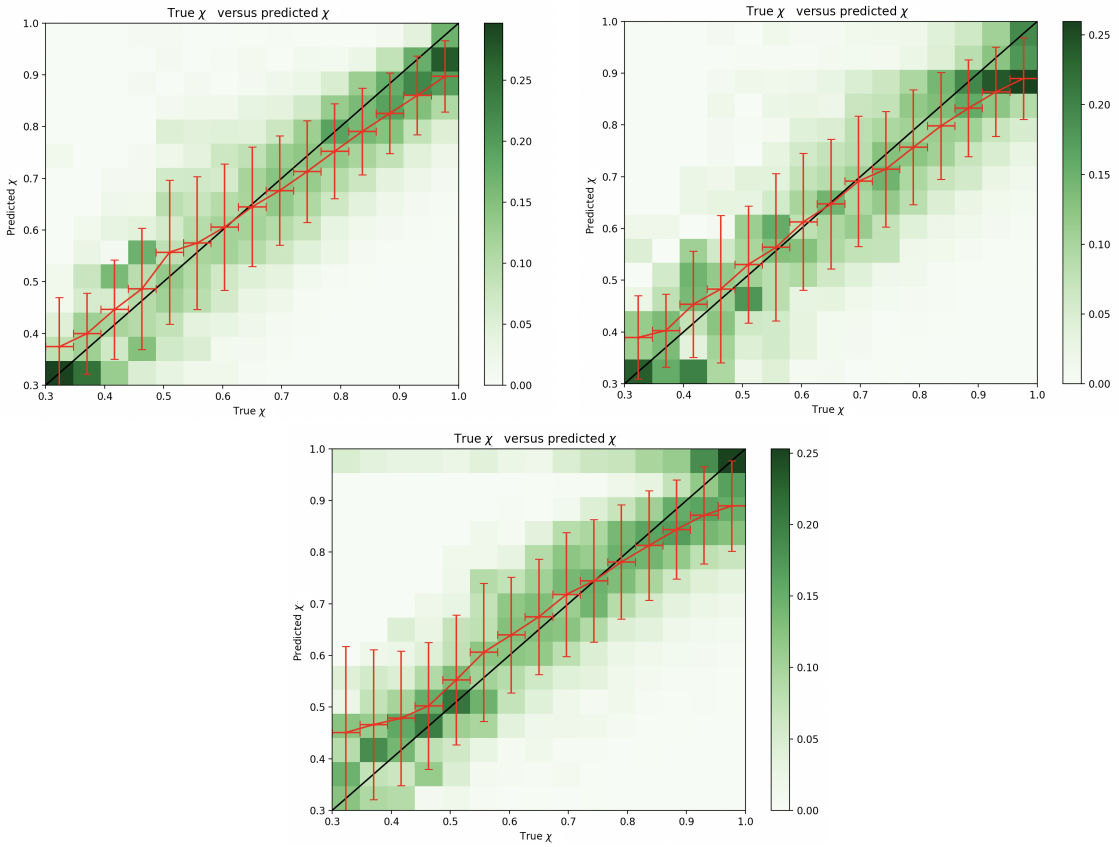


Figure 7.27: Prediction performance for the hadron level jets from the realistic jet spectrum after applying a k_T cut. *Upper Left:* With a cut at $k_T = 0.1$GeV. *Upper Right:* With a cut at $k_T = 1.0$GeV. *Lower Middle:* With a cut at $k_T = 2.0$GeV.

We proceed with training the network on hadron-level jets derived from the realistic spectrum while progressively applying larger k_T cuts. The resulting prediction performances are illustrated in Figure 7.27. Upon comparing these performances with the results from Figure 7.17, where no k_T cut was applied, we observe an overall deterioration in performance. The predicted values are more spread out, and the error bars are slightly larger. However, we still observe the same trends of lower predicted value for high true value and higher predicted value for low true value. Although the results

are not disastrous, there is a noticeable difference between the performances before and after applying k_T cuts.

Furthermore, as the k_T cut increases, we observe a further decline in prediction performance. This degradation is particularly pronounced in the lower middle plot of Figure 7.27, corresponding to the largest k_T cut. The application of this cut reduces the graph to only a few nodes, resembling the structure of parton-level graphs, and in turn, our results look similar to that of the parton-level graphs.

Input	Average MSE Loss
No cut	0.00777
$k_{T,min} = 0.1$GeV	0.01018
$k_{T,min} = 1.0$GeV	0.01188
$k_{T,min} = 2.0$GeV	0.01575

Table 7.6: Table of average MSE loss for hadron-level jets from the jet spectrum dataset, after applying different k_T cuts. Including the result without any cut from Table 7.4.

To effectively compare the performance of jets with different k_T cuts, we calculate the average MSE loss for each configuration, as presented in Table 7.6. As we introduce k_T cuts to the jets, the average loss consistently increases. However, applying a k_T cut of 2.0 results in a significant spike in the average loss, indicating that the model suffers when such a large amount of information is removed from the jets.

The model exhibits progressively worse performance as we remove information through k_T cuts, even more so than when we introduced additional features in the previous section. Despite the expectation that the hardest splittings are the most crucial and removing soft particles would have little impact on the network's performance, the importance of the soft radiation to the network is evident. The network prefers to add more information through additional features rather than remove soft particles.

7.5 Robustness Towards Embedded Background

Up until now, the jets we have examined were only modified by medium and contained the additional particles from the wake. However, in a realistic experimental setting, it is inevitable to obtain a medium-modified jet without the additional background radiation resulting from collision pileup and the hadronization of the QGP. These processes introduce additional soft, thermal background particles that obscure the signals of interest. Various grooming techniques are employed to mitigate the effect of background radiation on the jets, but completely removing all background while preserving the interesting signals is incredibly challenging. Therefore, machine learning methods such as LundNet must be somewhat resilient to background radiation for them to be applicable to experimental settings.

In this section, we will investigate the model's robustness to the additional background particles by embedding the medium-modified jets into a heavy-ion environment. The heavy-ion environment is simulated using the blast-wave model discussed in Chapter 2.3.2. We will begin with the original jets and gradually introduce more background particles until we reach realistic levels of background radiation. In heavy-ion collisions, an average of 1400 charged background particles per rapidity are produced

[24]. When considering neutral particles as well, the estimated additional background particles is approximately 2100 particles per rapidity.

The first step is to generate new datasets that incorporate the additional background particles. Both the jets and the background particles are clustered using the anti-k_T algorithm and then matched to their corresponding jets without background particles. Specifically, we match the medium-modified jets with a background to the medium-modified jets without a background, rather than comparing them to vacuum jets with background particles. This approach avoids the possibility of obtaining an energy loss ratio greater than 1. By comparing the medium-modified jets with background and vacuum jets with background, one could potentially include different amounts of background radiation in the two jets leading to a change in the energy loss ratio. Therefore, by using the same energy loss ratio calculated for the jets without background, we ensure that the energy loss ratio remains between 0 and 1.

Essentially, this section primarily examines how well the network handles the presence of additional soft particles, given the same labels as before. Given our understanding that the network tends to favor including additional information rather than removing any data, we expect a slow decrease in performance as additional particles are introduced. However, this decline should be slower compared to when applying jet substructure techniques to remove background radiation from the jets. To demonstrate this, we will also create a corresponding dataset where we apply constituent subtraction to the jets, allowing us to compare the performance of the network.

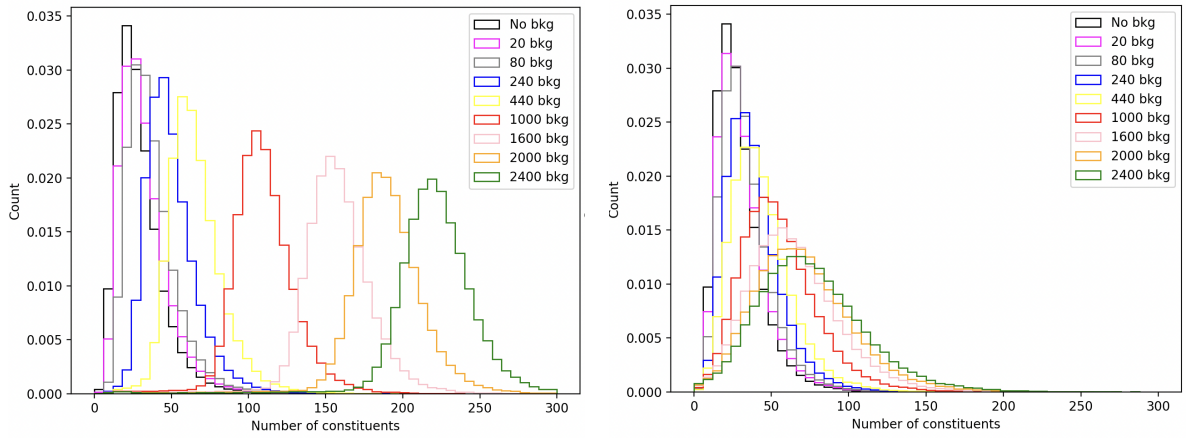


Figure 7.28: Distribution of the number of constituent particles for jets with different amounts of background particles. *Left:* Distribution without applying subtraction to the jets. *Right:* Distribution after applying constituent subtraction. Legend show the number of background (bkg) particles added is per rapidity.

