UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE FÍSICA

Deep Neural Networks applications in experimental physics data analyses

João Pedro da Palma Martins Pires

Mestrado em Engenharia Física

Dissertação orientada por:

Professora Doutora Helena de Fátima Nunes Casimiro dos Santos

2023

Acknowledgements

This work, as most of other related works, was a work progress based on a lot of analyses, sometimes leading to some regressions. Therefore a few times was necessary to do the very same analyses that either showed strange results, or either were a consequence of an error source. During this process an understanding of the results was necessary and for that a special thanks is dedicated to professor and investigator Helena Santos which was the difference between months and days of work.

Sometimes problems related to the code and questions related to its functioning arose and the CERN community had the kindness of helping me and thus a special thank for them too.

Another big thanks for all the friends and family that not only supported me on the construction of this paper, but helped me to desanuviate by get me out of home, for this a sepecial thanks for Cátia Principe, Gonçalo Silva, Margarida Sacadura, Nádia Chaves, Pedro Nobre and Rute Curto.

Abstract

One of the head investigations at the LHC and ATLAS Experiment is related to the study of the Quark Gluon Plasma (QGP), formed in ultra-relativistic heavy-ion collisions. Due to its short lifetime and spatial limitation, the QGP properties are impossible to be measured directly, and therefore, indirect methods must be used. One method is related to the measured quantities of the collimated sprays of particles, so-called jets, generated immediately after the collision takes place. The particle shower develops in the QGP, so understanding the jet energy loss processes and the modification of the fragmentation functions is crucial to infer the properties of this state of matter. In particular, jets resulting from the bottom (b) quark fragmentation are expected to interact with the QGP differently from the other jets providing additional information about the nature of the QGP so identifying them is a must. Distinguishing bottom-jets from charm- and light-jets produced in proton-proton collisions is difficult, but the huge environment of Pb+Pb collisions makes the task particularly challenging. In the ATLAS Experiment the Neural Networks, such as the DL1 - Deep Learning 1, are the most promising tools to provide an efficient jet flavour discrimination and b-tagging.

Keywords: ATLAS Experiment, b-tagging, DL1 - Deep Learning 1, DIPS - Deep Impact Parameter Sets, Pb+Pb collisions.

Resumo Alargado

As redes neuronais (NNs) são amplamente utilizadas nas mais diversas áreas para fazer previsões e estudos, tendo por base um conjunto de dados que à partida parece aleatório. Estas previsões são úteis em diversas áreas, tal como a economia e finanças, em que se tenta, por exemplo, construir modelos para identificação de fraudes, e nas grandes indústrias, como a Amazon e a Google, em que se tentam fazer estudos relativamente aos itens preferenciais para os consumidores na esperança de se prever qual o produto mais desejado. Assim na área da investigação, tenta-se também aplicar NNs para retirar dados de interesse de uma dada amostra.

Atualmente, no *Large Hadron Colider* (LHC) no CERN existem diversas investigações científicas na área da física de partículas que fazem uso de redes neuronais, mas uma das principais investigações está relacionada com o estudo do chamado Plasma de Quarks e Gluões (QGP) nas colisões de iões pesados, que têm lugar na Experiência ATLAS. Este plasma é formado nas estrelas de neutrões e em colisões altamente energéticas entre nucleões (protões e neutrões), no entanto, enquanto que no primeiro o QGP tem uma densidade incrivelmente elevada e uma baixa temperatura, no segundo não só tem uma grande densidade (estima-se pelo menos um fator de 10 superior à densidade nuclear) como também apresenta uma elevada temperatura, propriedades semelhantes ao que se pensa ter havido durante os primeiros momentos após o big bang, e como tal uma janela para o estudo das interações que poderão ter existido entre partões (quarks e gluões). Relativamente a estas partículas, diversas questões surgem associadas à sua interação, entre as quais, por exemplo, como é que os chuveiros de partões perdem a sua energia e torna-se crucial puder dar resposta a muitas destas questões. No entanto, devido ao muito reduzido tempo de vida e volume do QGP, torna-se impossível de medir diretamente as propriedades relativas a este estado da matéria e interação entre os partões, assim torna-se crucial a busca por um método de estudo indireto.

Desta forma surge o método *Hard Scattering Probing* que faz uso de *sprays* de colimados partículas, ao qual se dá o nome de jatos de partículas, gerados através da fragmentação dos diversos quarks que se formam na consequência da interação com o QGP. Através da medição das propriedades associadas a estes jatos, é possível reconstruir as interações que ocorream durante o QGP. No entanto, dependendo do tipo de jato que é normalmente associado ao quark pai que o originou, devido ao diferente comportamento dos quarks, mostra-se que jatos de partículas formados pela fragmentação do quark *bottom* interagem menos com as partículas provenientes da chuva de partículas geradas e são os primeiros a ser gerados, imediatamente após a colisão, o que faz com que haja uma maior probabilidade das propriedades medidas no detetor estarem relacionadas com as do plasma.

Assim, não só é necessário obter as propriedades dos jatos, como também é necessário identificar jatos de parículas resultantes da fragmentação do quark *bottom* (b-jets) dos restantes gerados da fragmentação de outros quarks.

Ao longo dos anos, na Experiência ATLAS, foram sendo desenvolvidos diversos algoritmos com o objetivo de detetar jatos resultantes da fragmentação do quark *bottom*, os chamados algoritmos de *b-tagging*. Estes algoritmos baseavam-se nos mais diversos parâmetros medidos diretamente do detetor, sejam estes os parâmetros de impacto, o tempo médio de vida da partícula, ou a distância média de decaimento associado para construir algum tipo de variável discriminante capaz de diferenciar os jatos de partículas formados. Assim, de acordo com os parâmetros de estudo, foram surgindo diversos algoritmos como o *Impact Parameter 2D* (IP2D) e o *Impact Parameter 3D* (IP3D), que fazem uso dos parâmetros de impacto, e *Secondary Vertex* (SV1) e *JetFitter* que fazem uso das distâncias de decaimento.

Pelo facto destas ferramentas utilizarem diferentes parâmetros apresentavam também diferentes eficiências e alcances que levavam a uma penalização na identificação de b-jatos numa perspetiva individualista. Desta forma começaram a surgir os primeiros algoritmos baseados em redes neuronais, os algoritmos de alto-nível, que inicialmente apenas correlacionavam as variáveis discriminantes construídas pelos algoritmos de baixo-nível, de uma forma bastante simplista, mas à medida que se foi aprofundando o desenvolvimento das redes neuronais, começaram a surgir diferentes NNs que, fazendo uso de diferentes métodos, permitiram um cada vez melhor desempenho no que toca ao *b-tagging*.

Para colisões entre protões os algoritmos de alto-nível foram sendo cada vez mais aplicados e estudados, apresentando resultados bastante prometedores no que toca ao objetivo final que é a discriminação do jato originado pelo quark b, no entanto fica a faltar dar o grande passo que correponde a estudar o comportamento destes algoritmos quando submetidos a um ambiente bastante mais carregado, em partículas, como é o caso das colisões entre iões pesados. Neste tipo de colisões existe um maior número de nucleões envolvidos o que leva à criação de um QGP com um maior tempo de vida e volume. Desta forma os partões no QGP vão ter um maior número de interações que poderá ser reconstruído através dos jatos de partículas e possibilitará melhores conclusões. No entanto, em contrapartida, também haverá uma maior corrupção dos dados associada a particulas geradas antes da colisão (*underlying event*).

Desta forma, para se puder analisar o comportamento dos algoritmos em iões pesados, este trabalho compromete-se a abordar dois tipos de algoritmos baseados em redes neuronais que fazem uso de diferentes propriedades e características associadas aos jatos de partículas, sejam estes algoritmos o *Deep Learning 1* (DL1) e o *Deep Impact Parameter Sets* (DIPS). Tanto o DL1 como o DIPS são algoritmos que fazem uso de redes neuronais e constroem três probabilidades referentes ao jato resultar da fragmentação de um jato *bottom*, *charm* ou de sabores leves (b-, c- e u-jatos), mas o DL1 é considerado um algoritmo de alto-nível, fazendo uso de variáveis construídas por algoritmos de baixo-nível, onde o DIPS está incluído. Enquanto o DL1 faz uso das propriedades dos jatos, isto é, das propriedades associadas às partículas envolventes de um feixe colimado de partículas, o DIPS consegue trabalhar as características associadas a cada partícula individualmente. Deste modo, são esperadas uma complementaridade e uma dependência entre os dois. A grande mais valia de se usar o DIPS é que este permite uma comparação com o DL1 e permite retirar conclusões relativamente ao impacto do valor mínimo do momento transverso (p_T) nas partículas escolhido.

Ao longo desta dissertação, para além da observação dos resultados associados a colisões Pb+Pb e da comparação com os resultados em pp, também vai ser feita uma breve explicação do pré-processamento dos dados, onde as partículas resultantes de colisões pouco energéticas vão ser eliminadas e vai ser aplicado o chamado downsampling para retirar a dependência das propriedades cinemáticas (p_T e η) do treino da rede neuronal. Pois no caso de não ser feito o downsampling os resultados seriam dependentes do número de jatos gerados, que muitas se torna impossível devido à probabilidade de acontecimento associada ao evento ser reduzida.

Uma vez esclarecido o pré-processamento aplicado, é feita a análise detalhada referente à correlação entre as propriedades utilizadas durante o treino de ambos os algoritmos. Estas correlações vão ter um impacto direto no treino das redes neuronais, pois estas vão tirar vantagem das dependências de cada uma das variáveis em questão beneficiando ou prejudicando o desempenho dos algoritmos no que toca à identificação dos b-jets.

Ainda como forma de identificar o impacto de cada uma das variáveis e/ou propriedades dos jatos de partículas no treino, utilizando a definição de discriminante de b-jatos utilizado amplamente em estudos associados a colisões pp, são construídos três novos discriminantes para identificarem o grau de fiabilidade na identificação de b-, c- e u-jatos. Ao fazer-se uso destes novos discriminantes é possível ver os gradientes dos discriminantes em ordem às propriedades e observar quais as propriedades que maior impactam positivamente e negativamente o desempenho da rede neuronal no que toca à identificação de b-jatos. Relativamente a esta questão é esperada também uma correlação entre os diversos gradientes e o desempenho dos algoritmos, pois como se pode imaginar uma correta identificação dos c-jatos e dos u-jatos vai provocar uma melhor identificação dos b-jatos.

Adicionalmente, da diferença associada à amostra de pp e de Pb+Pb é feito um outro estudo, tanto para DL1 como para DIPS, associado à centralidade das colisões, que devido ao número de nucleões envolvidos é absurdo ser tratado em colisões pp, mas que em Pb+Pb tem todo o sentido ser abordado.

Keywords: Experiência ATLAS, b-tagging, DL1 - Deep Learning 1, DIPS - Deep Impact Parameter Sets, Pb+Pb collisions.

Contents

List of Figures

List of Tables

Glossary

Δr Track Δr. 34, 35

d_0 Transverse impact parameter, distance of closest approach of the track to the primary vertex point in the r-ϕ projection. 34

IP2D/IP3D$_{bu}$ IP2D/IP3D Log-likelihood ratio between the probability of a jet be a b-jet and the probability of a jet be a u-jet. The u-jet probability is in the denominator.. 11

IP2D/IP3D$_{bc}$ IP2D/IP3D Log-likelihood ratio between the probability of a jet be a b-jet and the probability of a jet be a c-jet. The c-jet probability is in the denominator.. 11

IP2D/IP3D$_{cu}$ IP2D/IP3D Log-likelihood ratio between the probability of a jet be a c-jet and the probability of a jet be a u-jet. The u-jet probability is in the denominator.. 11

JF $\Delta R(\mathbf{p}_{jet}, \mathbf{p}_{Vtx})(JF)$ Delta R between the jet axis and the vectorial sum of momenta of all tracks associated to the displaced vertex. 14

JF E_{frac} Fraction of the charged jet energy in the secondary vertices. 14

JF $f_E\left(2^{nd}/3^{rd}vtx\right)$ Fraction of charged jet energy in 2^{nd} or 3^{rd} vertex. 16

JF $E_{Trk}\left(2^{nd}/3^{rd}vtx\right)$ Energy fraction of the tracks associated with the 2^{nd} or 3^{rd} vertex. 16

JF $L_{xy}\left(2^{nd}/3^{rd}vtx\right)(JF)$ Transverse displacement of 2^{nd} or 3^{rd} vertex. 16

JF $L_{xyz}\left(2^{nd}/3^{rd}vtx\right)$ Distance of 2^{nd} or 3^{rd} vertex from the primary vertex. 16

JF mass Invariant mass of the tracks fitted to the vertices with at least two tracks. 14

JF $m_{Trk}\left(2^{nd}/3^{rd}vtx\right)$ Invariant mass of tracks associated with the 2^{nd} or 3^{rd} vertex. 16

JF N_{2Tpair} Number of two-track vertex candidates (prior to decay chain fit). 14

JF $N_{TrackAtVtx}$ Number of tracks from multi-prong displaced vertices. 14

JF $N_{TrkAtVtx}\left(2^{nd}/3^{rd}vtx\right)$ Number of tracks associated with 2^{nd} or 3^{rd} vertex. 16

JF $N_{SingleTracks}$ Number of single track vertices. 14

JF N_{Vtx} Number of vertices with more than one track. 14

JF S_{xyz} Significance of the average distance between PV and displaced vertices, considering all multi-prong vertices or (if there are none) of all single-track vertices. 14

JF Y_{trk}^{min}, Y_{trk}^{max}, Y_{trk}^{avg} $\left(2^{nd}/3^{rd}vtx\right)$ Minimum, maximum and average track rapidity of tracks at 2^{nd} or 3^{rd} vertex. 16

JF+SV η_{trk}^{max} Maximum track relative η. 35

JF+SV η_{trk}^{min} Minimum track relative η. 35

JF+SV η_{trk}^{aver} Average track relative η. 35

JF+SV η_{jet}^{max} Maximum jet relative η. 35

JF+SV η_{jet}^{min} Minimum jet relative η. 35

JF+SV η_{jet}^{aver} Average jet relative η. 35

JF+SV E Energy of charged tracks associated to secondary vertex. 35

JF+SV E_{frac} Fraction of charged jet energy in secondary vertex. 35

JF+SV L_{xy} Transverse displacement of the secondary vertex from primary vertex. 35

JF+SV L_{xyz} Distance of the secondary vertex from primary vertex. 35

JF+SV mass Invariant mass of tracks associated to secondary vertex. 35

JF+SV N_{Tracks} Number of tracks associated to secondary vertex. 35

N_{Hits} Combined number of hits in the pixel layers (including the IBL). 34

$N_{InnerHits}$ Number of hits in the IBL. 34

$N_{NextToInnerHits}$ Number of hits in the next-to-innermost pixel layer. 34

$N_{SCTHits}$ Combined number of hits in the SCT layers (since 2 strip hits are required for a full SCT spacepoint, this number is divided by two in the track selection). 34

$N_{SharedHits}$ Number of shared hits (contributing to the track fit and to another track plus the ones not marked as split hit) in the pixel layers (including the IBL). 34

$N_{SharedInnerHits}$ Number of shared hits (contributing to the track fit and to another track) in the IBL. 34

$N_{SharedSCTHits}$ Number of shared hits (contributing to the track fit and to another track) in the SCT layers (since 2 strip hits are required for a full SCT spacepoint, this number is divided by two in the track selection). 34

$N_{SplitHits}$ Number of split hits in the pixel layers (including the IBL; split hit = hit is identified as being created by multiple charged particles during ambiguity solver stage at pattern recognition level). 34

$N_{SplitInnerHits}$ Number of split hits in the IBL. 34

$p_{T_{frac}}$ Fraction of the jet pt carried by the track. 34

SV1 $\Delta R(\mathbf{p}_{jet}, \mathbf{p}_{Vtx})$ ΔR between the jet axis and the direction of the secondary vertex relative to the primary one. 13

SV1 E_{frac} Energy fraction of tracks associated with the secondary vertex. 13

SV1 L_{xy} Primary and secondary vertex transverse distance. 13

SV1 L_{xyz} Primary and secondary vertex distance. 13

SV1 mass Invariant mass of tracks at the secondary vertex, assuming pion mass. 13

SV1 N_{2Tpair} Number of two-track vertex candidates. 13

SV1 N_{Track} Number of tracks used in the secondary vertex. 13

SV1 S_{xyz} Primary and secondary vertex distance weighted by its uncertainty. 13

S_{d_0} **IP3D** Signed transverse impact parameter significance from IP3D algorithm. 34

S_{z_0} **IP3D** Signed longitudinal impact parameter significance from IP3D algorithm. 34

$z_0 \sin\theta$ Longitudinal impact parameter projected onto the direction perpendicular to the track. 34

Chapter 1

Introduction

This chapter presents the motivation, objectives, general methodology, and contributions of this dissertation, as well as the overall document structure.

1.1 Motivation

Currently, at the ATLAS Experiment, one of the head investigations is related to the study of the so called Quark Gluon Plasma (QGP), formed in ultra-relativistic nucleon-nucleon collisions. This plasma is interesting because it is very similar to what is expected to have existed at the first moments of the universe, and thus, it is a window to study the interactions between particles immediately after the Big Bang. However, due to its own lifetime and spatial limitation, the QGP properties are impossible to be measured directly, and therefore an indirect method must be used. There are quite a few interesting methods that could be used for this objective, but one of the most promising is the Hard Scattering Probing method related to the measurement of the particle jets properties formed through the quark fragmentation. These jets start to be formed immediately after the collision takes place and interact with the QGP constituents. However, depending on the jet's quark parent, different physic behaviors are expected and some jets will be more relevant than others. In fact, the jets resulting on the bottom quark fragmentation, from now on referenced as b-jets, are excellent sources of information.

A flavor jet identification method is needed to discriminate b-jets from the others, in highly energetic heavy ion collisions. Such an identification is challenging due to the huge amount of jets that are generated as a consequence of the large number of binary nucleon-nucleon collisions. Furthermore, another problem appears on heavy ions where a much meaningful number of soft particles are generated. These particles, characterized by their low transverse momentum, define the so called underlying event and are generated by interactions between quarks and gluons emanated from the collision itself. Usually the effect of such particles is so intense and inflict such a big data pollution that the total soft particle energies need to be statically studied and taken into account.

The identification of b-jets, is, currently, one of the most relevant challenges that influence much of

the analyses results and conclusions. This identification is crucial in many physic areas in the ATLAS and the CMS experiments at the Large Hadron Collider (LHC), and is directly influenced by the detection efficiency of b-jets over a large background composed by many other jets, originate in charm quark hadrons but no bottom quark hadrons (c-jets), or neither bottom nor charm hadrons (light-flavor jets also called u-jets). This identification is possible due to b-tagging algorithms that take advantage of the impact parameters directly related to the long lifetime ($\tau \approx 1.5\,\mathrm{ps}$, $c\tau \approx 450\,\mu\mathrm{m}$), high mass and high decay multiplicity of b-hadrons, as well as taking into account the b-quark fragmentation properties.

The main problem related to the previous algorithms reside on their efficiency and range. Once each algorithm is based on different parameters, they have distinct behaviors that somehow complement themselves. Therefore, the idea of constructing one algorithm capable of correlating all the previous ones started to be elaborated, and thanks to the developments of the neural networks (NN) it was possible to be implemented. In fact, only in ATLAS Run 2 the objective of increase the b-tagging algorithm performances, instead of simply use the above algorithms further enhanced and now called low-level b-tagging algorithms, the idea of complementation between these algorithms started to grow and the first multi-variable algorithms, the high-level b-tagging algorithms, making use of several variables outputted by each low-level algorithm began to be developed. Soon the ATLAS collaboration understood the high potential of such models, once these models provide improvements in the u- jet and c-jet flavor rejections.

Performance evaluations demonstrate that the probability of identifying a c-jet (ε_c) and a light-jet flavor (ε_u) with a b-jet tagging probability, also called as b-jet efficiency (ε_b), of $70\,\%$ reach 2.5 and 10 times more rejection rate on high-level algorithms, respectively for c-flavor and u-flavor jets, when compared to low-level algorithms [Aad, 2019].

1.2 Objectives

Currently, there are two different high-level b-tagging algorithms, the MV2 [Collaboration, 2017b] and DL1, that offer different analysis types sustained by their intrinsically different constructions and quite different algorithm approach perspective. While the MV2 is based in a boosted decision tree (BDT) algorithm, the DL1 is based on a deep-forward neural network (NN) trained using Keras with the Theamo backend and the Adam optimizer [Aad, 2019], both making use of the low-level algorithms outputs as inputs.

The DL1, due to its NN based algorithm, offers a much more beneficial contribution when it comes to the b-tagging identification process and therefore it is presented as a good tool when it comes to the study of lead-lead collisions (Pb+Pb) analyses.

This work intends to understand the differences between pp and Pb+Pb DL1 performances and gather relevant conclusions regarding the training variables dependence, which are expected to be distinct from pp to Pb+Pb, as well as study the underlying event influence, shown mainly in Pb+Pb collisions.

The DIPS [Collaboration, 2020] low-level algorithm is also studied, with the objective of making the above same conclusions, but now regarding the track features.

1.3 Document Structure

In order to cover all the specified objectives this work is separated in three main topics where the first one is intended to give the reader a certain b-tagging understanding, the second is related to the NN training samples preprocessing and the last one is related to the training results and studies. The overall separation is summarized as below.

- **Chapter 2** (Related Work) introduces the basic concepts on b-tagging algorithms as well as the description of the detector and measured quantities.
- **Chapter 3** (Sampling preprocessing) describes the training samples generation, detailing the simulated data samples for both pp and Pb+Pb collisions and the applied selection cuts, as well as the generation of the samples used in training.
- **Chapter 4** (Neural Network training) details both the NN training features and their training relevance and correlation, as well as the NN training hyperparameters. The pp and Pb+Pb results obtained with both DL1 and DIPS taggers are presented.
- **Chapter 5** (Conclusion) summary of the overall conclusions made.

Chapter 2

Related Work

This chapter presents the basic concepts relative to b-tagging, since the description of the measured quantities, on which the discriminant methods are based, to the construction of the algorithms. Once all quantities are obtained with the ATLAS detector, there is too a brief description of this detector.

2.1 ATLAS Detector

At a particle collision, from the moment immediately after the collision until the moment where a particle reaches the detector, millions of particles can be generated, consequence of particle interactions or unstable states leading to decays, with this effect being far greater for massive particles, being proportional to the particles rest energy. These generated particles can have very much similar properties, ensured by the same origin/parent particles, and thus, it can be difficult to discriminate them.

The ATLAS detector [Collaboration and et al., 2008], at the LHC, plays a main role on the b-jet identification first steps, once much of the variables inputted on low-level b-tagging algorithms, algorithms which don't make use of NNs, are measured quantities obtained by the different ATLAS modules/sub detectors.

The ATLAS detector is a very complex detecting system that is constantly being monitored and updated, but it has several main modules that are very much indispensable when dealing with particle identification. Due to this works nature, there are four main modules that require a better understanding, this ones being

1. The **Inner Detector (ID)** - involving the particle interaction point and greatly contributing for the particle tracking.
2. The **Electromagnetic and Hadronic calorimeters (ECal and HCal)** - immediately after the ID, where the HCal is located after the ECal from the beam transverse view.
3. The **Muon Spectrometer** - The last detector layer where muons interact allowing their energy contabilization, decreasing greatly the energy loss given by the undetected particles such as neutrinos.

Each above component has its own important contribution to the overall performance and detection efficiency and as so, a deeper understanding on these modules may become quite relevant.

2.1.1 Inner Detector (ID)

The inner tracking detector (ID) is one of the main components relative to the particles tracking and by being located near the interaction point, it is the first component with which the generated particles interact.

It is involved by a superconducting solenoid's axial magnetic field of 2 T that enables charged-particle tracking, with its different sub detectors designed to interact as less as possible with the particles, while having the highest segmentation/resolution possible.

The ID is made by high-granularity silicon pixel detectors displaced near the primary vertex region (pseudorapidities[1] $|\eta| < 2.5$), typically allowing four measurements per track.

The first layer, the innermost one usually refereed to as insertable B-layer (IBL) [Aad, 2019], is the layer with the highest resolution and hence, lower pixel size, to provide the best resolution for a particle hit at the smallest detector radius. This decreases the possibility of identifying two different particles as a single one, once the solid angle separation between two different particles will be much less at the innermost layers. In truth this layer is so important that its absence would make the u-jet rejection decreases by a factor of 4 on b-tagging algorithms for a fixed b-tagging efficiency single-cut OP.[Aad, 2019].

Another very important sub detectors are the silicon microchip tracker (SCT), that increases the number of tracks having 8 layers of silicon microstrip arranged in cylinders at the barrel region and 18 layers arranged in a disk at the endcap regions, and the transition radiation tracker (TRT), a straw tube tracker which allows the track reconstruction for bigger radial ranges at $|\eta| < 2$, while providing electron identification information.

2.1.2 Calorimeter

The calorimeter is where particles will deposit most of their energy, and therefore allow the b-tagging algorithm variables energy related measurement. It is divided into two main sub calorimeters intrinsically different, these being the electromagnetic and the hadronic calorimeters, and covers in total a $|\eta| < 4.9$ region.

The electromagnetic calorimeter (ECal) reach a $|\eta| < 3.2$ region and is composed by a main tile barrel (EB) and two endcap (EECal) high-granularity lead/liquid-argon (LAr) sampling calorimeters,

[1] ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the z-axis along the beam pipe. The x-axis points from the IP to the centre of the LHC ring, and the y-axis points upwards. Cylindrical coordinates (r, ϕ) are used in the transverse plane, ϕ being the azimuthal angle around the z-axis. The pseudorapidity is defined in terms of the polar angle θ as $\eta = -\ln[\tan(\theta/2)]$. Angular distance is measured in units of $\Delta R \equiv \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$.

further complemented with a thin LAr presampler ($|\eta| < 1.8$) to account for energy losses inside the calorimeter first layers.

The hadronic calorimeter (HCal) splits into 4 different modules, a steel/scintillator tile sampling calorimeter (HB), that in turn splits into 3 barrels at $|\eta| < 1.7$, two copper/LAr hadronic endcap (HCal) and a cooper/LAr and tungsten/LAr forward calorimeter, inserted for electromagnetic and hadronic measurements optimization. In total the hadronic calorimeters covers almost completely the entire solid angle.

2.1.3 Muon Spectrometer

The muon spectrometer is the last sub detector on which particles went through and incorporates 3 large superconducting magnets, one on the barrel and two on each endcap region, with a separate trigger and high-resolution tracking.

The combination between the bending power of the magnets, ranging from $1\,\mathrm{Tm}$ to $7.5\,\mathrm{Tm}$, with the isolated muon tracking and trigger systems, made by resistive-plate chambers in the barrel as well as thin-gap chambers in the endcap regions, provides a precise reconstruction of the muon ($\mu^{\pm}$) momentum, as well as a good muon deflection tracking on the $|\eta| < 2.7$ region. The tracking provides a good muon deflection tracking with 3 tracking layers, each one with drift tubes complemented by the presence of cathode-strips in the forward region.

2.2 Measured quantities

The b-jet identification process starts after the collision take place, where all particles are obtained by Monte Carlo generated particles. In order to identify the possible b-jets formed immediately after the interaction point, or primary vertex, originated from the b-hadron fragmentation, as well as other particles besides jets, a data reconstruction process must first be done where much of the discriminant properties will be collected and filtered. Here the advantage is to know the particles trajectory, given by the ID detector, and the particle's energy deposition, measured by the calorimeter.

For particle tracking, the objective is, at first instance, look for segments measured at the ID, which are defined as a set of pixels that had their energy changed due to a particle interaction. Such information, in addition to the calorimeter energy depositions characterized by the presence of clusters analogous to the ID segments, are then used for particle trajectories identification, playing a vital role when trying to find possible vertices indicative of a decay process and formed by two or more tracks, where each track is reconstructed from signal segments originated on the ID, commonly named as "hits" [Collaboration, 2016].

As tracks are proportional to the amount of generated particles, a dimensional problem arise, resulting in memory pressure problems. However, as much of the tracks are generated from the same primordial particle decay, much of the tracks maintain similar properties that could be statistically analyzed and

treated as a very same set, with collimated particles and called as particle jets. Thus, in order to construct a jet, each track is correlated by the expression of ΔR (equation 2.1), which define the solid angle separation between tracks and is used to identify possible jet's tracks.

