Collider signatures for Hidden Valley jets

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Pushkar Saoji

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

December, 2023

Supervisor: Dr. Nishita Desai

Certificate

This is to certify that this dissertation entitled Collider signatures for Hidden Valley jets towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Pushkar Saoji at Tata Institute of Fundamental Research, Mumbai under the supervision of Dr. Nishita Desai, Ramanujan Fellow, Department of Theoretical Physics , during the academic year 2023.

Dr. Nishita Desai

Committee:

Dr. Nishita Desai

Prof. Sourabh Dube

I dedicate this thesis to my parents and my sister, who have always supported and encouraged me in my academic pursuits. Their unconditional love and belief in me have been my greatest source of inspiration.

Declaration

I hereby declare that the matter embodied in the report entitled Collider signatures for Hidden Valley jets are the results of the work carried out by me at the Department of Theoretical Physics, Tata Institute of Fundamental Research, Mumbai, under the supervision of Dr. Nishita Desai and the same has not been submitted elsewhere for any other degree.

Pushkar Saoji

Acknowledgments

I would like to begin by expressing my sincere gratitude to my supervisor, Dr.Nishita Desai, for her unwavering guidance and invaluable support throughout this research project. Dr. Desai not only provided essential insights into the research process but also served as an inspirational mentor who continually encouraged and challenged me. I am fortunate to have had the privilege of working under her guidance.

I would like to express my sincere appreciation to Prof. Sourabh Dube, whose invaluable insights and guidance have been of paramount importance. Prof. Dube's willingness to assist and share his knowledge in response to my numerous queries has greatly contributed to the depth and quality of this research.

I extend my heartfelt appreciation to Deepesh and Ashish for their constant support. Deepesh has been a constant pillar of assistance and encouragement throughout this thesis, as well as the five years preceding it. Ashish's technical expertise and dedication to resolving Linux-related issues have been invaluable. Their contributions have significantly enriched the research journey and the successful completion of this thesis.

I would also like to express my gratitude to all my friends at IISER Pune, who have made my academic journey enjoyable and memorable. Although it's not possible to name each one individually, I will forever cherish the bonds and experiences we've shared.

Most importantly, I owe a debt of gratitude to my parents and my sister, whose firm belief in my abilities, and emotional encouragement have been the bedrock of my journey throughout these years.

The collective guidance and support of these individuals have played an indispensable role in the completion of this thesis. Their high-priority assistance and unwavering belief in

my capabilities have been invaluable.

Abstract

This thesis delves into the realm of Hidden Valley models, which introduce the intriguing concept of a new gauge group alongside the Standard Model gauge group, connected by a heavy mediator particle. Seeking to discern their signatures at the Large Hadron Collider (LHC), a binary classification MLP (Multi-Layer Perceptron) neural network serves as a key tool for signal discrimination. The introduction of the SIFT algorithm enhances the extraction of dark meson mass scales and discriminating the Hidden Valley jets against ordinary quark or gluon jets. We also introduce other benchmarks to explore sensitivity to heavy and light quark decays of Hidden Valley mesons. The use CNNs, with jet-images as inputs, offers an alternative avenue to study the properties of Hidden Valley jets. This comprehensive analysis unveils the potential to decipher elusive signals of the Hidden Valley.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Large Hadron Collider (LHC) is a proton-proton collider situated at CERN, operating at a center-of-mass energy of 13.6 TeV during its run 3. Within the data generated at the LHC, we seek clues that may uncover the secrets of physics beyond the Standard Model (BSM). A particularly intriguing avenue of exploration is the Hidden Valley model. This model predicts existence of a new gauge group in addition to SM gauge group, and connects the two sectors using a heavy mediator particle.

This thesis embarks on a quest to identify the distinctive signatures of the Hidden Valley model within simulated LHC data. To provide essential background information, this chapter lays the foundation with discussions of prerequisite concepts. In Chapter 2, we delve into the intricacies of the machine learning tools along with high-energy event generator PYTHIA 8. Chapter 3 explores our application of Neural Networks to distinguish Hidden Valley phenomena, accompanied by a presentation of the corresponding results. Chapter 4 serves as a summary of the outcomes observed in Chapter 3.

Looking ahead, Chapter 5 outlines the limitations encountered during this study and offers a glimpse into our future research plans.

1.1 Standard Model

The quest to understand the fundamental building blocks of the universe and the forces that govern their interactions has been one of the most profound scientific endeavors in human history. At the heart of this quest lies the Standard Model of particle physics, a remarkably successful theoretical framework that provides a comprehensive description of the elementary particles and fundamental forces that shape the universe.

The Standard Model describes the universe's building blocks as follows:

1. Quarks: Quarks are the constituents of protons, neutrons, and other strongly interacting particles. There are six flavors of quarks: up (u), down (d), charm (c), strange (s), top (t), and bottom (b).

2. Leptons: Leptons are another family of elementary particles, including electrons (e), muons (μ), taus (τ), and their associated neutrinos (ν_e, ν_μ, ν_τ).

3. Gauge Bosons: These are the force carriers that mediate the fundamental forces in the universe:

   - Photon (γ) mediates the electromagnetic force.
   - W and Z bosons ($W\pm, Z^0$) mediate the weak nuclear force responsible for processes like beta decay.
   - Gluons (g) mediate the strong nuclear force that binds quarks together inside protons, neutrons, and other hadrons.

4. Higgs Boson: The Higgs boson (H) is a unique particle associated with the Higgs field. It is responsible for endowing other particles with mass through the Higgs mechanism.

Standard Model follows a $SU(3)_C \times SU(2)_L \times U(1)_Y$ group. In this the $SU(3)_C$ group corresponds to QCD, and the 'C' in the subscript stands for 'color' charge. $SU(2)_L$ group corresponds to electroweak interaction and the $U(1)_Y$ corresponds to electromagnetic interaction.

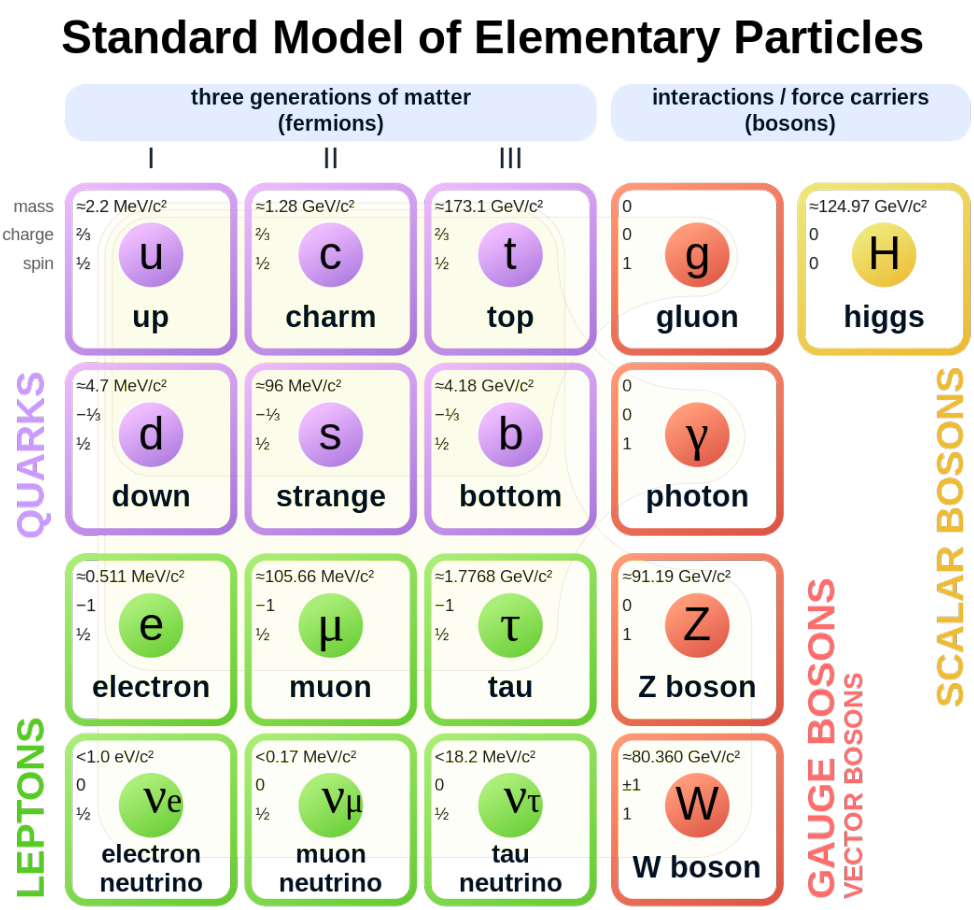


Figure 1.1: Particle content of Standard Model of Particle Physics

1.1.1 Quantum Chromodynamics (QCD)

QCD is the theory of the strong interaction between quarks mediated by gluons. QCD lagrangian is given by,

$$\mathcal{L}_{\text{QCD}} = -\frac{1}{4} G^a_{\mu\nu} G^{a,\mu\nu} + \sum_f \bar{q}_f (i\gamma^\mu D_\mu - m_f) q_f \tag{1.1}$$

Where,

$$D_\mu = \partial_\mu - i g_s \frac{\lambda^a}{2} A^a_\mu,$$
$$\text{and } G^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g_s f^{abc} A^b_\mu A^c_\nu,$$

- ∂_μ and ∂_ν are partial derivatives with respect to spacetime coordinates.
- g_s is the strong coupling constant, which characterizes the strength of the strong force.
- λ^a are the Gell-Mann matrices, which are generators of the SU(3) color group for QCD. They are indexed by a and represent the color charge of quarks and gluons.
- A^a_μ is the gluon field, representing the eight different gluon fields (indexed by a), which are the carriers of the strong force.

In the Lagrangian, the first term describes the kinetic energy of the gluons, which are the gauge bosons of the strong force. a is an index representing the eight different gluon color charges. $G^a_{\mu\nu}$ is the gluon field strength tensor. And the second term represents the kinetic and mass terms for quarks. f runs over the different flavors of quarks (up, down, strange, charm, bottom, and top). $\bar{q}_f$ and q_f are the quark fields and their antiparticles, respectively. $i\gamma^\mu D_\mu$ is the Dirac operator coupled to the gluon field, which describes the interaction between quarks and gluons. m_f is the mass of the quark.

Peculiar properties of QCD:
Color confinement is the property which states that quarks and gluons are never found in isolation, and are forever bound within color-neutral particles known as hadrons. And Asymptotic freedom refers to a steady reduction in the strength of interactions between quarks and gluons as the energy scale of those interactions increases.

The Standard Model of particle physics has proven to be a remarkably successful theory, describing the fundamental particles and their interactions with astonishing precision. However, it leaves some crucial questions unanswered, including the existence of 'Dark Matter'. Hidden Valley is a model that offers a potential solution to this mystery.

1.2 Hidden Valley

There are two broad candidates of physics Beyond Standard Model (BSM). One in which you have heavy particles with the coupling of the order of standard model, and the other one, in which there are light particles which do not have any SM charge. These particles

couple to SM through heavy mediator particles carrying both the SM charge and the charge of the new gauge group. Hidden Valley models [1–10] lie in the second category.

In the hidden valley scenarios, SM gauge group is extended by a gauge group G. This new gauge group can be a broken $U(1)$ (also denoted as $\not{U}(1)$) or an unbroken $SU(N)$. The gauge boson corresponding to $\not{U}(1)$ case is a light γ_v and the one corresponding to $SU(N)$ case is massless g_v. For this study, the new gauge group is SU(3).

The particle content consists of two kind of particles, q_v and F_v. F_v particles are charged under both SM gauge group and the new gauge group G, and q_v particles are charged only under the new gauge group. For this study q_v is spin 1/2 and F_v is spin 0. 12 F_v particles are introduced which mirror the SM fermions. And two pure dark quarks q_v's are introduced, named q_{v1} and q_{v2}

Dark sector production is through a Z_v resonance. Z_v boson is charged under both SM and HV gauge group.

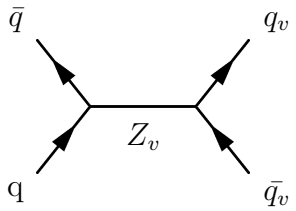
The interaction term between the mediator Z_v and the SM quarks is given by,

$$\mathcal{L} = g_{SM}\ \bar{q}\gamma_\mu q {Z_v}^\mu, \tag{1.2}$$

whereas for the Z_v and the HV quarks is given by,

$$\mathcal{L} = g_{HV}\ \bar{q_v}\gamma_\mu q_v {Z_v}^\mu, \tag{1.3}$$

Pair production of dark quarks is through the following process,



We also need a mechanism for dark sector decaying back to SM. This is achieved through a fraction of dark mesons decaying to SM quarks $q\ \bar{q}$. For this, the diagonal dark meson are considered to decay back to SM, whereas off-diagonal dark mesons are considered to be stable. This gives rise to jets with both visible and invisible content. These are called

semi-visible jets[11]. This results in significant missing energy along one of the jets.

Mesons of interest here are π_v's and ρ_v's dark sector partners to π and ρ mesons. By helicity conservation an extra m_q^2 factor comes in the π_v to $q\ \bar{q}$ coupling compared to ρ_v to $q\ \bar{q}$. Hence the branching ratios for π_v is proportional to squared mass of the decay quarks, whereas ρ_v decays are democratic.

For simplicity all the HV quarks are chosen to have same mass, and all the mesons are also chosen to have same mass.

1.3 LHC Experiments and Detector Schematic

In collider physics, the accurate detection and measurement of particles produced during high-energy collisions are crucial for understanding the fundamental processes that govern the universe. The LHC, located at CERN (European Organization for Nuclear Research), is the world's largest and most powerful particle accelerator. It consists of 9 detectors, which includes CMS, ATLAS, LHCb, ALICE and 5 other specialised and small scale detectors. I will briefly explain LHCb and ALICE before moving on to CMS and ATLAS, which are more relevant for our analysis.

LHCb's primary goal is to unravel the mysteries of CP violation, a phenomenon that explains why there is more matter than antimatter in the universe. By studying the decays of particles containing bottom and charm quarks, LHCb seeks to understand the fundamental differences in the behavior of matter and antimatter.

