UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Physics and Astronomy

Investigating Jet Physics through Machine Learning-Enhanced Representation Space

by

Giorgio Cerro

ORCiD: 0009-0009-1162-5739

A thesis for the degree of
Doctor of Philosophy

February 2025

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Physics and Astronomy

Doctor of Philosophy

Investigating Jet Physics through Machine Learning-Enhanced Representation Space

by Giorgio Cerro

The exploration of fundamental particles and their interactions has been central to particle physics for centuries, evolving from ancient philosophical concepts to rigorous scientific inquiry. The Standard Model (SM), formulated in the 1960s, represents a monumental achievement in describing three of the four fundamental forces and predicting the properties of elementary particles. Despite its successes, the SM leaves several critical questions unanswered, including issues related to quantum gravity, the hierarchy problem, mass generation, and the cosmological constant.High-energy particle collisions at facilities such as the Large Hadron Collider (LHC) have been instrumental in advancing our understanding of the subatomic world. These collisions generate vast amounts of data, necessitating sophisticated analysis techniques, including clustering and anomaly detection algorithms, to unravel the complexities of particle interactions. The integration of artificial intelligence (AI) and machine learning (ML) into high-energy physics represents a transformative shift in data analysis. By leveraging advanced algorithms, AI enhances the speed and precision of data processing, offering new pathways for discovery and optimisation. This thesis addresses two primary objectives: first, the development of a novel jet clustering algorithm using machine learning that adheres to physical constraints, providing transparency and interpretability compared to traditional deep learning approaches; and second, the application of innovative methods to mitigate performance bias in synthetic data through the exploitation of inherent data symmetries. The structure of this thesis encompasses a comprehensive exploration of the theoretical underpinnings of jets, the impact of machine learning on high-energy physics, the application of spectral clustering techniques, the influence of physical variables on advanced ML models, and strategies for reducing bias in synthetic datasets. Through these contributions, this work aims to enhance the analytical capabilities in particle physics and further our understanding of the universe's fundamental nature.

Contents

List of Figures

List of Tables

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

Signed:..... Date:.....

Part I

Introduction and Background Theory

Chapter 1

Introduction

The quest to understand the universe at its most fundamental level has been a driving force in particle physics for centuries. The concept of matter being composed of indivisible particles dates back to ancient Greece, but it wasn't until the late 19th century that this philosophical notion transformed into scientific inquiry, ushering in an era of groundbreaking particle discoveries—from the electron in 1897 to the Higgs boson in 2012 at the Large Hadron Collider (LHC).

Our most comprehensive understanding of these fundamental particles and their interactions is encapsulated in the Standard Model (SM), a theoretical framework developed in the 1960s by Sheldon L. Glashow, Steven Weinberg, and Abdus Salam. The SM is a remarkably successful quantum field theory that describes three of the four known fundamental forces: the electromagnetic, weak, and strong interactions. It has not only organised the known elementary particles into a coherent structure but has also accurately predicted the existence and properties of new particles, solidifying its position as a cornerstone of modern physics.

However, despite its triumphs, the Standard Model is not the final word. It leaves several profound questions unanswered and is widely considered an effective theory, valid only at lower energy scales. Among the most pressing puzzles are:

- quantum gravity: the SM describes three of the four fundamental interactions at quantum level, but it does not include gravity, for which no satisfactory treatment of a QFT is known;
- hierarchy problem: it is not clear why there are vastly different energy and mass scales into the SM, namely the electroweak scale $M_{EW} \sim 10^2$ GeV and the reduced Planck scale $M_P \sim 10^{18}$ GeV (with $M_P = 1/\sqrt{8\pi G}$ in natural units);

- mass generation problem: while SM successfully explains how particles acquire mass, it does not explain why they have the masses they do and why there is such a wide range of masses in the particle spectrum;
- cosmological constant: the vacuum energy density predicted by the SM is much smaller than the observed cosmological constant of the universe. These are only few of the puzzling problems we face; other issues concern for example the strong CP problem, the origin of the parameters of the SM, the mass of the Higgs boson, the nature of dark matter and dark energy, and others.

High-energy particle collisions are the crucible of modern particle physics, serving as the primary tool for investigating the fundamental building blocks of matter and their interactions. At facilities like CERN's Large Hadron Collider (LHC), Fermilab's Tevatron (now decommissioned), and KEK's SuperKEKB, particles such as protons or electrons are accelerated to nearly the speed of light before being smashed together in powerful collisions. These collisions unleash a cascade of secondary particles, including rare and exotic species that exist only fleetingly.

The analysis of these complex collision events is a monumental undertaking, requiring a sophisticated toolkit of techniques. Clustering algorithms are employed to group particles into jets, collimated sprays of particles that originate from the fragmentation and hadronisation of quarks and gluons. Tagging algorithms are used to identify specific particle types based on their characteristic properties, such as energy deposition, decay products, and interaction patterns. Anomaly detection algorithms are leveraged to sift through vast amounts of data in search of rare events that could signal new physics beyond the Standard Model. Each of these techniques plays a crucial role in extracting meaningful information from the intricate tapestry of particle collisions, enabling scientists to piece together the underlying dynamics of the subatomic world.

The advent of artificial intelligence (AI), particularly machine learning (ML), is transforming the landscape of high-energy physics. The field is characterised by enormous datasets, often petabytes in size, generated by high-luminosity colliders and large-scale detector systems. Traditional analysis methods, while powerful, can be computationally intensive and time-consuming, often requiring significant human intervention.

AI offers a paradigm shift in data analysis by leveraging advanced algorithms to automate and accelerate various tasks. Machine learning models can be trained on vast datasets to recognise patterns, classify particles, and even predict the outcomes of complex interactions. This not only enhances the speed and efficiency of analysis but also opens up new avenues for discovery. AI-powered algorithms can identify subtle anomalies in data that might otherwise go unnoticed, potentially revealing hints of

new particles or phenomena. Furthermore, AI can aid in the design of more efficient detectors and optimise data collection strategies, further pushing the boundaries of experimental capabilities. The integration of AI into high-energy physics research promises to unlock new insights into the fundamental nature of the universe and usher in an era of data-driven discovery.

However, as physicists embracing these powerful tools, we must remain vigilant and remember that our primary goal is to explain nature by providing reasonable explanations for observed phenomena. Therefore, in the work presented in this thesis, we focus on two main objectives: first, to propose a novel algorithm for jet clustering, utilising a well-known machine learning method that has been adapted to respect physical constraints. Unlike many deep learning methods, this algorithm is not a black box; it is designed to be transparent and easy to explore and understand. Second, we employed innovative methodologies to reduce the performance bias on synthetic data in some state-of-the-art neural networks. This was achieved by exploiting the availability of large datasets and leveraging the inherent symmetries in the data.

This thesis is structured as follows:

1. **Theoretical Framework of Jets:** An overview of the theoretical foundations related to jets, collimated sprays of particles produced in high-energy processes, including methods to observe, detect, and study them. This section provides the necessary background on the physics of jets and their significance in high-energy physics (HEP), covering both the underlying principles and the experimental techniques used in their analysis.

2. **Machine Learning in High-Energy Physics:** Exploration of how machine learning is revolutionising data analysis, with an overview of state-of-the-art models addressing challenges in HEP. This includes discussions on clustering, tagging, anomaly detection, matching, and simulation, highlighting the role of machine learning in enhancing the precision and efficiency of data processing in complex HEP experiments.

3. **Spectral Clustering for Jet Physics:** Presentation of our recent work on spectral clustering, an ad hoc implementation aimed at improving jet clustering [8]. This section discusses the competitive results of this approach compared to standard techniques, offering insights into its potential advantages despite its slower computational performance. The implications of these findings for future jet analysis are also considered. Main contributions:

    - Introduction of a Spectral Clustering Method for Jets: A novel alternative to traditional jet clustering algorithms, using multidimensional representations of particle kinematics.

- Infra-Red Safety Confirmation: Demonstration of the method's robustness against IR divergences, ensuring its theoretical soundness.
- Performance Comparison with $Anti - k_T$: Detailed evaluation showing competitive performance in reconstructing relevant final states in MC simulations of complex processes.
- Parameter Independence: Evidence that the method does not require process-specific parameter adjustments, unlike the $anti - k_T$ algorithm, which relies on cone size tuning.
- Applicability to Diverse Scenarios: Demonstrated utility across different physics scenarios, emphasizing the method's adaptability and potential to simplify jet clustering in high-energy physics.

4. **Obscuring Physics Variables:** An examination of advanced machine learning methods, particularly graph neural networks, with a focus on how physical variables influence these models [9]. This chapter delves into the complexity of integrating physical principles into machine learning models and the challenges of ensuring that these models generalise well while maintaining interpretability. The potential for such methods to obscure or enhance the understanding of underlying physical phenomena is critically analysed. Main contributions:
   - Introduction of JetLOV Framework: A novel composite model that combines MLP and LundNet to explore physics-agnostic representations in jet tagging.
   - Demonstrating Physics Independence: Evidence that high-performance jet tagging can be achieved without relying on pre-computed physical variables, emphasizing the network's capacity to learn new representations.
   - Addressing Model Dependence: Insights into mitigating the dependence on pre-defined physics features through generalization and training on diverse datasets.
   - Trade-off Analysis: A critical exploration of the balance between leveraging domain knowledge and allowing machine learning models to independently discover patterns.
   - Implications for Interpretability and Generalization: Advancing the discussion on how machine learning can obscure or enhance the understanding of underlying physical phenomena, with significant implications for the future of jet physics and high-energy physics.
5. **Reducing Bias on Synthetic Data:** A detailed investigation into the use of unsupervised methods for exploring and learning symmetries in data, with the goal of reducing bias in the performance of models trained on synthetic data. This section discusses the significance of identifying and mitigating biases that

arise from synthetic datasets, which are often used to supplement complicated and noisy real-world data in HEP. Main contributions:

- Application of VICReg to Graph Neural Networks: Introduction of an unsupervised learning method to address biases in synthetic datasets used in high-energy physics.
- Balancing Performance and Bias Mitigation: Demonstration of a trade-off between slightly reduced jet tagging performance and improved generalization across datasets with different biases.
- Leveraging Unlabelled Data: Highlighting the potential of pre-training on unlabelled data and fine-tuning on minimal labelled data to improve model adaptability and resilience.
- Generalization Across Simulations: Insights into how this approach enhances model robustness across diverse simulation conditions, reducing dependency on specific dataset characteristics.
- Foundation for Future Work: Establishing a framework for further exploration and refinement, with the goal of developing bias-resistant models that maintain high performance in real-world applications.

Chapter 2

Jet Physics

Jet physics is a crucial aspect of high-energy particle physics, providing essential insights into the fundamental interactions occurring at the smallest scales. This chapter delves into the mechanisms behind jet formation, the algorithms used for jet clustering, and the application of these concepts at the Large Hadron Collider (LHC).

2.1 Jet Formation

Due to Quantum Chromodynamics (QCD) colour confinement, quarks and gluons cannot exist in isolation and are only found as hadronic bound states. In high-energy collider experiments, these partons undergo multiple processes, ultimately manifesting as sprays of colourless hadrons known as jets. This section will cover the fundamental stages of jet formation, including parton showering and hadronisation.

2.1.1 Parton Showering

The first step in jet production is parton showering, which is a sequence of small-angle splits from a parton. The probability of a parton (denoted X) emitting a quark or gluon is given by:

$$P(X \to Xg) \sim \alpha_s \int \frac{dE}{E} \frac{d\theta}{\theta} \tag{2.1}$$

where α_s is the coupling, θ is the angle of emission, and E is the outgoing energy. As the equation suggests, the probability diverges at low θ, indicating that emissions are more likely to occur at small angles. This leads to a series of collimated emissions, forming the initial stages of jet development.

Generalising to various kinds of splittings, the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) equation [10–13], which encodes the behaviour of partons in hadron collisions via the parton distribution functions (PDF) $f(x, \mu)$, at some energy scale μ, can be written down

$$\mu \frac{\partial}{\partial \mu} f_i(x, \mu) = \sum_j \int_x^1 \frac{dz}{z} \frac{\alpha_s}{2\pi} P_{ij}(z) f_j(\frac{x}{z}, \mu). \tag{2.2}$$

Here, the sum over j accounts for all possible types of partons that can split, and $P_{ij}(z)$ are the splitting functions that describe the probability of a parton j splitting into a parton i, where i takes a fraction z of the total momentum of j. The splitting functions for various processes - such as a quark (or antiquark) radiating another quark (or antiquark), a quark splitting into a gluon and a quark, a gluon splitting into a quark-antiquark pair, or a gluon radiating another gluon - are given by:

$$P_{qq}(z) = C_F \left(\left[\frac{1+z^2}{1-z} \right]_+ + \frac{3}{2} \delta(1-z) \right), \tag{2.3}$$

$$P_{qg}(z) = C_F \left(\frac{1+(1-z)^2}{z} \right), \tag{2.4}$$

$$P_{gq}(z) = T_R \left(z^2 + (1-z)^2 \right), \tag{2.5}$$

$$P_{gg}(z) = C_A \left(\frac{z}{1-z} + \frac{1-z}{z} + z(1-z) \right) + \delta(1-z) \frac{11 C_A - 4 n_f T_R}{6}. \tag{2.6}$$

In the preceding equations, z and $(1-z)$ are again the energy fractions, with $C_F = \frac{4}{3}$, $C_A = 3$ and $T_R = \frac{1}{2}$ being the QCD "colour factors" and n_f being the number of fermions coupling to the gluons.

This repetitive splitting leads to the formation of parton shower, where the partons are collinear and soft, causing the final partons to be collimated in the direction of the initial ones. As energy of the initial collision decreases and approaches Λ_{QCD}, quarks and gluons can no longer exist as separate entities and perturbative theory fails due to the running of the QCD coupling constant α_s. This leads to the generation generation of stable colourless hadrons (such as kaons and pion) from coloured partons, a process known as hadronisation. The end result of parton showers and hadronisation is collimated sprays of hadrons known as jets.

In this context, the term $\left[\frac{1+z^2}{1-z}\right]_+$ employs the plus distribution notation, which regulates the divergence of $\frac{1+z^2}{1-z}$ as $z \to 1$. For our particular case, the plus distribution is defined as:

$$\int_0^1 g(z) \left[\frac{1+z^2}{1-z}\right]_+ dz = \int_0^1 g(z) \frac{1+z^2}{1-z} dz - g(1) \int_0^1 \frac{1+z^2}{1-z} dz, \tag{2.7}$$

where $g(z)$ is a smooth test function. The second term subtracts the contribution at z = 1, ensuring the integral is finite. This regulation is crucial in QCD because the divergence of $\frac{1+z^2}{1-z}$ would otherwise lead to undefined results in physical calculations.

The $\delta(1-z)$ term in the splitting function accounts for soft-collinear singularities and ensures proper normalization of the splitting process. Together, these components make $P_{qq}(z)$ well-defined and physically meaningful for parton splitting probabilities.

Colour and Spin Propagation

Parton showers use a branching structure where each splitting creates "daughter" partons from a "mother" parton. This genealogy is crucial for tracking the evolution of the shower, implementing momentum conservation and propagating colour and spin information.

In parton showers, the propagation of colour and spin information is crucial for accurately modelling the evolution of quarks and gluons. As partons split, the colour charge must be conserved and properly distributed among the daughter particles. This affects the subsequent radiation patterns and hadronisation process. The spin of the parent parton influences the angular distribution of its daughters. Incorporating spin effects leads to more accurate predictions of jet substructure and particle correlations.

Types of Parton Showers

There are several approaches to implementing parton showers [14], each with its own characteristics:

- **Angular-Ordered Showers** [15; 16]. Angular-ordered showers implement colour coherence by ordering emissions according to their angular separation from the initiating parton. In this scheme, emissions are ordered in decreasing angle, meaning that soft emissions tend to occur at larger angles relative to the direction of the initial parton. This ordering accurately captures wide-angle soft gluon effects, which are critical for understanding phenomena such as jet broadening and the evolution of jet structures. The angular ordering helps to maintain the physical characteristics of the emitted particles, ensuring that they exhibit the expected correlations that arise from the conservation of colour charge.

- **p_T-Ordered Showers** [17; 18]. In contrast to angular-ordered showers, p_T-ordered showers arrange emissions based on their transverse momentum. This approach simplifies the implementation of the showering algorithm, as emissions are generated in order of decreasing p_T. However, while this method can provide a computationally efficient way to model parton showers, it may necessitate additional corrections to fully account for colour coherence effects that are naturally incorporated in angular-ordered schemes. Consequently, the p_T-ordered approach may lead to discrepancies in the predictions of observables that are sensitive to colour correlations.

- **Dipole Showers** [19]. Dipole showers represent emissions as arising from colour dipoles rather than individual partons. This framework inherently incorporates some of the coherence effects observed in parton showers, as the interactions between dipoles account for the colour structure of the emitted particles. Dipole showers can simplify the momentum conservation equations, making them more straightforward to implement in simulations. Additionally, this approach allows for a more natural incorporation of both soft and collinear emissions, providing a flexible framework that can be adapted to different aspects of the parton showering process.

Higher-Order Accuracy

Parton shower algorithms are primarily based on leading-order (LO) approximations, but significant advancements have been made in incorporating higher-order corrections. These corrections are computed using fixed-order perturbative QCD, where the cross-section is expanded in powers of the strong coupling constant, α_s. The terms in this expansion, known as next-to-i-leading order (N^iLO), account for additional emissions or virtual corrections at each order. Recent breakthroughs have enabled precise calculations at NNLO and even N^3LO accuracy for specific processes, such as Higgs production via gluon-gluon fusion. However, at small scales, large logarithmic corrections arising from soft or collinear emissions can dominate and invalidate fixed-order predictions. To address this, resummation techniques are employed to account for these logarithms systematically, restoring theoretical accuracy. Furthermore, matching and merging schemes combine fixed-order precision with parton showers, providing reliable predictions across a wide phase space while maintaining infrared and collinear safety.

2.1.2 Hadronisation

Hadronisation is the process by which partons transform into hadrons. This non-perturbative phenomenon occurs when the energy scale of the partons falls below Λ_{QCD}. During hadronisation, quarks and gluons form colourless hadrons through the

creation of quark-antiquark pairs and gluon emissions, leading to the formation of mesons and baryons. The resulting hadrons are then detected as jets in collider experiments, providing a crucial link between the theoretical framework of QCD and observable phenomena in high-energy physics.

As we have established, the QCD coupling α_s runs with the energy scale Q. Inside a detector, the primary interaction vertex where the accelerated partons collide is at high energy, placing us in the perturbative regime of QCD where quarks and gluons are (almost) free. By the time particles reach the detector, however, we are at larger distance scales, and hence lower energy scales, where quarks and gluons are strongly coupled, and perturbation theory no longer holds. This is where hadronisation (also referred to as fragmentation) occurs, resulting in coloured partons forming stable colourless hadrons (such as pions or kaons) which are eventually detected.

A key concept for approaching hadronisation models is the local parton-hadron duality (LPHD) [20], which states that there is a similarity between global jet features computed at the parton level (after showering, but before hadronisation) and those measured after hadronisation. This concept is fundamental in understanding how theoretical predictions at the parton level translate to observable jet characteristics in experiments. In Appendix A, we show a simple exercise that demonstrates the LPHD. This is achieved using a novel metric used in jet physics is the Earth Mover's Distance (EMD) [7; 21], which is based on the Wasserstein distance [22], a measure of how "distant" two probability distributions are from one another. The Wasserstein distance is particularly effective for comparing distributions by quantifying the minimum "cost" to transform one distribution into another. In the case of jets, we refer to this metric as the Earth Mover's Distance because it measures the amount of energy (analogous to "earth") that needs to be moved from one jet to match another in the eta-phi space. This makes the EMD especially useful for comparing energy distributions between particles in jets, capturing their physical differences in a meaningful way.

2.1.2.1 Longitudinal Tube Model

A particular simple example of a hadronisation model is the longitudinal tube model [23], where hadrons arising from a pair of colour-connected partons are confined to a cylinder in (y, p_T). Here y is the rapidity coordinated definite by [24]

$$y = \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right), \tag{2.8}$$

for a particle with energy E and momentum along the $z-$axis (taken to be the beam line in colliders) p_Z, and

$$p_T = \sqrt{(p_x^2 + p_y^2)} \tag{2.9}$$

being the magnitude of the momentum vector in the $x-y$ plane, with the four-momentum defined as $p = (E, p_x, p_y, p_z)$. Defining the hadron density $\rho(p_T)$, the energy and momentum for a jet are written as

$$E = \int_0^Y dy d^2p_T \rho(p_T) p_T \cosh y = \lambda \sinh Y \tag{2.10}$$
$$P = \int_0^Y dy d^2p_T \rho(p_T) p_T \sinh y = \lambda(\cosh Y - 1) \sim E - \lambda$$

where we are integrating over a tube of length Y, and

$$\lambda = \int d^2p_T \rho(p_T) p_T \tag{2.11}$$

sets the scale for hadronisation. Notice that the jet momentum P receives a negative hadronisation correction of relative order $\lambda/E = 2\lambda/Q$ for a two-jet configuration of total energy Q. Thus one generally expects hadronisation effects to scale with energy like $1/Q$. From these equations we expect a mean-square hadronisation contribution to jet mass of

$$< M^2 >= E^2 - P^2 \sim \lambda Q. \tag{2.12}$$

Comparing the perturbative predictions for jet masses with experiment, one finds that a hadronisation correction corresponding to $\lambda \sim 0.5$ GeV is required. Note that this implies a fairly large jet mass in addition to the perturbative contribution.

2.1.2.2 String Model

Another method to describe the hadronisation process is the String Model [25–29], which is most easily described for e^+e^- annihilation. A schematic picture can be seen in Figure 2.1. The produced quark and antiquark move out in opposite directions, losing energy to the colour field, which collapses into a string-like configuration between them. The string has a uniform energy per unit length, corresponding to a linear quark confining potential, which is consistent with quarkonium spectroscopy. The string may be broken up starting at either the quark or the antiquark end, or both simultaneously (the breaking points have space-like separations, so their temporal sequence is frame-dependent), and it proceeds iteratively by $q\bar{q}$ pair creation, as in independent fragmentation.

If we label the initial quark antiquark pair as $q_0\bar{q}_0$, then after a splitting we produce a new quark pair $q_1\bar{q}_1$, which are organised into bound states $q_0\bar{q}_1$ and $q_1\bar{q}_0$. The newly produced hadron $h(q_0, \bar{q}_1)$, carries a fraction z of momentum modelled by the distribution

$$f(z) \sim \frac{(1-z)^a}{z} e^{-b\frac{m_h^2}{z}} \tag{2.13}$$

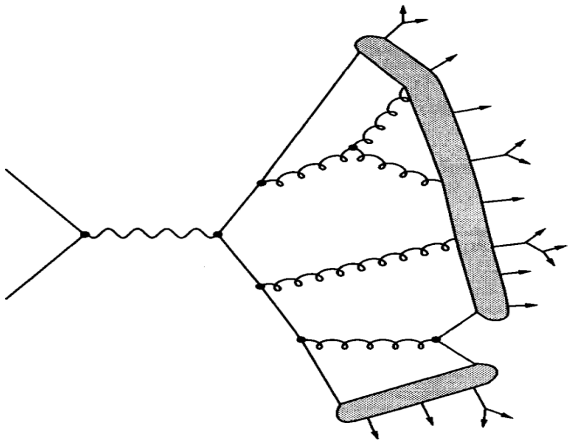


FIGURE 2.1: Parton shower with string hadronisation model for $e^+e^- \rightarrow hadrons$ [1].

where a and b are parameters tuned via experimental data.

2.1.2.3 Cluster Model

A key feature of the parton branching process is the preconfinement of colour, which plays an important role in hadronisation. Preconfinement ensures that, after the perturbative phase of jet development, pairs of colour-connected neighbouring partons tend to form clusters with relatively small masses. These clusters have a mass distribution that rapidly decreases at high masses and remains independent of the energy scale Q^2, meaning it behaves similarly regardless of the energy of the collision. Importantly, this mass distribution is universal, applying across different energy scales and processes [30].

This property forms the basis of cluster hadronisation models. After the perturbative parton showering process, where quarks and gluons are emitted, colour-singlet clusters of partons are formed. These clusters are created by grouping nearby colour-connected partons into colour-neutral combinations, known as preconfinement clusters. As these clusters have relatively low masses due to preconfinement, they can then decay into the final observed hadrons (such as pions, kaons, etc.). This process effectively bridges the gap between the high-energy parton-level dynamics and the low-energy hadronic states observed in experiments.

The most straightforward mechanism for the formation of colour-singlet clusters following parton branching involves the non-perturbative splitting of gluons into $q\bar{q}$ pairs. Neighbouring quarks and antiquarks can then combine to form singlets. The resulting cluster mass spectrum is again universal, with a steep decline at high masses. Its exact shape is determined by the QCD scale Λ, the cut-off scale t_0, and, to a lesser extent, the specifics of the gluon-splitting process. Typically, cluster masses are about two to three times $\sqrt{t_0}$. A schematic representation of this process is shown in

Figure 2.2, which illustrates the cluster hadronisation of the same parton shower depicted in Figure 2.1.

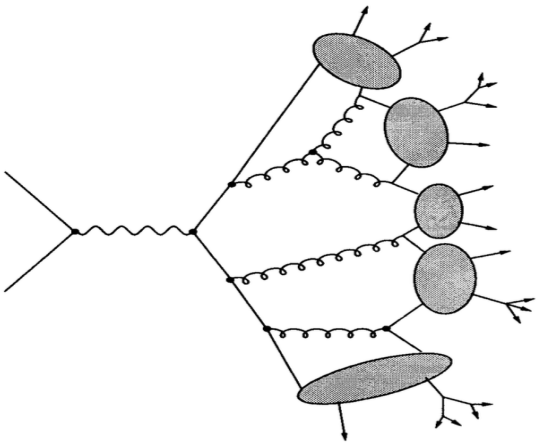


FIGURE 2.2: Parton shower with cluster hadronisation model for $e^+e^- \rightarrow hadrons$ [1].

2.2 Jet Clustering Algorithms

In high-energy particle collider experiments, the large-distance behaviour of QCD results in parton showering and fragmentation, which produce collimated sprays of colour-neutral hadrons detected by the experiments. However, jets are not fundamental objects in the theory. To study them, we need a precise definition, which is achieved through jet clustering algorithms.

Jet clustering algorithms map the complex detector environment, containing these sprays of hadrons, back to the original hard interactions. Once defined, jets can be tagged based on the originating parton, allowing us to infer the nature of the underlying interactions.

For example, consider the Standard Model production of a single Higgs boson decaying into a $b\bar{b}$ pair. If the Higgs is produced at rest, the outgoing b-quarks will be back-to-back, forming two well-resolved jets. These jets can be b-tagged (a process discussed later). By plotting the invariant mass of the dijet system (combining the two b-jets), we should observe a peak around the Higgs boson mass.

Assuming we can accurately identify these jets as originating from b-quarks (thus labelling them as b-jets), we can construct the invariant mass of the dijet system to determine the source of the $b\bar{b}$ pair. By generating a sample of Monte Carlo events, we can predict the mass distribution for a pair of b-jets from a Higgs boson, as illustrated in Figure 2.3. The definition of jets in this context is achieved using a jet algorithm, which will be reviewed in the next section.

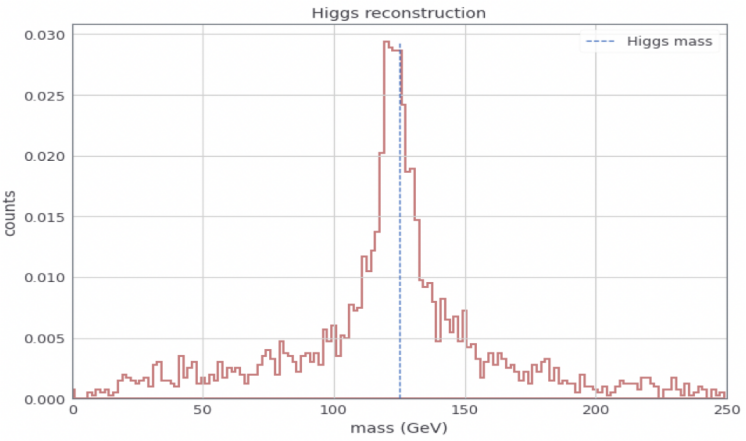


FIGURE 2.3: The $m_{b\bar{b}}$ distribution from the decay of an SM Higgs boson (m_H = 125 GeV).

2.2.1 Sterman-Weinberg Jets

The Sterman-Weinberg clustering jet algorithm [31], introduced by George Sterman and Steven Weinberg in 1977, is one of the earliest jet algorithms (in particular cone jet algorithms) developed to identify and study jets in high-energy physics experiments. This algorithm is particularly significant because it laid the groundwork for subsequent jet definition methods. The Sterman-Weinberg algorithm works by identifying clusters of particles within a cone of fixed opening angle and requires a certain fraction of the total energy to be contained within these cones. Specifically, it defines a jet as any group of particles whose total energy within a cone of half-angle δ exceeds a threshold E_{min}. Additionally, the algorithm stipulates that the energy outside the cones but within a broader region must not exceed a small fraction ϵ of the total energy. Mathematically, this constraint can be expressed as:

$$E_{\text{outside}} \leq \epsilon E_{\text{total}}, \tag{2.14}$$

where E_{outside} is the energy outside the jets, and E_{total} is the total energy of the event. This can be seen in Figure 2.4. This method is infrared and collinear safe, we will talk about this more in the next section, meaning it is not sensitive to the addition of soft particles or the splitting of a particle into collinear fragments, making it robust for practical applications. The Sterman-Weinberg algorithm, while conceptually simple, has been instrumental in advancing the understanding of jet formation and has influenced the development of more sophisticated jet algorithms used in contemporary particle physics research.

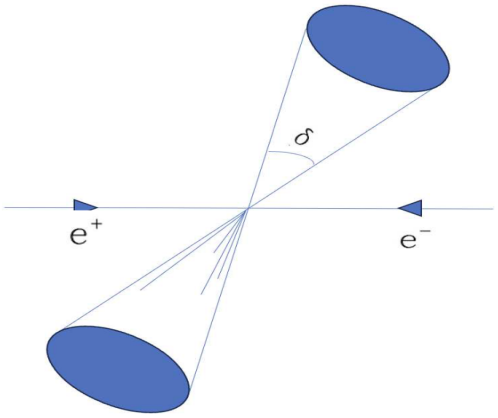


FIGURE 2.4: Diagrammatic representation of Sterman-Weinberg cone jets.