This ΔR quantity is also related with the jet p_T, due to the decay process be more collimated, and therefore the highest the jet's p_T, the narrower the solid angle, for example whilst for $20\,\mathrm{GeV}$ p_T jets common values round the $\Delta R = 0.45$, while for $150\,\mathrm{GeV}$ p_T $\Delta R = 0.26$ [Collaboration, 2016]. Thus, the necessary cuts on the ΔR variable are jet p_T dependent.

$$\Delta R(track, jet) \equiv \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{2.1}$$

By making use of the ΔR variable, all identified tracks are associated to one jet or to none. However it's possible to have a track common to more than one jet, on this occasions the criteria is to relate the track to the jet with the smallest ΔR, and so guarantee that only one jet can be related to a track. It's by applying further discriminant methods to the tracks, present on the b-tagging algorithms, that b-tagging identification is possible.

2.3 *b*-tagging algorithms

With all tracks associated to a jet, tracks are submitted to several additional requirements that are made to discard possible non interesting tracks, related to tracks formed by a single charged particle, long-lived particle tracks (K_s, Λ, and other hyperon decays) and material interactions from well measured tracks, with this last being relevant for analysis. These requirements are restricted to the particle properties and can be more or less restrictive, depending on which analysis type one wants to make. For example, the b-tagging baseline quality level requires a track's p_T bigger than $1\,\mathrm{GeV}$ with at least 7 pixel or micro-strip hits on the ID detector, with at least 2 of those hits on the pixel layer and with on eof them on the innermost layer [Collaboration, 2016].

Once selected the interesting jets it's possible to apply even more specific requirements, these given by algorithms that, due to their purpose, are named as b-tagging algorithms. There are several b-tagging algorithms that were developed during ATLAS Run 1 that are still used at ATLAS Run 2. Some of these were left behind, due to systematic errors, whilst others were tuned to enhance the b-tagging performances, but being basically the same as on Run 1.

The b-tagging algorithms tend to be, also, classified as lifetime based algorithms [Collaboration, 2016], due to these algorithms be capable of exploiting the relatively long lifetimes of b-hadrons, hadrons resulting from the b-quark fragmentation, while compared to other detected particles, usually called background. For example, for a b-hadron particle with a transverse momentum (p_T) rounding $50\,\mathrm{GeV}$, the corresponding mean flight path of such a particle, before decaying, is about $3\,\mathrm{mm}$ at the transverse plane, inside the detector, ensuring that at least one vertex would be displaced from the interaction point.

Lifetime based algorithms are divided into two main categories, depending if they either base their

study on the b-hadron tracks large impact parameters characteristic from b-hadron decays, or they reconstruct the displaced vertices.

Latter on, during ATLAS run 2, two b-tagging algorithms were implemented using the results of the other algorithms as inputs for multi-variable classifiers, increasing the b-tagging performance, and thus providing better results. As a consequence, a new algorithm classification was created and the base algorithms were classified as low-level algorithms and the ones making use of base algorithms were classified as high-level algorithms.

This type of algorithms are influenced by errors related to the primary vertex displacement, taken into account on impact parameters, which can be greatly mitigated by increasing the track multiplicity associated to each vertex. This is specially important for the second algorithm class, section 2.4.1.1, for multi vertex displaced based algorithms, where the error associated to the primary vertex used in each instance is propagated to the next one and is observed that significantly better resolutions can be achieved on events with high p_T jets or leptons. On this type of algorithms a selection criteria must be further applied to select the vertex from a set of reconstructed vertices, by taking the vertice with the maximum associated tracks' p_T^2 sum. Studies show that the probability of identifying the primary vertex on $t\bar{t}$ events rounds $98\,\%$, while lower multiplicity final states are considerably lower [Collaboration, 2016].

Through out this section two different low-level algorithm approaches will be presented, Impact parameters at 2.3.1.1 and vertex displaced based algorithms at 2.3.1.2 where quite a few low-level algorithms examples are shown, as well as two High-level algorithm taggers at2.3.2.

2.3.1 Low-level b-tagging algorithms

2.3.1.1 Impact parameters based algorithms

Due to the b-hadrons long lifetime, high mass and high decay multiplicity, it is possible to develop an algorithm that makes use of the so called impact parameters, that are directly related to the hadron path made inside the detector. Once the b-hadron average lifetime rounds the $\tau = 1.5\,\mathrm{ps}$, the average path made by a b-hadron before decay is given by $\langle l \rangle = \beta\gamma c\tau$ and therefore it's possible to define two different impact parameter variables: the transverse impact parameter d_0 and the longitudinal impact parameter z_0, one per plane.

These d_0 and z_0 variables are proportional to the particle's flight path and are the distance of the closest approach of a track to the primary vertex point on the plane transverse to the beam line ($r - \phi$ plane) and on the the plane were the beam line is contained (z plane), respectively, being either positive or negative depending on the track direction over the primary vertex disposition.

Therefore the sign can be positive if the track segment intersects the jet axis after the primary vertex, consequence of a hadron decay, or negative otherwise (shown in figure 2.1), further increasing the discrimination power, since b- and c-hadron decays tend to have a positive sign, as well as long-lived particles, but a reasonable prompt background tend to have a negative sign. This jet's axis are obtained

on the calorimeter and correspond to the average cluster disposition direction, however if a secondary vertex is found this axis is defined to be the line which intersects the two vertex. Thus, the identification of whether the tracks are formed by b-hadron particle decays or background is possible by limiting the accepted track's impact parameters value. Nonetheless, instead of simply using z_0, $|z_0| \sin\theta$ is broadly used, since the z_0 variable has a big dependence with the polar angle θ and the $\sin\theta$ is capable of decreasing this dependency. This criteria provides a relatively good amount of b-tagging quality tracks associated to the jets, rounding 3.5 and 7 tracks, whether working with a $50\,\mathrm{GeV}$ or $200\,\mathrm{GeV}$ jet's p_T respectively [Collaboration, 2016].

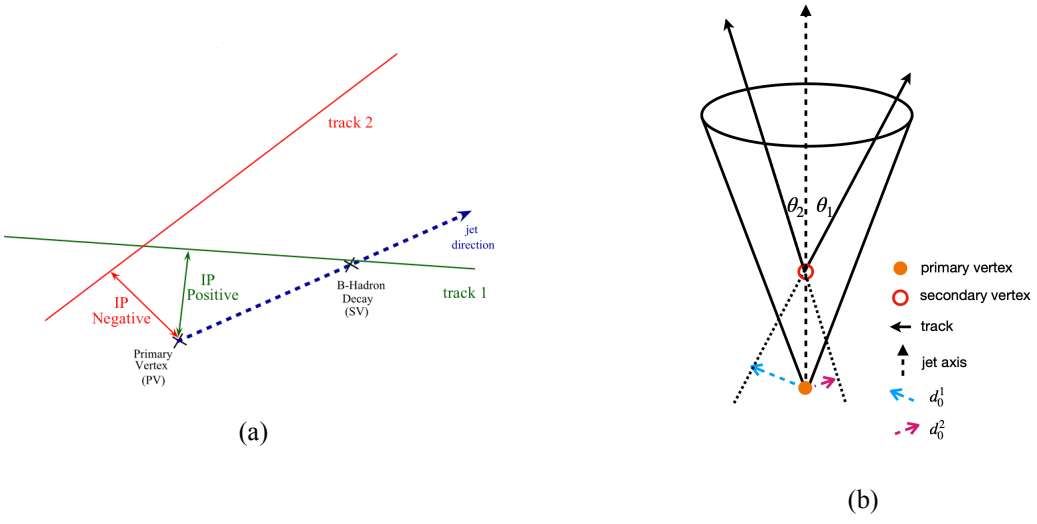


Figure 2.1: (a) Impact parameter signal, related to the track and jet.[Obikhod and Petrenko, 2020] (b) d_0 track's impact parameter.[Collaboration, 2022]

There are two complementary impact parameters based algorithms used at current state: the Impact Parameters 2D (IP2D) and the Impact Parameters 3D (IP3D), both being based on the JetProb algorithm, one of the first developed b-tagging algorithms, but also using the impact parameters significance representative of the impact parameters statistical variation. The main difference between these two algorithms is that, while the former algorithm constructs a discriminating variable based on the transverse impact parameter significance signed by the impact parameter signal, most likely JetProb, the latter one, goes even further, and, by taking into count additionally the longitudinal signed impact parameter significance, constructs a two dimensional grid were a correlated discriminating variable is defined.

The JetProb algorithm uses the transverse impact parameter, more specifically the transverse impact parameter significance given by $S_{d_0} \equiv d_0/\sigma_{d_0}$, with σ_{d_0} representing the reconstructed d_0 uncertainty. The code is based on the S_{d_0} jet's tracks discriminant power that is combined with a pre-determined resolution function $R\left(S_{d_0}\right)$ defined with prompt tracks, tracks generated before the collision due to partons

interactions and mostly associated to negative impact parameters, which allows the calculation of the probability of a jet be a b-jet. The calculations for the b-jet probability are related to the probability of each jet's track be produced from the primary vertex and not from prompt decays, however, to be able to make such distinction a discriminant variable is needed and a function $R\left(S_{d_0}\right)$ is defined. This cumulative density function is defined previously and takes into count the experimental data prompt tracks distribution characterized with negative impact parameters, assuming negligible heavy-flavor particles contribution. Only then, this probability density function (PDF), represented on equation 2.2, will allow the track's primary vertex generation probability determination that is then used to get the overall b-jet probability, equation 2.3. This data treatment provides a uniform P_{jet} for u-jets and a peak P_{jet} distribution around zero for b-jets, nevertheless this distinction fails if the data collected has a reasonable quantity of long-lived particle decays and material interactions, which is by far the most common.

$$P_{trk,i} = \int_{-\inf}^{-|S^i_{d_0}|} R(x)\, dx \tag{2.2}$$

$$P_{jet} = P_0 \sum_{j=0}^{N-1} \frac{(-\ln P_0)^j}{j!} \qquad P_0 = \prod_{i=1}^{N} P_{trk,i} \tag{2.3}$$

It was latter on, with the IP2D algorithm, that a more reasonable u-, b- and c-jet discrimination was attainable by discarding the $R(S_{d_0})$ function and by taking the PDF reference histograms from the Monte Carlo (MC) simulations instead. With the MC simulations it was possible to simulate all flavor-jet types for each impact parameter with a reasonable small uncertainty, which could be even further mitigated by the PDF classification based on the pattern hit of the tracks. Furthermore, this PDFs enabled the calculation of the per-track probability ratios of each flavor-jet, arising the possibility of defining the Log-Likelihood Ratio variable (LLR), of great importance on high-level b-tagging algorithms and defined as the sum over the per-track probability ratios for each jet-flavor hypothesis represented on equations 2.4, 2.5 and 2.6 to separate b- from u-jets (IP2D/IP3D$_{bu}$), b- from c-jets (IP2D/IP3D$_{bc}$) and c- from u-jets (IP2D/IP3D$_{cu}$), where N is the total number of jet's tracks and p_i, with $i =$ u, b and c , the MC PDF for i-jet flavor hypothesis.

$$\log(P_b/P_u) = \sum_{i=1}^{N} \log\left(\frac{p_b}{p_u}\right) \tag{2.4}$$

$$\log(P_b/P_c) = \sum_{i=1}^{N} \log\left(\frac{p_b}{p_c}\right) \tag{2.5}$$

$$\log(P_c/P_u) = \sum_{i=1}^{N} \log\left(\frac{p_c}{p_u}\right) \tag{2.6}$$

The IP3D in addition to the d_0 IP2D treatment, takes advantage of the z_0 impact parameter and of the correlation between them, comparing a $S \equiv (d_0/\sigma_{d_0},\ z_0/\sigma_{z_0})$ with a bi-dimensional PDF MC simulation, which defines the track-weight with the LLR variable aid.

2.3.1.2 Vertex displaced based algorithms

Due to b- and u-jets similarities, it can be quite difficult to discriminate such jets with just the impact parameters' jets signatures. To give answer to this problems, some additional algorithms were developed based on an inclusive three dimensional approach of the decay products originated from b-hadrons.

Related to the vertex displaced based algorithms, there are two different types, categorized as secondary vertex algorithms, once they can only obtain the secondary vertex and examples are the SV0 and SV1 algorithm, and as multi-vertex displaced algorithms, the case of the JetFitter.The main difference between the two types is that, while secondary vertex tagging algorithms make use of the tracks of the ID detector to identify the secondary vertex, the multi-vertex algorithms use the topological structure of weak b- and c-hadron decays inside a jet to reconstruct the complete b-hadron decay.

All the secondary vertex algorithms first start from selecting all tracks displaced by a certain quantity[2] from the primary vertex associated to a jet, and consequently, define possible vertex candidates formed by pairs of tracks with a correspondent vertex fit of $\chi^2 < 4.5$. From this group of vertices some will be generated from long-lived particles and material interactions and must be identified as such, thus an invariant mass discriminant and $r - \phi$ plane vertex position criteria is applied.

By using the invariant mass of the charged-particles' track four-momenta, it is most likely to separate long-lived particle decays and photon conversions, and by comparing the projection of the vertex's position in $r - \phi$ plane with the first hits inside the innermost pixel detector layers, it's possible to reject secondary vertices formed as a consequence of material interaction. Additionally a looser track selection is implemented, maximizing the "fake" track detection by implementing more variable restrictions and at least one hit on the pixel layer detector. However, the criteria implemented is algorithm dependent and must be changed depending on the utility.

With the possible b-hadron track-pairs identified, the requirements to start analyzing data are complete and the construction of the vertex can be done by combining the two-track vertices into one vertex and interactively removing the track with the largest χ^2 contribution to the vertex fit, this until a certain threshold be achieved.

Similarly to the JetProb algorithm, described in section 2.3.1.1, the SV0 algorithm was the first secondary vertex algorithm of its type and it was from it that all vertex-based codes started to flourish, and thus an explanation regarding this code may prove beneficial.

The SV0 algorithm, by being one secondary vertex algorithm, has the exact same procedure as specified above, however now the threshold is the observable discriminant variable L/σ_L defined as flight

[2] $d_{3D}/\sigma_{d_{3D}} > 2$, where d_{3D} is the three dimensional distance between the primary vertex and the point of closest approach of the track to this vertex, and $\sigma_{d_{3D}}$ its uncertainty.[Collaboration, 2016]

length significance and where L corresponds to the distance between the primary vertex and the inclusive secondary vertex signed by the respective jet direction, as on impact parameters. But despite a much smaller mistag ratio when compared to impact parameters based algorithms, it had a much limited secondary vertex finding efficiency.

SV1 algorithm tries to remedy such limitations by basing its analyze on the very same IP2D/IP3D likelihood ratio variables, besides exploiting other properties associated to the vertex, as well as the SV0 flight length significance related variables, all presented in table 2.1. The SV1 has a clear vertex finding efficiency dependency related to the event topology, however, by considering the simulated b- and c-jet efficiencies, better results can be achieved when compared to its parent SV0.

Table 2.1: SV1 low-level b-tagging algorithm output variables.[Aad, 2019]

Variable	Description
SV1 L_{xy}	Transverse distance between the primary and secondary vertex
SV1 L_{xyz}	Distance between the primary and the secondary vertex
SV1 S_{xyz}	Distance between the primary and the secondary vertex divided by its uncertainty
SV1 mass	Invariant mass of tracks at the secondary vertex, assuming pion mass
SV1 E_{frac}	Energy fraction of the tracks associated with the secondary vertex: energy of vertex / energy of jet, considering charged tracks)
SV1 N_{Track}	Number of tracks used in the secondary vertex
SV1 N_{2Tpair}	Number of two-track vertex candidates
SV1 $\Delta R(\mathbf{p}_{jet}, \mathbf{p}_{Vtx})$	ΔR between the jet axis and the direction of the secondary vertex, relative to the primary one

This algorithm analyses tracks with $p_T > 400\,\text{MeV}$, $|d_0| < 3.5\,\text{mm}$ and no cut on z_0 [Collaboration, 2016] excluding the possible two-track groups related to photon conversions, long-lived particle decays and interactions between particles or/with detector materials more precisely, and then, evaluate iteratively if each of this tracks define a precise track-vertex using a χ^2 statistical test. On each algorithm iteration, the track-vertex with a larger χ^2 is removed and the vertex fit is remade, making it possible to get a good vertex approximation at the end with the additional constraint of this vertex invariant mass be less than 6 GeV, once higher vertex invariant mass energies are not likely to be originated from b- or c-hadrons [Aad, 2019].

Another relevant vertex based algorithm is the JetFitter that is intrinsically different from SV1 and doesn't make use of statistical data analyses. Instead, it relies on a modified Kalman filter that provides the approximating chain vertices by initially finding the common line defined between the primary vertex and the secondary bottom and charm vertices and then, by obtaining the approximating vertex positions of tracks selected from jets with $p_T > 500\,\text{MeV}$, $|d_0| < 7\,\text{mm}$ and $z_0 \sin\theta < 10\,\text{mm}$ [Collaboration,

2016], meaning that both b- and c-hadron vertices will always be isolated from each other, even tough only one track be attached to each one.

Besides some similar variables defined for SV1, this algorithm outputs too some new variables related to the decay topology shown in table 2.2. These variables have their values greatly influenced by the jets characteristics and, therefore, contribute to a better discrimination between jet-flavors identification.

Table 2.2: JetFitter low-level b-tagging algorithm output variables.[Aad, 2019]

Variable	Description
JF S_{xyz}	Significance of the average distance between PV and displaced vertices, considering all multi-prong vertices or (if there are none) of all single-track vertices
JF mass	nvariant mass of the tracks fitted to the vertices with at least two tracks
JF E_{frac}	Fraction of the charged jet energy in the secondary vertices
JF $N_{TrackAtVtx}$	Number of tracks from multi-prong displaced vertices
JF N_{2Tpair}	Number of two-track vertex candidates (Prior to decay chain fit)
JF $N_{SingleTracks}$	number of single track vertices
JF N_{Vtx}	Number of vertices with more than one track
JF $\Delta R(\mathbf{p}_{jet}, \mathbf{p}_{Vtx})(JF)$	ΔR between the jet axis and the vectorial sum of momenta of all tracks associated to the displaced vertex

On both methods there was a clear performance development from Run 1 to Run 2, consequence of the pile-up rejection and high-jet p_T performance enhancement, and therefore the overall algorithms performance was increased. Despite all the tagging code developments made on SV1, further track-cleaning requirements, that allowed a better data analysis, were implemented for jets in $|\eta| \geq 1.5$, in order to minimize the detector material interaction influence which was badly constraining the vertex finding efficiency, and additional track specifications were applied, such as consider the 25 highest-p_T tracks only. On JetFitter, besides the track selection optimization and material interaction mitigation, a vertex-mass dependent criteria selection was inserted, thus increasing the detection efficiency of vertices.

2.3.2 High-level *b*-tagging algorithms

Given that both impact parameters and vertex displaced based algorithms have more or less the opposite behaviors, in the means that the first ones have higher b-tagging efficiencies and the second have lower mistag rates, a combined model would have a much better tagging performance.

At the beginning several models were developed by combining two of the previous algorithms, due to lack of resources, however, latter on, the possibility of combining more than two methods started to be achievable mediated by artificial Neural Networks (NN), which allowed more complex input variable correlations.

One of the first so called high-level b-tagging algorithms to be developed was the mainly straightforward LLR-based IP3D+SV1 algorithm. Once both IP3D and SV1 resembled their functioning on the LLR variables, only a few dependencies were needed to be accounted for and no complex correlations were necessary, however this code was kind of restrictive and it was with the following algorithms, the IP3D+JetFitter and MV NN-based ending on the DL1, that outstanding improvements in b-tagging performance were seen.

The IP3D+JetFitter is implemented as a shallow copy of the JetFitter code but now with the IP3D output taken as a weighted input node and the NN structure of the algorithm defined with two intermediate layers with 9 and 14 nodes, respectively, where the discriminating variables used to discard light-flavor jets and c-jets from b-jets are defined as on equation 2.7. This method ensured great performance improvements, once it provided a better discrimination power given by the new outputed discrimination variables [Collaboration, 2016]

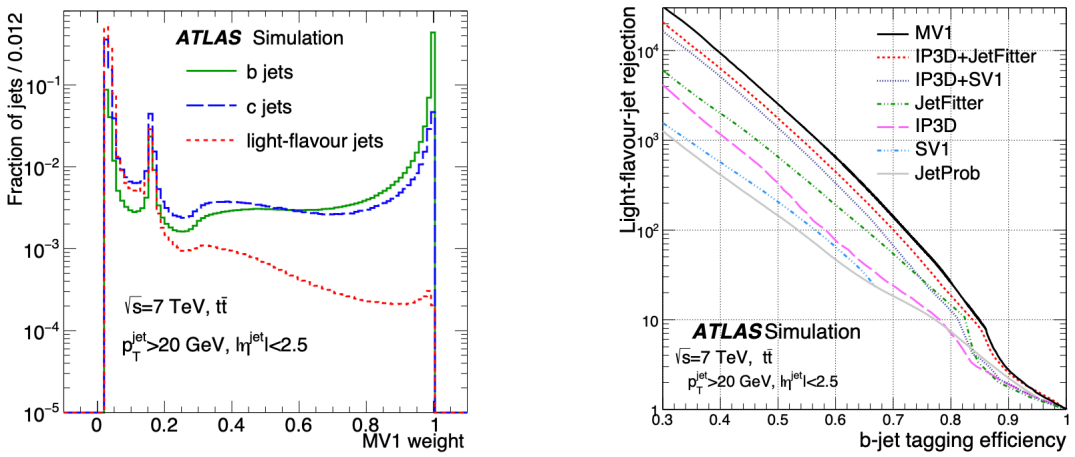
$$w_{IP3D+JetFitter} = \log\left(\frac{p_b}{p_u}\right) \qquad w_{IP3D+JetFitter} = \log\left(\frac{p_b}{p_c}\right) \tag{2.7}$$

The MV1 is a more powerful and complex algorithm, which is based on a deep neural network (DNN) and makes use of the IP3D and SV1, as well as the combined IP3D+JetFitter, algorithm discriminating variables and their correlations. The deep neural network implemented is constructed using the MLP code, a neural network code, from TMVA package [Hoecker et al., 2007] and consists on three different layers: two hidden layers, with three and two nodes respectively, and one output layer, with one node, where the final discriminant variable is constructed.

The MV1 DNN relies on a training process and therefore it's imperative to receive data with no kinematic dependencies, or else the jet-flavor kinematic different distributions would affect the b-tagging performance. For that a technique very much used is applied, by binning the p_T and the η jet distributions and selecting the exact same number of jets per bin in order to define a flat kinematic distribution for all jet-flavors. With the kinematic reweighting done, the data is inputted on the DNN that is trained on a back-propagation algorithm based in simulated b- (signal hypothesis) and u-jet (background hypothesis) samples. Results can be seen on figure 2.2 for $t\bar{t}$ simulated events, where the outputted variable weight and the u-jet rejection ROC curves are presented, the first for each jet-flavor and the last for each one of the previous algorithms.

There are two different high-level tagging algorithms, the MV2 and the DL1, that are much more beneficial in terms of performance improvements when compared to the previous ones. These algorithms combine the outputted low-level variables, described in this section (tables 2.1 and 2.2), and the jet kinematic properties (p_T and $|\eta|$) with different methods, increasing the overall performance that previously was restricted only to each low-level algorithm.

The MV2 tagger [Collaboration, 2017b], which follow the MV1, is a second generation multi-variable algorithm based on a so called Boosted Decision Tree (BDT). In this code the training process is made with $t\bar{t} + Z'$ samples, previously reweighted in p_T and η spectrum using the ROOT Toolkit for Multivariate



(a) (b)

Figure 2.2: (a) Distribution of the tagging weight obtained with the MV1 algorithm, for three different flavors of jets. (b) Light-flavor-jet rejection versus b-jet tagging efficiency, for various tagging algorithms.[Collaboration, 2016]

Table 2.3: Output variables produced by JetFitter c-tagging and used by DL1 high-level tagger. [Aad, 2019]

Variable	Description
JF $L_{xy}\left(2^{nd}/3^{rd}vtx\right)(JF)$	Transverse displacement of 2^{nd} or 3^{rd} vertex
JF $L_{xyz}\left(2^{nd}/3^{rd}vtx\right)$	Distance of 2^{nd} or 3^{rd} vertex from the primary vertex
JF $m_{Trk}\left(2^{nd}/3^{rd}vtx\right)$	Invariant mass of tracks associated with the 2^{nd} or 3^{rd} vertex
JF $E_{Trk}\left(2^{nd}/3^{rd}vtx\right)$	Energy fraction of the tracks associated with the 2^{nd} or 3^{rd} vertex
JF $f_E\left(2^{nd}/3^{rd}vtx\right)$	Fraction of charged jet energy in 2^{nd} or 3^{rd} vertex
JF $N_{TrkAtVtx}\left(2^{nd}/3^{rd}vtx\right)$	Number of tracks associated with 2^{nd} or 3^{rd} vertex
JF Y_{trk}^{min}, Y_{trk}^{max}, $Y_{trk}^{avg}\left(2^{nd}/3^{rd}vtx\right)$	Minimum, maximum and average track rapidity of tracks at 2^{nd} or 3^{rd} vertex

Data Analysis (TMVA) [Hoecker et al., 2007]. This reweight process is identical to the one described for MV1 and is very important in all high-level algorithms, once it makes all jet-flavors uniform on the p_T and η spectrum, particularly important on training because the learning process shouldn't be jet-flavor dependent.

The BDT model needs several parameters, usually named as hyperparameters, to define the training performance output and the model properties. Commonly, as a way of maximizing the tagging rate and learning process, optimization studies are conducted and MV2 was no exception.

The DL1 is relatively different from the MV2, and uses a deep-forward neural network trained with reweigthed $t\bar{t} + Z'$ samples, using keras [F. Chollet et al., 2015] with Theano backend [The Theano Development Team and et al., 2016] and with the Adam optimizer [Kingma and Ba, 2014]. The major distinction between MV2 and DL1 relies on the output topology, that whilst on the MV2 is one single output with the discriminant variable, on DL1 is a multi-dimensional output with the jet-flavor probabilities.

On DL1, besides the input variables associated to the MV2 - kinematics, IP2D/IP3D, SV1 (Table 2.1) and JetFitter (Table 2.2) output variables - a few additional output variables from the JetFitter c-tagging algorithm (Table 2.3), a variant of the JetFitter which finds the secondary or tertiary vertice properties, is used. Furthermore, similarly to the MV2 model, the DL1 needs a hyperparameters specification, now related to the NN structure, the training epochs, the learning rate and the batch size. By maximizing the b-tagging performance, one can optimize the hyperparameters, obtaining the parameters presented in table 2.4.

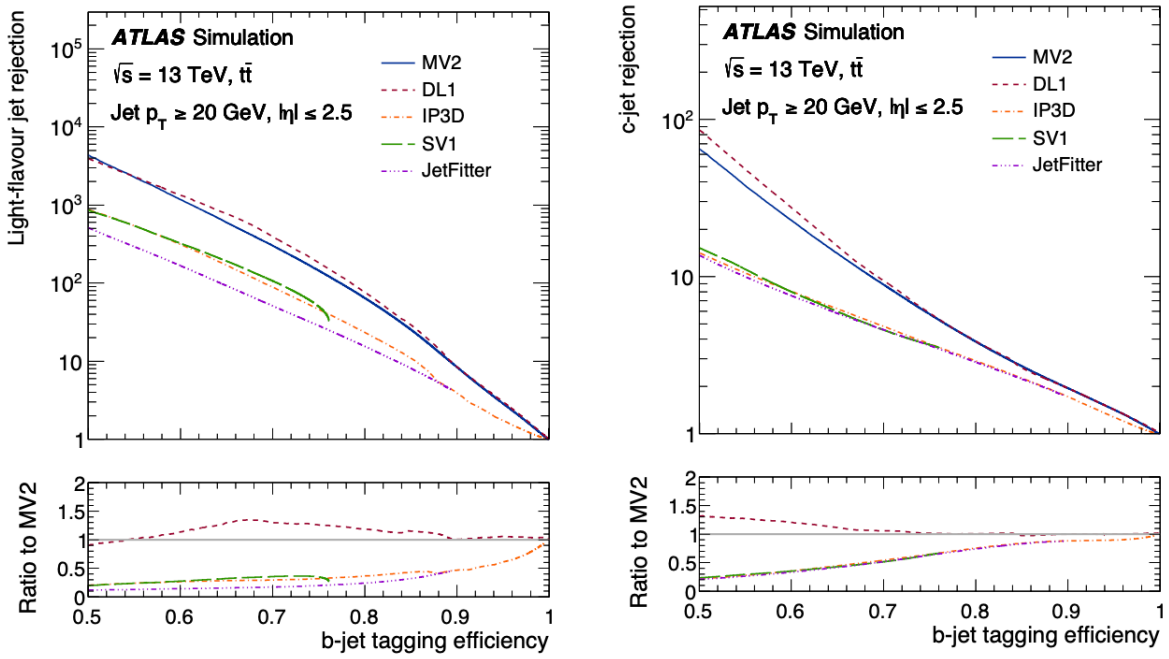
Table 2.4: List of optimized hyperparameters used in the DL1 b-tagging algorithm.[Aad, 2019]

Hyperparameters	**Value**
Number of input variables	28
Number of hidden layers	8
Number of nodes [per layer]	[78, 66, 57, 48, 36, 24, 12, 6]
Number of Maxout layers [position]	3[1, 2, 6]
Number of parallel layers per Maxout layer	25
Number of training epochs	240
Learning rate	0.0005
Training minibatch size	500

To evaluate the different algorithm performances, a discriminant that takes into count the b-tagging efficiency detection and the other flavor-jet rejection efficiency was developed. For DL1 this discriminant is defined as on equation 2.8 and depends on b-, c- and u-jet probabilities (p_b, p_c and p_u respectively) as well as on the effective c-jet fraction (f_c) which is predefined on the background training sample.

$$D_{DL1} = \ln\left(\frac{p_b}{f_c\, p_c + (1 - f_c)\, p_u}\right) \tag{2.8}$$

Despite both algorithms have more or less the same performance, as shown in figure 2.3, it's preferable to use the DL1. This one, due to the use of a multi-class network architecture, provides improvements in memory related issues.