The primary objective of the ALICE experiment is to study and recreate the conditions that prevailed in the universe microseconds after the Big Bang. During this early phase, the universe existed in a state of extreme temperature and energy density, wherein protons and neutrons had not yet formed, and quarks and gluons—the fundamental building blocks of matter—roamed freely. ALICE aims to recreate this state of matter, known as quark-gluon plasma (QGP), by colliding heavy ions, such as lead nuclei, at ultra-relativistic speeds. These collisions generate temperatures and energy densities that are orders of magnitude higher than those typically encountered in everyday matter. By studying QGP, ALICE seeks to answer fundamental questions about the universe's evolution, the behavior of matter under

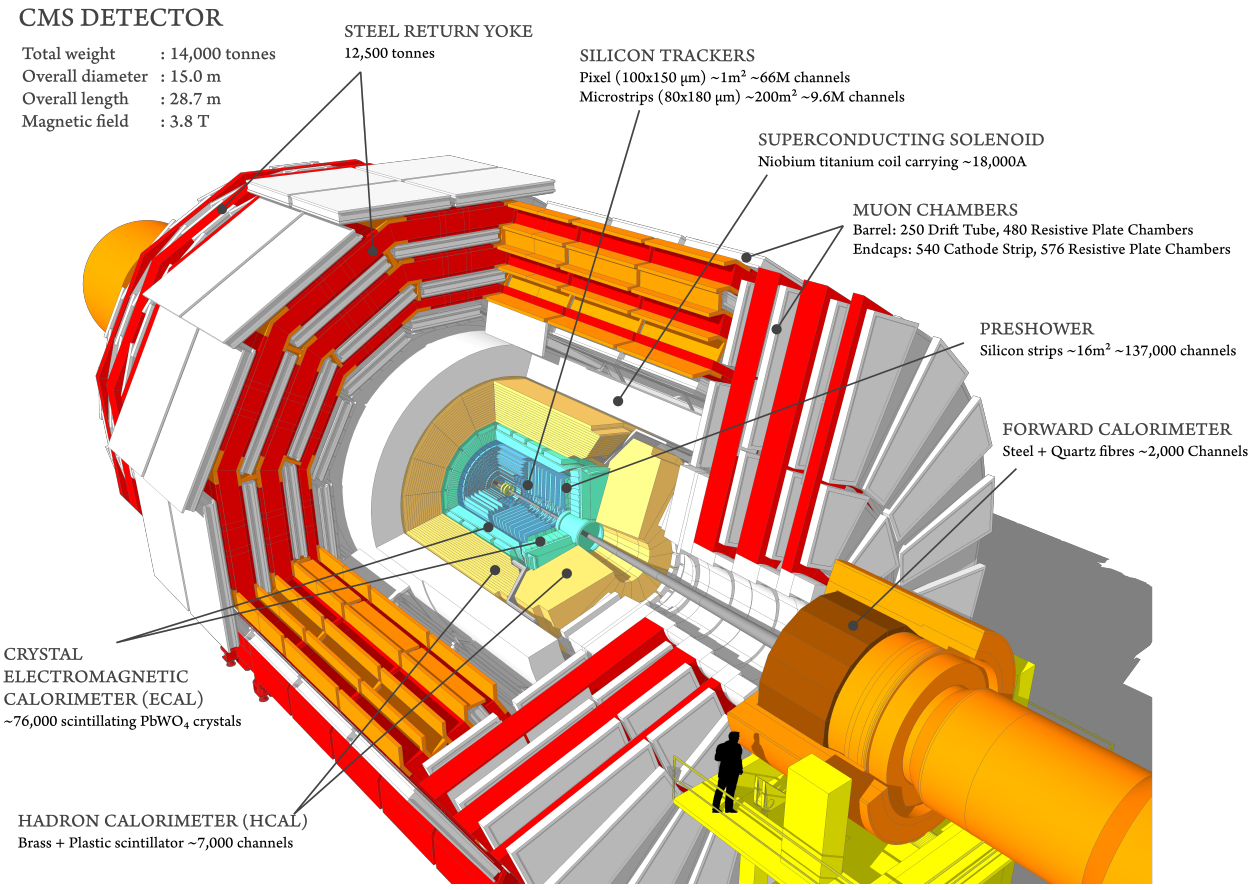


Figure 1.2: CMS Detector

(Source: cms.cern)

extreme conditions, and the strong force that governs quarks and gluons.

The CMS and ATLAS are general purpose detectors. We will be trying to understand the detector schematic of these detectors in this section.

The key components of the detector schematic include:

Inner Tracking System:

The innermost layer of the detector is the Inner Tracking System. It is responsible for precisely measuring the trajectories of charged particles as they emerge from collisions. This information is vital for determining particle momenta and identifying their types.

Calorimeters:

Calorimeters are used to measure the energy of particles. The detector includes two types of calorimeters:

- **Electromagnetic Calorimeter (ECAL)**: Primarily designed to measure the energy of electrons and photons.
- **Hadron Calorimeter (HCAL)**: Designed to measure the energy of hadrons, including protons, neutrons, and other strongly interacting particles.

Magnet Systems:

Magnet systems, including superconducting magnets, are used to bend the paths of charged particles, allowing for their momenta to be determined based on the curvature of their trajectories.

Muon Detectors:

Muons are relatively penetrating particles that can traverse through several layers of detectors. Specialized muon detectors are strategically placed at the outermost layer of the detector to identify muons and measure their momentum accurately.

Trigger System:

Given the immense rate at which collisions occur in the LHC, it is impossible to record and analyze every event in real-time. The trigger system is responsible for selecting the most interesting events for further analysis based on predefined criteria.

A detailed understanding of the detector components is essential for the analysis of Hidden Valley jet signatures and the development of discrimination methods. In the following sections, we will explore the jet algorithms used for reconstructing particle jets and various observables employed for characterizing these jets in the detector environment.

1.3.1 Jet algorithms

In the high-energy environments created at particle accelerators like the LHC, high-velocity protons or other heavy ions collide with each other, generating a cascade of subatomic

interactions. These collisions produce a myriad of particles, often including quarks and gluons, which are the fundamental constituents of matter and carriers of the strong nuclear force. Due to the nature of the strong force, quarks and gluons cannot exist as isolated particles but instead undergo a process known as hadronization. This results in the creation of sprays of particles moving in roughly the same direction, which are called jets.

Understanding the behavior of these jets is essential for making precise measurements and testing theoretical predictions in particle physics. This is where jet algorithms come into play.

Jet algorithms are a set of computational techniques and procedures designed to identify and characterize the individual jets originating from high-energy collisions. Jet algorithms have evolved and diversified over the years to meet the increasingly demanding requirements of modern particle physics experiments [12].

Infrared and Collinear (IRC) Safety

Infrared and collinear safety are required properties of self-consistent jet clustering algorithms. These properties ensure the robustness and reliability of the algorithms in identifying and characterizing jets.

Infrared Safety: Infrared safety refers to the insensitivity of the algorithm to soft (low-energy) particles or radiation. In high-energy collisions, many low-energy particles are produced as a result of the underlying QCD processes.

An infrared-safe jet algorithm is one that does not produce significantly different jet structures when soft particles are added to the event or when they are removed. In other words, the algorithm shouldn't be significantly affected by the presence or absence of low-energy radiation.

Collinear Safety: Collinear safety, on the other hand, relates to the behavior of the jet algorithm when particles are closely aligned in direction, forming collinear configurations. In particle collisions, quarks and gluons can radiate additional particles that are almost parallel in direction to the parent particle.

A collinear-safe jet algorithm should produce stable and consistent jet reconstructions

regardless of whether closely aligned particles are treated as a single constituent or separate constituents

IRC safety ensures that running the jet algorithm on event generators and detector outputs gives analogous results.

1.3.2 Rapidity and Psuedorapidity

Rapidity(w): It is commonly used as a measure of relativistic velocity. It is defined as,

$$w = arctanh(v/c) \tag{1.4}$$

Thus,

$$\cosh w = \cosh\left(\operatorname{arctanh}\frac{v}{c}\right) = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} = \gamma$$

$$\sinh w = \sinh\left(\operatorname{arctanh}\frac{v}{c}\right) = \frac{\frac{v}{c}}{\sqrt{1-\frac{v^2}{c^2}}} = \beta\gamma,$$

As we know,

$$E = mc^2 \cosh w$$
$$|\mathbf{p}| = mc \sinh w.$$

We get,

$$w = \operatorname{arctanh}\frac{|\mathbf{p}|c}{E} = \frac{1}{2}\ln\frac{E+|\mathbf{p}|c}{E-|\mathbf{p}|c}\ .$$

But in collider physics we define rapidity in a slightly different way. We use rapidity w.r.t beam axis (y), i.e $|\mathbf{p}|$ is replaced by p_z.

$$y = \frac{1}{2}\ln\frac{E+p_z c}{E-p_z c}.$$

Psuedorapidity η: It is defined as,

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right],$$

Where, θ is the angle between 3 momenta of the particle and beam axis.

$$\eta = \frac{1}{2} \ln \left(\frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z} \right)$$

In the limit, $m \ll |\mathbf{p}| \Rightarrow E \approx |\mathbf{p}| \Rightarrow \eta \approx y$. Therefore, both these terms are used equivalently in collider physics.

1.3.3 Anti-k_T

The anti-k_T algorithm [13] is a widely used jet clustering algorithm in particle physics. The anti-k_T jet algorithms work as follows:

1. For each protojet (i) and each pair of protojets (i, j) a distance measure is defined as

$$d_{ij} = \min \left(\frac{1}{p_{T,i}^2}, \frac{1}{p_{T,j}^2} \right) \frac{\Delta R_{ij}^2}{R^2} \tag{1.5}$$
$$d_i = \frac{1}{p_{T,i}^2},$$

   R is called radius of the jet as it roughly decides the size of the jet. And $\Delta R_{i,j} = \sqrt{(\eta_j - \eta_i)^2 + (\phi_j - \phi_i)^2}$, where, η_i is the pseudorapidity and ϕ_i is the azimuthal angle of the particle i. To initialize the algorithm, input particles are taken as protojets.

2. Find smallest of all d_{ij} and d_i and call it d_{min}

3. If d_{min} is for pair i, j then combine the two protojets i, j to form a new protojet k. If d_{min} is d_i for protojet i, then remove i from list of protojets and add it to list of jets.

4. Repeat until the list of protojets is empty.

To understand the functionality of anti-k_T algorithm, consider an event with few well-separated hard particles and large number of soft particles. The d_{ij} between a hard particle and a soft particle depends exclusively on transverse momenta of the hard particle and ΔR_{ij} between the two particles. Importantly, this distance measure does not take into account the transverse momentum of the soft particle. It is also worth noting that d_{ij} is significantly larger for pairs of soft particles. Therefore, before any soft-soft clustering takes place, all the

hard-soft clustering should have take place. So, for a hard particle, if there were no hard particle within $2R$ neighbourhood, it would produce a perfectly conical jet. If another hard particle exists in within distance R to $2R$, such that $p_{T2} \ll p_{T1}$, then the harder particle will produce a conical jet and the softer one will produce a partly conical jet. From these we can see that, in general anti-k_T algorithm will usually produce jets that are closer to conical shape.

Note that, in some cases I have also used C-A[1] jet algorithm, specially while reclustering jet constituents to obtain subjets.

I have used SlowJet within PYTHIA 8 to implement anti-k_T (and C-A) algorithms. Throughout this project, $R = 0.7$ has been used for clustering jets using anti-k_T algorithms, while $R = 0.3$ has been used for subjets reclustered using C-A algorithm. '`p` = -1' gives anti-k_T and '`p` = 0' gives C-A. There are a few more parameters that can be set while using SlowJet. `'pTjetMin'` puts a cut on minimum p_T required for an entity to be called a jet. So the list of jets will contain only those jets satisfying this condition. `'etaMax'` puts an upper cut on the η of the input particles to be considered as inputs, i.e the input particles with $|\eta| >$ `etaMax` are ignored. This condition is required keeping in mind the limitations on the range of the detectors.

SIFT

SIFT (Scale-Invariant Filtered Tree) [14] is a jet clustering algorithm that was introduced to maintain the resolution of substructure for collimated decay products at large boosts. One of the main drawbacks of anti-k_T (and most other algorithms) is that it requires radius of a jet (R) as parameter, which defines the scale for the algorithm. Whereas in case of SIFT, as the name suggests, the scale is not defined using any external parameter.

The main factor responsible for the dependence of traditional jet clustering methods on the conjugate momentum scale is the specification of an angular size parameter R. In order to develop a scale-independent algorithm, it is necessary to eliminate this factor from the clustering measure. However, we also want it to mimic successful approaches like anti-k_T in the asymptotic limit. Therefore, desirable behaviors include giving preference to pairs

[1]C-A algorithm works in the same way as anti-k_T, with the only difference being, distance measure is defined as $d_{ij} = \frac{\Delta R_{ij}^2}{R^2}$ and $d_i = 1$

of particles that are close in angular proximity and favoring pairs where one particle has a large transverse boost. To fulfill the former objective, the concept of mass-square difference is introduced, which is defined as follows:

$$\begin{aligned}\Delta m_{AB}^2 &\equiv (p_\mu^A + p_\mu^B)^2 - m_A^2 - m_B^2 = 2p_\mu^A p_\mu^B \\ &\simeq 2E^A E^B \left(1 - \cos\Delta\theta_{AB}\right) \simeq E^A E^B \Delta\theta_{AB}^2 \end{aligned} \tag{1.6}$$

The latter objective is achieved through suppression of denominator by the summed transverse energy-square:

$$\begin{aligned}\sum (E_T)^2 &\equiv (E_T^A)^2 + (E_T^B)^2 \\ (E_T)^2 &\equiv p_T^2 + m^2 = E^2 - p_z^2 \end{aligned} \tag{1.7}$$

Putting it together, the measure is given by

$$\delta_{AB}^{SIFT} = \frac{\Delta m_{AB}^2}{(E_T^A)^2 + (E_T^B)^2} \tag{1.8}$$

After that the procedure is somewhat similar to anti-k_T, i.e to find the pair with smallest value of measure and to sequentially combine it.

But there are a few issues that needs to be taken care of,

- Extraneous wide and soft radiation is assimilated very early, which in turn distorts the kinematic reconstruction
- Without any halting condition, it will end up clustering all the particles

For the first point, we can take a hint from 'SoftDrop' to solve the issue, which, as it turns out, also takes care of the second point.

SoftDrop iteratively declusters C-A jets, dropping the softer objects till the following condition is met,

$$\frac{\min(p_{T,i}, p_{T,j})}{p_{T,i} + p_{T,j}} > z_{\text{cut}} \left(\frac{\Delta R_{ij}}{R}\right)^\beta, \tag{1.9}$$

where R is the jet radius.

Before going further, let us first factorize the SIFT measure by defining 2 other quantities,

$\Delta\widetilde{R}^2_{AB}$ and ϵ^{AB}, such that $\delta_{AB} = \epsilon^{AB} \times \Delta\widetilde{R}^2_{AB}$,

$$\Delta\widetilde{R}^2_{AB} \equiv \frac{\Delta m^2_{AB}}{E^A_T E^B_T} \tag{1.10}$$

This factor plays the role of angular factor, i.e it promotes the merger at small angular separation.

$$\epsilon^{AB} \equiv \frac{E^A_T E^B_T}{(E^A_T)^2 + (E^B_T)^2} = \left\{ \left(\frac{E^A_T}{E^B_T}\right) + \left(\frac{E^B_T}{E^A_T}\right) \right\}^{-1}. \tag{1.11}$$

Notice that this has a maxima when $E^A_T = E^B_T$, $\epsilon^{AB} = 1/2$. And decreases as scale disparity increases. Therefore, this factor will try to defer the clustering of mutually hard pair.

Clustering condition for SIFT is given as follows:

$$\frac{\Delta\widetilde{R}^2_{AB}}{2} < \left\{ \left(2\,\epsilon^{AB}\right) \leq 1 \right\} \tag{1.12}$$

This condition can fail in 2 ways,

1. The transverse scales may be too hierarchically separated, i.e ($\epsilon^{AB} \ll 1$) and ($\Delta\widetilde{R}^2_{AB} \sim 1$) such that ($\delta_{AB} \ll 1$). In this case we would want put aside the softer candidate (Drop).

2. Angular opening may be too wide, i.e ($\Delta\widetilde{R}^2_{AB} \gg 1$) and ($\epsilon^{AB} \sim 1$). In this case appropriate thing to would be to consider both the candidates as jets (Isolate).

Thus, we get the following conditions:

$$\begin{aligned} \text{Drop:} \quad & \left\{ \left(2\,\epsilon^{AB}\right)^2 \leq 1 \right\} \leq \delta_{AB} < \left\{1\right\} \\ \text{Isolate:} \quad & \left\{1\right\} \leq \delta_{AB} \end{aligned} \tag{1.13}$$

Figure 1.3 gives visual intuition to the above criteria. The blue region stands for $1 \leq \delta_{AB}$, where isolation is to be performed. Dropping is to be performed in the red region, and the clustering criteria is satisfied by the green region.

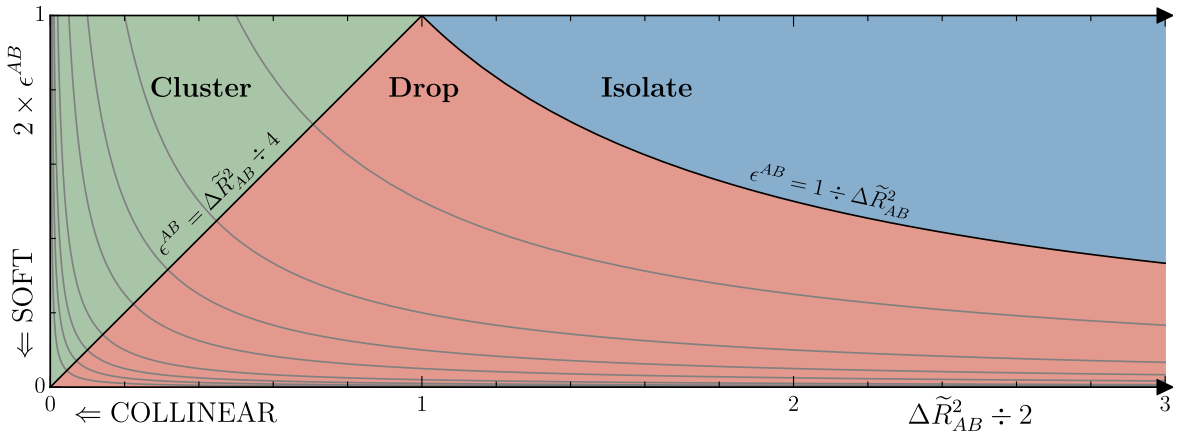


Figure 1.3: Phase diagram displaying the conditions for object merging, filtering, and isolation responses.