We begin by examining the distribution of the number of constituents in jets with varying amounts of background radiation, as depicted in the left plot of Figure 7.28. Evidently, the effective number of constituents in the jets, after incorporating background particles, is significantly smaller than the number of particles added per rapidity. That is a result of our relatively small jet radius of $R = 0.4$, and an uneven distribution of background particles across the (η, ϕ) space.

Then we explore the impact of constituent subtraction on the number of constituents in the jets, as illustrated in the right plot in Figure 7.28. In this case, all distributions have shifted towards lower values of the number of constituents. The peaks of the

distribution are much closer to the peak of the distribution with no background particles. While we still observe a broader spread of the number of constituents for jets with higher levels of background radiation, the overall distribution has shifted significantly towards lower values. This indicates that a large number of soft particles have been removed through the constituent subtraction process.

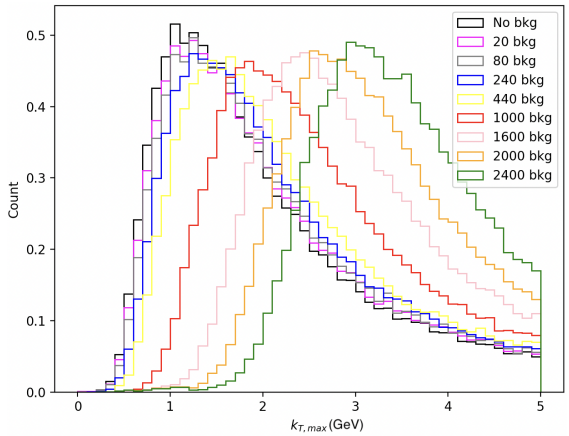


Figure 7.29: Distribution of $k_{T,max}$ for jets with background radiation and no background subtraction.

Based on our previous findings, we have found that the network does not respond well to the removal of information, and while constituent subtraction removes soft radiation without ruining the jet's internal substructure, it still results in information loss. Therefore, we wish to compare the sophisticated subtraction method to a more crude removal of information and explore whether the method of information removal matter or if the network simply struggles with any form of information removal.

To explore this, we apply a k_T cut to the un-subtracted jets with varying levels of background radiation, following the methodology employed in the previous section. To determine the optimal cut, we examine the distribution of $k_{T,max}$, as shown in Figure 7.29. The $k_{T,max}$ distribution differs significantly across different amounts of background particles. To ensure that the cut does not cause extreme energy loss for any of the distributions, we select a cut at $k_T = 1.0$GeV. It is worth noting that the k_T distributions demonstrate considerable resilience to additional background radiation until we reach 440 background particles.

Next, we train the network using these datasets and evaluate the average MSE loss, as presented in Table 7.7. In general, the performance decrease as the number of background particles increases. Moreover, applying constituent subtraction leads to a further decrease in performance, and the decline is even more pronounced when implementing a k_T cut at 1.0 GeV.

To better visualize this difference in performance, we plot the average loss for the various amounts of background particles per rapidity in Figure 7.30. Here, we observe that jets without subtraction or cuts consistently perform best, with the loss stabilizing at approximately 0.011. In contrast, for jets subjected to constituent subtraction, the loss continues to increase with the addition of more background particles. The worst results belong to the jets with a k_T cut instead of constituent subtraction, which tells us that applying a k_T cut is exceedingly more invasive than constituent subtraction, removing vital information from the jets.

dN_{bkg}/dy	No Subtraction	Constituent Subtraction	No Subtraction and $k_T > 1.0$GeV
0	0.00777	0.00945	0.01188
20	0.00977	0.01082	0.01197
80	0.01013	0.01037	0.01211
240	0.01126	0.01045	0.01162
440	0.01045	0.01111	0.01282
1000	0.01138	0.01252	0.01206
1600	0.01110	0.01206	0.01427
2000	0.01179	0.01267	0.01269
2400	0.01159	0.01283	0.01346

Table 7.7: Table of average MSE loss for jets with different amounts of background particles. The first column is without applying any background subtraction, the second column is after applying background subtraction and the third is without background subtraction but applying a k_T cut instead.

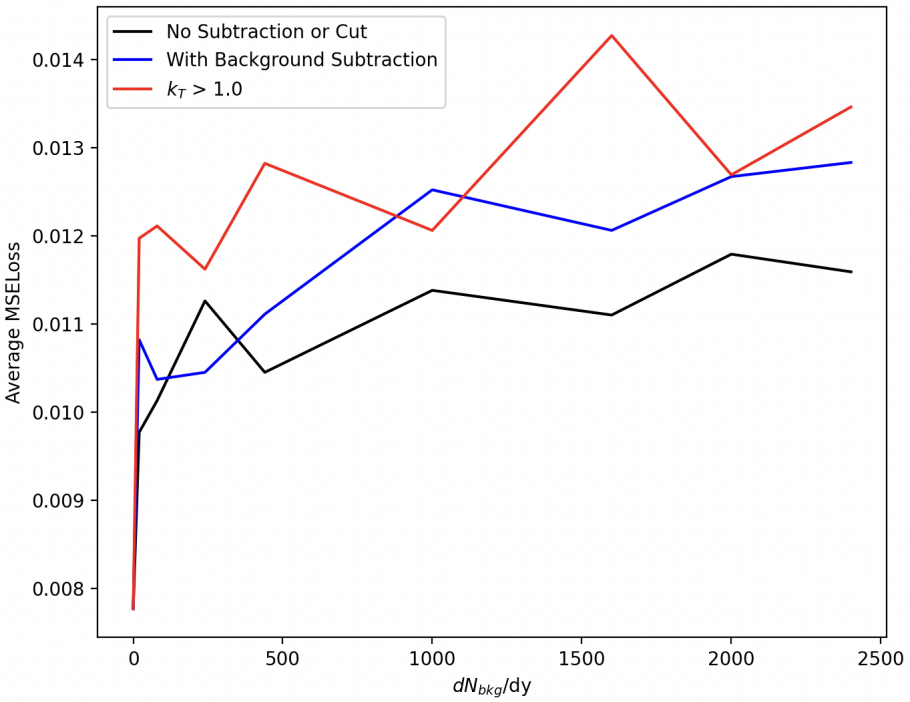


Figure 7.30: Plot of the average MSE loss for different numbers of additional background particles per rapidity. The distributions show jets with background particles but no background subtraction or cuts, jets with additional background and background subtraction, and jets with additional background and a k_T cut.

Overall, the LundNet model handled the embedding of jets in a heavy-ion environment quite well. The additional noise introduced by the background radiation is preferred over the removal of soft particles. We can conclude that the soft particles are equally as important to the network's ability to predict the energy loss ratio as the hard particles. The best performance is achieved when background particles are added to the modified jets without utilizing any subtraction techniques. Although the loss increases by approximately 30 percent, the average loss remains relatively stable as the background density increases.

7.6 Future Considerations

After finishing our results, we wish to provide an insightful outlook on the future prospects and potential topics of discussion to further explore the use of GNNs to predict the energy loss of jets traveling through a QGP.

Among the things we would like to further explore is the odd behavior of the network when dealing with the parton-level jets from the realistic spectrum. As of now, we can only assume this happens due to the limited information in the sparse graphs, but with further investigation, one may discover approaches to mitigate this effect.

We also discussed how the graphs of the jets look very different for vacuum and medium-modified jets. This discussion led us to consider the importance of clustering algorithms and their impact on the graph representation of jets. Since the graphs of the corresponding vacuum and medium-modified jets appear so different, we question whether the choice of jet clustering algorithm is as important as initially suggested. To explore this further, we propose the utilization of non-deterministic jet algorithms, such as Q-jets [66], to free the network from the deterministic structure of clustered jets. The concept of Q-jets is shortly introduced in Appendix C, and considers multiple graphs weighted by some metric. By training the network on jets with varying graph structures, we may uncover the most optimal structure for accurate predictions.

