SAPIENZA
UNIVERSITÀ DI ROMA

Search for New Physics with Anomaly Detection approach at the LHC

Facoltà di Scienze Matematiche, Fisiche e Naturali
Bachelor's Degree in Physics

Valerio Tinari
ID number 1998628

Advisors
Stefano Giagu
Graziella Russo

Academic Year 2023/2024

Search for New Physics with Anomaly Detection approach at the LHC
Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: tinari.1998628@studenti.uniroma1.it

A Francesco e Paola,
mie guide

Contents

Chapter 1

Introduction

It was the 4 July 2012 when Joseph Incandela, CMS spokesperson, affirmed: "*They combine to give us a combined significance of five standard deviations*". With these exact words, he announced, confirmed by ATLAS spokesperson Fabiola Gianotti a few minutes later, the first observation of the Higgs boson. Twelve years have passed since then and it remains the last big breakthrough in High Energy Physics. What happened after that? Has Particle Physics come to an end and everything that needed to be discovered has been found?
In reality this field of research is as active as it was a few years back, currently pursuing precision's measurements and search for deviations from the Standard Model, such as Dark Matter. In particular this second kind of research has been conducted using traditional data analysis techniques but, unfortunately, anything has been found yet. There are many reasons for this unexpected outcome: wrong Physical model for experiments, wrong triggering while selecting data or deviations too small to be observed through normal techniques.
A possible solution could be Anomaly Detection, that is the search for data deviating from what is already known. In our field of interest, Particle Physics, we apply this method to look for anomalies (signal data), deviating from well known reacting and decaying particles (background data). The strength of this method is that it is model agnostic, meaning that it does not rely on a specific theoretical model to analyse data. This allows us to explore a more vast range of events and gives us a first hint of the region where to look more in deep for New Physics with precise data analysis techniques. In this process, Machine Learning (ML) and Deep Learning architectures (DL) come to help and speed up the process.
An example case has been investigated in this thesis in order to understand the potential of this tool. Using Monte Carlo generated data as sample, anomalies in particles decay have been studied, implementing a DL architecture, in particular a Transformer. This thesis has been an opportunity to dive into this topic and become confident with tools used in this field of research.
The thesis is structured as follows. Chapter 2 introduces the LHC and ATLAS, experiments to investigate particle collisions. Chapter 3 presents Anomaly Detection approach. In chapter 4 the dataset analysed is presented. In chapter 5 is possible to find a detailed description of the architecture implemented. In conclusion, chapter 6 shows results of this project.

Chapter 2

The LHC and ATLAS experiments

2.1 The LHC

The Large Hadron Collider (LHC), with its 27 kilometres ring [4], is the world largest particle accelerator. It is part of the CERN's accelerator complex and it is situated around 100 metres underground to shield from radiation and for both economic and geological reasons. With its first run on 10 September 2008, it currently accelerates hadrons, in particular protons or lead nuclei. The LHC is composed by two circular beam pipes where two particle beams travel separately in opposite direction, close to the speed of light. Due to the very high speed, when the two beams collide, the energy of the center of mass, known as $\sqrt{s}$, is twice the energy of each beam (which is the same). In fact, knowing that $E^2 = m^2c^4 + p^2c^2$ and using the natural unit system ([m]=eV/c^2, [p]=eV/c), we get: $E^2 = m^2 + p^2 \simeq p^2$, where at the end an ultra-relativistic particle (p»m) was assumed. This information is used in the norm of the four-momentum of two colliding particles [10]:

$$\sqrt{s} = \sqrt{P_\mu P^\mu} = \sqrt{E_{tot}^2 - |\vec{p_{tot}}|^2} = \sqrt{(E_1 + E_2)^2 - |\vec{p_1} - \vec{p_2}|^2} \simeq 2E \qquad (2.1)$$

where we assumed $E_1 = E_2$. At the LHC protons are accelerated with an energy of 6.8 TeV (10^{12} eV), which confirms the assumption p»m and it leads to $\sqrt{s} = 13.6$ TeV [22]. Such high energies are reached through a series of increasing accelerators, shown in figure 2.1, where Radio-frequency cavities accelerate particles, through an oscillating electric field.
Accelerated beams are directed and curved by superconducting electromagnets with a magnetic field of up to 8.3 T [15]: 1232 dipole magnets and 392 quadrupole magnets are used. Although its huge contribution, magnetic field also represents one of the main limitations of this apparatus. In fact from the Lorentz force $\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$, it is possible to get a connection between external magnetic field $\vec{B}$, particle momentum and its radius of curvature, which is [10]:

$$p = qBR \qquad (2.2)$$

where q is the charge of the particle and R the radius of curvature. It is evident

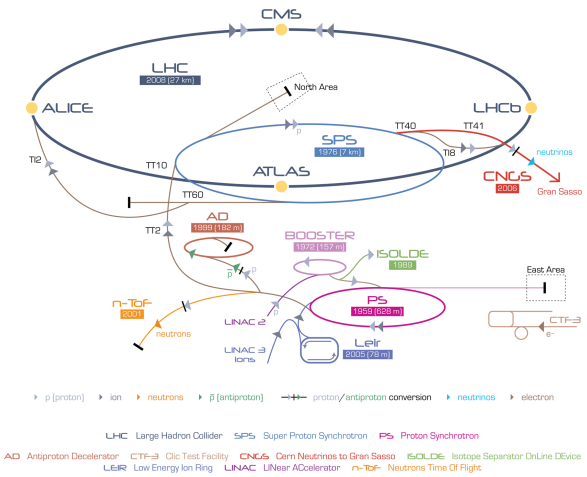

Figure 2.1 CERN accelerating system. Figure from Ref. [15]

that, working with high energy particles, to reduce R we need to generate a bigger magnetic field, but current technologies set a limit to this value.
Performance of the LHC are evaluated based on an essential parameter: the instantaneous luminosity $\mathcal{L}$, measured in $cm^{-2}s^{-1}$ and computed as follows [22].

$$\mathcal{L} = \frac{N_b^2 n_b f_{rev} \gamma_r}{4\pi\epsilon_n \beta^*} F \tag{2.3}$$

where: N_b is the number of particles in a bunch, n_b is the number of bunches in a beam, ϵ_n is the normalized transverse beam emittance, β^* is the beta function at the collision point, γ_r is the relativistic factor, f_{rev} is the revolution frequency. F represents the geometric luminosity reduction factor:

$$F = \frac{1}{\sqrt{1 + \left(\frac{\theta_c \sigma_z}{2\sigma^*}\right)^2}} \tag{2.4}$$

where θ_c is the full crossing angle at the interaction point, σ_z is the RMS bunch length and σ^* is the transverse RMS beam size at the interaction point. Luminosity is linked with the number of collisions per second [10]:

$$\frac{dN}{dt} = \mathcal{L}\sigma_{tot} \tag{2.5}$$

where σ_{tot} is the total cross section. If the instantaneous luminosity goes below a fixed threshold, beams are dumped.

2.2 ATLAS

A Toroidal LHC ApparatuS (ATLAS) [3] is the world largest particle detector, situated at interaction point 1 at CERN. It has many layers in a cylindrical symmetry

[1] Building LHC on the surface would have required the purchase of properties to do so and an under the open sky structure with a 27 km circumference would have been more expensive than agreeing with landowner and excavate a tunnel underground.

around the beam pipe. 44m in length and 25m in height are required to investigate collisions between particles accelerated by LHC, to study phenomena such dark matter, the Higgs boson and top quark. The main components of the detector are: an inner detector, an electromagnetic and a hadronic calorimeter, and the muon spectrometer, as shown in Figure 2.3.
The inner detector relies on a 2 T solenoidal magnetic field, parallel to the beam axis, to reconstruct the tracks of charged particles. This information is used to measure the momentum of them using the sagitta method and the relation between momentum and radius of curvature, presented in (2.2).
The electromagnetic calorimeter [20] measures the energy of electrons, photons and hadrons. Once a particle among them hits the detector, an electromagnetic shower is generated and the energy of the particle is transferred to the material through radiation, as shown in figure 2.2. Passive components of the calorimeter such as lead induce the loss of energy and the active ones, made by Liquid Argon (LAr) kept at a temperature of -184 °C, measure it. The shower is fully contained in this part of the detector, which is longer then 20 radiation lengths. For this reason this technique to measure energy is defined *destructive*.