Comparison

I have compared the p_T and Radius of the jets formed by using anti-k_T and SIFT algorithms. For this comparison QCD samples were used and `pTjetMin` is set to 50 GeV for both the case. 'R' for anti-k_T is set to 0.7.

Radius of the jet is defined here as, the maximum value of ΔR between a jet and its constituents. The comparison plots can be seen in Figures 1.4, 1.5 and 1.6. From this it can be seen that the SIFT jets are much thinner than the anti-k_T jets. It can also be seen that the SIFT algorithm is much more efficiency in extracting dark meson mass.

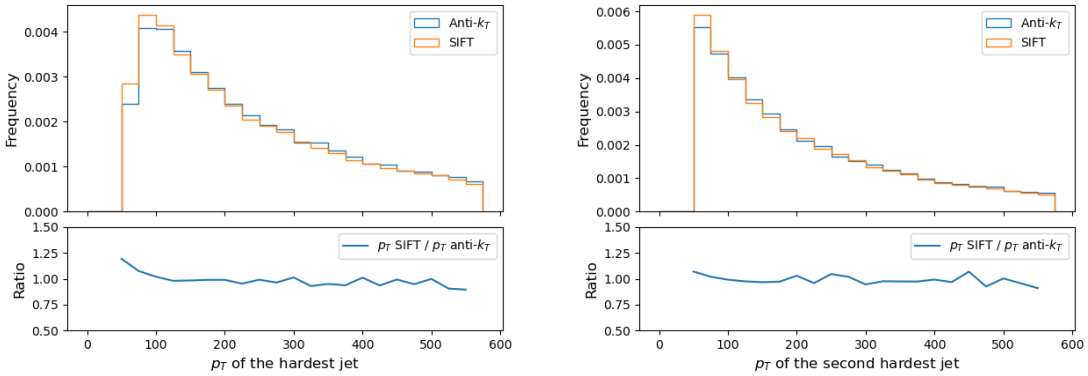


Figure 1.4: Comparing p_T of the jets clustered with anti-k_T algorithm and SIFT algorithm, for the two hardest jets

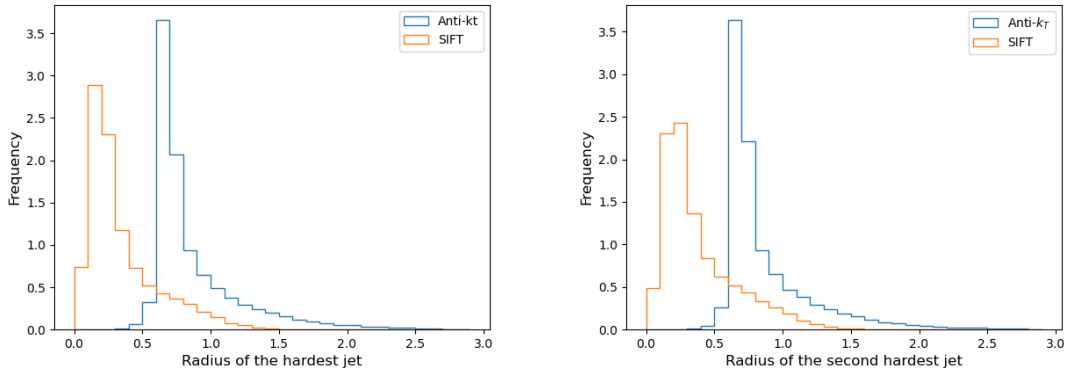


Figure 1.5: Comparing radius of the jets clustered with anti-k_T algorithm and SIFT algorithm, for the two hardest jets

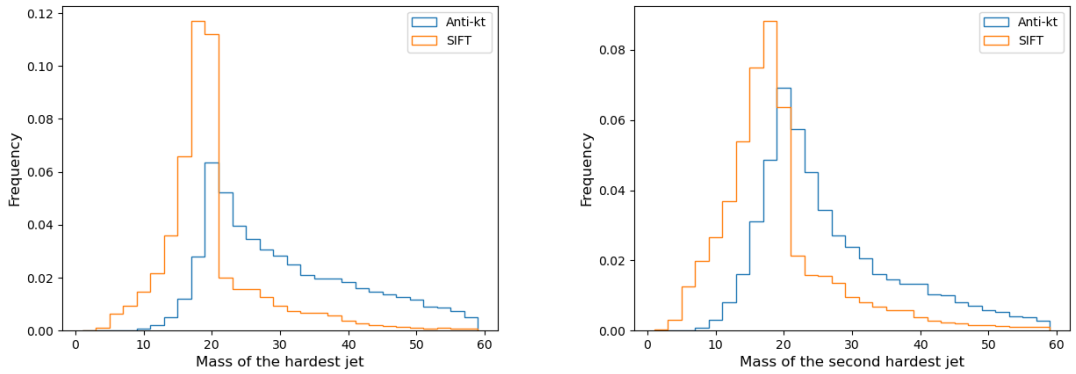


Figure 1.6: Comparing invariant mass of the jets clustered with anti-k_T algorithm and SIFT algorithm, for the two hardest jets

1.3.4 Other observables

Missing Transverse Energy (MET)

MET is a concept in collider physics, which represents the imbalance in visible transverse momentum in an event. In searches for new particles or interactions beyond the Standard Model, MET is a crucial discriminant. The presence of undetected, massive particles, such as dark matter candidates, can manifest as significant MET signatures.

The MET is calculated as the negative vector sum of the transverse momenta (p_T) of all

visible particles in the event:

$$MET = -\sum_i \vec{p}_T^{\,(i)} \tag{1.14}$$

Where, $\vec{p}_T^{\,(i)}$ is the transverse momentum vector of the i-th visible particle.

Invariant mass

Invariant mass is a fundamental concept in particle physics used to investigate the properties and dynamics of particle systems. It remains unchanged under Lorentz transformations. For a particle, this is also called rest mass, as it gives the mass of the particle at rest. The invariant mass (m) is given by:

$$m^2 = p^\mu p_\mu \tag{1.15}$$

Where, for a system of particles (like jets),

$$p^\mu = \sum_{i \in \{particles\}} p_i^\mu,$$

p_i^μ being p^μ (4-momentum) of the particle i. In the context of jets, the term mass is used interchangeably with invariant mass.

Energy-Correlation Functions

Energy-Correlation Functions (ECFs) are used to characterize the internal structure and substructure of jets. It provides valuable insights into the distribution of energy within a jet.

Jets exhibit distinct substructures depending on the nature of the initiating parton or their parent particle. For example, quark jets, originating from quarks, tend to have a more localized energy distribution. While, gluon jets, originating from gluons, typically exhibit a broader energy distribution.

The Energy Correlation Function aims to quantify these differences by examining the energy distribution within a jet, specifically focusing on the arrangement of subjets and their energy-sharing patterns. It is a powerful discriminator in identifying the underlying

parton of a jet and distinguishing signal jets from background.

The Energy Correlation Function for a jet is defined as follows:

$$e_2^\beta = \sum_{i<j\in jet} z_i z_j \times \Delta R_{i,j}^\beta, \tag{1.16}$$

$$e_3^\beta = \sum_{i<j<k\in jet} z_i z_j z_k \times \Delta R_{i,j}^\beta \Delta R_{j,k}^\beta \Delta R_{i,k}^\beta, \tag{1.17}$$

.

.

.

$$e_N^\beta = \sum_{i_1<\ldots<i_N\in jet} \left(\prod_{j=1}^{N} z_{i_j}\right)\left(\prod_{k<l=1}^{N} \Delta R_{i_k,i_l}^\beta\right), \tag{1.18}$$

Where,

- $z_i = \frac{p_{T,i}}{p_T^{\text{jet}}}$
- n is the number of subjets to consider.
- $p_{T,i}$ represents the transverse momentum of the i-th subjet.
- p_T^{jet} is the transverse momentum of the entire jet.
- β is an exponent that controls the sensitivity to subjet energy sharing.
- $\Delta R_{i,j}$ is the angular separation between the i-th and j-th subjets.

The ECFs can be used to characterize the substructure of Hidden Valley jets. Deviations from typical quark and gluon jet ECF distributions may indicate the presence of hidden particles. We have used $\beta = 1$ throughout this thesis, as this is the most commonly used value in literature.

Lund Jet Plane (LJP)

Since we use a sequential recombination algorithm to cluster jets, it is in principle possible to reverse the clustering history to obtain values of momenta at each radiation splitting. This

is captured in the 'Lund Jet Plane'. Since the clustering history of the anti-k_T algorithm is not the reverse of QCD showering, the jet has been reclustered using C-A algorithm with same radius.

To plot the LJP [15], we cluster first with anti-k_T ($R = 0.8$), recluster the constituents with C-A and uncluster the last step. We refer to the softer protojet as emission and the harder protojet as core. Then, k_T is defined as transverse momentum of emission w.r.t the core. At each step, note down the k_T and the angular distance ($\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2}$) between emission and core. Next, consider the harder protojet and continue with unclustering its last step, continuing to note down k_T and ΔR. Follow the same procedure till no protojets remain. I then plot the $ln(k_T/GeV)$ vs $ln(R/\Delta R)$, each point on the plot corresponding to one unclustering step.

The goal of this exercise is to determine if the components of HV jets (which come from the decays of HV mesons) have a different clustering history compared to QCD jets (where the clustering history mimics QCD emissions). The LJP's for `HV2` and `QCD` cases can be seen in figure 1.7. We can see that there is some difference in large-R emissions in `HV2` case compared to `QCD` case.

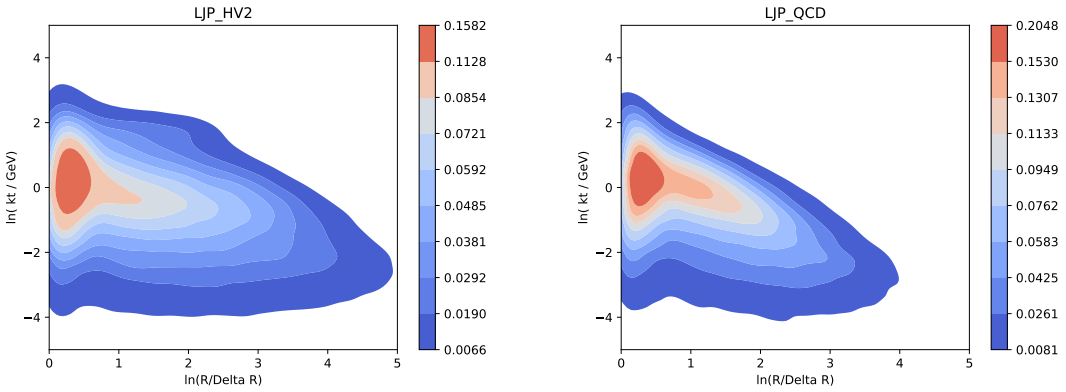


Figure 1.7: Comparing LJPs for `HV2` (on the left) and `QCD` (on the right)

Chapter 2

Computational Setup

2.1 PYTHIA 8

PYTHIA 8 [16] is a tool that is widely used for the generation of events in high-energy collisions between particles, where the QCD effects are highly important. In particle physics, an 'event' refers to the outcome of a collision between two incoming particles or the isolated decay of a particle. At its core, an event comprises a collection of outgoing particles that could be observed in an idealized detector snapshot.

I have used PYTHIA 8.3 to generate QCD and HV events.

2.1.1 Event Generation

QCD

For Generating QCD events, following settings are used:

- '`Beams:idA`' and '`Beams:idB`' are used to determine the particles to collide. Both are set to 2212, which is the PDG ID [1] for protons. This is to say that both the colliding particles are protons, as is the case for LHC.

[1] PDG ID is a unique code given to each type of particle by Particle Data Group

- '`Beams:eCM=13600`' sets the energy scale of the collision. It is set to 13600 GeV, the energy at LHC for run 3.
- '`HardQCD:all=on`' turns on all QCD $2 \to 2$ quark/gluon production processes.
- '`pTHatMin=15`' sets lower cut on transverse momentum of the interacting partons. It is set to 15 GeV. As these processes diverge at $p_T \to 0$, it is necessary to put some lower cut.
- '`PhaseSpace:bias2Selection=on`' It switches on biased phase space sampling for $2 \to 2$.
- '`PhaseSpace:bias2SelectionPow`' and '`PhaseSpace:bias2SelectionRef`' are bias parameters and are set to 4.5 and 100 respectively. To get suitable p_T distribution for the QCD sample[2].

As, p_T scale is model dependent, we don't want the Machine Learning techniques to discriminate based on p_T distributions of the model. Therefor to remove dependence of p_T from the analysis, bin-wise matching of p_T distributions was done. For this, the histograms were plotted for QCD and HV with bin size 25 GeV. And ratio of the value of the bins for HV and the value of the bins for QCD was stored as weights. QCD events within each bin were then accepted with probability equal to the weight of that bin. After doing this exercise, the p_T distributions were closely aligned, and hence dependence on the p_T distributions from the analysis was removed. The resulting sample will be called bin-wise matched QCD, and it will be understood that it is matched with corresponding HV counterpart under consideration.

HV

For generating HV, I have used different benchmarks, common settings include,

- '`HiddenValley:ffbar2Zv=on`' turns on pair production of HV quarks via Z_v.
- '`HiddenValley:FSR = on`' , turns on final state radiation in hidden sector.

[2]This will not affect the analysis as bin-wise normalization is applied anyway. This just allows to achieve bin-wise normalization with generating fewer number of events

		Decay to				
		$u\ \bar{u}$	$d\ \bar{d}$	$s\ \bar{s}$	$c\ \bar{c}$	$b\ \bar{b}$
Mesons	rhov11	0.2	0.2	0.2	0.2	0.2
	rhov22	0.2	0.2	0.2	0.2	0.2
	rhov21	0	0	0	0	0
	piv11	0	0	0	0.089	0.911
	piv22	0	0	0	0.089	0.911
	piv21	0	0	0	0	0

Decay to				
$u\ \bar{u}$	$d\ \bar{d}$	$s\ \bar{s}$	$c\ \bar{c}$	$b\ \bar{b}$
0.333	0.333	0.333	0	0
0.333	0.333	0.333	0	0
0	0	0	0	0
0.333	0.333	0.333	0	0
0.333	0.333	0.333	0	0
0	0	0	0	0

Table 2.1: Decay modes for `HV3` benchmarks with dark pions decaying to heavy SM quarks (left), and to light SM quarks (right)

		Decay to					
		$\nu_e\ \bar{\nu}_e$	$u\ \bar{u}$	$d\ \bar{d}$	$s\ \bar{s}$	$c\ \bar{c}$	$b\ \bar{b}$
Mesons	rhov11	$\frac{r}{1-r}$	0.2	0.2	0.2	0.2	0.2
	rhov22	$\frac{r}{1-r}$	0.2	0.2	0.2	0.2	0.2
	rhov21	$\frac{r}{1-r}$	0.2	0.2	0.2	0.2	0.2
	piv11	$\frac{r}{1-r}$	0	0	0	0.089	0.911
	piv22	$\frac{r}{1-r}$	0	0	0	0.089	0.911
	piv21	$\frac{r}{1-r}$	0	0	0	0.089	0.911

Table 2.2: Decay modes for `HV3` benchmarks with fixed r_{inv} ($r_{inv} = r$) and dark pions decaying to heavy SM quarks (HQ case)

		Decay to					
		$\nu_e\ \bar{\nu}_e$	$u\ \bar{u}$	$d\ \bar{d}$	$s\ \bar{s}$	$c\ \bar{c}$	$b\ \bar{b}$
Mesons	rhov11	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0
	rhov22	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0
	rhov21	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0
	piv11	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0
	piv22	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0
	piv21	$\frac{r}{1-r}$	0.333	0.333	0.333	0	0

Table 2.3: Decay modes for `HV3` benchmarks with fixed r_{inv} ($r_{inv} = r$) and dark pions decaying to light SM quarks (LQ case)

- '`HiddenValley:fragment = on`' turns on string fragmentation of the HV partonic system. So hadrons can form.