(a) (b)

Figure 2.3: The (a) u-flavor jet and (b) c-jet rejections versus the b-jet tagging efficiency for the IP3D, SV1, JetFitter, MV2 and DL1 b-tagging algorithms evaluated on the baseline $t\bar{t}$ events.[Collaboration, 2016]

2.4 DL1 algorithm

Given that tagger studies concluded that the DL1 tagger had a better performance when compared to its fellow high-level taggers [Lanfermann et al., 2017], and due to the huge development on the NN area, DL1 rapidly became the only eligible high-level tagger, and thus, it is of great importance understand its functioning, starting by the NN basis functioning.

A NN [Ian Goodfellow, 2016] is a group of hidden layers with an arbitrary number of hidden nodes, that can be distinct in number from layer to layer. The most simple way to explain a NN functioning is by knowing that it receives a certain input and constructs a variable output, which depends on the data and objectives, and for that it has to be trained.

Commonly a NN is based on two different processes called forward and backward propagation, where the former one is used to obtain the results straightforward, and the last one to calculate the training parameters, with both of these processes to be done on each layer and therefore node as illustrated in figure 2.4.

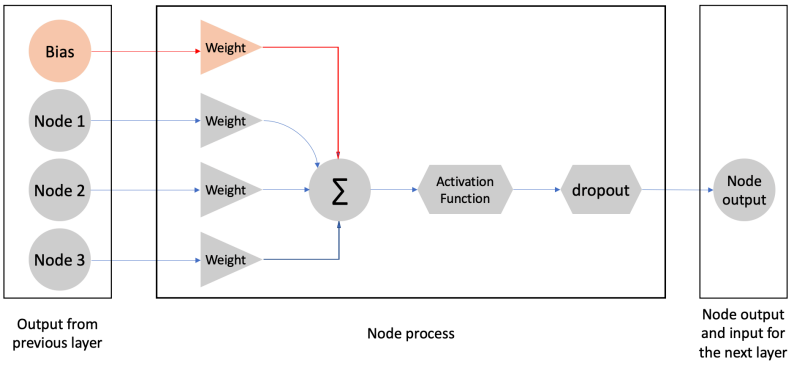


Figure 2.4: Perceptron functioning scheme of a fully connected layer.

At forward propagation, at each node, a bias term and all the data outputted by the previous layers, with the exception of the input layer where the inputs are taken instead, will be submitted into a weighted sum operation which result will then be multiplied by an activation function and a dropout value, with these last defined by the user. Such operation is not trivial and implies a lot of data analysis pressure, once per each input node (from the previous layer), each output node (which value is to be calculated on the next layer) and on each layer there will be one weight initially random that will be tuned at the NN training as the NN loss function is minimized. Besides, this weighted sum operation, generally represented in equation 2.9 where it's assumed that there are N input nodes which values are represented by vector $\mathbf{x}$ and a weight vector $\mathbf{w_i}$ formed by the input node weights for the i-th output layer node, can very much differ depending on the base code implemented and thus the f and g functions presence in figure 2.9.

$$\Sigma\left(\mathbf{x}, \mathbf{w_i}\right) = \sum_{j=1}^{N} f(x_j) \cdot g(w_{i,j}) \tag{2.9}$$

While the weights are given by the training process itself, formed by consecutive forward and backward propagation, the same does not happen for the activation function and dropout, but they have the very same important role, as they allow the appliance of non linear combinations between layers, decreasing the data correlation factor influence.

The backward propagation is deeply related to the forward one, being initialized at the end of the forward one. Once the final output had been obtained, the NN resulting predictions are compared to the real data ones, saved as well but not entering in the train itself. This will allow the calculation of the error referent to the train results, giving the accuracy of the prediction or/and the model loss, referent to a loss function, but mainly it shall allow the obtainment of the updated weights and bias which are in turn dependent of a predefined parameter called learning rate.

A NN training is achieved when a forward followed by a backward propagations are recursively made, with the total number of recursions called epochs, resulting in better training accuracy results and

more precise training weights.

The Deep Learning 1 or DL1 Neural Network, presented in figure 2.5, is a more complex NN with a mixture of fully connected hidden layers and Maxout layers [Goodfellow et al., 2013] normalized with the Batch distribution, that are activated with a rectified linear unit function [Nair and Hinton, 2010; Bishop, 2006] with the exception of the output, where a SoftMax activation function [Bishop, 2006] is used instead to transform the outputs into probability distributions.

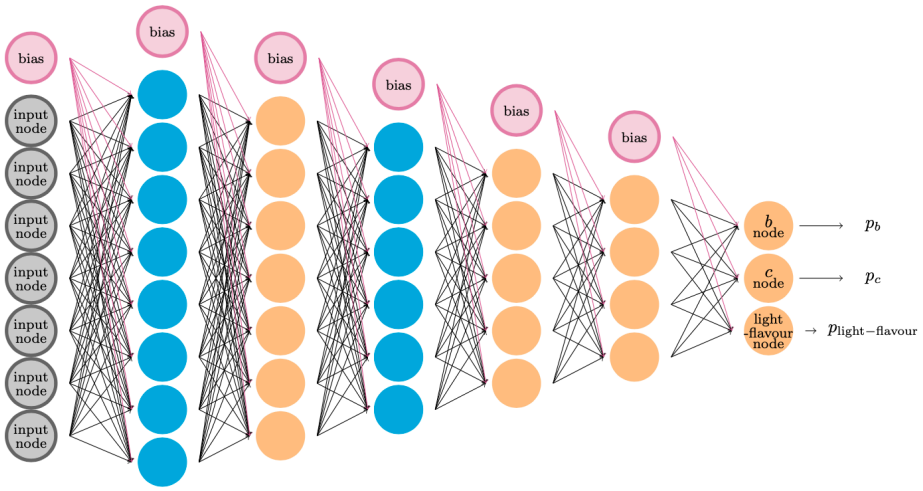


Figure 2.5: DL1 baseline layers and node disposition.

The main DL1 difference to the NN base described above is that now, in addition to the fully connected hidden layers structure, there are maxout pooling layers too, represented in 2.6, that proportionate the propagation of the element-wise maximum leading to a superior training performance.

Hence, the DL1, by receiving a jet feature sample, a set of data represented by N vectors with n jet feature values (N the total number of jets and n the total number of features), proportionate the training of the jet features that will result in a two dimensional output, where a jet-flavor probability (b-jet, c-jet and u-jet), granted by the softmax activation function, will be predicted by the model, for each jet inputted [Lanfermann et al., 2017].

Finally, by using this output, with b-, c- and light-flavor probabilities (p_b, p_c and p_u), a discriminant variable is constructed and the flavor-tagging process is possible. Such discriminant is presented in equations 2.10 and 2.11 for both b-tagging and c-tagging, respectively.

$$DL1_b = \ln\left(\frac{p_b}{f_c \cdot p_b + (1 - f_c) \cdot p_u}\right) \tag{2.10}$$

$$DL1_c = \ln\left(\frac{p_c}{f_b \cdot p_c + (1 - f_b) \cdot p_u}\right) \tag{2.11}$$

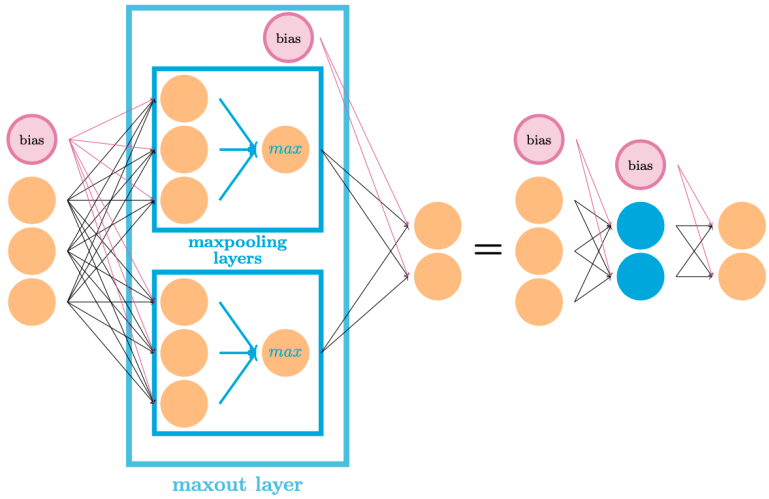


Figure 2.6: Maxout pooling layer. [Lanfermann et al., 2017]

As DL1 is trained over the jet features values for a certain number of jets, its performance is very much dependent on which features and on how many jets it uses.

At the moment there are four different DL1 versions (2.7) that depending on the jet features used, outputted from the low-level algorithms, can be classified as DL1 baseline, the original one, DL1r or DL1rnnip, which makes use of the RNNIP tagger, DL1mu, that takes muon data into count, and at last the DL1d, which replaces the RNNIP by the DIPS tagger.

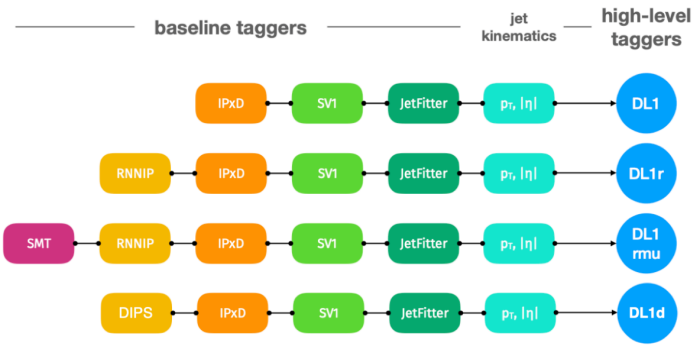


Figure 2.7: DL1 algorithm updates that result on the construction of different high-level b-tagging algorithms.

This algorithm first started by only using the low-level tagger algorithms, referenced before, but soon some limitations were identified and, later on, several DL1 releases were produced. One of the limitations identified in the DL1 arose from the fact of, due to technical limitations related to computational power, the correlations between other track features, besides the impact parameters, not be taken into count

during the NN training. In order to decrease memory problems [Collaboration, 2017c, 2020], the binning size taken during the likelihood algorithm (used by the IP2D/IP3D) for impact parameters needed to be broad, which in addition to the not studied impact parameters track correlations [Collaboration, 2017a, 2020], implied direct consequences on the resulting data.

Thus, to give answer to such problems the scientific community tried to implement a new Neural Network called Recurrent Neural Network (RNN) that would be trained with just impact parameter based features and would then replace the IP based low-level algorithms, originating the second generation DL1 tagger called DL1r or DL1rnnip.

The RNN algorithms have been used on different investigation areas such as on natural language processing and time-series analyses, and this is due to its possibility of extension to sequence-based and temporal domains. This type of NN architecture has an encapsulated cell with an internal state vector with a fixed number of inputs that is initialized to zero and, at each step, is recurrently combined with the next instance, formed by the inputted features, with the predefined training precepts/rules obtained during the training phase. Only at the end of this process, a fixed-dimension vector will be outputted, latter processed by a normal feed-forward NN.

Once this process is recurrently done, no dimension or variable length is necessary, meaning that it offers a great advantage to track features training, given that the number of tracks per jet is not a robust variable because many of the tracks are associated to ”fake” tracks.

So by defining a b-jet tracking features as a variable-length vector it was possible to apply it in a Recurrent Neural Network (RNN) architecture, which was latter consolidated as RNNIP which architecture is shown on figure 2.8. The RNNIP tagger now is inputted with the very same features taken by IP3D, which is ensured by the application of the exact same jet quality criteria applied to IP3D to RNNIP, and new additional tracking features, this being the track's transverse momentum fraction (p_T^{frac}) and the $\Delta R(track, jet)$, which is expected to increase the b-tagging performance.

However, it was only with Deep Impact Parameter Sets (DIPS) algorithm that more tracking features were possible to be trained. This algorithm, that would latter on replace the RNNIP and provide the appearance of the DL1d, due to its new architecture, allowed the consideration of more features with no computing power penalization. Such model was first applied in particle physics on the identification of different types of jets [Komiske et al., 2019] and was formalized as on equation 2.12, where the $\Omega(\{\mathbf{p_1}, ..., \mathbf{p_n}\})$ means the set of flavor-jet probabilities obtained by the n track features and $\mathbf{p_i}$ the ith track features' vector.

$$\Omega\left(\{\mathbf{p_1}, ..., \mathbf{p_n}\}\right) = \mathcal{F}\left(\sum_{i=1}^{n} \Phi\left(\mathbf{p_i}\right)\right) \tag{2.12}$$

DIPS can be defined in two separated processes represented by the functions Φ and $\mathcal{F}$, which are basically two different NN's with different architectures, shown in figure 2.9.

Having in mind that for each jet there will be several tracks (n total tracks), each one of them with a certain number of tracking features (m features), on DIPS, first, the inputted data is separated per tracks,

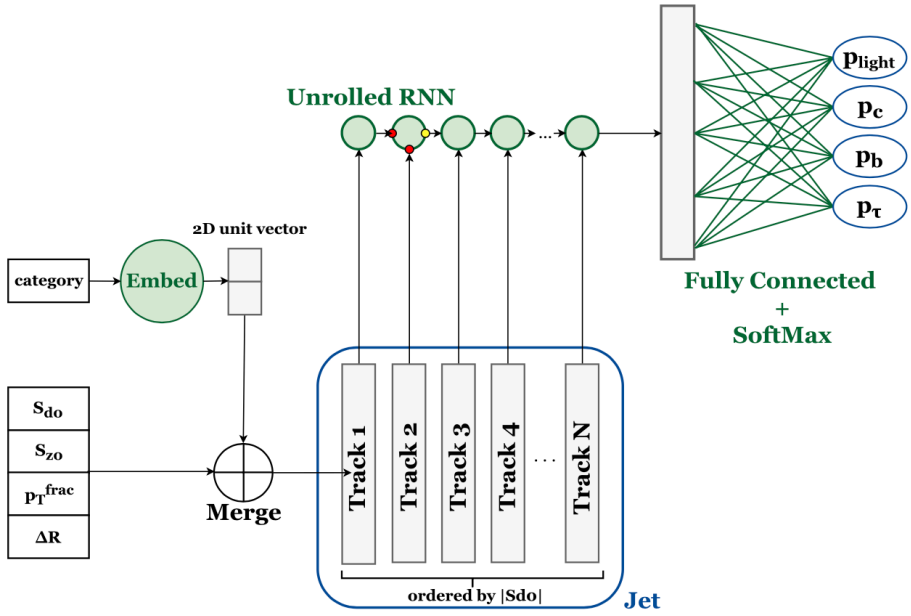


Figure 2.8: RNNIP Neural Network architecture scheme.[Collaboration, 2017a]

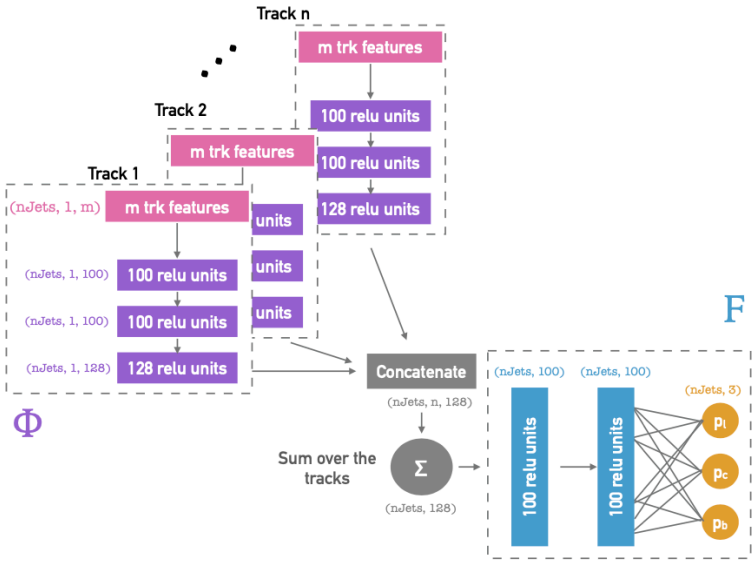


Figure 2.9: DIPS low-level algorithm architecture scheme.[Collaboration, 2020]

resulting in n sets, that are individually feed into a forward NN called Φ, and secondly, with all tracks processed by Φ, the resulting vectors are concatenated and summed over the tracks and a vector with the

overall track feature correlations is obtained, which is then processed by $\mathcal{F}$ NN, intended to study the track correlations.

The main distinction between this last algorithm and RNNIP is the process/training efficiency and the data processing. Since RNNIP has a sequential training, the tracks that are used must be ordered with this process be time consuming and purely none physical, given that the b-hadron decay products do not exhibit any intrinsic sequential ordering [Collaboration, 2020].

Chapter 3

Sampling Preprocessing

With all high-level b-tagging algorithms already applied on pp data collisions and with the technical improvements conducted on NNs, it is now possible to apply these algorithms to heavy ions data collisions, where taggers will work on an extreme environment with lots of particles emitted.
This chapter will be centered on the explanation of the sample preprocessing that is subdivided in several stages, from the Analyses Object Data files (AOD) extraction to the data treatment required for the neural networks training process.

3.1 Methodology

At section 2 was described the ATLAS b-tagging algorithms evolution and some results obtained for pp data collisions, but these results were specific for a given tagger, its architecture and input training variables, not to speak on the sample processing, characterized by the event selection.

Given the Pb+Pb b-tagging performance study interest, all of the previous enumerated parameters will change, starting by the event selection, explained by the underlying event influence, and the training input features, where some of them are much more meaningful for Pb+Pb.

Hence, during the production of this work, it was used a package called UMAMI [The ATLAS Flavor Tag Group, 2022b], produced by the ATLAS Flavor Tagging Group (FTAG group), that made the preprocessing of the samples and the training of the Neural Networks possible. Despite this module has been developed for flavor tagging proton collision data analyses, the application to Pb+Pb collisions data was possible after small modifications.

Actually this module not only selects the interesting events, by restricting the features values at choice, but applies the so called undersampling, previously explained, which samples are then enabled to proceed to the training step.

The first steps toward the b-tagging on Pb+Pb data are related to the construction of the training, validation and testing samples. These three samples are made by events selected with a certain number of predefined cuts, usually related to the jet object characterization/definition, and with additional constraints

given by the study in question, which imply cuts on kinematic variables such as p_T.

In order to grant a good NN learning performance the three specified samples must have the exact same preprocessing, the data treatment applied previous to the training step. This is important because, while the training sample is only used during the NN training, and therefore for NN weight calculations, the validation and testing samples are the ones that are used to obtain the results ensured by the weights given by the training sample, with the only difference between these samples be on the dimension size, where the validation is only a fraction of the testing sample.

Consequently, if any of the last two samples had a different sample preprocessing regarding the training sample, it would mean that the NN wouldn't had been trained for the testing sample and the results would be rather penalized, due the the NN bad performance. As such, to ensure a good NN performance there were applied the exact same preprocessing cuts across all samples, shown in section 3.2.

Another very important and necessary data treatment is related to the already mentioned downsampling or undersampling that is done on section 3.3. This step, following the preprocessing cuts appliance, is only needed due to the particle jet physics and detector limitations, and is used as a way of maximizing the NN learning performance. The goal of the downsampling is to provide a non kinematic dependent flavor tagging, just by producing a new data sample were the kinematic variables, such as the jet p_T and the jet pseudorapidity have their contributions eliminated or deeply minimized. This final sample will then be used at the training stage and will allow the NN to give results in a non specific kinematic range.

3.2 Monte Carlo Samples

In order to compare pp with Pb+Pb data collision results, two datasets were used with these being gathered through dijet generation for five different ranges of truth p_T of leading simulated jet, where pp and Pb+Pb data are obtained differently, with pp data collision being generated by Pythia8 [T. Sjöstrand, 2015; Collaboration, 2014] and Pb+Pb the same as on pp, but also embedded in MinBias real Pb+Pb data to have a reliable underlying event description. The used AODs are presented in tables 3.1 and 3.2.

Additionally, for Pb+Pb with the intend of maximizing the NN learning rate and consequently tagger performance, three other sets of samples were used, each one enriched in b, c and light flavor-jets, which would be grouped in one sample.

The AODs are separated by the range of the truth p_T of leading simulated jet where a good jets statistic is generated on each p_T range enumerated bellow and which JZ tags are identified at the AODs:

- JZ1 : 20–60 GeV
- JZ2 : 60–160 GeV
- JZ3 : 160–400 GeV
- JZ4 : 400–800 GeV

- JZ5 : 800–1300 GeV

Table 3.1: List of AODs used for pp data samples

AODs
mc16_5TeV.420011.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ1R04.recon.AOD.e6608_s3238_r11199
mc16_5TeV.420012.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ2R04.recon.AOD.e6608_s3238_r11199
mc16_5TeV.420013.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3R04.recon.AOD.e6608_s3238_r11199
mc16_5TeV.420014.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4R04.recon.AOD.e6608_s3238_r11199
mc16_5TeV.420015.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5R04.recon.AOD.e6608_s3238_r11199

Table 3.2: List of AODs used for Pb+Pb data samples

AODs enriched on b-jets
mc16_5TeV.800893.Py8EG_A14N23LO_jetjet_JZ1WithSW_bfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800894.Py8EG_A14N23LO_jetjet_JZ2WithSW_bfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800895.Py8EG_A14N23LO_jetjet_JZ3WithSW_bfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800896.Py8EG_A14N23LO_jetjet_JZ4WithSW_bfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800897.Py8EG_A14N23LO_jetjet_JZ5WithSW_bfilter.merge.AOD.e8366_d1521_r11472_r11217
AODs enriched on c-jets
mc16_5TeV.800888.Py8EG_A14N23LO_jetjet_JZ1WithSW_cfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800889.Py8EG_A14N23LO_jetjet_JZ2WithSW_cfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800890.Py8EG_A14N23LO_jetjet_JZ3WithSW_cfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800891.Py8EG_A14N23LO_jetjet_JZ4WithSW_cfilter.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.800892.Py8EG_A14N23LO_jetjet_JZ5WithSW_cfilter.merge.AOD.e8366_d1521_r11472_r11217
AODs enriched on light flavor-jets
mc16_5TeV.801117.Py8EG_A14N23LO_jetjet_JZ1WithSW_c_b_veto.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.801118.Py8EG_A14N23LO_jetjet_JZ2WithSW_c_b_veto.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.801119.Py8EG_A14N23LO_jetjet_JZ3WithSW_c_b_veto.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.801120.Py8EG_A14N23LO_jetjet_JZ4WithSW_c_b_veto.merge.AOD.e8366_d1521_r11472_r11217
mc16_5TeV.801121.Py8EG_A14N23LO_jetjet_JZ5WithSW_c_b_veto.merge.AOD.e8366_d1521_r11472_r11217

3.3 Event Selection

Once much of the detected events are related to ”fake” track identifications, several events are contaminated with jets originated from material interactions, and thus a data selection is necessary to separate consistent jets from background. Hence, by applying a certain number of variable cuts, it is possible to retain the interesting events with quite effectiveness. The trick here is to neither do much cuts, implying interesting data elimination or the origin of non physical tendencies or bias, and neither do negligible cuts, which provides data corruption.

The training-dataset-dumper code [The ATLAS Flavor Tag Group, 2022a], developed by the ATLAS flavour tag group was used to filter data and apply the first set of cuts done on the tracks, restricting the amount of data in AOD files, with the jet cuts being applied on the UMAMI package and on the sample preprocessing that originates the training, validation and testing samples.

- **Track cuts**

  Track $p_T : > 2\,\text{GeV}$

  $|\eta| : < 2.5$

  $d_0 : < 3.5\,\text{mm}$

  $z_0 : < 5.0\,\text{mm}$

  $N_{Si} : > 7$

  $N_{Si}^{Share} : < 1$

  $N_{Si}^{Hole} : < 2$

  $N_{Pix}^{Hole} : < 1$

- **Jet cuts**

  JetFitterSecondaryVertex mass $: < 25\,\text{GeV}$

  JetFitterSecondaryVertex energy $: < 100\,\text{GeV}$

  $\Delta R : < 0.6$

  Jet $p_T : > 50\,\text{GeV}$

Hence, by applying the above enumerated cuts less data will be retained, where, hopefully, most of the non physical data will vanish. In order to see if the cuts are being effective, variable distributions analysis at the different cut stages are quite helpful and the jet p_T is a great probe for this objective. Since its spectrum is very well study with a well known characteristic distribution, a power law with a negative exponent decay distribution alike is expected, however, if all initial files would be used to plot the jet p_T distribution, the obtained plot would not be the expected and it would be more like the represented in

figure 3.1a instead, in this case for pp data collisions but it would be the same for Pb+Pb. In this figure each JZ range file contribution is observed, where five peaks could be seen: three of them very much clear (JZ3 near the 200 GeV, JZ4 near the 400 GeV and JZ5 near the 800 GeV) and two of them looking as one single peak, but being two in fact (JZ1 and JZ2).

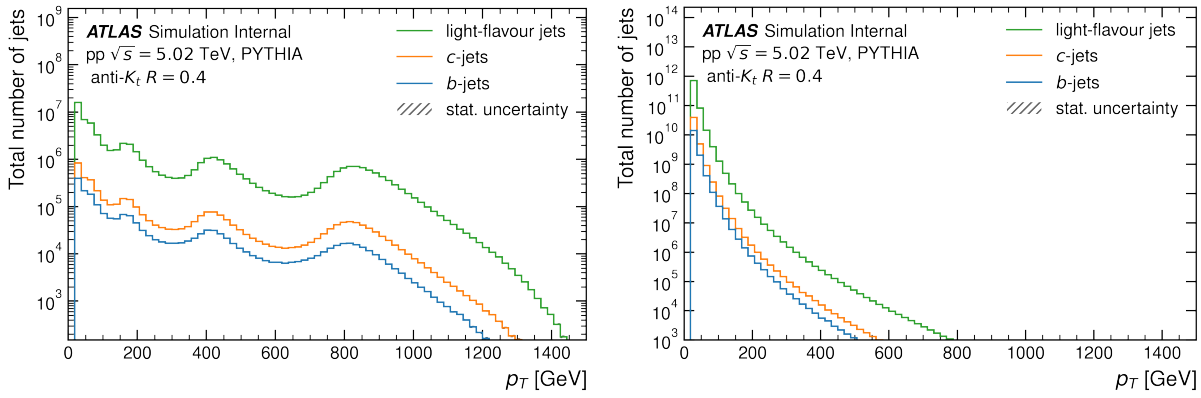


Figure 3.1: (a) The transverse momentum distribution (p_T) of pp jets with no weight applied and across all JZ samples. (b) The same as on (a) but now the contributions from each JZ file are weighted, ensuring a consistent pT distribution.