Another approach to free the jets from the constraints of clustering algorithms would be to use ParticleNet. ParticleNet was briefly described in Chapter 4.5.1, and treats the jets as particle clouds, using EdgeConv operations on the k nearest neighbors of each particle, thereby aviding the need for clustering algorithms. This alternative method may allow us access to even more valuable information from the jets. Ref. [6] reveal that ParticleNet performs similarly to LundNet in jet tagging studies, suggesting its potential efficacy in our analysis as well.

Furthermore, we can explore the network's response to grooming techniques such as SoftDrop, or Dynamical Grooming [67]. As discussed in Chapter 6, SoftDrop grooming effectively removes soft radiation to recover the desired signal. While we are aware that the network suffers when information is removed from jets, we wonder if the application of SoftDrop may stabilize performance, particularly in scenarios with a higher density of background radiation.

While this thesis primarily focuses on one specific model for jet quenching, we are intrigued by the possibility of the network's dependence on the model used to generate the medium-modified jets. We hope to capture universal features rather than model-specific features. Though we have not discussed other jet quenching models, we know that hybrid models, such as the hybrid strong/weak model we employ in this thesis, commonly contain both hard and soft components, but specific details about the implementation of the different models could vary. Therefore, future work might benefit from attempting to identify any indications or patterns that might shed light on a potential relationship between the learning process and the model of jet quenching

Conclusion to the Main Study of the Thesis

We observed that our model achieved performance comparable to, or slightly worse than, the previous work conducted in Ref. [5]. This suggests that our ap-

proach is consistent with existing methods.

This thesis focused on studying the **sensitivity of the model** to various cuts and the influence of additional features and background radiation from the **embedded heavy-ion environment**. We found that the network is "greedy" for additional information, and **prefers additional noise** rather than having information removed.

We conducted a comparison between **constituent subtraction** and **applying a k_T cut**. Our analysis revealed that the removal of information through k_T cuts s a highly invasive procedure, leading to a greater decrease in performance compared to constituent subtraction. This finding underscores the importance of considering how information is removed, as constituent subtraction aims to preserve the internal structure of the jet, while a k_T cut blindly discards information below a certain threshold.

Surprisingly, our investigation of the most realistic scenario involving hadron-level jets originating from the realistic spectrum, with medium modifications embedded in a heavy-ion environment, demonstrated **decent performance**. Despite a slight decrease in performance with increasing background radiation, we observed a stable performance level even in the presence of significant background radiation.

Conclusion and Outlook

This thesis focuses on the application of machine learning techniques to predict the energy loss of jets traversing a **quark-gluon plasma**. Our approach involves representing the jets as graphs derived from the declustered Lund tree and training a graph neural network model, known as LundNet, to perform the energy loss predictions. Overall, the network has exhibited promising performance in its predictions.

Properties of the QGP can be studied through the effect of medium modifications on jet observables. However, jets that are suffering great energy losses are typically underrepresented in statistics as they often are lost due to the **selection bias effect**. Therefore, it is desirable to be able to recognize how much energy a jet has lost in order to uncover part of this information.

Previous work has successfully utilized convolutional neural networks to predict energy loss based on jet images. However, these approaches faced limitations when it came to embedding jets into a heavy-ion environment. In experimental settings, measurable jets are influenced by the thermal background generated by other detector processes and the thermal background originating from the quark-gluon plasma itself. Consequently, there is a need to extend energy loss prediction to jets embedded in heavy-ion environments, accounting for these additional factors.

To improve upon this previous work, we opted to use a new approach by utilizing a different representation of jets, moving away from jet images. While jet images effectively capture radiation patterns, they lack the ability to incorporate comprehensive information about the jets. To address this limitation, we chose to work with graph data structures, which offer greater flexibility. Specifically, we employed a **graph representation of the Lund tree**, which is great at encoding both the radiation patterns and substructure of jets. By using graphs we can essentially include a wide range of observables at any point within the graph, thereby allowing us to provide more information into our machine-learning model. However, using a graph representation of jets introduces a dependency on the **clustering algorithm** used to construct the jets. As no clustering algorithm perfectly aligns with the true branching history, there is a possibility that the machine learning model becomes reliant on the artificial structure generated by the chosen clustering algorithm.

Considering our goal to incorporate the embedding of jets into a heavy-ion environment, we conducted a preliminary study to examine how jet observables are influenced when a **vacuum jet is embedded within a heavy-ion environment**. We found that there was quite a difference between the distribution of observables with and without embedded background, especially for the mass, the opening angle, and the momentum-sharing fraction. To mitigate these disparities, we applied SoftDrop grooming to the jets. This grooming technique significantly improved the resultant distributions of ob-

servables, making the distributions more similar between jets with and without embedded background radiation. In regard to this study, we also discussed the effect of different clustering algorithms when performing SoftDrop grooming. We found that the use of Cambridge-Aachen clustering in SoftDrop grooming is advantageous as a decent amount of soft radiation is removed.

Subsequently, we moved on to the main study of this thesis, which involved training our model to predict jet energy loss. We used two different datasets, one created by a parton gun and one by a realistic dijet spectrum. First, we focused on the **parton gun** dataset and trained the network on both parton-level and hadron-level jets. Notably, we observed that the network exhibited superior performance when trained on hadron-level jets. This can be explained by the inclusion of particles from the wake in hadronic jets, resulting in larger graphs replete with informative content. In contrast, partonic jets contained only a sparse number of particles, limiting the information available to the network.

Next, we trained the network using the jets derived from the **realistic dijet spectrum**. Surprisingly, the network's performance on parton-level jets from the realistic dijet spectrum proved to be unsatisfactory, performing considerably worse than the partonic jets from the parton gun. We expect the poor performance to be due to the sparse graphs belonging to the parton-level jets, containing only a few nodes. The hadronic jets from the realistic spectrum are the most realistic jets for experimental measurements that we have considered so far. The network performs very well on this data, as visualized in Figure 7.17. The network still predicts a slightly too-high energy loss for the jets with very low energy loss and too low energy loss for the jets with very high energy loss.

In order to gain insight into the network's decision-making process and identify what it considers the most important features, we conducted several brief studies. First, we **visualized the graphs** provided to the network and observed notable differences between the vacuum jet graph and its corresponding medium-modified jet. This is to be expected since the medium-modified jets contain a larger number of particles. Furthermore, we recognized that the choice of clustering algorithm significantly impacts the graph structure. However, given the large difference between the vacuum jet and medium-modified jet graphs, we concluded that the clustering algorithm's influence may not be as crucial as initially believed.

Next, we tried to change **the number of features** assigned to each node. Introducing additional features resulted in a performance decrease, although at a slower rate compared to when removing a feature. Then we explored the effect of applying a k_T **cut** on the graphs to remove soft radiation. Due to the IRC-safe jet definition, we expect that the hard splittings are the most important to accurately predict the energy loss. Surprisingly, removing the soft splittings led to a significant decline in performance, indicating that the network considers soft radiation crucial for accurate predictions.

Finally, we **embedded the medium-modified jets into a heavy-ion environment** and trained the network using samples with varying quantities of background particles. As the number of background particles increased, the network's performance gradually decline. Furthermore, applying background subtraction to remove some of the soft background particles resulted in a more rapid decline in performance. Overall, the network displayed a preference for receiving additional information rather than removing information through k_T cuts and background subtraction tools.

To improve upon our results, we would like to eliminate the dependence on clustering algorithms by introducing the Q-jet algorithm, which is briefly outlined in Appendix C. Although the reliance on clustering algorithms may not be as significant as we feared, we wish to free ourselves from such dependencies. Additionally, grooming methods like SoftDrop could be employed in future work to close the performance gap between jets with embedded background radiation, with and without background subtraction. Finally, we would also like to investigate whether the machine learning models are dependent on the jet quenching model.

These results are promising, and with further improvements, the network will be capable of sufficiently predicting energy loss for realistic experimental measurements. The precise estimation of energy loss on a jet-by-jet basis can significantly contribute to our understanding of the quark-gluon plasma by amplifying the signals from heavily modified jets. This, in turn, opens up new avenues to explore new aspects of the QGP and gain deeper insights into its properties.

Acknowledgements

First and foremost, I want to express my heartfelt appreciation to my supervisor, Konrad Tywoniuk. His invaluable guidance and mentorship throughout my master's thesis have been instrumental in my academic journey. Additionally, I am immensely grateful for his continuous support in applying for summer internships and now for Ph.D. positions. Working with him has been a pleasure, and I am truly thankful for his assistance.

I would also like to extend my gratitude to Daniel Pablos for his contribution in generating the hybrid model data, as well as to Yilun Du for engaging discussions on machine learning methods.