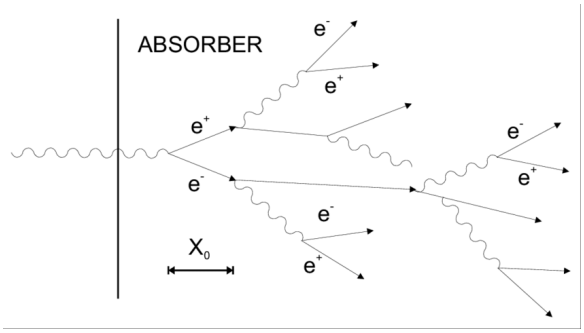

Figure 2.2 Schematic of an electromagnetic shower starting from a photon

The hadronic calorimeter has been placed to measure the energy of hadrons that survive to the electromagnetic calorimeter. It relies on the principle of particle shower, which is converted into an electric current and measured.
The muon spectrometer is the last component of the ATLAS experiment and measures the momenta of muons. In fact, they are similar to electrons, but with a greater mass, which leads only to weak interactions and more depth of penetration. This is why they are not revealed by inner systems.
The coordinate system adopted at ATLAS to describe events is discussed in detail in section 4.1.

2.3 Problems in Particle Physics

The 20th century was a golden era in Particle Physics, due to the high number of discoveries happening during it. Thanks to them we now have a complete overview of how the subatomic world works. The overview was completed in 2012 with the first observation of the Higgs boson. Everything we know is contained in the Standard

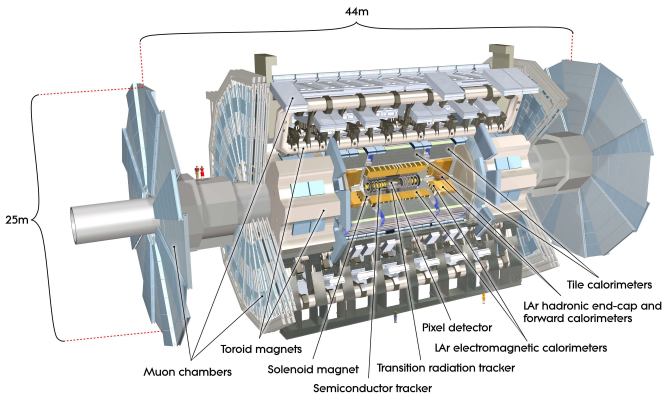

Figure 2.3 ATLAS schematic. Figure form Ref. [20]

Model of particle physics[1]: 6 quarks (up, down, charm, strange, top, bottom), 6 leptons (electron, muon, tau and corresponding neutrinos), 4 bosons that mediate the 4 fundamental forces (strong, weak, electromagnetic, gravitational) and the Higgs boson giving them a mass by the homonymous mechanism. The SM describes very well these phenomena but there are still many problems that need to be solved.

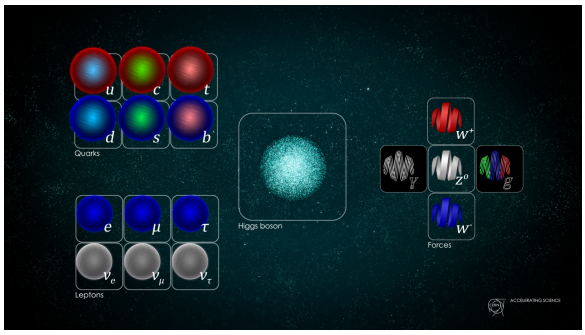

Figure 2.4 Particles of the Standard Model. Figure from Ref. [1]

Among all unsolved mysteries, we can find: unification of gravitational interaction and other interactions, the mass of neutrinos, the baryon - anti-baryon asymmetry in the Universe and Dark Matter. In particular the last is giving physicists a hard time. From cosmological observation it is known that ordinary matter constitutes only 5% of the mass of the Universe, 27% is Dark Matter and 68% is Dark Energy. If we investigate the last two using particle accelerators, we expect them to leave the experiment undetected due to their low ratio of interactions and then from missing energy we can observe their presence. Anyway, using traditional data analysis techniques, they have not been observed yet on the Earth. One possible explanation for this "failure" could be the limits of model depending theoretical model used. In this situation a model agnostic approach such as Anomaly Detection comes to help.

Chapter 3

Anomaly Detection

3.1 What is it?

Normally, search paradigms in Particle Physics use to investigate collisions happening at the LHC target a specific physical model. This approach has a limited field of application and so far it showed being ineffective to find New Physics (NP). There are two possible reasons behind this failure: there is no New Physics at current LHC energy or "traditional" search paradigms are not efficient with NP. At this point it is time to try something different in order to hope to understand something more about Beyond Standard Model phenomena: Anomaly Detection could be the solution.
When applying Anomaly Detection [2] [16], we are looking for outliers or rare events, which typically have a small deviation from already known events. In this case data is divided into two categories: "background", containing Standard Model events or noise, and "signal", including the hypothetical New Physics. All SM phenomena are classified within background without any distinction based on the origin of data. It is now possible to understand why Anomaly Detection is defined "model agnostic": in fact it does not rely on a specific theoretical model. This feature allows to scan a bigger range of events but it is less sensitive to any specific model.

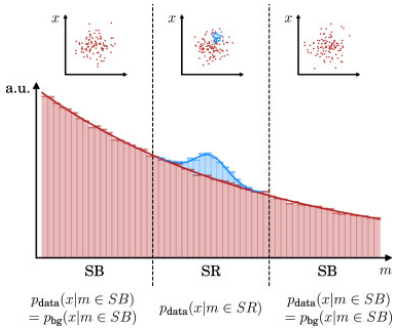

Figure 3.1 Search for anomalies in particle physics. Figure from Ref. [12]

To understand Anomaly Detection, we can use an example. Assuming we need to find a church in a town we do not know, without any technology. Instead of

desperately walking around, we decide to climb an hill where we can scan the whole town with not so much effort. It is possible to identify the neighbourhood of the church or the "spatial region" where it is situated. We can also get some information we would not have got in the town, such as the colour of the roof or of the facade of the building we are looking for. Then we climb down the hill and go straight to the point we observed. Thanks to the details we obtained up there, we are able to find the church quite quickly and more easily. In other words, the church represents New Physics, climbing the hill is Anomaly Detection, climbing down is using normal data analysis techniques. It is quite powerful, isn't it? To apply Anomaly Detection it is necessary to hypothesize that anomalies are rare and they represent a small portion of a whole dataset. This assumption is realistic because it is what happens everyday in airports, finance and particle physics: normal events are way more common then deviations. In addition, if anomalies in Particle Physics were more frequent, we would have already observed them through common data analysis.

3.2 Deep Learning and Anomaly Detection

While working on a Particle Physics problem, it is common to deal with a huge amount of data collected during collisions at the LHC. Machine Learning (ML) algorithms, which are able to learn from past by using vast datasets, can help in this sense. Among all categories of ML, Deep Learning (DL) is the best for our task [16]. Based on a multi-layer Neural Network, DL can build complex data representation, which is optimal when dealing with high-dimensional dataset. Furthermore due to its implementation, that will be discussed in chapter 5, it can be exploited for model agnostic applications. [12]
ML and DL methods can be classified based on how the labels of data are used to train models. Main categories are: Supervised Learning, Unsupervised Learning and Semi-Supervised Learning.[2]
In Supervised Learning, shown in figure 3.2, labeled data is used to train DL models and it can be implemented only with sufficient normal and anomalous data.

Figure 3.2 Data used for Supervised Learning. Figure from Ref. [2]

Unsupervised Learning is based on not labeled data, as shown in figure 3.3, and models implementing it are trained to find a structure among the input features. Then model performance is tested using a mixed dataset containing both background and signal data.
Semi-Supervised Learning is a hybrid of previous techniques, implemented in the most varied ways.
In this project we implemented an Unsupervised Anomaly Detection, because labels have not been used and model was trained only over background data.

Figure 3.3 Data used for Unsupervised Learning. Figure from Ref. [2]

3.3 How is it implemented?

The process of training a DL model is divided into three main phases: training, validation and test loop. They are defined "loop" because in each of them a different section of the dataset is used and the code loops over it.
The most important stage is the training loop: in our case the chosen DL model is trained to reconstruct arrays containing information about particle jets, using only data with normal behaviour (background events) [2]. In this phase, parameters of the model are adjusted in order to best fit to data. By doing so, DL model is able to recognise very well normal data and their structure. Our hope is that, once it faces an anomaly (signal event), it is able to identify it due to its difference from what it knows. The second step is the validation loop, where a valuation of the performance of the model is elaborated using the validation set. The valuation and the search for the best weights of the model are executed computing distance between output and input over training and validation set. When this distance is minimum within the validation loop, we save the model's weights.
The trained and tuned model is then tested, using a dataset containing unseen data, the test set, where there are both background and signal events. The purpose of this phase is to validate results previously obtained and get a preview of what will happen once the model will be running with completely new data. The task is completed computing the Anomaly Score (AS). In this thesis, we implement it as a distance between input and output, with different dimension: instead of calculating it over a batch of data, we do so for each event. In this way we are able to plot a reconstruction probability graph: we expect that the distributions of AS from background and signal events are different, because signal data contains anomalies which our model does not know. By means of these distributions we can identify a threshold of separation between them. For any event we pass to our trained model, we compute AS: if it higher (lower) than the chosen threshold, we are looking at a background (signal) event. Since AD is distinguishing signal events from background, we are basically executing a classification task with two classes. Implementation of Anomaly Detection is discussed more in depth in chapters 5 and 6.