The reason for such a different distribution in figure 3.1a is justified by the use of the non weighted JZ range contributions, showed in section 3.2, that have different interaction probabilities (cross-sections) and filter efficiencies. A meaningful jet pT distribution is obtained by weighting each file contribution with the product cross-section and filter efficiency, both represented on 3.3, and the expected jet p_T distribution arise, shown in figure 3.1b, where, now, the number of jets decrease with a negative exponent power law as the jet p_T increases. Another conclusion is that the u-flavor jets are dominant, followed by the c- and b-flavor jets.

Table 3.3: pp data Pythia cross-sections and filter efficiencies values simulated for each p_T range (JZ).

PYTHYA	**JZ1**	**JZ2**	**JZ2**	**JZ4**	**JZ5**
Cross Section (nb)	6.79×10^{7}	6.40×10^{5}	4.72×10^{3}	2.66×10^{1}	2.25×10^{-1}
Filter Efficiency	2.8748×10^{-3}	4.2952×10^{-3}	5.2994×10^{-3}	4.5901×10^{-3}	2.2846×10^{-3}

However, even though the non weighted jet p_T distribution be quite different from the physical one it helps on cut effectiveness observation, and therefore, from now on, all jet p_T plots won't be weighted, with no impact on b-tagging effectiveness. This decision is also supported due to the additional treatment that the weight process implies. So, at training samples production, where the second set of cuts is applied,

an even further data restriction can be seen, as shown in figure 3.2, for the training sample. Now, both p_T and η distributions for pp and Pb+Pb data were plotted and clearly the pp data had a decrease on the amount of data, observed on the total number of jets per bin when comparing 3.1a and 3.2a. Additionally, in figure 3.2, as a consequence of the jet p_T cut on > 50 GeV, the JZ1 file contributions almost disappears. This shows to be beneficial on Pb+Pb, even though much of the interesting phenomena happen on low jet p_T ranges, given that most of the underlying events noisy contributions lives on the low level jet p_T ranges.

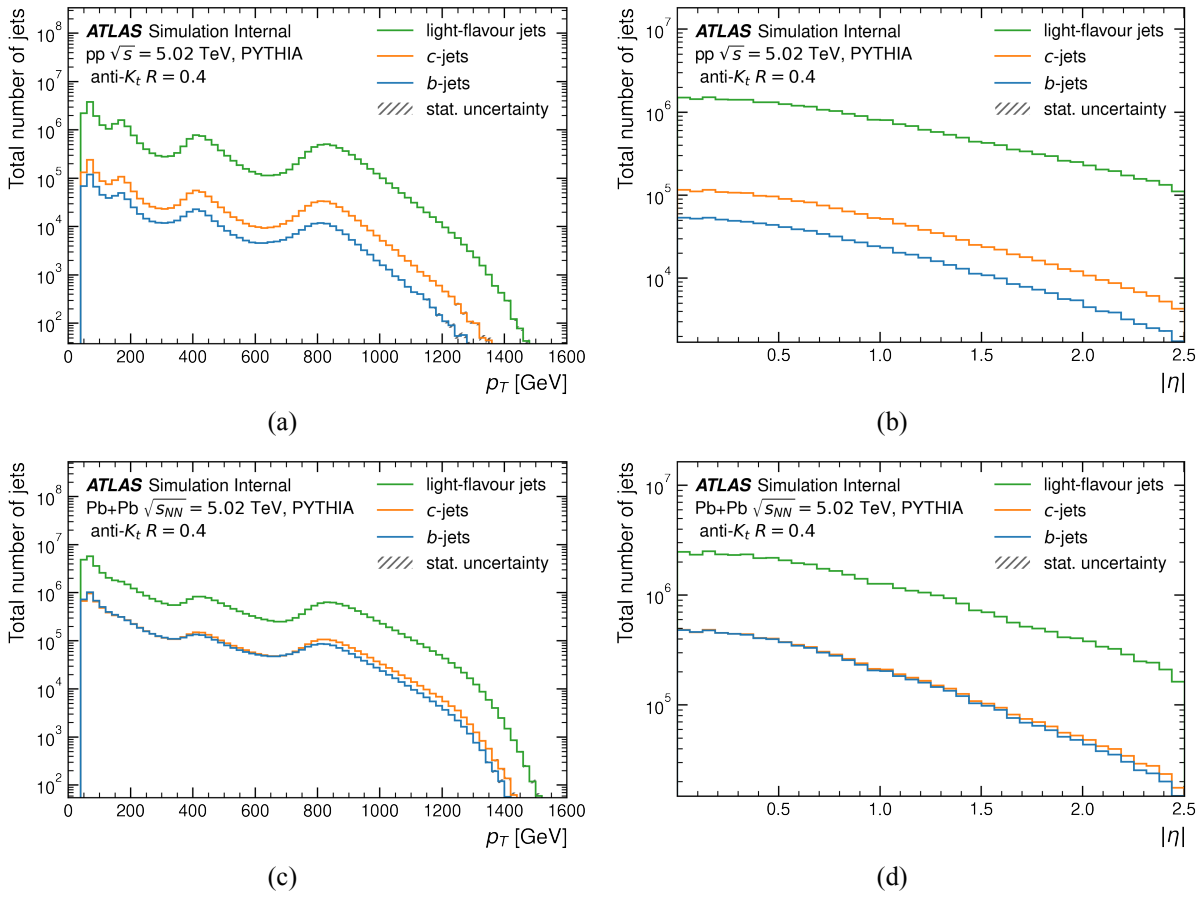


Figure 3.2: p_T and η variable distributions of the constructed train samples for pp, (a) and (b), and Pb+Pb, (c) and (d), respectively and across all JZ samples.

With all cuts applied, the generated files are submitted to the undersampling step. The way this process is made is quite easy to understand and it's based on a binning process, where both η and p_T have a certain number of bins, previously defined. For a given η bin range and a certain p_T bin range the exact same number of jets of each flavor will be imposed. By doing this recurrently for each variable

bin combination, a new sample with the exact same number of flavor-jets with no η and p_T variables influence will be constructed.

Thus, the flavor-jet with the minimum number of jets will tend to be dominant and will define the total amount of jets at the final downsample sample. Therefore, given that the lower number of jets per p_T and η bin is ensured, for pp, by the b-jet distribution, the final samples distributions for every flavor will be similar to the b-jet distribution, meaning that, for example, for the train sample the distributions will be the shown in image 3.3a and 3.3b, but the same can not be said about Pb+Pb. Due to the huge amount of Pb+Pb data that would arise computational stress latter during training, it is not possible to use all jet's statistic, and, as a consequence, a limitation on the number of jets must be applied.

Since the data files are separated by transverse momentum ranges, as previously specified, by default, a sum operation is applied over all JZ files at the downsampling stage to ensure a final file with a p_T range of $20 - 1300\,\mathrm{GeV}$. While for pp the weights used are defined to maximize the amount of jets, explained by the short amount of data at dispose and granting the above distribution, for Pb+Pb, a number of maximum jets was specified as a limit, choose as 5 million, and each JZ file contributed with a percentage of their jets to the overall jets' quantity present in the complete sample. Thus, as a consequence, the Pb+Pb downsample sample variable distributions will have a proportional lower number of jets per bin with the very same distributions as on the b-jet flavor distribution for Pb+Pb, as seen in 3.3c and 3.3d. In this case this jet's limitation was considered in order to minimize the computational training exhaustion.

The downsampling, is applied to the training sample, ensuring that the NN weights are optimized for flavor-jets discrimination, and to validation and test samples, which must have the exact same data treatment as on the training sample. For exemplification only the training sample distributions were plotted, but the other sample distributions can be seen in the Appendix A.

At this point with the undersample, validation and testing samples constructed, all requirements are achieved and we are able to continue to the training process that will be the main theme on the next chapter.

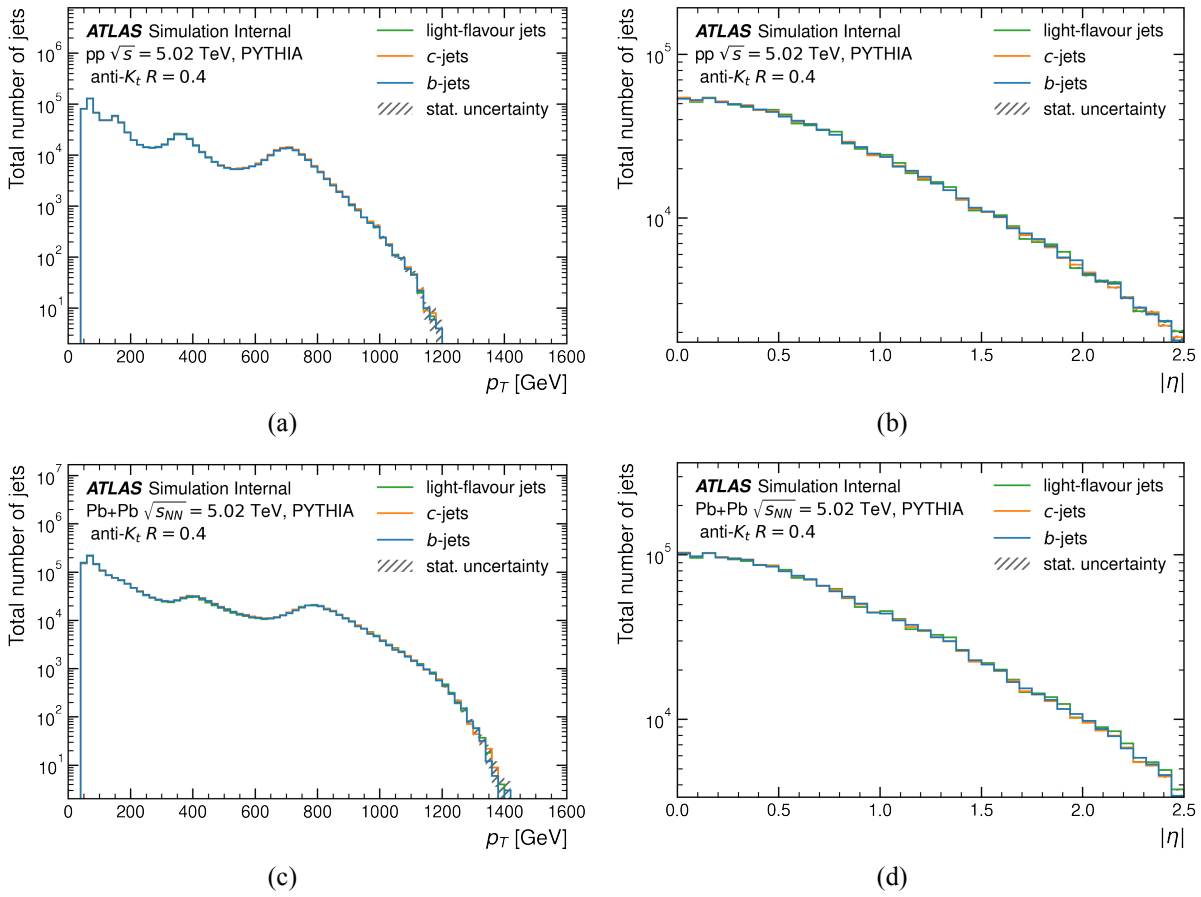


Figure 3.3: p_T and η variable distributions of the constructed downsample samples for pp, (a) and (b), and Pb+Pb, (c) and (d), respectively and across all JZ samples.

Chapter 4

Neural Network training

Once most of the undergoing scientific investigations are related to b-tagging identification on Pb+Pb collisions, such results are greatly desired by the scientific community. Hence, to ensure the results veracity, several tests and simulations are made with proton data collision results as comparison.
This chapter will board the training specifications done for two different NNs, DL1 and DIPS, and will discuss their results in a few studies, some of them related to Pb+Pb specific studies such as the analysis of the b-tagging performance as a function of collisions centrality.

4.1 Training Features

One of the very first things that one must think before training a Neural Network is the so called training variables or features set, used to discriminate flavor-jets and being related to the NN performance. In b-tagging there are a lot of features, related to b-tagging lower-level algorithms, that can be exploited, some of them referenced in the related work section. In general, all of these variables are either related to jets or tracks and, therefore it is usual to separate these variables in two sets of features that are explained bellow.

1. **Jet Features** - Related to jet properties and so related to JetFitter and SV1 algorithm outputs and combinations between them, as well as on the kinematic features and jet related impact parameter algorithms (IP2D, IP3D) output. These features are used by the DL1 baseline high-level tagger algorithm and its upgraded versions.

2. **Track Features** - Related to track properties obtained by the calorimeters and the Inner Detector (ID) module. This variables are used by the DIPS tagger.

Ideally, one would want a neural network capable of using all disposable features, and the NN evolution is certainly tending for that, but at the moment it is impossible. Hence, with the goal of making use of both track and jet features, currently in ATLAS, the best choice would be to make use of DIPS and

DL1 taggers. These algorithms, explained in section 2.4, are based in track and jet features, respectively, which can be defined differently depending on the approach, with these training variables having a direct impact at the NN learning. Besides, for a meaningful comparison between pp and Pb+Pb results, the same jet and track features should be applied to both sets of data.

Thus, following the UMAMI package, the training variables applied on tracks are the ones presented in table 4.1 and for jets are used the kinematic, IP2D and IP3D, JetFitter and SV1 features already explained in chapter 2, with the addition of the JetFitter+SV1 features listed in table 4.2.

Table 4.1: List of track related features used on DIPS training.

Track Features	**Description**
$p_{T_{frac}}$	Fraction of the jet pt carried by the track
Δr	Track Δr
d_0	Transverse impact parameter, distance of closest approach of the track to the primary vertex point in the r-ϕ projection
$z_0 \sin\theta$	Longitudinal impact parameter projected onto the direction perpendicular to the track
S_{d_0} IP3D	Signed transverse impact parameter significance from IP3D algorithm
S_{z_0} IP3D	Signed longitudinal impact parameter significance from IP3D algorithm
N_{Hits}	Combined number of hits in the pixel layers (including the IBL)
$N_{InnerHits}$	Number of hits in the IBL
$N_{NextToInnerHits}$	Number of hits in the next-to-innermost pixel layer
$N_{SharedInnerHits}$	Number of shared hits (contributing to the track fit and to another track) in the IBL
$N_{SplitInnerHits}$	Number of split hits in the IBL
$N_{SharedHits}$	Number of shared hits (contributing to the track fit and to another track plus the ones not marked as split hit) in the pixel layers (including the IBL)
$N_{SplitHits}$	Number of split hits in the pixel layers (including the IBL; split hit = hit is identified as being created by multiple charged particles during ambiguity solver stage at pattern recognition level)
$N_{SCTHits}$	Combined number of hits in the SCT layers (since 2 strip hits are required for a full SCT spacepoint, this number is divided by two in the track selection)
$N_{SharedSCTHits}$	Number of shared hits (contributing to the track fit and to another track) in the SCT layers (since 2 strip hits are required for a full SCT spacepoint, this number is divided by two in the track selection)

Most of these features are entailed and have some correlations, which will ultimately allow a better NN identification as they will very much complement and fortify each other. Hence, such features will

Table 4.2: List of JetFitter+SV1 related features used during DL1 versions and Umami taggers training.

JetFitter+SV1 Features	**Description**
JF+SV N_{Tracks}	Number of tracks associated to secondary vertex
JF+SV mass	Invariant mass of tracks associated to secondary vertex
JF+SV E	Energy of charged tracks associated to secondary vertex
JF+SV E_{frac}	Fraction of charged jet energy in secondary vertex
JF+SV L_{xy}	Transverse displacement of the secondary vertex from primary vertex
JF+SV L_{xyz}	Distance of the secondary vertex from primary vertex
JF+SV η_{trk}^{max}	Maximum track relative η
JF+SV η_{trk}^{min}	Minimum track relative η
JF+SV η_{trk}^{aver}	Average track relative η
JF+SV η_{jet}^{max}	Maximum jet relative η
JF+SV η_{jet}^{min}	Minimum jet relative η
JF+SV η_{jet}^{aver}	Average jet relative η

increase the NN performance.

By analyzing the jet feature correlations between 1 million jets from the training samples, for pp and Pb+Pb, some interesting results can be seen in figures 4.1 and 4.2, respectively for pp and Pb+Pb. These diagrams are very much similar, between each other, and show two distinct sets of correlations, one related to SV1, JetFitter and JetFitter+SV1 features and another related to impact parameter algorithms. Some correlations such as a clear anti-correlation between η - p_T and SV1 E_{frac} - p_T variables can be seen, besides the anti-correlations between IP2D$_{bc}$ - IP2D$_{cu}$ and IP3D$_{bc}$ - IP3D$_{cu}$. These results are explained by the particle jet physics that sustains those anti-correlation behaviors. For η, as the p_T increases, for energy and momentum conservation, the particle will be limited into a certain emission angle which explain the $|\eta|$ decrease. This effect is responsible for the η decrease as the jet p_T increases, shown in figure 4.3.

As per tracks, the results obtained for 200k jets of the training sample are presented in figure 4.4, for pp and Pb+Pb. Again, a similar correlation is observed for both data types centered around the inner detector measured quantities, associated to the number of pixels transposed, with the only exception residing on the IP3D impact parameters significance that are directly correlated, once both share the same detector resolution, and on a few inner detector properties that are anti-correlated to the Δr.

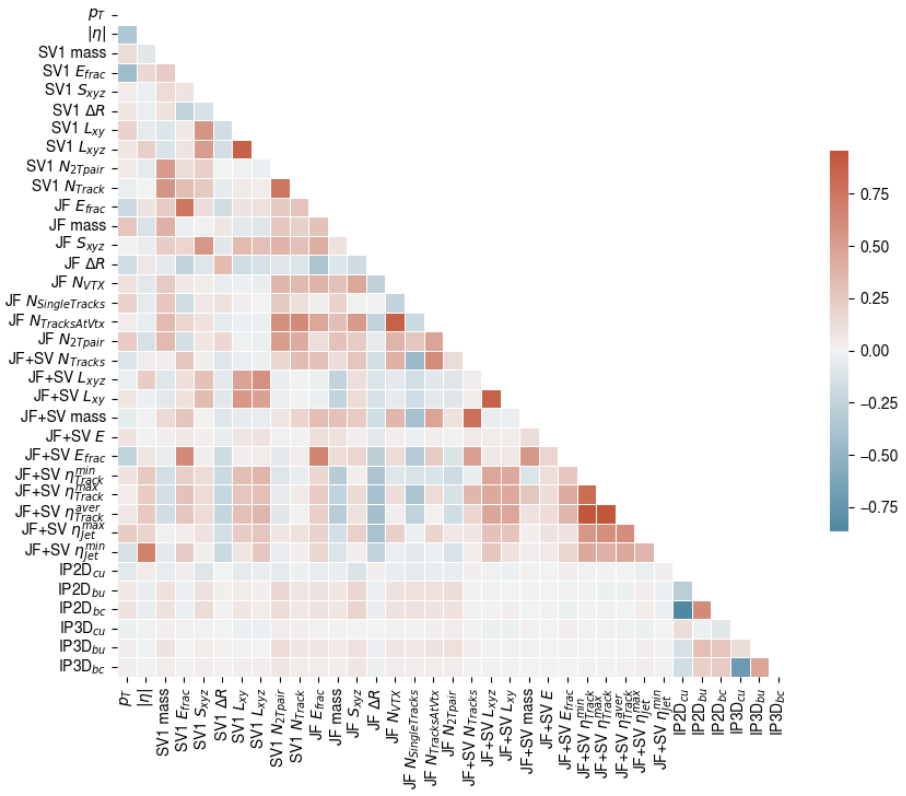


Figure 4.1: pp jet feature correlations obtained for 1 million jets from the train sample. The red color means large correlation whilst blue means large anti-correlation.

Even though the features correlation is a good hint of the impact of the training features on the NN performance, it may not prove to be enough, meaning that another method may be needed to do such conclusions.

So, to give answer to the questions related to which variables may or may not increase the NN performances and to how different can the features discriminant power be dependent on the collisions data, another relevant study that might be interesting to see regards the contribution that each feature gives to the neural network discriminant value calculated in section 2.4.

By training each NN with the features used by default on UMAMI, it is possible to construct a flavor-tagging discriminant variable, equal for DIPS and DL1 and presented in equation 4.1 for a general k-flavor tagging, with k-, l- and m-flavor being the b-, c- or u-flavor jet depending on the flavor-tagging that one is interested. For b-tagging, the discriminant is represented as in equation 4.2 and relates each NN outputted flavor probability (for each jet inputted) in order to construct a single value, function of the c-jet fraction (f_c). Such variable helps to understand how confident a NN can be when identifying a b-jet, by ideally

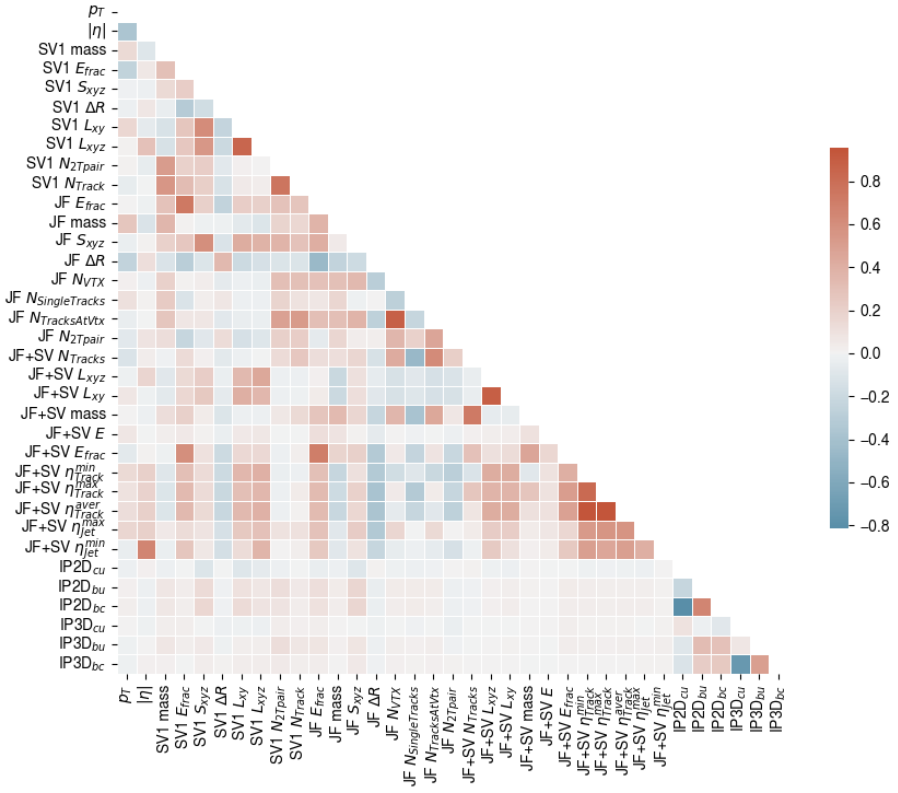
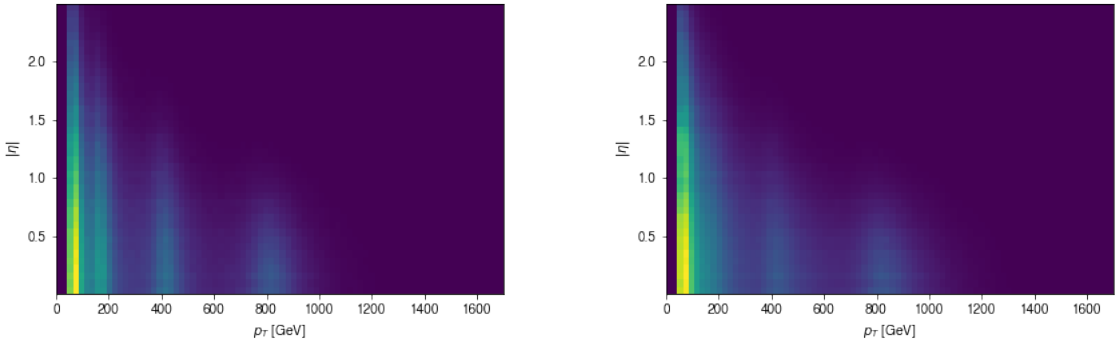


Figure 4.2: Pb+Pb jet feature correlations obtained for 1 million jets from the train sample. The red color means large correlation whilst blue means large anti-correlation.

tending to $+\infty$ when a jet is identified with a 100 % probability of being a b-jet, and tending to $-\infty$ for c-jets and u-jets. However the NN is not perfect and it never gives a probability of 100 %, meaning that a b-jet will have a high probability of being a b-jet and a residual probability of being a c-jet and a u-jet that can be much different from jet to jet. Figure 4.5 shows the b-tagging discriminant distribution for the DL1 algorithm. Besides the significant difference between figures 4.6a and 4.6b, three separate areas can be seen where, on average, the b-jets have the bigger discriminant values and the u-jets the lower ones, as expected from the b-tagging discriminant definition. The differences between both distributions are related to the underlying event contribution.

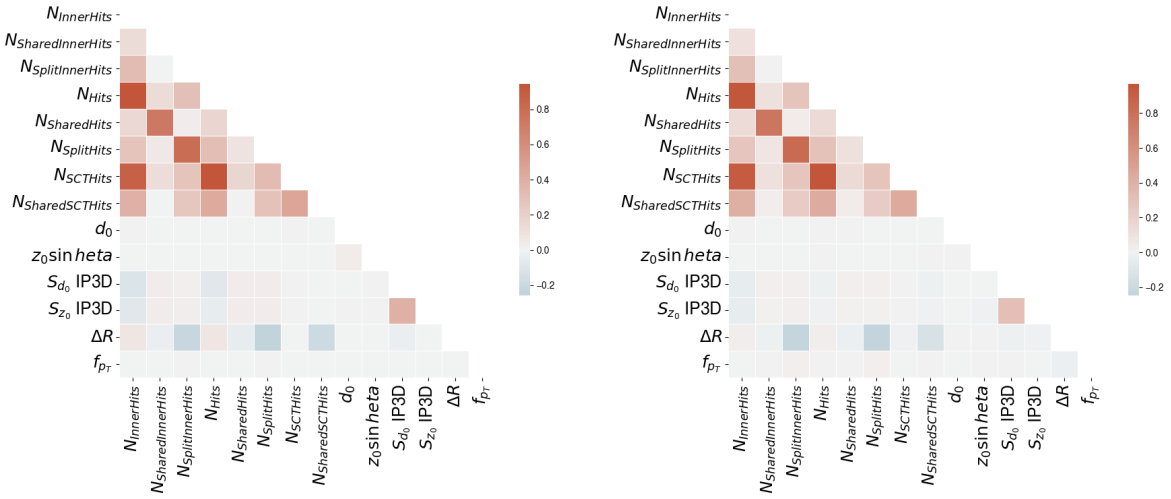
$$D_{k\text{-}flavor\ tagging} = \log\left(\frac{p_{(k\text{-}flavor)}}{f_{(l\text{-}flavor)} \cdot p_{(l\text{-}flavor)} + \left(1 - f_{(l\text{-}flavor)}\right) \cdot p_{(m\text{-}flavor)}}\right) \tag{4.1}$$

$$D_{b-tagging} = \log\left(\frac{p_b}{f_c \cdot p_c + (1 - f_c) \cdot p_u}\right) \tag{4.2}$$



(a) (b)

Figure 4.3: (a) pp and (b) Pb+Pb $|\eta|$ distribution as a function of the jet p_T for 2.6M and 5M train sample jets, respectively.



(a) (b)

Figure 4.4: (a) pp and (b) Pb+Pb track correlations for 200 thousand jets from the train sample. The red color means large correlation whilst blue means large anti-correlation.

The b-tagging discriminant, is a great tool to evaluate how much each variable influences the NN performance. As already observed this variable is entailed with all NN outputs, meaning the flavor-jet probabilities, and therefore the gradient of the b-tagging discriminant in order to each feature will represent the NN performance impact of such a feature. Besides, this study can be made for both track

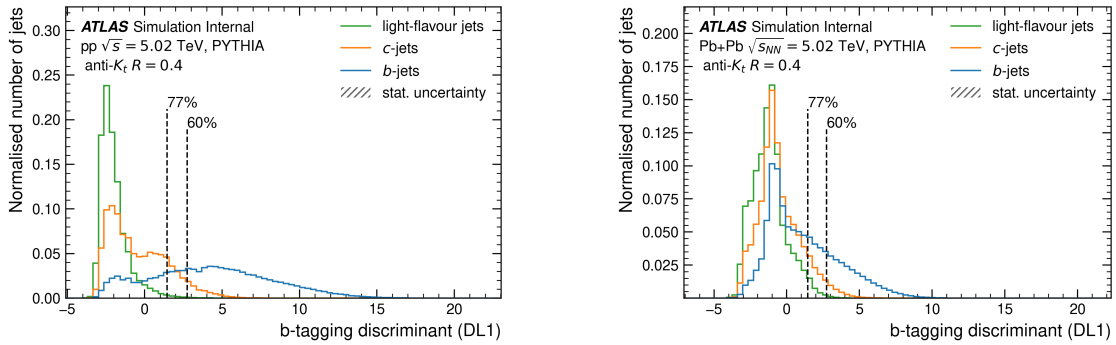


Figure 4.5: B-tagging discriminant distribution on (a) pp and (b) Pb+Pb. Both 60 % and 77 % b-tagging efficiencies are plotted.