2.2.2 JADE Algorithm

The first example of a sequential recombination algorithm is the JADE algorithm [32]. These algorithms work by iteratively combining the final state particles. An intuitive way to understand this is by imagining the reversal of the showering process, tracing back to the parton from which the jet originated.

To implement such an algorithm, we must define a distance measure between a pair of particles i and j. For JADE, we use

$$d_{ij} = \frac{m_{ij}^2}{E_{tot}^2} \sim \frac{2E_iE_j(1-\cos^2\theta_{ij})}{E_{tot}^2} \tag{2.15}$$

where E_{tot} represents the energy in the entire event. The algorithm then proceeds as follows:

1. Calculate the all the pairwise distances d_{ij}.
2. Find the pair with the minimum d_{ij}, referred to as $i_0$and j_0.
3. Define a cut off distance measure d_0.
4. If $d_{i_0j_0} < d_0$, particles i_0 and j_0 are combined into a single new object called a pseudojet z with four-momentum equal to the sum of the two merged particles' four-momentum: $p_z = p_{i_0} + p_{j_0}$.
5. Repeat steps 1-4 until $d_{i_0j_0} \geq d_0$, at which point all the remaining particles are declared jets.

The distance measure vanishes for both collinear pairs ($\theta_{ij} \to 0$) and for soft particles ($E_{ij} \to 0$). In practice, as the algorithm processes an event, it prioritises clustering regions with collinear and soft emissions. However, this can potentially cause issues

where two widely separated soft particles ($E_i, E_j \to 0$) are clustered before another pair, since the energy of both particles is accounted for in d_{ij}.

Clearly, this algorithm is significantly more computationally intensive than cone algorithms but offers advantages in the performance. The flexibility of having a single input parameter d_0 allows the JADE algorithm to handle multijet events effectively, which can be challenging for cone algorithms.

2.2.3 k_T Algorithm

Following the development of the JADE algorithm, the next significant sequential recombination algorithm to emerge was the k_T Algorithm [33; 34]. This algorithm was primarily developed for use in $e^+e^- \to$ hadrons events and represents an incremental improvement over JADE rather than a complete redesign.

The main issue with the JADE algorithm was that its distance measure d_{ij} could cause wide-angle, soft particle pairs to be clustered before more appropriate choices. To address this, the k_T algorithm introduced a modified distance measure:

$$d_{ij} = \frac{2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})}{E_{\text{tot}}^2}. \tag{2.16}$$

In this modification, the product E_iE_j is replaced by $\min(E_i^2, E_j^2)$. The use of the min function ensures that only the energy of the softer particle in the pair i and j is considered, which prevents widely separated soft particles from being clustered together prematurely. Instead, particularly soft particles are more likely to be clustered with nearby neighbours.

The algorithm proceeds similarly to the JADE algorithm, iteratively clustering pairs that minimise d_{ij} until all remaining pairs exceed the cutoff d_0. This adjustment allows the k_T algorithm to handle soft emissions and wide-angle particles more effectively, resulting in more accurate jet clustering.

2.2.4 Generalised k_T Algorithm

An enhancement to the clustering methods discussed earlier involves refining the distance measure to better capture the similarities between particles. The generalised form of the k_T algorithm achieves this by modifying the distance metric:

$$d_{ij} = \frac{\min(p_{T_i}^n, p_{T_j}^n)\Delta R_{ij}^2}{R^2} \tag{2.17}$$

Here,

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2 \tag{2.18}$$

represents the angular separation between particles i and j, where y and ϕ denote the rapidity and azimuthal angle of the particles, respectively. The parameter R serves as an input that determines the scale of the clustering, analogous to d_0 in the JADE algorithm, while n is a tunable parameter chosen a priori. The clustering process follows the same steps as previously mentioned: creating a distance matrix, identifying the closest pair of particles, merging them, and repeating until a stopping condition is met.

Despite the seemingly minor change in the distance measure, this modification significantly improves performance and leads to two of the most widely used and robust clustering algorithms: the anti-k_T algorithm and the Cambridge-Aachen algorithm. These algorithms are distinguished by different choices of the parameter n.

2.2.4.1 The Anti-k_T Algorithm

The anti-k_T algorithm [5; 33; 35], a variant of the k_T algorithm, arises when the parameter n is set to -1. This specific choice inverts the behaviour of the distance measure, prioritising the clustering of high-energy particles first. The anti-k_T algorithm effectively reverses the original k_T approach, which weighted the distance measure by the minimum energy of the particle pair to prevent widely separated, soft particles from clustering together.

This inversion leads to the formation of jets with well-defined, conical shapes, making the anti-k_T algorithm particularly stable and reliable for reconstructing jets in high-energy physics experiments. Its robustness in maintaining jet shapes has made it a preferred method in many analyses, especially in environments where precise jet reconstruction is crucial.

2.2.4.2 The Cambridge-Aachen Algorithm

In contrast, the Cambridge-Aachen algorithm [36] emerges when the parameter n is set to 0. This choice results in a purely geometric distance measure, which depends solely on the angular separation between particles, without considering their energies.

The Cambridge-Aachen algorithm clusters particles based on their relative angles, making it highly suitable for studying jet substructure and exploring the internal dynamics of jets. Its focus on angular proximity rather than energy allows for a detailed examination of the finer details within jets, providing insights into complex jet structures that are not as easily captured by energy-based clustering methods.

2.2.4.3 Variable-R Jets

A key parameter that appears in all the sequential recombination algorithms discussed so far is R , which is responsible for the stopping condition and is related to the cone size. This parameter needs to be chosen a priori; however, not all jets fit neatly into a single cone size, making the choice of this parameter challenging.

In fact, a rule of thumb for selecting a good value is to approximate the cone size using the following relation:

$$R \sim \frac{2m}{p_T}, \tag{2.19}$$

where R depends on the jet's transverse momentum p_T and its mass m. For hard (high p_T) jets, one can expect tightly compact, narrow cone jets, while for softer (low p_T) jets, the resulting jet constituents are more spread out. Therefore, a jet definition that accounts for this behaviour should be able to accurately cluster jets of different sizes.

A novel solution, the variable-R algorithm [37], removes the need to settle on a single fixed cone, which might not be suitable for multi-jet events involving boosted objects. The distance measure is modified as follows:

$$d_{ij} = \frac{\min(p_{T_i}^n, p_{T_j}^n)\Delta R_{ij}^2}{R_{eff}^2(p_{T_i})}, \tag{2.20}$$

where R_{eff} replaces the fixed input parameter R in traditional algorithms:

$$R_{eff}(p_{T_i}) = \frac{\rho}{p_T}. \tag{2.21}$$

We immediately notice that the new measure R_{eff} encodes the dependence shown in the equation above, where ρ is a dimensionful input parameter, generally set to match the p_T scale of the jets in the event being clustered.

The algorithm proceeds in the same way as the generalised k_T algorithm. Additionally, we note the appearance of n in d_{ij}, which allows the Variable-R algorithm to be modified to mimic the anti-k_T or Cambridge-Aachen algorithms.

Some recent works [38; 39] compare the performance of this method with the traditional generalised k_T algorithm, with a particular focus on the 2HDM model, a beyond Standard Model solution.

2.2.5 Modern Clustering Methods

All the clustering methods discussed so far are foundational for jet clustering, being among the oldest, most robust, and most reliable techniques. In recent years, with the exponential growth of Machine Learning, the physics community has begun implementing new models that leverage the vast amount of data and computational resources necessary to train sophisticated algorithms. In Section 3, we will explore some of these methods in detail. Additionally, we will discuss a recently developed method, Spectral Clustering for Jet Clustering, in Section 4.

While some of these new methods exhibit high performance, the physics community still tends to prefer and use these time-tested clustering methods due to their simplicity, interpretability, and speed. Despite the revolutionary advancements in AI and computational techniques, physicists strive to keep pace by developing increasingly sophisticated models. However, it is crucial to recognise that in physics, "good" is not enough. As we aim to represent and explain the natural world, we must incorporate physical constraints into our models, as discussed in Section 2.2.6. This is not always the case with newer methods, making it imperative to balance performance with adherence to fundamental physical principles.

2.2.6 Infrared and Collinear Safety

Insensitivity to soft or collinear emissions is a crucial attribute for jets, known as infrared (IR) safety. Formally, a soft emission refers to the emission of a massless particle with very low energy. As the energy of the emitted particle approaches zero, the probability calculation diverges, making perturbative calculations impossible. However, since a massless particle with zero energy is unobservable, this does not pose a practical problem. Similarly, a singularity arises when a particle undergoes collinear splitting, producing two particles with the same momentum direction. This decay is also undetectable because two particles in the exact same location with the same combined momentum will be measured as indistinguishable from the original particle [40].

There are three primary reasons for requiring an algorithm that maintains the measurable properties of jets in the presence of IR radiation [41]:

1. If a jet clustering algorithm were sensitive to soft or collinear splittings, it would be impossible to make reliable predictions about jet properties using (QCD). Such sensitivity would undermine many important theoretical comparisons, making IR safety essential for adhering to the general properties of jet definitions established with the "Snowmass Accords" [42].

2. Monte Carlo simulations are based on probabilities calculated from QCD. These simulations must be tuned and compared to experimental data using measurable quantities. If a simulation's accuracy suffers in the IR limit, it becomes unreliable. Therefore, jets derived from Monte Carlo data must be designed to be independent of the IR behaviour of the simulation.

3. The specific energy at which a soft particle becomes undetectable or a collinear splitting becomes indistinguishable depends on the detector used. This dependency complicates the predictions of an IR-unsafe algorithm, as its behaviour would be unique to each detector.

Cone algorithms, which have not been examined in detail here, often face challenges with infrared and collinear (IRC) safety. Fortunately, the sequential recombination algorithms discussed thus far are IRC-safe. IRC safety remains a relevant concern for recent algorithms, particularly within the context of Machine Learning, where controlling the model's behaviour can be complex. A key question is whether to prioritise a clustering method that offers superior performance but lacks IRC safety, or to choose one with potentially lower performance but adheres to this fundamental constraint. In Section 4, we will investigate a novel clustering technique and discuss how IRC safety has been integrated into the algorithm's design. Meanwhile, in Section 3, we will examine a modern Graph Neural Network developed specifically for jet tagging, with a focus on ensuring IRC safety within its architecture.

2.3 Jets at LHC

So far we have been discussing jets and their reconstruction from a theoretical point of view. However, it is important to understand how jets originate in an experimental setup since they hold the key to discovering new physics and particles at the LHC. In this section, we will review the concept of jets at the LHC, with particular emphasis on the CMS detector phenomenology.

2.3.1 The Particle Detector

The LHC is the world's largest and one of the most powerful particle colliders, featuring a 27 km circular ring of superconducting magnets. Inside the detector, two high-energy proton beams collide, producing hadrons that emerge from parton showers and hadronisation, dispersing in all directions from the primary vertex. The detector's cylindrical shape around the beamlines helps confine the particles and captures emissions close to the beamline.

Several concurrent experiments operate at the LHC, including ALICE, ATLAS, CMS, and LHCb, each with distinct objectives. Without going into details of the description of the various components of a detector, as we are mainly interested in its geometry, Figure 2.5 shows what a detector looks like, for example the CMS (Compact Muon Solenoid) detector [43].

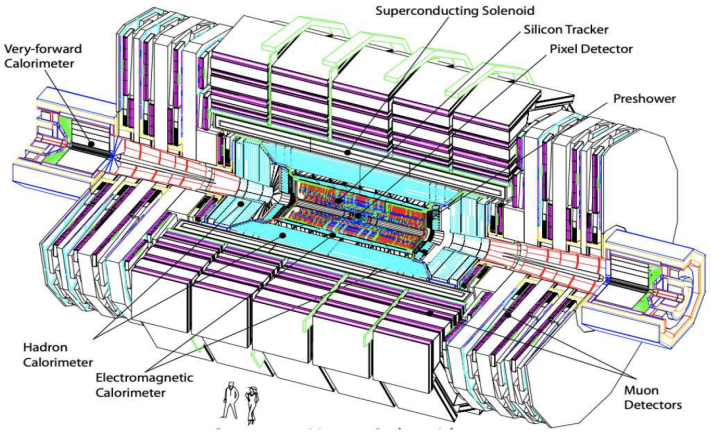


FIGURE 2.5: A pictorial representation of the CMS detector [2].

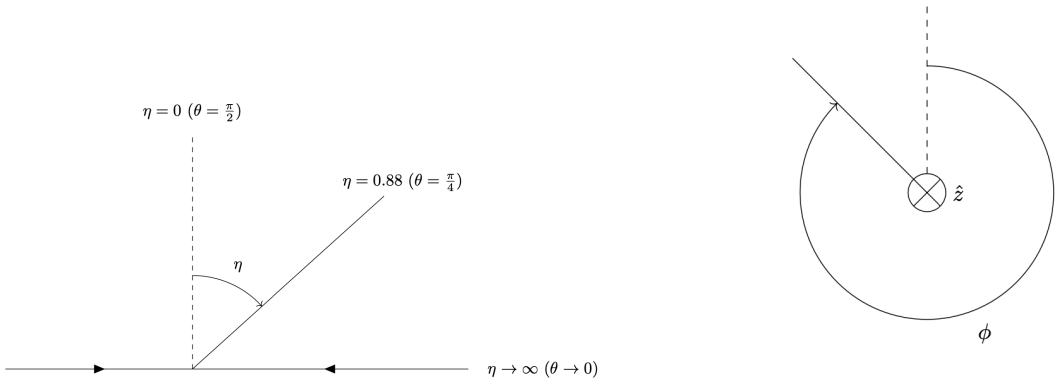
For the geometry of the detector, the beamline is considered the z-direction. The angular coordinates are the rapidity y, defined in Equation 2.8 and the pseudorapidity η, defined as

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) = \frac{1}{2}\ln\left(\frac{|p| + p_z}{|p| - p_z}\right). \tag{2.22}$$

In the pseudorapidity definition, E is replaced with the magnitude of the three-momentum $|p|$, making it a massless approximation of rapidity. Both quantities measure the momentum an object has in the z-direction. The angle θ is the angle relative to the beamline (z-direction). Another crucial coordinate around the beamline is the rotation angle ϕ, which starts from the x-axis and rotates in the x-y plane. In Figure 2.6 we can see an illustration of the ranges for the values of the two variables.

2.3.2 Jet Tagging

To properly analyse the underlying physics at the LHC, jet taggers are employed to extract detailed information about the original partons from which the final state jets originated. This is a crucial yet complex task that has seen significant performance improvements with the advent of Machine Learning (ML) supervised methods. For example, CMS results from a few years ago demonstrated a 15% increase in tagging efficiency after incorporating ML techniques [44]. This marked the beginning of a



(A) (B)

FIGURE 2.6: Demonstration of the psuedorapidity η (A) and of the azimuthal angle ϕ (B) with respect to the beamline of an event at the LHC.

rapid advancement in the field, with the most advanced algorithms now achieving exceptionally high performance.

Typically, the first step in testing a new method is to address a simple binary classification problem. In many cases, this involves distinguishing between jets originating from W bosons or Top quarks against those from QCD jets. A more challenging task is to accurately classify quark jets versus gluon jets, as the differences in their substructures are more subtle and harder to distinguish. State-of-the-art machine learning models, as discussed in Section 3, can achieve an accuracy exceeding 95%, which is nearly perfect. However, in practical applications with real data, rather than synthetic data generated by simulation software, these impressive performance metrics often decline significantly. This drop in performance is a common issue for any model deployed in a production environment.

Several factors contribute to this performance degradation. First, real-world scenarios require handling multi-class classification tasks, which are inherently more complex than binary classification. Additionally, the data encountered in actual experiments is much messier, introducing a level of noise and variability not present in synthetic datasets. Furthermore, errors in the clustering methods themselves can adversely affect tagging performance.

Despite these challenges, continuous improvements in ML models and techniques hold the promise of bridging the gap between theoretical performance and practical application. Ongoing research focuses not only on enhancing the accuracy and robustness of jet taggers but also on integrating physical constraints and domain knowledge into the models, ensuring that they remain grounded in the underlying physics principles.

2.3.3 Jet Contamination Sources

So far, we have discussed the reconstruction of jets at the LHC in the simplest scenario, where the incoming radiations originate from the hard interaction of interest. However, the LHC environment is filled with additional hadronic events and unwanted radiation, complicating the jet reconstruction process. These unwanted contributions come from several sources, such as Pile-Up (PU), Multi-Parton Interactions (MPI) and the Soft Underlying Event (SUE).

2.3.3.1 Multi-Parton Interactions

Multi-Parton Interactions refer to the phenomenon where multiple independent scatterings occur between partons within a single proton-proton collision. Although the primary hard scattering is the focus of most analyses, these additional, softer interactions can contribute extra hadronic radiation, filling the detector with low-energy particles that overlap with the jet. MPI can introduce spurious energy into jets, affecting both their reconstructed mass and substructure.

The complexity of modeling MPI is crucial for accurate jet reconstruction, as it must be accounted for in both simulations and data. Tools like PYTHIA8 [45] are often used to simulate MPI, with specific "tunes" designed to better reproduce experimental results by adjusting parameters such as the parton distribution functions and parton shower models [46]. Additionally, jet grooming algorithms like trimming and pruning help mitigate the contributions of MPI by removing soft and wide-angle radiation that is more likely to come from MPI than the hard scatter.

2.3.3.2 Underlying Event

The Underlying Event is the collection of particles produced by the remnants of the initial proton-proton collision that are not directly involved in the primary hard scattering. These remnants consist of soft hadrons from the break-up of the initial-state protons, and while their energy is typically low compared to the jet from the hard scatter, they are numerous and can affect the overall energy deposition in the detector.

UE contributes an isotropic, soft energy background that can inflate the energy scale of jets, making it essential to remove or correct for these contributions to accurately measure jet properties. Theoretical models and event generators like HERWIG [47] and PYTHIA [45] include parameters for tuning the UE, ensuring that it is well modelled in simulations [48]. In addition to jet grooming techniques, specific UE subtraction methods are used to estimate and remove the energy contribution from UE, leaving behind the more relevant signal from the hard scatter.

2.3.3.3 Pile-Up

Pile-Up results from multiple proton-proton collisions occurring in the same bunch crossing. In high-luminosity environments like the LHC, where dozens of interactions can occur simultaneously, PU introduces a large number of soft particles, especially low-energy hadrons, which contaminate the detector and overlap with jets of interest. PU introduces both charged and neutral particles into the jet cone, increasing its reconstructed transverse momentum (p_T) and distorting its substructure.

Mitigation of PU is critical for accurate jet reconstruction. The Pile-Up Per Particle Identification (PUPPI) algorithm is one of the most effective tools used at the LHC to reduce PU contamination. PUPPI assigns weights to each particle based on its likelihood of originating from PU events by considering both its kinematic properties and proximity to the primary interaction vertex [49]. The method helps suppress contributions from PU, allowing for a clearer identification of the jets from the hard scatter.

2.3.4 Jet Grooming

When reconstructing jets at the LHC, the presence of additional radiations can obscure the features relevant to the analysis, complicating the task of distinguishing the hard scattering products from unwanted background radiation. This separation is particularly challenging for hadronic final states, where soft radiation from sources such as Multiple Parton Interactions (MPI), the Underlying Event (UE), and Pile-Up (PU) can significantly contaminate the jet structure. The overlapping nature of these background contributions can inflate the energy of jets, blur their boundaries, and hinder the identification of meaningful physics signals.

More generally, separating the hard scattering events from these unwanted contributions—often referred to as “junk”—poses a significant challenge. This is especially true in the case of hadronic final states, where the inherent complexity of the LHC environment exacerbates the difficulties. The presence of soft and collinear radiation from Multiple Parton Interactions can create a dense background of additional particles that overlap with the jets of interest. Moreover, the Underlying Event can generate a fluctuating and non-uniform distribution of soft particles that further complicate the identification of hard scatterings. Pile-Up events, resulting from multiple simultaneous proton-proton collisions, add yet another layer of complexity by contributing additional jets and particles that can obscure the features of the primary jet.

These overlapping contributions can lead to an inflation of jet energy, misidentification of jet boundaries, and the dilution of signal characteristics, making it

challenging to extract the meaningful physics signals that are critical for analyses. As a result, it is essential to apply cleaning techniques to isolate the relevant jets and mitigate these unwanted contributions. Jet grooming and PU subtraction methods are employed at the LHC to refine the jet structure, ensuring that only particles pertinent to the jet substructure study are retained.

In this section, we briefly discuss some key grooming and PU subtraction methods that help disentangle the true jet signal from the complex environment of the LHC, allowing for more accurate analyses of hadronic final states.

2.3.4.1 Jet Trimming

Jet trimming, introduced in [50], is a technique designed to clean up fat jets by removing soft and wide-angle radiation. This method involves reclustering the constituents of a fat jet using a smaller radius parameter R and keeping only a subset of subjets that pass a certain transverse momentum (p_T) threshold:

$$p_{T,i} > f_{cut}\Lambda_{hard} \tag{2.23}$$

where f_{cut} is a fixed cut-off parameter chosen arbitrarily and Λ_{hard} is some hard scale chosen depending upon the kinematics of the event. The trimmed jet is formed by summing the remaining subjets that meet this criterion. The rationale is that radiation from hard interactions of interest should be concentrated in clusters, and reclustering with a smaller R helps isolate this radiation from unwanted contributions. While the k_T algorithm is commonly used for this purpose, the Cambridge/Aachen and anti-k_T algorithms can also be employed to recluster the constituents into subjets.

2.3.4.2 Jet Pruning

Another effective grooming method is jet pruning [51]. Jet pruning addresses the issue of soft, wide-angle emissions that can significantly affect the resultant jet mass and, consequently, the ability to extract meaningful physics from jet variables.

Pruning operates by first reclustering the jet constituents using the Cambridge/Aachen (CA) algorithm. The algorithm then iteratively unravels the clusters, forming a series of splittings. For a specific splitting $k \to ij$, two key parameters are computed:

$$z = \frac{\min(p_{T,i}, p_{T,j})}{p_{T,k}} \tag{2.24}$$

This parameter measures the relative softness of the emissions and is compared with an input value z_{cut}. Additionally, the angular separation ΔR_{ij} of the splitting is measured against a cut-off value:

$$\Delta R_{ij} > D_{cut} \tag{2.25}$$

For pairs of splittings ij that satisfy $z < z_{cut}$ and $\Delta R_{ij} > D_{cut}$, the softer particle (the one with lower p_T) is removed, and the process continues until a sufficiently hard splitting is found. This method helps in reducing the influence of soft and wide-angle radiation, leading to a cleaner and more accurate jet structure.

2.3.4.3 Softdrop

More recently, the pruning algorithm has been modified into the so-called Softdrop algorithm [52]. Softdrop follows a similar procedure to pruning but employs a different criterion for removing particles. The condition for retaining a pair of particles ij in Softdrop is that they satisfy:

$$\frac{\min(p_{T,i}, p_{T,j})}{p_{T,i} + p_{T,j}} > z_{cut} \left(\frac{\Delta R_{ij}}{R_0} \right)^{\beta}, \tag{2.26}$$

where the softer of i and j is removed if the condition is not met. In this equation, z_{cut} and β are input parameters that can be tuned to optimise Softdrop performance. This method allows for a flexible approach to grooming, where adjusting z_{cut} and β enables fine-tuning to retain more of the relevant jet substructure while removing unwanted soft and wide-angle radiation.

2.3.4.4 PU Mitigation Techniques

One of the widely used methods for mitigating pile-up is the Pile-Up Per Particle Identification (PUPPI) algorithm [49]. PUPPI capitalises on the ability to trace back the interaction vertex from which charged tracks originate, allowing the removal of charged tracks that come from vertices displaced from the primary interaction point of interest. However, since charged particles account for only about 60% of the emissions in a proton-proton (pp) interaction, it is also necessary to address the neutral radiation arising from PU events.

The PUPPI method starts by defining a shape parameter, α, to estimate the likelihood that a particle originates from a PU event. This parameter is calculated as:

$$\alpha_i = \log\left(\sum_{j\in\text{event}} \frac{p_{T,j}}{\Delta R_{ij}}\Theta(R_{\text{min}} \leq \Delta R_{ij} \leq R_0)\right), \tag{2.27}$$

where Θ is the Heaviside function, R_{min} is the minimum cut-off that governs the collinear splittings from i, and R_0 defines the cone surrounding the particle i. The parameter α can be plotted to identify particles originating from PU events.

The neutral PU particles usually exhibit similar patterns to the charged ones, allowing for effective distinction from particles of interest. This identification procedure significantly minimises the impact of PU events, enhancing the ability to isolate and study particles originating from the primary interaction.

Moreover, PUPPI incorporates information from both charged and neutral particles to create a more comprehensive picture of the event. By doing so, it enhances the accuracy of jet reconstruction and subsequent analyses. The algorithm uses local shape variables, such as α, to assign weights to each particle, effectively suppressing those likely to be from PU while retaining those from the primary interaction. This method not only cleans up the jets but also preserves the essential features needed for detailed physical analyses, making it a crucial tool in high-energy physics experiments.

Overall, PUPPI and similar PU mitigation techniques are vital for ensuring that analyses conducted at the LHC and other high-energy physics experiments are accurate and reliable, free from the distortions introduced by extraneous PU interactions.

2.3.5 Jet Structure and Substructure

2.3.5.1 Jet Shapes Variables

There are predictions that can be made for the distributions of jets from QCD. Event shape variables are an example of this. Comparisons between various MC event generation algorithms show the shape variables measured on jets to be relatively insensitive to the details of the MC simulation used [53]. Thus, these variables are good comparisons, even including the uncertainties of simulation. Here, 6 common shape variables are described;

- **Jet mass:** a jet's momentum is the combined 4-momentum of all reconstructed particles assigned to the jet. The invariant jet mass spectrum is simply the invariant mass of the momentum of one or more of the jets. It can be calculated for more than one jet in each event, or just of the highest p_T jet of each event.

- **Thrust:** is a description of how much of the jet momentum goes along the dominant axis. It is calculated as

$$T = \min_{n_t} \frac{2\sum_i p_i \cdot \hat{n}_t}{\sum_i |p_i|}, \tag{2.28}$$

where i sums over all jets, or sometimes only a subset of the jets, in the event. The factor of 2 being customary, but sometimes omitted.

- **Thrust major and minor:** measures of thrust in other directions. Thrust major is defined as

$$T_M = \min_{\hat{n}_M \in \hat{n}_t \cdot \hat{n}_M = 0} \frac{2\sum_i p_i \cdot \hat{n}_M}{\sum_i |p_i|}, \tag{2.29}$$

so it is the same as thrust, only its axis, $\hat{n}_M$, is required to be perpendicular to the thrust axis, $\hat{n}_t$. Thrust minor has its axis, $\hat{n}_m$ perpendicular to both $\hat{n}_t$ and $\hat{n}_M$. Its axis is $\hat{n}_m = \hat{n}_t \times \hat{n}_M$, thus no minimisation is needed. It is calculated as

$$T_m = \frac{2\sum_i p_i \cdot \hat{n}_m}{\sum_i |p_i|} \tag{2.30}$$

- **Oblateness:** this is a property also used in earth science to describe the shape of the earth. It is calculated from the thrust major and the thrust minor as [54]

$$O_b = T_M - T_m. \tag{2.31}$$

Conceptually, this measures how squished the event is.

- **Sphericity:** heuristically this can be seen as a measure of how far the event deviates from a spherical configuration. It is calculated by first constructing the momentum tensor;

$$S^{\alpha\beta} = \frac{\sum_i p_i^\alpha p_i^\beta}{\sum_i |p_i|^2} \tag{2.32}$$

where α, β are x, y or z, thus $S^{\alpha\beta}$ is a 3 by 3 tensor, and the sum over i sums over all momentum vectors of the jets (or some subset). By calculating the eigenvalues of the momentum tensor, $\lambda_1 \geq \lambda_2 \geq \lambda_3$, the event sphericity can be written as $S = \frac{3}{2}(\lambda_2 + \lambda_3)$ [55]. This is equivalent to calculating

$$S = \min_{\hat{n}_s} \frac{3\sum_i (p_i - p_i \hat{n}_s)^2}{2\sum_i (p_i)^2}. \tag{2.33}$$

The vector $\hat{n}_s$ is referred to as the sphericity axis [56].

- **Spherocity:** has a definition that at first glance appears similar to sphericity [56]

$$S' = \min_{\hat{n}_{s'}} \left(\frac{4\sum_i (p_i - p_i \hat{n}_{s'})}{\pi \sum_i p_i} \right)^2 \tag{2.34}$$

However, on closer inspection, it becomes clear that taking the square out of the sum means that jet momentum vectors going in opposite directions will cancel. This is now measuring how spherical and how balanced the event is.

These quantities are sensitive to IR behaviour, and so having IR safe jets is required to make predictions about their distributions.

2.3.5.2 N-Subjettiness

N-subjettiness [57] is a powerful and widely-used method for analysing the substructure of jets in high-energy physics. This parameter is designed to quantify how consistent a jet is with the hypothesis that it contains a certain number of subjets, offering insight into the jet's internal composition.

The N-subjettiness of a jet k, denoted as τ_{Nk}, is defined by the following measure:

$$\tau_{Nk} = \frac{1}{d_0} \sum_i p_{T,i} \min_{j \in \text{subjets of jet } k} (\Delta R_{ij}), \tag{2.35}$$

where the label j identifies the candidate subjets inside jet k for which τ_{Nk} is being computed, and the sum iterates over all constituent particles i within jet k. The restriction $j \in$ subjets of jet k ensures that the min function considers only those subjets associated with jet k. The $\min(\ldots)$ function selects the subjet j with the smallest angular distance ΔR to a given constituent particle i. The leading factor d_0 normalizes this measure and is given by:

$$d_0 = \sum_i p_{T,i} R, \tag{2.36}$$

where R is the characteristic radius parameter used during the jet clustering process.

Interpreting τ_{Nk} is straightforward: the value of τ_{Nk} indicates how much the jet k deviates from being composed of exactly N subjets. A smaller τ_{Nk} value implies that the jet is well-aligned with the hypothesis of having N subjets, meaning the jet's radiation is concentrated around these subjets. Conversely, a larger τ_{Nk} value suggests significant radiation exists away from the N candidate subjets, indicating the presence of additional substructure or more complex internal dynamics.

For example, if for a given jet k, we compute $\tau_{2k} \gg 0$, this indicates substantial radiation not aligned with the two candidate subjets, suggesting a more complex internal structure. On the other hand, if $\tau_{3k} \sim 0$, it implies that most of the radiation

within jet k is concentrated around the three candidate subjets, consistent with the jet having a three-pronged structure.

