

From strange-quark tagging to fragmentation tagging with machine learning

Yevgeny Kats and Edo Ofir

Department of Physics, Ben-Gurion University, Beer-Sheva 8410501, Israel

E-mail: katsye@bgu.ac.il, edoof@post.bgu.ac.il

ABSTRACT: We apply advanced machine learning techniques to two challenging jet classification problems at the LHC. The first is strange-quark tagging, in particular distinguishing strange-quark jets from down-quark jets. The second, which we term fragmentation tagging, involves identifying the fragmentation channel of a quark. We exemplify the latter by training neural networks to differentiate between bottom jets containing a bottom baryon and those containing a bottom meson. The common challenge in these two problems is that neither quark lifetimes and masses nor parton showering provide discriminating tools, making it necessary to rely on differences in the distributions of the hadron types contained in each type of jet and their kinematics. For these classification tasks, we employ variations of Graph Attention Networks and the Particle Transformer, which receive jet and all constituent properties as inputs. We compare their performance to a simple Multilayer Perceptron that uses simple variables. We find that the more sophisticated architectures do not improve s -quark versus d -quark jet differentiation by a significant amount, but they do lead to a significant gain in b -baryon versus b -meson jet differentiation.

Contents

1	Introduction	1
2	Event simulation	4
2.1	Event generation	4
2.2	Detector simulation	4
2.3	Reconstruction of K_S and Λ decays	5
2.4	Jet clustering and preprocessing	6
3	Basic discriminating variables	7
3.1	Strange vs. down jets	7
3.2	Bottom baryon vs. meson jets	12
4	ML-based taggers	16
4.1	NN inputs	16
4.2	NN architectures	17
4.3	Performance	18
4.3.1	Strange tagging	18
4.3.2	Fragmentation tagging	21
4.4	Robustness to measurement errors	23
5	Summary and discussion	23
A	NN details	24
A.1	Graph Attention Network (GAT)	24
A.2	Particle Transformer (ParT)	25
A.3	Multilayer Perceptron (MLP)	29
A.4	Hyperparameters	29

1 Introduction

Jets, which are collimated sprays of particles, are among the most ubiquitous objects produced at the Large Hadron Collider (LHC) at CERN and other high-energy colliders. Determining a jet’s origin is often crucial for deciphering the underlying physical process that occurred in the collision. In other cases, one would like to focus on particular hadrons that might have been produced, with their decay products contained in the jet. However, these tasks are complicated by the stochastic nature of the processes governing the evolution of the original particle into a jet, which typically ends up containing tens of particles. Moreover, due to the limitations of detectors, only certain properties of the jet constituents

can be measured. Machine learning (ML) tools are therefore a natural fit for analyzing jet data. They have been already proven effective in many such tasks (e.g., refs. [1, 2]).

We are interested in strange-quark tagging as part of a broader effort to distinguish between different types of quarks, gluons, and other objects that produce jets. In many cases, various established tagging techniques exist. Bottom and charm quarks travel several millimeters within the detector because of their long lifetimes. Their decays produce secondary vertices, aiding in their recognition and differentiation from other particles [3, 4]. Another example involves distinguishing gluon from light-quark (u, d, s) jets. This is done by noting that gluon jets are usually wider, have a larger number of constituents, and exhibit more uniform energy fragmentation [5]. Similarly, up and down-quark jets can be partially distinguished using the p_T -weighted track charge [6]. Moreover, in scenarios where a heavy hadronically decaying particle (e.g., a W, Z or Higgs boson or a top quark) is boosted, such that its decay products are collimated into a single jet, it can be distinguished from quark/gluon jets by using the jet mass or substructure variables (e.g., refs. [7, 8]). Distinguishing between strange and down-quark jets, on the other hand, remains a challenge since these quarks have small masses and identical QCD and electromagnetic interactions, so the only difference between them is found in their hadronization and subsequent decay processes.

Simple methods for strange-quark tagging were implemented by DELPHI [9, 10] and OPAL [11] at LEP and by SLD [12] at SLAC in measurements of the strange-quark forward-backward asymmetry in e^+e^- collisions near the Z pole. These methods relied on the fact that the particle with the highest energy in a jet tends to carry the flavor of the primary quark. Therefore, strange jets can be characterized by them containing an energetic charged or neutral kaon or a Λ baryon. More recently, a combination of such inputs was proposed for identifying Higgs boson decays to $s\bar{s}$ in a future e^+e^- collider [13]. A recurrent neural network (RNN) was proposed for the same task in ref. [14]. And a short while ago, a transformer-based neural network was proposed for jet flavor tagging, including strange-quark tagging, for the FCC-ee [15].

Our focus will be on the multipurpose LHC detectors ATLAS [16] and CMS [17], where the strange-tagging challenge is exacerbated by the inability to determine the identity of charged hadrons. This implies, in particular, that charged kaons (K^\pm) cannot be distinguished from charged pions and protons. Additionally, for jet transverse momentum (p_T) above a few tens of GeV, most of the energetic short-lived neutral kaons (K_S) and Λ baryons reach the hadronic calorimeter (HCAL) without decaying, like the long-lived neutral kaons (K_L), which makes them indistinguishable from neutrons or a collection of soft neutral hadrons. Still, the presence of these energetic neutral strange hadrons in s -quark jets leads to an increased energy fraction, on average, deposited in the HCAL, while d -quark jets have a greater energy fraction deposited in the electromagnetic calorimeter (ECAL) due to energetic neutral pions decaying to photons. However, the distributions of these quantities for the two classes of jets overlap significantly, which limits their discriminating power.

Several studies on possible s -tagging strategies for ATLAS and CMS have been reported in the literature [18–21]. Ref. [18] considered processing the tracks in the jet with a Long Short-Term Memory (LSTM) RNN. Refs. [20, 21] extended this study to use calorime-

ter information, again with LSTM RNNs. Ref. [21] also implemented a feedforward neural network (FNN) with multiple properties of the jet and of reconstructed K_S and Λ hadrons as inputs. The resulting s tagger was calibrated on ATLAS $t\bar{t}$ samples with hadronic W decays, and then used to constrain the CKM matrix elements $|V_{ts}|$ and $|V_{td}|$ by analyzing data from top-quark decays. Ref. [19] considered Boosted Decision Tree (BDT) classifiers with whole-jet energy fractions as inputs, as well as Convolutional Neural Networks (CNNs) applied to jet images, and found the CNNs to outperform the BDTs by a small amount. In all these cases, the discriminating power was found to be quite limited, with AUC scores not exceeding 0.64.