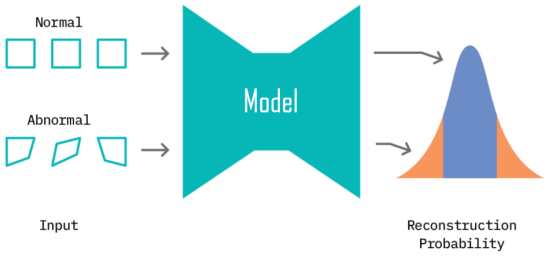

Figure 3.4 Test loop of Anomaly Detection. Figure from Ref. [2]

Chapter 4

LHC Olympics 2020 Dataset

4.1 ATLAS coordinate system

The dataset we use to develop this thesis is composed by positions and momenta of simulated particles originating after a collision. For this reason, prior to discussing its structure, it would be appropriate analysing the coordinate system active in a particle detector, such as the ATLAS experiment [15] [20]. In figure 4.1 an overview of the system is presented.

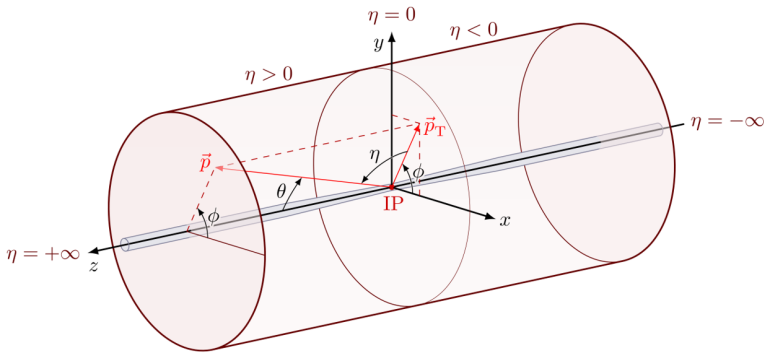

Figure 4.1 Schematic of ATLAS coordinate system

Cartesian axes (*x,y,z*) are constructed as follows. The origin is set in the interaction point (IP). From it, the z-axis departs parallel to the beam tunnel with positive z-values in anticlockwise direction. As direct consequence, the x-y plane is perpendicular to it, with positive x-values towards the center of the LHC and positive y-values upward. On this Cartesian system, a cylindrical coordinate system (r, η, ϕ) is defined to describe geometric position and components of particle momenta. ϕ is the azimuth angle, measured around the z-axis, with values $[-\pi, \pi]$. η is named pseudo-rapidity and it is linked to the polar angle θ by the equation below:

$$\eta = -ln\Big(tan\frac{\theta}{2}\Big) \tag{4.1}$$

The pseudo-rapidity and the azimuth angle, known as geographical coordinates [20], are used to define a geometrical distance, applied when computing the separation

between two physics objects:

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \tag{4.2}$$

The third coordinate, r, is the radius from the center of the system. Actually, in this thesis, we use another information: the transverse momentum p_T. It is a 2D vector defined as the projection of the momentum in the transverse plane of the interaction. This parameter is related to the radius and it is important because it brings information about energy. In figure 4.2 it is possible to observe how p_T is calculated.

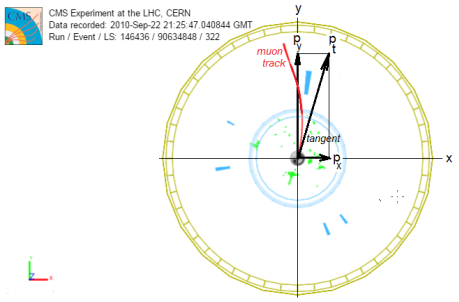

Figure 4.2 Schematic of p_T computing in the CMS experiment, analogous to ATLAS

4.2 LHC Olympics Dataset

In this thesis we deal with the R&D dataset created for the LHC Olympics 2020 [16], a challenge promoted by CERN collaboration to contribute to New Physics searches. It is made up of 1 million background events and 100 000 signal events, the anomalies we are looking for. The background is associated to SM phenomena, especially quantum chromodynamics (QCD) dijet events. The signal is about the resonance of Z' particle: $Z' \to XY$, with $X \to q\bar{q}$ and $Y \to q\bar{q}$. The masses of the particles are: $m_{Z'}$= 3.5 TeV, m_X= 500 GeV, m_Y= 100 GeV. The topology of the interaction is presented in figure 4.3.

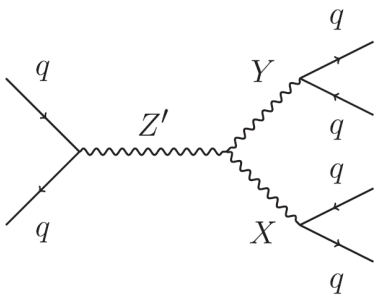

Figure 4.3 Topology of the signal interaction. Figure from Ref. [16]

A resonance is used in the signal dataset because the dijet final state offers a complex topology for hiding new physics.
The dataset is produced in Monte Carlo (MC) simulations using Pythia 8.219 and

Delphes 3.4.1 with default settings, simulating an ideal particle detector. Events are selected using an anti-k_t jet trigger with R=1 and a p_T threshold of 1.2 TeV.
For each event we have information about up to 700 particles and for each of them, we have three features, p_T, η and ϕ. Zero padding is applied, meaning that if an event has less then 700 physical data, zeros are added in order to get a perfect rectangle. Together with spatial information, a truth value is appended to every event in order to know whether it is a background (value=0) or a signal event (value=1). In conclusion the dataset has shape: (1.1 M, 2101).

Jet clustering

Quarks in the final state of our events (both background and signal) hadronize in the calorimeters. This phenomenon induces sprays of particles in the detector, defined jets. Prior to applying AD, data needs to be pre-processed in order to reconstruct jets. The reconstruction is performed applying the anti-k_t algorithm [21]. This method relies on the distance d_{ij} between two constituents (i, j) and the distance d_{iB} between a jet i and the beam B:

$$\begin{aligned} d_{ij} &= min(p_{Ti}^{2k}, p_{Tj}^{2k}) \frac{\Delta_{ij}^2}{R^2} \\ \Delta_{ij}^2 &= (y_i - y_j)^2 + (\phi_i - \phi_j)^2 \\ d_{iB} &= k_{ti}^{2p} \end{aligned} \tag{4.3}$$

where, working with the constituent i, p_{Ti} is its transverse momentum , y_i is its rapidity and ϕ_i its azimuth; R is the radius parameter of the algorithm and k is an additional parameter, equal to -1 in case of anti-k_t algorithm, about the power of energy.

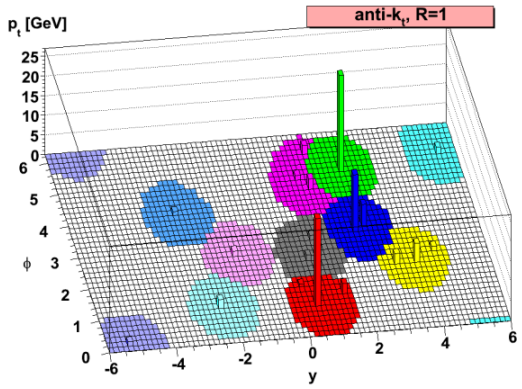

Figure 4.4 Example of anti-k_t application. Figure from Ref. [21]

We apply the anti-k_t algorithm in this thesis using the Fastjet implementation and the radius parameter is set to R=1.
After the reconstruction, for each "physical" event we select the two most intense jets,

defined as leading jet and sub-leading jet, if present. This means that the dimension of the whole dataset is multiplied by 2, becoming 2.2 M events. For each event we select up to 50 constituents and for each of them we still have the three features (p_{Tfrac}, η, ϕ), where $p_{Tfrac} = \frac{p_{Tconst}}{p_{tjet}}$. Zero padding is applied and the truth value is preserved. The new shape of the dataset is (2.2 M, 50, 3), with 2 M background events and 200k signal events.
In figure 4.5 it is possible to look at the distributions of the three features constituting the reconstructed dataset, for both background and signal events. In this plot events with p_{Tfrac}, and then η and ϕ, equal to zero are excluded, because they are not physical but just padding data.