I am grateful to the theory group for their stimulating seminars and insightful discussions over the past year. In particular, I would like to thank the PhD students Adam Takacs, Johannes Hamre Isaksen, and Alexandre Da Fonseca Falcao for their valuable discussions and feedback on the project, and engaging conversations during lunch breaks.

I would also like to express my gratitude to the friends I have made through my studies. A special thanks to Esther, whose lunchtime walks and Mario Kart sessions created a wonderful study environment, and to Tiril and Sebastian, who kept me active with our weekly bouldering sessions.

Lastly, I want to extend a special thank you to my family for their unwavering support throughout my academic journey. Whether it was hiking to help me relax, baking homemade study snacks, or my professor in genetics father assisting me with scientific writing. A special thank you to my grandmother, whose door is always open to provide me with the extra encouragement I need. Their love and support have been invaluable to my academic success.

Appendix A

Cautionary Tale of Applying Cuts on Jet Observables

In this thesis, we have discussed the inherent bias introduced when removing jets or their constituents based on a threshold set for a particular observable. One of the most prominent examples is the selection bias resulting from the transverse momentum (p_T) cut applied to jets, which serves as the primary motivation for conducting this thesis. It is important to recognize that if we just cut away jets, or constituents, naively then we end up with a very biased set of jets that no longer represent the true jet distribution.

If we return to Chapter 6, we see that after applying SoftDrop grooming the distribution of observables has become more cohesive, but there is still room for improvement if we wish to mitigate the effect of background completely. To delve deeper into this matter, we will examine the impact of imposing a cut on a specific jet observable for the vacuum jets embedded in a heavy-ion environment, the same dataset as discussed in Chapter 6.

To decide what observable we wish to apply a cut on, we take another look at the distribution of observables in Figure 6.6 and 6.7. We want to apply a cut to an observable which still has a considerable difference between the jets with and without background radiation. We are looking for a border where we can separate the majority of the distribution without background on one side of the cut, and as much as possible of the distribution with background on the other side.

Possible candidates can be removing the tail of the mass distribution or removing the tail of the k_T distribution. Another possibility is to place a cut on the distribution of the number of constituents or the ratio plot, which shows the distribution of constituents in the first two subjets. There is a much clearer border between jets with and without background in these plots, however, it is quite risky to put too much faith in these observables as they are not IRC-safe observables. As the differences for the mass and k_T distributions are so small, we will try to apply cuts to the observables related to the number of constituents in the jets and in each of the first subjets.

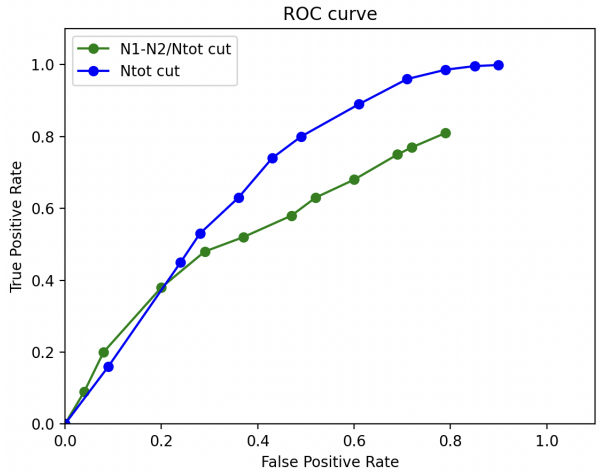
To determine the optimal cut, we calculate a ROC curve, which stands for Receiver Operating Characteristic curve. The ROC curve provides a graphical representation of the performance of a classification model across various classification thresholds [68]. In our specific case, it demonstrates the performance of each cut option considered. The effectiveness of the cut can be evaluated based on how well it retains jets without background while removing as many jets with background as possible. By achieving

this outcome, we can approximate the distribution of jets with background to closely resemble the distribution of jets without background.

This curve is constructed by plotting two parameters: the True Positive Rate (TPR) and the False Positive Rate (FPR), given by:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}. \tag{A.1}$$

In our plot, the True Positive (TP) corresponds to the number of surviving samples for jets without background when the cut is applied, while the False Negative (FN) denotes the number of samples that did not meet the criteria. Furthermore, we determine the False Positive (FP) as the number of surviving samples for the distribution with background particles, and the True Negative (TN) as the number of samples that were removed by the cut.



	cut_1	cut_2	cut_3	cut_4	cut_5	cut_6	cut_7	cut_8	cut_9	cut_{10}	cut_{11}	cut_{12}
N_{tot} cut	0	5	10	11	13	15	17	20	25	30	35	40
$\frac{N_1-N_2}{N_{tot}}$ cut	-1	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2

Figure A.1: The ROC curve for $\frac{N_1-N_2}{N_{tot}}$ cut and N_{tot} cut. The points on the graph represent the cuts in the table. The table does not contain the TPR or FPR of the distributions, only the value of the cuts.

Next, we implement multiple cuts on our SoftDrop jets and evaluate their performance using the ROC curve depicted in Figure A.1. To achieve optimal separation between the signal and background, we aim for a high True Positive Rate (TPR) and a low False Positive Rate (FPR). Based on the curve, the most favorable point is a cut that eliminates all jets with $N_{\text{tot}} > 17$.

Upon implementing this cut, the distributions of observables exhibit larger differences between jets with and without background radiation. It appears that the applied cut has removed crucial information from the jets, leading to noticeable differences in the observables. Thus, we can call this a cautionary tale of what not to do if we wish to keep the most crucial information of a jet.

In the upper left plot of Figure A.2, the impact of the cut is clearly visible. The distributions of N_{tot} with background exhibit a sharp cutoff at 17 particles per jet.

However, when examining the ratio $\frac{N_1-N_2}{N_{\text{tot}}}$ in the upper right plot, we observe a striking similarity to the same plot where only SoftDrop had been applied. The primary difference is that the jets with background now closely resemble the distribution

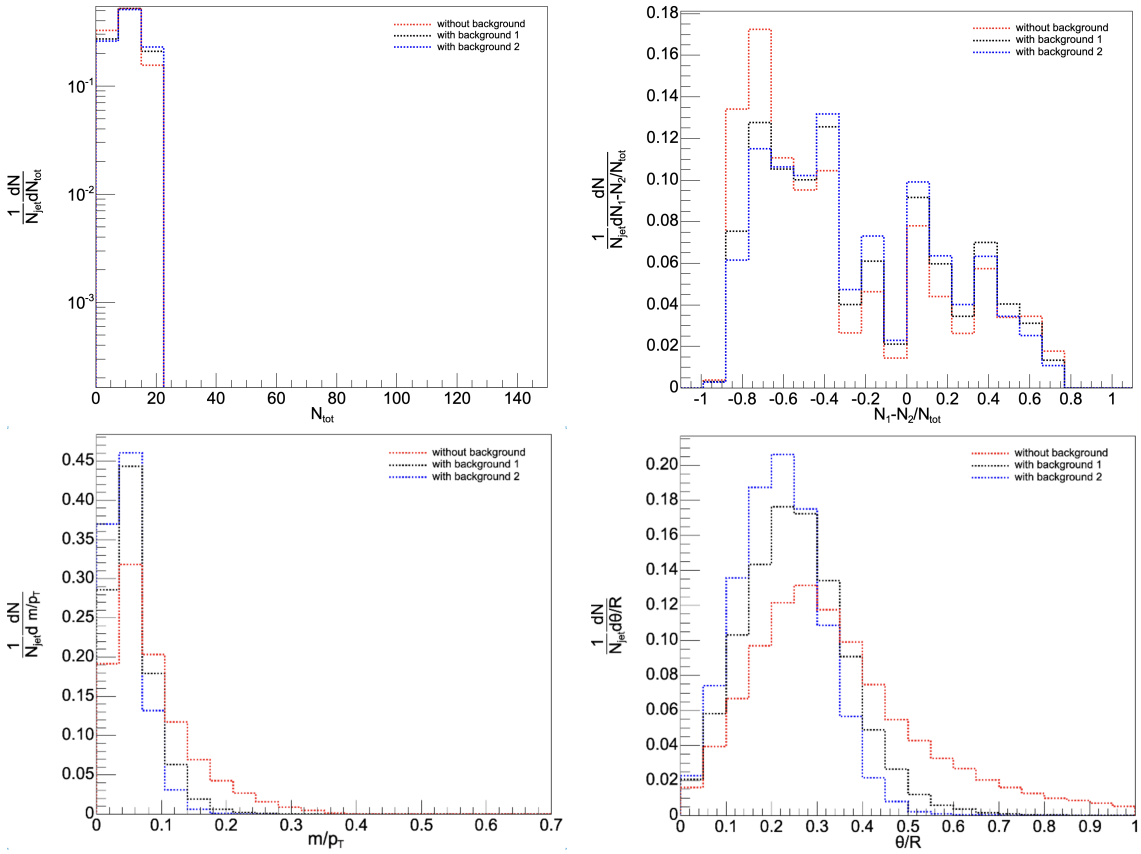


Figure A.2: Plots of observable distribution for $\hat{p}_{t\,min} = 200$GeV when applying SoftDrop grooming and cut $N_{tot} > 17$. *Upper Left*: N_tot distribution. *Upper Right*: $N1 - N2/N_tot$ distribution. *Lower Left*: Mass distribution. *Lower Right*: θ distribution.

without background, aligning more with its shape compared to when only SoftDrop was employed.