In this paper, we make another attempt to tackle the difficult problem of strange-jet tagging by employing different and more sophisticated neural network (NN) architectures that utilize *attention mechanisms*. One is a variant of the Graph Attention Network (GAT) [22, 23], which is a powerful Graph Neural Network (GNN) [24, 25] architecture. Another NN we use is based on the attention mechanisms utilized in the famous *transformer* architecture [26, 27]. Transformers were shown to outperform GNNs and CNNs in sequence transduction problems [26] and they are at the heart of state-of-the-art artificial intelligence applications, such as ChatGPT [28]. In collider physics, the recently introduced Particle Transformer (ParT) [29] (see also refs. [30–33] for more sophisticated versions) was shown to be very effective in many jet classification tasks, and we will closely follow its architecture. Similar architectures are also being explored by the ATLAS and CMS collaborations for various jet tagging tasks [30, 33–36].

The second problem that we will address with the same tools is *fragmentation tagging*. By this we mean determining the fragmentation channel of a quark, e.g., identifying the type of b hadron that was present in a b jet. The motivation for developing fragmentation taggers comes from several directions. One is measurements of parton fragmentation functions (FFs), which describe how partons transform into hadrons [37, 38]. The FFs, which are determined by non-perturbative QCD matrix elements, describe the probability of a parton producing a specific hadron carrying a certain fraction of the parton momentum. Measuring FFs is useful for improving our understanding, or at least the description, of QCD dynamics, as well as for tuning Monte Carlo generators. While FF measurements can be done using specific clean decay channels of the corresponding hadrons (see, e.g., refs. [39, 40]), it is interesting to ask whether ML methods can help doing more inclusive measurements. Such measurements might be particularly useful at high p_T , where statistics is limited.

Another motivation for fragmentation tagging is analyses targeting jets with (decay products of) specific hadrons, where jets with other hadrons form a background. One such case is the proposed measurements of b -quark polarization and/or $b\bar{b}$ spin correlations in various samples in ATLAS and CMS [41–43]. These proposals rely on reconstructing semileptonic decays of Λ_b baryons in b jets. A fragmentation tagger could help reduce the large background from semileptonic decays of B mesons.

We will demonstrate fragmentation tagging using the example of distinguishing between b jets containing a b baryon vs. those containing a b meson. Various other definitions of classes (e.g., focusing on specific hadrons as signal, distinguishing between particles and

antiparticles, etc.), or narrowing down to a particular subset of jets (e.g., those containing a lepton), could be relevant to specific applications of fragmentation tagging. They can be addressed with the same general approach.

The rest of the paper is organized as follows. Section 2 describes the simulated event samples used in our study. In section 3, we analyze a number of simple variables that can be used for classification. In section 4, we present the neural network architectures we used (whose more detailed descriptions are provided in appendix A) and the resulting performance. Section 5 summarizes our results and conclusions.

2 Event simulation

To obtain representative samples of the jet types of interest, we proceed as follows.

2.1 Event generation

We simulate dijet events in proton-proton collisions with a center-of-mass energy of 14 TeV. We use MadGraph5 3.5.1 [44] to create the hard scattering process and Pythia 8.308 [45] to handle the parton showering, hadronization and decays. We define two kinematic regions for the jets (clustered as described below): $p_{T,\text{jet}} > 200$ GeV and $p_{T,\text{jet}} > 45$ GeV. These jet p_T cuts follow generation-level cuts on the partons of $p_T > 180$ GeV and $p_T > 35$ GeV, respectively, and pseudorapidity $|\eta| < 4$.

To obtain samples of s and d jets, we generate $s\bar{s}$ and $d\bar{d}$ events. We include only the gg and $u\bar{u}$ initial states so that the only difference between the s and d jets in the resulting samples will be their flavor. We do it to ensure that the taggers rely solely on the intrinsic differences between the jets and not on differences between their p_T and η distributions in particular samples.

To obtain samples of b jets, we generate $b\bar{b}$ events from gg and $q\bar{q}$ (with $q = u, d, s$) initial states. The b jets are separated into a sample containing b baryons and a sample containing b mesons. The baryon samples are dominated by the Λ_b^0 (with smaller contributions from Ξ_b^0 , Ξ_b^- , and Ω_b^-), and the meson samples consist of \bar{B}^0 ($\sim 45\%$), B^- ($\sim 45\%$), and \bar{B}_s^0 ($\sim 10\%$). Their antiparticles are included.

2.2 Detector simulation

We consider particles in the range $|\eta| < 4$, which is approximately the range that will be covered by the ATLAS and CMS tracking detectors at the HL-LHC [46, 47]. In addition, charged particles need to satisfy $p_T > 0.5$ GeV. Particles are treated as stable if they do not decay within 1 m from the beam axis and 3 m along the beam axis from the interaction point. Based on them, we form the following detector-level objects with the help of the Monte Carlo truth information:

- Charged hadrons.

We simulate track reconstruction efficiency as a function of the track production radius r_{prod} (relative to the beam axis) according to the expectations for the ATLAS tracker at the HL-LHC [48]. This efficiency starts at about 95% for $r_{\text{prod}} = 0$,

decreases gradually to 65% at $r_{\text{prod}} \approx 38$ cm, and then drops sharply. For $r_{\text{prod}} > 50$ cm, we set the reconstruction efficiency to zero. Charged hadrons that fail track reconstruction are counted as neutral hadrons.