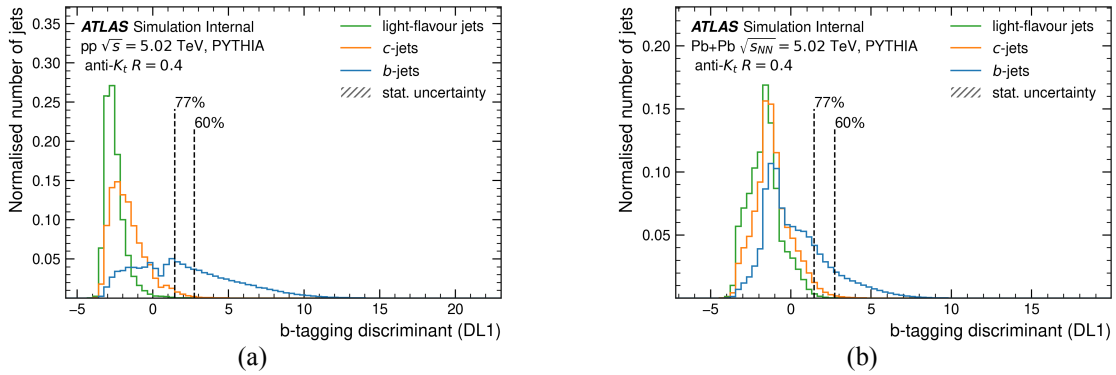


Figure 4.6: B-tagging discriminant with no denominator weights on (a) pp and (b) Pb+Pb. Both 60 % and 77 % b-tagging efficiencies are plotted.

and jet features, since the DL1 algorithm makes use of jet features and the DIPS algorithm makes use of track features. Once the b-tagging discriminant increases with the probability of a jet be identified as a b-jet, its gradients for b-jets show the confidence level of the NN and what training features penalize the NN performance on the identification of b-jets. Nonetheless, for c-jets and u-jets the same analysis can not be implemented, as now the gradient of the discriminant is expected to ideally decrease since the b-tagging discriminant tends to $-\infty$. Thus, for both c- and light-flavor jets the features with a negative b-tagging discriminant gradient are the ones that ensure a better identification of c-jets and u-jets, granted by the decrease in the probability of a jet be considered a b-jet.

The problem related to the b-tagging discriminant is that no conclusion regarding the exact performance of the NN in the identification of c- and u-jets is possible and a c-tagging or u-tagging discriminant

is necessary, since these probabilities are camouflaged when using the expression in equation 4.2. Therefore, instead of using the fraction of each flavor as a weight for the probabilities in the denominator as in equation 4.1, that are not attainable for all flavor-jets once the conducted studies are for b-tagging, a good way to study these other flavor-tagging performances is to use no weight, providing the construction of three analogous expressions serving as a b- c- and u-jet discriminant and represented in equations 4.3. By not using the usual weights the results are expected not to change much, however differences will exist, observed when comparing figures 4.5 and 4.6, as now the non tagging flavor probabilities are equally important. When using these discriminants on each jet-flavor sample, conclusions regarding the track and jet feature impact during the NN training are made.

$$\begin{cases} D_{b-flavor\ tagging} = \log\left(\frac{p_b}{p_c+p_u}\right) \\ D_{c-flavor\ tagging} = \log\left(\frac{p_c}{p_b+p_u}\right) \\ D_{u-flavor\ tagging} = \log\left(\frac{p_u}{p_b+p_c}\right) \end{cases} \tag{4.3}$$

By using the gradient of a jet-flavor discriminant in order to each feature on the same jet-flavor sample, the negative gradient values will represent the variables that decrease the flavor-discriminant, penalizing the jet-flavor identification, and contrarily if its gradient increase the variable improves both the jet-flavor identification and the NN performance for that flavor. These analysis are presented in figures 4.7 and 4.8 where now the negative gradients are represented with a red color and the positive ones with a green color. These results show that, when looking to the b-tagging discriminant, in b-jets, several features increase the b-jet identification whether others decrease it. However, those negative variables are good c- and u-jet discriminators, observed by the c-tagging and u-tagging gradient discriminants, ultimately improving the b-tagging efficiency. Hence, the NN will increase the desired b-tagging performance, identifying the b-jets with the positive b-tagging discriminant gradient features, and separate c-jets from u-jets from the other non b-flavor jets, making use of the positive c- and u-tagging discriminant gradient features. Such gradient discriminant differences are related to the life-time of each flavor hadrons and to the structural decay shower differences that make some of the ID features more beneficial for c- and u-jets, which have a much lower life-time, and causing differences in some jet features.

Prove of the positive impact of the negative gradient b-tagging features, positive for c- or u-flavor tagging gradients, is that when considering the positive b-tagging features only the NN performance is greatly penalized when compared to the train with all features, as can be observed in figure 4.9. The DL1 NN has a 10 % and 5 % increase in u- and c-flavor rejections, respectively, when trained with all features and when compared to the b-tagging features training.

Another relevant conclusion is that for Pb+Pb some pp features change their relative importance for a specific jet-flavor identification but none become useless, as they instead become discriminators of other flavor-jets. For example, the $N_{InnerHits}$ track feature variable is a good u-jet discriminator for pp, and become a good discriminator of b-jets and c-jets for Pb+Pb.

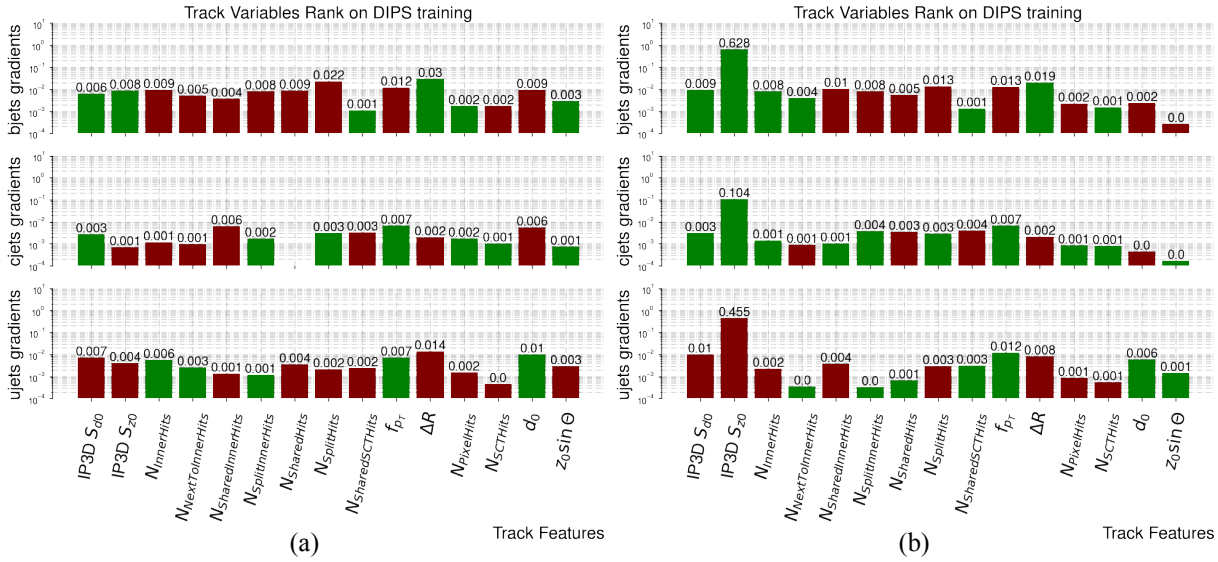


Figure 4.7: (a) pp and (b) Pb+Pb DIPS flavor-tagging discriminant gradient of each track feature for b-, c- and u-jets (top, middle and bottom respectively for each sub diagram) passing a flavor-tagging efficiency of 77% and with exactly 8 valid tracks. For this study, 300k test sample jets were used. The green color mean an increase on the flavor-tagging discriminant gradient value, and the red color mean its decrease.

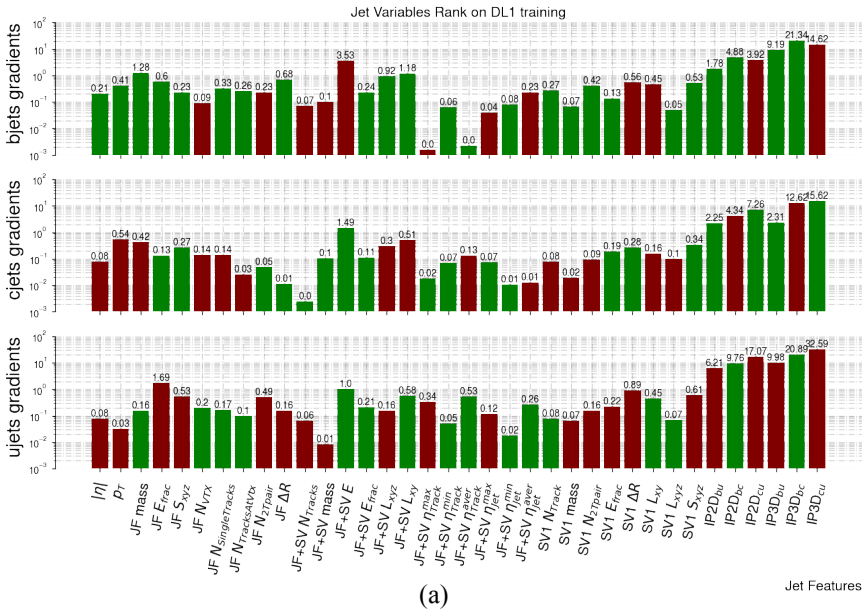
Furthermore, by analyzing the track features rank for pp and Pb+Pb, an outstanding difference could be seen between each data type for the IP3D S_{Z_0}. While for pp a homogeneous distribution is observed, meaning that much of the variables have the same absolute discriminant values, for Pb+Pb one variable has a big absolute discriminant gradient with all others being negligible, around 15 to 37 times more than the medium value of the absolute gradients.

4.2 DL1

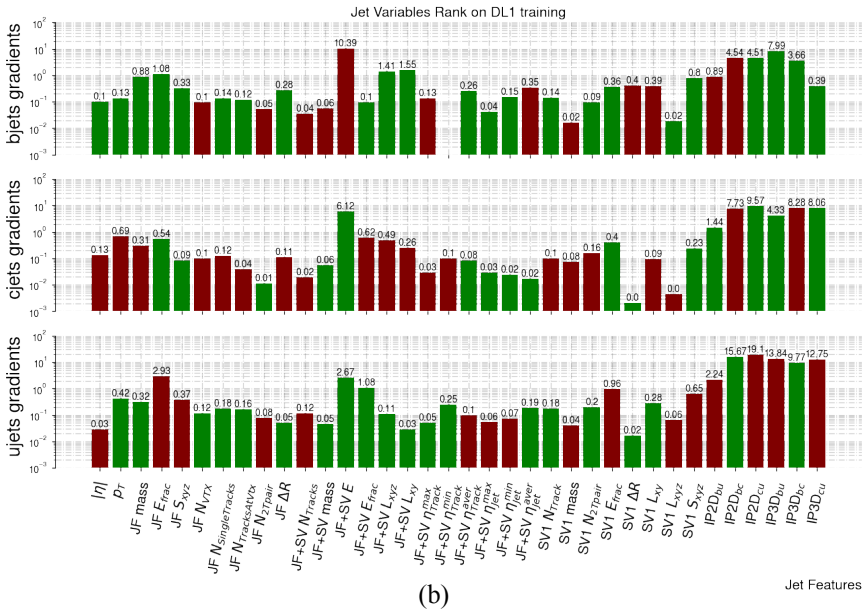
The DL1 neural network is based on jet reconstruction related features that are obtained through tracks, as detailed in the previous sections, being of uttermost importance when there is a b-tagging performance study interest.

As already seen, there are plenty of different DL1 architectures, with the DL1d being the most promising high-level algorithm, however, due to some technical problems related to the AODs download, mediated by ATHENA [The ATLAS Colaboration, 2022a] and pandas [The ATLAS Colaboration, 2022b], it was impossible to obtain data to ensure the utilization of DL1d and DL1r and consequently, the studies on this paper just make use of DL1 baseline and DIPS.

Since this work will evaluate Pb+Pb b-tagging performances by comparing these results to the pp



(a)



(b)

Figure 4.8: (a) pp and (b) Pb+Pb DL1 flavor-tagging discriminant gradient of each jet feature for b-, c- and u-jets (top, middle and bottom respectively for each sub diagram) passing a flavor-tagging efficiency of 77%. For this study, 300k test sample jets were used. The green color mean an increase on the flavor-tagging discriminant gradient value, and the red color mean its decrease.

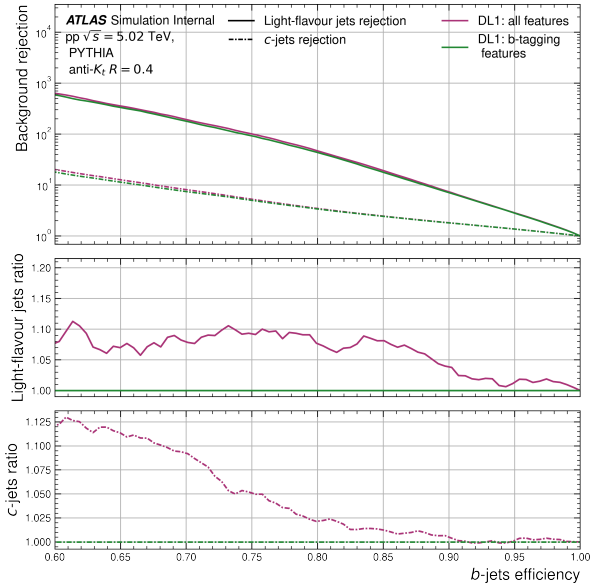


Figure 4.9: ROC plot obtained for pp on DL1 training. This ROC shows two different trainings, DL1: all features with all features and DL1: b-tagging features with all the positive b-tagging gradient features. Two ratio plots are shown for u-jets (middle) and c-jets (bottom), where the reference is considered as the DL1: b-tagging features training results. All results are generated with the test samples, containing 0.6M jets, with the training weights obtained on epoch 300. For this an $f_c = 0.05$ was used.

ones, the very same NN architecture was applied for both data types, with the chosen hyperparameters being dictated by the available studies in pp btagging performances by the ATLAS FTAG group, which are presented in table 2.4 and are the default values used in this package.

Table 4.3: DL1 hyperparamiters used for pp and Pb+Pb data training.

DL1	
Hyperparameters	**Values**
Number of input variables	41
Number of hidden layers	8
Number of nodes [per layer]	[256, 128, 60, 48, 36, 24, 12, 6]
Number of training epochs	300
Learning rate	0.001
Training minibatch size	15000

Is by making use of the very same training features presented previously for both pp and Pb+Pb, which correlations have already been observed, and by using these hyperparameters that the DL1 training

performance studies will work on.

4.2.1 Pb+Pb performance comparison

By training the DL1 NN with the above specifications for pp and Pb+Pb quite some comparison studies can be done, where the pp sample results are used mainly as reference.

Starting by the analysis of the NN training performance, by making a training process over each DL1 algorithm recursion or epoch where the training weights are calculated with the training sample and, consequently, used to predict the jet flavor of the validation sample, the figure 4.10 can be obtained. This figure has two different plots regarding the training NN performance, one for each data type, where the training loss plots, referent to the amount of mistag flavor jets using the categorical cross-entropy function, and the training accuracy plots, calculated through the total number of correct identified flavor-jets, are observed. Results show that the pp data collisions training offers a much better performance when compared to the Pb+Pb data, for both the training loss and accuracy have worse values. Whilst for pp the lower number of collision generated particles provide less measurement errors in tracking, for Pb+Pb this number is several orders bigger leading to worse resolution. Additionally, due to the huge underlying event contribution on Pb+Pb, the jet features are more contaminated with fake tracks, justifying tight jet and track p_T variable cuts.

These plots show that the NN is not only predicting correctly the results with a certain accuracy, but is increasing its performance on each epoch. Besides no overfitting is shown, otherwise the validation curve would start to decrease/increase it's accuracy/loss and the model would start to have the exact same training sample fluctuations losing the ability to predict results consistently. Underfitting is also not present, for which case the validation wouldn't have enough recurrences or epochs to stabilize at a certain value.

The test sample is used to evaluate the b-tagging perfomance. The proxy is the Receiver Operator Characteristic curve (ROC curve) shown in figures 4.11 and 4.12 for pp and Pb+Pb collisions simulation, where the first is the reference obtained by the ATLAS Flavor Tag group and the latter the obtained straight forward from the previous training performance plots. Both the c- and u-jet rejections are much higher on pp than on Pb+Pb, with a clear higher rejection for u-jets explained by the similarity between b- and c-jets. For example, at $\varepsilon_b = 75\,\%$, about 4.7 and 1.7 times more light-flavor and c-flavor jets are rejected on pp than on Pb+Pb.

Another important and complementary study is the study of the dependence of the flavor jet rejection on jet p_T, which is summarized in figure 4.13. Both the c- and u-jet rejection vary with the jet p_T in such a way that the former one monotonically increases and stabilizes, with the exception of Pb+Pb where the underlying event contribution is responsible for the low jet p_T spike, and the latter first increases and then decreases. The reason for this behavior is that most of the jet features used in the training are very much jet p_T dependent, in such a way that for u-jets most of the variables will loose their discriminant power as the jet p_T increases, meaning that lower u-jet rejections will be attainable. On the other hand,

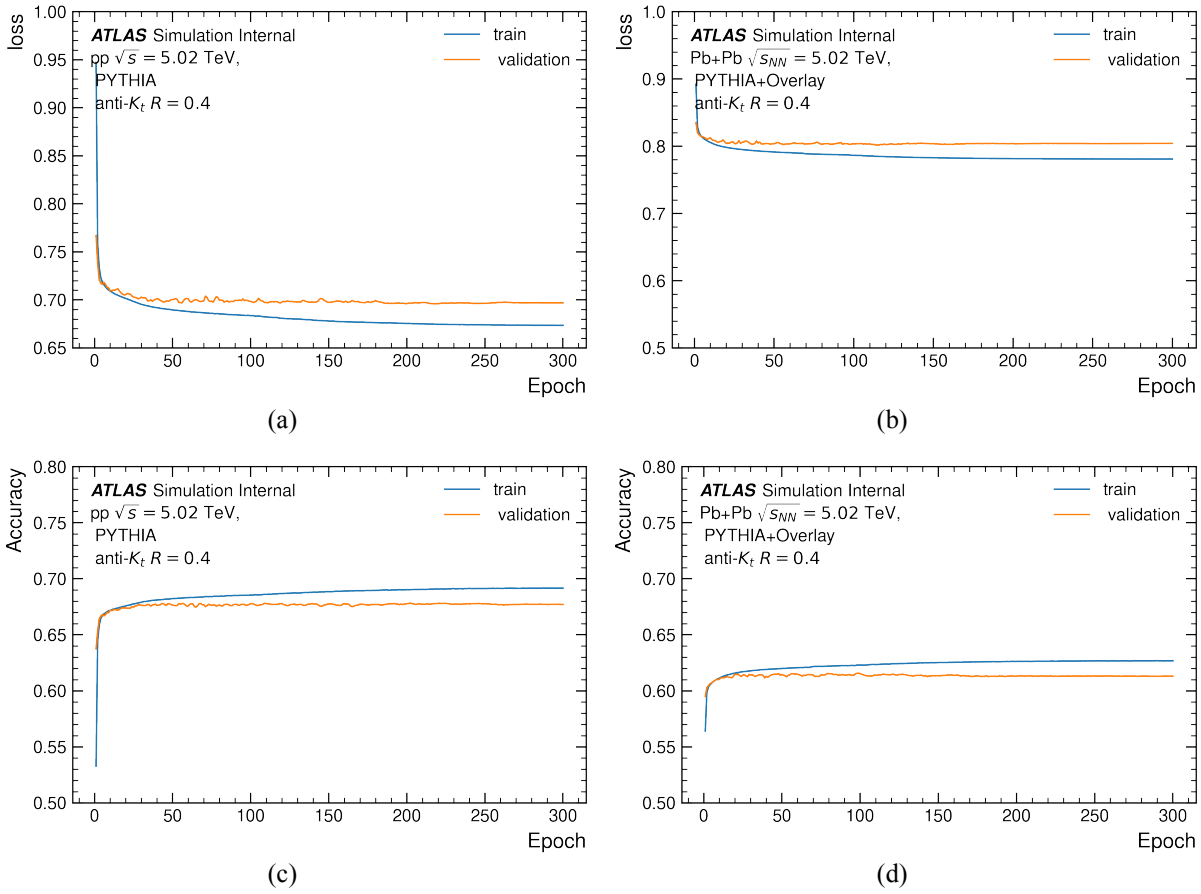
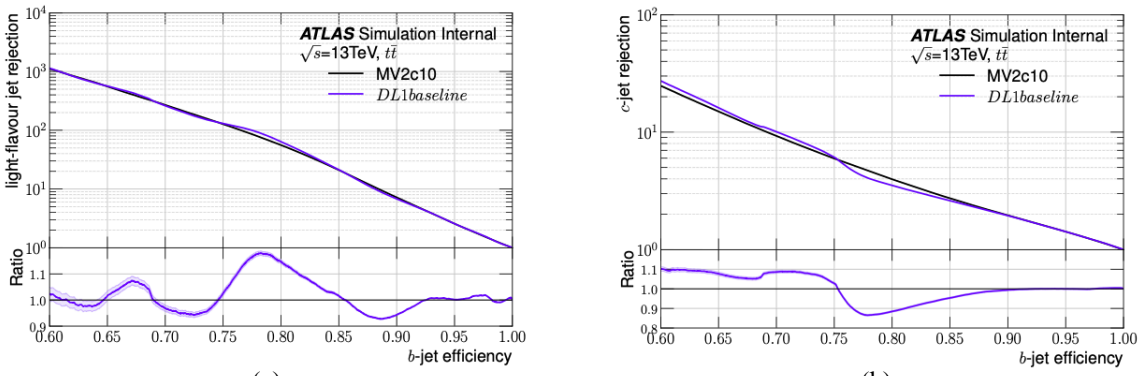


Figure 4.10: DL1 training loss for pp (a) and Pb+Pb (b) and accuracy for pp (c) and Pb+Pb (d). There are two curves on each graph, where the blue one is the training sample, with 2.6M and 5M jets for pp and Pb+Pb respectively, and the orange one is the validation, with 300k jets for both pp and Pb+Pb. Each sample have the very same preprocessing treatment, meaning that both were downsampled.

for c-jets these variables are not likely affected, as can be seen in figure 4.14 for both pp and Pb+Pb. When comparing the b- and c-jets variable ranks with the u-jet ones, a bigger impact on u-jet variables is observed with the jet p_T increase, as the IP2D$_{bc}$ and IP3D$_{bc}$ variables loose much of their positive discriminant power and the other IP2D and IP3D variables having a much stronger negative effect for this specific flavor. This behavior is not observed so intensely in other flavor-jets.

4.2.2 Pb+Pb centrality performance studies

While on pp collisions only two nucleons interact, on Pb+Pb there are 208 nucleons colliding with 208 nucleons, meaning that, in this case, there are simultaneous collisions (within the time scale of the



(a) (b)

Figure 4.11: ATLAS Flavor Tag (a) light-flavor jet rejection and (b) c-jet rejection in proton collisions for $t\bar{t}$ events. In this case the results are obtained using the DL1 baseline algorithm and a $f_c = 0.10$.

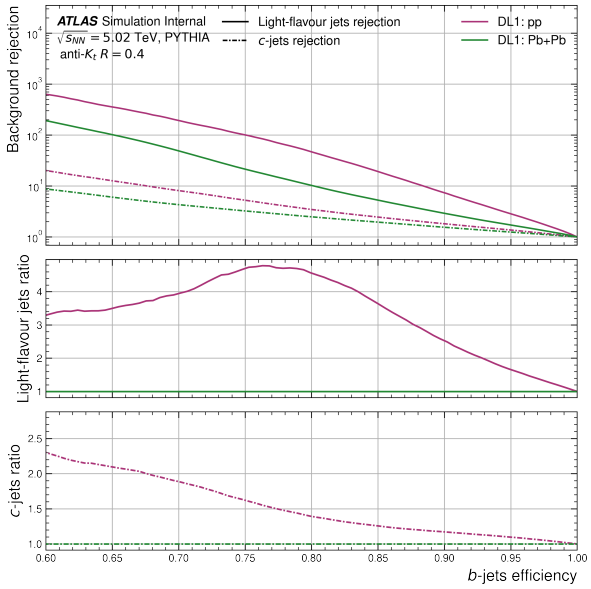
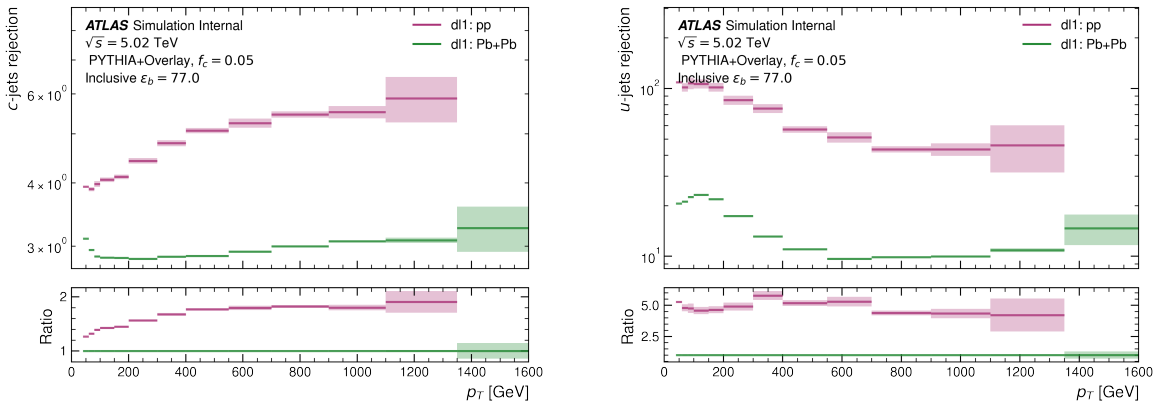


Figure 4.12: ROC plot obtained for DL1 training. Two ratio plots are shown for u-jets (middle) and c-jets (bottom), where the reference is considered as the Pb+Pb training results. All results are generated with the test samples, containing 0.6M and 5M jets for pp and Pb+Pb respectively, with the training weights obtained on epoch 300. For this an $f_c = 0.05$ was used.

data processing) that can take place, where all nucleons or a fraction can collide. This characteristic associated to the Pb+Pb collisions will generate different data, and therefore, change the training performance associated to the NN in question depending on which features its train is based on.

As DL1 makes use of the jet features, the collision centrality is expected to affect the training per-



(a) (b)

Figure 4.13: (a) c- and (b) u-jet rejection distribution as a function of the jet p_T for pp and Pb+Pb test samples, with 0.6M and 5M jets respectively. For this an inclusive value of $\varepsilon_b = 77\,\%$ and an $f_c = 0.05$ were used.