Moving to the lower left plot in Figure A.2, we analyze the mass distribution post-cut application. All distributions appear narrower than before the cut was implemented. Notably, jets with background radiation exhibit a significantly narrower distribution compared to jets without background, which display a slightly broader distribution.

Lastly, in the lower right plot of Figure A.2, we examine the distribution of the opening angle θ after applying the cut. The distributions appear less concentrated, featuring wider spreads and a peak at lower values.

The Key Concepts of This Appendix

By applying cuts on the observables, we essentially discard jets in a crude manner, resulting in observable distributions that no longer accurately represent the true underlying jet distributions.

When a cut is imposed on the number of constituents within a jet, the observable distributions experience a shift, altering their characteristics. As a consequence, the difference between jets with and without background radiation increase compared to when only SoftDrop grooming is employed.

Appendix B

Correlations between χ and Observables

In order to gain a deeper understanding of the graphs used as input for the network, we aim to conduct a brief study on the correlations between the energy loss ratio and various observables. This study serves as an indirect investigation into the impact of cuts on the graphs, as we examine the behavior of different observables after undergoing various degrees of modification caused by the medium.

Specifically, we will focus on the characteristics of the first node within each graph, dividing them into χ bins and analyzing the ratio relative to the corresponding vacuum feature. This analysis will provide valuable insights into the relationship between the energy loss ratio and observables within the graphs.

Figure B.1 provides insight into the distribution of features associated with the first nodes, as well as their ratio to the vacuum k_T values. The full medium-modified dataset has been subdivided into smaller datasets based on the extent of modification experienced by the jets, or how large the energy loss is. Through this analysis, our objective is to identify any observables that exhibit notable sensitivity to specific levels of modification.

First, let's consider the k_T distribution in the upper right plot. There is a noticeable difference between medium-modified jets and those with extensive modifications compared to the vacuum jets, particularly for high k_T values. The distributions are shifted towards lower values. The ratio plot shows that jets with minimal energy loss have a higher proportion of high k_T values.

Moving on to the opening angle θ in the upper middle plot, which reveals a shift towards larger opening angles for medium-modified jets. The degree of shift increases with the level of modification, with a higher concentration of jets with significant modifications in the tail of the distribution at high θ values.

Examining the distribution of the momentum-sharing fraction z in the upper left plot, we observe a shift towards lower z values for all modified jets. Interestingly, jets with minimal modifications exhibit the most distinct distribution for z compared to vacuum jets.

The lower left plot shows the mass distribution, which is shifted towards smaller values for medium-modified jets. The average mass decreases as the level of modifications increases

As expected, the distribution of the number of constituents in the lower right plot increases with higher levels of modifications. Vacuum jets have relatively few constituents, while jets with extensive modifications contain a larger number of soft wake

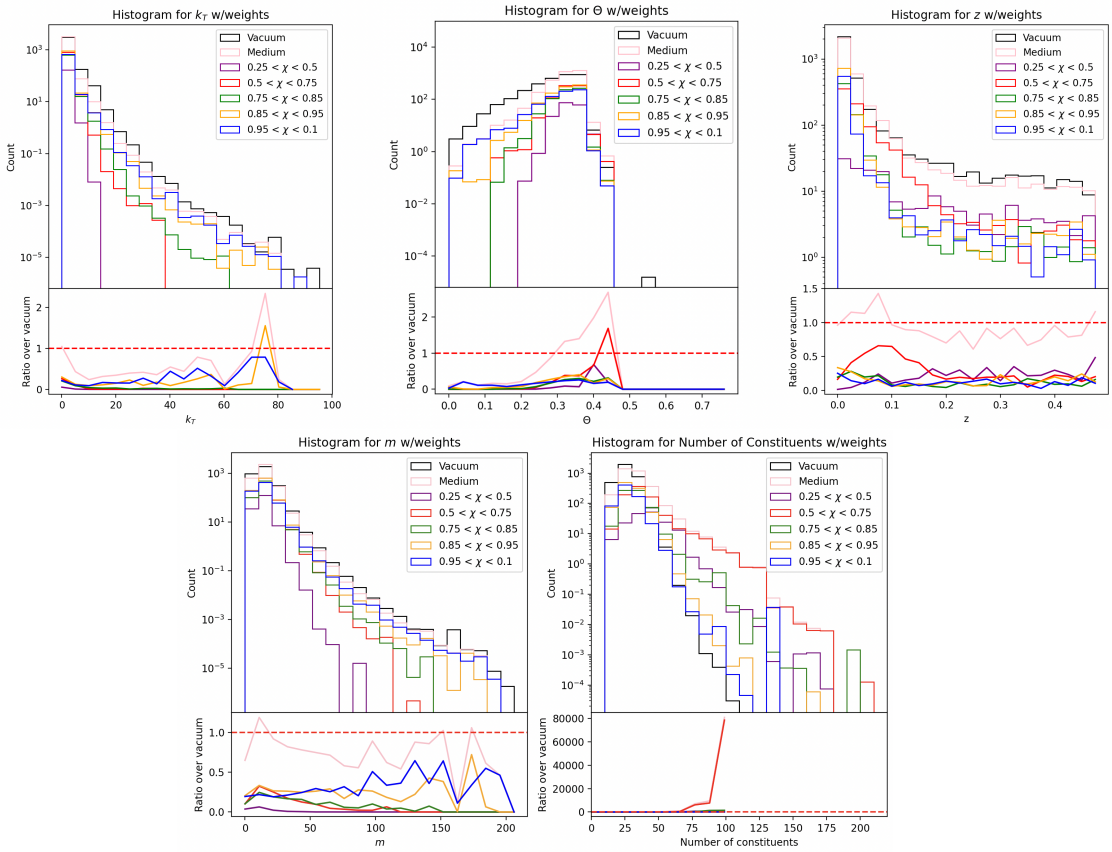


Figure B.1: Distribution of observables for hadron-level jets from the jet spectrum for different χ bins. *Upper Left:* The k_T distribution. *Upper Middle:* The distribution of the opening angle θ. *Upper Right:* The distribution of the momentum-sharing fraction z. *Lower Left:* The distribution of the mass m. *Lower Right:* The distribution of the number of jet constituents.

particles.

In summary, we find that the observables are primarily influenced by jets with the most significant modifications, except for the momentum-sharing fraction z, where jets with minimal modifications display the most distinct distribution compared to the vacuum counterpart.

The features of the first node may not necessarily be the most reliable jet observables due to the changes in jet structure caused by additional soft particles. To mitigate this, we employ SoftDrop grooming to remove some of the soft branches and focus on the distributions of jet features corresponding to the modified observables after applying SoftDrop. These SoftDrop observables provide a more informative characterization of the system compared to the features of the first nodes, which can be heavily influenced by the choice of jet algorithm. The distributions of these observables after applying SoftDrop are presented in Figure B.2.

Starting with the k_T distribution in the upper left plot, we observe that the distribution after applying SoftDrop remains largely unchanged compared to before. The distributions follow similar trends, and no significant differences are noticeable.