- '`PartonLevel:ISR=off`' , turns off initial state radiation.

- '`HiddenValley:nFlav=2`' this determines the no. of flavours of hidden valley quarks to include in the model. It is set to 2.

For the benchmarks `HV2` and `HV3`:
The Diagonal mesons (piv11,piv22 and rhov11,rhov22) are set to decay to SM quarks namely u (up), d (down), s (strange), c (charm), b (bottom). The t (top) quark is not considered as it is extremely heavy and not kinematically accessible. And off-diagonal mesons (piv21 and rhov21) are not allowed to decay. Vector mesons (rho mesons) decay democratically, whereas pseudoscalar mesons (pi mesons) branching ratios are proportional to mass of the decay products. for `HV2`, the mass of the dark meson, 'm_{dark}' is set to 10 GeV. Whereas for `HV3`, it is set to 20 GeV.

For the benchmarks `HV3_HQ` and `HV3_LQ`, the decay modes are as shown in the tables 2.1, respectively.

Some samples with fixed $r_{invisible}$, or r_{inv} in short, is defined as the total invisible branching fraction of the dark hadrons. In the default PYTHIA 8 case all the off-diagonal mesons remain in the invisible sector while the diagonal mesons do not. So, for this case, where all mesons are degenerate, the r_{inv} will be the ratio of number of off-diagonal dark mesons to the total number of dark mesons. Another computational work-around to get an exact invisible fraction is to assign an arbitrary decay table to all dark mesons keeping only the fraction r_{inv} into either neutrino or dark matter states. This is what we use to test our NN's on different values of r_{inv}.

For these cases the decay modes for HV model with '$r_{inv} = r$' are as shown on tables 2.2 and 2.3.

2.2 Machine Learning techniques

In this section I will discuss the various Machine Learning techniques that have been used for the analysis.

2.2.1 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is a type of artificial neural network commonly used in machine learning. It consists of multiple layers of interconnected nodes, also known as artificial neurons. The network is organized into three main types of layers: input, hidden, and output layers. The input layer serves as the entry point for data and contains nodes corresponding to the features or variables of the input data. The number of nodes in the input layer is determined by the number of input features.

In between the input and output layers, there can be one or more hidden layers. These hidden layers are named as such because they are not directly connected to the input or output, but they play a crucial role in the network's ability to learn complex patterns and relationships within the data. Each node in a hidden layer is connected to every node in the previous and subsequent layers.

The output layer is responsible for producing the network's predictions or outputs. The number of nodes in the output layer depends on the specific problem the network is designed to solve. For binary classification, there is a single output node, while for multi-class classification, there can be multiple output nodes, each representing a class or category.

Each node in an MLP performs a simple computation. It calculates a weighted sum of its inputs, applies an activation function to this sum, and then passes the result to nodes in the next layer. The activation function introduces non-linearity into the model, allowing the network to learn complex relationships in the data. The 'Rectified Linear Unit(ReLU)' is the most commonly used activation function in MLP models. The function returns 0 if it receives any negative input, but returns the value back for any positive value x. So, mathematically, It can be expressed as $f(x) = max(0, x)$. '`Sigmoid`' and '`tanh`' are other popular alternatives. But as you can see in the Figure 2.2, the gradient of the sigmoid and tanh functions disappears quickly in the region outside (-2,2). This makes it difficult

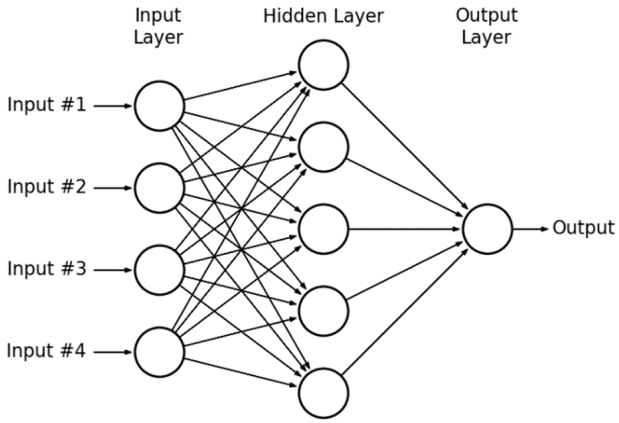


Figure 2.1: A schematic of a Multi-Layer Perceptron with one input layer(4 nodes), one hidden layer(5 nodes) and one output layer(single node)

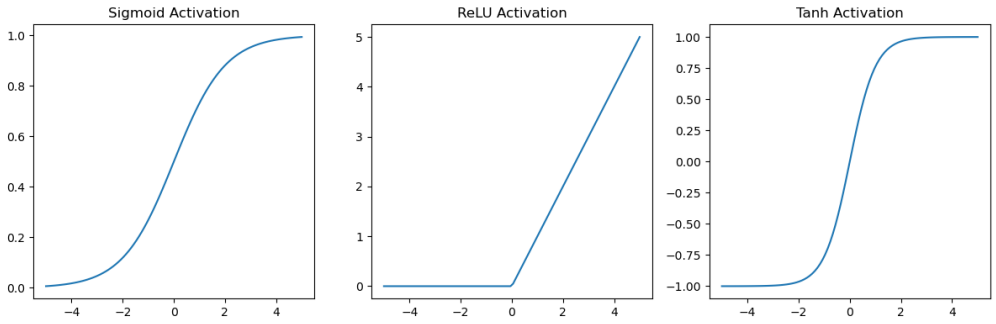


Figure 2.2: Graphical representations of various activation functions

to improve weights using gradient descent. This problem is known as 'vanishing gradient problem'. This problem is not applicable for `ReLU`, hence it has become more and more popular in the recent years. Therefore, `ReLU` has been used as the activation function for the input and hidden layers.

A loss function (also known as a cost function) is used to measure the difference between the predicted output and the actual target values. For binary classification tasks, a commonly used loss function is binary cross-entropy. The formula for binary cross-entropy is as follows:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(p(y_i) + (1 - y_i) \cdot log(1 - p(y_i)] \tag{2.1}$$

where, y_i is the true binary label, and $p(y_i)$ is the predicted probability of y_i.

Gradient descent is an optimization algorithm used in machine learning to minimize the cost or loss function of a model. It works by iteratively adjusting model parameters (weights and biases) in the direction of the steepest descent of the cost function. This adjustment is made in proportion to the negative gradient of the cost function, effectively seeking the lowest point (minimum) of the cost surface. The learning rate (α) is a small positive value that determines the step size in each iteration. Adam (Adaptive Moment Estimation) optimizer is a variant and improvement over the basic gradient descent optimization algorithm. Adam dynamically adjusts the learning rate for each parameter, making it well-suited for a wide range of neural network training tasks. It helps speed up convergence and improve training efficiency by maintaining moving averages of past gradients and their square values.

The '`He_normal`' initializer is a weight initialization technique commonly used in deep neural networks, including MLPs. The purpose of weight initialization is to set the initial values of the network's weights to suitable values to facilitate efficient training. The '`He_normal`' initializer initializes the weights using a normal distribution with a mean of 0 and a standard deviation calculated as the $\sqrt{\frac{2}{n}}$, where n is the number of nodes in the previous layer. This initialization method helps mitigate the vanishing gradient problem and promotes faster and more stable convergence during training, especially in deep networks with multiple hidden layers.

To train an MLP, you need a labeled dataset. The network's weights and biases are adjusted through backpropagation and optimization algorithms to minimize a loss function, which quantifies the discrepancy between the predicted output and the actual target values.

Overtraining is a phenomenon that occurs when a machine learning model learns to fit the training data too closely, capturing noise and specific patterns rather than general trends. This leads to poor performance on new, unseen data because the model doesn't generalize well. To have a check on overtraining, the input dataset is divided into two datasets, training dataset and testing dataset, randomly. The model is trained only on the training dataset, and then tested on the training as well as testing dataset. An overtrained model will perform much better on trained dataset compared to testing dataset. Higher the difference between the performance of the model on the two datasets, higher is the overtraining.

Dropout is a regularization technique used in neural networks to prevent overfitting. It involves randomly 'dropping out' (setting to zero) a fraction of the neurons or units in a neural network layer during each training iteration. These dropped out units do not

contribute to the forward or backward pass of that iteration. 'Dropout rate', the primary hyper-parameter of the layer determines the probability of dropping out each neuron during training iteration.

It's important to note that while MLPs can solve a variety of problems, they may not be the best choice for all tasks, especially when dealing with images. In such cases, more advanced architectures like Convolutional Neural Networks (CNNs) are required.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing and analyzing visual data, such as images and videos. They have revolutionized computer vision tasks and found applications in various domains beyond image analysis. CNNs consist of several key components, including convolutional layers, pooling layers, fully connected layers, and activation functions.

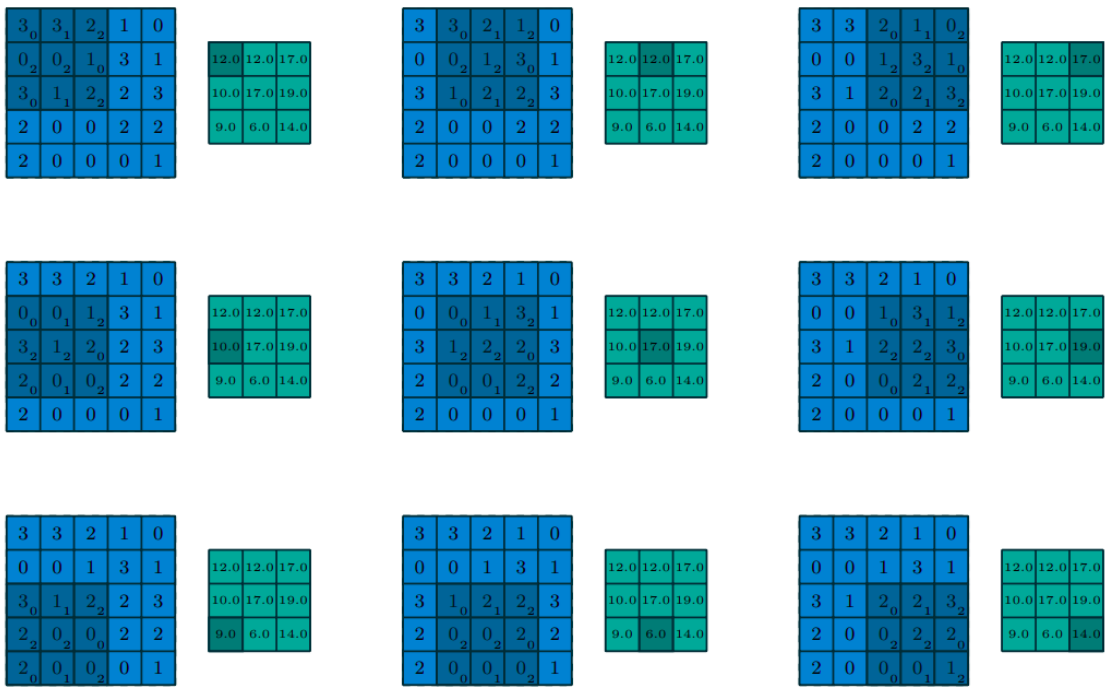


Figure 2.3: Visual schematic for calculating output of a convolutional layer [17]

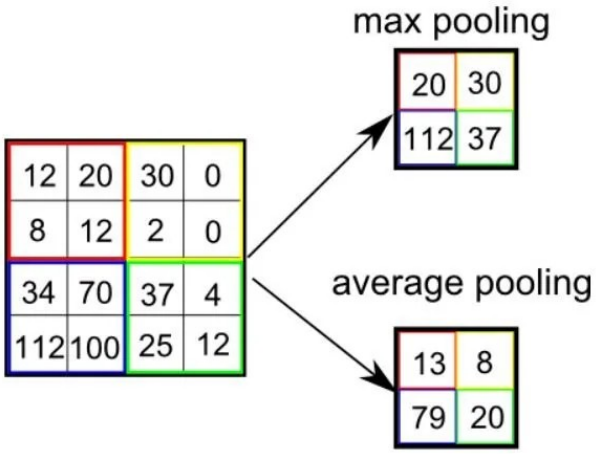


Figure 2.4: Visual schematic for pooling layer [18]

In the convolutional layers, small filters (kernels) slide over the input image, and output is obtained by computations as shown in Figure 2.3. It enables the network to detect features like edges, textures, and patterns. Filters are the learnable parameters, each highlighting specific patterns. The stride determines the units by which filter moves during convolution, affecting the spatial dimensions of the feature maps.

Pooling layers are used to reduce the spatial dimensions of the feature maps while retaining crucial information. Techniques like max-pooling and average-pooling help in this process. These layers contribute to reducing computational complexity and preventing overfitting. The working of these layers can be visualised from Figure 2.4.

Padding is technique that involves adding extra rows and columns of zeros around the input image or feature maps. It is used to preserve the dimensions of the image under the application of convolution. It also ensures that the information in the corner and edges of an image is read more often.

In a neural network, as data flows through layers, the distribution of inputs to each layer can change. This phenomenon, known as internal covariate shift, can make training deep networks challenging because it requires careful initialization of weights and the use of small learning rates to prevent divergence or slow convergence. Batch Normalization addresses this problem by normalizing the inputs within each batch of data. For each feature, Batch Normalization calculates the mean and standard deviation across the batch. It then normalizes the feature by subtracting the mean and dividing by the standard deviation.

This effectively centers the distribution of the feature around zero and scales it to have unit variance. After normalization, the feature is scaled by a learnable parameter called '`gamma`' and shifted by another learnable parameter called '`beta`'. These parameters allow the network to learn the appropriate scale and shift for each feature. They provide flexibility in handling variations in the data distribution.

Fully connected layers, similar to those in a Multi-Layer Perceptron (MLP), come after a series of convolutional, pooling and batch normalization layers. They are responsible for classification or regression tasks based on the extracted features from previous layers.

2.2.3 Analysing output of NN

Various plots that are used to analyze the performance of NN, are discussed below.

NN score plot

NN score is the final output the model. For binary classification tasks, the value closer to the label of a class implies higher probability of the input being of that class. Usually signal is labelled '1' and the background is labelled '0'. In that case, higher NN score implies higher probability of being signal. NN score plot is the histogram of NN score of the model. It is plotted as separate curves for the inputs labelled 0 and 1. This plot helps in determining the cut on NN score above which a the input is to be considered as signal.

ROC curve

The ROC plot is a graphical representation of the classifier's performance at various thresholds. It displays the trade-off between True Positive Rate (sensitivity) and False Positive Rate (1−specificity) as you vary the decision threshold. A model with a ROC curve that approaches the top-left corner has better classification accuracy and is more capable of discriminating between the classes. The Area Under the ROC Curve (AUC) summarizes the model's overall performance. An AUC value of 0.5 suggests a random classifier, while a value of 1 indicates a perfect classifier.

Signal efficiency vs background rejection plots

Signal efficiency is defined as fraction of signal events that are classified as signal by the model. And the background efficiency is defined as the fraction of background events that are classified as signal by the model. For this plot, the signal efficiency is plotted on the x-axis and the inverse of background efficiency is plotted on the y-axis. This plot helps us find the balance between the trade off between sensitivity and background rejection.

In the subsequent chapters we will use the term 'network' instead of 'model', as the term 'model' is also used to refer different HV models.

Chapter 3

Discriminating Hidden Valley jets

In this chapter we will use various machine learning tools to discriminate HV jets against the background of QCD jets.

3.1 Discriminating HV jets clustered with anti-k_T using MLP

Here, the `QCD` and `HV2` events are clustered using anti-k_T jet algorithm with $R = 0.7$. For the analysis, **pTjetMin** is set to 100 GeV and the requirement of minimum numbers of jets to be 2 is also put.

Based on our preliminary investigations, the following quantities were chosen as inputs to the neural network:

- Number of constituents in the two hardest jets. Here, I have used anti-k_T to cluster jets.
- Scalar sum of p_T of photons in a jet for the two hardest jets in the event.
- Subjet-p_T of upto three hardest subjets of the two hardest jets in the event.
- ΔR between subjets, for up to three hardest subjets of the two hardest jets.