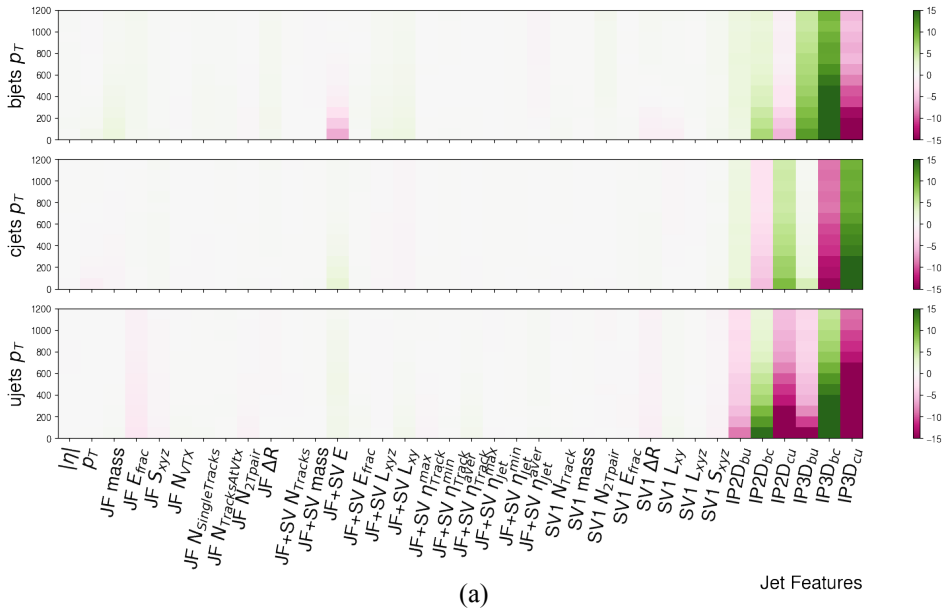
formance. The variable used as a proxy for collisions centrality is the transverse energy deposited in the forward calorimeter (E_T^{FCal}). This variable is influenced by the collision centrality with the larger the energy deposition, the larger the nuclear overlap [G. Aad et all, 2012]. Thus, to study this dependence, a new variable/feature, intrinsic to the event itself, is used associated to the forward calorimeters total transverse energy deposition (E_T^{FCal}) which is divided into percentiles where $0\,\%$ corresponds to an $E_T^{FCal} \to \infty$ and refers a central collision, and $100\,\%$ corresponding to $E_T^{FCal} = 0$ and referring a peripheral collision [G. Aad et all, 2012].

Pb+Pb collisions data were divided in three centrality ranges shown in table 4.4, where the 0 - 20 %, 20 - 50 %, 50 - 80 % are the central, semi-central/semi-peripheral and peripheral centrality samples, that were submitted to the very same sample preprocessing explained in section 3.3 with the additional variable cuts presented in table 4.4.

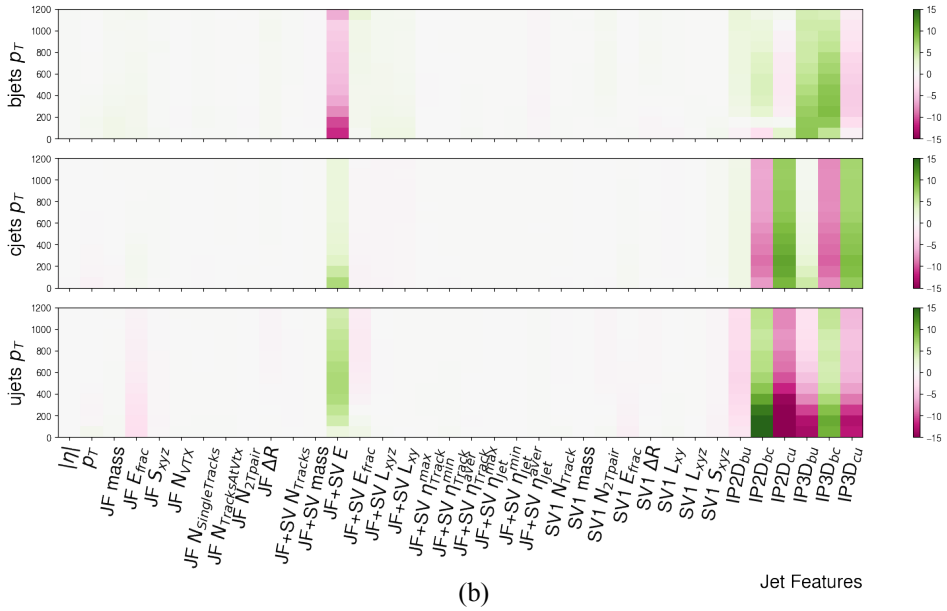
Table 4.4: Samples generated for Pb+Pb in order to study the collision centrality impact during the NN training.

Percentile range	$\mathbf{E_T^{FCal}}$ **(TeV)**
0 % - 20 %	∞ - 2.05
20 % - 50 %	2.05 - 0.53
50 % - 80 %	0.53 - 0.06

Figure 4.15 shows the training Loss and Accuracy as a function of the number of epoch for the three centrality ranges, in this case for the DL1 algorithm, with no overfitting and underfitting. These results show that a clear difference is observed when dealing with different centrality samples and that the performance tend to decrease as the centrality increase, consequence of the underlying event influence.



(a)



(b)

Figure 4.14: (a) pp and (b) Pb+Pb jet features rank vs jet p_T for each flavor-jet. The color base values are referent to each jet feature contribution on the DL1 flavor-tagging discriminant gradients presented in equations 4.3, just as on section 4.1, with the top subfigure related to the b-tagging, the middle one to the c-tagging and the bottom one to the u-tagging. The green colors mean a positive discriminant contribution and overall training performance increase and the pink colors, the opposite.

Such behavior is very much expected since as the centrality increase so does the number of particles generated. The detector resolution cannot cope efficiently with the larger multiplicities and the number of fake tracks and fake jets increases distorting the set of features used in the training. The consequence is the degradation observed. Some of the variables will have more than one track or particle contribution, originating none physical data or false tracks, which will be used at the NN and will corrupt the learning process.

Figure 4.16 shows the ROC curves for pp and the three Pb+Pb centrality ranges. These results show that the Pb+Pb peripheral test sample almost reach the pp rejections, sustained by the different tracking setups used on Pb+Pb data collisions. In fact, for a b-jet efficiency of $75\,\%$ about 2.1, 4 and 8.6 times more u-jet are rejected and about 1.1, 1.3 and 1.9 times more c-jet are rejected for Pb+Pb semi-central, Pb+Pb peripheral and for pp collisions, respectively when compared to the Pb+Pb central collisions.

As for the jet p_T dependency study, presented in figure 4.17, the same performance hierarchy can be seen with better jet-flavor rejections as the centrality decreases, but now, whilst the same u-jet rejection distributions are observed for both pp and Pb+Pb, the c-jet rejections have a clear different distribution for the central collisions. For this last a monotonic c-jet rejection decrease with p_T is observed (for $p_T \geq 100\,\mathrm{GeV}$), tending into a constant. While for u-jet rejections the same pattern is observed on all centralities, on c-jets rejection both semi-central and peripheral samples have the same distribution as on pp, with the exception of the central sample where a inverted behavior could be seen and the rejection decrease with the jet p_T. Again, such tendency is explained by the jet features vs p_T dependence that for central collisions is dominated by the JF+SV Energy which starts by providing a good c-tagging discrimination that rapidly vanishes with the jet p_T increase, shown in figure 4.18.

4.2.3 $\mathbf{E_T^{FCal}}$ variable training

As a consequence of the training performances centrality impact, the idea of training the DL1 with the E_T^{FCal} feature might prove beneficial, by inputting such dependencies in the train.

To answer such a question, first a variable rank, much likely the ones presented in section 4.1, may prove handy showing E_T^{FCal} train impacts on the overall DL1 discriminant. Therefore, in order to understand not only the variable training impact, but how well does it perform in several centralities a E_T^{FCal} rank plot was made for an inclusive E_T^{FCal} sample and for each one of the previous E_T^{FCal} samples, being presented in figure 4.19. From this analysis the E_T^{FCal} might improve the train performance and thus the b-tagging performance, as it provide a better b-jet discrimination while decreasing the other jet-flavor identification. Once this variable increase the b-tagging performance at other jet-flavors cost, the NN is expected to be capable of increasing the b-jet identification while decreasing the c- and u-jet discrimination, which will ultimately make the NN to mistag c-jets as u-jets and u-jets as c-jets. So the E_T^{FCal} feature does increase the learning performance a bit, and prove beneficial for the inclusive Pb+Pb sample, increasing on average $2.5\,\%$ and $2\,\%$ the u- and c-flavor rejections respectively for b-tagging efficiencies bigger than $80\,\%$, as can be seen from the ROC in figure 4.20. The other Pb+Pb collision

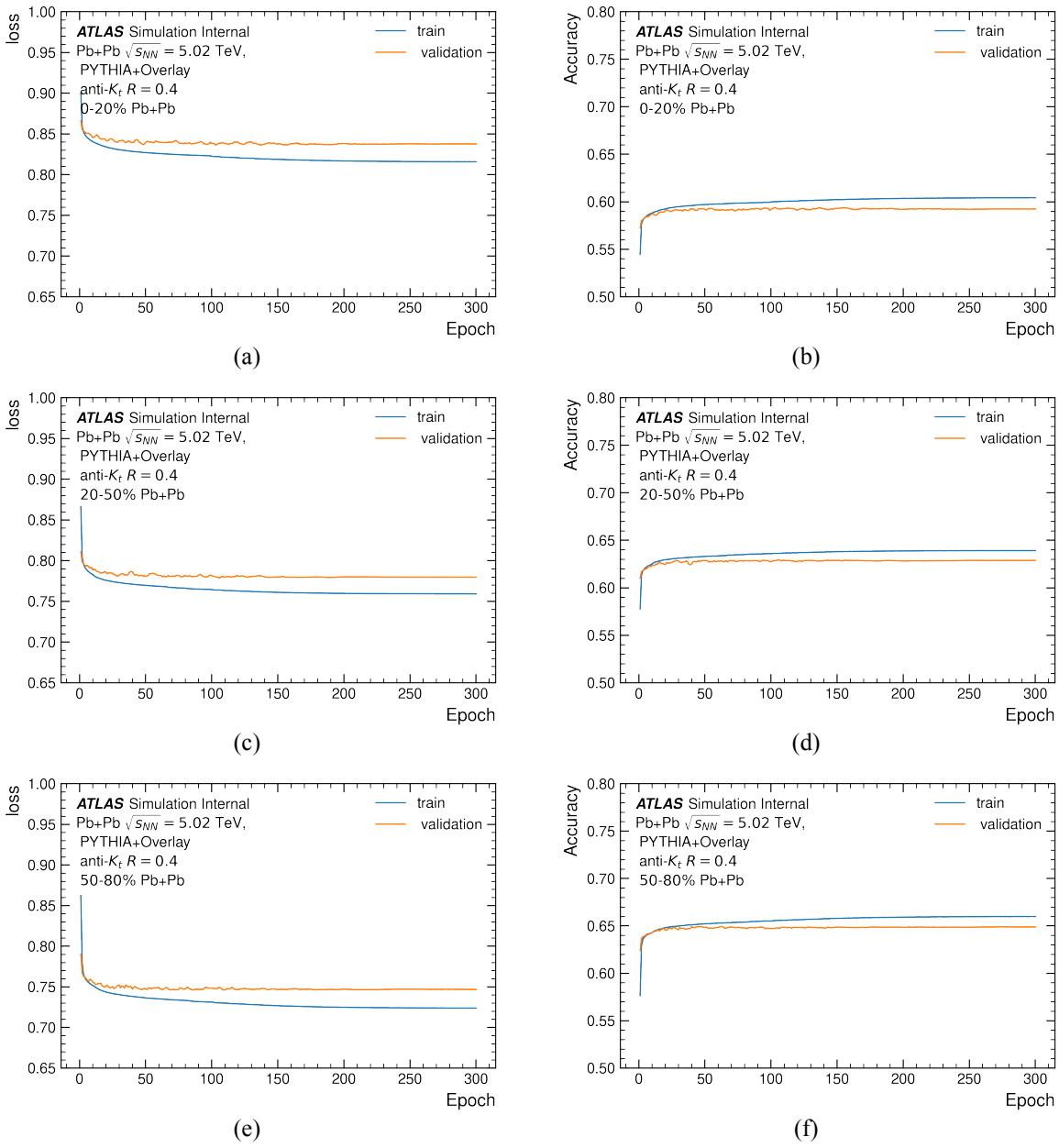


Figure 4.15: DL1 training loss for central (a), semi-central (c) and peripheral (e) centralities and accuracy for central (b), semi-central (d) and peripheral (f) centralities. There are two curves on each graph, where the blue one is the training sample (with 5M, 5M and 3.6M jets for central, semi-central and peripheral samples respectively) and the orange one is the validation (with 300k jets for all samples). Each sample has the very same preprocessing treatment, meaning that both were downsampled.

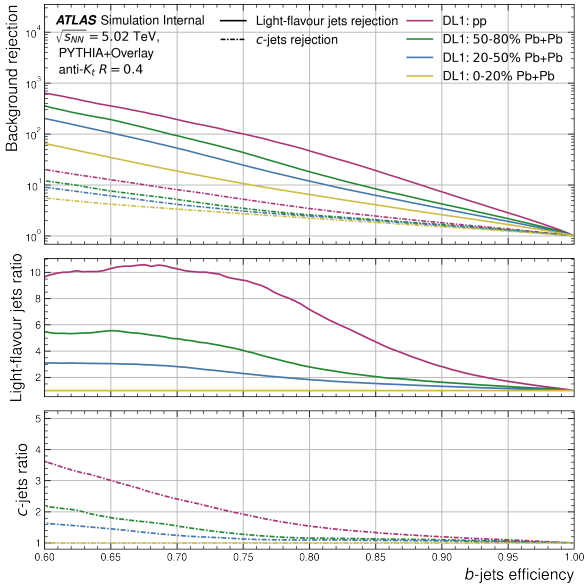


Figure 4.16: ROC plot obtained for central, semi-central and peripheral Pb+Pb collisions on DL1 training. Two ratio plots are shown for u-jets (middle) and c-jets (bottom), where the reference is considered as the central Pb+Pb training results. All results are generated by the test samples (with 2.7M, 1.6, and 0.9M jets for central, semi-central and peripheral samples respectively) and by making use of the training weights obtained on epoch 300. In this study, an $f_c = 0.05$ has been used.

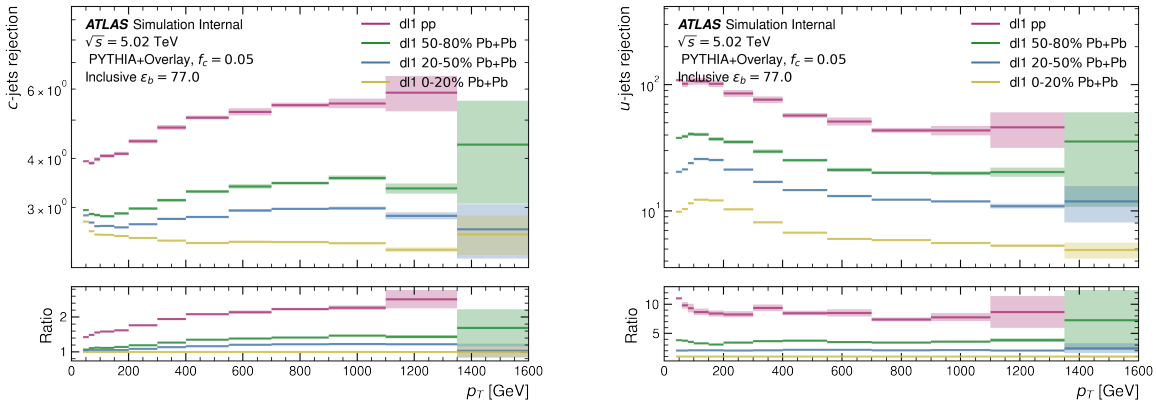


Figure 4.17: (a) c- and (b) u-jet rejections as a function of jet p_T. Two ratio plots are shown for u-jets and c-jets, where the reference is considered as the central Pb+Pb training results.

centrality samples, might have the same behavior, which in this work was impossible to study due to the lack of statistic in this samples.

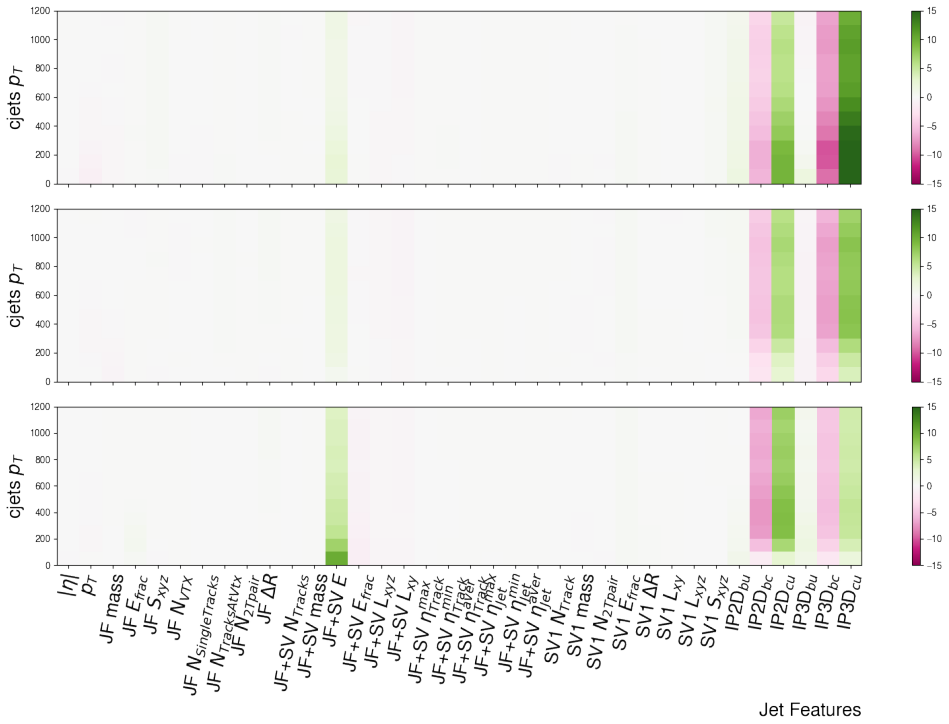


Figure 4.18: Pb+Pb peripheral, semi-central and central (top, middle and bottom, respectively) jet features rank and their jet p_T dependency for c-jets. The color base values are referent to each jet feature contribution on the DL1 c-tagging discriminant gradient presented in equations 4.3, just as on section 4.1. The green colors mean a positive gradient discriminant and overall training performance increase and the pink colors, the opposite.

4.3 DIPS

As for track study, the best study is obtained by DIPS low-level algorithm, that as explained in section 2.4, allows the exploiting of several other tracking features that are not only related to the impact parameters, but are also directly related to the particle interactions, measured in the inner detector.

For this NN, in order to define the hyperparameters that increase the NN learning process and b-tagging performance, a study regarding the training performances over different sets of hyperparameters was conducted for both pp and Pb+Pb.

Starting from the hyperparameters used at the Umami package for DIPS, by varying the learning rate and minibatch size while keeping the other parameters fixed, several train performance plots were produced (presented in appendix 6) which results are shown in figure 4.21.

Several conclusions regarding the hyperparameter sets can be taken, where for a higher learning rate a lower/higher train sample loss/accuracy is obtained and for a lower learning rate the same but for the validation sample. As the objective is to obtain the best results with an arbitrary sample different from

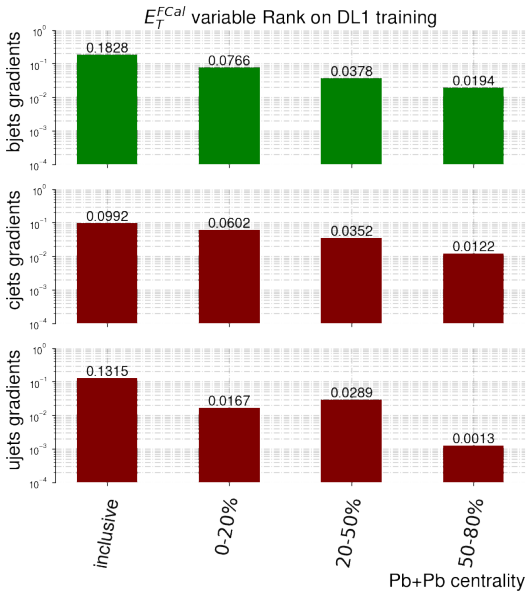


Figure 4.19: DL1 discriminant gradients in order to the E_T^{FCal} variable for b-, c- and u-jets (top, middle, and bottom respectively) passing a flavor-tagging efficiency of 77 % for the previous centrality samples and inclusive E_T^{FCal} sample. During this study, 1M test sample jets were used. The color base values are referent to each jet feature contribution on the DL1 flavor-tagging discriminant gradients presented in equations 4.3, just as on section 4.1, with the top subfigure related to the b-tagging, the middle one to the c-tagging and the bottom one to the u-tagging. The green colors mean a positive discriminant contribution and overall training performance increase and the pink colors, the opposite.

the one used for training, the hyperparameters that maximize the validation parameters are the ones that interest the most, which are related to the lower learning rate and minibatch size.

However, more can be said about those results that can't be seen from just a simple table. Thus, a few training performance plots are presented in figure 4.22, where the rest could be observed in the Appendix B. From the training loss plots produced by a learning rate of 0.001 and 0.0001 with the same minibatch size of 5000 for pp and Pb+Pb, a typical overfitting distribution can be seen, for both data types, when using a higher learning rate (figures 4.22a and 4.22c). In fact this distribution can be seen through out all trains with a learning rate of 0.001, for both pp and Pb+Pb, and are the main reason that justify the use of a learning rate of 0.0001, which have a much smoother distribution reaching a desired plateau. Thus, for DIPS the hyperparameters chosen for pp and Pb+Pb are the ones that maximize the validation results, which are summarized in table 4.5, and have the same values for both data types.

4.3.1 p_T track cut study

By training this NN using different track p_T cut samples, several conclusions related to the track variables training and DIPS tagger performances can be taken for Pb+Pb with pp as reference.

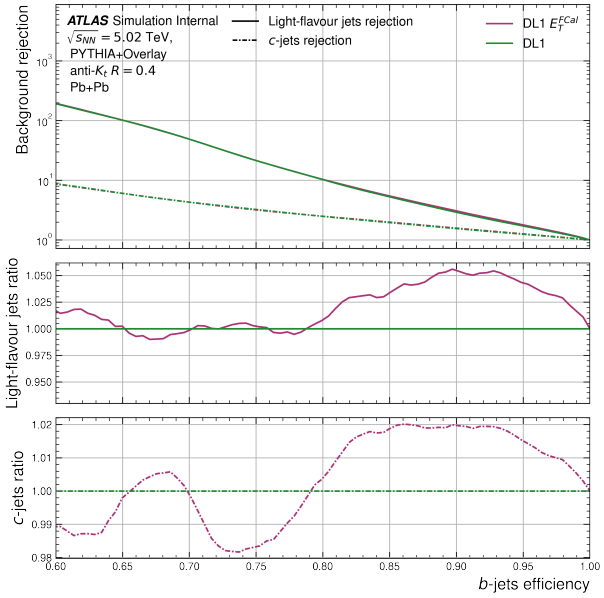


Figure 4.20: ROC plots obtained for inclusive Pb+Pb collisions on DL1 training when including the E_T^{FCal} as a train feature. The ROC is copulated with a ratio plot for both u-jets and c-jets, where the reference is considered as the DL1 base results. All results are generated by the test samples (with 5M jets) and by making use of the training weights obtained at epoch 300. In this study, an $f_c = 0.05$ has been used.

Table 4.5: DIPS hyperparameters used for pp and Pb+Pb data training.

DIPS	
Hyperparameters	**Values**
Number of input variables	15
Number of Φ hidden layers	3
Number of Φ nodes [per layer]	[100, 100, 128]
Number of F hidden layers	4
Number of F nodes [per layer]	[100, 100, 100, 30]
Number of training epochs	300
Learning rate	0.0001
raining minibatch size	5000

In addition to the already mentioned samples, another two samples were generated for each data type, where the 1 GeV and 2 GeV minimum track p_T cuts were applied during the sample selection on the construction of each AOD enumerated in section 3.2

The training loss performances presented in figure 4.22, obtained using the final hyperparameters in

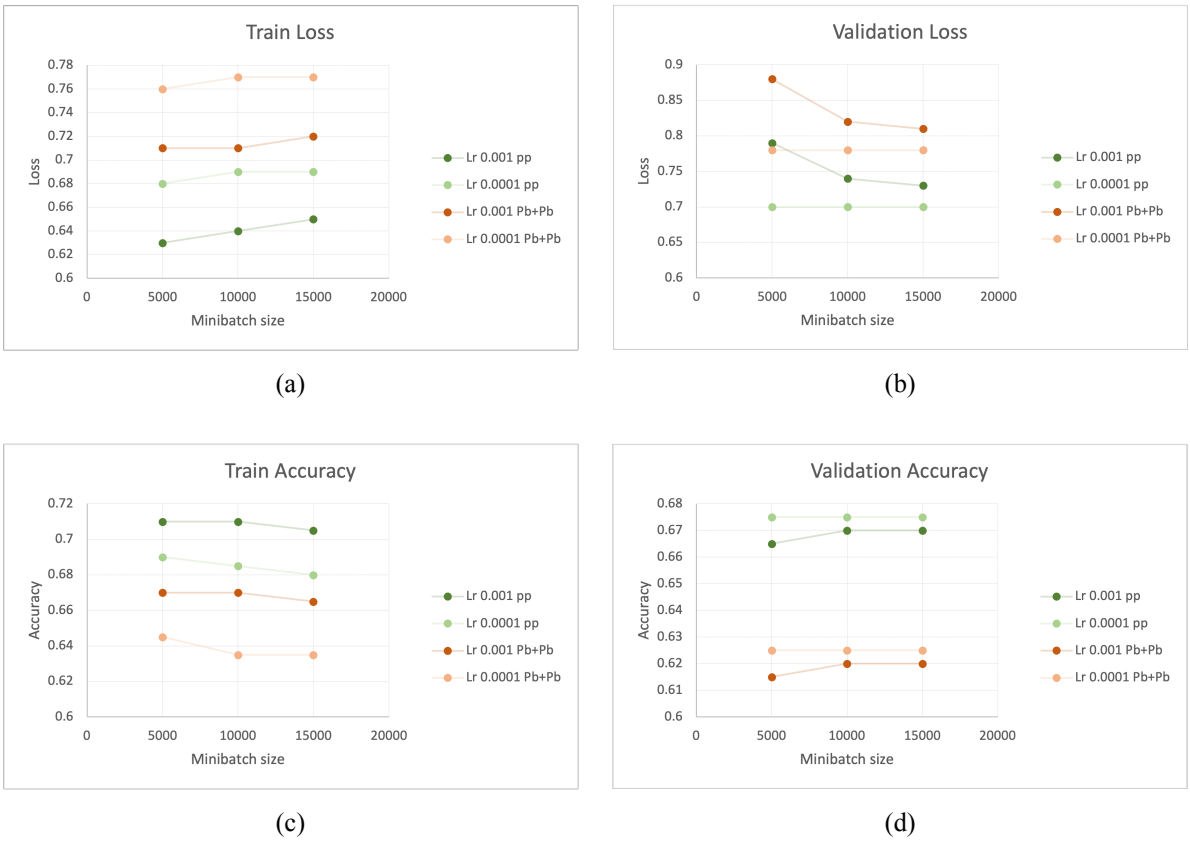


Figure 4.21: pp and Pb+Pb train loss (a), validation loss (b), train accuracy (c) and validation accuracy (d) values obtained in epoch 300 for a 2 GeV track p_T cut, while changing the learning rate and the minibatch sizes

table 4.5, by default make use of the 2 GeV minimum track p_T cut samples. Thus, by using the same hyperparameters for the 1 GeV minimum track p_T cut samples, the same training performance is expected and the ROC plots in figure 4.23 are obtained. Both pp and Pb+Pb 1 GeV (with a minimum 1 GeV track p_T cut) samples have better performances, when compared to the 2 GeV (with a minimum 2 GeV track p_T cut) samples, justified by the harmful effect of the track p_T cut where the 2 GeV cut is enough to waste relevant track information. However, more conclusions regarding the minimum track's p_T cut can be obtained on the track's p_T cut sample features rank study in figure 4.24. For pp, in figure 4.24a, all positive gradient features get worse with the 2 GeV track p_T cut and most of the negative gradient features increase their negative impact, with these effects being noticed also for c-jets. The exception to this are the u-jets where mainly the positive gradients become negative as the track p_T cut increase. As for Pb+Pb, represented in figure 4.24b, despite a bigger influence in the IP3D$_{z_0}$ which had already been observed, analogous conclusions can be made and for b-, c- and u-jets some features do increase the NN

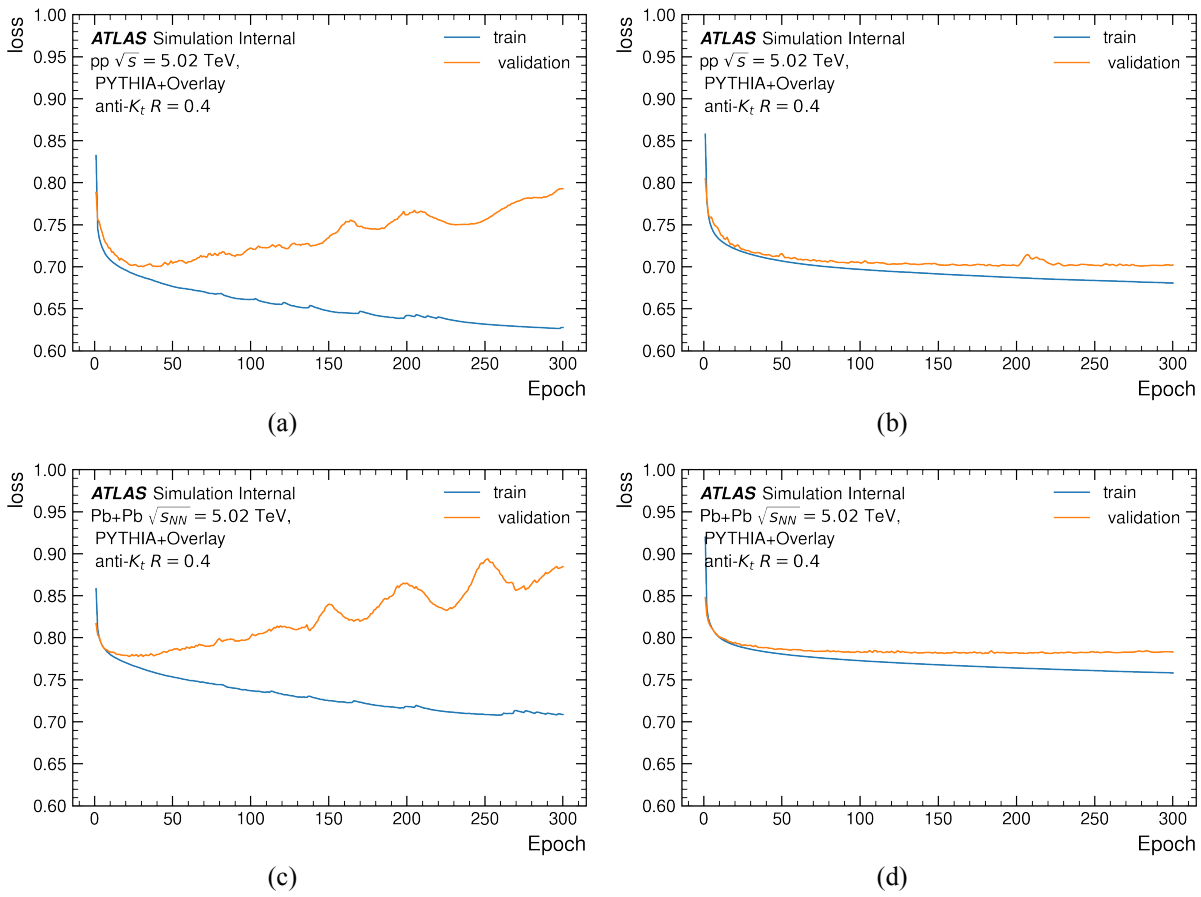


Figure 4.22: DIPS training loss curves when using a learning rate of 0.001, for (a) pp and (c) Pb+Pb, and 0.0001, for (b) pp and (d) Pb+Pb, with a minibatch size of 5000 for train and validation samples.

flavor discrimination, even thought being less noticed when compared with pp.