However, in the upper middle plot depicting the distribution of the opening angle θ after applying SoftDrop, a distinct change is apparent. The distributions no longer exhibit a peak at high θ values but instead display a flatter distribution across the entire

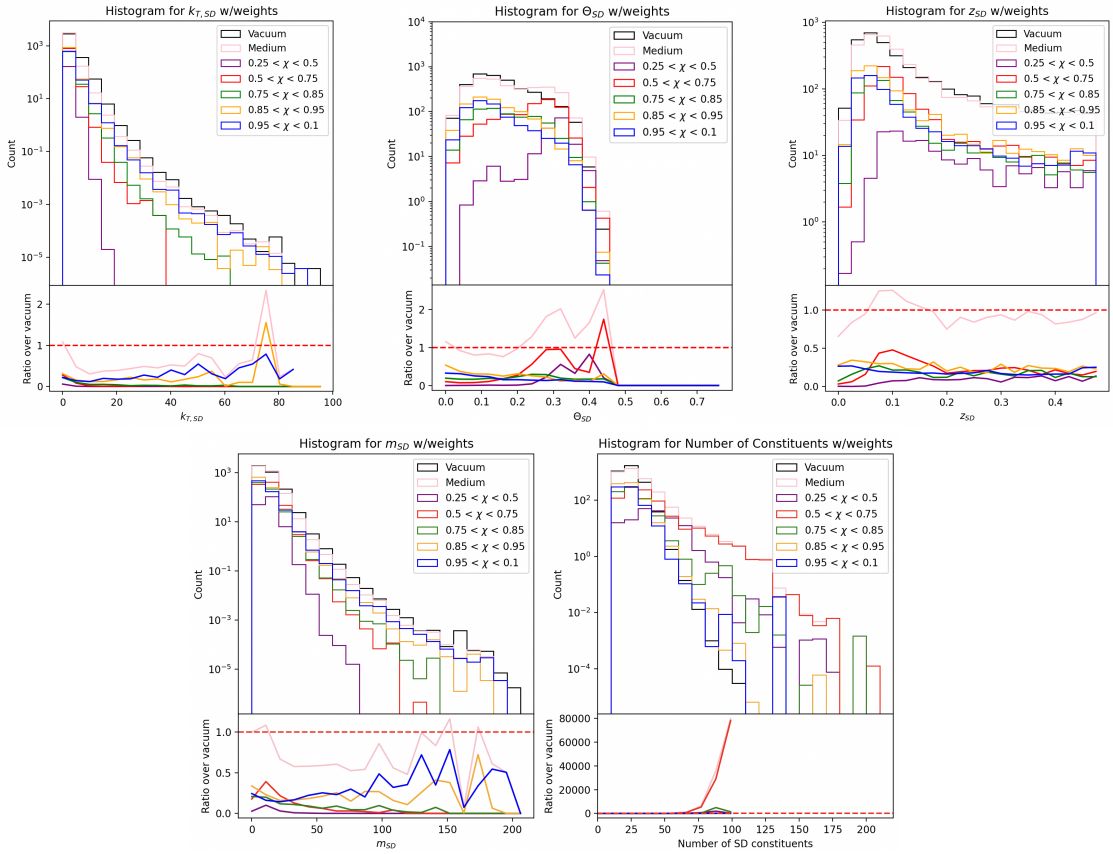


Figure B.2: Distribution of jet observables for hadron-level from the jet spectrum dataset for different χ bins, after applying SoftDrop. *Upper Left:* The k_T distribution. *Upper Middle:* The distribution of the opening angle θ. *Upper Right:* The distribution of the momentum-sharing fraction z. *Lower Left:* The distribution of the mass m. *Lower Right:* The distribution of the number of jet constituents.

range. Jets with significant modifications, characterized by small χ values, are shifted closer to their pre-SoftDrop distributions and exhibit the most noticeable deviations from the vacuum distribution.

Moving to the upper right plot, we examine the momentum-sharing fraction z distributions after applying SoftDrop. These distributions are quite similar to the pre-SoftDrop distributions, with the exception that the jets with the most modifications now display the least resemblance to the vacuum distribution, showing a shift towards larger z values.

In the lower left plot, we find the mass distributions after applying SoftDrop, which closely resemble the distributions obtained without SoftDrop. The distributions exhibit similar trends and characteristics as observed previously.

Finally, the distribution of the number of constituents in the lower right plot remains highly consistent with the previous distributions. The distributions of medium-modified jets continue to exhibit the most significant differences from the vacuum jets, as they still contain a substantial number of soft particles that have not been entirely removed by SoftDrop grooming.

Overall, applying SoftDrop did not have large effects on the observable distributions in the sense that the difference between the vacuum jets and the medium-modified jets still have quite different observable distributions after SoftDrop.

The Key Concepts of This Appendix

Our analysis reveals that the observables are primarily influenced by jets experiencing the most substantial modifications. Specifically, jets with significant energy loss, indicating large modifications, exhibit observable distributions that are least similar to those of vacuum jets. In these cases, the distributions tend to be shifted towards lower values for observables such as k_T, mass, and momentum-sharing fraction z.

Furthermore, we find that the application of SoftDrop grooming to the jets has minimal impact on the distributions of observables. The observed trends in the distributions remain consistent with those obtained for jets subjected to medium-modifications, indicating that SoftDrop does not introduce significant alterations to the observable behavior.

Appendix C

Q-jets

The clustering of particles into jets is typically done using deterministic clustering algorithms, such as Cambridge-Aachen or anti-k_T. These jets take a graph-like structure when clustered. These clustering algorithms create one deterministic graph for each jet based on some distance measure between each particle.

Instead of considering just one graph per jet, one could use a non-deterministic approach called Q-jets [66]. This approach considers multiple graphs for each jet, weighted by an appropriate metric. Then each jet in an event produces a distribution for each observable, rather than one single value [66].

This could be useful to free the graph neural network from the constraint of the choice of a clustering algorithm. If we can supply the network with different-looking graphs for each jet, then we could potentially make the network more robust and not as dependent on the clustering algorithm.

C.1 Definition of Q-jet

The algorithm assembles a graph using a series of $2 \to 1$ mergings. It works as follows

1. At every stage of the clustering, a set of weights ω_{ij} for all pairs $\langle ij \rangle$ are computed, using the formula

$$\omega_{ij}^{(\alpha)} = \exp\Big\{-\alpha \frac{(d_{ij} - d^{min})}{d^{min}}\Big\} \tag{C.1}$$

   where the distance measure is $d_{ij} = \min\{p_{T,i}^{2p}, p_{T,j}^{2p}\}\frac{\triangle R_{ij}^2}{R^2}$ in Eq. 3.16, with $p = 1$ for k_t distance measure or $p = 0$ for C/A distance measure, and α is the rigidity parameter.

2. Then compute the probability for each pair

$$\Omega_{ij} = \omega_{ij}/N \tag{C.2}$$

   where $N = \sum_{\langle ij \rangle} \omega_{ij}$

3. Generate a random number and use it to choose a pair $\langle ij \rangle$ with probability Ω_{ij}.

4. Merge the chosen pair and repeat the procedure until all particles are clustered.

The algorithm produces a graph with a weight $W_{graph} = \prod_{mergings} \Omega_{ij}$. To produce a distribution of graphs for each jet, repeat the algorithm N_{graph} times.

The algorithm reduces to a traditional clustering algorithm when $\alpha \to \infty$. The algorithm is dependent on what distance measure one use, for example using the geometric distance measure of the C/A algorithm.

C.2 Dependence on the Rigidity Parameter and Distance Measure

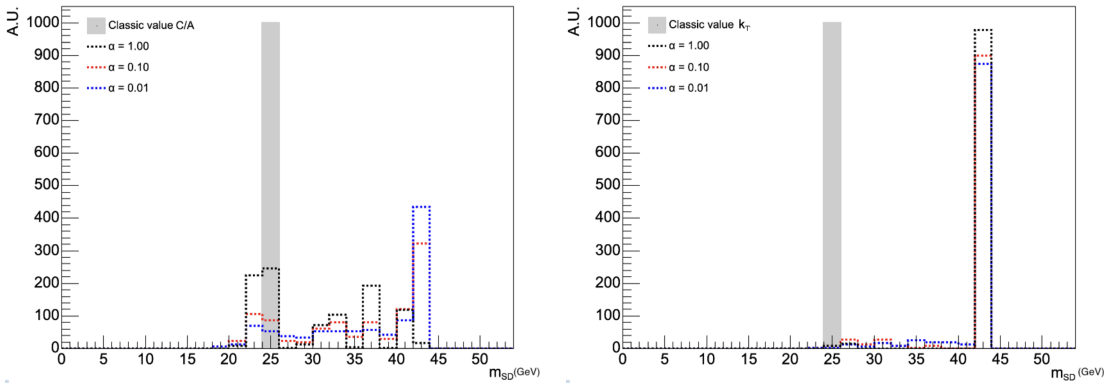


Figure C.1: Distribution of the SoftDrop mass for a jet with $p_T = 440GeV$ and $m = 43GeV$, for 1000 Q-jets. Left: With the C/A distance measure. Right: With the k_T distance measure. The classical value of the respective deterministic clustering algorithm is marked in gray.

The Q-jet algorithm provides us with a distribution of graphs representing the same jet, allowing us to examine the distribution of each observable for each jet. To illustrate this, we select a specific jet with a transverse momentum of $p_T = 440$ GeV and a mass of $m = 43$ GeV, and take a look at one observable.