For b jets, we distinguish between charged hadrons originating from the primary vertex (PV), which are produced simultaneously with the b hadron during hadronization, and those produced from the decay of the b hadron, thus originating from a secondary vertex (SV). This distinction is achieved in our simulation based on truth information. If a charged hadron has a b hadron as one of its ancestors, it is categorized as originating from the SV. Otherwise, it is classified as originating from the PV.

- Neutral hadrons.

Associated with the typical HCAL granularity, we implement a grid with cell sizes of 0.1×0.1 in the η - ϕ space and calculate the energy contributed by the neutral hadrons to each cell within this grid. Each populated cell is then described as a single neutral hadron, regardless of how many neutral hadrons actually fell within its boundaries. While charged hadrons deposit their energy in the HCAL as well, it can be approximately subtracted based on the momentum measurement of their tracks in the tracker. Hence, in our simulation, we exclude the energy of charged hadrons from the HCAL measurements, except for those that fail track reconstruction.

- Photons (γ).

Associated with the typical ECAL granularity, we implement a grid with cell sizes of 0.02×0.02 in the η - ϕ space. Photons contributing to the same cell are considered a single photon. We assume that contributions from electrons (except for those that fail track reconstruction) are subtracted based on their track measurements and neglect the electromagnetic energy depositions due to muons and charged hadrons.

- Electrons (e^\pm), except for those failing track reconstruction and then counted as photons.
- Muons (μ^\pm), except for those failing track reconstruction.

2.3 Reconstruction of K_S and Λ decays

Since energetic K_S mesons and Λ baryons are more common in strange than in down-quark jets, it is useful for the purpose of strange-quark tagging to attempt identifying them from their decay products. In addition, since Λ baryons are more common in b -baryon than in b -meson decays, while K_S mesons are more common in b -meson decays, reconstructing K_S and Λ decays is also useful for the purpose of b -baryon/ b -meson discrimination.

The K_S meson decays as $K_S \rightarrow \pi^+\pi^-$ with a 69% branching ratio, and $K_S \rightarrow \pi^0\pi^0$ with a 31% branching ratio. The Λ baryon decays as $\Lambda \rightarrow p\pi^-$ with a 64% branching ratio, and $\Lambda \rightarrow n\pi^0$ with a 36% branching ratio. We attempt to reconstruct K_S and Λ as intermediate particles for decays to charged hadrons that occur at a distance greater than 0.5 cm and less than 50 cm from the beam axis. The lower bound helps to avoid confusion

with prompt tracks that originate from the interaction point, while the upper bound represents the radius beyond which track reconstruction becomes essentially impossible due to an insufficient number of tracker layers that can be used. As mentioned above, we simulate the track reconstruction efficiency as a function of the track production radius based on ref. [48]. If both charged hadron tracks from the K_S or Λ decay are reconstructed, we remove them from the list of charged hadrons and store them as a reconstructed K_S or Λ object instead.

Apart from the Λ baryon, other relatively long-lived hadrons that are common in b -baryon decays are the Σ^+ and Σ^- baryons [49, 50]. The dominant decays of these particles, $\Sigma^+ \rightarrow p\pi^0$, $\Sigma^+ \rightarrow n\pi^+$, and $\Sigma^- \rightarrow n\pi^-$, produce kinked track signatures. Each of the two segments of the kinked track may or may not be reconstructible, depending on the tracker layers it passes through. Due to this nontrivial dependence on the specifics of the tracking detector and tracking algorithms, we will not consider the identification of these signatures in this paper, for simplicity. See, however, related studies in refs. [51–56].

2.4 Jet clustering and preprocessing

The objects defined in sections 2.2 and 2.3 are clustered into jets using the anti- k_t algorithm [57, 58] with a radius parameter $R = 0.4$.

We consider the two leading jets in each event. In the case of $s\bar{s}$ or $d\bar{d}$ production, we assume the two leading jets to be s -quark or d -quark jets, respectively. Quark and antiquark jets are included together in our samples, but we note that for some applications it can be useful to treat them separately. For $b\bar{b}$ production, we include b hadrons as soft ghost particles during jet clustering, and then examine which jets contain a b baryon and which ones contain a b meson. Jets without b hadrons are discarded. After this procedure, the ghost particles are removed from the jets.

Properties of the jets and their constituents are recorded for the analysis. Recorded jet properties are p_T , η , the number of constituents, and the fractions of the jet energy contributed by each type of constituent: photon energy E_γ , electron energy E_e , muon energy E_μ , charged hadron energy E_{CH} , neutral hadron energy E_{NH} , reconstructed $K_S \rightarrow \pi^+\pi^-$ energy E_{K_S} , and reconstructed $\Lambda \rightarrow p\pi^-$ energy E_Λ (including $\bar{\Lambda}$). In the case of b jets, instead of E_{CH} , we use two separate variables, $E_{\text{CH,PV}}$ and $E_{\text{CH,SV}}$, for charged hadrons originating from the primary vertex and those from the secondary vertex, respectively, as detailed in section 2.2.

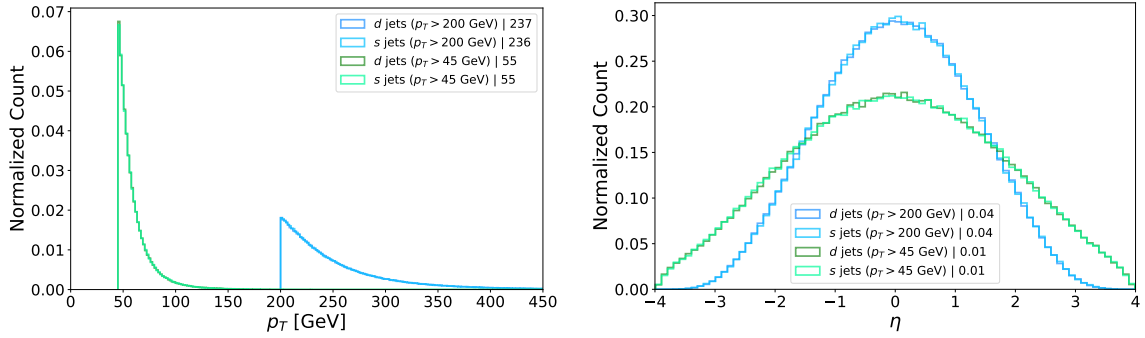
For the jet constituents, the identities are recorded in binary form. We also record the transverse momentum p_T of each constituent i , normalized with respect to the jet p_T ,

$$p_{T,i}^{\text{norm}} = \frac{p_{T,i}}{p_{T,\text{jet}}} . \quad (2.1)$$

The positions of the jet constituents in the η - ϕ space, (η_i, ϕ_i) , are expressed in terms of polar coordinates (r, α) centered on the jet axis, such that