- Energy correlation functions [19, 20] e_2 and e_3.
- Lund jet plane variables (ΔR and k_T) for the first five unclusterings.

Histograms for few of the input quantities are shown in Figures (3.1-3.2). And the LJP plots are shown in Figure 1.7.

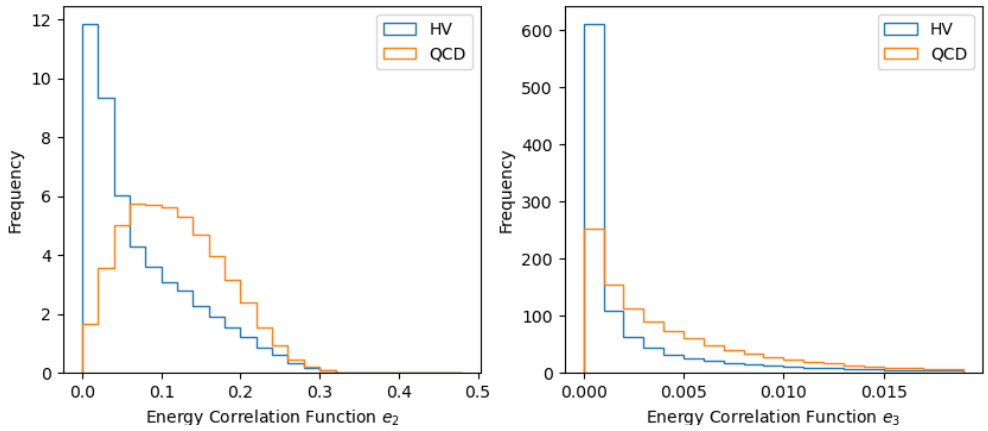


Figure 3.1: Energy Correlation Functions, e_2 on the left and e_3 on the right

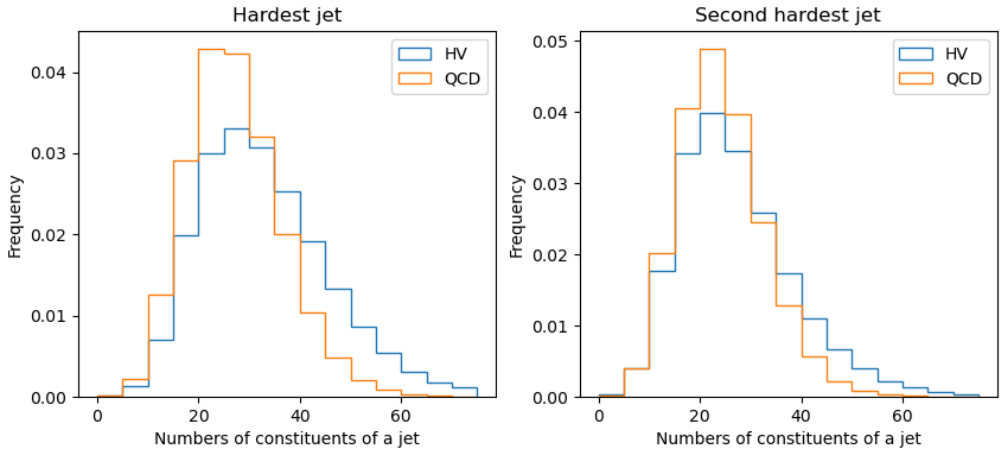


Figure 3.2: Number of constituents in the hardest (left) and second hardest (right) jet

3.1.1 NN architecture

Sequential network have been used with 7 Dense layers, containing 32, 32, 16, 8, 4, 2, 1 neurons respectively. Of which, the first 6 layers use activation function '`ReLU`' and the last

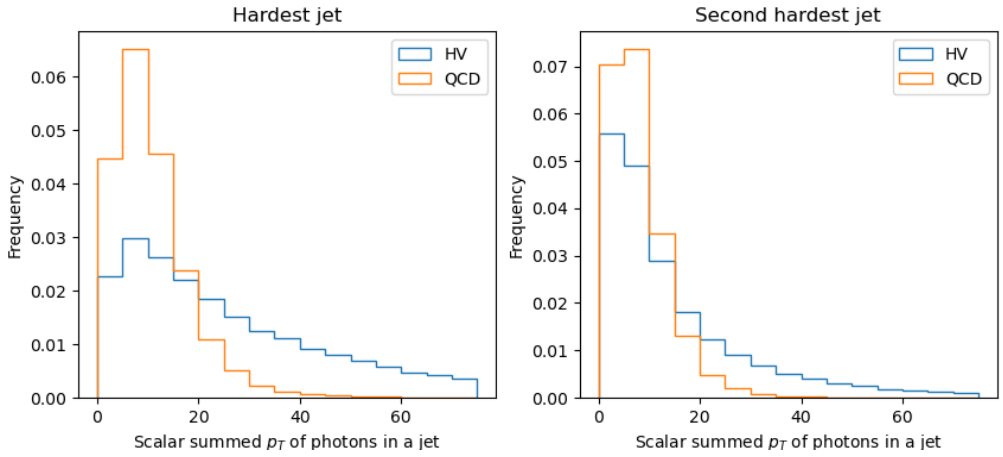


Figure 3.3: Scalar sum of p_T of gamma particles in the hardest (left) and second hardest (right) jet

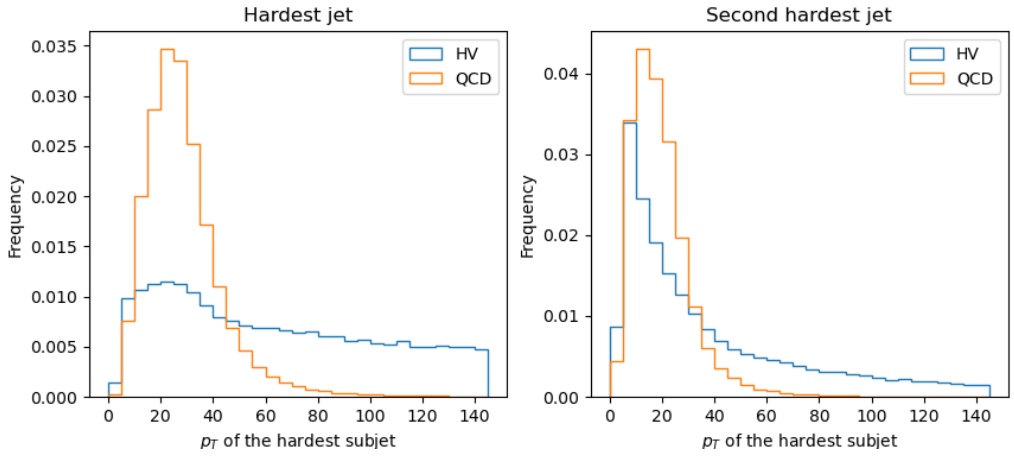


Figure 3.4: Hardest subjet p_T of the hardest (left) and second hardest (right) jet

(output) layer uses '`sigmoid`' activation function. `Adam` optimizer is used for compiling the network.

The input size is about 40000 (20000 `QCD` + 20000 `HV3`) events, 80% of which is used for training and the remaining 20% is used for testing. The network is trained for 400 epochs with batch size of 100. Early stopping condition is applied with `patience` = 20, along with `restore_best_weights` set to `True`, monitor being `val_accuracy`.

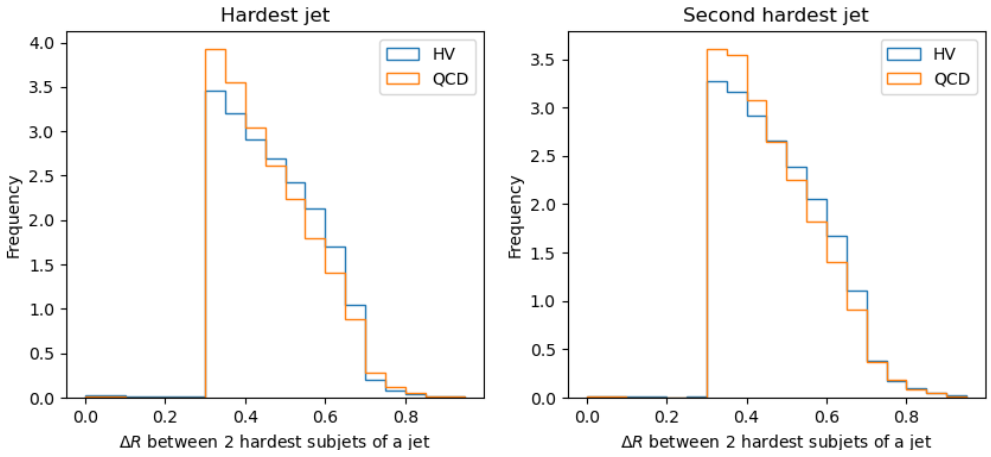


Figure 3.5: ΔR between 2 hardest subjets of the hardest (left) and second hardest (right) jet

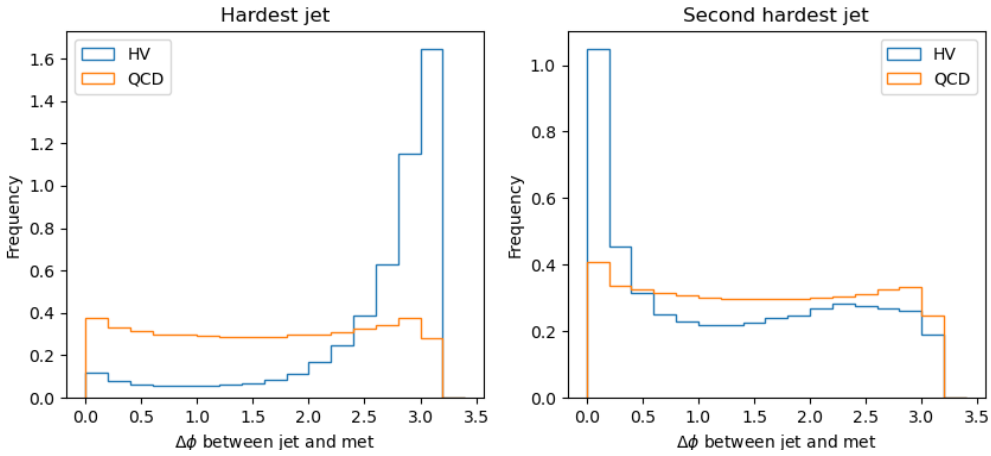


Figure 3.6: $\Delta\phi$ of MET with the hardest jet (left) and the second hardest jet (right)

3.1.2 NN output

The output of the neural network is shown in figure 3.7. As can be seen, there is a good discrimination between signal and background, with consistent performance for both the training and testing samples.

I have plotted 1/background efficiency ($1/\epsilon_{bkg}$) vs signal efficiency (ϵ_{sig}), as well as the ROC(Receiver Operating Characteristic) curve in Figure 3.7.

Then `QCD` background is bin-wise matched, and the network was retrained (with exactly

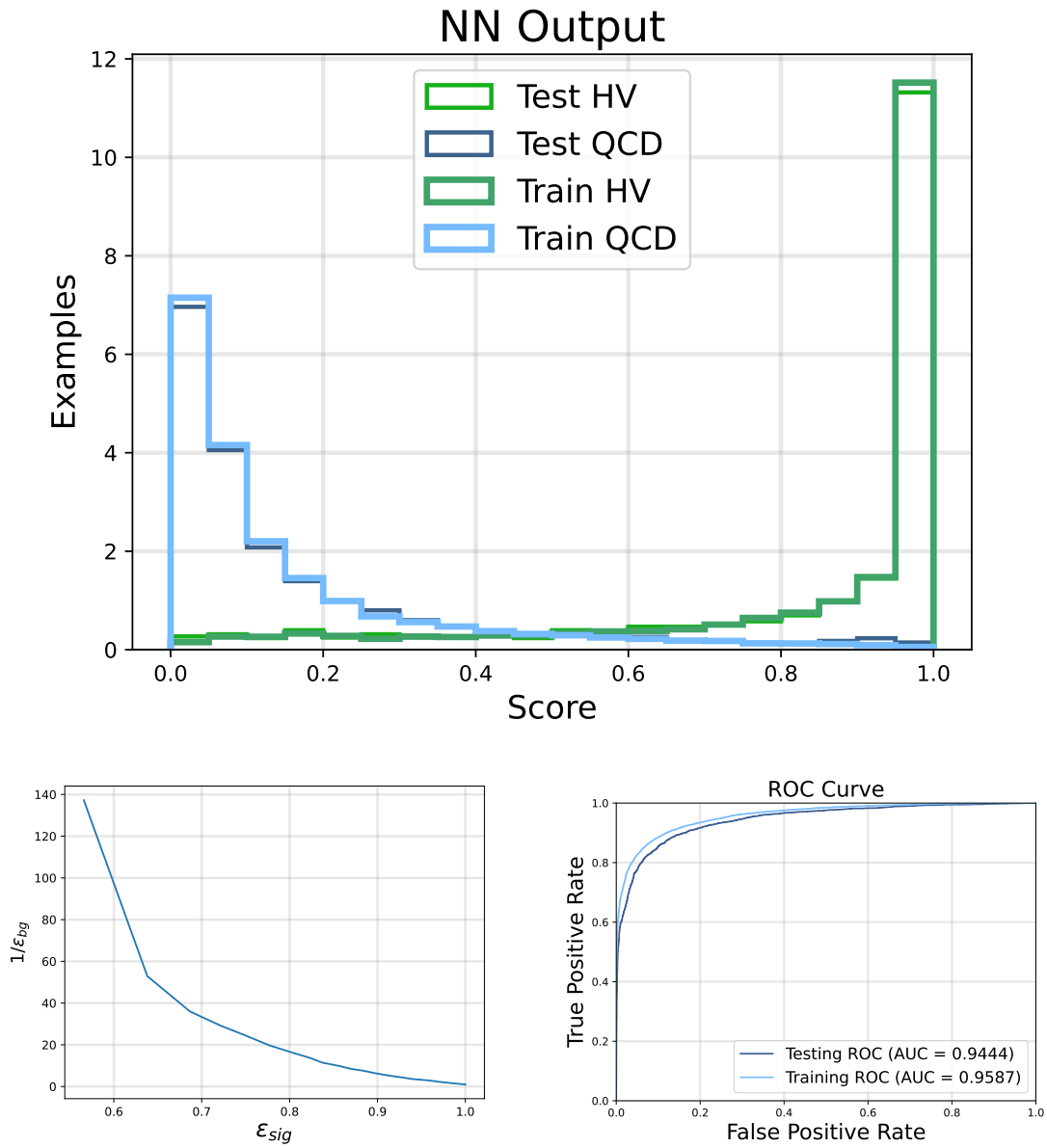


Figure 3.7: Plot on the top shows the histogram for NN score, the bottom left shows signal efficiency vs background rejection plot, and ROC curve on the bottom right, for the MLP network on `HV2` model, with anti-k_T algorithm used for clustering jets

same NN architecture), and the output can be seen in the Figure 3.8.