Another relevant conclusion is that, as for DL1, in DIPS a better performance is achieved for pp, observed in the background rejection values for pp and Pb+Pb in figures 4.16 and 4.23. Conclusions supported by the jet features, taken in DL1, being obtained by the track features that were modified and treated by the low level algorithms just to guarantee a better variable discrimination.

4.3.2 Pb+Pb centrality performance studies

Despite some differences being observed in Pb+Pb $1\,\mathrm{GeV}$ and $2\,\mathrm{GeV}$ track p_T samples, for a given specific E_T^{FCal} centrality this difference could be improved, which can bring advantage during a certain Pb+Pb centrality training.

By analyzing the DIPS behavior for each centrality sample cut with a $1\,\mathrm{GeV}$ and a $2\,\mathrm{GeV}$ track p_T

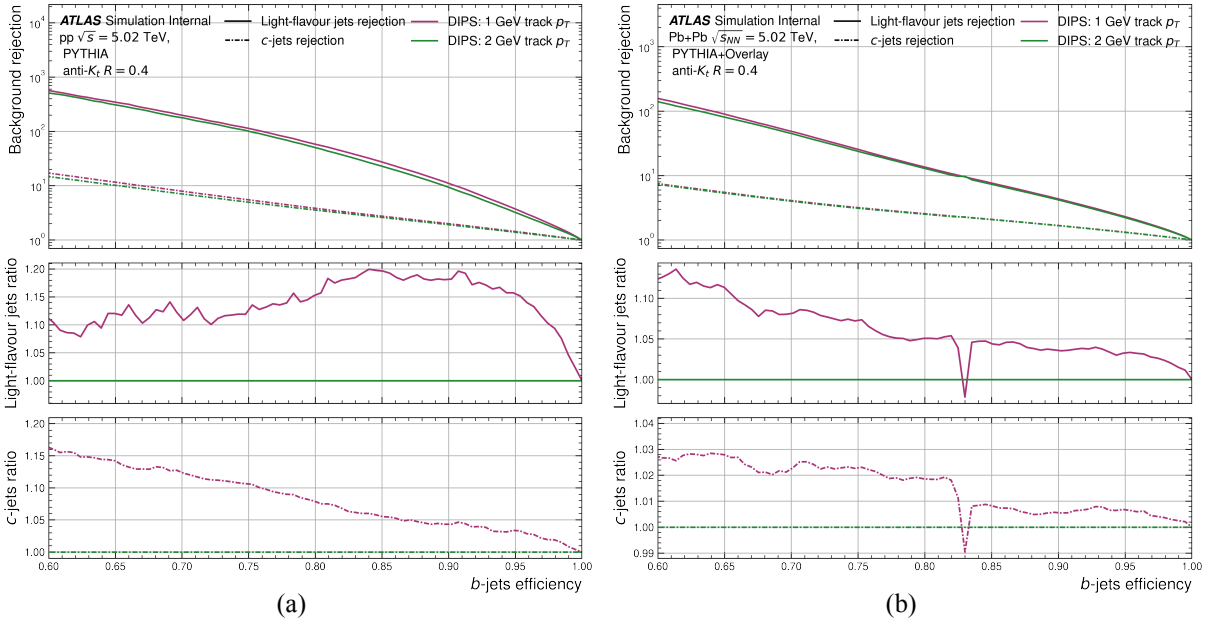


Figure 4.23: ROC plots obtained for (a) pp and (b) Pb+Pb 1 GeV and 2 GeV cut test samples on DIPS training. Each ROC is copulated with two ratio plots, for both u-jets and c-jets, where the reference is considered as the DIPS 2 GeV minimum track p_T results. All results are generated with the test samples, and thus 1.5M jets, for pp and Pb+Pb respectively, with the training weights obtained on epoch 300. In this study, an $f_c = 0.05$ has been used.

cuts individually applied, new results are found and are presented in figure 4.25. When comparing the performance results as a function of collisions centrality, better achievements can be seen as centrality gets lower. Results show that the 1 GeV sample had an overall increase rounding 1.05 to 1.15 times from central to peripheral collision in a $\varepsilon_b = 75\,\%$ for u-jets, meaning a 15 % u-jet rejection increase for the peripheral 1 GeV sample compared to the 2 GeV one, and a constant 3 % c-jet rejection increase for 1 GeV sample compared to the 2 GeV for all collision centralities in a $\varepsilon_b = 75\,\%$. The reason for this performance is justified by the feature rank and track p_T cut dependence, presented in figures 4.26 for each collision centrality. As centrality increases, both 1 GeV and 2 GeV samples lose much of their jet-flavor discrimination, mainly represented by the IP3D S_{z_0} which has a decrease in the positive b-tagging discriminant gradient and a slight increase in the negative u-tagging discriminant, but the 1 GeV track p_T cut has a slight better flavor-jet discrimination than the 2 GeV, observed in peripheral collisions. This results prove to be statistically dependent and a study with a bigger statistic might be beneficial, but this prove to be impossible due to memory pressure problems.

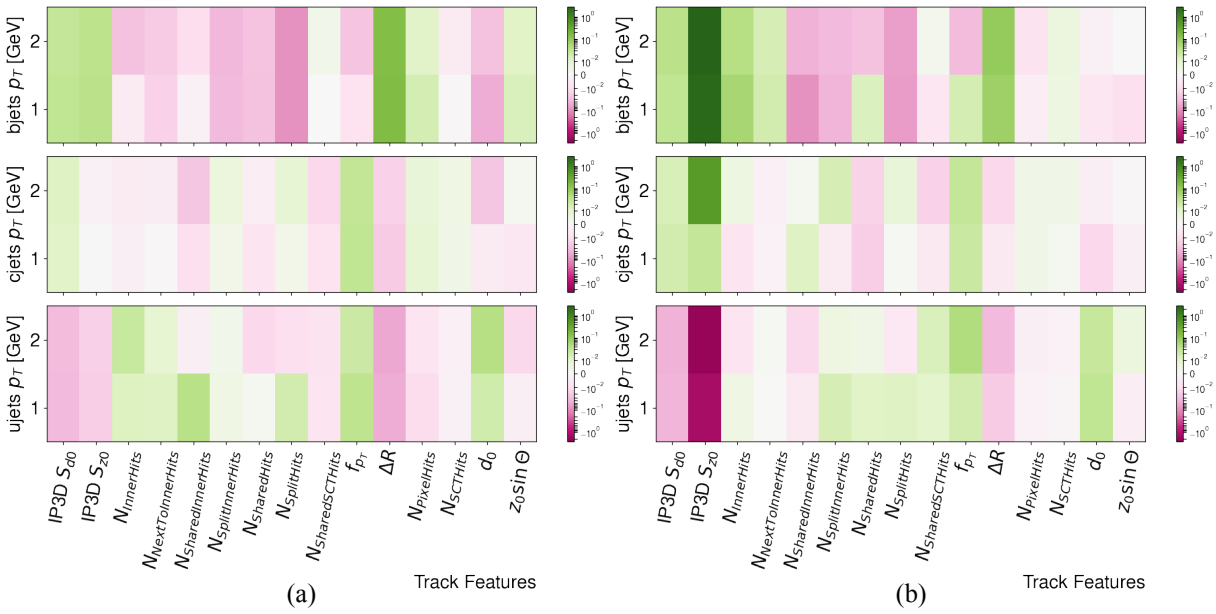
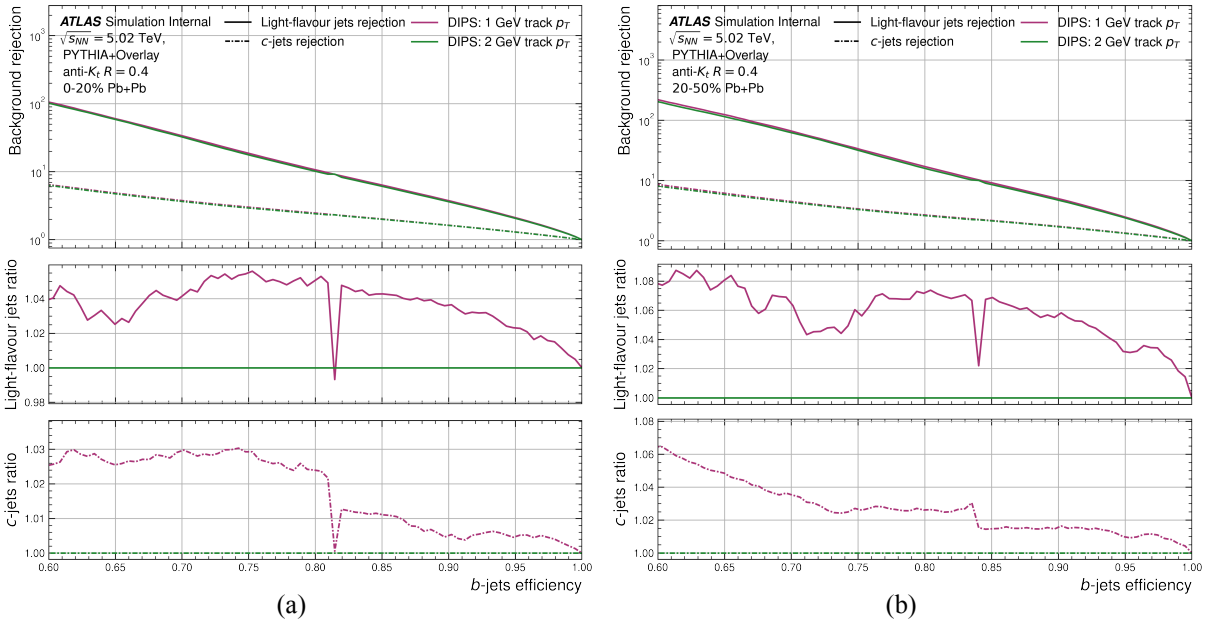


Figure 4.24: Track features rank for (a) pp and (b) Pb+Pb 1 GeV and 2 GeV track p_T cut samples on DIPS for b-, c- and u-jets (top, middle and bottom respectively) for jets passing a flavor-tagging efficiency of 77 %. For this study 500k test sample jets were used. The color base values are referent to each track feature contribution on the DIPS flavor-tagging discriminant gradients presented in equations 4.3, just as on section 4.1, with the top subfigure related to the b-tagging, the middle one to the c-tagging and the bottom one to the u-tagging. The green colors mean a positive discriminant contribution and overall training performance increase and the pink colors, the opposite.



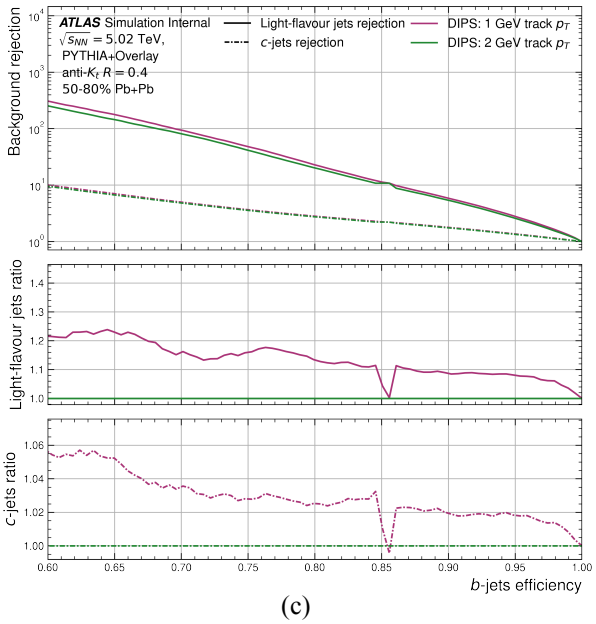


Figure 4.25: ROC plots obtained for 1 GeV and 2 GeV (a) central, (b) semi-central and (c) peripheral Pb+Pb collisions on DIPS training. Each ROC is copulated with two ratio plots, for both u-jets and c-jets, where the reference is considered as the DIPS 2 GeV minimum track p_T results. These results are generated by the 1.5M test sample jets and by making use of the training weights obtained on epoch 300. In this study, an $f_c = 0.05$ has been used.

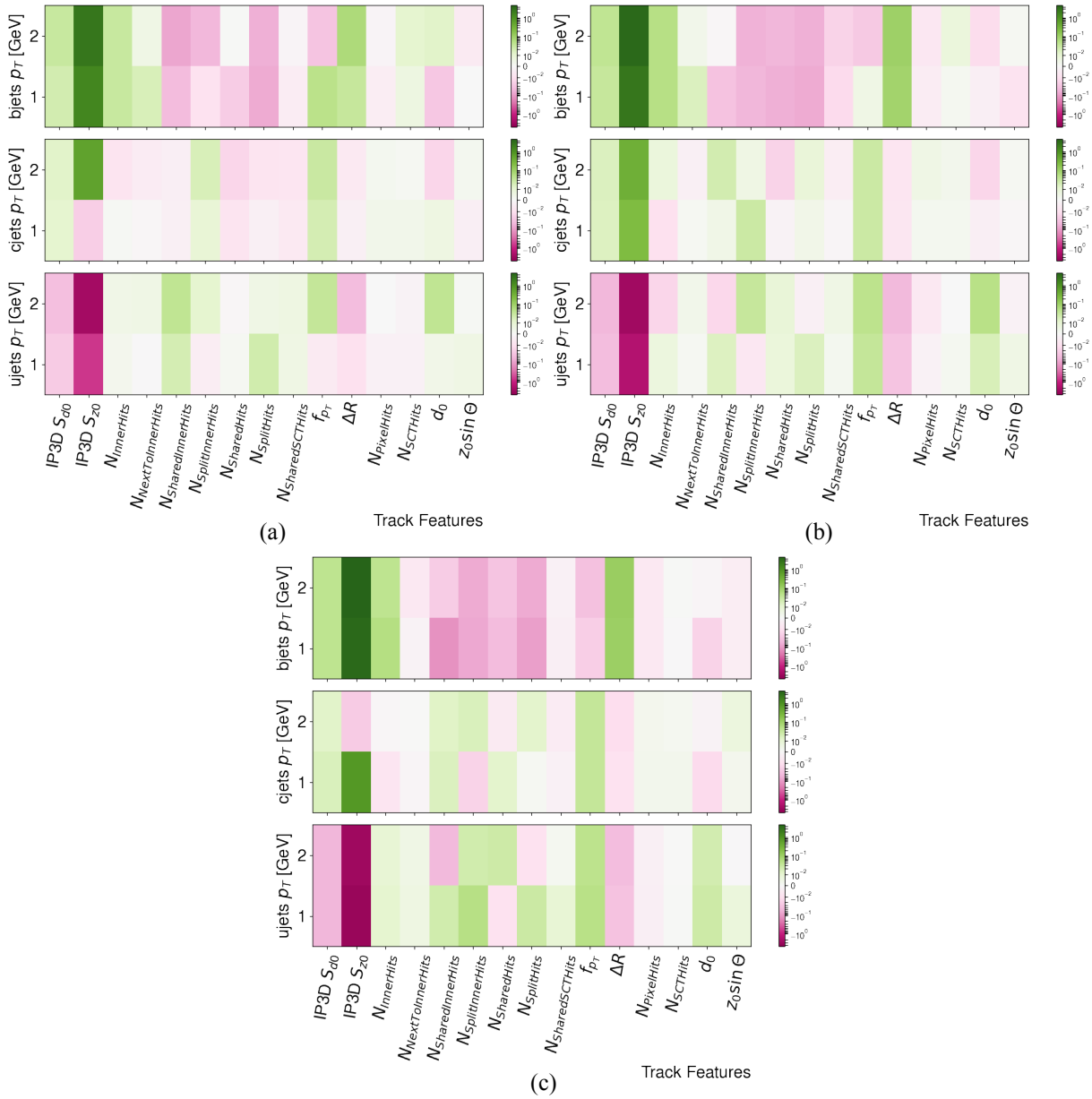


Figure 4.26: Track features rank for 2 GeV and 1 GeV track p_T cut samples on DIPS for (a) central, (b) semi-central and (c) peripheral Pb+Pb collision centralities with jets passing a flavor-tagging efficiency of 77 %. For this, 500k test sample jets were used. The color base values are referent to each track feature contribution on the DIPS flavor-tagging discriminant gradients presented in equations 4.3, just as on section 4.1, with the top subfigure related to the b-tagging, the middle one to the c-tagging and the bottom one to the u-tagging. The green colors mean a positive discriminant contribution and overall training performance increase and the pink colors, the opposite.

Chapter 5

Conclusion

Neural Networks are extremely powerful tools that offer huge investigation possibilities, allowing the accomplish of milestones previously impossible or hard to overcome. At the ATLAS experiment, at CERN, these tools have been studied and applied with the objective of increasing the particle jets identification in Pb+Pb collisions and thus, the reliability of new theories. Until now, the results taken were applied for pp collisions, but a clear bigger motivation was related to the performances in heavy ions.

Through out this work several analyses were made for Pb+Pb data collisions with pp data as reference, where two very distinct NN based algorithms, DL1 and DIPS, were used to answer to much of the questions related to the heavy ion b-tagging performances. In fact, this work covered all the training process made by each NN (section 4), from the different jet and track training variable rank and correlation analysis (section 4.1) to the performance and training results (sections 4.2 and 4.3) of each tagger.

From this process several conclusions were taken, related to the jet and track variable contributions, that have much differences for pp and Pb+Pb across all trainings.

Concerning track and jet variable correlations (section 4.2), no significant difference was observed between pp and Pb+Pb. Contrarily to variable ranks, several variables were much less meaningful on heavy ions, contributing to a lower b-tagging discrimination, with the same happening for pp as well. Nonetheless, such variables, commonly used in pp studies, contributed to the c- and light-jet flavor tagging and were used during each NN training, providing the expected results with a much better performance on pp data collisions, as explained by the lower amount of particles and the much smaller underlying event contribution.

Relatively to the dependence on Pb+Pb collisions centrality of training performances of the DL1 algorithm (section 4.2.2), despite the same expected dominant hierarchy, differences regarding the c-jet rejection and jet p_T distributions were observed. While the peripheral and semi-central Pb+Pb samples followed the pp distribution monotonally increasing and then stabilizing with the jet p_T, the central one had an opposite decreasing behavior, justified by the variables discriminant power variance with the jet p_T. As a consequence, the E_T^{FCal} variable is included as a training variable and results regarding its

train influence (variable rank) and performance impact are analyzed in section 4.2.3 where better results were obtained for peripheral, semi-central, and central samples, respectively, increasing the u-jet flavor rejection.

For DIPS, a hyperparameter optimization study is conducted in section 4.3 and a track p_T cut influence study is made with the optimized NN parameters. Results show that the 1 GeV track p_T cut sample had a better training performance when compared to the 2 GeV track p_T cut sample, which could be improved by the collision centrality separation and is sustained by the track variable ranks study (section 4.3.1).

5.1 Future Work

Despite these work efforts, at the heavy ion collisions flavor tagging, several additional conclusions and studies weren't possible and are as important as the ones presented.

Regarding the conducted studies, most of the analysis were statistically dependent, but due to lack of computer power and memory pressure, the jet/track samples used were sometimes characterized with a lower statistics. Examples are the track feature vs track p_T cut studies made on tracks, using DIPS, where no big conclusions can be retrieved, contrarily to the same analysis in ROC curves that made use of a bigger amount of jets.

Additionally it would be much more useful to do the exact same studies conducted in this work for DL1d, the most updated version of DL1 tagger, instead the used DL1 baseline with a different architecture and variables, however due to data collection problems this showed to be impossible.

Appendices

Appendix A

Appendices to preprocessing section

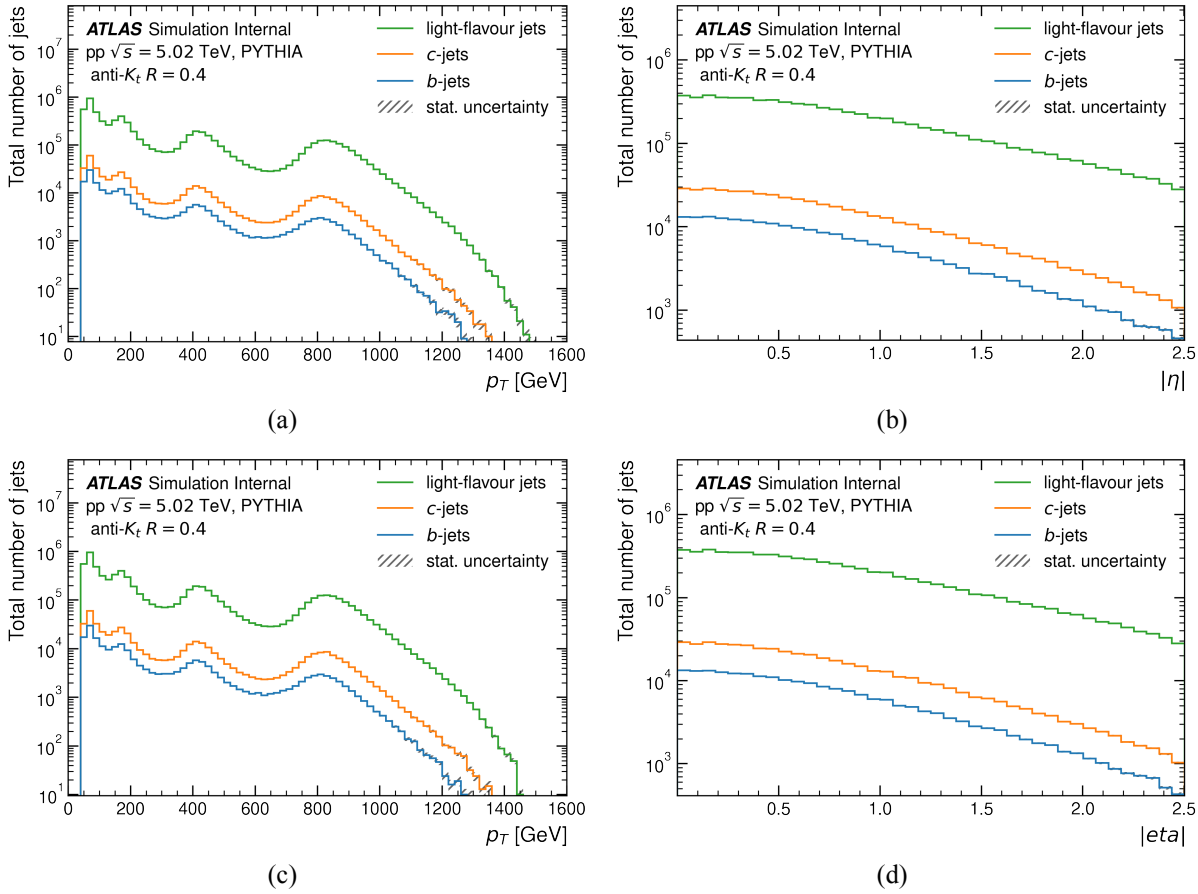


Figure A.1: pp (a) jet p_T and (b) $|\eta|$ validation sample distributions. pp (c) jet p_T and (d) $|\eta|$ test sample distributions.

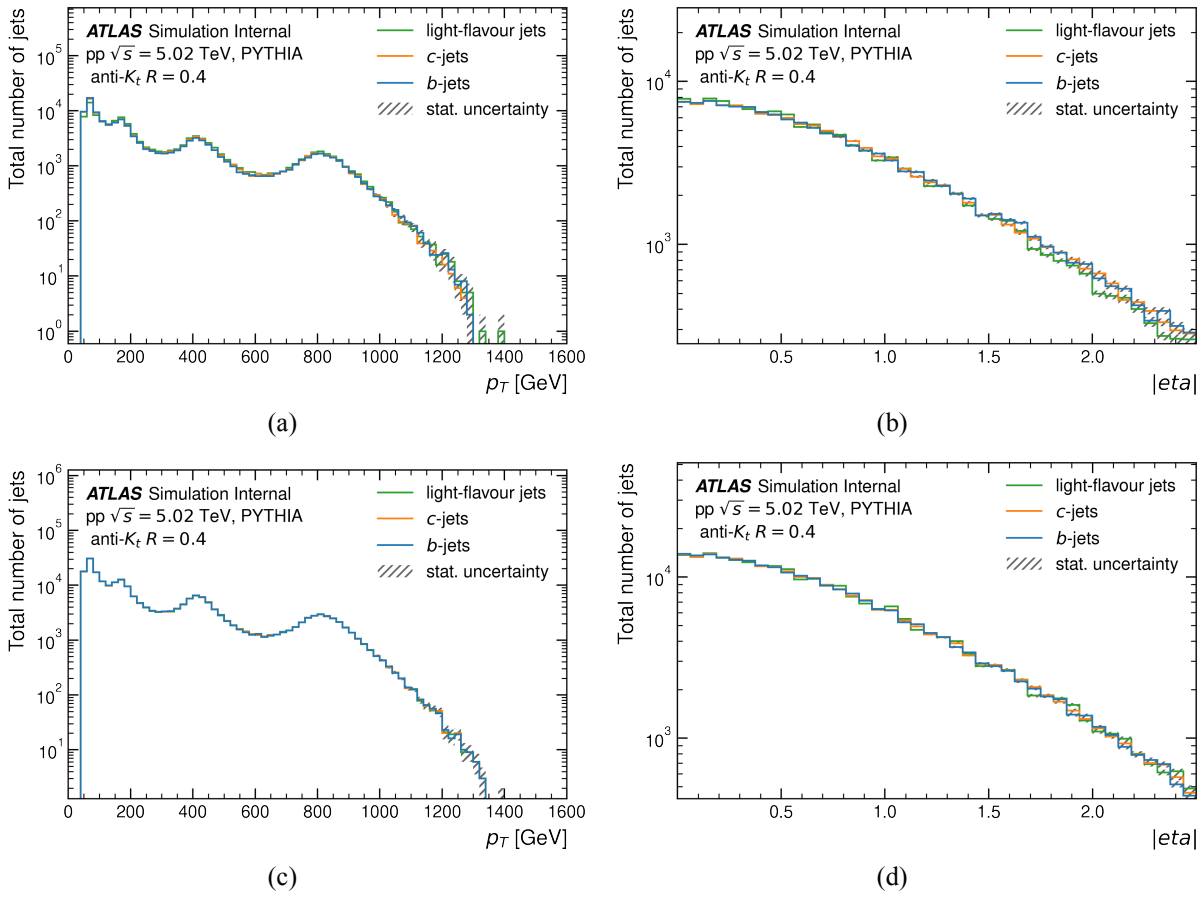


Figure A.2: pp (a) jet p_T and (b) $|\eta|$ validation resample distributions. pp (c) jet p_T and (d) $|\eta|$ test resample distributions.