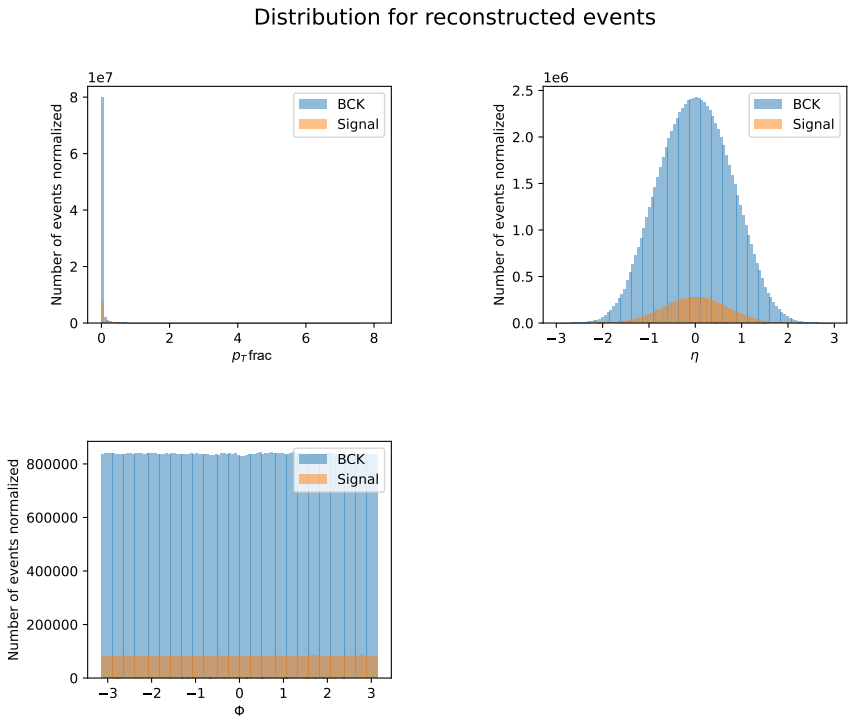

Figure 4.5 Distribution for reconstructed events

To properly train our model and avoid misclassification error, data is scaled before moving to the next phase of the project. Data scaling is executed applying a standardization, implemented through the Standard Scaler function [9], leading to a dataset with mean equal to 0 and standard deviation equal to 1. The task is completed passing each data x to the following equation:

$$z = \frac{x - \mu}{\sigma} \tag{4.4}$$

where μ is the mean of the original dataset, σ its standard deviation and z is the scaled data. The new data distribution is presented in figure 4.6, where also jets with p_T=0 are plotted.

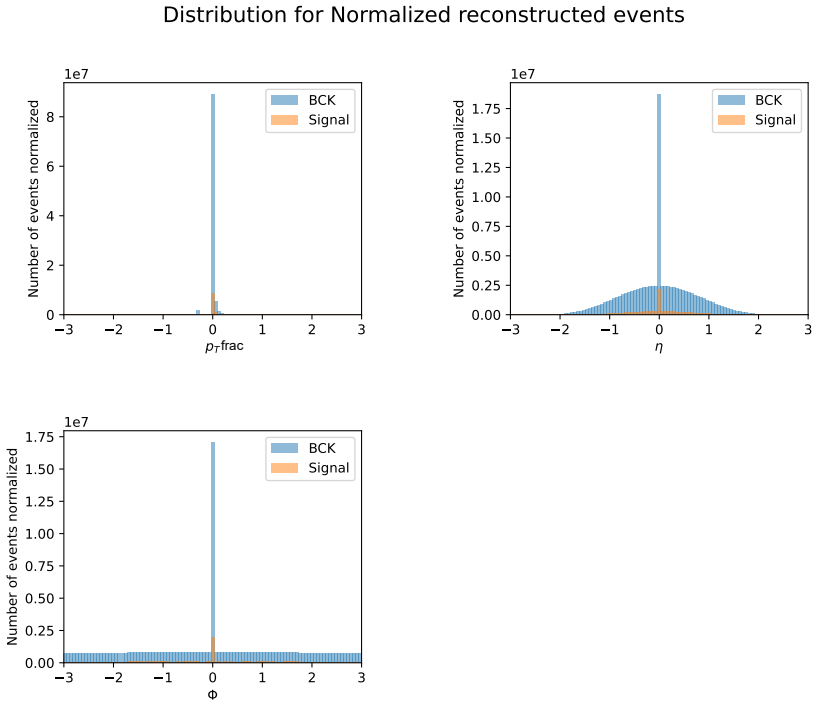

Figure 4.6 Distribution for reconstructed and normalized events

Jet transformation

At this point of the thesis it is essential to admit that we want to focus on the micro-structures of a jet, based on the assumption that New Physics could be found there. Therefore the absolute position of the jet in the (η, ϕ) plane is irrelevant, resulting in rotational symmetries among data. To study the impact of these symmetries either we let our model learn them (using the dataset described above) or we prepare new dataset removing them. One dataset is created applying a rotation; we will refer to it as the "rotated" dataset. The other is created to determine the correlation of mass with training results; we will call it the "transformed" dataset.
The rotation, is implemented using two Lorentz boosts. Given the coordinates of the jets η_{jet} and ϕ_{jet}, two Lorentz boost are applied consecutively in order to obtain $\eta'_{jet} = 0$ and $\phi'_{jet} = 0$. They are applied to all constituents of the jet. In figure 4.7 and 4.8 new distribution plots are presented.
The transformation consists of few steps [17].
Prior to describing this method, a proper notation needs to be introduced. P_J^μ $(\vec{P}_J)$ is the four-momentum (three-momentum) of the jet, p_i^μ $(\vec{p}_i)$ is the same entity but related to the i^{th} constituent. These vectors are used to describe the energy of the jet:

$$P_J^\mu = \sum_{i=1}^{N_J} p_i^\mu \ , \quad \mathrm{m_J^2} = |\mathrm{P_J^\mu}|^2 = \left(\mathrm{P_J^0}\right)^2 - \left(\overrightarrow{P}_\mathrm{J}\right)^2 \ , \quad \mathrm{E_J} = \mathrm{P_J^0} \tag{4.5}$$

where N_J is the number of constituents, m_J is the jet mass and E_J the jet energy.

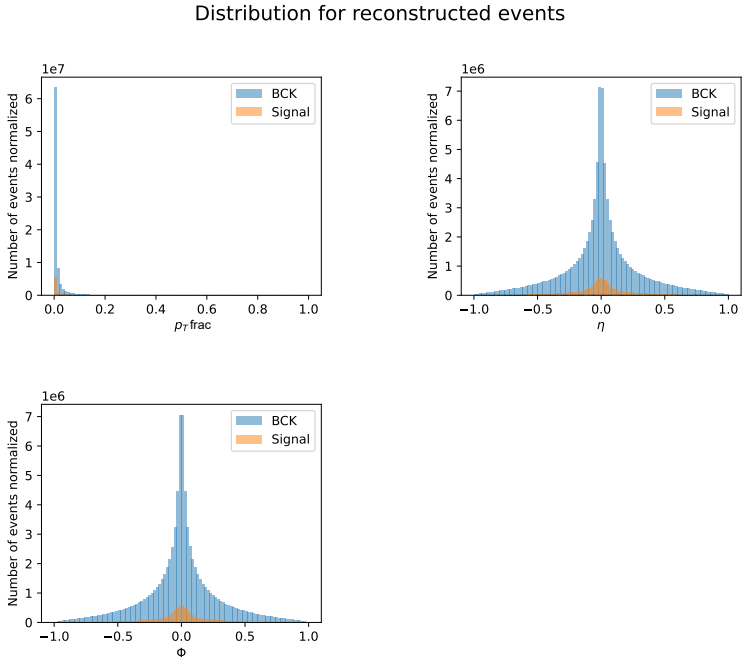

Figure 4.7 Distribution for reconstructed and rotated events

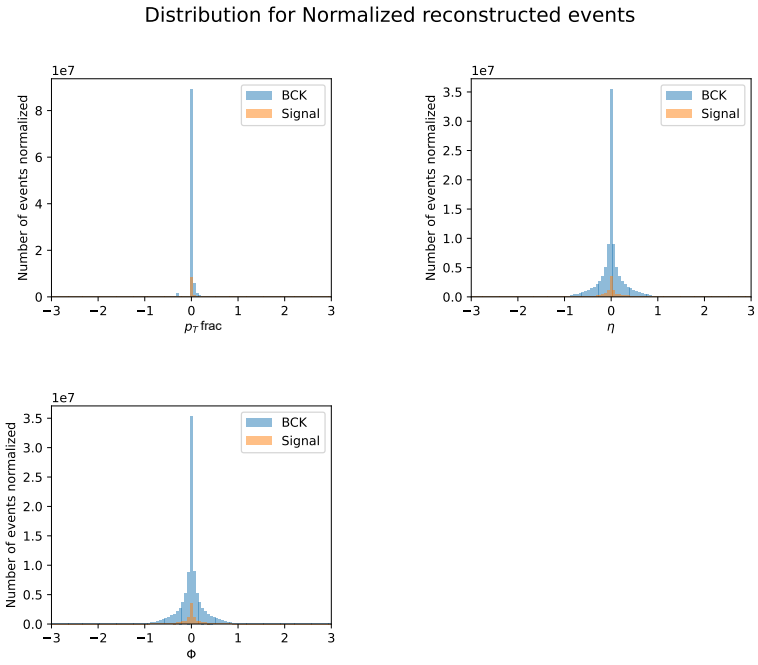

Figure 4.8 Distribution for reconstructed, rotated and normalized events

First of all, we rescale the jet constituent four-momenta in order to set the jet mass to a value m_0=250 MeV:

$$p_i^\mu \to p_i^{'\mu} = \frac{m_0}{m_J} p_i^\mu \ , \quad \mathrm{P_J^\mu} \to \mathrm{P_J^{'\mu}} = \frac{\mathrm{m_0}}{\mathrm{m_J}} \mathrm{P_J^\mu} \ , \quad \mathrm{E_J} \to \mathrm{E_J^{'}} = \frac{\mathrm{m_0}}{\mathrm{m_J}} \mathrm{E_J} \tag{4.6}$$