The ability to discern the internal structure of jets using N-subjettiness makes this method a crucial tool in particle physics, particularly for distinguishing between jets originating from different sources. For instance, quark and gluon jets typically exhibit different N-subjettiness distributions due to their differing radiation patterns. Moreover, jets originating from heavy particles, such as top quarks or W bosons, often exhibit specific substructure signatures that can be effectively captured and analyzed using N-subjettiness. This capability is especially valuable in searches for new physics, where distinguishing signal jets (like those from new particles) from background jets (such as those from QCD processes) is critical.

Overall, N-subjettiness provides a quantitative and interpretable means of probing the fine details of jet structure, making it an indispensable technique in the study and identification of jets in high-energy particle collisions.

Chapter 3

Machine Learning in High Energy Physics

The unprecedented scale and complexity of data generated by modern particle physics experiments, such as those at the LHC, have outpaced traditional analysis methods. Machine learning has emerged as a vital tool in this context, offering powerful solutions for event reconstruction, jet tagging, anomaly detection, and more. By leveraging ML techniques, particularly advanced models like Graph Neural Networks (GNNs), researchers can efficiently analyse vast datasets, uncovering patterns and insights that are crucial for advancing our understanding of high-energy physics.

3.1 The Big Challenge in HEP

Modern particle physics experiments, such as those conducted at the LHC at CERN, generate unprecedented amounts of data. The LHC, the world's largest and most powerful particle accelerator, can produce petabytes of data annually. Up to about 1 billion particle collisions can take place every second inside the LHC experiment's detectors. It is not possible to read out all of these events in real-time. Therefore, a 'trigger' system is employed to filter the data and select events that are potentially interesting for further analysis.

Even after the drastic data reduction performed by the experiments, the CERN Data Centre processed an average of one petabyte (one million gigabytes) of data per day during LHC Run 2. The LHC experiments plan to collect more data during LHC Run 3 than they did in the first two runs combined. This increase means that the computing challenge during Long Shutdown 2 was to prepare for storing and analysing more than 600 petabytes of data (600 million gigabytes), which is equivalent to over 20,000

years of 24/7 HD video recording [58]. This data encompasses billions of collision events, each containing intricate details of particle interactions and decay processes.

Traditional data analysis techniques, which have served the field of high energy physics well in the past, are now facing significant challenges due to the sheer volume and complexity of the data. Standard methods rely heavily on manual tuning and predefined criteria, which become increasingly difficult to manage as data grows in scale and complexity. The process of sifting through this massive dataset to identify significant events or rare phenomena is akin to finding a needle in a haystack.

Moreover, the complexity of the data itself poses another layer of difficulty. Each collision event can produce a multitude of particles, whose interactions need to be meticulously tracked and analysed. This task demands not only substantial computational resources but also sophisticated algorithms capable of discerning subtle patterns and correlations within the data. A classic example is the difficulty presented by pileup, where multiple proton-proton collisions occur in the same bunch crossing, making it challenging to disentangle individual particle tracks.

Given these challenges, there is a pressing need for innovative approaches that can handle large-scale data efficiently and effectively. Machine learning emerges as a powerful solution to this problem. Unlike traditional methods, machine learning algorithms are designed to learn from data, recognising patterns and making predictions without being explicitly programmed for each task. This capability makes ML particularly well-suited for analysing complex and high-dimensional data typical in HEP.

By leveraging machine learning, researchers can automate the analysis process, enabling them to uncover insights that would be difficult, if not impossible, to detect through conventional means. Machine learning models can be trained to identify specific signatures of interest, classify events, and even detect anomalies that may indicate new physics phenomena. This adaptability and efficiency make machine learning an indispensable tool in the modern particle physicist's toolkit.

In summary, the massive data output from contemporary particle accelerators presents a significant challenge to traditional analysis techniques. Machine learning offers a promising avenue to meet this challenge, providing the means to extract meaningful insights from the data deluge and pushing the boundaries of our understanding in high energy physics.

3.2 Machine Learning as a Solution

Machine learning is a subset of artificial intelligence focused on developing algorithms that allow computers to learn from and make predictions based on data.

Unlike traditional programming, where explicit instructions are provided to achieve a specific outcome, machine learning algorithms identify patterns and relationships within data autonomously. This capability to learn and adapt without being explicitly programmed for each task makes ML particularly powerful for handling large and complex datasets.

In the context of high energy physics, machine learning has become an invaluable tool for analysing the massive amounts of data generated by particle accelerators. Machine learning algorithms can be trained to recognise complex patterns in particle physics data that might be difficult, if not impossible, for humans to discern. For instance, identifying rare particle interactions or distinguishing between different types of particles based on their decay signatures requires analysing a vast number of variables simultaneously. ML excels at such high-dimensional data analysis, making it an ideal solution for these challenges.

Machine learning techniques can be broadly categorised into supervised and unsupervised learning, each offering unique advantages for different types of problems in HEP.

Supervised Learning

Supervised learning involves training a model on a labelled dataset, where the correct output (label) is provided for each input example. The model learns to map inputs to outputs by minimising the difference between its predictions and the actual labels. In HEP, supervised learning is often used for tasks such as classification and regression. For example, classifiers can be trained to identify whether a collision event corresponds to a specific particle type or to predict certain properties of particles based on their observed behaviours.

Example Application:

- Jet tagging, where jets resulting from particle collisions are classified as originating from quarks, gluons, or other particles, can benefit from supervised learning. By training on a labelled dataset of jets, the model can learn the distinguishing features of each type, improving the accuracy of event classification. We will talk more about this in Chapter 5.
- Jet Clustering, where events, which are the results of particle collision will be grouped into jet clusters, identifying and removing the background noise from the jets. We will talk more about this in Chapter 4.

Unsupervised Learning

Unsupervised learning, on the other hand, deals with unlabelled data. The goal is to uncover hidden patterns or structures within the data without prior knowledge of the

outcomes. This approach is particularly useful for exploratory data analysis and anomaly detection in HEP. By clustering similar data points or identifying outliers, unsupervised learning algorithms can reveal new insights and potentially discover unknown physics phenomena.

Example Application:

- Anomaly detection in collision events can be approached using unsupervised learning. By identifying events that significantly deviate from the norm, researchers can pinpoint rare or unexpected occurrences that might indicate new physics beyond the Standard Model.
- Jet Tagging, same example as in the supervised method, but this time the we leverage symmetries to learn about different types of jets, Chapter 5.

3.3 Applications of Machine Learning in HEP

Machine learning has become an indispensable tool in high energy physics, offering novel solutions to a wide range of challenges. From event reconstruction to anomaly detection, ML algorithms are transforming how physicists analyse and interpret vast amounts of data. In this section, we explore several key applications of machine learning in HEP: event reconstruction, jet classification, anomaly detection, and simulation and data generation.

3.3.1 Event Reconstruction

Event reconstruction involves interpreting the raw data collected from particle detectors to infer the properties and trajectories of particles produced in collisions. Traditional methods rely on deterministic algorithms and manual tuning, which can be limited in handling the complexity and volume of data produced by modern experiments. Although these methods do not require any training, which can be an advantage, they are limited in that they cannot extrapolate hidden features or symmetries from the data.

In Chapter 4, we will discuss a novel ML method used for reconstructing events, specifically clustering the final state particles into jets. The main advantage of this new method, spectral clustering, is that it projects the particles into a new space by considering their distances in the collider. In this new embedding space, a different metric can be used to infer the proximity of particles that belong to the same cluster. Although it is a relatively slow algorithm, it achieves impressive results, outperforming the most robust and popular clustering methods. Unlike most popular

machine learning models, especially deep learning models, it is not a black box, allowing for further studies to better understand the clustering process.

If we are willing to trust a black box, other ML algorithms, particularly deep learning models, can achieve extremely high performance. These models are adept at handling the high-dimensional and noisy data typical in event reconstruction and are capable of extracting hidden features from the data. The advantage of these methods is that they can be trained on huge amounts of data, from which they can infer and learn hidden or not immediately visible features, helping the model resolve events and cluster particles effectively.

Various models can be used depending on how the event data is represented, as discussed in Section 3.4. Some of the most popular include Graph Neural Networks [59–61], which are particularly useful because the particles in the event can be naturally described as a graph if connections are made between them, for example, based on their distances. These models are extremely helpful as they can handle events of different sizes, with each event containing a different number of particles. Other methods include Convolutional Neural Networks, which are useful if we want to represent the data as an image [62].

3.3.2 Jet Classification

Jet classification, or jet tagging, is the process of identifying the origin of jets produced in high-energy collisions. Jets can originate from various particles, such as quarks, gluons, or more complex objects like top quarks and Higgs bosons. Accurate classification is crucial for understanding the underlying physics of collision events.

For this specific task, deep learning methods are by far the most powerful and reliable. The concept is as simple as training a neural network to distinguish an image of a dog from an image of a cat. In our case, instead of presenting millions of animal images, the dataset comprises millions of jets. Typically, models are initially trained for binary classification, such as distinguishing W boson (or top quark) jets from QCD jets. Once again, Graph Neural Networks are state-of-the-art for solving classification problems, as we will explore later in this chapter. Before representing jets as graphs became popular, several studies focused on representing jets as images, which is why many Convolutional Neural Networks perform exceptionally well for this task [63; 64].

With the recent introduction of one of the most effective architectures ever invented—transformers—the current best-performing model is ParT [65]. This complex model achieves impressive results in classifying several classes. These remarkable results have been achieved not only due to the powerful architecture itself but also thanks to the incredibly large datasets used and the resources required to train such a model over a long period, which necessitates many GPUs.

3.3.3 Anomaly Detection

Anomaly detection in high-energy physics is a crucial technique aimed at identifying rare or unexpected events in particle collision data that deviate from known physics processes. This approach is particularly valuable for discovering potential new particles or interactions without relying on specific theoretical models. However, anomaly detection in HEP faces significant challenges, including the need to process vast amounts of high-dimensional data, the rarity of potential anomalous events, and the complexity of background processes. Machine learning techniques have emerged as powerful tools to address these challenges, offering the ability to analyse complex data patterns, handle high-dimensional spaces, and identify subtle deviations from expected behaviours. [66] provides a comprehensive review of different approaches to anomaly detection.

3.3.4 Anomaly Matching

Anomaly matching is a technique in HEP that builds upon anomaly detection methods to not only identify unusual events but also to categorise and match them to potential new physics scenarios. The primary challenge in anomaly matching lies in developing algorithms that can effectively search the parameter space of theoretical models to fit the model to the anomaly. Machine learning techniques are becoming increasingly popular as they are well-suited to address these challenges due to their ability to learn complex patterns and relationships in high-dimensional data. [67] provides a comprehensive review of the machine learning techniques used in anomaly matching.

An approach we worked on leverages machine learning based on a multi-objective active search method called b-CASTOR [68] (an adaptation of the constraint active search [69]), which achieves high sample efficiency and diversity due to the use of probabilistic surrogate models and a volume-based search policy, outperforming competing algorithms, such as those based on Markov-Chain Monte Carlo (MCMC) methods [70; 71]. Additionally, a remarkable result is the use of Gaussian Processes [72] to create a surrogate model, which avoids the repeated use of physics software to evaluate data points, significantly speeding up the search and providing data with uncertainty.

3.3.5 Simulation and Data Generation

Simulation is an essential part of high-energy physics, providing synthetic data crucial for experiment planning, detector design, and hypothesis testing. Traditional simulation methods, such as those employed in well-established software packages

like Pythia [45; 73] and Herwig [47], use Monte Carlo methods [74] to model the complex processes occurring in particle collisions. These simulations are highly detailed and accurate, allowing researchers to predict the outcomes of various high-energy interactions. However, they are computationally intensive and can become a bottleneck in large-scale studies, especially when vast amounts of data are required for comprehensive analyses.

Pythia and Herwig are among the most widely used tools in the HEP community for event generation. They simulate the physics of particle collisions, including the parton shower, hadronisation, and decay processes, based on well-understood theoretical models. These programs rely heavily on Monte Carlo simulations, a method that uses random sampling to simulate the probabilistic nature of particle interactions. While these traditional methods are robust and have been indispensable in the field, they require significant computational resources, particularly when generating large datasets or running simulations for complex event topologies.

Given these challenges, machine learning methods have emerged as powerful tools for data generation. Generative models, such as Generative Adversarial Networks (GANs) [75] and Variational Autoencoders (VAEs) [76; 77], offer a promising alternative to traditional simulation techniques. These models can learn from existing datasets and generate high-fidelity synthetic data that closely mimics the properties of actual collision events. The main advantage of using ML-based generative models is their ability to produce large-scale datasets more efficiently, reducing the computational load and accelerating the simulation process.

For example, [78] demonstrated the potential of GANs in generating realistic jet images, offering a complementary approach to traditional methods. Similarly, [79] explored the use of VAEs. These models not only speed up data generation but also introduce new possibilities for creating diverse datasets that can be used for training and validating other ML models in HEP. Moreover, ML-generated data can be fine-tuned to explore rare scenarios.

3.4 Representations of Jets

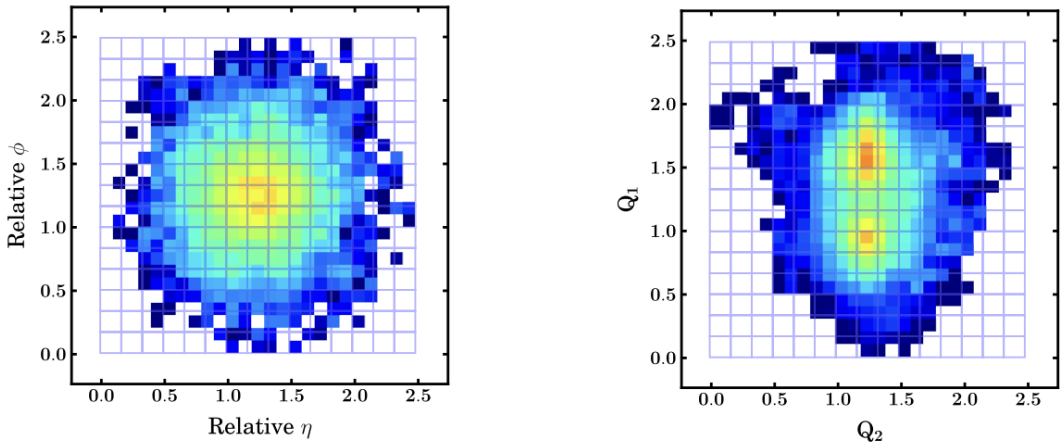
Jets, collimated streams of particles resulting from high-energy processes, are fundamental objects of study in high energy physics. Accurately representing jets is crucial for analysing their properties and behaviours. Various representations have been developed to capture the complex structure of jets, each with its advantages and challenges. Below, we explore four common representations: jet images, point clouds, trees, and graphs.

3.4.1 Jets Images

Jet images represent jets as two-dimensional histograms or images, where the pixel intensities correspond to the energy deposits of particles within a detector. This approach leverages techniques from computer vision, making it possible to apply convolutional neural networks (CNNs) [80] for jet classification and analysis.

- Advantages: Jet images allow for the application of mature image processing techniques and deep learning models developed for image recognition tasks. This can lead to high accuracy in tasks like jet tagging.
- Challenges: One limitation of jet images is the potential loss of information due to discretisation when converting continuous particle data into pixel values. Additionally, the spatial resolution is limited by the pixel size.

Jet images were first introduced in [3] and have since been utilised and refined by subsequent studies [81–83]. The initial step in creating jet images is to approximate the calorimeter as a single-layered grid. For example, cells of $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ spanning $[-2.5, 2.5]$ in η and $[0, 2\pi]$ in ϕ can be used to show the amount of transverse momentum (p_T) carried by the particle position in each cell.



(A) (B)

FIGURE 3.1: Image [3] of the average jet with a transverse momentum (p_T) of approximately 200 GeV, shown (A) before pre-processing and (B) after pre-processing, where the new coordinate system ($Q1 - Q2$) is introduced following the rotation.

Pre-processing is paramount to accurate jet image classification. In addition to standard physics pre-processing steps, several image-specific augmentations can enhance model performance. These augmentations help the CNN recognise symmetries during training. Typical augmentations include:

- **Rotation**: Rotation is performed to account for the stochastic nature of the decay angle relative to the $\eta - \phi$ coordinate system.
- **Translation**: The jet image is translated such that the energy centroid or leading subjet is centred in the same pixel.
- **Reflection**: The image is reflected over the vertical axis, ensuring that the side of the image with maximal transverse energy always appears on the right side.

These augmentation techniques are illustrated in Figure 3.1.

3.4.2 Point Clouds

Point clouds represent jets as a collection of points in space, where each point corresponds to a detected particle with attributes such as momentum, energy, and spatial coordinates. This representation maintains the continuous nature of particle data and can be processed using techniques from 3D data analysis.

- Advantages: Point clouds preserve detailed information about each particle, allowing for more precise analysis of jet substructure. Furthermore, point cloud methods can handle the unordered nature of particle sets, ensuring that the order of the particles does not affect the analysis.
- Challenges: Handling variable-sized point clouds and ensuring efficient processing can be computationally demanding. Moreover, processing large point clouds requires significant computational resources. Finally, developing robust algorithms to process point clouds requires addressing issues related to permutation invariance and local neighbourhood structures.

Jets can be effectively analysed as point clouds, capturing the full kinematic and spatial details of particles. The paper [21] introduces a novel approach to processing and learning from collider events using point clouds. This method respects the variable-length and unordered nature of particle sets, addressing the limitations of traditional neural network architectures.

In their work, they introduce Energy Flow Networks (EFNs) and Particle Flow Networks (PFNs). EFNs specifically target infrared and collinear safety by incorporating particle energy or transverse momentum as weights in the summation over particles. PFNs, on the other hand, provide a more general framework that allows for the inclusion of various particle-level features such as charge and flavour.

To create input data as point clouds, each particle in a jet is represented by its features, such as transverse momentum, rapidity, and azimuthal angle. These features form a

multidimensional point for each particle, which collectively constitute the point cloud of the jet. The network architecture processes these point clouds by mapping each particle to an internal latent space and summing these representations to form an overall event representation. This approach preserves the permutation invariance and variable length properties of the particle sets, making it well-suited for jet physics applications.

3.4.3 Trees

Tree representations of jets model the hierarchical nature of particle showers. In this approach, jets are represented as binary or multi-branch trees, where each node represents a particle, and branches represent the splitting processes during the jet formation.

- Advantages: Tree structures naturally align with the physical process of jet formation, making them intuitive for modelling particle decays and splittings. Recursive neural networks and tree-based models can exploit this hierarchical information effectively.
- Challenges: Constructing accurate tree representations requires detailed tracking of particle interactions and decay processes. Ensuring that the tree structure faithfully represents the underlying physics can be complex, especially in dense jet environments.

Tree representations of jets are typically constructed by re-clustering the jet constituents using classical clustering algorithms. These algorithms, which are greedy in nature, merge pairs of particles iteratively until a complete tree structure is formed. Here is an outline of the process:

- Initial Jet Constituents: The starting point is the set of particles that form the jet, each characterised by its four-momentum.
- Clustering Algorithm: Algorithms such as k_T, Cambridge/Aachen, or anti-k_T are applied. These algorithms determine which pairs of particles to merge based on distance metrics that favour the merging of nearby particles.
- Binary/Multi-Branch Tree Formation: At each step of the clustering process, the closest pair of particles is merged into a single pseudo-particle. This process is repeated iteratively, creating a binary or multi-branch tree that traces back the merging steps.

- Tree Nodes and Branches: Each node in the tree represents a particle or pseudo-particle, and each branch represents the merging of two particles, reflecting the hierarchical nature of the particle shower.

Tree-based representations have been used in various studies to enhance jet analysis. One notable example [84] uses recursive neural networks (RNNs) [85] are applied to tree structures. This study demonstrates that different tree topologies can lead to different performances in jet tagging tasks. The hierarchical structure of the tree allows the RNN to capture complex dependencies and patterns in the jet substructure, improving the accuracy of particle identification and classification.

Another innovative application is LundNet [86], which employs graph neural networks to augment tree representations. LundNet embeds the tree into the Lund Plane, a theoretical framework that organises the splittings in a jet according to their scales. By augmenting the tree with additional features in this embedding, LundNet effectively captures the intricate details of the jet formation process. More details on this approach will be discussed in Section 3.5.3.

In summary, tree representations provide a powerful and intuitive way to model the hierarchical nature of jets, and when combined with advanced machine learning algorithms, they offer significant improvements in jet analysis and classification tasks.

3.4.4 Graphs

Graph representations of jets capture the relational structure between particles by modelling them as nodes connected by edges. Each node represents a particle, and edges encode the relationships (e.g., proximity, kinematic correlations) between particles.

- Advantages: Graphs provide a flexible and powerful way to model complex dependencies and interactions between particles. Graph neural networks can be employed to learn from these relational structures, offering state-of-the-art performance in various jet analysis tasks.
- Challenges: Designing effective graph architectures and determining appropriate edge connections are non-trivial tasks. Computational efficiency and scalability can also be concerns when dealing with large and complex graphs.

Graph representations of jets can be constructed by defining particles as nodes and their relationships as edges. One common method to establish these relationships is by using the k-nearest neighbours (kNN) algorithm in the $\eta - \phi$ space, as shown in Figure 3.2.

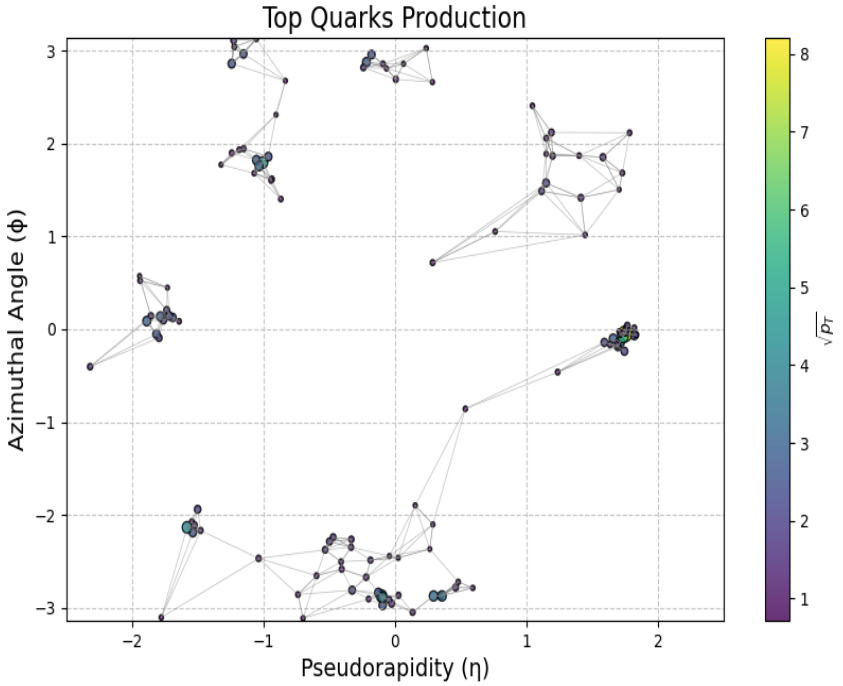


FIGURE 3.2: An example of a graph representation of a particle event, where particles are nodes and the edges represent relationships established using the k-nearest neighbours algorithm in the $\eta - \phi$ space.

- Initial Jet Constituents: Start with a set of particles that form the jet, each characterised by their kinematic properties (e.g., transverse momentum p_T, pseudorapidity η, and azimuthal angle ϕ).
- Defining Nodes and Edges:
  - Nodes: Each particle in the jet is represented as a node in the graph.
  - Edges: Use the kNN algorithm to determine the edges. For each node, identify the k nearest particles in the $\eta - \phi$ space, creating edges between them. The distance metric typically used is $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$, where $\Delta\eta$ and $\Delta\phi$ are the differences in pseudorapidity and azimuthal angle, respectively.
- Edge Features: The edges can also be augmented with features such as the distance ΔR or kinematic correlations between the connected nodes, providing additional information to the GNN.

Graph-based representations can be applied both for clustering the entire event or for tagging individual jets. When used for clustering the entire event, graphs can model the interactions and relationships between all particles in the event. This holistic approach allows for comprehensive analysis of the event structure, aiding in tasks

such as event classification and anomaly detection. For jet tagging, graphs are constructed for individual jets, focusing on the relationships between particles within the jet. This localised graph structure enables precise identification of the jet's origin and characteristics. GNNs can exploit these relational structures to enhance the performance of jet tagging algorithms. Graphs will be discussed in more details in Section 3.5.

3.5 Focus on Graph Neural Networks (GNNs)

3.5.1 Definition of a Graph

A graph G is formally defined as an ordered pair (V, E) where:

- **V:** A finite, non-empty set of vertices (also called nodes or points). This set represents the objects or entities within the system being modelled. We denote the cardinality of this set as $|V|$, representing the number of vertices in the graph.
- **E:** A set of edges (also called links or arcs), where each edge is a connection between two vertices in V. Mathematically, an edge is typically represented as an unordered pair u, v, where u and v are distinct vertices in V. The cardinality of this set, $|E|$, indicates the number of edges in the graph.

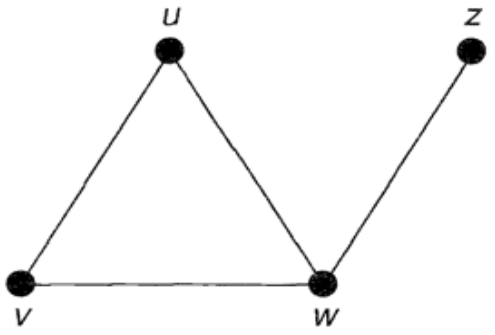


FIGURE 3.3: A simple example of a graph.

In Figure 3.3 we can see an example of a simple graph G whose vertex set $V(G)$ is u, v, w, z, and whose edge set $E(G)$ consists of the edges uv, uw, vw and wz.

Types of Graphs

Graphs can be classified based on the properties of their edges:

- **Undirected Graph:** Edges have no direction, meaning the edge u,v is equivalent to v,u. The relationship between connected vertices is symmetric.

- **Directed Graph (Digraph):** Edges have a specific direction. The edge (u,v) indicates a connection from vertex u to vertex v, but not the reverse. The relationship is asymmetric.
- **Weighted Graph:** Each edge is assigned a numerical value or weight, representing the strength, cost, or some other property of the relationship between the vertices it connects.
- **Simple Graph:** An undirected graph with no self-loops (edges connecting a vertex to itself) and no multiple edges between the same pair of vertices.

Basic Graph Notation

Before continuing, just some basic notation for simplifying further reading:

- **Adjacency:** we say that two vertices u and v are adjacent if there is an edge connecting them
- **Neighbourhood:** The neighbourhood of a vertex v is the set of all vertices adjacent to v.
- **Degree:** In an undirected graph, the degree of a vertex is the number of edges incident to it (i.e., the number of edges connected to it). In a directed graph, we distinguish between in-degree (number of incoming edges) and out-degree (number of outgoing edges).

Graphs as a Universal Representation

Graphs offer a versatile framework for modelling complex systems, enabling precise analysis and algorithm development for tasks like clustering, classification, and prediction. Interestingly, almost anything can be represented as a graph:

- **Images**: Pixels as nodes, edges connecting neighbouring pixels based on proximity or similarity.
- **Text**: Characters, words, or tokens as nodes, edges connecting them based on sequence or co-occurrence.
- **Molecules**: Atoms as nodes, bonds as edges, potentially weighted by bond strength.
- **Particle Jets**: Particles as nodes, edges representing interactions or proximity.

In the following chapters, we'll delve deeper into the power of graph representations, especially when it comes to analysing the complex structure of particle jets

3.5.2 Graph Neural Networks

A Graph Neural Network is a machine learning model designed to operate on data structured as graphs. Graphs consist of nodes (entities) connected by edges (relationships), making them a natural way to represent many real-world datasets, from social networks to molecular structures.

At its core, a GNN takes two primary inputs:

- **Node Embedding (X):** These are vector representations of the nodes, capturing their individual features or attributes.
- **Adjacency Matrix (A):** This matrix encodes the connectivity of the graph, indicating which nodes are linked by edges.

The GNN then processes these inputs through a series of layers, typically referred to as convolutional layers in the context of Graph Convolutional Networks (GCNs). Each layer updates the node embedding to incorporate information from the surrounding neighbourhood within the graph. This process continues iteratively, producing intermediate "hidden" representations (H_i) at each layer. The final layer outputs the final node embedding (H_K), which now encapsulate information about each node and its context within the broader graph structure.

We can express each layer of a GCN as a function $F[\cdot]$ with parameters Φ:

$$\begin{aligned} H_1 &= F[X, A, \phi_0] \\ H_2 &= F[H_1, A, \phi_1] \\ &\vdots \\ H_K &= F[H_{K-1}, A, \phi_{K-1}] \end{aligned}$$

where X is the input, A the adjacency matrix, H_k contains the modified node embedding at the k^{th} layer, and ϕ_k denotes the parameters that map from layer k to layer $k+1$.

Permutation Equivariance

A fundamental property of graphs is that the indexing of nodes is arbitrary. The structure remains the same regardless of how we order the nodes. Therefore, it's crucial that GNNs respect this permutation invariance.

In other words, each layer of the GNN must be permutation equivariant. This means that if we permute (reorder) the nodes in the graph, the resulting node embedding at each layer will be permuted in the same way. This ensures that the GNN's output is

consistent with the underlying graph structure and does not depend on arbitrary node ordering.

Parameter Sharing

Fully connected networks applied to images can be inefficient, as they require learning object recognition at each image position independently. Convolutional layers offer a more suitable approach by processing every image position uniformly. This reduces the number of parameters and imposes a bias that enforces consistent treatment of different image regions.

A similar rationale applies to nodes within a graph. Learning with separate parameters for each node would require the network to independently grasp the meaning of connections at every position, necessitating extensive training on graphs with identical structures. Instead, constructing a model with shared parameters for every node not only reduces the parameter count but also enables knowledge sharing across the entire graph.

Convolution operations involve updating a variable through a weighted sum of neighbouring information. This can be interpreted as each neighbour sending a message, which is then aggregated to form an update. While images involve fixed-size square regions around each pixel, graphs present varying neighbour counts and no consistent spatial relationships. Therefore, unlike images, there's no inherent notion of weighting information from one node differently based on its relative position to another.

Tasks

Supervised learning tasks on graph data generally fall into three distinct categories, each with unique goals and applications within the field of jet physics, Figure 3.4.