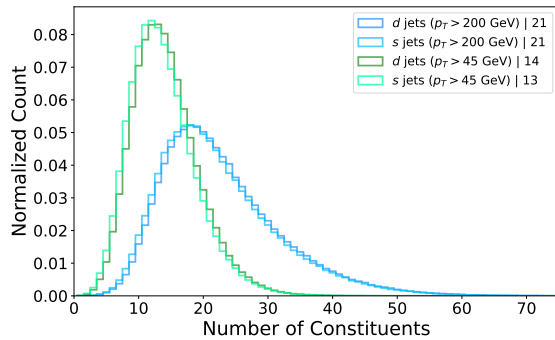
$$\eta_i - \eta_{\text{jet}} = r_i \cos \alpha_i , \quad (2.2)$$

$$\phi_i - \phi_{\text{jet}} = r_i \sin \alpha_i , \quad (2.3)$$



(a) Jet transverse momentum distributions.

(b) Jet pseudorapidity distributions.



(c) Number of constituents distributions.

Figure 1: Properties of d -quark and s -quark jets in our samples for $p_{T,\text{jet}} > 200$ GeV and $p_{T,\text{jet}} > 45$ GeV. Median values are given in the legends.

where $\alpha = 0$ is defined by the location of the most energetic constituent. In addition, we flip the signs of all α_i values if the total momentum on the left side of the $\alpha = 0$ line is greater than on the right side. This standardizes the data and removes physical redundancies before feeding the data into the neural networks.

3 Basic discriminating variables

In this section, we look at the distributions of several simple variables that characterize the jets. Some of them could potentially be used to distinguish between s -quark and d -quark jets, or between b -baryon and b -meson jets.

3.1 Strange vs. down jets

We first examine variables that characterize the entire jet, including $p_{T,\text{jet}}$, η_{jet} , and the number of constituents, for s -quark and d -quark jets. Their distributions are shown in figures 1a, 1b, and 1c, respectively. The distributions of $p_{T,\text{jet}}$ and η_{jet} are essentially identical between the d -quark and s -quark jets in our samples, as expected from our choice of the production processes. The number of constituents is higher for higher- p_T jets, as expected, and tends to be slightly higher in d -quark jets than in s -quark jets.

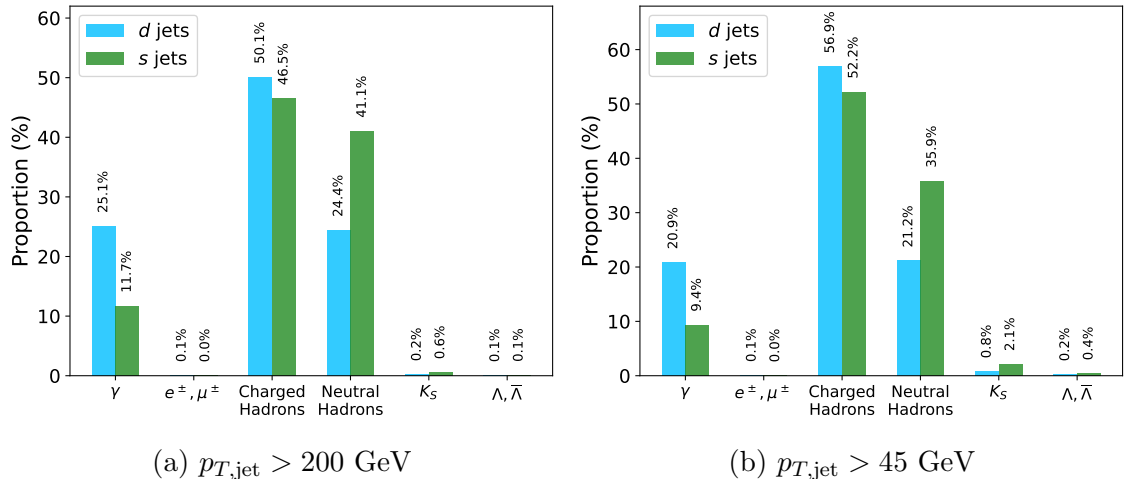


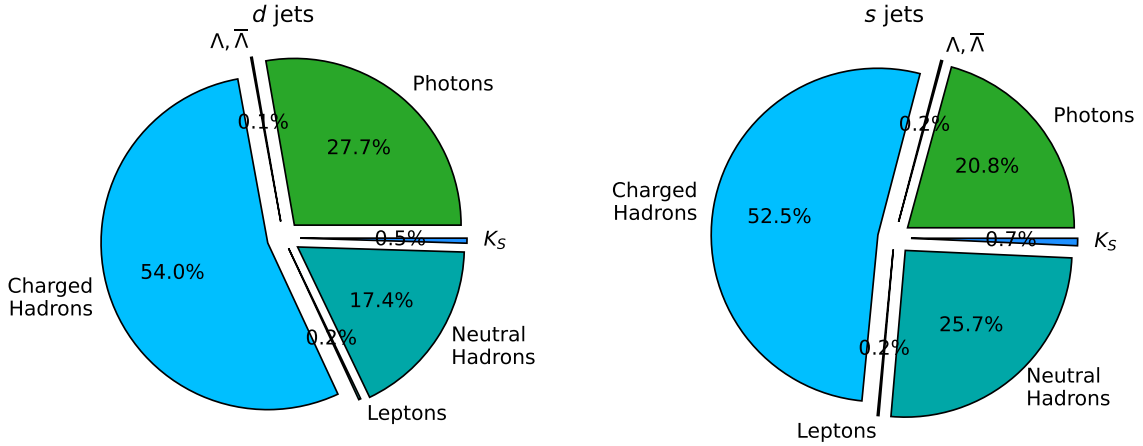
Figure 2: Distributions of the constituent types with the highest transverse momentum (p_T) in s -quark and d -quark jets for (a) $p_{T,\text{jet}} > 200$ GeV, (b) $p_{T,\text{jet}} > 45$ GeV.