Secondly, a Lorentz boost (Λ_ν^μ) is performed to set the energy of all jets to a value E_0= 1 GeV in the new frame of reference:

$$p_i^{'\mu} \to \mathbf{p}_i^\mu = \Lambda_\nu^\mu p_i^{'\nu} \ , \quad \mathrm{P_J^{'\mu}} \to \mathbf{P}_\mathrm{J}^\mu = \Lambda_\nu^\mu \mathrm{P_J^{'\nu}} \ , \quad \mathbf{E}_\mathrm{J} = \mathbf{P}_\mathrm{J}^0 = \mathrm{E}_0 \tag{4.7}$$

where letters in bold are associated to boosted jets. In conclusion, the rotation is applied. The distribution of the transformed dataset is visible in figure 4.9 and 4.10.

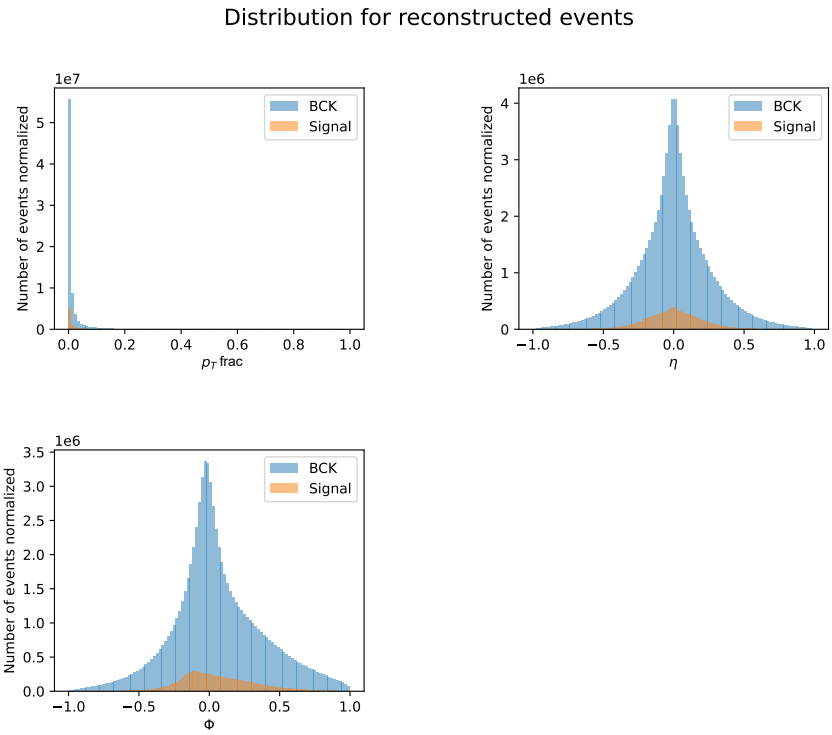

Figure 4.9 Distribution for reconstructed and transformed events

At this point, we expect our model to perform better and to be able to distinguish between the two kind of events. Our hope relies on the fact that we removed rotational symmetries and for kinematic reasons [10], all jets will be pushed forward and they will have a smaller spatial aperture, given the fact that the new energy of jets E_0 is lower than original energy. This feature will help our model during training, allowing it to focus on details and small differences.

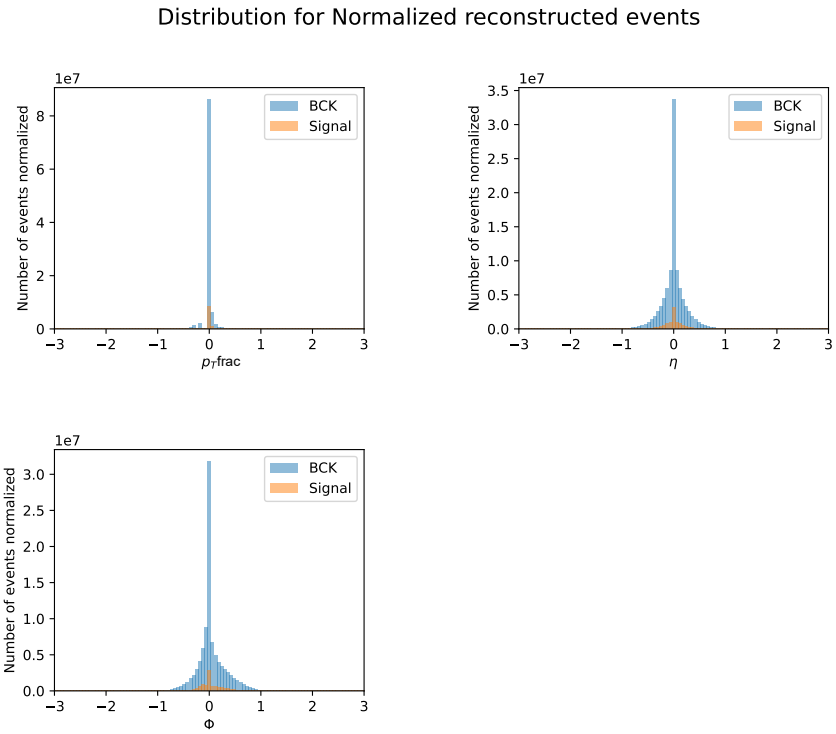

Figure 4.10 Distribution for reconstructed, transformed and normalized events

Chapter 5

Deep Learning and Transformer architecture

5.1 Introduction to Deep Learning

Does our model magically learn to reconstruct input jets and distinguish between background and signal? No, it does not: it is nothing more than math formulas and in this chapter it is briefly discussed how it is done.
Deep Learning tries to replicate the behaviour of a human brain and, analogously, its basic unit is named "neuron" [18] [19]. From its dendrites it receives input values; they are processed and then summed in a weighted sum. After applying an activation function (described later), the signal is sent in output through the axon, towards the next dendrites. Neurons are organized in layers, constituting a network. The first one is defined *input layer* and it receives original data as input; the last one is the *output layer* and its output is the output of the model. Between them there are hidden layers, performing mathematical computations. A model with just one hidden layer is Perceptron. If the number of layers is large, the model is defined as Deep Network.

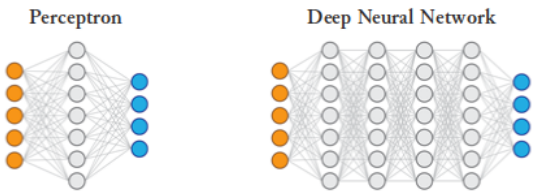

Figure 5.1 Comparison between a Perceptron and a Deep Network. Figure from Ref. [18]

The aim of a model training is to find the best function that connects input and output data. In order to achieve this goal, we reconstruct the function connecting a layer to the following, which is [17]:

$$O_i^L = \sigma\Big(\sum_j W_{ij}^L I_j^L + B_i^L\Big) \tag{5.1}$$

where I_i^L is the input vector, O_i^L the input vector, W_{ij}^L is the weight matrix (which

has to be optimized during training), B_i^L represents the bias vector for the layer L and σ is the activation function applied before signal leaves the neuron. When dealing with an hidden layer, we assume $I_i^L = O_j^{L-1}$. The activation function is a mathematical function setting a threshold to determine whether the neuron is active or not. In this thesis we used a GELU (Gaussian Error Linear Unit) activation [13]. It is computed as follows:

$$\sigma(x) = x \cdot \frac{1}{2}\Big[1 + erf\Big(\frac{x}{\sqrt{2}}\Big)\Big] \simeq x \cdot \frac{1}{2}\left[1 + tanh\Big[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\Big]\right] \tag{5.2}$$

where the erf function is defined as:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{5.3}$$