- **Graph-Level Tasks:** In these tasks, the goal is to predict a property or label for the entire jet, treating it as a single entity. The model considers both the jet's internal structure (how its constituent particles are connected) and the features of these particles to make a holistic prediction.
  Example: Jet Tagging. A classic example of a graph-level task is jet tagging. Here, the objective is to determine the origin of a jet, distinguishing between jets originating from specific particles (e.g., W bosons, top quarks, Higgs bosons) and those arising from background processes like QCD. The model analyses the relationships between particles within the jet, as well as their individual properties (such as energy and momentum), to assign a probability for each possible jet origin.

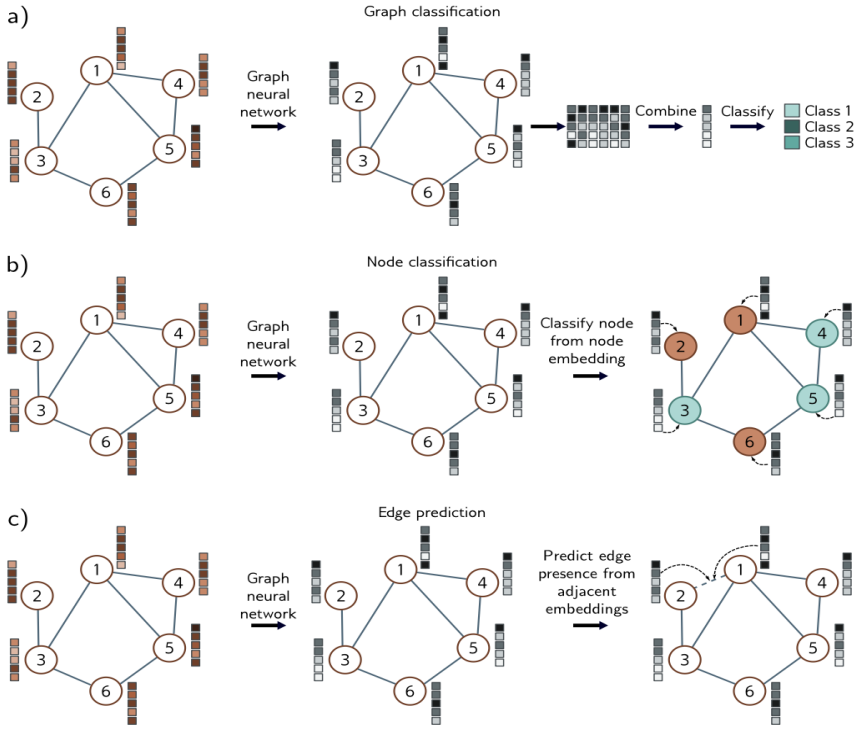


FIGURE 3.4: Common tasks for graphs. In each case, the input is a graph represented by its adjacency matrix and node embedding. The graph neural network processes the node embedding by passing them through a series of layers. The node embedding at the last layer contain information about both the node and its context in the graph. a) Graph classification. The node embedding are combined (e.g., by averaging) and then mapped to a fixed-size vector that is passed through a softmax function to produce class probabilities. b) Node classification. Each node embedding is used individually as the basis for classification (cyan and orange colours represent assigned node classes). c) Edge prediction. Node embedding adjacent to the edge are combined (e.g., by taking the dot product) to compute a single number that is mapped via a sigmoid function to produce a probability that a missing edge should be present.

- **Node-Level Tasks:** Node-level tasks focus on making predictions about the individual particles within a jet. This involves assigning a label or value to each particle based on its features and its interactions with other particles in the jet. **Example: Particle Identification and Clustering.** In jet physics, node-level tasks are commonly used for particle identification. By examining a particle's properties and its role in the jet's structure, a GNN can classify it as a quark, gluon, or other fundamental particle. Similarly, node-level predictions can be employed for clustering particles within a jet, grouping them based on their shared characteristics or relationships.

- **Edge Prediction Tasks:** Edge prediction tasks focus on the relationships between particles within a jet. The goal is to predict whether a connection (edge) exists between two particles or to estimate the strength of an existing connection. **Example: Jet Substructure Analysis.** Edge prediction can be applied to analyze the substructure of jets. By predicting which particles are most likely to be

directly related through the underlying physics processes, we can gain insights into the jet's origin and the dynamics of the particle collision that produced it. Additionally, the strength of predicted edges can offer valuable information about the hierarchical structure within the jet.

3.5.3 Jet Taggers with the use of GNNs.

3.5.3.1 ParticleNet

ParticleNet, a pioneering approach introduced in [87], has emerged as a leading architecture in the realm of Graph Convolutional Neural Networks for jet tagging tasks in high-energy physics. Its unique strength lies in its innovative representation of jets as "particle clouds," unordered sets of particles analogous to point clouds in computer vision. This representation intrinsically respects the permutation symmetry inherent in particle physics data, a crucial aspect often overlooked by traditional methods.

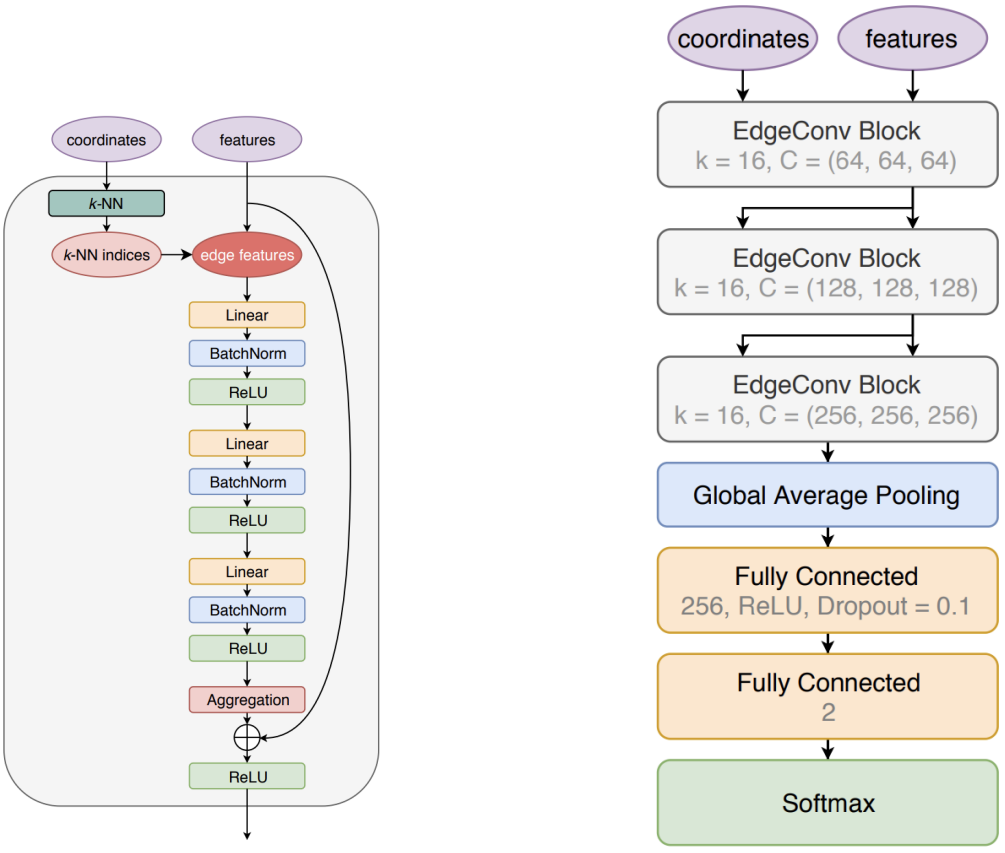
A key innovation of ParticleNet is its integration of Dynamic Graph CNN (DGCNN) [88], where the graph structure is dynamically updated after each EdgeConv block. This continuous refinement of particle neighbourhoods enhances the model's ability to capture evolving relationships between particles as it learns higher-level representations.

At the heart of ParticleNet's architecture is the Edge Convolution (EdgeConv) operation. This operation is designed to learn on graphs where nodes represent particles and edges connect each particle to its k nearest neighbours. By learning a shared function that transforms the features of each particle and its neighbours, EdgeConv ensures permutation invariance while effectively capturing local particle interactions. The EdgeConv operation for each node $x_i^{(l)}$, at step (l) for its update at step $(l+1)$ has the form:

$$x_i^{(l+1)} = \max_{j \in \mathcal{N}(i)} (\Theta \cdot (x_j^{(l)} - x_i^{(l)}) + \Phi \cdot x_i^{(l)}) \tag{3.1}$$

where $\mathcal{N}(i)$ is the neighbour of x_i, and Θ and Φ are linear layers. In Figure 3.5 we can see how an EdgeConv can be constructed explicitly. In this case it consists of three layers for this shared MLP, each consisting of a linear layer followed by a batch normalisation (BN) [89] and a ReLU activation [90]. Then, an aggregation step is performed for the node by taking an element-wise average of the learned edge features of all the incoming edges. A shortcut connection [91] is also added to take the original node features into account directly, and the node feature is then updated to the new value.

ParticleNet employs multiple EdgeConv blocks, Figure 3.5, in a hierarchical manner, enabling the network to learn features at various scales. The initial block utilises spatial coordinates (pseudo-rapidity and azimuth) to define particle neighbourhoods, while subsequent blocks leverage learned features as coordinates. This dynamic approach allows the network to adapt its understanding of particle relationships as it progresses through deeper layers, capturing both local and global jet features. After a few EdgeConv blocks, a global average pooling is applied to read out information from all nodes in the graph. This is followed by a fully connected layer before the final classification output.



(A) (B)

FIGURE 3.5: (A) The structure of the EdgeConv block. (B) The architecture of ParticleNet on the right.

It is worth mention that the input data for this model consists of only the kinematic information, i.e. the 4-momentum (p_x, p_y, p_z, E) , of each constituent of the jets. Based on these information, the features are enlarged with other variables derived from the 4-momentum, which are listed in Table 3.1. The $(\Delta\eta, \Delta\phi)$ variables are used as coordinates to compute the distances between particles in the first EdgeConv block. They are also used together with the other five variables, $logp_T$, $logE$, $log\frac{p_T}{p_T^{(jet)}}$, $log\frac{E}{E^{(jet)}}$ and ΔR, to form the input feature vector for each particle.

Variable	Definition
$\Delta\eta$	difference in pseudo-rapidity between the particle and the jet axis
$\Delta\phi$	difference in azimuthal angle between the particle and the jet axis
$\log p_T$	logarithm of the particle's p_T
$\log E$	logarithm of the particle's energy
$\log \frac{p_T}{p_T^{(\text{jet})}}$	logarithm of the particle's p_T relative to the jet p_T
$\log \frac{E}{E^{(\text{jet})}}$	logarithm of the particle's energy relative to the jet energy
ΔR	angular separation between the particle and the jet axis $\left(\sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}\right)$
q	electric charge of the particle
`isElectron`	if the particle is an electron
`isMuon`	if the particle is a muon
`isChargedHadron`	if the particle is a charged hadron
`isNeutralHadron`	if the particle is a neutral hadron
`isPhoton`	if the particle is a photon

TABLE 3.1: Variables used by ParticleNet for jet tagging, including kinematic and particle identification features (these only used for the quark-gluon tagging task.

ParticleNet's efficacy has been demonstrated through its state-of-the-art performance on benchmark jet tagging tasks, including the identification of top quarks and the discrimination between quark and gluon jets. In the top tagging task, ParticleNet significantly outperforms previous methods, achieving a background rejection rate at 30% signal efficiency that is roughly 1.8 and 2.1 times better than the Particle Flow Network (PFN) [21] and P-CNN models [92], respectively. It even surpasses the performance of the deep ResNeXt-50 model [91], which is considerably more complex.

Furthermore, ParticleNet's lightweight design, with significantly fewer trainable parameters than ResNeXt-50, makes it computationally efficient and well-suited for real-time applications in high-energy physics experiments. Its success has not gone unnoticed, as evidenced by its adoption by the CMS collaboration for jet tagging tasks.

Beyond jet tagging, ParticleNet's particle cloud representation and dynamic graph approach hold promise for a wide range of applications in particle physics. Its ability to model complex interactions and capture intricate patterns in data could prove invaluable in areas such as pileup identification, jet grooming, and jet energy calibration. As research in this field progresses, ParticleNet is poised to remain a key player in unlocking the potential of machine learning for unravelling the mysteries of the universe.

3.5.3.2 LundNet

LundNet, a graph neural network introduced in [93], is designed for jet tagging tasks in high-energy physics experiments. It leverages the Lund plane representation of jets [86], which maps the radiation patterns within a jet onto a two-dimensional plane. In

this representation, each node in the graph corresponds to a Lund declustering, a step in the jet clustering algorithm that splits a jet into two subjets. The nodes carry kinematic information about the splittings, such as the transverse momentum of the softer subjet and the angle between the two subjets.

The construction of the Lund plane [86] representation begins by reclustering the jet's constituents using the Cambridge/Aachen algorithm, Figure 3.6. This algorithm sequentially merges pairs of particles that are closest in angle, starting with the two closest and iterating until all particles are combined into a single jet. Each step of the reclustering process corresponds to a node in the Lund plane tree.

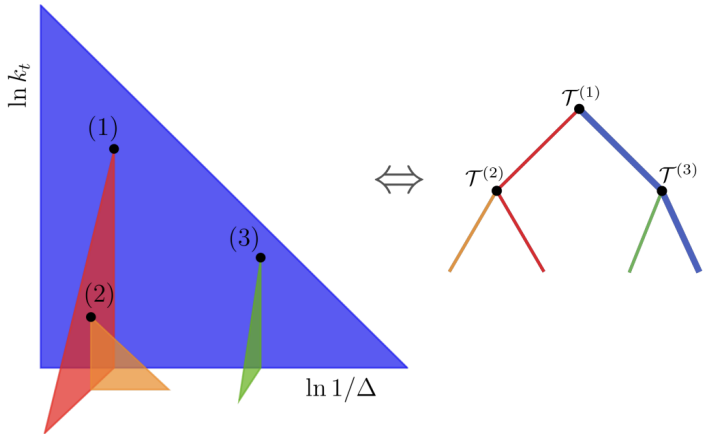


FIGURE 3.6: The Lund plane representation of a jet (left) where each emission is positioned according to its Δ and k_T coordinates, and the corresponding mapping to a binary Lund tree of tuples (right).

The kinematic information associated with each node, denoted as tuple $T^{(i)}$, is defined as follows:

$$T^{(i)} = \{k_T, \Delta, z, m, \psi\} \tag{3.2}$$

where:

- $k_T = p_{T,b}\Delta$ is the transverse momentum of the softer subjet (b) with respect to its emitter (a+b) in the collinear limit,
- Δ is the angular distance between the two subjets in the rapidity-azimuth plane,
- $z = \frac{p_{T,b}}{p_{T,a}+p_{T,b}}$ is the momentum fraction of the softer subjet,
- m is the invariant mass of the two subjets,
- $\Psi = \tan^{-1}(\frac{y_b - y_a}{\phi_b - \phi_a})$ is the azimuthal angle around the harder subjet's axis.

LundNet utilises the EdgeConv operation, introduced in Section 3.5.3.1, to process the information in the Lund plane graph. EdgeConv operates on each node and its neighbours, transforming their features into a new set of features. This process is

repeated through multiple layers, allowing the network to learn complex patterns in the jet's radiation. In Figure 3.7 we can see how this translates to a tree structure.

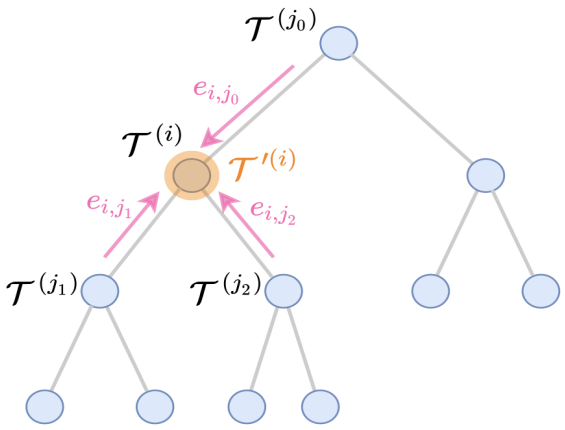


FIGURE 3.7: Illustration of the EdgeConv operation on a node of the Lund tree.

The architecture of LundNet consists of six EdgeConv blocks, followed by a global average pooling layer and fully connected layers for classification. The input features for each node are the kinematic variables associated with the Lund declustering, such as the transverse momentum, angle, and mass of the subjets, the one described in Equation 3.2

LundNet has been shown to outperform other state-of-the-art jet taggers, particularly in identifying top quark jets. This is attributed to its ability to effectively capture the complex radiation patterns within jets, which are crucial for distinguishing between different types of particles. The model's performance is further enhanced by its use of the full Lund plane, including both primary and secondary emissions, which provides a more complete picture of the jet's substructure.

In addition to its superior performance, LundNet is also computationally efficient, requiring significantly less training time than other GNN-based taggers. This is due to the efficient representation of jets in the Lund plane and the use of the EdgeConv operation, which avoids the need for computationally expensive nearest-neighbour searches. If we compare the LundNet model to ParticleNet, although the architecture is quite similar (they roughly have the same number of parameters, just less than 400 thousand), the training time of LundNet is 0.472 [ms/sample/epoch], only 15% of the training time of ParticleNet.

3.5.3.3 The Energy-Weighted Message Passing Neural Network (EMPN).

The Energy-weighted Message Passing Neural Network (EMPN) [4] is a graph neural network architecture designed for jet tagging tasks in high-energy physics. Its key

innovation lies in its inherent infrared and collinear (IRC) safety, a crucial property for theoretical consistency in perturbative QCD calculations.

In the context of jet physics, IRC safety means that the network's output should remain unchanged under soft (low-energy) or collinear (small-angle) emissions from particles within the jet. This is achieved in EMPN through a novel message-passing mechanism that incorporates energy weights. In order to achieve this important feature, three steps are needed: graph construction, message-passing and the node readout.

In order to be infra-red and collinear safe, an observable has to be equal in the presence or absence of soft or collinear particles. This applies to the construction of the graph as well, i.e. the graph constructed by the presence of a soft or a collinear particle should be equal to the one constructed in its absence. A k-nearest neighbour graph would not allow for an IRC safe message-passing since adding a particle in the vicinity of a node could change the neighbourhood drastically. As can be seen in Figure 3.8, when a particle q splits into particles r and s, the extra particle b, which was previously in the neighbourhood, is now excluded. Therefore, the only way to construct the graph and preventing from changing the structure of the neighbourhood, is to draw a radius R_0 around each particle and create the connections with all the other particles that have an euclidean distance in the (η, ϕ)-plane less than R_0. Details of how this is a good choice can be found in [4].

Once the graph has been constructed, the message function in EMPN is defined as plays the crucial role on keeping the network IRC safe. The updated node features are given by:

$$h_i^{(l+1)} = \sum_{j \in N[i]} \Phi(p_i, p_j). \tag{3.3}$$

where $\mathcal{N}[i]$ represents the set of neighbouring nodes for node i, and $\Phi^{(l)}(p_i, p_j)$ is the message function between nodes i and j at layer l.

The message passing needs to be defined such that it is zero whenever the contribution comes from a soft particle, and it is equal to the sum of two particle whenever these are the product of a collinear split. Therefore, for a splitting $q \in N[i]$ to $r, s \in N'[i]$, the requirements for the message function Φ are:

$$\textbf{IR Safety:} \quad \Phi(p_i, p_r) \to 0 \quad \text{as} \quad z_r \to 0 \tag{3.4}$$

$$\textbf{C safety:} \quad (p_i, p_r + p_s) = \Phi(p_i, p_r) + \Phi(p_i, p_s) \quad \text{as} \quad \Delta_{rs} \to 0. \tag{3.5}$$

where z_r is the energy fraction of particle r and where Δ_{rs} denotes the angular distance between the two particles r and s. In order to respect these conditions, we need to define a scale function for each particle contributing to the update of the particle's

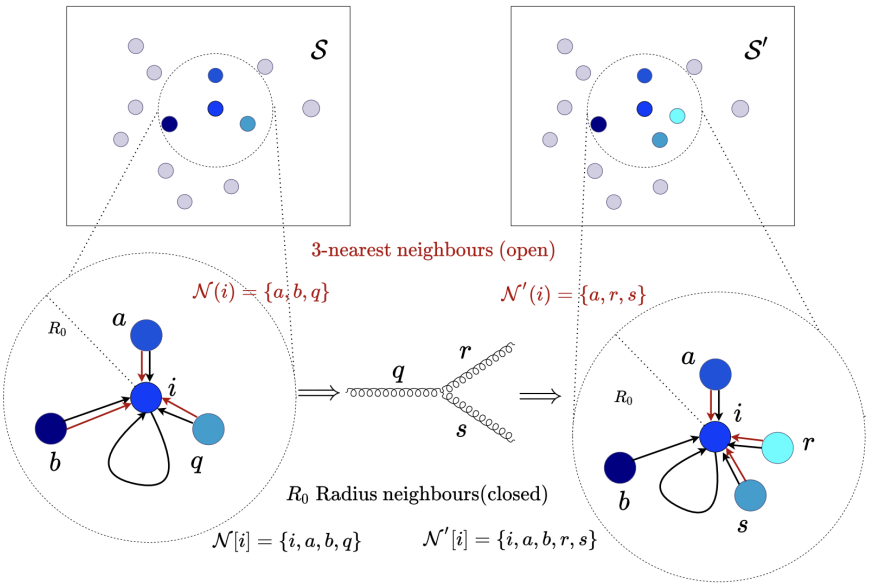


FIGURE 3.8: A k-nearest neighbour graph in the (η, ϕ)-plane will have a different structure when any particle q splits to r and s. The set S denote the particles in the jet when there is no splitting, while S' denotes the particles with q splitting. We show the directed edge connection to i from its three nearest neighbours with red on either side. The neighbourhood set $N(i)$ has b in it, however when q splits, $N'(i)$ does not contain b. Therefore, the graph's structure prevents a smooth extrapolation between the two scenarios in the infra-red and collinear limit. This is not the case for a radius graph with radius R_0 in the (η, ϕ)-plane, which is shown with black connections. We also include the self-loop of i, by using the closed neighbourhood sets $N[i]$ and $N'[i]$, since the node i could also split into two particles [4].

feature:

$$w_j = \frac{p_T^j}{\sum_{n \in N} p_T^n}, \tag{3.6}$$

where p_T^j is the transverse momentum of particle j, and the denominator normalizes this contribution by the total transverse momentum of all particles in the neighbourhood $\mathcal{N}[i]$, such that the contribution is weighted according to the proportion of its transverse momentum compared to the entire momentum of the neighbourhood. Hence, the updated node features that satisfy the IRC safety condition, can be written as:

$$h_i^{l+1} = \sum_{j \in \mathcal{N}[i]} \omega_j^{(\mathcal{N}[i])} \Phi^l(p_i, p_j). \tag{3.7}$$

where $\omega_j^{(\mathcal{N}[i])}$ ensures that the contributions of particles are weighted proportionally to their transverse momenta.

Finally, the graph readout needs also be of the form:

$$g = \sum_{i \in G} \omega_i h_i \tag{3.8}$$

where ω_i is similarly defined to ensure that the graph-level representation respects IRC safety

The EMPN architecture consists of multiple layers of energy-weighted message passing, followed by a global pooling operation and fully connected layers for classification. The authors apply EMPN to three jet tagging tasks: quark/gluon discrimination, W boson vs. QCD jet tagging, and top quark vs. QCD jet tagging.

The results demonstrate that EMPN achieves state-of-the-art performance on all three tasks, outperforming previous methods such as Energy Flow Networks [21]. Moreover, the authors demonstrate that EMPN is robust to variations in the jet clustering radius and the number of message-passing layers, indicating its stability and generalisability.

The key novelty of EMPN lies in its inherent IRC safety, which is a significant advancement in the application of GNNs to jet physics. By ensuring that the network output is insensitive to soft and collinear emissions, EMPN provides a more theoretically sound and reliable approach for jet tagging. This is crucial for reducing uncertainties in new physics searches and measurements of particle properties at the LHC.

Furthermore, EMPN's strong performance on benchmark tasks demonstrates its potential for real-world applications in high-energy physics experiments. Its ability to effectively capture the underlying physics of jet formation and evolution makes it a valuable tool for exploring the frontiers of particle physics.

Part II

Research and Results

Chapter 4

Eigenspace Projection for Jet Clustering

In this chapter, we delve into the methodology introduced in [8], which presents an innovative approach to jet clustering through Spectral Clustering. Unlike conventional methods, such as anti-k_T [35], that operate within the confines of the 2-dimensional eta-phi space, the key premise of this approach lies in projecting events into a novel feature space based on the kinematics of the constituent particles. This transformation allows for a fresh perspective on event structures, ultimately enhancing clustering performance significantly.

4.1 Introduction

In high-energy physics, accurately reconstructing the jets formed by particle collisions is crucial for understanding the underlying physics processes. Jets are collimated groups of particles resulting from the hadronisation of quarks and gluons. Traditional jet clustering algorithms, like the anti-k_T algorithm, operate in the eta-phi space (a cylindrical coordinate system adapted to the geometry of particle detectors). While effective, these methods can sometimes struggle with complex event structures, particularly when jets overlap or have varying densities.

Spectral Clustering is a powerful technique that has gained popularity across various fields, including machine learning and image processing. This method involves representing the data points as a graph, where each point is a node, and edges between nodes are weighted by an affinity measure reflecting their similarity. By computing the eigenvalues and eigenvectors of the graph's Laplacian matrix, the data is projected into a new space—known as the embedding space—where the clustering structure becomes more apparent.

The primary advantage of Spectral Clustering lies in its ability to adapt to the intrinsic structure of the data. This adaptability stems from the way it constructs the similarity graph and uses the spectral properties of the graph to determine clusters. Unlike traditional methods that rely on fixed geometric criteria, Spectral Clustering uses the eigenvalues and eigenvectors of the Laplacian matrix to capture the global structure of the data, allowing it to identify clusters based on the natural relationships within the data. This makes it particularly robust to variations in jet shapes and densities, which is beneficial in high-energy physics, where the kinematic properties of particle collisions can vary widely.

In this chapter, we detail our implementation of Spectral Clustering tailored specifically for jet physics. We discuss the necessary modifications to the standard algorithm to accommodate the unique challenges of particle physics data. Additionally, we present a comprehensive comparison of our results against traditional jet clustering methods, highlighting the improvements in clustering performance.

By projecting events into a novel feature space, our approach provides new insights into jet structures and offers a significant enhancement in clustering accuracy. The following sections will delve into the theoretical foundations of Spectral Clustering, describe our implementation in detail, and present the results of our analysis.

4.2 Method

In this section, we first introduce the original version of Spectral Clustering, detailing its principal features and elucidating the reasons behind its popularity and robustness as a clustering method. Subsequently, we present our modified version, specifically tailored for particle physics data. To enhance performance and ensure adherence to fundamental physics principles, several necessary modifications have been implemented. These adaptations not only improve the algorithm's accuracy but also optimise its application to the complex datasets characteristic of particle physics research.

4.2.1 Embedding into the Eigenspace

An excellent and comprehensive description of the theory behind spectral clustering can be found in [94]. This section provides a concise summary and highlights its application in jet clustering within particle physics.

Spectral clustering is an advanced method that transforms a set of data points into a new space, known as the embedding space, where the clusters are more

distinguishable. The transformation relies on the eigenvectors and eigenvalues of a Laplacian matrix derived from the data, which gives the method its name.

In spectral clustering, the input data is represented as a graph. This graph consists of nodes, which in our case represent particles, and edges, which represent the relationships between these particles. The edges can be weighted, meaning they carry a positive value known as affinity. Affinity quantifies the likelihood that the nodes connected by the edge belong to the same cluster. For jet clustering, this translates to the likelihood that the particles originated from the same parton shower.

The strength of spectral clustering lies in its ability to capture the global structure of the data through spectral properties of the graph, making it particularly robust in scenarios with complex data distributions. This robustness is why we decided to implement this method to jet clustering, where traditional methods like the anti-k_T algorithm may struggle with overlapping jets or varying jet densities.

Before delving into the theoretical foundations of spectral clustering, it's useful to visualise its effectiveness. Figure 4.1 compares the jets produced by the spectral clustering algorithm to those produced by the well-known Cambridge/Aachen (CA) algorithm across three different events. These events are selected to represent various challenging scenarios for jet formation algorithms.

Event 1 features jets with uniform density that blend smoothly into each other, illustrating how spectral clustering can maintain clear boundaries in homogeneous conditions. Event 2 presents three jets in very close proximity, showcasing the algorithm's capability to distinguish and accurately separate closely spaced jets. Event 3 contains jets of variable density, demonstrating spectral clustering's adaptability to different jet shapes and densities. These examples highlight why spectral clustering is a compelling method for jet formation, offering superior performance in diverse and challenging environments.

By transforming the original particle data into an embedding space that better reveals the underlying cluster structure, spectral clustering provides a powerful tool for high-energy physics. Its ability to handle complex, high-dimensional data makes it an invaluable method for accurately identifying jets, ultimately enhancing our understanding of particle collisions and the fundamental processes of the universe.

The theory.