Figure 2 shows the distributions of the identities of the constituents with the highest p_T within each jet. As expected, we observe that it is more common for s -quark jets than for d -quark jets to have a neutral hadron or a reconstructed K_S as the most energetic constituent, while it is the other way around for photons. It is related to the fact that an s quark often produces an energetic K_L or K_S meson, while d quarks produce energetic π^0 mesons (which decay to photons) more frequently.

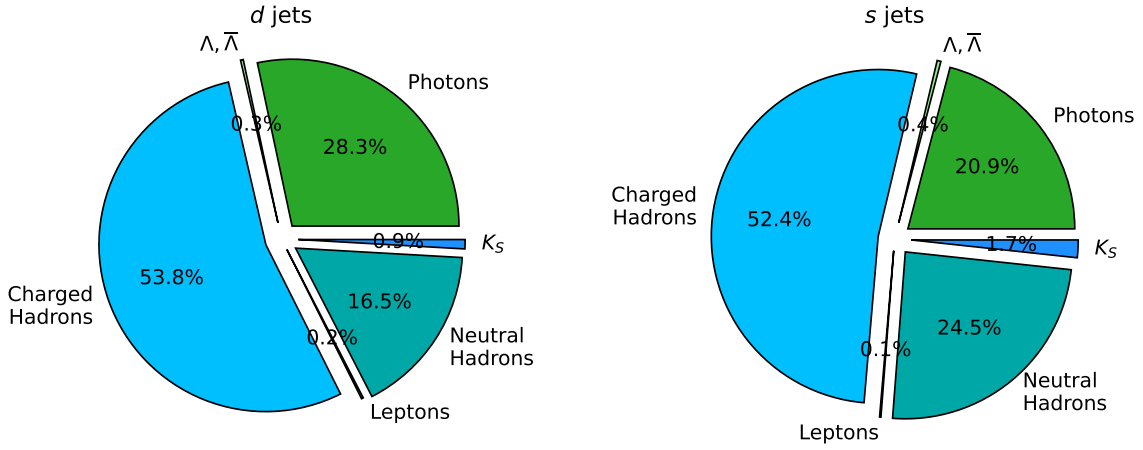
Figure 3 shows the mean energy fraction contributed by each type of constituent within the jet. Related to the previous observations, we see that the energy in neutral hadrons (as well as reconstructed K_S decays, especially for low- p_T jets) is greater for s -quark jets, while the energy in photons is higher for d -quark jets. Consequently, s -quark jets deposit a larger proportion of their energy in the HCAL, while d -quark jets tend to deposit a greater fraction of their energy in the ECAL.

The final step in our analysis uses Receiver Operating Characteristic (ROC) curves. An ROC curve characterizes the discriminating power of applying a threshold to a given variable. It presents the signal efficiency (ε_s) vs. the background efficiency (ε_b) achieved at varying threshold values. In the present case, the signal efficiency corresponds to correctly identifying s -quark jets, and the background efficiency indicates the fraction of d -quark jets incorrectly identified as s -quark jets. For each of the variables, we construct an ROC curve and compute the Area Under the Curve (AUC). The AUC serves as a metric for the discriminative power of the variable. Random guessing would give an AUC of 0.5, whereas an AUC of 1 indicates perfect discrimination, and an AUC of 0 also signifies perfect discrimination but with an opposite threshold direction. Figure 4 presents the ROC curves for the ten most discriminative jet and constituent features.¹ The neutral

¹Features of constituents beyond the fifth most energetic one are not considered for this plot since they are less likely to be meaningful as individual variables and because they are not available in all jets. However, features of all the constituents of each jet will be made available to the advanced neural networks.



(a) $p_{T,\text{jet}} > 200 \text{ GeV}$



(b) $p_{T,\text{jet}} > 45 \text{ GeV}$

Figure 3: The mean energy fractions contributed by the various types of constituents in d -quark jets (left) and s -quark jets (right) for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$, (b) $p_{T,\text{jet}} > 45 \text{ GeV}$.

hadron energy and photon energy, followed by the identity of the most energetic particle, whether it is a neutral hadron or a photon, show the highest/lowest AUC values, implying they are the most discriminative features. Figure 5 shows the distributions of the three most discriminative features identified in figure 4.