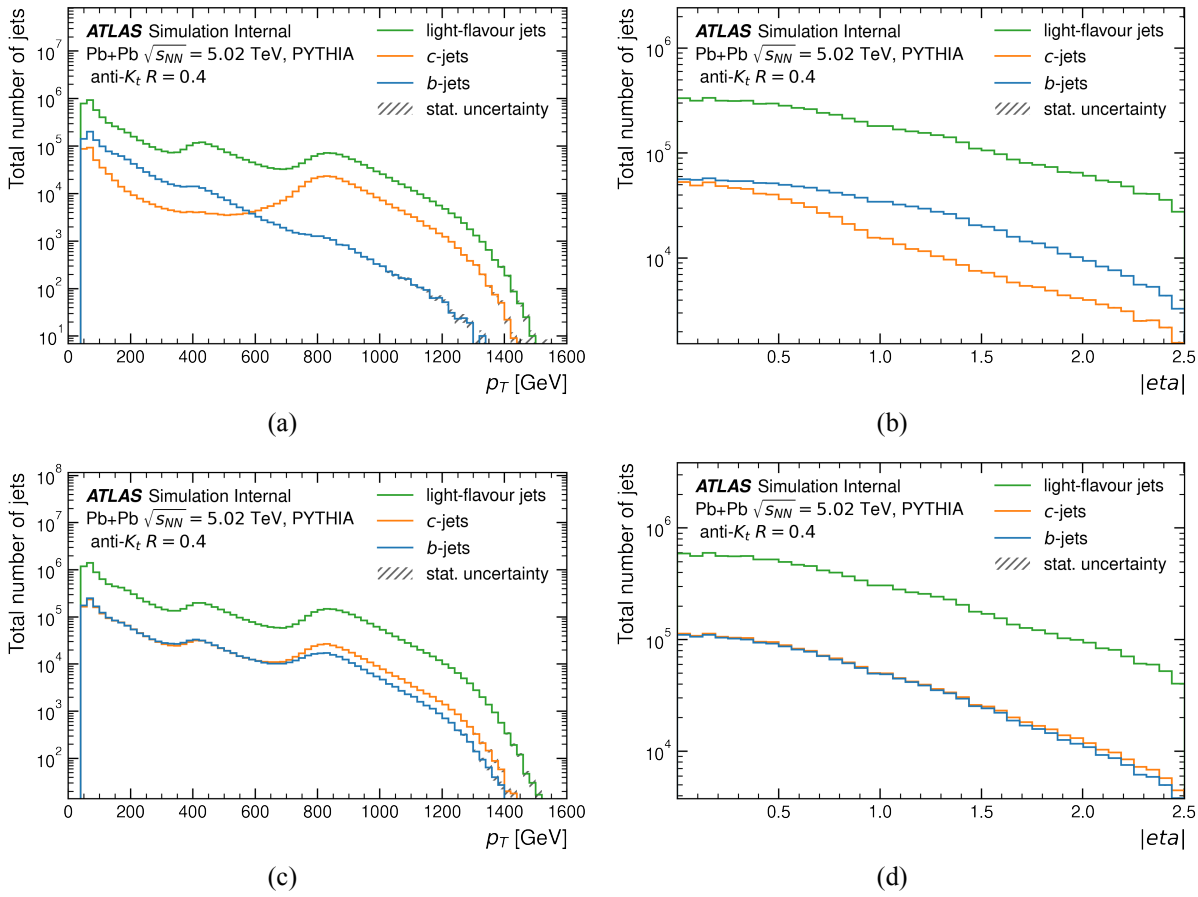


Figure A.3: Pb+Pb (a) jet p_T and (b) $|\eta|$ validation sample distributions. Pb+Pb (c) jet p_T and (d) $|\eta|$ test sample distributions.

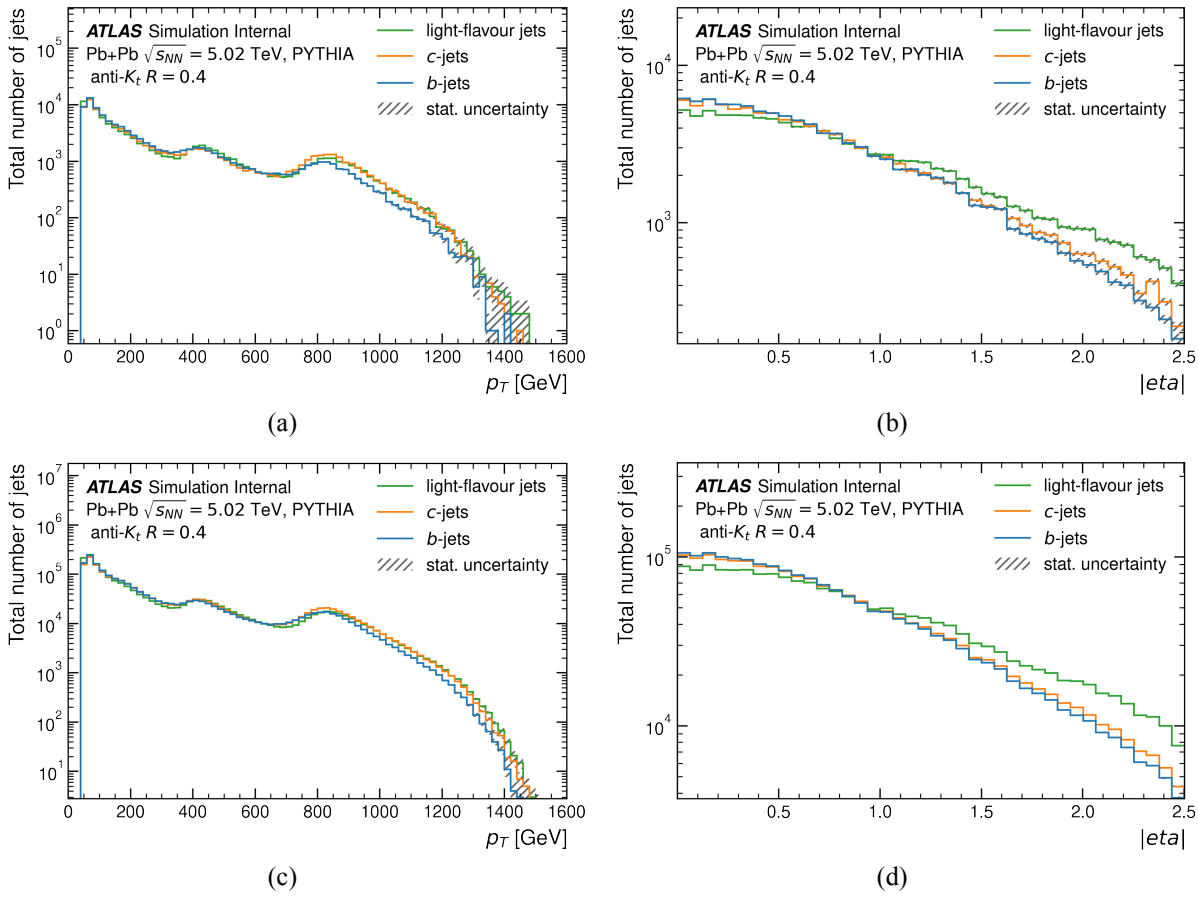


Figure A.4: Pb+Pb (a) jet p_T and (b) $|\eta|$ validation resample distributions. Pb+Pb (c) jet p_T and (d) $|\eta|$ test resample distributions.

Appendix B

Appendices to training section

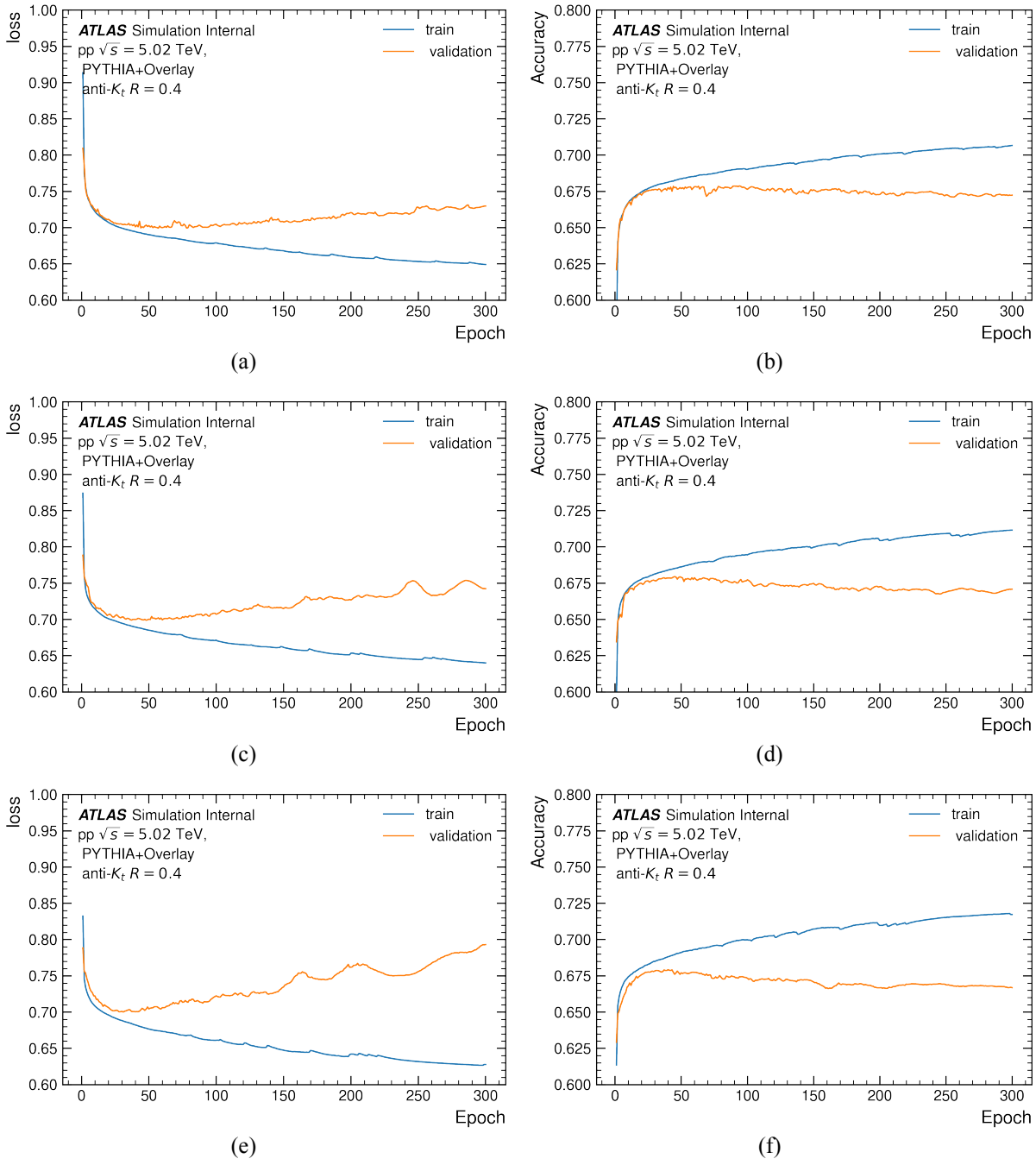


Figure B.1: DIPS training loss curves obtained with a learning rate of 0.001 and a minibatch size of (a) 15000, (c) 10000 and (e) 5000 and training accuracy curves obtained with a learning rate of 0.001 and a minibatch size of (b) 15000, (d) 10000 and (f) 5000 for pp train and validation samples.

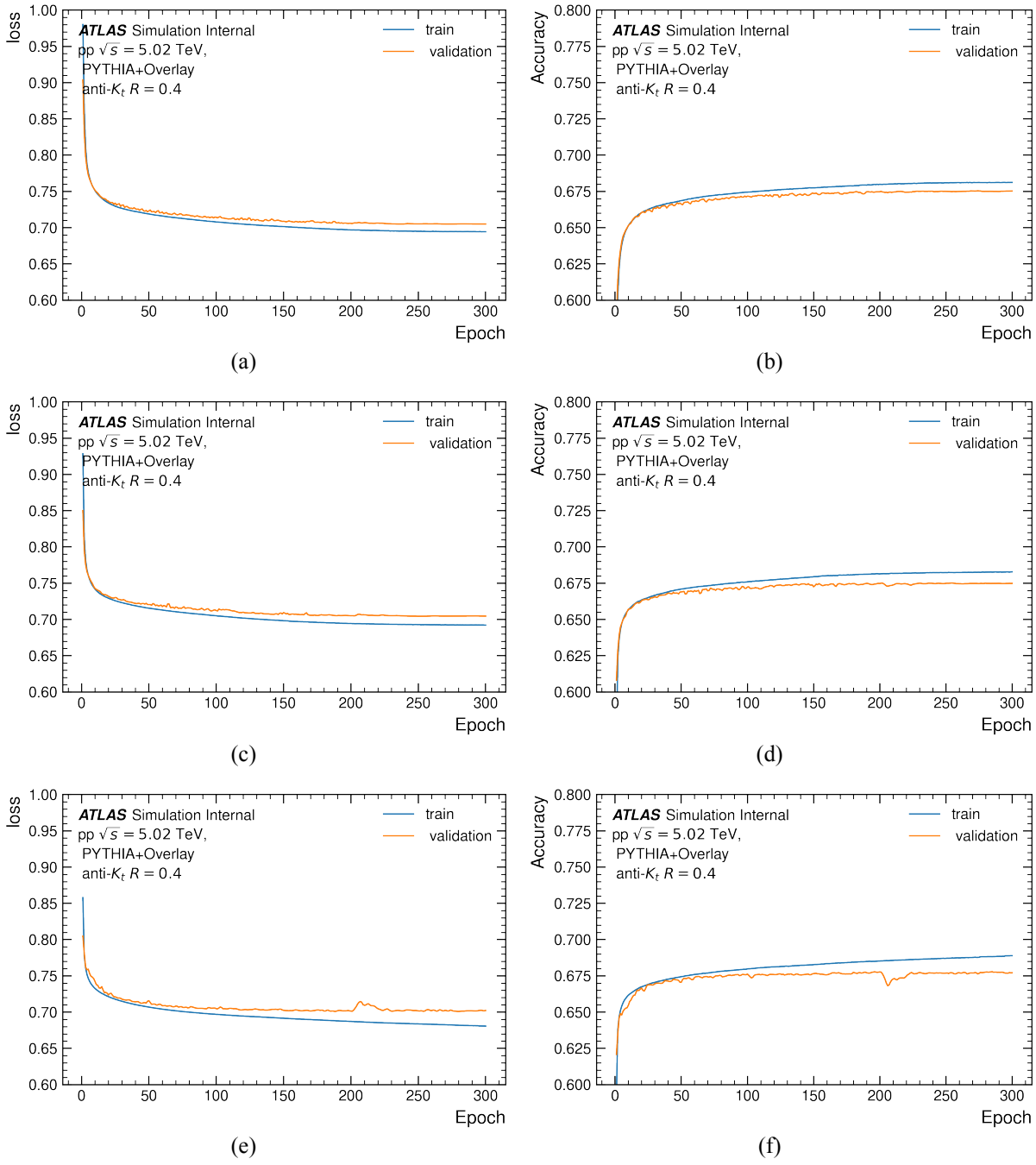


Figure B.2: DIPS training loss curves obtained with a learning rate of 0.0001 and a minibatch size of (a) 15000, (c) 10000 and (e) 5000 and training accuracy curves obtained with a learning rate of 0.001 and a minibatch size of (b) 15000, (d) 10000 and (f) 5000 for pp train and validation samples.

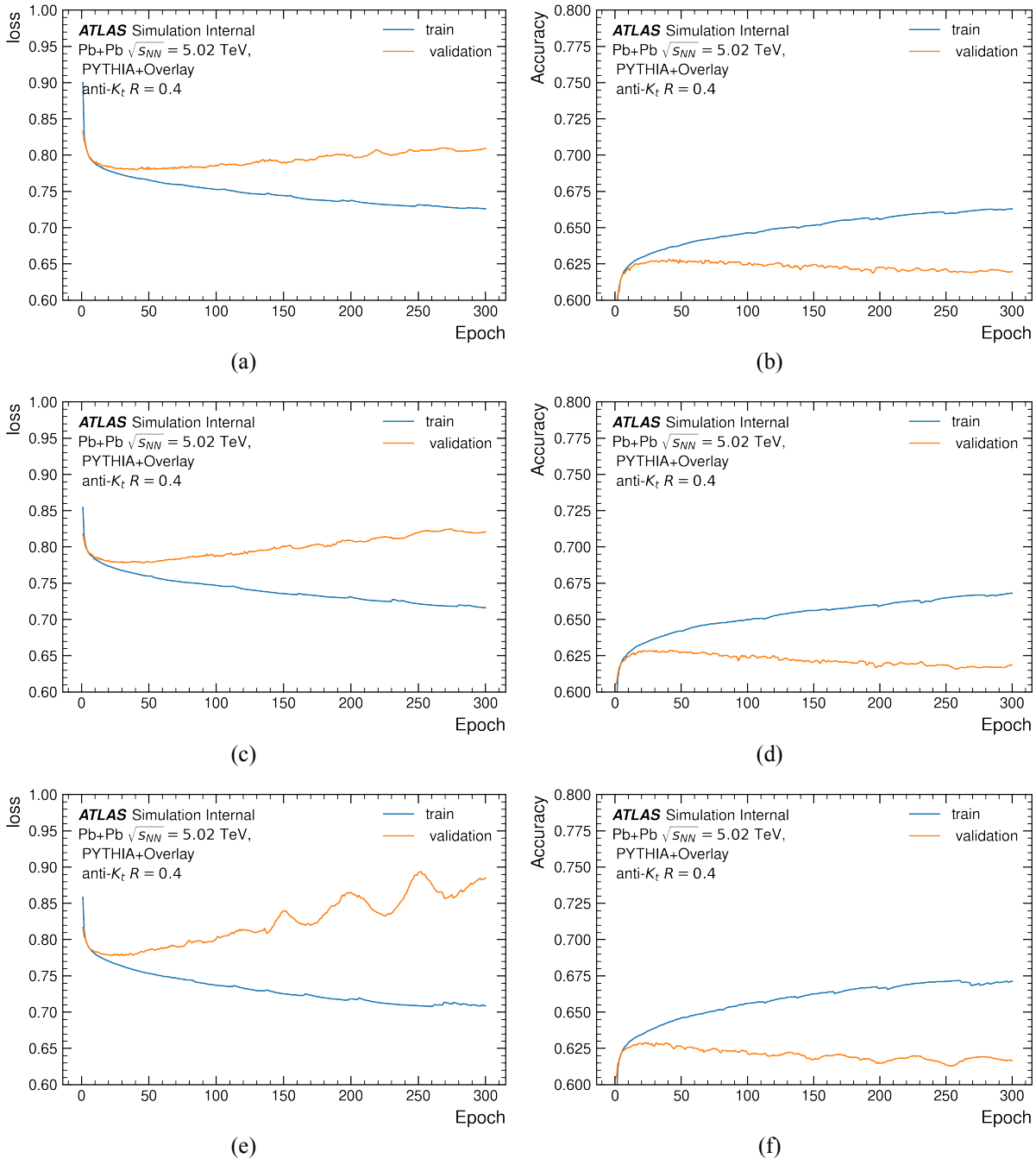


Figure B.3: DIPS training loss curves obtained with a learning rate of 0.001 and a minibatch size of (a) 15000, (c) 10000 and (e) 5000 and training accuracy curves obtained with a learning rate of 0.001 and a minibatch size of (b) 15000, (d) 10000 and (f) 5000 for Pb+Pb train and validation samples.

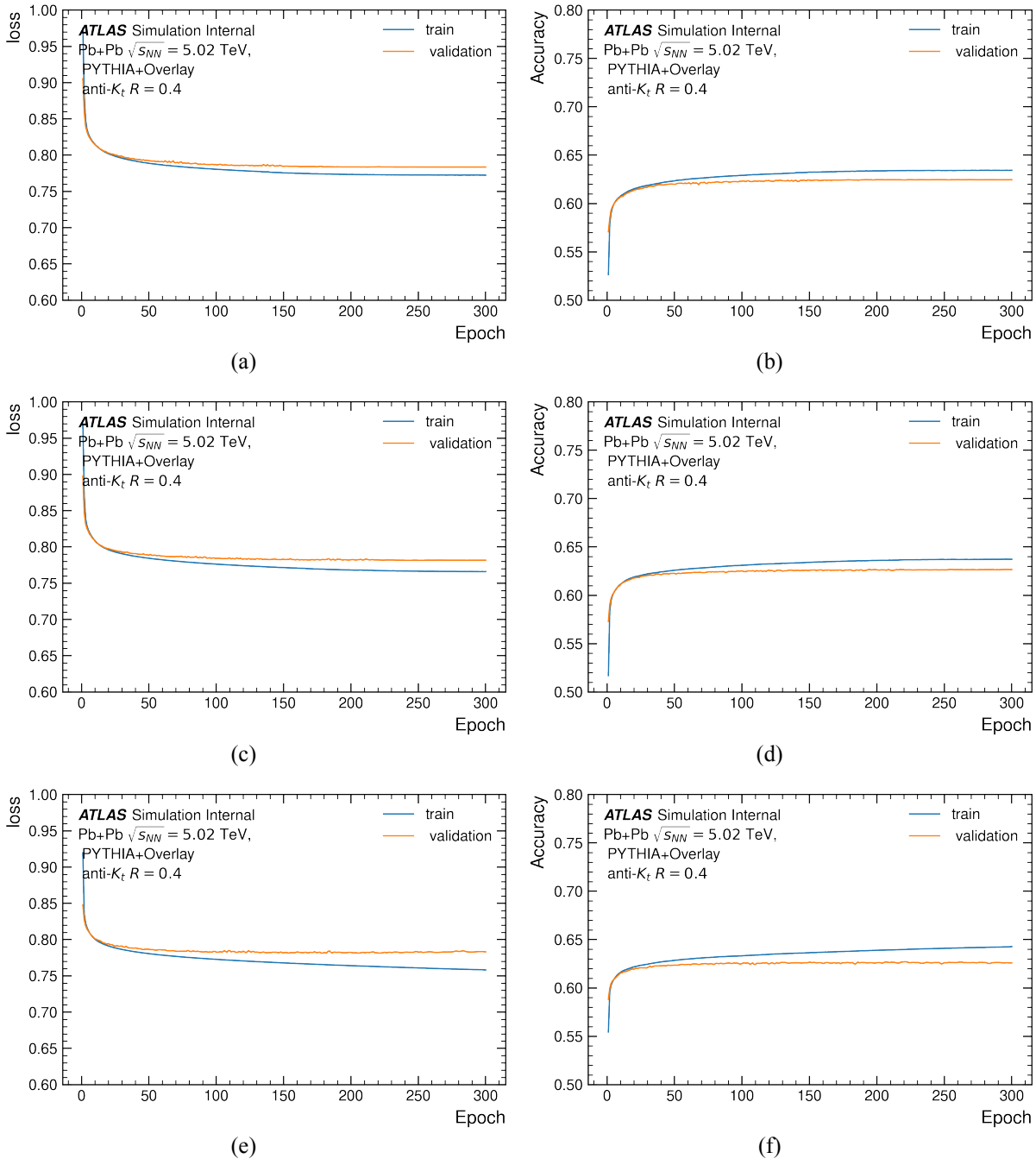


Figure B.4: DIPS training loss curves obtained with a learning rate of 0.001 and a minibatch size of (a) 15000, (c) 10000 and (e) 5000 and training accuracy curves obtained with a learning rate of 0.001 and a minibatch size of (b) 15000, (d) 10000 and (f) 5000 for Pb+Pb train and validation samples.

References

Aad, G. e. a. (2019). Atlas b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ tev. *The European Physical Journal C*, 79(11). 2, 6, 13, 14, 16, 17

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer. 20

Collaboration, T. A. (2014). ATLAS Pythia 8 tunes to 7 TeV data. Technical report, CERN, Geneva. 26

Collaboration, T. A. (2016). Performance of b-jet identification in the atlas experiment. *Journal of Instrumentation*, 11(04):P04008–P04008. XI, 7, 8, 9, 10, 12, 13, 15, 16, 18

Collaboration, T. A. (2017a). Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment. Technical report, CERN, Geneva. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-003. 22, 23

Collaboration, T. A. (2017b). Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run. Technical report, CERN, Geneva. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-013. 2, 15

Collaboration, T. A. (2017c). Optimisation and performance studies of the atlas b-tagging algorithms for the 2017-18 lhc run. Technical report, CERN, Geneva. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-013. 22

Collaboration, T. A. (2020). Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS. Technical report, CERN, Geneva. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2020-014. 2, 22, 23, 24

Collaboration, T. A. (2022). Measuring the b-jet identification efficiency for high p_T jets using multijet events in proton–proton collisions at $\sqrt{s} = 13\,\mathrm{TeV}$ recorded with the ATLAS detector. Technical report, CERN, Geneva. 10

Collaboration, T. A. and et al., G. A. (2008). The ATLAS experiment at the CERN large hadron collider. *Journal of Instrumentation*, 3(08):S08003–S08003. 5

F. Chollet et al. (2015). Keras. Technical report. 17

G. Aad et all (2012). Measurement of the pseudorapidity and transverse momentum dependence of the elliptic flow of charged particles in lead–lead collisions at $\sqrt{s_{NN}} = 2.76\,\mathrm{TeV}$ with the ATLAS detector. Technical report, CERN, Geneva. 47

Goodfellow et al. (2013). Maxout networks. 20

Hoecker et al. (2007). Tmva - toolkit for multivariate data analysis. 15, 16

Ian Goodfellow, Yoshua Bengio, A. C. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org. 18

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. 17

Komiske, P. T., Metodiev, E. M., and Thaler, J. (2019). Energy flow networks: deep sets for particle jets. *Journal of High Energy Physics*. 22

Lanfermann et al. (2017). Deep Neural Network based higher level flavour tagging algorithm at the ATLAS experiment. Technical report, CERN, Geneva. 18, 20, 21

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814. 20

Obikhod, T. and Petrenko, I. (2020). B-tagging and searches for new physics beyond the standard model. *Problems of Atomic Science and Technology*, pages 3–7. 10

T. Sjöstrand, e. a. (2015). An introduction to PYTHIA 8.2. *Computer Physics Communications*, 191:159–177. 26

The ATLAS Colaboration (2022a). Athena. 41

The ATLAS Colaboration (2022b). ATLAS PanDa. 41

The ATLAS Flavor Tag Group (2022a). Training-Dataset-Dumper. 28

The ATLAS Flavor Tag Group (2022b). UMAMI. 25

The Theano Development Team and et al., A.-R. (2016). Theano: A python framework for fast computation of mathematical expressions. 17