The theory behind constructing the embedding space in spectral clustering is based on relaxing the optimization criteria that aim to best partition nodes into separate, disconnected subgraphs by grouping them into clusters. In a typical non-physics application, we start with points having coordinates, which need to be divided into a predetermined number of clusters, s. These points are represented as nodes in a

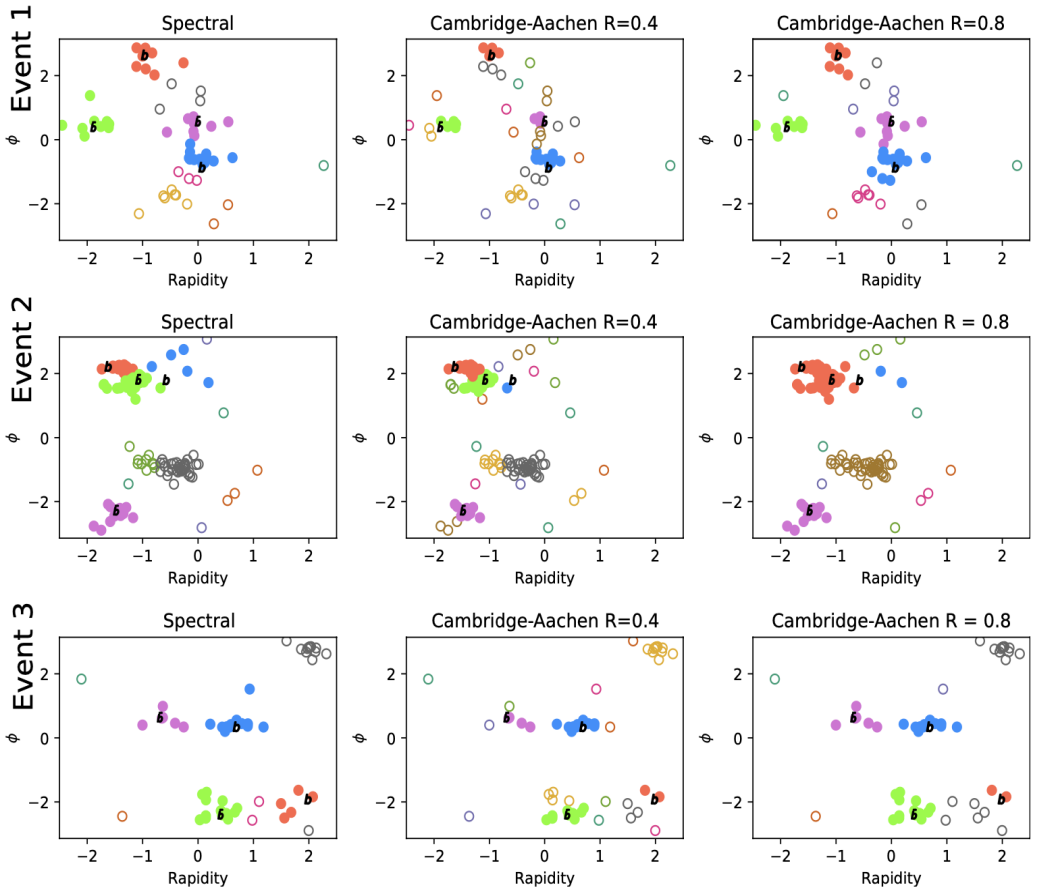


FIGURE 4.1: Behaviour of the spectral algorithm is compared to the well known Cambridge-Aachen algorithm using three events from our dataset. Each row contains an event, each column is a clustering algorithm. Circle colour indicates jet membership, filled circles indicates a b-quark jet.

graph, and the edge joining nodes i and j is assigned a weight $a_{i,j}$, which increases with the probability of i and j belonging to the same cluster.

To identify clusters, the graph is partitioned into subgraphs G_k, where $k = 1, ..., s$. The goal is to ensure that points connected by high-affinity edges remain in the same subgraph, while also avoiding clusters of very uneven sizes. This is achieved by minimizing the Normalized Cut (NCut) objective, which balances the cut size and the cluster size. NCut is defined as:

$$NCut = \frac{1}{2}\sum_{k}\frac{W(G_k, \bar{G}_k)}{\text{vol}(G_k)}, \tag{4.1}$$

where $W(G_k, \bar{G}_k)$ is the sum of all the edge weights that must be dropped to separate the cluster G_k from the rest of the graph, $\bar{G}_k$, so that $W(G_k, \bar{G}_k) = \sum_{i \in G_k, j \in \bar{G}_k} a_{i,j}$, and $\text{vol}(G_k)$ is the volume of G_k, defined as the sum of the weights of all edges connected to nodes in G_k. The denominator is used to penalise the formation of small clusters.

In order to determine which point will go in which G_k, a set of indicator vectors must be found. Membership of cluster G_k will be recorded in the indicator vector h_k:

$$h_{k_i} = \begin{cases} 1/\sqrt{\text{vol}(G_k)} & \text{if point } i \in G_k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

To find these indicator vectors the graph is represented by the graph Laplacian, L, a square matrix with as many rows and columns as there are points. To construct this Laplacian we define two other matrices, an off diagonal matrix $A_{i,j} = (1 - \delta_{i,j})a_{i,j}$ and a diagonal matrix $D_{i,j} = \delta_{i,j} \sum_q a_{i,q}$. Then the symmetric Laplacian can be simply written as

$$L = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}. \tag{4.3}$$

Considering just one cluster, G_k, when the Laplacian is multiplied by its indicator vector, the result is the term that NCut seeks to minimise for that cluster,

$$h_k' L h_k = \frac{1}{\text{vol}(G_k)} \sum_{i \in G_k, j \in \bar{G}_k} \left(\delta_{i,k} \sum_l a_{l,i} - a_{i,j} \right) = \frac{W(G_k, \bar{G}_k)}{\text{vol}(G_k)}. \tag{4.4}$$

To obtain the sum of all the terms, stack the indicator vectors into a matrix, $h_k' L h_k = (H'LH)_{kk}$, and the NCut aim described earlier becomes the trace

$$NCut(G_1, G_2, \ldots, G_n) \equiv \frac{1}{2} \sum_{k=1}^{n} \frac{W(G_k, \bar{G}_k)}{\text{vol}(G_k)} = Tr(H'LH), \tag{4.5}$$

where $H'H = I$.

Relaxation problem.

However, if we relax the requirements on the indicator vectors h in eq. 4.2, allowing elements of h to take arbitrary values, then the Rayleigh-Ritz theorem provides a solution. Trace minimisation in this form is done by finding the eigenvectors of the Laplacian L with smallest eigenvalues,

$$\lambda_{min} = \min_{||x|| \neq 0} \frac{x^H L x}{x^H x}, \tag{4.6}$$

where x is the relaxed indicator vector and an eigenvector of L.

Notice that L is a real symmetric matrix and, therefore, all its eigenvalues are real. Due to the form of the Laplacian, there will be an eigenvector with components all of the same value and its eigenvalue will be 0. This corresponds to the trivial solution of considering all points to be in one group. The next c = s eigenvectors of L, sorted by smallest eigenvalue, can be used to allocate points to s clusters. These eigenvectors are then used to determine the position of the points in the embedding space. Each

eigenvector has as many elements as there are points to be clustered, so the coordinates of a point are the corresponding elements of the eigenvectors. The standard method above is designed to form a fixed number of clusters, but typically we do not know how many jets should be created in an event. We will create an alternative algorithm, beginning with the principles of spectral clustering and adjusting to the needs of the physics being studied. Using the positions in embedding space, the points can be gathered agglomeratively, so that we do not need to choose a predetermined number of clusters.

Distance in the embedding space

When the relaxed spectral clustering algorithm is used to create an embedding space, points in each group will be distributed in this embedding space. Each point can be seen as a vector, its direction indicating the group to which this point should be assigned. Changes in magnitude of the vectors cause the Euclidean distance between the corresponding points to grow, however, an angular distance is invariant to changes in magnitude, therefore it is a suitable measure to use.

Information in the eigenvalues.

When clusters in the data are well-separated, the affinities between groups approach zero, and consequently, the eigenvalues will also be closer to zero. A small eigenvalue indicates that the corresponding eigenvector effectively separates particles based on their affinities. This information can be leveraged for clustering.

In traditional spectral clustering, the desired number of clusters, s, is predetermined. The embedding space is formed by selecting the $c = s$ eigenvectors with the smallest eigenvalues, excluding the trivial eigenvector. The resulting embedding space has c dimensions.

However, when forming jets, the number of clusters in the dataset is not known a priori, making it difficult to determine the appropriate number of eigenvectors to retain. We cannot simply set $c = s$. While choosing an arbitrary, fixed number of eigenvectors is possible, it is not optimal. A better approach is to select all non-trivial eigenvectors associated with eigenvalues smaller than a limiting value, λ_{limit}. For a symmetric Laplacian, the eigenvalues are $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2$, and λ_k is related to the quality of forming k clusters [95]. Removing eigenvectors with eigenvalues close to zero would discard valuable information, while keeping those with eigenvalues close to two would increase noise. Therefore, values of $0 < \lambda_{limit} < 1$ are sensible choices, and the specific choice within this range is not critical. In this way, the dimensionality of the embedding space will vary based on the number of non-trivial eigenvectors with corresponding $\lambda < \lambda_{limit}$.

We can further leverage the information contained in the eigenvalues to refine the embedding space. The dimensions of this space are not of equal importance, as

eigenvectors with smaller eigenvalues typically capture more meaningful cluster structure. We can account for this by scaling each eigenvector by a power, β, of its corresponding eigenvalue. Let x_{ni} denote the i-th component of the n-th eigenvector, which satisfies the eigenvalue equation:

$$\sum_j L_{i,j} x_{nj} = \lambda_n x_{ni}. \tag{4.7}$$

where $L_{i,j}$ are the elements of the Laplacian matrix. We then define the coordinates of the $j-th$ point in the c-dimensional embedding space as:

$$m_j = (\lambda_1^{-\beta} h_{1j}, \ldots, \lambda_c^{-\beta} h_{cj}). \tag{4.8}$$

This scaling effectively compresses the magnitudes of the vectors m_j in the n-th dimension by a factor of λ_n^{β}. Consequently, the larger the eigenvalue λ_n, the greater the compression applied to that dimension.

Stopping conditions

When employing a recursive clustering algorithm such as the generalised k_T algorithm, a robust stopping condition is essential. Initially, a stopping condition based on the smallest distance between points in the embedding space was explored. However, this approach proved unstable, as it was not possible to define a single threshold suitable for all events.

The minimum separation between the last two points joined before forming the final jets varies considerably across events, rendering it an unreliable stopping criterion. In contrast, the average distance between points before this last joining is more stable, being influenced by two opposing factors:

- Increase in Average Distance: As points are merged in a fixed-dimensional embedding space, the average distance between the remaining points increases. In physical space, this increase would roughly correspond to the decreasing number of points.
- Decrease in Embedding Space Dimensions: As fewer high-affinity combinations remain, the number of dimensions in the embedding space decreases, leading to a reduction in average distance.

These opposing effects create a balance in the average distance within the embedding space. As points merge, the average distance initially rises, but when only low-affinity combinations remain, the diminishing dimensionality of the embedding space counteracts this rise. This self-regulating behavior makes the mean distance in the embedding space a natural and effective stopping criterion. The validity of these assertions is supported by empirical evidence presented in 4.3.3.

4.2.2 Spectral Clustering for Jet Physics

This section delves into the spectral clustering methodology applied to the selection and merging of pseudojets. The algorithm presented here leverages the mathematical framework of spectral clustering to efficiently group pseudojets based on their spatial and kinematic properties. This approach facilitates a refined analysis of particle collisions by identifying and merging pseudojets into jets in a manner that respects underlying physical principles. The process involves constructing a normalised Laplacian matrix, computing eigenvectors and eigenvalues for embedding pseudojets, evaluating distance metrics in the transformed space, and applying stopping conditions to finalise jet formation. Detailed steps and mathematical formulations are provided to elucidate the procedures used to dynamically adjust clusters and manage varying particle interactions, ultimately enhancing the accuracy and interpretability of jet clustering in high-energy physics experiments.

- **Graph Construction and Affinity Calculation:**
  We begin by constructing a graph where nodes represent pseudojets. The edges between nodes are based on the spatial proximity of these pseudojets and are weighted by an affinity metric, which reflects their closeness. The distance $d(t)_{i,j}$ between pseudojets i and j at step t is defined as:

$$d(t)_{i,j} = \sqrt{(y(t)_i - y(t)_j)^2 + \delta(\phi(t)_i, \phi(t)_j)^2}, \tag{4.9}$$

  where $y(t)_i$ and $\phi(t)_i$ denote the rapidity and the angle in the transverse plane of the pseudojet i at step t, respectively, and δ represents the distance measure for the cyclic coordinate. Unlike many traditional jet clustering methods, there is not any transverse momentum p_T dependence. The affinity between two pseudojets is then computed:

$$a(t)_{i,j} = exp(-d(t)^{\alpha}_{i,k}/\sigma_v), \tag{4.10}$$

  where $\alpha = 2$ is the standard Gaussian kernel as used in [96]. With this choice of the affinity functions, pseudojets become more similar as the distance becomes shorter.

- **Affinity Pruning and Noise Reduction:**
  To enhance the clarity of clustering, affinities between pseudojets that are far apart are set to zero, preserving only the k_{NN} nearest neighbors for each pseudojet. This pruning step significantly reduces noise and focuses the clustering process on the most likely candidates for merging.

- **Construction and Dynamics of the Normalised Laplacian:**
  Now it is time to create the normalised Laplacian matrix, which is proportional to $-a(t)_{i,j}$ in the i^{th} row and j^{th} column. For ease of notation, let $z(t)_j$ be a

measure of the contribution of pseudojet j to a cluster. Before the first merge, $z(1)_j = \sum_k a_{j,k}$. Define three square matrices: $A(t)_{i,j} = (1 - \delta_{i,j})a(t)_{i,j}$, known as the adjacency matrix; $B(t)_{i,j} = \delta_{i,j} \sum_k a(t)_{i,k}$, known as the degree matrix; and $Z(t)_{i,j} = \delta_{i,j} z(t)_i$, which normalises the Laplacian. The Laplacian is then expressed as:

$$L(t) = Z(t)^{-\frac{1}{2}}(B(t) - A(t))Z(t)^{-\frac{1}{2}}. \tag{4.11}$$

With each merging step, the Laplacian matrix reduces in size by one row and one column, corresponding to the reduction in particle numbers. If pseudojets i and j from step $t-1$ merge to form pseudojet i at step t, their sizes update as follows:

$$z(t)_i = s_{i,j}(t-1)(z(t-1)^i + z(t-1)_j) + (1 - s_{i,j}(t-1))b(t)_i. \tag{4.12}$$

. The sizes for all other pseudojets adjust accordingly:
$z(t)_q = s(t)z(t-1)_q + (1 - s(t))b_q$. This mechanism ensures that the size of a pseudojet accumulates over time, but resets when soft or collinear particles merge into it. To manage these merges, singularity factors are introduced:

$$s_{i,j}(t) = 1 - \frac{\kappa}{\kappa + min(p_T(t-1)^i, p_T(t-1)_j)d(t-1)_{i,j}}, \tag{4.13}$$

where κ is a constant, here chosen to be 0.0001.

- **Spectral Embedding of Pseudojets Using Laplacian Eigenvectors:**
The eigenvectors of the Laplacian

$$L(t)h(t)_q = \lambda(t)_q h(t)_q, \quad q = 1, \ldots, c \tag{4.14}$$

are used to create the embedding of the pseudojets. The eigenvector corresponding to the smallest eigenvalue represents the trivial solution, which would cluster all the points into a single group. All non-trivial eigenvectors, corresponding to eigenvalues less than a predefined limit,
$\lambda(t)_c < \lambda_{limit} < \lambda(t)_{c_+1}$, are retained. If no eigenvectors meets this criterion, the clustering process terminates at this point. Each eigenvector is divided by its corresponding eigenvalue raised to the power of β. To prevent zero division errors, the smallest eigenvalues are clipped to 0.001, such that

$$\lambda'_q = min(\lambda_q, 0.001). \tag{4.15}$$

This operation to compresses the dimensions containing less information. The embedding space is then formed with the eigenvectors having as many elements as there are pseudojets. The coordinates of the j^{th} pseudojet at step t are defined as follows:

$$m(t)_j = \lambda'_1(t)^{-\beta} h_1(t)_j, \ldots, \lambda'_c(t)^{-\beta} h_c(t)_j. \tag{4.16}$$

- **Calculation of Distance Metrics in the Embedding Space:**
  The embedding space utilises angular distances as the most appropriate metric for measuring distances between pseudojets. This is mathematically expressed as:
  $$d'(t)_{i,j} = s(t)_{i,j} \arccos\left(\frac{m(t)_i \cdot m(t)_j}{\|m(t)_i\| \|m(t)_j\|}\right), \tag{4.17}$$
  where $\|m\|$ represents the Euclidean norm of m. The scaling factor $s(t)_{i,j}$ is designed to ensure that soft and/or collinear particles are merged early in the clustering process. This early merging is crucial as it resets the size of the pseudojets based on their recent merging history.

- **Stopping condition:** The clustering process incorporates a stopping condition to determine when to halt merging. This condition is evaluated by comparing the mean of the square roots of the distances $d'(t)_{i,j}$ to a threshold parameter R. Mathematically, this is represented as:
  $$\frac{2}{c(c-1)} \sum_{i \neq j} \sqrt{d'(t)_{i,j}} < R, \tag{4.18}$$
  where c is the number of pseudojets currently considered. If this condition is satisfied, the two pseudojets with the smallest embedding distance are merged. The two pseudojets are combined using the E-scheme: a new pseudojet, which will replace the other two, has a 4-momenta equal to the sum of the two joiners pseudojets' 4-moments, i.e. $p(t+1)_k = p(t)_i + p(t)_j$.

When the mean of the distances in the embedding space exceeds the threshold R, all remaining pseudojets are promoted to jets. Jets containing fewer than two tracks are classified as noise and removed from further consideration. This procedure dynamically forms a variable number of jets from a variable number of particles, adapting to the specific data characteristics of each event. An illustrative example of the constructed first embedding space is shown in Figure 4.2. This visualisation effectively highlights how the embedding space delineates the clusters, providing a clear depiction of the clustering dynamics.

4.2.3 Hyperparameters

Unlike most deep learning methods currently used in particle physics, spectral clustering does not rely on extensive arrays of learned parameters. Instead, it uses a small, interpretable set of parameters. The optimal values for these parameters were identified by conducting scans and observing how changes affected the formation of jets.

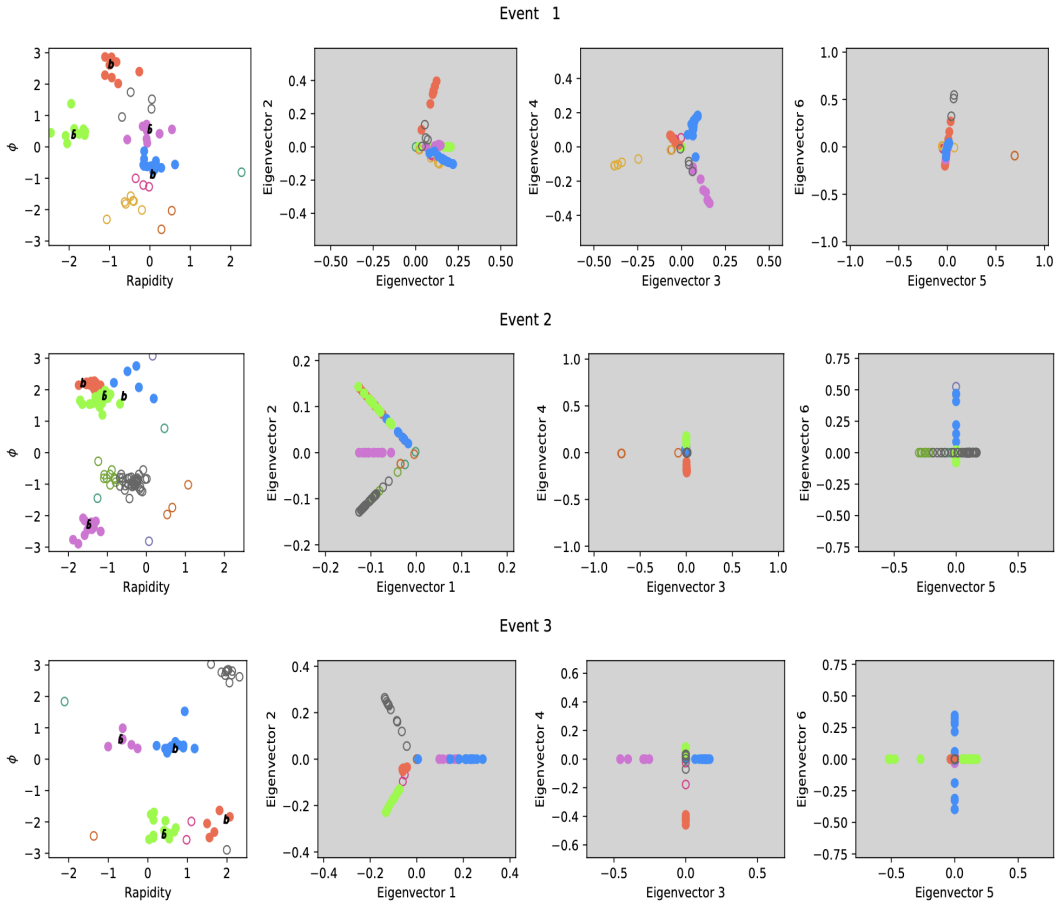


FIGURE 4.2: Example of the construction of the embedding space with Spectral Clustering at the first step. To the left, the white plots show the particles in the events as points on the unrolled detector barrel. The colour of each point indicates the jet is is assigned to., filled circles are b-jets. On the right, three grey plots show the first 6 dimensions of the embedding space and the location of the points within the embedding space.

Section 4.2.2 discusses six parameters: $\sigma_v, \alpha, k_{NN}, \lambda_{\text{limit}}, \beta, and R$. Although there are more parameters than those used in the generalised k_T algorithm, they do not require precise values to achieve good performance. Detailed explanations of these parameters are provided.

- σ_v is a scale parameter in physical space. It represents an approximate average distance between particles in the same shower, or alternatively, the size of the neighbourhood around each particle. It is closely related to the stopping parameter for the generalised k_T algorithm R_{k_T}, as both influence the width of the jets formed. The value of should be of the same order of magnitude as R_{k_T}.

- α changes the shape of the distribution used to describe the neighbourhood of a particle. Higher values reduces the probability of joining particles outside σ_v. In particular, $\alpha = 2$ defines a Gaussian kernel.

- k_{NN} dictates the minimum number of non-zero affinities around each point. Lower values create a sparser affinity matrix, reducing noise at the potential cost of lost signal. Values above 7 are seen to have little impact.
- λ_{limit} is a means of limiting the number of eigenvectors used to create dimensions in the embedding space. Only eigenvectors corresponding to eigenvalues less than λ_{limit} are used. Thus, the number of dimensions in the embedding space can be increased with a larger values of λ_{limit}. However, as the eigenvalues will be influenced by the number of clear clusters available, there will not be the same number of dimensions in each event. Values of $0 < \lambda_{limit} < 1$ are sensible choices.
- β accounts for variable quality of information in the eigenvectors, as given by their eigenvalues, in such a way that the dimensions of the embedding spaces corresponding to higher eigenvalues are compressed, as they contain lower quality information.
- **R** determines the expected spacing between jets in the embedding space. As the number of dimensions in the embedding space grows with increasing number of clear clusters, it will not results in the same or similar number of clusters each time.

To assess the impact of varying parameters on clustering behaviour, scans were conducted on a small sample of 2,000 events with numerous parameter configurations. Instead of employing more sophisticated methods, a straightforward random scan was utilised.

Using Monte Carlo (MC) truth information, we can establish a success metric. For each target object (e.g., a b-quark), MC truth reveals which visible particles in the detector originated from that object. Often, a detected particle may have been produced by more than one object, such as a particle from a $b\bar{b}$ pair; in these instances, both originating objects are considered jointly. The complete set of visible particles emanating from these objects is termed their descendants. The objective in jet clustering is to encapsulate all descendants in the same number of jets as there were originating objects. Thus, the descendants of a $b\bar{b}$ pair should ideally be encapsulated within exactly two jets. This approach of using MC information was also explored in previous studies [59], particularly for jets that originate from a colour singlet hard particle, such as a W boson. Furthermore, we aim to identify jets arising from systems with a colour charge. By allowing descendants from groups of interacting showers to be clustered in any configuration that yields the correct number of jets, we circumvent the necessity of uniquely associating each descendant with a specific object, such as a b-quark, which is impractical for colour-charged objects as noted in [59].

Jet finding algorithms can err in two primary ways during the task of reconstructing objects from their descendants. The first error occurs when some descendants are omitted, resulting in jets with less mass than expected. The second error involves the inclusion of unrelated particles, such as initial state radiation or particles from other interactions, which increases the jet's mass erroneously. Although these mistakes might cancel each other in the total mass calculation of the jet, they are undesirable individually. Therefore, separate metrics are developed for each error type: "Signal mass lost" quantifies the mass discrepancy caused by missing descendants, and "Background contamination" measures the excess mass due to unrelated particles. To address these issues, a "Loss" function is constructed as a weighted combination:

$$Loss = \sqrt{\omega(\text{Background contamination})^2 + (\text{Signal mass lost})^2}, \tag{4.19}$$

where ω is a weight that balances the preference between minimising "Signal mass lost" and reducing "Background contamination". In scenarios using an anti-k_T algorithm, adjusting ω can decrease "Signal mass lost" at the cost of increasing "Background contamination". We have chosen to compare this with $R_k = 0.8$ because our sample dataset features well-separated jets and minimal background noise. For this setup, the weight ω is set to 0.53 to reflect these conditions.

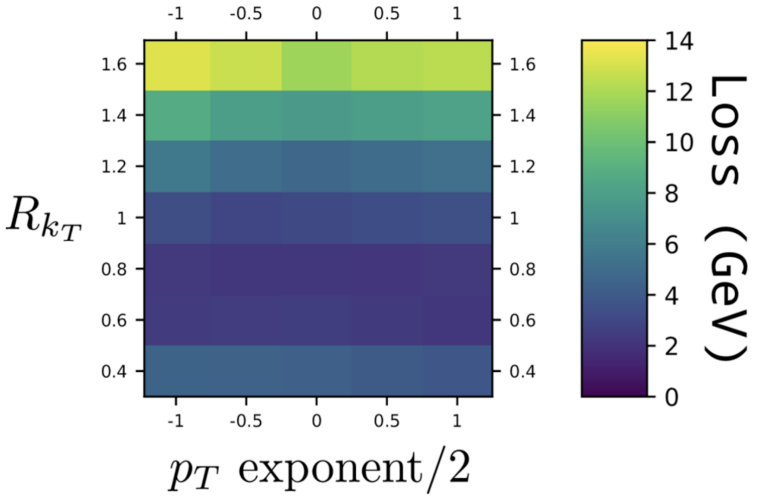


FIGURE 4.3: The generalised k_T algorithm has 2 parameters that can be varied. The stopping condition, R_{k_T}, and a multiple for the exponent of the p_T factor. When the exponent of the p_T factor is -1 the algorithm becomes the anti-k_T algorithm.

An illustration of this evaluation using the generalised k_T algorithm is shown in Figure 4.3, where it is evident that various p_T exponent values can yield satisfactory results, though there is a slight preference for suppressing "Signal mass lost" to achieve clearer mass peaks.

For spectral clustering, which involves managing more than two variables, two-dimensional slices of the parameter space are analysed. These slices represent the

best-performing combinations and are displayed in Figure 4.4, using the same colour scale as Figure 4.3 for easy comparison. As shown, the parameter choices are not finely tuned, as many configurations lead to successful outcomes. For instance, parameters like α, k_{NN}, β and λ_{limit} show flexibility, performing well across a broad range of values. Even parameters such as R and, notably, σ show significant signal "Loss", say for $R = 1.35$ and $\sigma_v = 0.4$, this happens in very narrow ranges. Definitively, the parameters set for the remainder of the study are $\alpha = 2$, $k_{NN} = 5$, $R = 1.26$, $\beta = 1.4$, $\sigma_v = 0.15$ and $\lambda_{limit} = 0.4$.

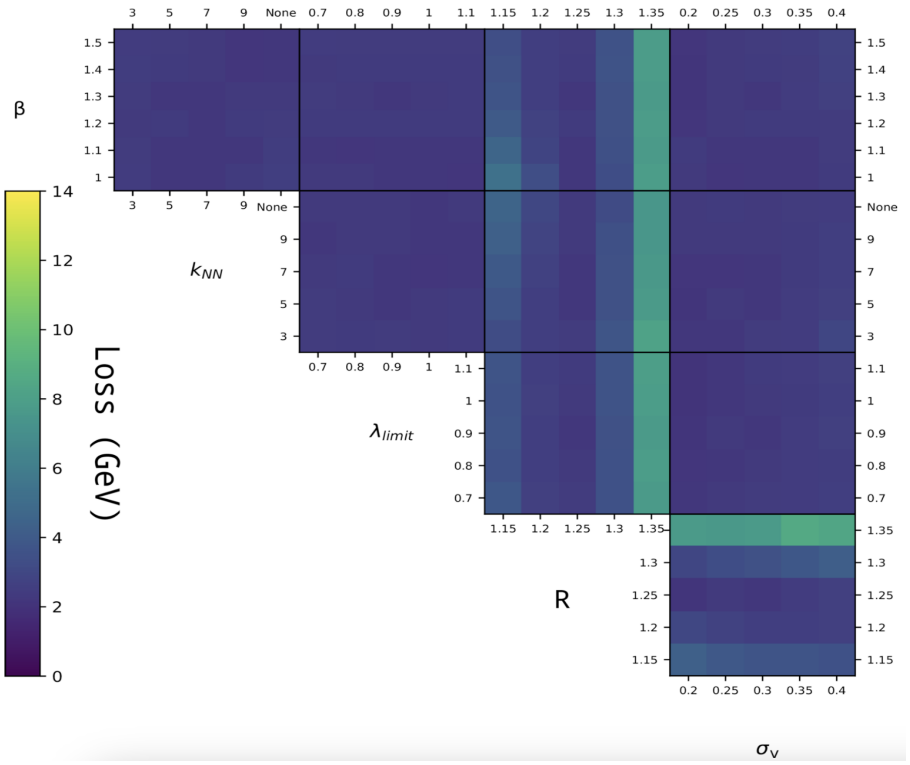


FIGURE 4.4: The spectral clustering algorithm has 6 parameters that can be varied (described in the text).

4.3 Results and Limits

In this section, we present the results of applying the spectral algorithm to physics data. We compare the performance of our method with benchmarks obtained using both the anti-k_T and the CA algorithms at two different jet radii, $R_{k_T} = 0.4$ and $R_{k_T} = 0.8$.

To evaluate the behaviour of our model, we used three datasets. The first two uses a 2-Higgs Doublet Model (2HDM) setup [1], as described in details in [97], and the last is purely a Standard Model (SM) processes:

[1] We will not delve into the details of this physics model as our focus is to demonstrate the performance of the spectral clustering method on physics data.

- *Light Higgs*: A SM-like Higgs boson with a mass of 125 GeV decays into two light Higgs states with mass 40 GeV, which further decay into $b\bar{b}$ quark pairs. The process is $gg, q\bar{q} \to H_{125GeV} \to h_{40GeV}h_{40GeV} \to b\bar{b}b\bar{b}$.
- *Heavy Higgs*: A heavy Higgs boson with a mass 500 GeV decays into two SM-like Higgs states with mass 125 GeV, which further decay into $b\bar{b}$ quark pairs. The process is $gg, q\bar{q} \to H_{500GeV} \to h_{125GeV}h_{125} \to b\bar{b}b\bar{b}$.
- *Top*: A $t\bar{t}$ pair decays semileptonically, i.e. one $W^{\pm}$ decays into a pair of quark jets jj and the other into a lepton-neutrino pair $l\nu_{e/\mu}$. The process is $gg, q\bar{q} \to t\bar{t} \to b\bar{b}W^+W^- \to b\bar{b}jjl\nu_{e/\mu}$

We selected these datasets because the events each contain four distinct jets, presenting a high multiplicity, which serves as another metric for testing our method.