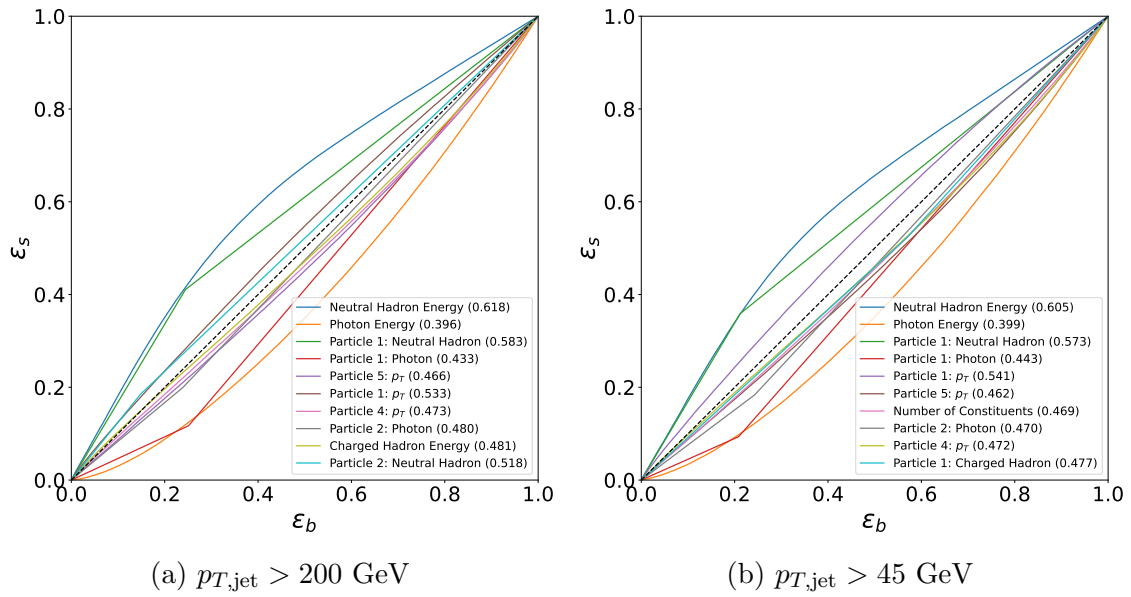
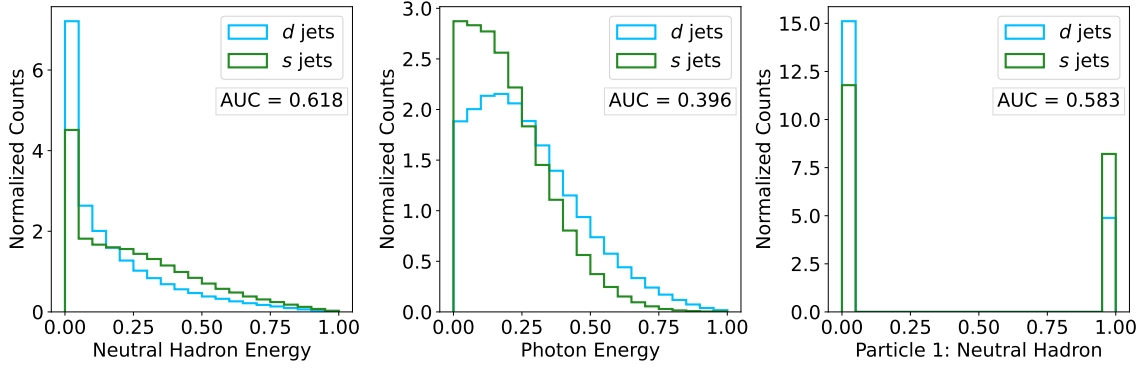
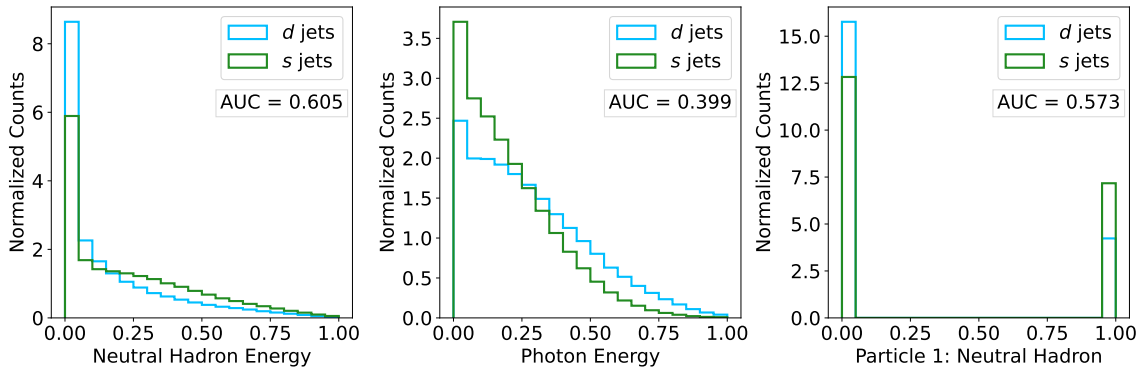


Figure 4: ROC curves for the ten most discriminative features in s -quark (signal) and d -quark (background) jets for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$ and (b) $p_{T,\text{jet}} > 45 \text{ GeV}$. Both particle and jet-level features from section 2.4 are included. The particles are numbered by decreasing p_T values. The kinks present in curves corresponding to particle identities are due to the binary nature of the variable. The AUC values are given in parentheses in the legends, where the features are ordered based on the absolute distance of their AUC from 0.5.

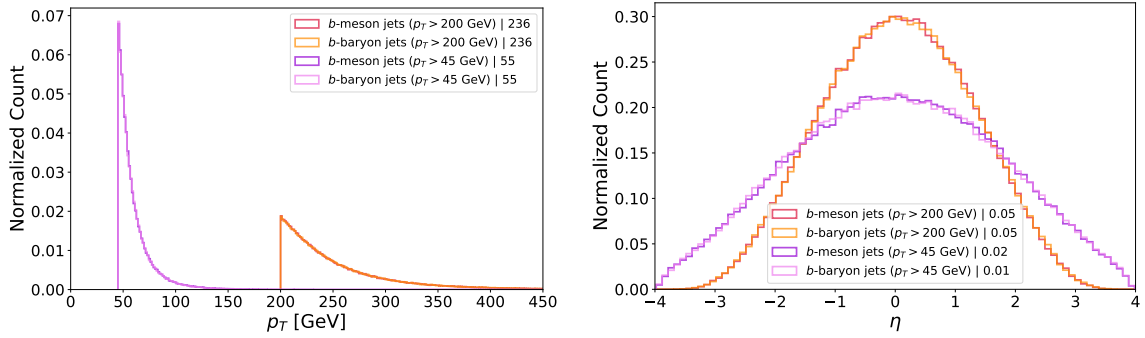


(a) $p_{T,\text{jet}} > 200$ GeV



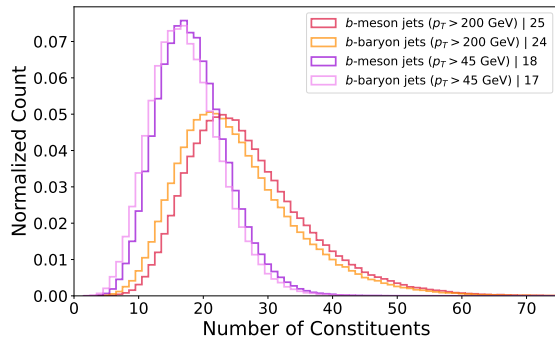
(b) $p_{T,\text{jet}} > 45$ GeV

Figure 5: Distributions of the three most discriminative features from figure 4 in d -quark and s -quark jets for (a) $p_{T,\text{jet}} > 200$ GeV, (b) $p_{T,\text{jet}} > 45$ GeV.