We examine the SoftDrop mass, which corresponds to the mass of the first SoftDrop splitting. Figure C.1 displays the distribution of SoftDrop masses for 1000 Q-jets obtained using both k_T and C/A distance measures. The gray area indicates the observable value obtained when clustering the jet with a deterministic algorithm. Notably, the classic SoftDrop mass value m_{SD} is smaller than the true mass $m = 43$GeV.

In the left plot of Figure C.1, we observe the SoftDrop mass distribution for Q-jets using the C/A distance measure. The distributions are spread out, and only the distribution corresponding to $\alpha = 1.00$ exhibits a peak around the original value. As the value of α decreases, the distributions of m_{SD} progressively shift closer to the true mass m and deviate from the classical value of the SoftDrop mass m_{SD}.

In the right plot of Figure C.1, we examine the SoftDrop mass distribution for Q-jets employing the k_T distance measure. In this case, all the distributions peak at the true mass m value, significantly differing from the classical m_{SD} value.

Once again, we observe the notable differences between jets clustered using the k_T and C/A distance measures. The SoftDrop mass distribution m_{SD} displays a wider distribution when the C/A distance measure is used, while the distribution associated with the k_T distance measure is more focused around the true mass m value. This

indicates that the C/A algorithm and distance measure are better suited for SoftDrop observables.

For large values of α, the graphs are constrained to resemble classical graphs, resulting in the SoftDrop mass m_{SD} remaining close to the classical value. As α decreases, a broader spectrum of graphs can be considered, leading to a wider range of SoftDrop mass values.

As α decreases further, the differences between the distributions of graphs generated from the C/A and k_T distance measures diminish. In reference [66], it is suggested that for sufficiently small rigidity parameter α, the Q-jets become independent of the choice of distance measure. In conclusions, a non-deterministic jet reconstruction algorithm such as Q-jets allows to explore a wider variety of probable jet trees constructed from the same set of final-state particles, and could thereby be more robust to the choice of a specific jet algorithm.

The Key Concepts of This Appendix

Q-jets is a non-deterministic approach to jet clustering, where multiple weighted graphs are created for each jet. Q-jets can be used to mitigate the dependence on clustering algorithms. By decreasing the rigidity parameter the Q-jets can become more robust towards the choice of clustering algorithm.

Bibliography

[1] Karen M. Burke et al. "Extracting the jet transport coefficient from jet quenching in high-energy heavy-ion collisions". In: *Phys. Rev. C* 90.1 (2014), p. 014909. DOI: 10.1103/PhysRevC.90.014909. arXiv: 1312.5003 [nucl-th].

[2] Luke de Oliveira et al. "Jet-images — deep learning edition". In: *Journal of High Energy Physics* 2016.7 (July 2016). DOI: 10.1007/jhep07(2016)069. URL: https://doi.org/10.1007%2Fjhep07%282016%29069.

[3] Patrick T. Komiske, Eric M. Metodiev, and Matthew D. Schwartz. "Deep learning in color: towards automated quark/gluon jet discrimination". In: *JHEP* 01 (2017), p. 110. DOI: 10.1007/JHEP01(2017)110. arXiv: 1612.01551 [hep-ph].

[4] Daniel Guest et al. "Jet Flavor Classification in High-Energy Physics with Deep Neural Networks". In: *Phys. Rev. D* 94.11 (2016), p. 112002. DOI: 10.1103/PhysRevD.94.112002. arXiv: 1607.08633 [hep-ex].

[5] Yi-Lun Du, Daniel Pablos, and Konrad Tywoniuk. "Deep learning jet modifications in heavy-ion collisions". In: *JHEP* 21 (2020), p. 206. DOI: 10.1007/JHEP03(2021)206. arXiv: 2012.07797 [hep-ph].

[6] Frédéric A. Dreyer and Huilin Qu. "Jet tagging in the Lund plane with graph networks". In: *JHEP* 03 (2021), p. 052. DOI: 10.1007/JHEP03(2021)052. arXiv: 2012.08526 [hep-ph].

[7] Ida-Marie Johansson. *MasterProject*. All code available on GitHub. URL: https://github.com/idamarie1303/MasterProject.

[8] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. "A Brief Introduction to PYTHIA 8.1". In: *Comput.Phys.Commun.* 178 (2008), pp. 852–867. DOI: 10.1016/j.cpc.2008.01.036. arXiv: 0710.3820 [hep-ph]. URL: http://weblib.cern.ch/abstract?CERN-LCGAPP-2007-04.

[9] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "FastJet User Manual". In: *Eur. Phys. J. C* 72 (2012), p. 1896. DOI: 10.1140/epjc/s10052-012-1896-2. arXiv: 1111.6097 [hep-ph].

[10] Simone Marzani, Gregory Soyez, and Michael Spannowsky. *Looking inside jets: an introduction to jet substructure and boosted-object phenomenology*. Vol. 958. Springer, 2019. DOI: 10.1007/978-3-030-15709-8. arXiv: 1901.10342 [hep-ph].

[11] R.Keith Ellis, W.James Stirling, and B.R. Webber. *QCD and collider physics*. Vol. 8. Cambridge University Press, 1996. ISBN: 978-0-511-82328-2, 978-0-521-54589-1. URL: http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=0521581893.

[12] Paul Caucal. "Jet evolution in a dense QCD medium". Other thesis. Oct. 2020. arXiv: 2010.02874 [hep-ph].

[13] Sinead M. Ryan. "Introduction to QCD: basics". Talk at PANDA Phyiscs Winter School, GSI December 2017. 2017. URL: https://indico.gsi.de/event/6430/sessions/4605/attachments/21409/26974/Lecture1and2.pdf.

[14] Gavin P. Salam. "Towards Jetography". In: *Eur. Phys. J. C* 67 (2010), pp. 637–686. DOI: 10.1140/epjc/s10052-010-1314-6. arXiv: 0906.1833 [hep-ph].

[15] Chun Shen. "The standard model for relativistic heavy-ion collisions and electromagnetic tomography". PhD thesis. Ohio State U., July 2014.

[16] K. Yagi, T. Hatsuda, and Y. Miake. *Quark-gluon plasma: From big bang to little bang*. Vol. 23. 2005.

[17] Derek A. Teaney. "Viscous Hydrodynamics and the Quark Gluon Plasma". In: *Quark-gluon plasma 4*. Ed. by Rudolph C. Hwa and Xin-Nian Wang. 2010, pp. 207–266. DOI: 10.1142/9789814293297_0004. arXiv: 0905.2433 [nucl-th].

[18] CERN. "Heavy ions and quark-gluon plasma". Article from CERN website. 2023. URL: https://home.cern/science/physics/heavy-ions-and-quark-gluon-plasma.

[19] Ekkard Schnedermann, Josef Sollfrank, and Ulrich Heinz. "Thermal phenomenology of hadrons from 200A GeV S+S collisions". In: *Phys. Rev. C* 48 (5 Nov. 1993), pp. 2462–2475. DOI: 10.1103/PhysRevC.48.2462. URL: https://link.aps.org/doi/10.1103/PhysRevC.48.2462.

[20] Premomoy Ghosh et al. "Indication of transverse radial flow in high-multiplicity proton–proton collisions at the Large Hadron Collider". In: *J. Phys. G* 41 (2014), p. 035106. DOI: 10.1088/0954-3899/41/3/035106. arXiv: 1402.6813 [hep-ph].

[21] Panjeh. "Monte Carlo Accept-Reject Method". Article from Medium.com. 2020. URL: https://panjeh.medium.com/monte-%20carlo-accept-reject-method-cb6cc2a76840.

[22] Andrew J. Larkoski, Ian Moult, and Benjamin Nachman. "Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning". In: *Phys. Rept.* 841 (2020), pp. 1–63. DOI: 10.1016/j.physrep.2019.11.001. arXiv: 1709.04464 [hep-ph].

[23] Guido Altarelli and G. Parisi. "Asymptotic Freedom in Parton Language". In: *Nucl. Phys. B* 126 (1977), pp. 298–318. DOI: 10.1016/0550-3213(77)90384-4.

[24] Carlos A. Salgado. "Lectures on high-energy heavy-ion collisions at the LHC". In: *2008 European School of High-Energy Physics*. July 2009, pp. 239–280. arXiv: 0907.1219 [hep-ph].

[25] David d'Enterria. "Jet quenching". In: vol. 23. 2010, p. 471. DOI: 10.1007/978-3-642-01539-7_16. arXiv: 0902.2011 [nucl-ex].

[26] Miklos Gyulassy and Michael Plumer. "Jet Quenching in Dense Matter". In: *Phys. Lett. B* 243 (1990), pp. 432–438. DOI: 10.1016/0370-2693(90)91409-5.