The data, comprising a total of $\mathcal{O}(10^5)$ events for each process, was generated using MadGraph [98] for the partonic processes and Pythia [45] for the showering. Instead of a full detector simulation, we applied particle cuts to mimic detection: the reconstructed particles must have a pseudorapidity $|\eta| < 2.5$ and a transverse momentum $p_T > 0.5$ GeV. The events are required to have $p_T > 15/30/50$ GeV, depending on the dataset.

4.3.1 Mass Peak Reconstruction

Once we have clustered the particles in an event, as we expect to end up with more than one jet, we need to tag each jet to the particle it most closely resembles. To achieve this, we introduce a tagging distance metric based on the Monte Carlo truth:

$$d_{tag} := \sqrt{(y_{quark} - y_{jet})^2 + (\delta(\phi_{quark}, \phi_{jet}))^2}, \tag{4.20}$$

with y being the pseudorapidity and ϕ the angular distance, where $\delta(\phi_{quark}, \phi_{jet})$ is a distance measure accounting for the periodic nature of the ϕ component, ensuring the difference remains less than 2π. After clustering and identifying the b quarks, we combine them to recreate the heavier particle, based on the dataset, i.e., either into Higgs or Top quarks.

In Figure 4.5, three selections are plotted for the Light Higgs MC sample. We display events where all four b-jets are combined into the total invariant mass of the event, thus reconstructing the mass of the SM Higgs boson. Each event also contains two light Higgs states. These are differentiated by the mass of the particles (generated by them) that pass the particle cuts. The light Higgs boson reconstructed from the $2b$-jet system with the greater mass visible to the detector is termed the "Light Higgs with stronger signal," while the one reconstructed with less mass visible is called the "Light

Higgs with weaker signal." The correct jets for each Higgs mass reconstruction are identified using MC truth, ensuring accurate pairings. (If two such dijet systems are not found, the event is not included in the plots). Overall, it can be seen that spectral clustering forms the best peaks, narrow and close to the correct mass. In fact, its performance is comparable to that of anti-k_T with $R_{k_T} = 0.8$ and is clearly better than the $R_{k_T} = 0.4$ option. The parameters for spectral clustering were specifically chosen to minimise a loss that was based on the performance of CA with $R_{k_T} = 0.8$ in this Light Higgs dataset. Given this, the similarity of the mass peaks is not surprising. It will be more interesting to see how the algorithm performs on a different dataset.

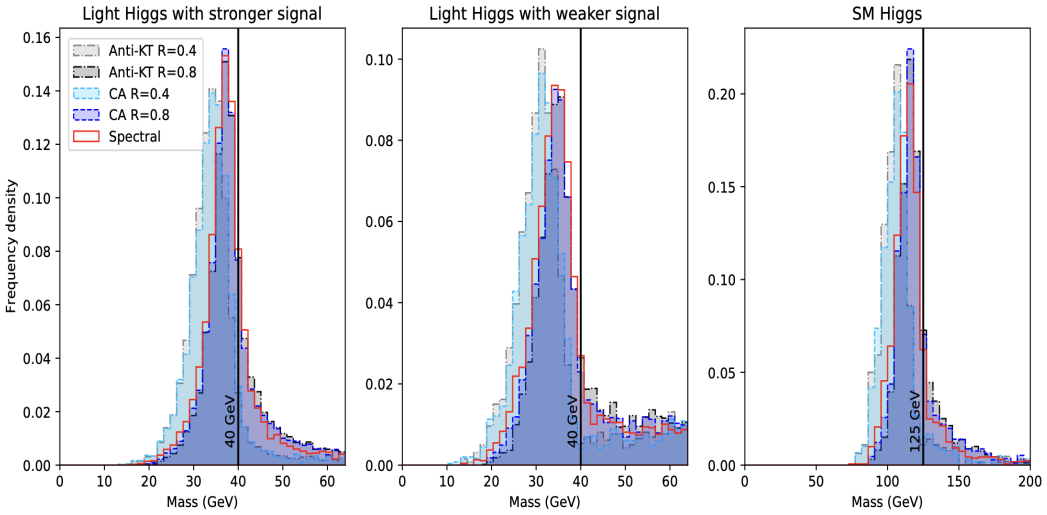


FIGURE 4.5: Three mass selections are plotted for the Light Higgs dataset. From left to right we show: the invariant mass of the 4b-jet system, of the 2b-jet system with heaviest invariant mass and of the 2b-jet system with lightest invariant mass (as defined in the text). Three jet clustering combinations are plotted as detailed in the legend. The spectral clustering algorithm is consistently the best performer in terms of the narrowest peaks being reconstructed and comparable to anti-k_T / CA with $R_{k_T} = 0.8$ in terms of their shift from the true Higgs mass values, with anti-k_T / CA with $R_{k_T} = 0.8$ being the outlier.

In Figure 4.6 the exercise is repeated for the Heavy Higgs MC dataset. All the parameters of spectral clustering are the same as in the Light Higgs MC sample yet we note that its performance is still excellent, with very sharp peaks at the correct masses, although the three clustering algorithms are overall much closer in performance. So, we are again driven to conclude that spectral clustering is probably the best performer overall with the added benefit of not requiring any adjustment of its parameters to achieve this.

Finally, in Figure 4.7, the W and t mass peaks for semi-leptonic $t\bar{t}$ decays are shown. Three mass reconstructions are given. The hadronic W is reconstructed from the jets that come from the quarks it decayed to. Correct decisions about which quarks correspond to which particle in the hard process are made by using information in the MC: this is to prevent performance evaluation of clustering to be confounded by

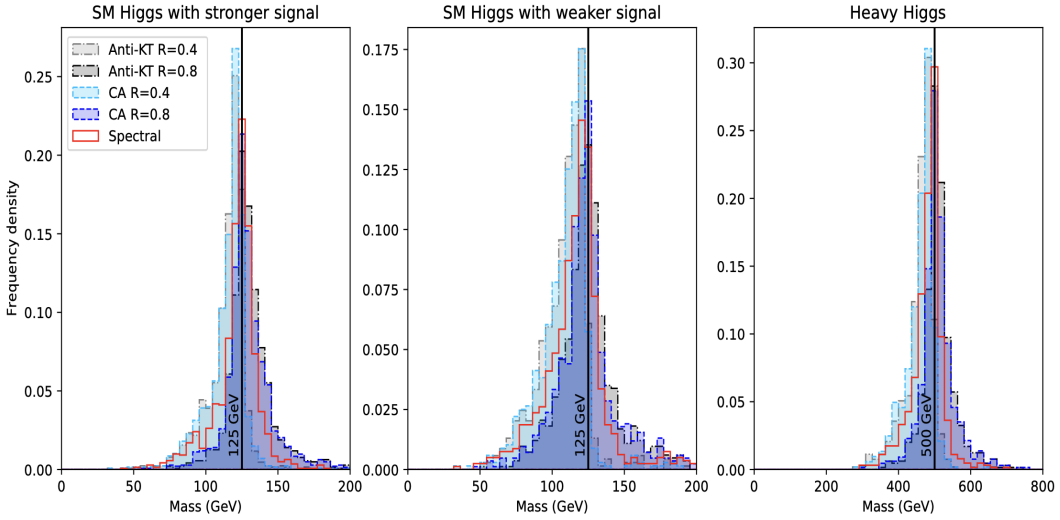


FIGURE 4.6: Same as Figure 4.5 for the Heavy Higgs dataset. Here, the performance of the spectral clustering and anti-k_T (with both 0.4 and 0.8 as jet radii) clustering algorithms is much closer to each other.

mismatching. To tag a jet with a quark we use the tagging distance measure d_{tag}. The W will always decay to a pair of quarks, which may be captured in one jet or separate jets. If either of the these quarks are too far away from the closest jet to tag it, that is d_{tag} ¿ 0.8, then it is not associated with any jet and the hadronic W is not reconstructed. The mass of the hadronic top is then reconstructed in events where the hadronic W could be reconstructed and the b-jet from the hadronic top is also found. The leptonic top is then reconstructed in events where a b-jet from the top is combined with the reconstructed W which decays leptonically. The leptonic reconstruction of the W uses the momentum of the electron p_l, the missing transverse momentum p_T^{miss} (identified with that of the neutrino) and the longitudinal neutrino momentum (p_L^{ν}, which is unknown) in a quadratic equation, $(p_l + p_{miss} + p_\nu)^2 = m_W^2$, of which only the real solutions are plotted. In this case, it can be seen that spectral clustering is adapting to jets of a different radius. In fact, while before its behaviour had mostly resembled anti-k_T with $R_{k_T} = 0.8$, it has now moved closer to the case with $R_{k_T} = 0.4$. (Semi-leptonic top events would typically be processed using anti-k_T with $R_{k_T} = 0.4$.) The peaks of spectral clustering are not quite as narrow as those from anti-k_T with R_{k_T} = 0.4, but they improve on $R_{k_T} = 0.8$ and their location is substantially correct.

4.3.2 Multiplicity

Jet multiplicities, that is, the number of reconstructed jets found per event, are given for the anti-k_T , CA and spectral clustering algorithms. These can be seen for the three datasets in Figure 4.8. Herein, it is seen that spectral clustering produces the best multiplicity (i.e., most events where 4 jets are found) for Top events while for the Light Higgs and Heavy Higgs MC samples it creates a multiplicity closer to that of anti-k_T

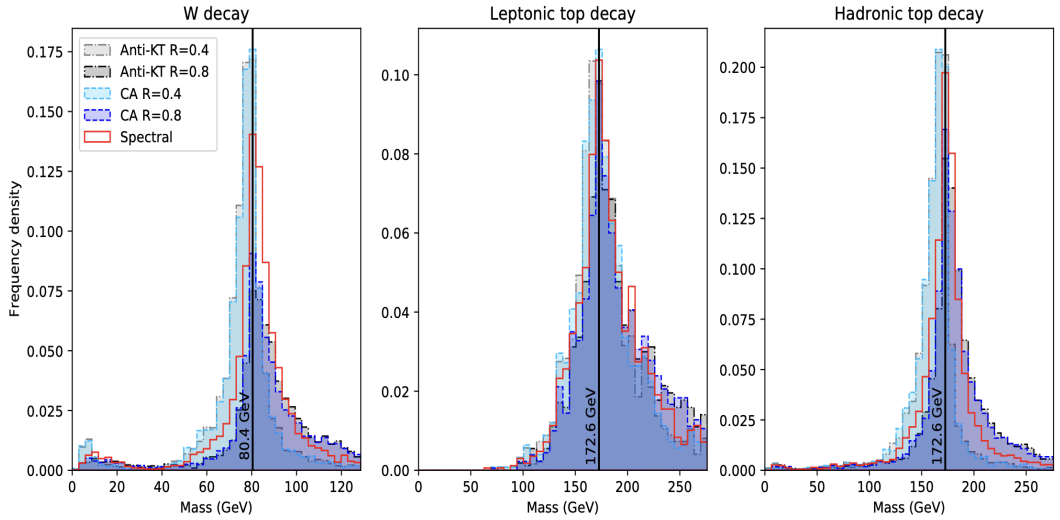


FIGURE 4.7: Three mass selections are plotted for the Top dataset. From left to right we show: the invariant mass of the light jet system, of the reconstructed leptonic W (as described in the text) combined with a b-jet and of the hadronic W combined with the other b-jet. Three jet clustering combinations are plotted as detailed in the legend. The spectral clustering algorithm consistently outperforms anti-k_T with jet radius 0.8 and is slightly worse than the anti-k_T /CA one with $R_{k_T} = 0.4$, but only in terms of sharpness, not of location of the mass peak.

/CA2 with $R_{k_T} = 0.4$ than $R_{k_T} = 0.8$, the first of these being the best performer of the two. This study provides evidence that spectral clustering, unlike anti-k_T , adapts to the different final states without having to adjust its parameters. The anti-k_T algorithm suggests 0.4 to be the best choice for all datasets, but this is in tension with the fact that different masses from different datasets do require the anti-k_T parameters to be adjusted, as we saw in the previous section.

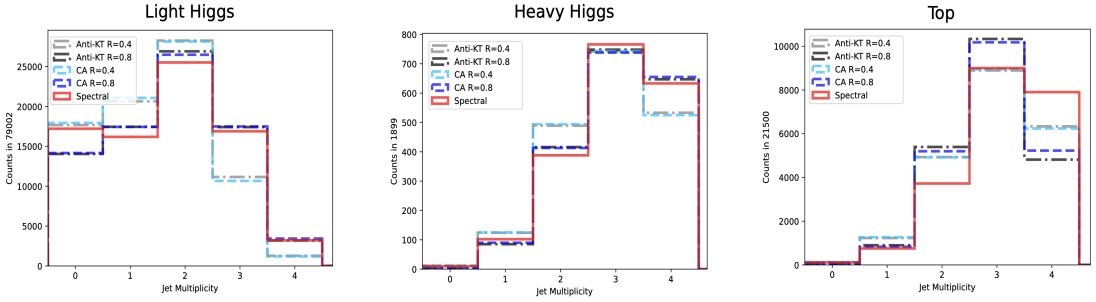


FIGURE 4.8: Jet multiplicities for the anti-k_T and CA (for two R_{k_T} choices) and spectral clustering algorithms on the Light Higgs, Heavy Higgs and Top MC samples. For all such datasets, the hard scattering produces 4 partons in the final state, so maximising a multiplicity of 4 jets indicates good performance.

4.3.3 Stopping Condition

To support the assertions made in Section 4.2.2, Figure 4.9 examines the behaviour of the mean distance during clustering. Clustering is performed on the Light Higgs dataset described earlier, using the spectral algorithm parameters specified at the end of Section 4.2.2.

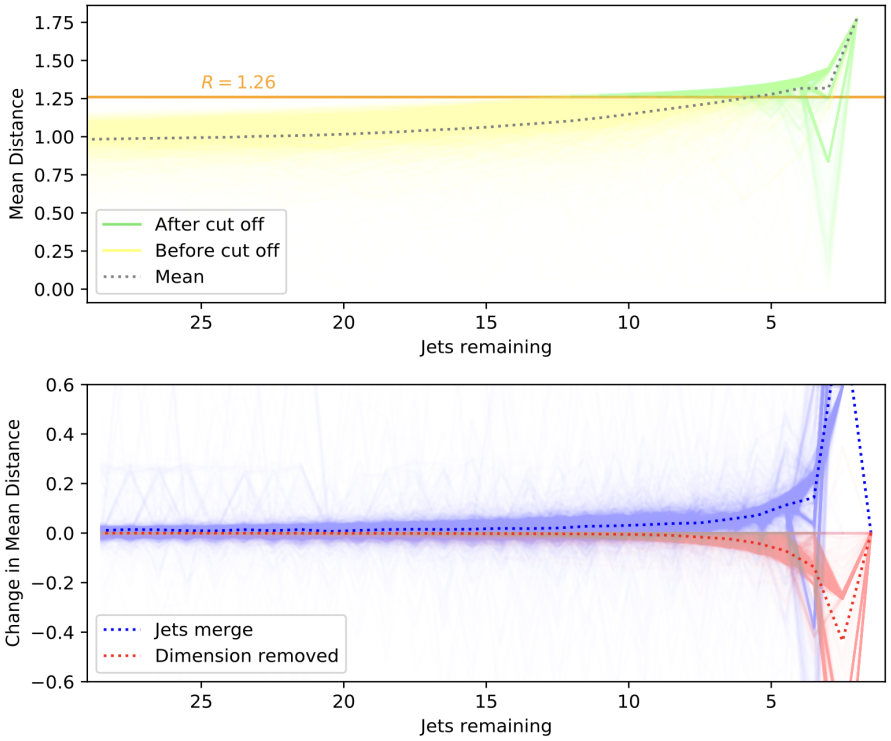


FIGURE 4.9: Mean distance between pseudojets in the embedding space during clustering for 2000 events. (Top) Mean distance vs. the number of remaining pseudojets. Lines are yellow until the mean distance exceeds the stopping condition (R = 1.26), then turn green. The dotted line represents the average across all events. (Bottom) Changes in mean distance due to merging pseudojets (blue) and reduction in embedding space dimensionality (red). Dotted lines show the respective averages.

The upper panel of Figure 4.9 displays the mean distance between pseudojets for 2000 events, plotted against the number of remaining pseudojets. Each line is coloured yellow until its value first exceeds R = 1.26 (the stopping condition), after which it turns green. In practice, the spectral clustering algorithm would typically halt at the end of the yellow segment when the stopping condition is met. However, the green section is included here to illustrate the behaviour beyond this point.

The transition from yellow to green occurs with approximately 3 to 13 pseudojets remaining, confirming that a mean distance stopping condition does not enforce a fixed number of jets per event. Notably, the mean distance increases smoothly for most of the clustering process, becoming erratic only when fewer than 5 pseudojets remain.

The lower panel delves into the factors influencing the mean distance. Again, each of the 2000 events is represented by a solid line. The blue lines depict changes in mean distance due to merging pseudojets. While merging typically increases the mean distance as the embedding space becomes sparser, there are occasional configurations where the mean distance decreases (blue lines dip below zero). The plot indicates that such configurations are less frequent than those that increase the mean distance.

The red lines in the lower panel show changes in mean distance due to a reduction in the dimensionality of the embedding space. This invariably decreases the mean distance, keeping the red lines at or below zero. Since not every algorithm step reduces dimensionality, the red lines for an event are often zero.

These two factors, the merging of pseudojets and the reduction in dimensionality, balance each other to produce a relatively stable trend in the mean distance.

(Note: A rare third possibility exists where the number of dimensions in the embedding space increases. This is not depicted in the plot, as the corresponding line would be indistinguishable from y = 0 and would clutter the figure.)

4.3.4 Run Time

One of the most crucial aspects of evaluating a new algorithm is its runtime and memory requirements. Typically, finding an optimal trade-off between performance and efficiency is key to selecting the best algorithm. In this section, we will discuss one of the significant limitations of this novel algorithm: the runtime.

Currently, the preferred methods for clustering particles into jets are the general k_T algorithms [5], which operate at an impressive time complexity of $\mathcal{O}(n \log n)$. This efficiency sets a challenging benchmark. In contrast, Spectral Clustering involves eigenvalue calculations, an $\mathcal{O}(n^2)$ operation, representing a significant bottleneck. This requirement renders our algorithm less competitive compared to the k_t family. The initial steps of the spectral algorithm are similar to those of the generalised k_T, and thus, they are expected to share the same runtime. However, the implementation used in this work does not incorporate the improvements that evolved generalised k_T from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. Consequently, these initial steps operate at $\mathcal{O}(n^2)$. Adding the eigenvector calculations further compounds the complexity, leading to a total expected runtime of $\mathcal{O}(n^4)$ with a naive implementation.

However, following this logic, the results presented in Figure 4.10 are quite surprising. We tested the execution times of the two distinct algorithms on datasets with varying numbers of particles. Interestingly, the spectral algorithm actually runs in $\mathcal{O}(n^3)$, not $\mathcal{O}(n^4)$, despite no specific optimisations being applied. The implementation of the spectral algorithm is a basic Pythonic implementation.

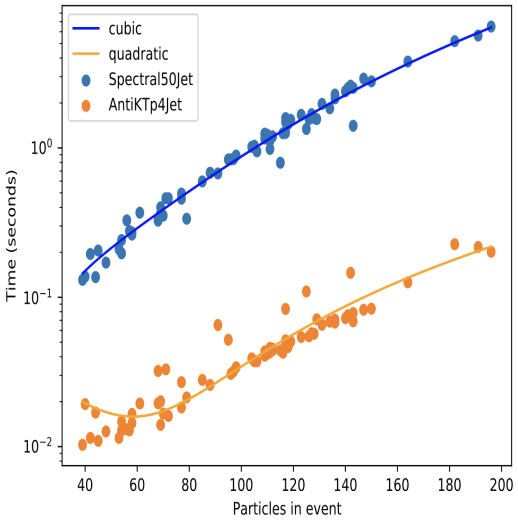


FIGURE 4.10: The run time of spectral, compared to a naive implementation of generalised k_T (without the performance refinements in [5]), on datasets of varying size, i.e. events with increasing number of particles. Cubic and quadratic fits are shown for each dataset respectively. This shows that spectral runs in $\mathcal{O}(n^3)$, instead of the the expected $\mathcal{O}(n^4)$.

Thus, the improvements that reduce the algorithm's complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ must be attributed to the intelligently designed libraries used. The eigenvector calculation was performed using the function *scipy.linalg.eigh* from *scipy* [99]. This function optimises the calculation by imposing two criteria: firstly, it requires that the input matrix be Hermitian, a condition met by the Laplacian matrix used in our tests; secondly, it allows the specification of the desired range of eigenvalues. As detailed in Step 5, Section 8.2.2, our spectral algorithm requires only the eigenvectors corresponding to a pre-specified range of eigenvalues. These optimisations appear to yield an $\mathcal{O}(n)$ runtime improvement.

There are several ways to enhance the speed of this algorithm, ranging from refining the code for greater efficiency to implementing it in a lower-level programming language, such as C, or optimising the algorithm itself—perhaps by finding a method to avoid recomputing the Laplacian Matrix at every step. Although many approaches have been proposed, they slightly deviate from the scope of our work; therefore, no further investigations have been pursued.

Chapter 5

Obscuring and Enhancing Jet Variables for Tagging

In this chapter, we delve into the latest research on how obscuring or leveraging physics knowledge affects the performance of machine learning models for the specific task of jet tagging.

We introduce JetLOV [9], an enhanced Graph Neural Network that achieves remarkable results by relying exclusively on the model's output, effectively replacing traditional physics reasoning with a machine learning black box.

5.1 JetLOV: Enhancing Jet Tree Tagging through Neural Network Learning of Optimal LundNet Variables

Machine learning has played a pivotal role in advancing physics, with deep learning notably contributing to solving complex classification problems such as jet tagging in the field of jet physics. In this experiment, we aim to harness the full potential of neural networks while acknowledging that, at times, we may lose sight of the underlying physics governing these models. Nevertheless, we demonstrate that we can achieve remarkable results obscuring physics knowledge and relying completely on the model's outcome. We introduce JetLOV, a composite comprising two models: a straightforward multilayer perceptron (MLP) and the well-established LundNet [93]. Our study reveals that we can attain comparable jet tagging performance without relying on the pre-computed LundNet variables. Instead, we allow the network to autonomously learn an entirely new set of variables, devoid of a priori knowledge of the underlying physics. These findings hold promise, particularly in addressing the issue of model dependence, which can be mitigated through generalisation and training on diverse data sets.

5.2 Bias in Synthetic Data

Jet tagging constitutes a fundamental yet intricate aspect of jet physics. Its primary objective is the accurate identification of high-energy particles responsible for initiating cascades, ultimately leading to the formation of particle clusters known as jets. Within this context, it is imperative to distinguish between various types of particles that can trigger such events. However, the complexity is amplified when dealing with extreme energy regimes, making the task of correct identification exceedingly challenging. Machine learning has made significant strides in aiding the physics community in improving models and achieving remarkable results in jet tagging. Some of the most recent state-of-the-art models, as discussed in Section 3.5, such as Particlenet [87] and LundNet [93] rely on Graph Neural Networks. Our focus in this experiment centres on assessing the performance of LundNet, a model that has excelled not only due to its complexity and sophistication but also because of the pre-processing of input data fed into the neural network.

LundNet utilises the Lund plane projection to transform the tree-like structure of jets into meaningful input features for each node in the tree [86]. By considering the four-momentum of the two particle's descendants, LundNet computes a set of five variables known as LundNet variables. These variables encode essential information about the energy distribution and flow within the tree structure. While it is often the case that more complex architectures yield superior classification performance, they often grapple with the issue of model independence. Recent findings, as presented at the BOOST 2023 workshop on behalf of the ATLAS collaboration [100], have underscored the challenges of generalisation for intricate taggers. These models tend to be sensitive to the specifics of synthetic data generation, where variations in simulation software, such as PYTHIA [73] and HERWIG [47], can significantly impact the outcomes; different softwares have their own algorithms for simulating physics processes, for example the parton shower and the hadronisation steps.

Motivated by the quest for model independence in physics, we introduce JetLOV, a composite of two models which are first trained separately and then put together for further training on the jet tagging: RegNet (a Multilayer Perceptron network) is the one responsible for learning new set of variables to feed into the second part of the model, LundNet, which is responsible for the tagging. JetLOV aims to discover a new set of variables that can yield state-of-the-art performance. This approach paves the way for future endeavours where RegNet can be leveraged to learn variable sets that are independent of the data type provided. Our objective is to train the model to achieve model independence, offering a promising path forward in the realm of physics modelling.

5.3 Experiment

The experiment is divided into two distinct steps. The initial step is dedicated to the training of the RegNet component (coded using Pytorch 1.7 [101]) and involves a regression task aimed at learning the five LundNet variables [86; 93], defined in Section 3.5.3.2, for each particle denoted as "k" based on the four-momentum vectors of their two descendants, "i" and "j":

$$(p^i_\mu, p^j_\mu) \rightarrow (\ln k_t, \ln \Delta, \ln z, \ln m, \psi)^k \tag{5.1}$$

RegNet, which takes the form of a MLP, is designed with five distinct branches, each responsible for independently learning one of the five variables. The branches handling "$\ln k_t$" and "$\ln m$" comprise a series of channels with dimensions (8, 128) and (128, 5), interconnected by ReLU activation functions. In contrast, the branches responsible for "$\ln \Delta$," "$\ln z$," and "ψ," which involve more intricate functions, employ a sequence of channels with dimensions (8, 128), (128, 128), and (128, 5), also incorporating ReLU activation functions for effective learning. This pre-training is performed in order to ensure that when we attach the second part of the architecture, which is already trained as well, we start not too far off the minimum. Our investigation confirms the necessity of this pre-trained part for achieving high performance, as the non pre-trained RegNet fails to discover such optimal local minima.

In the second step we attach RegNet to the pre-trained LundNet model. In this way when we feed the data into the first model, it produce the input for the second, which then produce the final output, i.e. the probability for classifying the type jet. Once the prediction is compared with the target, we do the back propagation all the way back in order to update the weights of the full model. The goal is to minimise the Cross Entropy Loss and achieve a good performance. We look at five metrics: accuracy, area under the ROC curve (AUC), background rejection at three signal efficiencies (0.3, 0.5 and 0.7).

5.4 Dataset

The data set is taken from [102]. In this project we worked only on the W-tagging, a standard binary classification problem, which consists of two classes: the signal (the W-jets, $pp \rightarrow WW$ process, and W required to decay hadronically) and the background (the QCD-jets, $pp \rightarrow jj$). Jets are clustered using the anti-kt algorithm [35] with a radius R = 1.0 using *FastJet 3.3.2* [5; 103], and are required to pass a selection cut, with transverse momentum $pt > 500\ GeV$ and rapidity $|y| < 2.5$. In each event,

only the two jets with the highest transverse momentum are considered, and are saved as training data if they pass the selection cuts.

5.5 Results

For the regression part, we trained the RegNet model with a data set of 100k events (50k signals and 50k backgrounds) and a smaller validation data set of 10k events. The training has been done over 100 epochs, until we reached a satisfying Mean Squared Error (MSE) Loss. In Figure 5.1 we can see the plots between target and prediction for each of the LundNet variables, the performance has been evaluated on a total number of particles of approximately 15000 and achieved a MSE of 0.037. All of them follow the expected linear trend, although it seems that the Neural Network struggles a bit on learning the third variable $\ln(z)$.

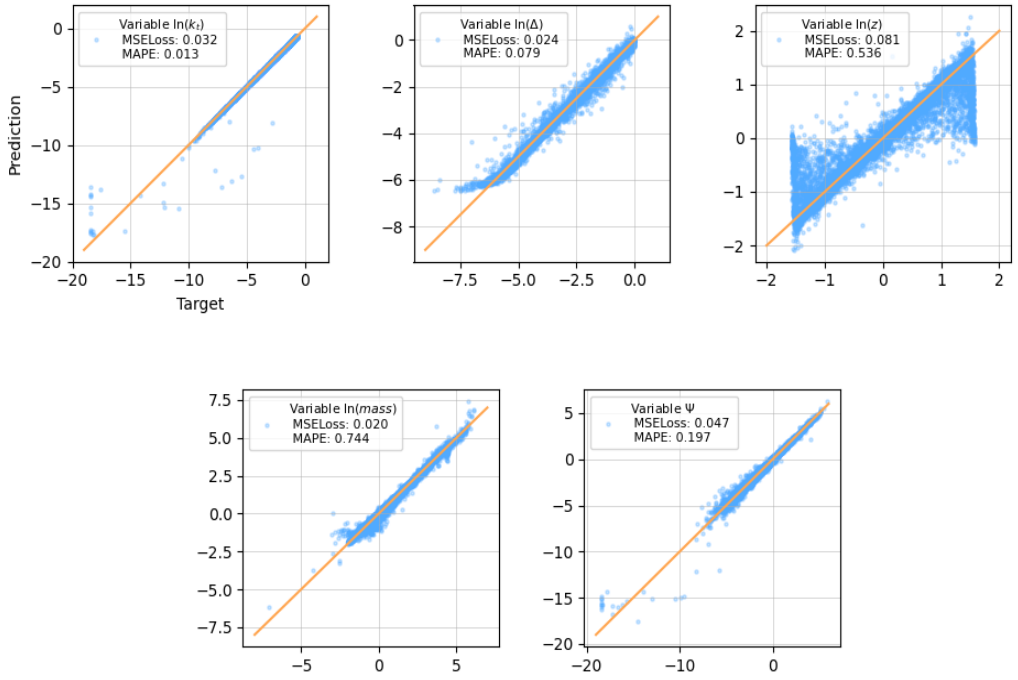


FIGURE 5.1: For each one of the LundNet variables we plot the prediction from the RegNet's outputs. We also report the Mean Squared Error (MSE) and the Mean Absolute Percentage Error (MAPE). The overall MSE for all the five variables is 0.037 on approximately 15000 points. Blu dots are the data points while the orange lines are the function $f(x) = x$.