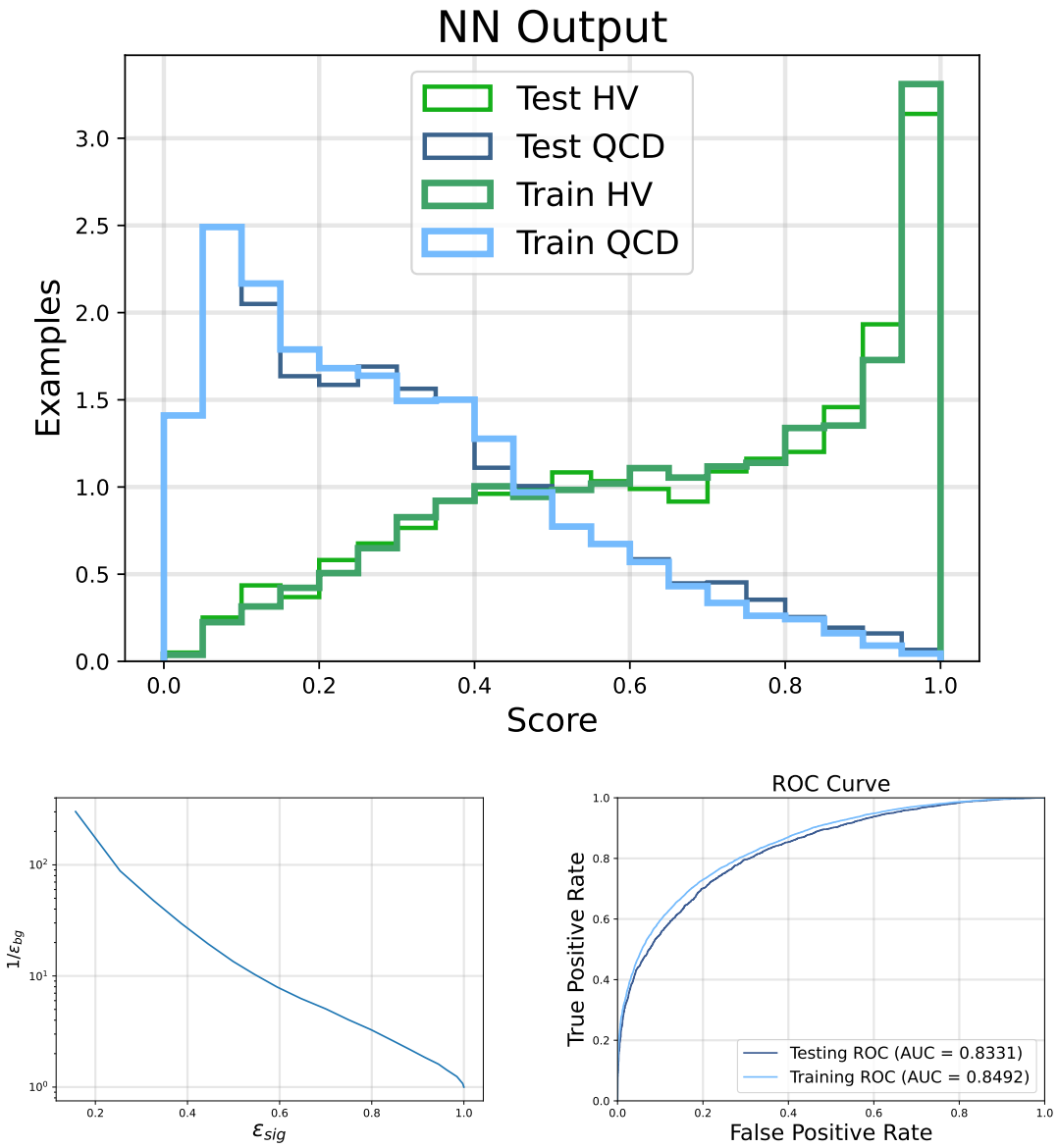


Figure 3.8: Plot on the top shows the histogram for NN score, the bottom left shows signal efficiency vs background rejection plot, and ROC curve on the bottom right, for the MLP network on `HV2` model with bin-wise matched p_T distribution, with anti-k_T algorithm used for clustering jets

3.2 Discriminating HV jets clustered with SIFT, using MLP

Here the jets has been clustered using the SIFT algorithm with '`pTjetMin`=50 GeV'. And the condition of having atleast 2 jets in each event has also been put. As the SIFT jets are very thin compared to HV, instead of going to subjets, the detector level quantities of the jets have been used for analysis i.e transverse momentum (p_T), invariant mass (m), pseudorapidity (η), azimuthal angle (ϕ) of the 2 hardest jets. And to see the effect of MET on the network, I have separately trained another network which has an extra input parameter MET, along with the other inputs. Here, the `HV3_HQ`, `HV3_LQ` samples, as described in sec 2.1.1, are differentiated against bin-wise matched `QCD` sample. The plot 3.9 verifies the bin-wise matching. The plot 3.10 gives a peak at 20 GeV, which is the dark meson mass. From this, it can be seen that the SIFT algorithm can be helpful in extracting mass scales of dark mesons.

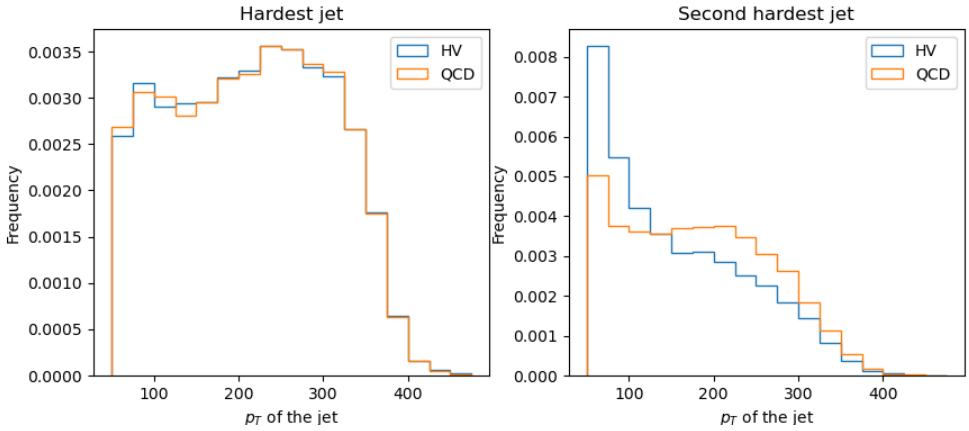


Figure 3.9: p_T of the hardest jet (left) and the second hardest jet (right)

This is to check the dependence of the network on dark mesons decaying to heavy quarks like, $b\,\bar{b}$. Hence, two different scenarios are considered, one in which the dark pions decay to light quarks only, and one in which it decays to heavy quarks only. The decays of the ρ mesons have not been altered. So, here, in total I have trained 4 networks, labelled as:

1. `HQ_wo_MET`: `HV3_HQ` sample is used and MET is not included in inputs.
2. `HQ_with_MET`: `HV3_HQ` sample is used and MET is included in inputs.

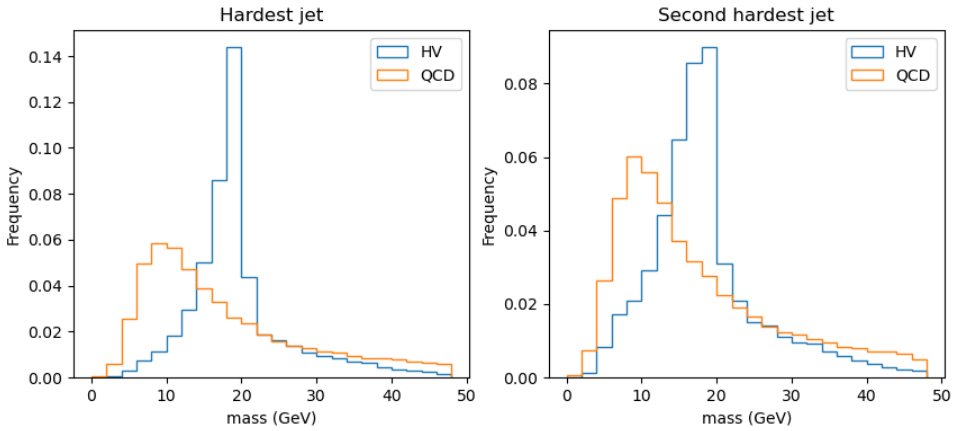


Figure 3.10: Invariant mass of the hardest jet (left) and the second hardest jet (right)

3. `LQ_wo_MET`: `HV3_LQ` sample is used and MET is not included in inputs.

4. `LQ_with_MET`: `HV3_LQ` sample is used and MET is included in inputs.

3.2.1 NN architecture

Sequential network have been used with 7 Dense layers, containing 32, 16, 16, 8, 4, 2, 1 neurons respectively. Of which, the first 6 layers use activation function '`ReLU`' and the last layer uses '`sigmoid`' activation function. `Adam` optimizer is used for compiling the network.

For each network the input size is about 50000 (25000 `QCD` + 25000 `HV3`) events, 80% of which is used for training and the remaining 20% is used for testing. The network is trained for 400 epochs with batch size of 100. Early stopping condition is applied with `patience = 20`, along with `restore_best_weights` set to `True`, monitor being `val_accuracy`.

3.2.2 NN output

The performance of all the four trained networks, `HQ_wo_MET`, `HQ_with_MET`, `LQ_wo_MET` and `LQ_with_MET` on the corresponding testing datasets can be seen in the figures 3.11, 3.12, 3.13, 3.14 respectively.

Note that, for the benchmarks `HV2`, `HV3`, `HV3_HQ` and `HV3_LQ`, the r_{inv} was not fixed. To

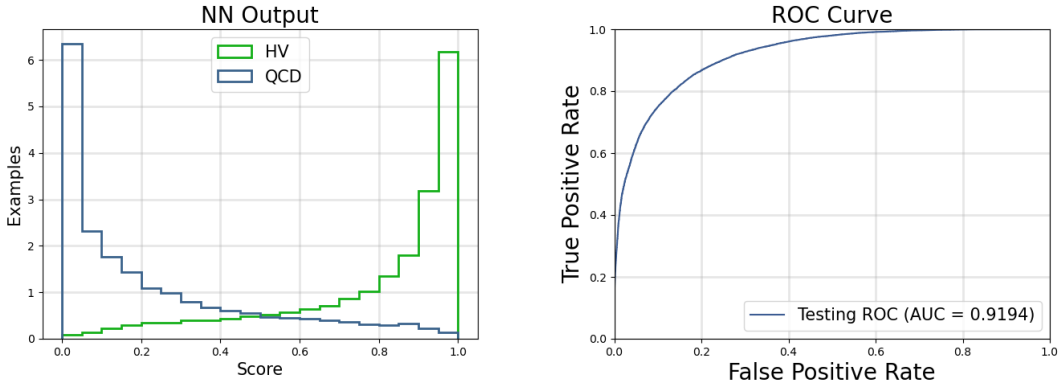


Figure 3.11: NN score plot (on the left) and signal efficiency vs background rejection plot (on the right), for the `HQ_wo_MET` network on the testing dataset

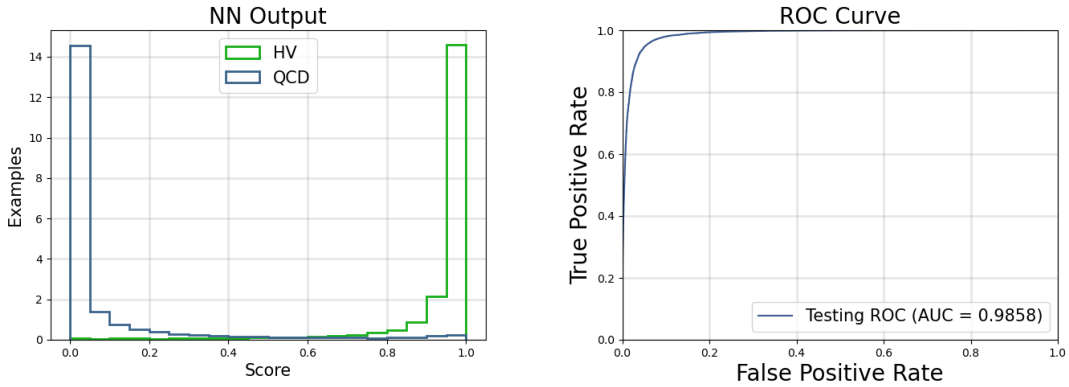


Figure 3.12: NN score plot (on the left) and signal efficiency vs background rejection plot (on the right), for the `HQ_with_MET` network on the testing dataset

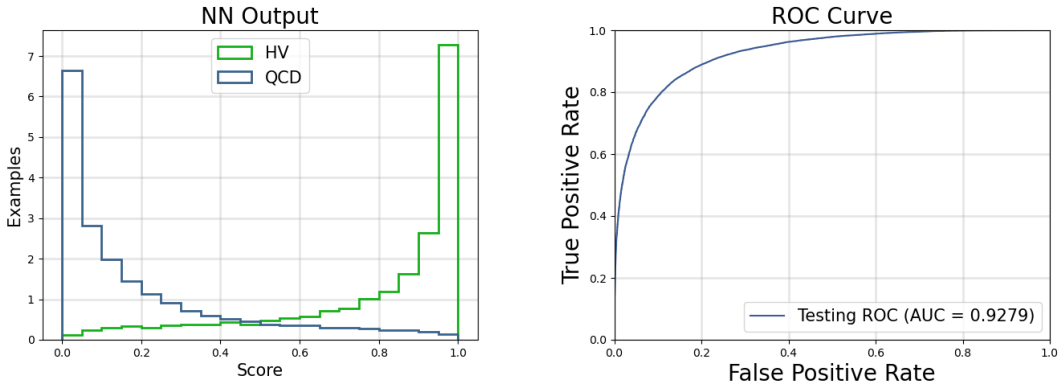


Figure 3.13: NN score plot (on the left) and signal efficiency vs background rejection plot (on the right), for the `LQ_wo_MET` network on the testing dataset

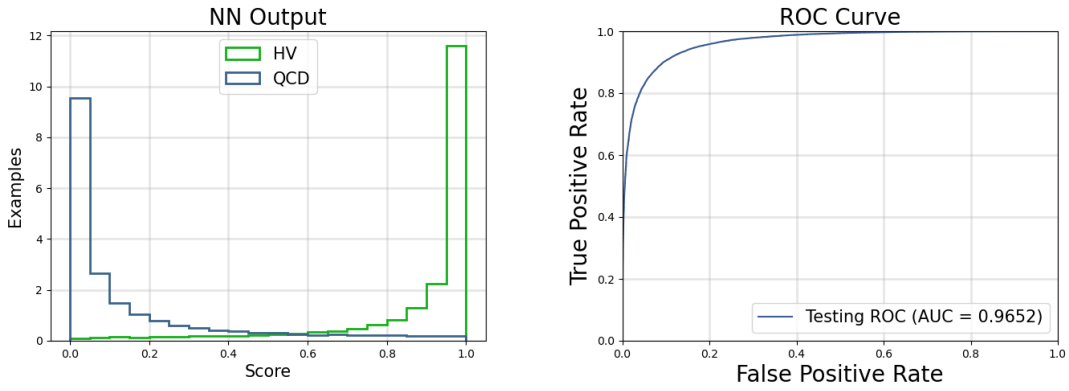


Figure 3.14: NN score plot (on the left) and signal efficiency vs background rejection plot (on the right), for the `LQ_with_MET` network on the testing dataset

see how the performance changes with changing r_{inv}, I used the benchmarks with fixed r_{inv} and tested their samples with these trained NN networks, and calculated the proportion of events accepted with the cut of 0.9 on NN score. This quantity will be referred to as 'acceptance efficiency'. This acceptance efficiency is normalized by dividing it with the acceptance efficiency of the benchmark on which the network was trained. The behaviour of this 'normalized acceptance efficiency' (to be denoted as ϵ_A) on changing the r_{inv} is shown in Figure 3.15.

As can be seen seen in the Figure 3.15, normalised acceptance efficiency increases with increase in r_{inv}.

3.3 Discriminating HV jets clustered with SIFT, using CNN

In this study, we harness the capabilities of Convolutional Neural Networks (CNNs) to differentiate between HV and QCD signature. The approach begins by representing the calorimetric deposits at the detector as pixelated images, as was previously done by [21–23]. Each pixel within these images corresponds to a calorimetric cell. These images serve as the input data for the CNNs. The CNN architecture, with its inherent ability to capture intricate patterns and spatial dependencies within images, is adept at finding subtle features that distinguish HV and QCD events.