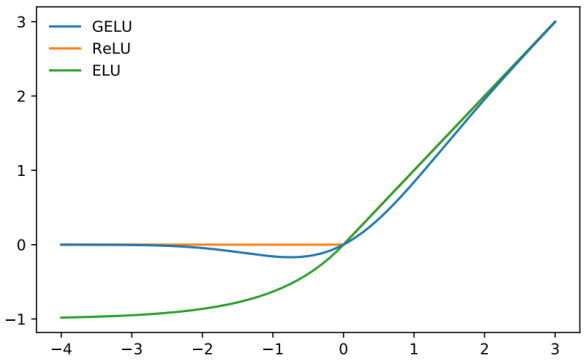

Figure 5.2 Comparison between GELU, ReLU and ELU [1]activation Figure from Ref. [13]

As explained above, the model implementation is based on math and also the process of learning does. A loss function $\mathcal{L}$ is defined from the distance between the input and the output of the model. During the training phase we try to minimize this function adjusting weights of the model. A possible method to do so is the stochastic gradient descent (SGD). Defining $\mathbf{p} \in \mathbb{R}^s$ as the vector containing weights and biases, SGD is computed [18]:

$$\mathbf{p} \to \mathbf{p} - \eta \left(\frac{\partial \mathcal{L}}{\partial p_r} \Delta p_r \right)_{x^{\{i\}}} \tag{5.4}$$

where η is the learning rate, an hyperparameter that has to be optimized, computed as the ratio between the step size and the absolute value of the gradient. Δp_r is a perturbation associated to the step. To calculate the gradient back-propagation is applied: weights are computed while forward feeding the model and then computed again when data passes backwards. Values are combined to present the best estimation of weights.

[1]ReLU = Rectified Linear Unit activation function; ELU = Exponential Linear Unit activation function

5.2 Proposed model: Transformer architecture

In order to complete the Anomaly Detection task, we implemented a Transformer architecture, a model based on the Multi-Head Attention mechanism (MHA)[11] [14]. It is defined "Multi-Head" because it is implemented h times simultaneously (h stands for number of heads); it is "Attention" because it relies on the attention function, which maps a query and a set of key-value pairs to an output. The core aspect of this mechanism is that its weights are dynamic, meaning they change depending on the data that is passed as input. This feature allows the model to learn only the relevant parts of the dataset and, in our case, of the jets.
All just said is converted into concrete actions using the "Scaled Dot-Product Attention", which is a parallelism with the process of the retrieval in a database of a value v based on a query q and on a key k. In our case, queries and keys, of a dimension d_k, are packed respectively into a matrix Q and K. Also values, of a dimension d_v, are presented as a matrix V. The dot product of Q with K is computed and then we divide by $\sqrt{d_k}$. The output is still a matrix and it is computed as:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}}\right)V \tag{5.5}$$

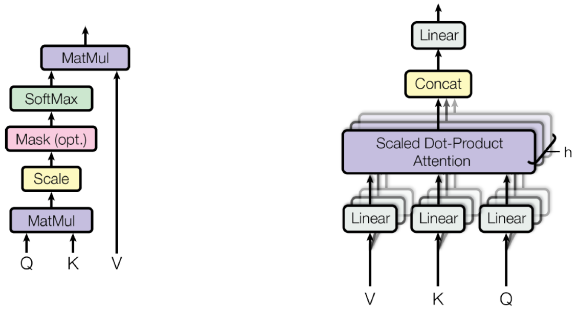

Figure 5.3 Scaled Dot-Product (left), MHA (right). Figure from Ref. [11]

Starting from the Scaled Dot-Product, we obtain the MHA computing in parallel multiple sets $i = 1, \ldots, h$ of keys, querys, and values, where h is the number of heads applied. In fact, Q, K, V are linearly projected in the respective dimension; then the attention mechanism is performed in parallel. Results are concatenated and projected, to give us MHA:

$$\begin{aligned} h_i(Q_i, K_i, V_i) &= Attention(QW_i^Q, KW_i^K, VW_i^V) \\ MHA(Q, K, V) &= concat[h_1, h_2, ..., h_h]W^O \end{aligned} \tag{5.6}$$

where the letter W stands for the projections in the different dimensions: $W_i^Q \in \mathbb{R}^{\mathrm{d_{model}} \times \mathrm{d_k}}$, $W_i^K \in \mathbb{R}^{\mathrm{d_{model}} \times \mathrm{d_k}}$, $W_i^V \in \mathbb{R}^{\mathrm{d_{model}} \times \mathrm{d_v}}$ and $W^O \in \mathbb{R}^{\mathrm{hd_v} \times \mathrm{d_{model}}}$.
The number of heads h is an hyperparameter that has to be optimized during the training phase, just as the learning rate.
MHA is the essential constituent of the Transformer architecture. This model has typically an encoder-decoder structure 5.4, meaning the input x is mapped to a

sequence of continuous representations z by the encoder and then turned into output by the decoder. We are going to focus only on the encoder because it is the part of the model that we implement in this thesis.

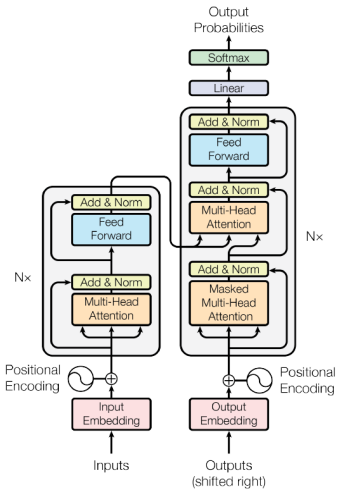

Figure 5.4 Schematic of the implementation of the Transformer model in the paper [11]

The model is a stack of L identical layers and each of them is composed by two sub-layers: MHA and a feed-forward network. The keys, values and queries to implement MHA in a layer are the output of the previous. The number of layers L is an hyperparameter that has to be optimized. The output will be an embedding for each input data.

Our Transformer implementation

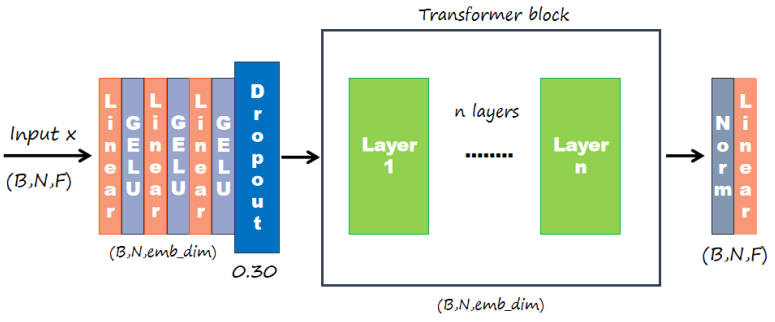

Figure 5.5 Schematic of our implementation of the Transformer model

Based on what is said above and on what is presented in the original paper [11], an implementation of the Transformer architecture is developed. Right after that, to go back to the input dimension, a linear layer is applied. In figure 5.5 a schematic of the implementation of the Transformer architecture is presented. Talking about dimensions, prior passing the training set of data to the model, the dataset is divided

into batches of shape (512, 50, 3) which corresponds to the shape (B, N, F) in the schematic. Once our data is in, using a series of three linear layers followed by GELU activation (5.2), it is taken to the desired dimension (512, 50, 128). At this point, a dropout layer is applied. It is about dropping out the nodes of a few randomly selected neurons in order to avoid overfitting, as shown in figure 5.6. The number of neurons involved depends on the probability associated to this process. In our case, a dropout of probability 0.30 is applied and then the vector is passed to L layers of MHA (green rectangles in the picture). As said before, after this crucial phase, our vector is taken back to the input dimension applying a Layer Normalisation and a Linear Layer.

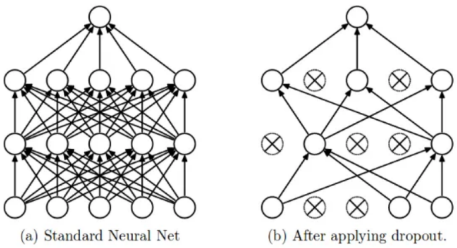

Figure 5.6 Schematic of dropout

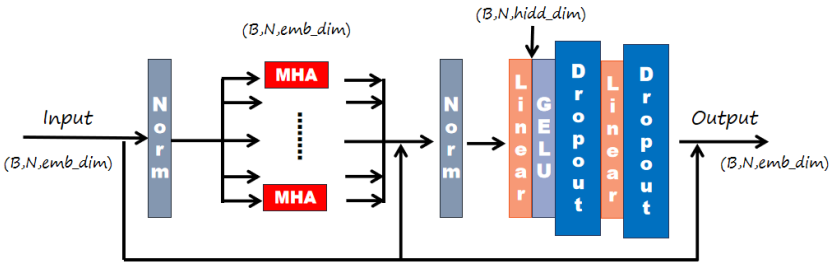

Figure 5.7 Schematic of our implementation of the MHA

Finally, in figure 5.7 our implementation of a layer of the model, containing the MHA is presented. It is basically quite similar to the original paper [11], where after a Layer Normalisation, MHA is applied h times. Then the input is summed to this output. The difference between the paper and this work is that we propose that at the end of the layer, after a few more linear and dropout layers, we add the input vector to the current vector of data, instead of adding the output of the second normalisation. This difference was presented assuming that an additional information about the macro-structure of the jet would have helped the model in the discrimination phase. However, this difference has been proved to have a minor impact on the final result.