For the classification, we have used an equally balanced data set of 1M events for the training, and 100k for the validation and 100k for the testing. The training, performed on a Nvidia GTX 1080 Ti graphics card with a minibatch size of 256, has been run for

30 epochs, with an initial learning rate of 0.001, and a scheduler that lowered the learning rate by a factor of 10 after the 20th and the 25th epochs. For the optimisation we have chosen to use Adam optimiser [104]. The best model is chosen when the highest validation accuracy has been reached. In Table 5.1, we can see that the models achieve same performances, with slightly preference of LundNet on higher signal efficiency and a preference of JetLOV for lower signal efficiency.

	AUC	Acc.	$1/\epsilon_B$ at ϵ_S=0.3	$1/\epsilon_B$ at ϵ_S=0.5	$1/\epsilon_B$ at ϵ_S=0.7
LundNet	0.938	0.872	4545.4	**602.4**	**70.2**
JetLOV	0.938	0.872	**6250.8**	555.6	63.9

TABLE 5.1: Results of several metrics for the two models (LundNet and JetLOV) for the W-tagging problem. The first column gives the area under the ROC curve, the second gives the accuracy, and the later three show the background rejection $(1/\epsilon_B)$ at three different signal efficiencies (ϵ_S), 30 %, 50 % and 70 % respectively. For each metric, larger values indicate better performance.

After training, our primary focus shifted towards the learned variables from the initial part of the model that feed into the second part. We aimed to assess the extent to which these learned variables differed from the original LundNet variables. To conduct this analysis, we employed Canonical Correlation Analysis (CCA) [105] to compare the original LundNet variables with the set learned by JetLOV.

CCA is a statistical method used to understand the relationship between two multivariate sets of variables. It finds linear combinations of the variables in each set such that the correlation between these combinations is maximized. Essentially, CCA identifies the directions in which the two sets of variables are most strongly correlated.

The results of this analysis, conducted on a dataset comprising 500 events, revealed a CCA value of 0.522, which indicates an absence of any significant correlation between the two sets of variables.

Delving deeper, we further investigated the matter using Singular Vector Canonical Correlation Analysis (SVCCA) [106] between the output layers of the fully trained LundNet model before and after its attachment to RegNet.

SVCCA extends CCA by combining it with Singular Value Decomposition (SVD). This approach not only considers linear correlations but also effectively handles high-dimensional data by reducing noise. SVCCA finds the most correlated directions (like CCA) but applies SVD first to reduce the dimensionality, making the analysis more robust, especially in neural network settings where high-dimensional activations are common.

This comparative analysis is visualised in Figure 5.2. The image, which is a matrix displaying all pairwise SVCCA values, shows no values close to 1 (excluding the input and the output layers), indicating little to no correlation between the parameters of

each layer of the network. This suggests that we have successfully identified a minimum point in our training process that meets the performance requirements for jet tagging. However, it is remarkable that the set of variables discovered at this minimum point deviates substantially from the original LundNet variables.

This intriguing outcome suggests that in certain cases, Machine Learning models can excel without relying on physics-derived features. While this may be advantageous as it hints at the potential for uncovering unconventional insights, there is a notable trade-off. The balance must be carefully struck between model interpretability and performance. Physics-based features are often interpretable, allowing researchers to comprehend the rationale behind a model's predictions. Conversely, when models learn novel features, they may become more opaque or 'black-box,' presenting challenges in interpreting their decision-making processes.

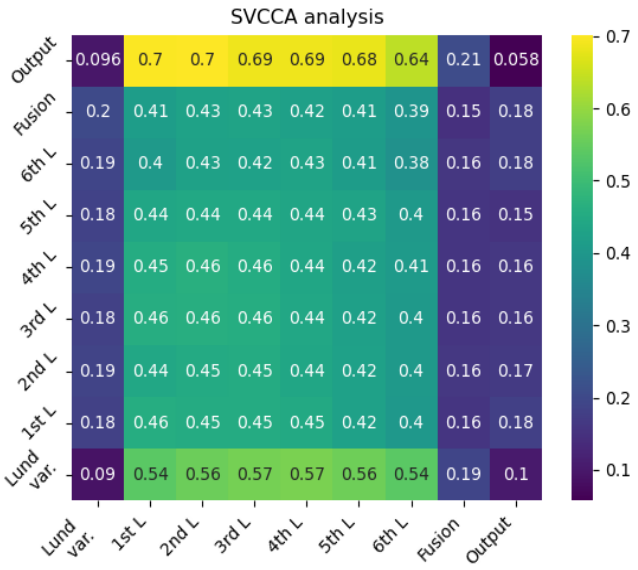


FIGURE 5.2: SVCCA analysis for each pair of the layers of the fully trained LundNet model before and after being combined with RegNet. Rows correspond to the model before being attached to RegNet, while cols correspond to the model after.

Chapter 6

Reducing the Bias of Machine Learning Models' Performance

In this chapter, we explore the application of a novel unsupervised learning method to reduce bias in machine learning models' performance when trained on synthetic data. By capturing symmetries in the data, this approach seeks to enhance the robustness of neural networks while mitigating biases introduced by varying simulation or physics models. Additionally, we examine the interplay between bias and generalisation, focusing on how domain shifts between training and test datasets impact model performance. Our investigation highlights the importance of reducing reliance on simulation-specific features to improve the applicability of machine learning models to real-world data.

6.1 Bias and Generalisation: Understanding Domain Shifts

Bias in machine learning models refers to systematic deviations in performance caused by differences in the training data distribution compared to the test or deployment scenarios. In the context of HEP, this bias often arises due to the use of synthetic datasets generated by different physics models or simulation tools. These differences can lead to significant performance degradation, particularly when the models encounter real-world data that may diverge from the training data distribution.

On the other hand, generalisation ability refers to a model's capacity to perform well on unseen data, particularly when the data distribution differs from the training set—a phenomenon often referred to as the domain shift problem. In HEP, domain shifts manifest when machine learning models are trained and tested on datasets generated using different showering algorithms or event generators. The goal is to

train models that remain stable and reliable across such shifts, enabling consistent performance across diverse physics scenarios.

Addressing bias and improving generalisation are closely intertwined but not identical tasks. Bias mitigation focuses on reducing the dependence on simulation-specific features or artefacts, while improving generalisation entails designing models or training schemes that can adapt to or remain robust against variations in the data distribution. Together, these efforts aim to produce models that can accurately interpret real-world experimental data despite the intrinsic differences from synthetic training datasets.

6.2 Comparison Between Twin Events

As mentioned in Section 5.2, complex and highly performing jet taggers often lack stability when trained and tested on data generated with different software or different physics models. This is a significant limitation, as ultimately, we seek a model that is robust when working with real-world data.

In our investigation, we focused on the impact of varying showering models within PYTHIA. Specifically, we generated samples using both the angular-ordered [15; 16] and p_T-ordered [17; 18] showering methods.

Angular-ordered parton showers arrange emissions based on the angle of emission relative to the initial parton direction. This approach is motivated by the concept of angular ordering, which aligns with colour coherence effects in QCD. Emissions in this model are naturally angular ordered due to colour coherence, providing a physical picture of QCD radiation that is particularly effective at simulating soft and collinear emissions. However, it can be less straightforward to accurately model harder emissions, where the transverse momentum (p_T) of the emissions becomes more critical.

In contrast, p_T-ordered parton showers arrange emissions based on the transverse momentum of the emitted parton. This method is more intuitive in terms of momentum scales and excels in simulating high transverse momentum processes and hard scatterings. The p_T-ordered approach offers a clearer separation of scales in parton emissions, making it particularly suitable for processes involving high transverse momentum. Nonetheless, this model may not fully capture the angular correlations induced by colour coherence, potentially affecting the accuracy of soft and collinear emission simulations.

Figure 6.1 illustrates the discrepancy in ROC curves for the same model (LundNet) trained five times on a dataset of 500k jets and tested on an unseen dataset of 50k jets.

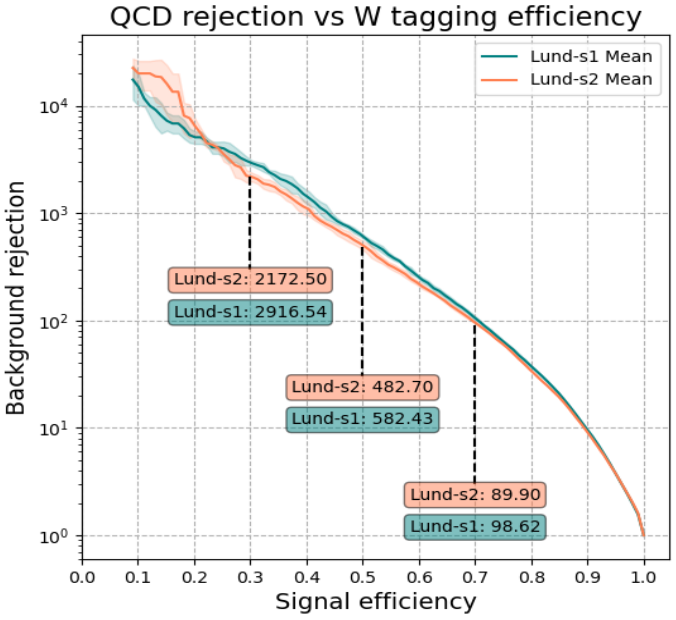


FIGURE 6.1: LundNet performance on two distinct showering models. The results are averaged over 5 runs per model.

The curves represent the averaged performance across the five training runs. To highlight the impact of the showering model, we have annotated the plot with the background rejection values at three key signal efficiencies: 0.3, 0.5, and 0.7.

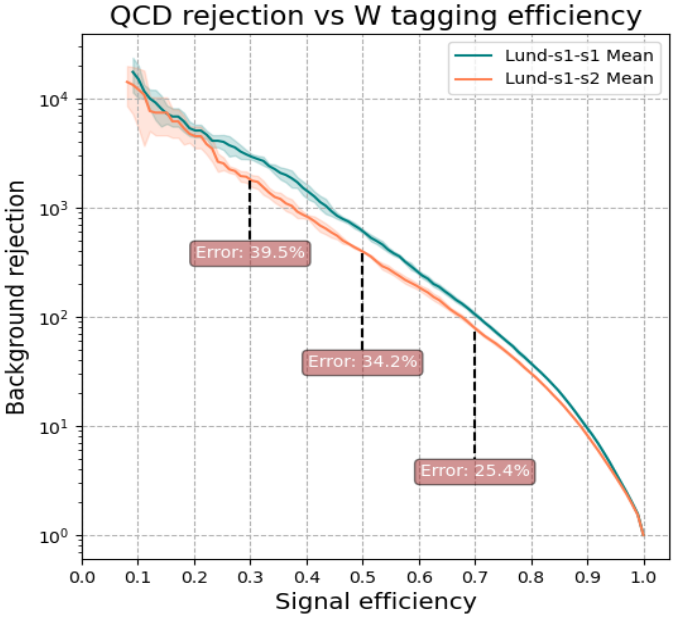


FIGURE 6.2: LundNet trained on dataset number 1 and then tested on dataset number 1 and dataset number 2. Results are averaged over 5 runs per model.

In contrast, Figure 6.2 illustrates the performance discrepancy of the LundNet architecture when trained on dataset number 1 and then tested on both datasets 1 and 2. Each curve represents the average performance across five independent training runs. We observe a significant percentage error between the two curves, indicating the model's sensitivity to the specific dataset used for training.

By analysing model performance across datasets generated with these two distinct showering methods, we aim to understand the extent to which domain shifts in simulation inputs affect bias in machine learning models. Our proposed unsupervised learning method leverages symmetries in the data to reduce simulation-specific dependencies, thereby improving generalisation across domain shifts. This approach not only mitigates bias but also enhances the adaptability of models to diverse datasets, bridging the gap between synthetic and real-world data.

6.3 Exploring Symmetries

Variance-Invariance-Covariance Regularisation (VICReg) [6] is a novel self-supervised learning method designed to explicitly prevent the collapse problem in representation learning, where models learn to produce trivial or constant outputs. This is achieved through two regularisation terms applied to the learned embedding. In contrast, popular alternative approaches like Contrastive Learning [107] demonstrate strong performance but require a large number of contrastive pairs for effective training. While Contrastive Learning has been successfully applied in jet physics through models like JetCLR [108], which leverages symmetries in jet data for tasks like jet tagging, VICReg offers a distinct advantage by not requiring negative samples or a large batch size.
The following results focus on adapting VICReg to GNNs. This adaptation necessitates four key components:

- **Augmented data** Each data point requires an augmented copy to provide diverse training examples.
- **Encoder** A neural network (in our case a GNN) that transforms input data into a latent representation.
- **Projector/Decoder**: A network that projects the latent representation into a higher-dimensional space, aiding in the separation of different features.
- **Loss Function**: A three-term loss function composed of:
  - Invariance: Minimises the mean squared distance between embeddings of augmented pairs, encouraging similar representations.

- Variance: Promotes diversity among embeddings of different samples within a batch.
- Covariance: Reduces correlations between different dimensions of the embeddings, further preventing collapse.

Method

VICReg utilises a joint embedding architecture, which can be either symmetric or asymmetric. Typically, a Siamese network architecture is used where two branches share weights and consist of an encoder f_θ and an expander h_ϕ. The encoder outputs representations for downstream tasks, while the expander maps these representations into an embedding space where the loss function is computed. The expander serves to eliminate differing information and expand the dimension non-linearly to reduce dependencies between variables of the representation vector. This can be seen in Figure 6.3.

Given a data point i sampled from a dataset D, two different views $x = t(i)$ and $x' = t'(i)$ are produced using random transformations. These views are encoded by f_θ into representations $y = f_\theta(x)$ and $y' = f_\theta(x')$, which are then mapped by h_ϕ to embeddings $z = h_\phi(y)$ and $z' = h_\phi(y')$.

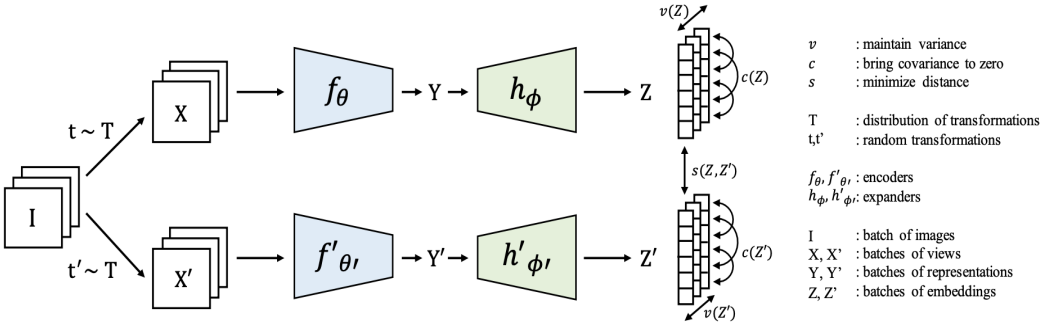


FIGURE 6.3: VICReg [6]: joint embedding architecture with variance, invariance and covariance regularisation. Given a batch of data points I, two batches of different views X and X' are produced and are then encoded into representations Y and Y'. The representations are fed to an expander producing the embeddings Z and Z'. The distance between two embeddings from the same data point is minimised, the variance of each embedding variable over a batch is maintained above a threshold, and the covariance between pairs of embedding variables over a batch are attracted to zero, decorrelating the variables from each other.

The loss is computed on these embeddings and consists of three terms: variance, invariance, and covariance.

The loss function in VICReg is defined as:

$$\mathcal{L}(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')] \tag{6.1}$$

where λ, μ, and ν are hyperparameters that control the contribution of each term to the total loss and $Z = [z_1, \ldots, z_n]$ and $Z' = [z'_1, \ldots, z'_n]$ denote batches of data points. The three terms are defined as follows:

- **Invariance Term**: This term ensures that the embeddings of augmented pairs are close to each other by minimising the mean squared distance between them.

$$s(Z, Z') = \frac{1}{N} \sum_{i=1}^{N} \|z_i - z'_i\|_2^2 \tag{6.2}$$

  where N is the number of samples, z_i and z'_i are the embeddings of the i-th sample and its augmented view, respectively.

- **Variance Term**: This term promotes diversity among embeddings of different samples within a batch by ensuring that the standard deviation of each dimension in the batch is above a given threshold γ.

$$v(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon)), \tag{6.3}$$

  where S is the regularised standard deviation defined by:

$$S(x, \epsilon) = \sqrt{Var(x) + \epsilon} \tag{6.4}$$

  where d is the dimensionality of the embeddings, and $\mathrm{Var}(z^j)$ is the variance of the j-th dimension of the embeddings z in the batch.

- **Covariance Term**: This term reduces correlations between different dimensions of the embeddings by minimising the sum of the squared off-diagonal elements of the covariance matrix.

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2 \tag{6.5}$$

  where $\mathrm{C}(Z)$ is the covariance matrix of Z (the indices run over the dimensions of the embeddings):

$$C(Z) = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \bar{z})(z_i - \bar{z})^T, \quad \text{where} \quad \bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i. \tag{6.6}$$

This comprehensive loss function ensures that the model learns meaningful and diverse representations without collapsing to trivial solutions.

In the next section, we will delve deeper into the application of these methods to jet tagging, exploring how VICReg can be adapted and optimised for use with Graph Neural Networks in this domain.

6.4 Experiment

In our work, we utilize a dataset of 500,000 events (balanced between signal and background) as augmented data. Crucially, groups of 2 jets within this dataset share the same hard scattering process, acting as "twin" samples. This differs significantly from the approach in [108], where data augmentation (e.g., rotations) is performed during training. In our case, the augmentation is done beforehand, due to the computationally expensive nature of re-showering events. Before delving into the complexity of using different re-showering methods, we generate the data re-showering the same hard scattering using a different seed. We do this with the two showering models, therefore we end up with two datasets: pt-ordered dataset and angular-ordered dataset.

For the encoder, we employ a stack of EdgeConv layers, proposed in [88], similar to the architectures used in ParticleNet [87] and LundNet [93]. Specifically, the EdgeConv blocks have (32, 32), (64, 64), (128, 128), and (128, 128) channels, respectively. These are followed by a two-layer Sequential network, with a ReLU in between, that expands the aggregated graph node features into a 256-dimensional space. The input data can be either 4-dimensional (for LundNet trees) or 5-dimensional (for KNN graphs).

A key strength of this unsupervised approach lies in its task formulation. We train the model to recognise "twin jets," or jets sharing the same hard scattering process. While learning this task, the model implicitly separates signal jets from background jets in the latent space. This separation allows us to achieve good tagging performance using only a simple linear classifier on top of the learned representations.

Model	**Acc.**	**AUC**	**1/ϵ_B at ϵ_S=0.3**	**1/ϵ_B at ϵ_S=0.5**	**1/ϵ_B at ϵ_S=0.7**
V-LundNet	0.888	0.944	463.3	183.3	46.2
V-ParticleNet	0.879	0.940	345.6	105.8	37.3

TABLE 6.1: Results of several metrics for the two models (V-LundNet and V-ParticleNet) for the W-tagging problem. The first column gives the area under the ROC curve, the second gives the accuracy, and the later three show the background rejection $(1/\epsilon_B)$ at three different signal efficiencies (ϵ_S), 30 %, 50 % and 70 % respectively. For each metric, larger values indicate better performance.

As shown in Table 6.1, the two models, V-LundNet and V-ParticleNet, were evaluated for the W-tagging problem, with V-LundNet outperforming V-ParticleNet across all metrics. We can see that passing as input LundNet trees, it clearly outperforms the other ways to pass the data as inputs, i.e. as plain graphs in the eta-phi space or passing tree with learnable features.

Although these are good results, they are not comparable to the state-of-the-art LundNet model, in fact, as we can see from Fig. 6.4, the supervised method, where we train on labels, outperforms the unsupervised one, where all we ask is to recognised pairs of twin-jets.

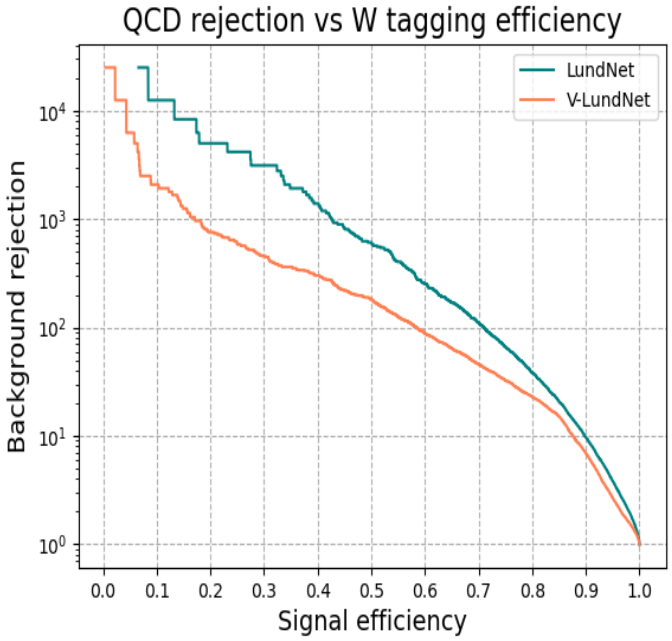


FIGURE 6.4: LundNet (supervised method) outperforms V-LundNet (unsupervised method).

6.5 Hyperparameter Tuning

Hyperparameter tuning is a critical step in the machine learning model development process. Unlike model parameters, which are learned during the training process, hyperparameters are set prior to the commencement of training and influence the behaviour and performance of the model. Properly tuning these hyperparameters can significantly improve the accuracy and efficiency of the model.

Hyperparameters are configurations that guide the training process of machine learning algorithms. They include aspects like learning rate, batch size, number of layers in a neural network, and more. Finding the optimal combination of these hyperparameters can be challenging due to the vast search space and the computational resources required.

Hyperparameter tuning involves systematically searching through a set of possible values for the hyperparameters and selecting the combination that yields the best performance on a validation set. The objective is to enhance the model's generalisation ability, ensuring it performs well on unseen data.

Several methods are commonly used for hyperparameter tuning, each with its own advantages and trade-offs. The most prevalent methods are grid search, random search, and Bayesian optimisation.

Grid Search is one of the simplest and most exhaustive hyperparameter tuning methods. It involves defining a finite set of values for each hyperparameter and evaluating the model performance for every possible combination of these values. Although grid search explores all possible combinations within the specified ranges, ensuring that the best combination is found, provided it lies within the grid, it is extremely computationally expensive. Moreover, it is quite inefficient as it may evaluate many combinations that do not contribute significantly to performance improvement.

Random Search [109], as the name suggests, involves randomly selecting values for each hyperparameter from specified distributions. The model is then evaluated on these randomly chosen combinations. It is more efficient than grid search, indeed it can often find good hyperparameter combinations with fewer iterations because it does not evaluate the entire search space. It is suitable for high-dimensional hyperparameter spaces. Random search may miss the optimal combination of hyperparameters, especially if the search space is vast and sparsely sampled.

Bayesian Optimisation [110] is a more sophisticated hyperparameter tuning method that builds a probabilistic model of the objective function and uses this model to select the most promising hyperparameters to evaluate. It systematically explores the search space and focuses on areas where the optimal hyperparameters are more likely to be found. Bayesian optimisation often finds the best hyperparameters in fewer iterations compared to grid and random search. It is more complex to implement and understand compared to grid and random search and it can be computationally expensive, especially for high-dimensional spaces.

In the context of our research, we can distinguish the hyperparameters into two sets: those aimed at improving the performance of the model, which refer to the architecture (e.g., learning rate, batch size, number of neurons), and those related to the stability of the training, specifically the scaling factors for the three terms in the loss function. In the initial stage of the project, our primary objective was to investigate the efficiency of the new methodology, so we focused solely on the second set of hyperparameters.

In the original work [6], the exact tuning procedure is not detailed. However, it is mentioned that to ensure stability in training, the value for the covariance loss should be very low compared to the other two, which are then chosen to be equal.

Since this is an unsupervised method, selecting the right metric can be quite challenging. Therefore, we decided to leverage the fact that in the second step, we could use the model to perform a classification task. With the help of the Optuna package [111], we conducted a simple random search for these three hyperparameters. We retained the best set of values whenever the trained architecture performed better on the classification dataset. After a few searches, we confirmed that the optimal

values for the three parameters are: 1 for the scaling factor for the covariance loss and 25 for the other losses.

6.6 Tuning on Smaller Datasets

Thus far, the unsupervised model has been trained solely on the task of recognising "twin jets" (jets sharing the same hard scattering process). To compare it against traditional supervised methods, we attached a single linear layer to learn the distinction between signal and background classes. Interestingly, this linear classifier does not require the full dataset used to train the underlying GNN; it achieves comparable performance with a significantly smaller subset. In fact, using only 10% of the original training set (50k jets instead of 500k) results in a negligible change in performance, as can be seen in Table 6.2.

Model	**Acc.**	**AUC**	**1/ϵ_B at ϵ_S=0.3**	**1/ϵ_B at ϵ_S=0.5**	**1/ϵ_B at ϵ_S=0.7**
V-LundNet-500k	0.888	0.944	463.3	183.3	46.2
V-LundNet-50k	0.888	0.944	438.4	162.9	45.0

TABLE 6.2: Results of several metrics for the V-LundNet models comparing tuning performance using the full dataset of 500k jets against a smaller subset of only 50k. As shown, there are negligible differences, highlighting the surprising advantages of this method for generalising across various types of data.

This approach of using smaller datasets for fine-tuning has significant implications for the generalisability of machine learning models in high-energy physics. By effectively leveraging pre-trained models and applying minimal fine-tuning, we can achieve robust performance across different datasets and physics simulations.

For instance, in our investigation, we focused on the impact of varying showering models within PYTHIA. Specifically, we generated samples using both the angular-ordered and p_T-ordered showering methods. If a model trained on jets produced by one showering method can accurately classify jets from another, it demonstrates strong generalisability. This is particularly important because different parton shower models can produce varying jet structures, and a model that performs well across these variations is more likely to be robust when applied to real-world data.

To further illustrate this, consider the scenario where additional data is generated using a different Monte Carlo generator, such as HERWIG. Instead of retraining the model from scratch on this new data, we can leverage the pre-trained model and apply fine-tuning with a relatively small amount of HERWIG-generated data. This process significantly reduces the computational resources and time required for training while maintaining high classification accuracy.

Such adaptability is critical in high-energy physics, where the ability to generalise across different datasets and simulation models can lead to more reliable and efficient data analysis. The use of unsupervised learning to pre-train models, followed by supervised fine-tuning on smaller datasets, represents a powerful strategy for developing robust jet taggers that perform well across a range of conditions and data sources.

By adopting this approach, we not only enhance the model's performance but also ensure that it remains flexible and capable of adapting to new data with minimal retraining. This is particularly useful in experimental physics, where obtaining large labelled datasets can be challenging and time-consuming.

This approach is closely related to transfer learning [112], a common technique in machine learning that involves transferring knowledge from one domain to another to reduce training complexity. Transfer learning allows a model trained on one task to be adapted for a different but related task with minimal retraining. In the context of high-energy physics, transfer learning has been demonstrated effectively in previous studies. For example, in [113], it was shown that a model trained on a top tagging dataset could be adapted for W-tagging without the need to retrain from scratch. This technique underscores the efficiency and versatility of transfer learning in developing adaptable and high-performing models in jet physics.

6.7 Exploring the Embedding Space

Figure 6.5 visualises the latent space representations learned by both LundNet and V-LundNet. The goal is to explore how well these models generalise across different datasets. We fed 10,000 jets from each of the two datasets into both models, even though the models were trained solely on dataset 1. The t-SNE dimensionality reduction algorithm [114] was used to project the 256-dimensional latent spaces down to two dimensions for easier visualisation.

t-SNE (t-distributed Stochastic Neighbour Embedding) is a machine learning algorithm specifically designed for visualising high-dimensional data by reducing it to two or three dimensions. It works by converting high-dimensional Euclidean distances into conditional probabilities representing similarities between data points. The algorithm minimises the divergence between these probability distributions in the original high-dimensional space and the low-dimensional projection, making it particularly effective for revealing structure and clustering similar data points together.

The figure shows a notable similarity in the overall structure of the latent spaces for both datasets, despite the models being trained only on dataset 1. This suggests that

the models capture generalisable features, explaining the small performance gap observed when switching datasets for classification. The analysis of these latent spaces gives us insight into how well the models can generalise across different data distributions, a key property for robust jet tagging.

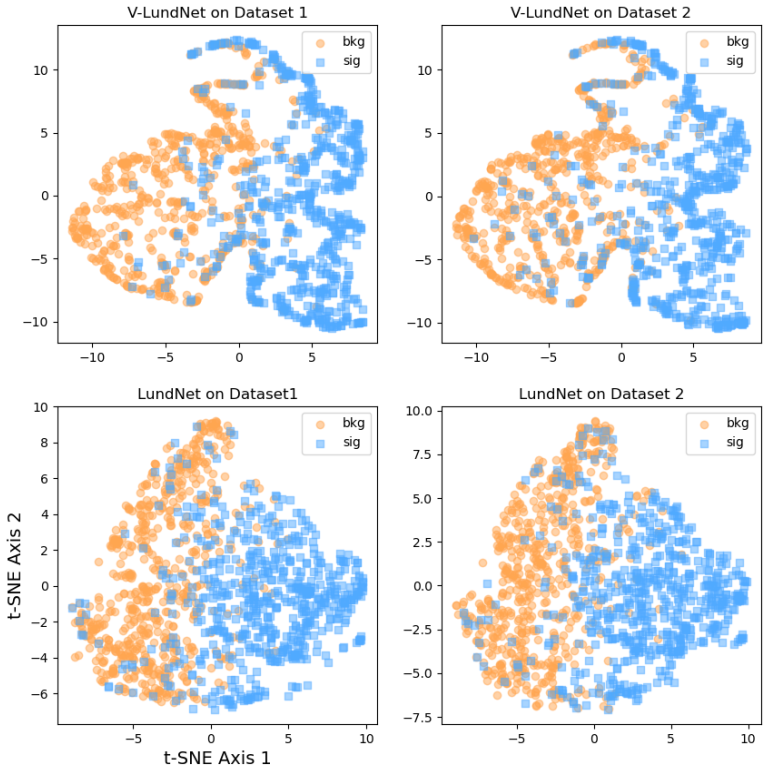


FIGURE 6.5: Projection of 10,000 jets from both dataset 1 (training set) and dataset 2 into the latent spaces learned by LundNet and V-LundNet. These 256-dimensional latent representations were reduced to two dimensions using t-SNE for visualisation. The figure highlights how the latent space structure remains consistent across datasets, suggesting that the learned representations generalise well beyond the training data.