[27] Jorge Casalderrey-Solana et al. "A Hybrid Strong/Weak Coupling Approach to Jet Quenching". In: *JHEP* 10 (2014). [Erratum: JHEP 09, 175 (2015)], p. 019. DOI: 10.1007/JHEP09(2015)175. arXiv: 1405.3864 [hep-ph].

[28] Robi Peschanski. "Introduction to String Theory and Gauge/Gravity duality for students in QCD and QGP phenomenology". In: *Acta Phys. Polon. B* 39 (2008). Ed. by R. Fiore, A. Papa, and C. Royon, pp. 2479–2510. arXiv: 0804.3210 [hep-ph].

[29] Jorge Casalderrey-Solana, Jose Guilherme Milhano, and Paloma Quiroga-Arias. "Out of Medium Fragmentation from Long-Lived Jet Showers". In: *Phys. Lett. B* 710 (2012), pp. 175–181. DOI: 10.1016/j.physletb.2012.02.066. arXiv: 1111.0310 [hep-ph].

[30] Jorge Casalderrey-Solana et al. "Predictions for Boson-Jet Observables and Fragmentation Function Ratios from a Hybrid Strong/Weak Coupling Model for Jet Quenching". In: *JHEP* 03 (2016), p. 053. DOI: 10.1007/JHEP03(2016)053. arXiv: 1508.00815 [hep-ph].

[31] J. Casalderrey-Solana et al. "Simultaneous description of hadron and jet suppression in heavy-ion collisions". In: *Physical Review C* 99.5 (May 2019). DOI: 10.1103/physrevc.99.051901. URL: https://doi.org/10.1103%2Fphysrevc.99.051901.

[32] Jorge Casalderrey-Solana et al. "Jet Wake from Linearized Hydrodynamics". In: *JHEP* 05 (2021), p. 230. DOI: 10.1007/JHEP05(2021)230. arXiv: 2010.01140 [hep-ph].

[33] Paul M. Chesler and Krishna Rajagopal. "On the Evolution of Jet Energy and Opening Angle in Strongly Coupled Plasma". In: *JHEP* 05 (2016), p. 098. DOI: 10.1007/JHEP05(2016)098. arXiv: 1511.07567 [hep-th].

[34] Paul M. Chesler and Laurence G. Yaffe. "The Wake of a quark moving through a strongly-coupled plasma". In: *Phys. Rev. Lett.* 99 (2007), p. 152001. DOI: 10.1103/PhysRevLett.99.152001. arXiv: 0706.0368 [hep-th].

[35] Jorge Casalderrey-Solana et al. "Angular Structure of Jet Quenching Within a Hybrid Strong/Weak Coupling Model". In: *JHEP* 03 (2017), p. 135. DOI: 10.1007/JHEP03(2017)135. arXiv: 1609.05842 [hep-ph].

[36] Yilun Du. "Deep learning jet modifications in heavy-ion collisions". Talk at Fysikermøtet 2021. 2021. URL: https://indico.uis.no/event/13/contributions/168/.

[37] John E. Huth et al. "Toward a standardization of jet definitions". In: *1990 DPF Summer Study on High-energy Physics: Research Directions for the Decade (Snowmass 90)*. Dec. 1990, pp. 0134–136.

[38] Harry Arthur Andrews et al. "Novel tools and observables for jet physics in heavy-ion collisions". In: (). arXiv: 1808.03689 [hep-ph].

[39] Bo Andersson et al. "Coherence Effects in Deep Inelastic Scattering". In: *Z. Phys. C* 43 (1989), p. 625. DOI: 10.1007/BF01550942.

[40] Frédéric A. Dreyer, Gavin P. Salam, and Grégory Soyez. "The Lund Jet Plane". In: *JHEP* 12 (2018), p. 064. DOI: 10.1007/JHEP12(2018)064. arXiv: 1807.04758 [hep-ph].

[41] Peter Berta et al. "Particle-level pileup subtraction for jets and jet shapes". In: *JHEP* 06 (2014), p. 092. DOI: 10.1007/JHEP06(2014)092. arXiv: 1403.3108 [hep-ex].

[42] Gavin P Salam Matteo Cacciari and Gregory Soyez. "The catchment area of jets". In: *Journal of High Energy Physics* 2008.04 (Apr. 2008), pp. 005–005. DOI: 10.1088/1126-6708/2008/04/005.

[43] Jonathan M. Butterworth et al. "Jet substructure as a new Higgs search channel at the LHC". In: *Phys. Rev. Lett.* 100 (2008), p. 242001. DOI: 10.1103/PhysRevLett.100.242001. arXiv: 0802.2470 [hep-ph].

[44] Mrinal Dasgupta et al. "Towards an understanding of jet substructure". In: *JHEP* 09 (2013), p. 029. DOI: 10.1007/JHEP09(2013)029. arXiv: 1307.0007 [hep-ph].

[45] Andrew J. Larkoski et al. "Soft Drop". In: *JHEP* 05 (2014), p. 146. DOI: 10.1007/JHEP05(2014)146. arXiv: 1402.2657 [hep-ph].

[46] Giuseppe Carleo et al. "Machine learning and the physical sciences". In: *Rev. Mod. Phys.* 91.4 (2019), p. 045002. DOI: 10.1103/RevModPhys.91.045002. arXiv: 1903.10563 [physics.comp-ph].

[47] Amol Mavuduru. "What "no free lunch" really means in machine learning". Article from Towards Data Science. 2020. URL: https://towardsdatascience.com/what-no-free-lunch-really-means-in-machine-learning-85493215625d.

[48] Rukshan Pramoditha. "The Concept of Artificial Neurons (Perceptrons) in Neural Networks". Article from Towards Data Science. 2021. URL: https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc.

[49] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pp. 315–323. URL: https://proceedings.mlr.press/v15/glorot11a.html.

[50] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: (Feb. 2015). arXiv: 1502.03167 [cs.LG].

[51] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[52] Matthew Stewart. "Simple Introduction to Convolutional Neural Networks". Article from Towards Data Science. 2019. URL: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac.

[53] Benjamin Sanchez-Lengeling et al. "A Gentle Introduction to Graph Neural Networks". In: *Distill* (2021). https://distill.pub/2021/gnn-intro. DOI: 10.23915/distill.00033.

[54] Justin Gilmer et al. “Neural Message Passing for Quantum Chemistry”. In: *CoRR* abs/1704.01212 (2017). arXiv: 1704.01212. URL: http://arxiv.org/abs/1704.01212.

[55] Huilin Qu and Loukas Gouskos. “ParticleNet: Jet Tagging via Particle Clouds”. In: *Phys. Rev. D* 101.5 (2020), p. 056019. DOI: 10.1103/PhysRevD.101.056019. arXiv: 1902.08570 [hep-ph].

[56] Yue Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: (Jan. 2018). arXiv: 1801.07829 [cs.CV].

[57] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[58] Minjie Wang et al. “Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs”. In: *CoRR* abs/1909.01315 (2019). arXiv: 1909.01315. URL: http://arxiv.org/abs/1909.01315.

[59] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *CoRR* abs/1912.01703 (2019). arXiv: 1912.01703. URL: http://arxiv.org/abs/1912.01703.

[60] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: (Feb. 2015). arXiv: 1502.01852 [cs.CV].

[61] A.Y. Ng. “Feature selection, l1 vs. l2 regularization, and rotational invariance”. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 78.

[62] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: Dec. 2014. arXiv: 1412.6980 [cs.LG].

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

[64] Yin Cui et al. *Class-Balanced Loss Based on Effective Number of Samples*. 2019. arXiv: 1901.05555 [cs.CV].

[65] Jasmine Brewer, Quinn Brodsky, and Krishna Rajagopal. “Disentangling jet modification in jet simulations and in Z+jet data”. In: *JHEP* 02 (2022), p. 175. DOI: 10.1007/JHEP02(2022)175. arXiv: 2110.13159 [hep-ph].

[66] Stephen D. Ellis et al. “Qjets: A Non-Deterministic Approach to Tree-Based Jet Substructure”. In: *Phys. Rev. Lett.* 108 (2012), p. 182003. DOI: 10.1103/PhysRevLett.108.182003. arXiv: 1201.1914 [hep-ph].

[67] Yacine Mehtar-Tani, Alba Soto-Ontoso, and Konrad Tywoniuk. “Dynamical grooming of QCD jets”. In: *Phys. Rev. D* 101.3 (2020), p. 034004. DOI: 10.1103/PhysRevD.101.034004. arXiv: 1911.00375 [hep-ph].

[68] Google Developers. “Classification: ROC Curve and AUC”. Tutorial from developers.google.com. 2022. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.