Chapter 6

Training and results

6.1 Training the model

After pre-processing data and defining the architecture, it is time to train the model to discriminate between background and signal. To complete properly this task, we divide the dataset containing clustered jets into three sub-dataset: training, validation and testing set, used in the respective phase of training. In table 6.1 the number of events in each set is presented. As said in previous chapters, training and validation sets contain only background data; signal data is all inserted in the testing set, together with a small portion of background.

	background	signal
training set	72.7% = 1.6M	0%
validation set	9.1% = 200k	0%
testing set	9.1% = 200k	9.1% = 200k

Table 6.1 Number of events in each set, for both background and signal. Note that the percentage is the absolute percentage.

In order to follow the progress of the model learning, a loss function $\mathcal{L}$ is defined from the distance between the input and the output of the model: we use the Mean Squared Error (MSE) from PyTorch [6]. It is implemented as:

$$MSE = ||x - \hat{x}||^2 \tag{6.1}$$

where x is the input of the neural network and $\hat{x}$ is the output. It is computed as an average over each element in the batch. The goal of the training phase is to find the weights and the hyperparameters of the model that minimize the MSE value.
The process of model learning is made up of three phases: training, validation and testing. In the first part a loop of n epochs is executed, where in our case n we set it equal to 50, because no significant contributions were observed with a higher value. In each epoch, a loop over the training set, divided into batches, is executed; during this loop the weights are calculated. At the end we compute the MSE loss. Keeping frozen the weights, a loop over the validation set is executed, to valuate performance of the model. Once again, MSE is calculated. After n epochs, a plot with training

and validation losses for each epoch is created. We save weights and parameters of the model when the validation loss is minimum. By doing so we are sure to be looking at the best and tuned model in this circumstance. During this phase, the Adam optimizer is used, with the learning rate hyperparameter optimized.
Using the best model, a loop over the testing set is run, keeping the weights frozen. This set acts as a bunch of new data, unseen by the model, and it is used to confirm results previously obtained. A qualitative estimate of performance come from computing the Anomaly Score (AS) in the *linalg.norm* implementation from Numpy [5]. It is basically similar to MSE:

$$AS = ||x - \hat{x}|| \tag{6.2}$$

Apart from the squared term, the main difference about our MSE and AS computation is that we calculate AS for each data separately and not averaged over the batch. These values are then plotted with two different colours for background and signal events. We should get two distinct distributions. A classifier has been build: if we pass a new data to the trained model, we compute the AS and we compare it with the threshold to determine whether it is a background or a signal event. In figure 6.1 an example of AS is presented in order to give the reader a visive record.

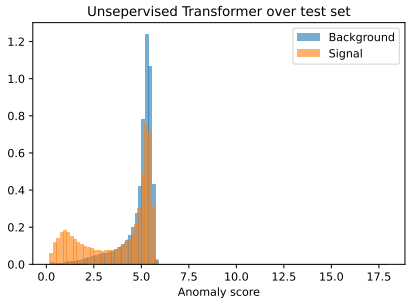

Figure 6.1 Example of AS distribution plot. In this case we would have set the threshold around 3

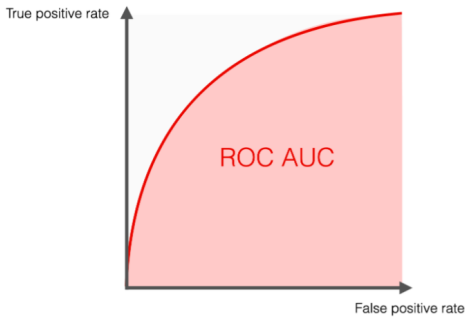

Figure 6.2 ROC curve and AUC explained

It is necessary to develop also a quantitative estimate of performance. We use the Area Under the Curve (AUC) [7] and the Receiver Operating Characteristic (ROC) curve [8]. They are computed starting from the confusion matrix, where four groups are presented to quantify misclassification errors: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). From them we can calculate rates:

$$TPR = \frac{TP}{TP + FN} \ , \quad FPR = \frac{FP}{TN+FP} \tag{6.3}$$

where TPR is the True Positive Rate and FPR is the False Positive Rate. ROC curve, as shown in figure 6.2, is a plot showing the trade-off between TPR and FPR at various classification thresholds. The curve is always over the line where TPR=FPR and the closer it is to the top left corner, the better our model is learning. AUC is, as the name suggests, the area under the ROC curve; it is computed using an integration method, for instance the trapezoidal rule. A random classifier can only obtain a AUC=0.5 (TPR=FPR); a perfect classifier flies to a AUC=1.

6.2 Results

As said in previous chapters, we need to optimize a few hyperparameters: learning rate η, number of heads in MHA h and number of layers L. We execute the optimization training our model with different choices for these values and looking at the ones giving us the best results in terms of ROC and AUC. The best values we found during the training of the model are presented in table 6.2, where "anti-k_T" stands for the not transformed dataset.

dataset	η	h	L	best epoch	AUC
anti-k_T	10^{-5}	8	5	10	53%
rotated	10^{-6}	8	32	27	66.4%
transformed	10^{-6}	2	4	47	69.0%

Table 6.2 Results of our model training: best hyperparameters, epoch for the best model and AUC

Finally, results of our study are presented in figures below and then commented.

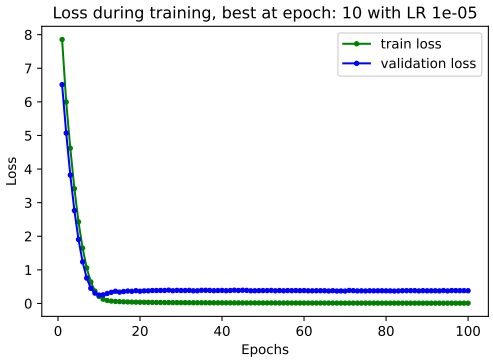

Figure 6.3 Loss evolution, anti-k_T dataset

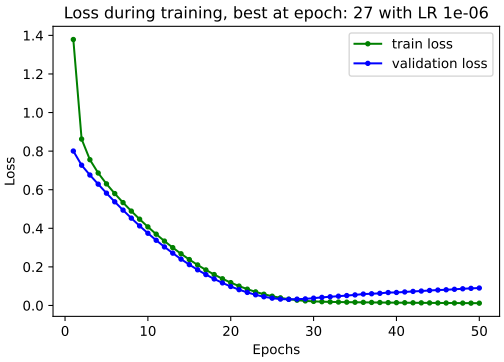

Figure 6.4 Loss evolution, rotated dataset

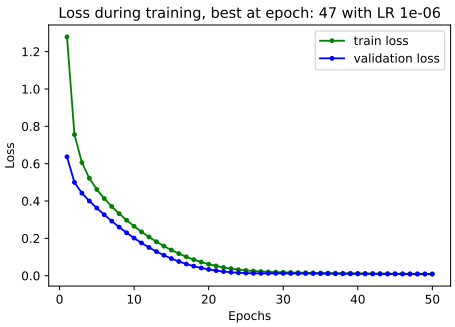

Figure 6.5 Loss evolution, transformed dataset

In figure 6.3, 6.4 and 6.5 you can look at training plots showing the evolution of the MSE loss over training and validation set for each dataset; in figure 6.6 , 6.7 and 6.8 we present Anomaly Score distributions. In conclusion, in figure 6.9 , 6.10 and 6.11