6.8 Conclusion

In this chapter, we embarked on an initial exploration of a novel approach to mitigating bias in synthetic data using an unsupervised learning method, specifically through the application of VICReg to Graph Neural Networks. The results, though preliminary, are promising. While the performance in traditional jet tagging tasks has seen some reduction compared to state-of-the-art supervised methods, this trade-off is a potential pathway to reducing performance gaps caused by training on biased synthetic datasets.

The ability to learn from unlabelled data and fine-tune models on a minimal amount of labelled data is a crucial aspect of this solution. By leveraging a pre-trained model and applying targeted fine-tuning, we can significantly enhance generalisability across different datasets and simulation conditions, all while maintaining a robust performance.

This work represents just the beginning of our efforts to address the challenge of bias in synthetic data. There is ample room for improvement, and further refinement of this methodology may lead to more substantial reductions in performance disparities. As we continue to develop and refine these techniques, the ultimate goal remains to build models that not only perform well but are also resilient to the biases inherent in the data on which they are trained.

Part III

Summary and Final Comments

Chapter 7

Conclusions

In this thesis, we have delved into the intricate and fascinating world of jet physics, exploring the formation, clustering, and tagging of jets in high-energy particle collisions. Our work has underscored the transformative power of machine learning in enhancing our understanding of these complex phenomena, particularly through the application of GNNs. Our journey has traversed both foundational theory and cutting-edge research, culminating in the development of novel algorithms aimed at unravelling the secrets of the subatomic realm.

We began by establishing a solid theoretical framework for jets, tracing their origins from the fundamental interactions of quarks and gluons to the collimated sprays of hadrons observed in detectors. We delved into the mechanisms of parton showering and hadronisation, highlighting the inherent challenges and intricacies involved in accurately modelling these processes. The complexity of jet formation necessitates robust clustering algorithms, and thus, we examined traditional methods like the anti-k_T algorithm alongside more recent approaches based on spectral clustering. In doing so, we emphasised the importance of infrared and collinear safety, ensuring that our algorithms remain consistent with the underlying principles of QCD and particle physics.

The advent of machine learning, particularly GNNs, has revolutionised the field of jet physics. We have demonstrated how GNNs can effectively capture the complex relationships between particles within jets, leading to significant advancements in jet tagging and clustering tasks. Throughout this thesis, we explored state-of-the-art GNN architectures such as ParticleNet, LundNet, and EMPN, each with its unique strengths and applications. These models have shown remarkable performance in distinguishing between different types of jets, identifying the origins of particles, and even uncovering subtle anomalies that could hint at new physics beyond the Standard Model.

Our research has contributed to this exciting frontier by developing and rigorously evaluating novel algorithms for jet clustering and tagging. In Chapter 4, we introduced a spectral clustering method that operates in a transformed feature space, offering improved performance compared to traditional methods. This approach leverages the power of eigenvectors and eigenvalues to capture the intrinsic structure of particle collisions, resulting in more accurate and robust jet clustering. We also provided a comprehensive analysis of how different hyperparameters affect the algorithm's performance, offering valuable insights for future optimisations and practical implementations.

In Chapter 5, we delved into the integration of physics knowledge with machine learning for jet tagging tasks, introducing JetLOV—a model designed to learn optimal variables for LundNet without relying on pre-computed physics features. This work highlighted the ability of machine learning to discover effective representations of jet data, even without explicit guidance from traditional physics-based features.

In Chapter 6, we shifted our focus to methods aimed at reducing bias in GNN-based jet taggers. Here, we employed unsupervised learning techniques, specifically VICReg, to exploit the inherent symmetries in jet data. This approach proved effective in mitigating model dependence, ensuring consistent performance across different datasets and simulation models, and enhancing the generalisability of our algorithms.

Throughout this thesis, we have consistently emphasised the importance of balancing performance with adherence to fundamental physics principles. While machine learning offers powerful tools for data analysis, it is crucial to ensure that our models remain grounded in the underlying physics. This involves integrating physical constraints, such as infrared and collinear safety, into our algorithms and rigorously evaluating their behaviour in diverse scenarios. By striking this balance, we can harness the full potential of machine learning to advance our understanding of jet physics, ultimately leading to new discoveries in high-energy physics.

Looking ahead, the future of jet physics research is exceedingly bright. The continuous advancements in machine learning, coupled with the ever-growing data volumes from experiments like the LHC, promise to revolutionise our understanding of the subatomic world. As we continue to develop more sophisticated algorithms and explore innovative representations of jet data, we can expect to uncover deeper insights into the fundamental forces and particles that constitute our universe. The integration of machine learning with theoretical physics holds the potential to not only push the boundaries of current knowledge but also to open up entirely new avenues of exploration, leading to groundbreaking discoveries and an expanded understanding of the cosmos.

In conclusion, this thesis has made significant contributions to the field of jet physics by bridging the gap between traditional physical theories and modern machine

learning techniques. The methodologies and algorithms developed here provide a solid foundation for future research, ensuring that as we advance into the next era of particle physics, our tools remain as rigorous, interpretable, and effective as possible. The synergy between machine learning and high-energy physics will undoubtedly play a pivotal role in the next generation of scientific discoveries, helping us to decipher the profound mysteries of the universe.

Appendix A

Demonstration of Local Parton-Hadron Duality Using the Earth Mover's Distance in Jet Physics.

In this appendix, we present a straightforward and insightful exercise that illustrates the concept of local parton-hadron duality (LPHD), as discussed in Section 2.1.2. This exercise utilises a novel metric in jet physics, known as the Earth Mover's Distance (EMD) or Energy Mover's Distance, which has been explored in several studies [115–119]. The exercise was proposed by [7], following the introduction of EMD in jet physics by [21].

Intuitively, EMD is a metric that measures how "difficult" it is to move a pile of dirt from one location, A, to another, B. The lower the value, the easier the task. When applied to distributions, such as in jet physics where points are dispersed in space, it becomes non-trivial to quantify the difficulty or even determine the optimal way to redistribute energy from distribution A to distribution B. Figure A.1 shows an example of how to rearrange the energy of one jet to another. A specific variant of the EMD has been introduced, enabling comparison between events with different total energies. The EMD represents the minimum "work" required to transform one event E into another E_0 by transferring energy f_{ij} from particle i in one event to particle j in the other:

$$EMD(\epsilon, \epsilon') = \min_{f_{ij}} \sum_{ij} f_{ij} \frac{\theta_{ij}}{R} + \left| \sum_i E_i - \sum_j E'_j \right| \tag{A.1}$$

subject to the following conditions:

$$f_{ij} > 0, \quad \sum_j f_{ij} \leq E_i, \quad \sum_i f_{ij} \leq E'j, \quad \sum ijf_{ij} = E_{\text{min}}, \tag{A.2}$$

where i and j index particles in events ϵ and ϵ', respectively, E_i is the particle energy, θ_{ij} is an angular distance between particles, and $E_{min} = min(\sum_i E_i, \sum_j E'_j)$ is the smaller of the two total energies. R is a parameter that controls the relative importance of the two terms. Note that either energy or transverse momentum can be used.

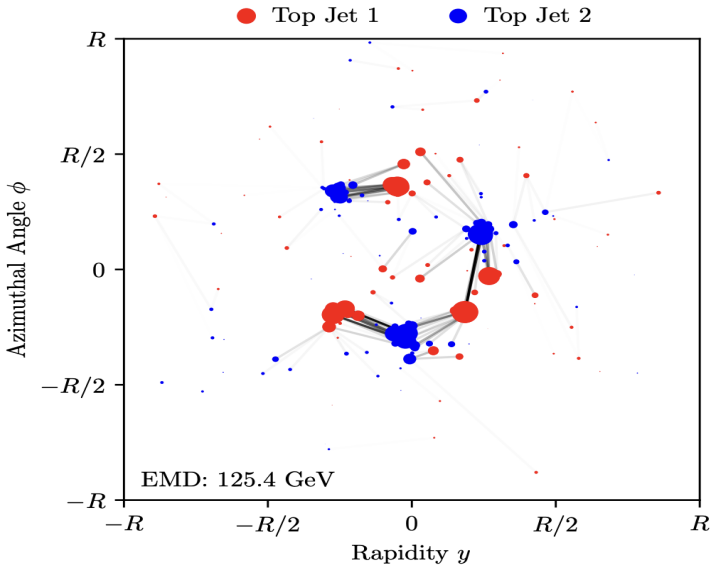


FIGURE A.1: The optimal movement to rearrange one top jet (red) into another (blue). Particles are shown as points in the rapidity-azimuth plane with areas proportional to their transverse momenta. Darker lines indicate more transverse momentum movement. The energy mover's distance is the total "work" required to perform this rearrangement [7].

The EMD can be related to additive IRC-safe observables through the Kantorovich-Rubinstein duality theorem [120]. By applying this theorem to the variant of EMD discussed here, it is possible to demonstrate how EMD constrains hadronisation modifications of jet angularities $\lambda = \sum_i p_{T,i}\theta_i$, where θ_i is the rapidity-azimuth distance to the jet axis. Consequently, we expect that:

$$\Delta\Lambda = |\lambda(\epsilon) - \lambda(\epsilon')| \leq EMD(\epsilon, \epsilon'). \tag{A.3}$$

To carry out this exercise, events are required at two stages of the generation process: before and after hadronization. Particle physics generator software, such as Pythia [45], can simulate the entire process asynchronously, providing access to both stages.

The dataset consists of proton-proton collision events at the LHC, generated using Pythia 8.235 [45] at $\sqrt{s} = 14$ TeV, including hadronisation and multiple particle

interactions. Anti-k_T jets [35] with a jet radius of 1.0 are clustered using FastJet 3.3.1 [103], and up to two jets with $p_T \in [500, 550]$ GeV and $|y| < 1.7$ are selected. This p_T range represents an intermediate energy level for jets at the LHC, allowing sensitivity to both terms in Eq. A.1. The jets are longitudinally boosted and rotated to centre the jet four-momentum at $(y, \phi) = 0$ and to vertically align the principal component of the constituent transverse momentum flow in the rapidity-azimuth plane. This step removes the dependence of the EMD on these jet isometries.

Figure A.2 presents the results obtained using the Python Optimal Transport library [121] to compute the EMDs between events.

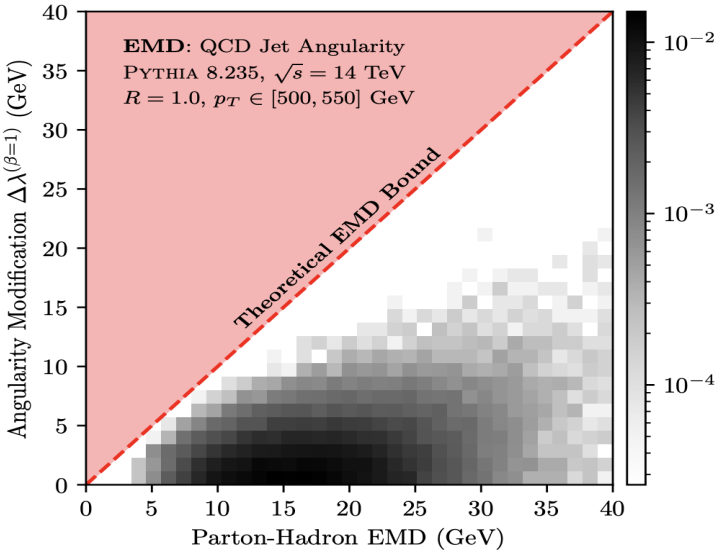


FIGURE A.2: Two-dimensional histogram of the EMD between 30k QCD jets before and after hadronisation versus the corresponding angularity modification. The red region is excluded based on the bound, shown as a dashed red line. The bound is clearly satisfied and is nearly saturated for EMD ≤ 10 GeV [7].

References

[1] R.K. Ellis, W.J. Stirling and B.R. Webber, *Fundamentals of qcd*, in *QCD and Collider Physics*, Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology, p. 1–21, Cambridge University Press (1996).

[2] CMS collaboration, *The CMS Experiment at the CERN LHC*, *JINST* **3** (2008) S08004.

[3] J. Cogan, M. Kagan, E. Strauss and A. Schwarztman, *Jet-images: computer vision inspired techniques for jet tagging*, *Journal of High Energy Physics* **2015** (2015) .

[4] P. Konar, V.S. Ngairangbam and M. Spannowsky, *Energy-weighted message passing: an infra-red and collinear safe graph neural network algorithm*, *Journal of High Energy Physics* **2022** (2022) .

[5] M. Cacciari and G.P. Salam, *Dispelling the $\mathcal{N}^3$ myth for the k_t jet-finder*, *Physics Letters B* **641** (2006) 57–61.

[6] A. Bardes, J. Ponce and Y. LeCun, *Vicreg: Variance-invariance-covariance regularization for self-supervised learning*, *arXiv preprint* (2022) [2105.04906].

[7] P.T. Komiske, E.M. Metodiev and J. Thaler, *The Hidden Geometry of Particle Collisions*, *JHEP* **07** (2020) 006 [2004.04159].

[8] G. Cerro, S. Dasmahapatra, H.A. Day-Hall, B. Ford, S. Moretti and C.H. Shepherd-Themistocleous, *Spectral clustering for jet physics*, *JHEP* **02** (2022) 165 [2104.01972].

[9] M.A. Diaz, G. Cerro, J. Chaplais, S. Dasmahapatra and S. Moretti, *JetLOV: Enhancing Jet Tree Tagging through Neural Network Learning of Optimal LundNet Variables*, in *Machine Learning for Physical Sciences workshop at the 37th Conference on Neural Information Processing Systems*, 11, 2023 [2311.14654].

[10] V.N. Gribov and L.N. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438.

[11] V.N. Gribov and L.N. Lipatov, *e^+e^- pair annihilation and deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 675.

[12] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, Nucl. Phys. B **126** (1977) 298.

[13] Y.L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e^+e^- Annihilation by Perturbation Theory in Quantum Chromodynamics.*, Sov. Phys. JETP **46** (1977) 641.

[14] S. Höche, *Introduction to parton-shower event generators*, 1411.4085.

[15] N. Armesto, G. Corcella, L. Cunqueiro and C.A. Salgado, *Angular-ordered parton showers with medium-modified splitting functions*, Journal of High Energy Physics **2009** (2009) 122–122.

[16] Z. Nagy and D.E. Soper, *Ordering variable for parton showers*, Journal of High Energy Physics **2014** (2014) .

[17] H. Brooks and P. Skands, *Coherent showers in decays of colored resonances*, Physical Review D **100** (2019) .

[18] H. Brooks, C.T. Preuss and P. Skands, *Sector showers for hadron collisions*, Journal of High Energy Physics **2020** (2020) .

[19] B. Cabouat and T. Sjöstrand, *Some dipole shower studies*, The European Physical Journal C **78** (2018) .

[20] Y.I. Azimov, Y.L. Dokshitzer, V.A. Khoze and S.I. Troyan, *The string effect and QCD coherence*, Physics Letters B **165** (1985) 147.

[21] P.T. Komiske, E.M. Metodiev and J. Thaler, *Energy flow networks: deep sets for particle jets*, Journal of High Energy Physics **2019** (2019) .

[22] L. Kantorovich and G.S. Rubinstein, *On a space of totally additive functions*, Vestnik Leningrad. Univ **13** (1958) 52.

[23] S. Larin and J. Vermaseren, *The three-loop qcd -function and anomalous dimensions*, Physics Letters B **303** (1993) 334.

[24] B. Webber, *Hadronization*, hep-ph/9411384.

[25] M.e.a. Derrick, *Measurement of α from jet rates in deep inelastic scattering at HERA*, Physics Letters B **363** (1995) 201–216.

[26] T. Brodkorb and E. Mirkes, *Complete $\mathcal{O}(\alpha_s^2)$ corrections to $(2+1)$ jet cross sections in deep inelastic scattering*, Zeitschrift für Physik C Particles and Fields **66** (1995) 141–149.

[27] T. Brodkorb and J.G. Korner, *Lepton - hadron correlations to $\mathcal{O}(\alpha_s^2)$ in $(2+1)$ jet production at electron - proton colliders*, Z. Phys. C **54** (1992) 519.

[28] E.A. De Wolf, A.T. Doyle, N. Varelas and D. Zeppenfeld, *QCD effects in hadronic final states*, *AIP Conf. Proc.* **407** (1997) 175 [hep-ex/9707038].

[29] D. Graudenz, *Jets and fragmentation*, *Journal of Physics G: Nuclear and Particle Physics* **25** (1999) 1289–1295.

[30] B. Webber, *A qcd model for jet fragmentation including soft gluon interference*, *Nuclear Physics B* **238** (1984) 492.

[31] G.F. Sterman and S. Weinberg, *Jets from Quantum Chromodynamics*, *Phys. Rev. Lett.* **39** (1977) 1436.

[32] JADE collaboration, *Experimental Studies on Multi-Jet Production in e^+e^- Annihilation at PETRA Energies*, *Z. Phys. C* **33** (1986) 23.

[33] S. Catani, Y.L. Dokshitzer, M. Olsson, G. Turnock and B.R. Webber, *New clustering algorithm for multi - jet cross-sections in e^+e^- annihilation*, *Phys. Lett. B* **269** (1991) 432.

[34] S.D. Ellis and D.E. Soper, *Successive combination jet algorithm for hadron collisions*, *Physical Review D* **48** (1993) 3160–3166.

[35] M. Cacciari, G.P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, *Journal of High Energy Physics* **2008** (2008) 063–063.

[36] Y. Dokshitzer, G. Leder, S. Moretti and B. Webber, *Better jet clustering algorithms*, *Journal of High Energy Physics* **1997** (1997) 001–001.

[37] D. Krohn, J. Thaler and L.-T. Wang, *Jets with variable R*, *Journal of High Energy Physics* **2009** (2009) 059–059.

[38] A. Chakraborty, S. Dasmahapatra, H. Day-Hall, B. Ford, S. Jain, S. Moretti et al., *Revisiting jet clustering algorithms for new Higgs Boson searches in hadronic final states*, *Eur. Phys. J. C* **82** (2022) 346 [2008.02499].

[39] A. Chakraborty, S. Dasmahapatra, H. Day-Hall, B. Ford, S. Jain and S. Moretti, *Fat b-jet analyses using old and new clustering algorithms in new Higgs boson searches at the LHC*, *Eur. Phys. J. C* **83** (2023) 347 [2303.05189].

[40] M. Srednicki, *Quantum Field Theory*, Cambridge University Press (2007).

[41] G.P. Salam, *Towards jetography*, *The European Physical Journal C* **67** (2010) 637–686.

[42] J.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)*, pp. 0134–136, 12, 1990.

[43] CMS collaboration, G.L.e.a. Bayatian, *CMS Physics: Technical Design Report Volume 1: Detector Performance and Software*, Technical design report. CMS, CERN, Geneva (2006).

[44] A.e.a. Sirunyan, *Identification of heavy-flavour jets with the cms detector in pp collisions at 13 tev*, *Journal of Instrumentation* **13** (2018) P05011–P05011.

[45] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An introduction to pythia 8.2*, *Computer Physics Communications* **191** (2015) 159–177.

[46] M.R. Aguilar, Z. Chang, R.K. Elayavalli, R. Fatemi, Y. He, Y. Ji et al., *Pythia 8 underlying event tune for rhic energies*, 2022.

[47] J. Bellm et al., *Herwig++ 2.7 Release Note*, *arXiv preprint* (2013) [1310.6877].

[48] R.D. Field, *The underlying event in hard scattering processes*, *arXiv preprint* (2002) [hep-ph/0201192].

[49] D. Bertolini, P. Harris, M. Low and N. Tran, *Pileup per particle identification*, *Journal of High Energy Physics* **2014** (2014) .

[50] D. Krohn, J. Thaler and L.-T. Wang, *Jet trimming*, *Journal of High Energy Physics* **2010** (2010) .

[51] S.D. Ellis, C.K. Vermilion and J.R. Walsh, *Techniques for improved heavy particle searches with jet substructure*, *Physical Review D* **80** (2009) .

[52] A.J. Larkoski, S. Marzani, G. Soyez and J. Thaler, *Soft drop*, *Journal of High Energy Physics* **2014** (2014) .

[53] G. Dissertori, F. Moortgat and M.A. Weber, *Hadronic Event-Shape Variables at CMS*, in *34th International Conference on High Energy Physics*, 10, 2008 [0810.3208].

[54] S. Chekanov, M. Demarteau, A. Fischer and J. Zhang, *Effect of pythia8 tunes on event shapes and top-quark reconstruction in e^+e^- annihilation at clic*, 1710.07713.

[55] ATLAS Collaboration collaboration, *Measurement of charged-particle event shape variables in inclusive $\sqrt{(s)}$=7 TeV proton-proton interactions with the atlas detector*, *Phys. Rev. D* **88** (2013) 032004.

[56] S. Brandt and H.D. Dahmen, *Axes and scalar measures of two-jet and three-jet events*, *Zeitschrift für Physik C Particles and Fields* **1** (1979) 61.

[57] J. Thaler and K. Van Tilburg, *Identifying boosted objects with n-subjettiness*, *Journal of High Energy Physics* **2011** (2011) .

[58] *CERN Storage - The Amount of Data to Collect*, .

[59] X. Ju and B. Nachman, *Supervised jet clustering with graph neural networks for lorentz boosted bosons*, *Physical Review D* **102** (2020) .

[60] J. Shlomi, P. Battaglia and J.-R. Vlimant, *Graph neural networks in particle physics*, *Machine Learning: Science and Technology* **2** (2020) 021001.

[61] J. Guo, J. Li, T. Li and R. Zhang, *Boosted higgs boson jet reconstruction via a graph neural network*, *Physical Review D* **103** (2021) .

[62] J. Li, T. Li and F.-Z. Xu, *Reconstructing boosted higgs jets from event image segmentation*, *Journal of High Energy Physics* **2021** (2021) .

[63] Y.-C.J. Chen, C.-W. Chiang, G. Cottin and D. Shih, *Boosted W and Z tagging with jet charge and deep learning*, *Physical Review D* **101** (2020) .

[64] J. Li and H. Sun, *An attention based neural network for jet tagging*, *arXiv preprint* (2020) [2009.00170].

[65] H. Qu, C. Li and S. Qian, *Particle transformer for jet tagging*, *arXiv preprint* (2024) [2202.03772].

[66] V. Belis, P. Odagiu and T.K. Aarrestad, *Machine learning for anomaly detection in particle physics*, *Reviews in Physics* **12** (2024) 100091.

[67] R. Baruah, S. Mondal, S.K. Patra and S. Roy, *Probing intractable beyond-standard-model parameter spaces armed with machine learning*, *arXiv preprint* (2024) [2404.02698].

[68] M.A. Diaz, G. Cerro, S. Dasmahapatra and S. Moretti, *Bayesian Active Search on Parameter Space: a 95 GeV Spin-0 Resonance in the (B − L)SSM*, 2404.18653.

[69] G. Malkomes, B. Cheng, E.H. Lee and M. Mccourt, *Beyond the pareto efficient frontier: Constraint active search for multiobjective experimental design*, in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, eds., vol. 139 of *Proceedings of Machine Learning Research*, pp. 7423–7434, PMLR, 18–24 Jul, 2021, https://proceedings.mlr.press/v139/malkomes21a.html.

[70] D.W. Hogg and D. Foreman-Mackey, *Data analysis recipes: Using markov chain monte carlo**, *The Astrophysical Journal Supplement Series* **236** (2018) 11.

[71] D. Luengo, L. Martino, M. Bugallo, V. Elvira and S. Särkkä, *A survey of monte carlo methods for parameter estimation*, *EURASIP Journal on Advances in Signal Processing* **2020** (2020) 25.

[72] M. Ebden, *Gaussian processes: A quick introduction*, *arXiv preprint* (2015) [1505.02965].

[73] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159 [1410.3012].

[74] R. Ciesielski and K. Goulianos, *MBR monte carlo simulation in PYTHIA8*, *arXiv preprint* (2012) [1205.1446].

[75] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al., *Generative adversarial networks*, *arXiv preprint* (2014) [1406.2661].

[76] D. Bank, N. Koenigstein and R. Giryes, *Autoencoders*, *arXiv preprint* (2021) [2003.05991].

[77] D.P. Kingma and M. Welling, *Auto-encoding variational bayes*, *arXiv preprint* (2022) [1312.6114].

[78] L. de Oliveira, M. Paganini and B. Nachman, *Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis*, *Computing and Software for Big Science* **1** (2017) .

[79] K. Dohi, *Variational autoencoders for jet simulation*, *arXiv preprint* (2020) [2009.04842].

[80] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, *arXiv preprint* (2015) [1511.08458].

[81] P.T. Komiske, E.M. Metodiev and M.D. Schwartz, *Deep learning in color: towards automated quark / gluon jet discrimination*, *Journal of High Energy Physics* **2017** (2017) .

[82] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, *Jet substructure classification in high-energy physics with deep neural networks*, *Physical Review D* **93** (2016) .

[83] J. Lin, M. Freytsis, I. Moult and B. Nachman, *Boosting $h \to b\bar{b}$ with machine learning*, *Journal of High Energy Physics* **2018** (2018) .

[84] G. Louppe, K. Cho, C. Becot and K. Cranmer, *QCD-aware recursive neural networks for jet physics*, *Journal of High Energy Physics* **2019** (2019) .

[85] A. Sherstinsky, *Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network*, *Physica D: Nonlinear Phenomena* **404** (2020) 132306.

[86] F.A. Dreyer, G.P. Salam and G. Soyez, *The lund jet plane*, *Journal of High Energy Physics* **2018** (2018) .

[87] H. Qu and L. Gouskos, *Jet tagging via particle clouds*, *Physical Review D* **101** (2020) .

[88] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein and J.M. Solomon, *Dynamic graph CNN for learning on point clouds*, *arXiv preprint* (2019) [1801.07829].

[89] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, *arXiv preprint* (2015) [1502.03167].

[90] X. Glorot, A. Bordes and Y. Bengio, *Deep sparse rectifier neural networks*, in *International Conference on Artificial Intelligence and Statistics*, 2011, https://api.semanticscholar.org/CorpusID:2239473.

[91] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, *arXiv preprint* (2015) [1512.03385].

[92] CMS Collaboration, *Boosted jet identification using particle candidates and deep neural networks. Available here*, .

[93] F.A. Dreyer and H. Qu, *Jet tagging in the lund plane with graph networks*, *arXiv preprint* (2021) [2012.08526].

[94] U. von Luxburg, *A tutorial on spectral clustering*, *arXiv preprint* (2007) [0711.0189].

[95] J.R. Lee, S.O. Gharan and L. Trevisan, *Multi-way spectral partitioning and higher-order cheeger inequalities*, *arXiv preprint* (2014) [1111.1055].

[96] M. Belkin and P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, *Neural Computation* **15** (2003) 1373.

[97] A. Chakraborty, S. Dasmahapatra, H. Day-Hall, B. Ford, S. Jain, S. Moretti et al., *Revisiting jet clustering algorithms for new higgs boson searches in hadronic final states*, *arXiv preprint* (2022) [2008.02499].

[98] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer and T. Stelzer, *Madgraph 5: going beyond*, *Journal of High Energy Physics* **2011** (2011) 128.

[99] P.e.a. Virtanen, *Scipy 1.0: fundamental algorithms for scientific computing in python*, *Nature Methods* **17** (2020) 261.

[100] M. LeBlanc, *Comparative performance of ATLAS boosted W taggers using different AI/ML algorithms*, .

[101] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, *Advances in neural information processing systems* **32** (2019) .

[102] S. Carrazza and F.A. Dreyer, *This is a git-lfs repository containing a range of jet-related data sets. this is available here*, .

[103] M. Cacciari, G.P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896 [1111.6097].

[104] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 1412.6980.

[105] D.R. Hardoon, S. Szedmak and J. Shawe-Taylor, *Canonical correlation analysis: An overview with application to learning methods*, *Neural Computation* **16** (2004) 2639.

[106] M. Raghu, J. Gilmer, J. Yosinski and J. Sohl-Dickstein, *Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability*, *arXiv preprint* (2017) [1706.05806].

[107] T. Chen, S. Kornblith, M. Norouzi and G. Hinton, *A simple framework for contrastive learning of visual representations*, *arXiv preprint* (2020) [2002.05709].

[108] B. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, *Symmetries, safety, and self-supervision*, *SciPost Physics* **12** (2022) .

[109] J. Bergstra and Y. Bengio, *Random search for hyper-parameter optimization*, *Journal of Machine Learning Research* **13** (2012) 281.

[110] J. Snoek, H. Larochelle and R.P. Adams, *Practical bayesian optimization of machine learning algorithms*, *arXiv preprint* (2012) [1206.2944].

[111] T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama, *Optuna: A next-generation hyperparameter optimization framework*, *arXiv preprint* (2019) [1907.10902].

[112] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu et al., *A comprehensive survey on transfer learning*, *arXiv preprint* (2020) [1911.02685].

[113] F.A. Dreyer, R. Grabarczyk and P.F. Monni, *Leveraging universality of jet taggers through transfer learning*, *The European Physical Journal C* **82** (2022) .

[114] T.T. Cai and R. Ma, *Theoretical foundations of t-sne for visualizing high-dimensional clustered data*, *arXiv preprint* (2022) [2105.07536].

[115] S. Peleg, M. Werman and H. Rom, *A unified approach to the change of resolution: space and gray-level*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11** (1989) 739.

[116] Y. Rubner, C. Tomasi and L.J. Guibas, *A metric for distributions with applications to image databases*, in *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, (USA), p. 59, IEEE Computer Society, 1998.

[117] Y. Rubner, C. Tomasi and L.J. Guibas, *The earth mover's distance as a metric for image retrieval*, *International Journal of Computer Vision* **40** (2000) 99.

[118] O. Pele and M. Werman, *A linear time histogram metric for improved sift matching*, in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr and A. Zisserman, eds., (Berlin, Heidelberg), pp. 495–508, Springer Berlin Heidelberg, 2008.

[119] O. Pele and B. Taskar, *The tangent earth mover's distance*, in *Geometric Science of Information*, F. Nielsen and F. Barbaresco, eds., (Berlin, Heidelberg), pp. 397–404, Springer Berlin Heidelberg, 2013.

[120] T.-Y. Hu and A.G. Hauptmann, *Multi-shot person re-identification through set distance with visual distributional representation*, in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, (New York, NY, USA), p. 262–270, Association for Computing Machinery, 2019, DOI.

[121] R.F. et al., *Pot: Python optimal transport*, *Journal of Machine Learning Research* **22** (2021) 1.