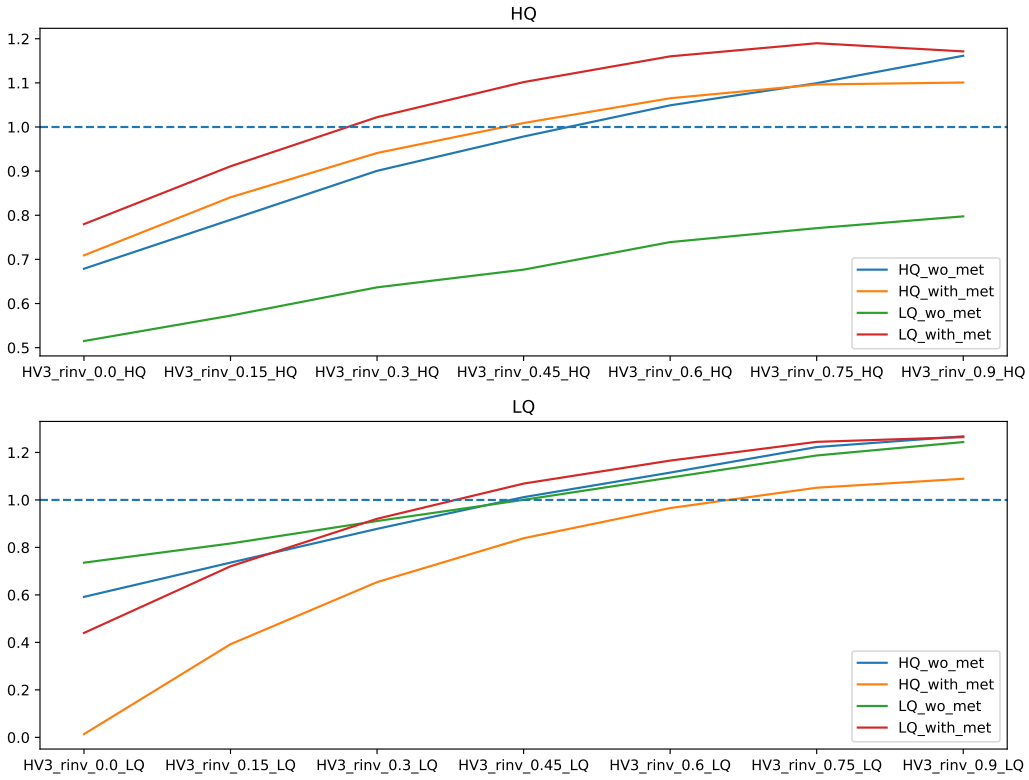


Figure 3.15: Normalized acceptance efficiency for MLP networks trained on SIFT jets. On x-axis lies the model on which the trained network is being tested and different plots are for different trained networks. The top figure have testing samples of the HQ type and bottom figure have testing sample of the LQ type.

3.3.1 Jet-image formation

For this analysis, a calorimetric cell is considered to be of dimension 0.05×0.05 in the $\eta - \phi$ plane. I have used PYTHIA 8 to generate the events, for the benchmarks `HV3_HQ`, `HV3_LQ` and `QCD` and clustered them using SIFT algorithm with `pTjetMin` = 50 GeV along with the condition of having atleast 2 jets in each event. The input data consists of a 21×21 pixel, square grid centered at jet-center of the hardest jet. Jet-images are then formed by interpreting energy depositions at each calorimetric cell as the pixel intensity of the corresponding image. An exampler image, is as shown in the figure 3.16.

Image pre-processing

To optimize the learning of distinguishing characteristics between signal and background, and to prevent the acquisition of spatial and temporal symmetries, the jet images need to be pre-processed.

Here, the pre-processing is done in 2 steps, translation and rotation. In translation the jet-image is translated so that the hardest calorimetric tower occupies the center of the jet-image. And in rotation, jet-image is rotated so that second hardest calorimetric tower lies on the positive x-axis. The pre-processed jet-image will be of size 13×13 pixels. An example of the pre-processed jet image is shown in the figure 3.16. The jet-images are then normalized so that the sum of the square of each pixel intensity of an image is 1.

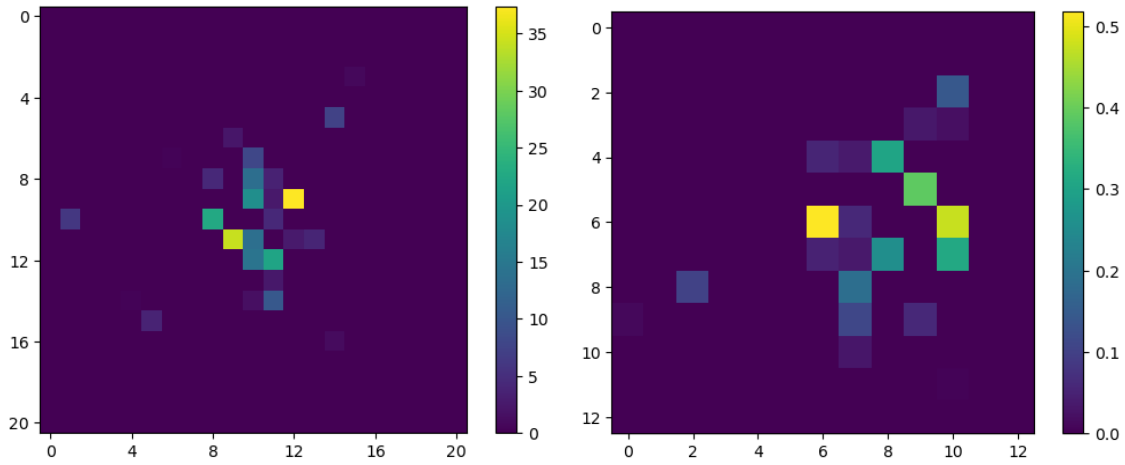


Figure 3.16: Exampler jet-image, before pre-processing on the left and after pre-processing on the right

CNN architecture

The Convolutional Neural Network (CNN) architecture, used here, is a sequential network featuring:

1. `Conv2D` Layer: 32 filters, (3,3) kernel size, (2,2) stride, 'he_uniform' weight initialization.
2. `AveragePooling2D` Layer: (2,2) pooling size.

3. `BatchNormalization` layer to enhance training stability.

4. `Dropout` layer: with dropout rate of 0.2 , to mitigate overfitting.

5. `Conv2D` Layer: 32 filters, (3,3) kernel size, (1,1) stride.

6. `MaxPooling2D` Layer: (2,2) pooling size and 'same' padding followed by a `BatchNormalization` layer

7. `Flatten` layer, which flattens the feature maps to arrays. Followed by a `Dropout` layer: with dropout rate of 0.2.

8. Followed by 3 `Dense` layers with 16, 8, 1 neurons respectively.

For each network (trained on `HV3_HQ` and `HV3_LQ`) the input size is about 50000 (25000 `QCD` + 25000 `HV3`) events, 80% of which is used for training and the remaining 20% is used for testing. Early stopping condition is applied with `patience` = 10, along with `restore_best_weights` set to `True`, monitor being `val_accuracy`.

Activation function used for the last layer, with a single neuron, is `Sigmoid`. For all the other layers `ReLU` is used as activation function. And padding is set to 'same' for all the convolutional and pooling layers.

CNN output

The performance of networks trained on `HV3_HQ` and `HV3_LQ` can be seen in Figure 3.17 and 3.18 respectively.

And then these two trained networks were tested on the various fixed r_{inv} samples, and the behaviour of normalized acceptance efficiency was shown in Figure 3.19. In this case as well the normalized acceptance efficiency can be seen to increase with increase in r_{inv}. It can also be noticed that the model trained on `HV3_LQ` does not perform well on testing samples of HQ type.

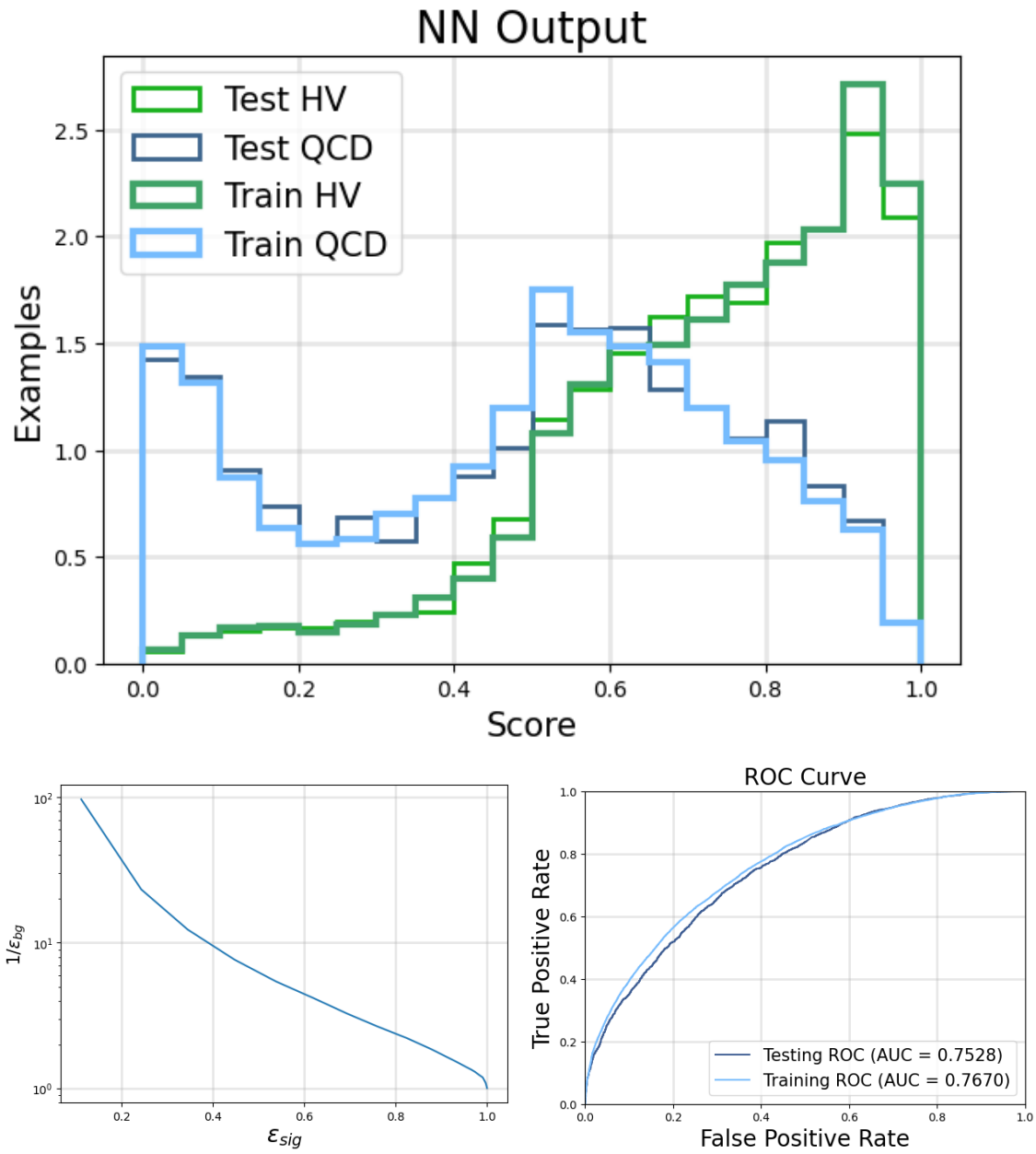


Figure 3.17: Plot on the top shows the histogram for NN score, the bottom left shows signal efficiency vs background rejection plot, and ROC curve on the bottom right for CNN network trained on the `HV3_HQ` samples.

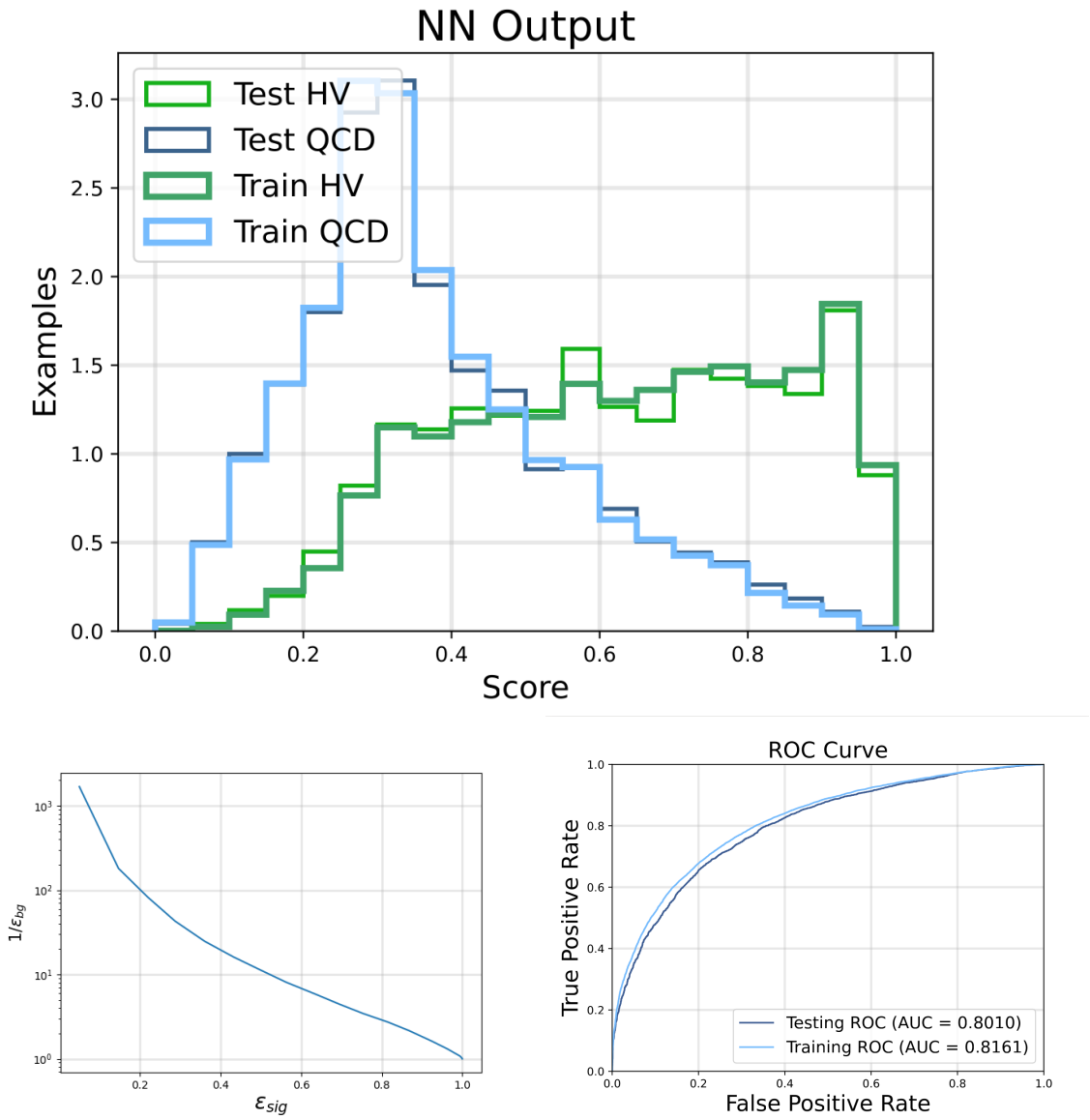


Figure 3.18: Plot on the top shows the histogram for NN score, the bottom left shows signal efficiency vs background rejection plot, and ROC curve on the bottom right, for CNN network trained on the `HV3_LQ` samples.

3.4 Discriminating HV jets clustered with SIFT, using CNN+MLP

As a last method in this study, I have tried to mix the CNN and MLP approach. In this case, the input consist of the combined inputs of last 2 sections (3.2 and 3.3), i.e. jet-image

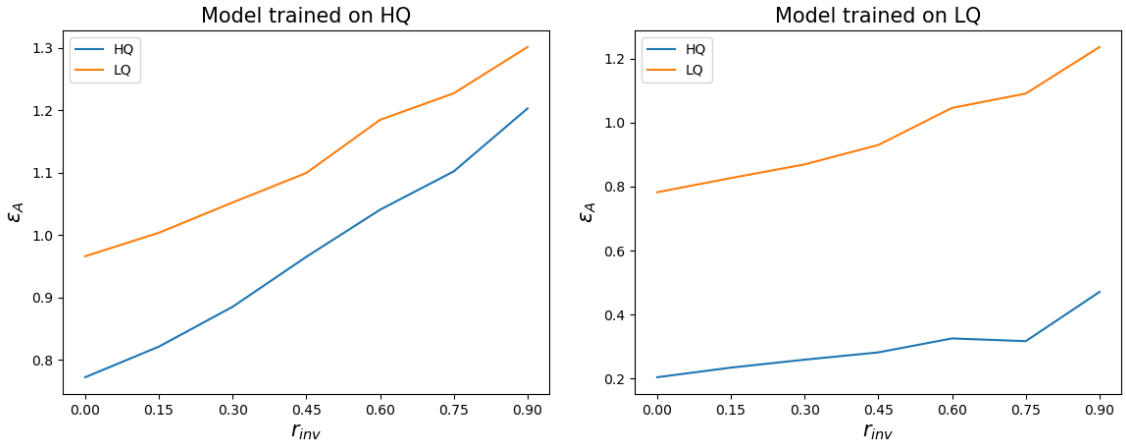


Figure 3.19: Plot showing the behaviour of normalized acceptance efficiency w.r.t different r_{inv} samples for networks trained on `HV3_HQ` (on the left) and `HV3_LQ` (on the right).

for the hardest jet and mass, p_T, η, ϕ of the 2 hardest jets. Benchmarks used for analysis are `HV3_HQ` and bin-wise matched `QCD`. All the pre-processing steps from section 3.3.1 are applied to the jet images

3.4.1 CNN+MLP architecture

In this case, the architecture is mostly similar to that of CNN (3.3.1), with the only difference being, after the flattening layer, the MLP input is appended to the flattened array. And then the Dense layers follows as usual.