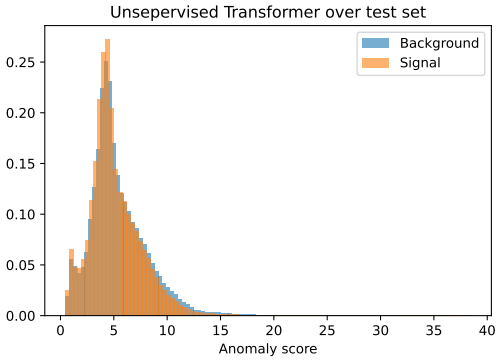

Figure 6.6 AS for the anti-k_T dataset

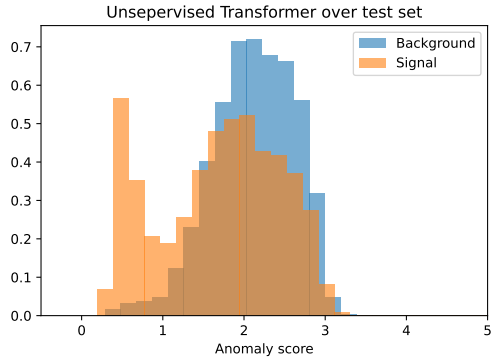

Figure 6.7 AS for the rotated dataset

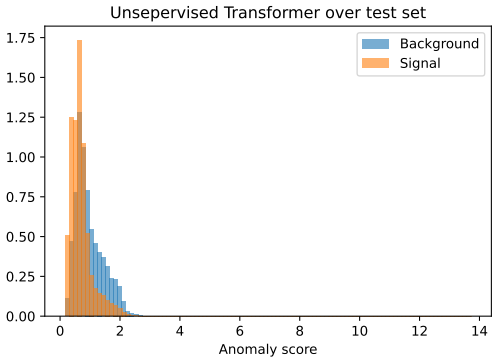

Figure 6.8 AS for the transformed dataset

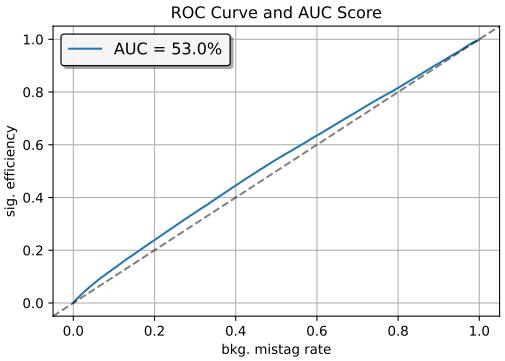

Figure 6.9 AUC for the anti-k_T dataset

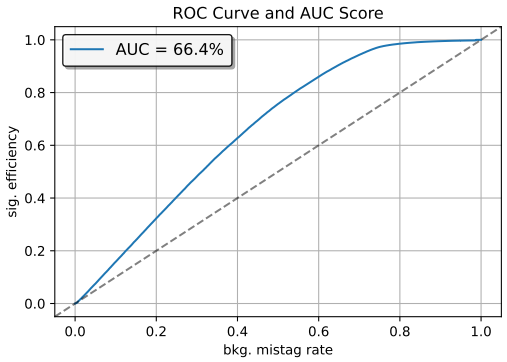

Figure 6.10 AUC for the rotated dataset

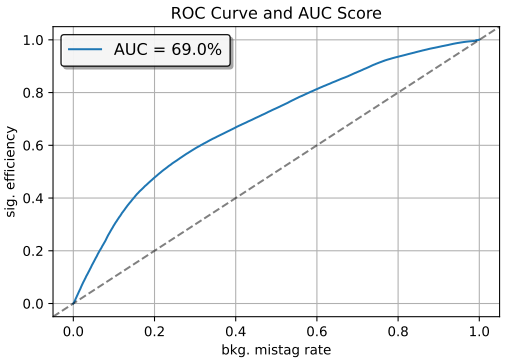

Figure 6.11 AUC, transformed dataset

the ROC curve and the AUC plots are shown.
In all three cases, the MSE loss follows a reasonable evolution, with a minimum among chosen epochs. Given the presence of overfitting (values increasing after the minimum), a small number of training epochs (50) was chosen.
As clearly visible in figure 6.6 and 6.9 where an AUC score of 53% is presented (random classifier), if we train our model using data neither rotated nor transformed, we will not be able to distinguish between background and signal region. The reason

behind this "failure" is the presence among data of jets with rotational symmetries, given the fact that for our purpose the position of the jet in the (η, ϕ) plane is irrelevant. In fact, as discussed in paragraph 4.2, we assume that New Physics relies in the micro-structures of jets. If we do not transform the dataset, during the training phase, our model wastes time and data to learn that these jets are similar, resulting in a learning of only macro-structures of jets and a consequent inability to discriminate. Given the small deviation of anomalies from normal events, it is necessary that our model focuses on details.
As expected, rotated and transformed datasets obtain a higher value of AUC, respectively of 66.4% and 69%. Qualitatively speaking, AS plots show distributions distinguished between background and signal. Overall, we can notice that the transformed dataset performs better than the other; in fact, applying Lorentz boosts and frame change, all jets are pushed forward resulting in a smaller spatial aperture, because the new energy of jets E_0 is lower than original energy. If jets are pushed forward, they cover a smaller space in the (η, ϕ) plan and for our model it is easier to learn to reconstruct background jets.
Anyway, in spite of the fact that we consider results acceptable for out project, it is important to note that perhaps the model we proposed and implemented is not the best choice for this task, given the not so high AUC score: the model can be optimized even more or we can choose another one.

Chapter 7

Conclusions

This thesis has been an opportunity to investigate a new technique used in the search for New Physics.
The main section of this project was about our implementation of Anomaly Detection with a specific dataset and its transformations. The most interesting aspect is, for sure, the comparison between different types of dataset, which was presented to try to understand how a physical transformation can impact on performance of a computer learning. In fact, we started the thesis discussing about the "anti-k_T" dataset (the not transformed dataset) and it was crystal clear that, with an AUC score of 53%, our model was acting as a random classifier. This failure was associated to rotational symmetries among data.
We decided to rotate data applying two Lorentz boosts in order to obtain $\eta'_{jet} = 0$ and $\phi'_{jet} = 0$ for each jet. This simple change resulted in a significant improvement in performance: we were able to get a 66.4% in AUC score, which is not perfect, but at least our model is learning to discriminate between background and signal jets.
The third and last attempt was made transforming the dataset moving to a specific frame of reference. Performance is better than the rotated dataset, we got a 69% AUC score, but, again, it is not perfect.
We can conclude that the transformed dataset is the one which gives us the best results, because the reduction of jet surface in the (η, ϕ) plane makes the classification task easier for the model.
In the end, it is true that we were able to discriminate between background and signal jets but the quantitative parameter, the AUC score, underlined that this implementation is not perfect: 69% of AUC in the best case scenario is not ideal if we want investigate small deviations from the Standard Model. This not brilliant result can be caused by not perfectly optimized hyperparameters or by the model itself. In fact, it is plausible that the architecture proposed does not fit very well for this task. Anyway, these results leave enough space for future development of the project.
We hope that one day, helped by Anomaly Detection, we will be able to find the church we are looking for, the New Physics that researchers have been investigating for years.

Bibliography

[1] CERN. The Standard Model. URL: https://home.cern/science/physics/standard-model.

[2] Cloudera. Deep Learning for Anomaly Detection. URL: https://ff12.fastforwardlabs.com/.

[3] ATLAS collaboration. The ATLAS Experiment. URL: https://atlas.cern/.

[4] Communications Education and CERN Outreach Group. LHC the guide FAQ. URL: https://home.cern/resources/brochure/knowledge-sharing/lhc-facts-and-figures.

[5] Numpy. Numpy linalg norm. URL: https://numpy.org/doc/stable/reference/generated/numpy.linalg.norm.html.

[6] PyTorch. MSE Loss function. URL: https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html.

[7] Scikit-learn. AUC. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html.

[8] Scikit-learn. ROC curve. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html.

[9] Scikit-learn. Standard Scaler function. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[10] Cesare Bini. Lezioni di Fisica Nucleare e Subnucleare, 2024.

[11] Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin. Attention Is All You Need, 2023.

[12] Vasilis Belis, Patrick Odagiu, and Thea Klaeboe Aarrestad. Machine learning for anomaly detection in particle physics. *Reviews in Physics*, 2023.

[13] D. Hendrycks and K. Gimpel. Gaussian Error Linear Units (GELUs), 2023.

[14] Stefano Giagu. Deep Learning for physicists: Transformers, 2023.

[15] Guglielmo Frattari. Investigating the nature of dark matter and of the Higgs boson with jets and missing transverse momentum at the LHC, 2021.

[16] Gregor Kasieczka, Benjamin Nachman, David Shih, Oz Amram, Anders Andreassen, Kees Benkendorfer, Blaz Bortolato, Gustaaf Brooijmans, Florencia Canelli, and Jack H Collins. The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics. *Reports on Progress in Physics*, 2021.

[17] T.S. Roy et A.H. Vijay. A robust anomaly finder based on autoencoders, 2020.

[18] Francesca Nuzzo. Sanity Checks for Explanations of Deep Neural Networks Predictions, 2020.

[19] Han, Song. *Efficient Methods and Hardware for Deep Learning*. PhD thesis, Stanford University, Stanford, CA, USA, 2017.

[20] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

[21] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. The anti-ktjet clustering algorithm. *Journal of High Energy Physics*, 2008(04), 2008.

[22] Oliver Sim Brüning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. CERN, 2004.

Acknowledgments

Un sentito ringraziamento va ai miei relatori, il prof. Stefano Giagu e la dott.ssa Graziella Russo, per avermi guidato nella realizzazione di questo lavoro. Vorrei inoltre ringraziarli per avermi accolto in *saletta* e fatto conoscere questa bella e stimolante realtà.