(a) Jet transverse momentum distributions.

(b) Jet pseudorapidity distributions.



(c) Number of constituents distributions.

Figure 6: Properties of b -meson and b -baryon jets in our samples for $p_{T,\text{jet}} > 200$ GeV and $p_{T,\text{jet}} > 45$ GeV. Median values are given in the legends.

3.2 Bottom baryon vs. meson jets

In this section, we extend our analysis to distinguishing between b -baryon and b -meson jets, which is an example of fragmentation tagging.

Figure 6 presents the distributions of $p_{T,\text{jet}}$, η_{jet} , and the number of constituents within b -meson and b -baryon jets. The distributions of $p_{T,\text{jet}}$ and η_{jet} are essentially identical between the b -meson and b -baryon jets, as expected. We also see that jets containing b mesons demonstrate a slightly higher constituent count, on average, than those containing b baryons.

Figure 7 shows the distributions of the identities of the constituents with the highest p_T in b jets containing a b baryon vs. those with a b meson. We see that neutral hadrons are more common as the leading constituents in b -baryon jets. This can be attributed to the Λ baryons and neutrons that are produced in many of the b -baryon decays, in line with baryon number conservation. While b -meson decays often produce neutral kaons, they will usually carry less energy due to their lower mass. Nevertheless, we see that reconstructed K_S decays appear more frequently as the leading constituents in b -meson jets, and reconstructed Λ decays in b -baryon jets, as expected.

Figure 7 also shows that photons are more common as the leading constituents in b -

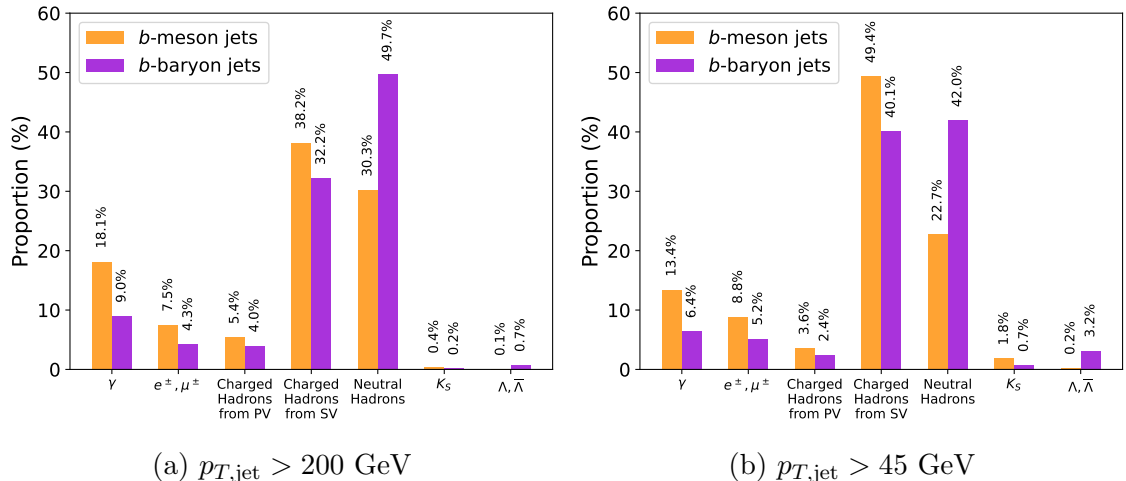


Figure 7: Distributions of the constituent types with the highest transverse momentum (p_T) in b -meson and b -baryon jets for (a) $p_{T,\text{jet}} > 200$ GeV, (b) $p_{T,\text{jet}} > 45$ GeV.

meson jets. This can be attributed to π^0 decays. While pions are common in decays of all b hadrons, the fact that b -baryon decay products necessarily include a baryon with a mass of about 1 GeV leaves less room for energetic pions. Leptons are also more common in b -meson jets. While leptons from $b \rightarrow c$ transitions are expected to contribute similarly to both types of jets, leptons from $c \rightarrow s$ transitions are less common in b -baryon decays due to the small leptonic branching ratio of the Λ_c^+ (about 4% per lepton flavor) relative to those of the D mesons (16%, 7%, and 6% per flavor for the D^+ , D^0 , and D_s^+ , respectively) [49]. They are also less energetic because the necessity of having a baryon in the final state leaves less energy available to leptons.

Figure 8 presents the mean energy fractions of each type of constituent within the jet. The behavior is similar to that observed for the most energetic constituent: the neutral hadronic energy and reconstructed Λ energy are higher in b -baryon jets, while the energy fractions in photons, leptons, and reconstructed K_S mesons are larger in b -meson jets.

Lastly, we construct ROC curves for the different features and compute the resulting AUC scores. Figure 9 presents the ten most discriminative features. The neutral hadron energy and photon energy, along with the identity of the most energetic particle in the jet (Particle 1), exhibit the highest/lowest AUC values. For low- p_T jets (figure 9b), the energy in reconstructed Λ baryons is the strongest discriminator for $\varepsilon_s \lesssim 14\%$. The efficiency here is limited by the probability for the jet to contain a Λ baryon and for its highly displaced $\Lambda \rightarrow p\pi^-$ decay to be reconstructed in the tracker. This discriminator is much less useful for high- p_T jets because the probability for the Λ to decay sufficiently early in the tracker becomes too small. Figure 10 shows the distributions of the three most discriminative features identified in figure 9.