3.4.2 CNN+MLP output

Performance for the CNN+MLP network can be seen in the Figure 3.20.

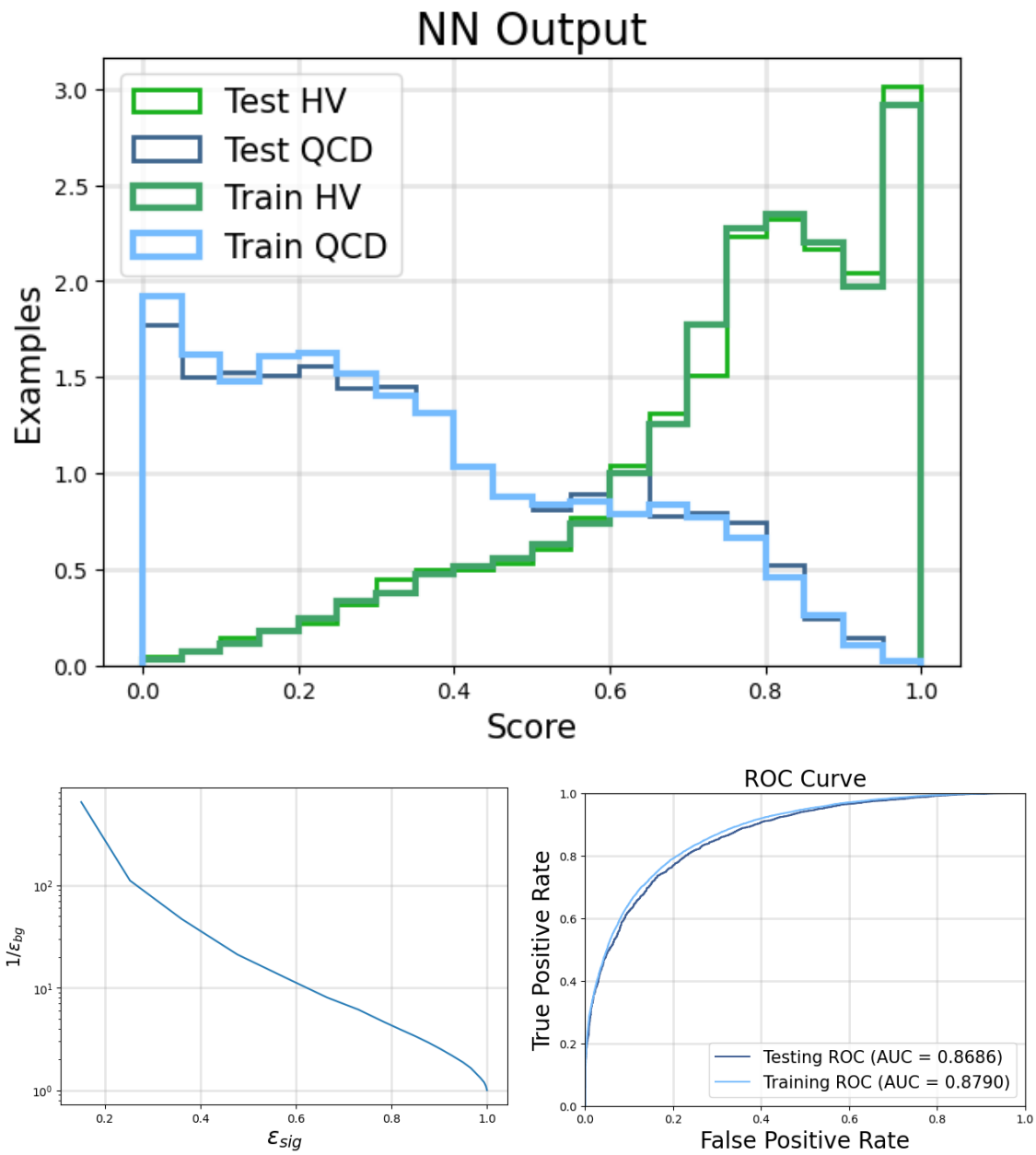


Figure 3.20: Plot on the top shows the histogram for NN score, the bottom left shows signal efficiency vs background rejection plot, and ROC curve on the bottom right, for CNN+MLP network on `HV3_HQ` model

Chapter 4

Summary and Conclusions

In this thesis, we've made significant progress in understanding Hidden Valley models by using advanced machine learning techniques. In this final chapter, we'll summarize the main findings of our research.

We began by identifying crucial jet variables, for anti-k_T jets, capable of effectively discriminating between HV jets and the QCD background. A binary classification MLP neural network was then trained using these variables as inputs, and its performance was assessed through the analysis of various plots. To mitigate scale dependence, we implemented a bin-wise matching approach, aligning the transverse momentum (p_T) distribution for the hardest jet of the background with that of the signal. The subsequent network, trained and analyzed for this purpose, exhibited a noticeable decrease in performance compared to the initial network.

We then replaced the anti-k_T algorithm by the SIFT algorithm to cluster jets. SIFT jets proved to be more useful in extracting the dark meson mass scale, and also in discriminating the HV model. Subsequently, another MLP was trained with the detector level quantities, (m, p_T, η, ϕ) of the first two jets, as inputs to the network. To study the dependence of the network in picking up heavy and light quark decays, two other benchmarks were introduced, one in which the dark pions decay to heavy quarks (c,b) only, and other one in which they decay to light quarks only. Each MLP network was separately trained on samples from these benchmarks. Additionally, we explored the influence of missing transverse energy (MET) by introducing two more networks with MET as an additional input.

Few more benchmarks were introduced, where the invisible fraction r_{inv} was artificially fixed to different values to test the response of the trained network to different fractions of missing energy. To examine the impact of r_{inv} on the network's performance, we tested all four trained networks on samples with varying fixed r_{inv} values. It was observed that the performance of networks improve with increase in r_{inv}.

To investigate whether HV events can be discriminated against QCD without relying on jet mass, we harnessed the power of Convolutional Neural Networks (CNNs). Jet images of the hardest jet served as inputs for the CNN, offering a promising avenue for future investigations. Here again, CNN's were trained for `HV3_HQ` and `HV3_LQ` models separately and the trained CNN networks were tested on the samples with varying fixed r_{inv} values, and it was again observed that the performance of networks improve with increase in r_{inv}.

In the end, we tried to mix the MLP and CNN network, for which the input was the combined input of CNN network and MLP network.

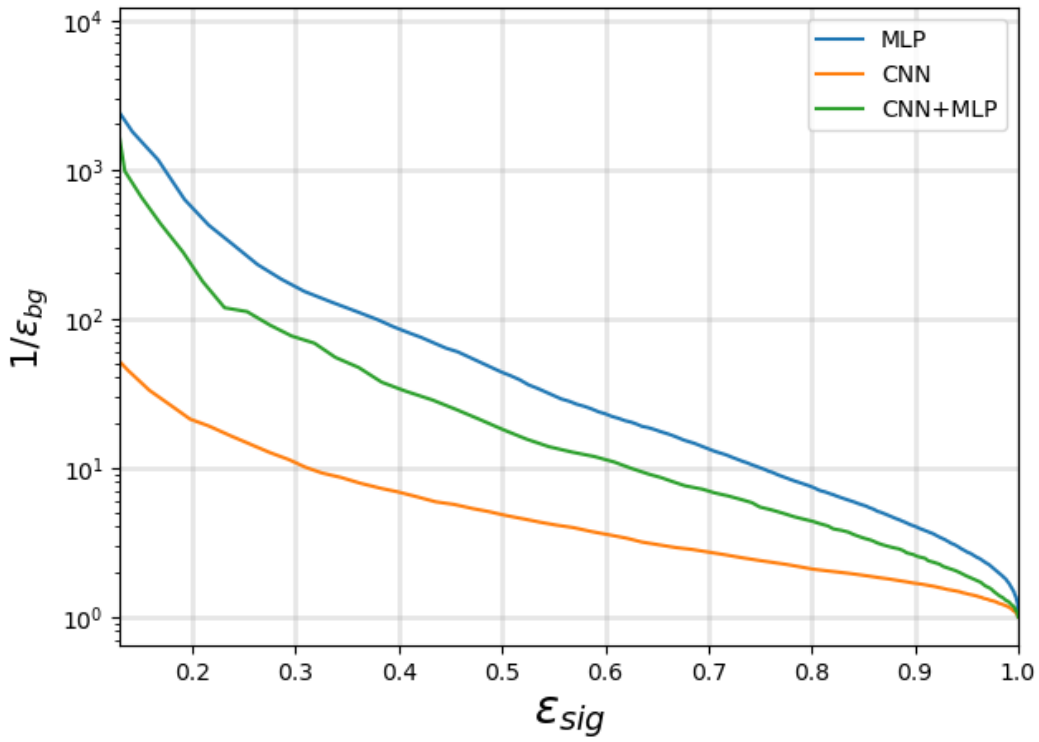


Figure 4.1: Comparing signal efficiency vs background rejection plots for different networks trained on `HV3_HQ` model, where SIFT clustering was used.

A comparison of the three approaches can be seen in Figure 4.1. Currently it seems that the MLP analysis does the best job of providing robust background rejection for given signal efficiency.

Chapter 5

Outlook

In this chapter, we will discuss about various short-comings and future plans for this this study.

- To make the analysis more realistic and robust, detector response simulation softwares like Delphes [24] will be used. This is especially required in the CNN case as, during the pre-processing of jet-images, we have rotate input images, but the rotated cells does not superimpose on the original grid. So we had to consider the calorimetric tower for each cell to be located at the centre of each cell. To check the robustness of this step it is essential to test this with detector simulations.

- For this study, the jet radius has been defined as the maximum value of ΔR between a jet and its constituents. But more accepted quantity is the jet area, which is based on ghost particles. But as our current implementation of SIFT is $\mathcal{O}(N^2)$, it is computationally not viable to look at jet area. Therefore we will be looking at ways to make the implementation faster, with the goal of making it $\mathcal{O}(N\ log(N))$.

- For this analysis the value of Z_v was set to 800 GeV. This analysis needs to be generalised by considering different mass scales for the mediator particle.

- In each case, the output of the NN was observed to be highly correlated with the input mass of dark mesons to the event generator. Hence, to search within the parameter space of HV models, one might need to train a separate NN for each mass hypothesis.

- Z_v couplings to both SM and HV quarks, are assumed to be at maximum value, and can in principle be constrained by searches in dijets, which would reduce the value of allowed couplings. But this is not expected to affect the kinematics, and hence the discrimination potential of the analysis.
- Other ways to extract hidden meson masses will also be explored in the future.

Bibliography

1. Carloni, L. & Sjostrand, T. Visible Effects of Invisible Hidden Valley Radiation. *JHEP* **09,** 105. arXiv: 1006.2911 [hep-ph] (2010).
2. Carloni, L., Rathsman, J. & Sjostrand, T. Discerning Secluded Sector gauge structures. *JHEP* **04,** 091. arXiv: 1102.3795 [hep-ph] (2011).
3. Strassler, M. J. & Zurek, K. M. Echoes of a hidden valley at hadron colliders. *Phys. Lett. B* **651,** 374–379. arXiv: hep-ph/0604261 (2007).
4. Han, T., Si, Z., Zurek, K. M. & Strassler, M. J. Phenomenology of hidden valleys at hadron colliders. *JHEP* **07,** 008. arXiv: 0712.2041 [hep-ph] (2008).
5. Collaboration, A. *et al.* Search for non-resonant production of semi-visible jets using Run˜ 2 data in ATLAS. *arXiv preprint arXiv:2305.18037* (2023).
6. Collaboration, C. *et al.* Search for resonant production of strongly coupled dark matter in proton-proton collisions at 13 TeV (2022).
7. Knapen, S., Shelton, J. & Xu, D. Perturbative benchmark models for a dark shower search program. *Physical Review D* **103,** 115013 (2021).
8. Kribs, G. D., Martin, A., Ostdiek, B. & Tong, T. Dark Mesons at the LHC. *JHEP* **07,** 133. arXiv: 1809.10184 [hep-ph] (2019).
9. Beauchesne, H., Bertuzzo, E., Grilli Di Cortona, G. & Tabrizi, Z. Collider phenomenology of Hidden Valley mediators of spin 0 or 1/2 with semivisible jets. *JHEP* **08,** 030. arXiv: 1712.07160 [hep-ph] (2018).
10. Beauchesne, H., Bertuzzo, E. & Grilli Di Cortona, G. Dark matter in Hidden Valley models with stable and unstable light dark mesons. *JHEP* **04,** 118. arXiv: 1809.10152 [hep-ph] (2019).

11. Cohen, T., Lisanti, M. & Lou, H. K. Semivisible Jets: Dark Matter Undercover at the LHC. *Phys. Rev. Lett.* **115,** 171804. arXiv: 1503.00009 [hep-ph] (2015).

12. Marzani, S., Soyez, G. & Spannowsky, M. *Looking inside jets: an introduction to jet substructure and boosted-object phenomenology* arXiv: 1901.10342 [hep-ph] (Springer, 2019).

13. Cacciari, M., Salam, G. P. & Soyez, G. The anti-k_t jet clustering algorithm. *JHEP* **04,** 063. arXiv: 0802.1189 [hep-ph] (2008).

14. Larkoski, A. J., Rathjens, D., Veatch, J. & Walker, J. W. Jet clustering with a scale-invariant filtered tree: A new algorithm for the substructure era. *Phys. Rev. D* **108,** 016005. https://link.aps.org/doi/10.1103/PhysRevD.108.016005 (1 July 2023).

15. Cohen, T., Roloff, J. & Scherb, C. Dark Sector Showers in the Lund Jet Plane. arXiv: 2301.07732 [hep-ph] (Jan. 2023).

16. Bierlich, C. *et al.* A comprehensive guide to the physics and usage of PYTHIA 8.3. arXiv: 2203.11601 [hep-ph] (Mar. 2022).

17. Dumoulin, V. & Visin, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285* (2016).

18. Saha, S. A Comprehensive Guide to Convolutional Neural Networks. *Towards Data Science.* https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (2018).

19. Larkoski, A. J., Salam, G. P. & Thaler, J. Energy Correlation Functions for Jet Substructure. *JHEP* **06,** 108. arXiv: 1305.0007 [hep-ph] (2013).

20. Marzani, S., Soyez, G. & Spannowsky, M. *Looking Inside Jets* https://doi.org/10.1007%2F978-3-030-15709-8 (Springer International Publishing, 2019).

21. Almeida, L. G., Backović, M., Cliche, M., Lee, S. J. & Perelstein, M. Playing Tag with ANN: Boosted Top Identification with Pattern Recognition. *JHEP* **07,** 086. arXiv: 1501.05968 [hep-ph] (2015).

22. Cogan, J., Kagan, M., Strauss, E. & Schwartzman, A. Jet-Images: Computer Vision Inspired Techniques for Jet Tagging. *JHEP* **02,** 118. arXiv: 1407.5675 [hep-ph] (2015).

23. De Oliveira, L., Kagan, M., Mackey, L., Nachman, B. & Schwartzman, A. Jet-images — deep learning edition. *JHEP* **07,** 069. arXiv: 1511.05190 [hep-ph] (2016).

24. De Favereau, J. *et al.* DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP* **02,** 057. arXiv: 1307.6346 [hep-ex] (2014).