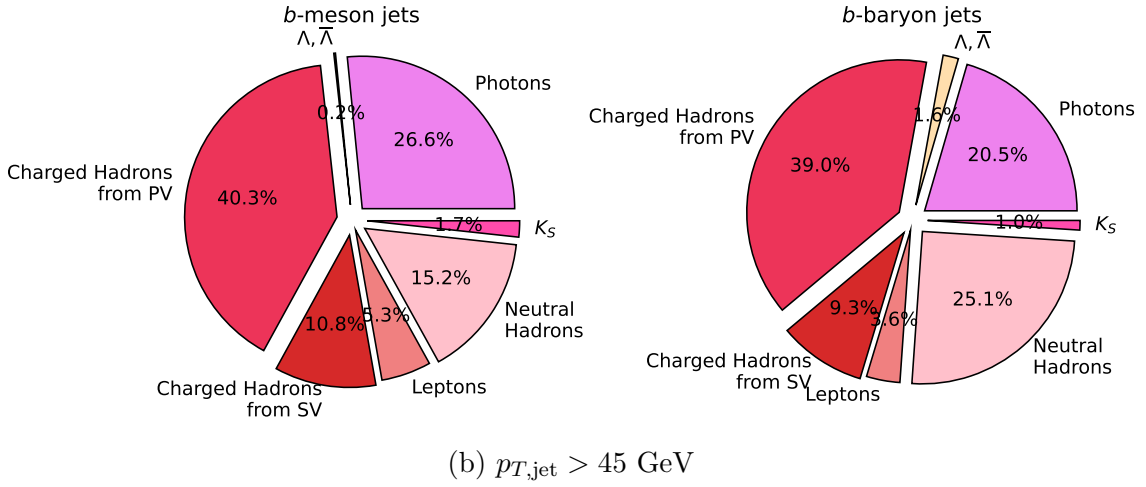
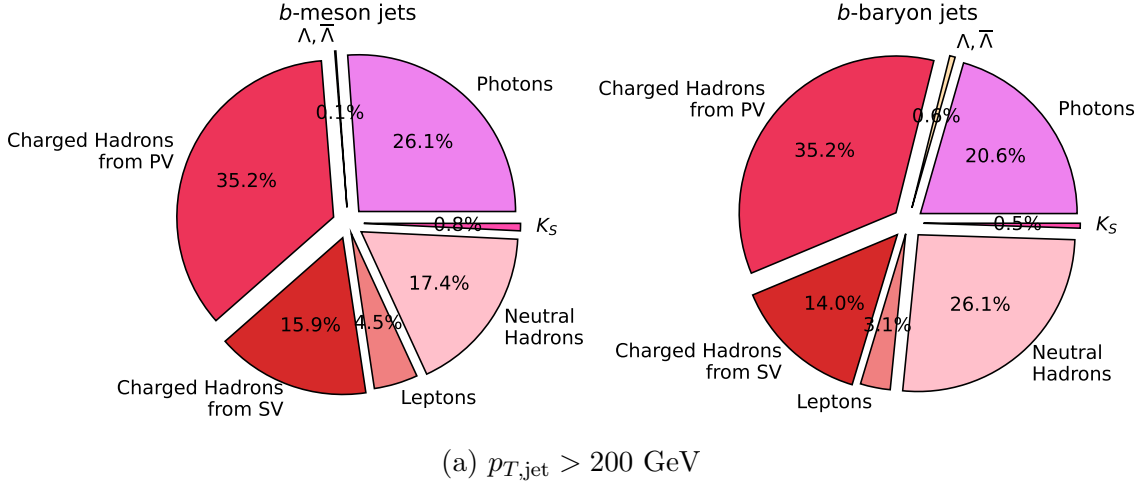


Figure 8: The mean energy fractions contributed by the various types of constituents in b -meson jets (left) and b -baryon jets (right) for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$, (b) $p_{T,\text{jet}} > 45 \text{ GeV}$. Leptons include electrons and muons.

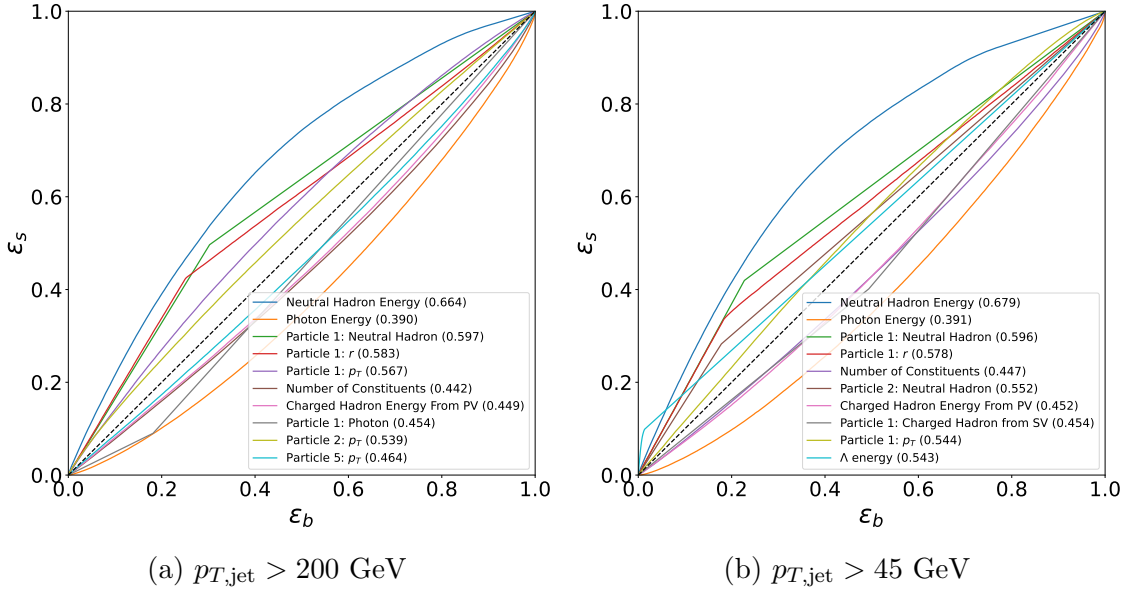


Figure 9: ROC curves for the ten most discriminative features in b -baryon (signal) and b -meson (background) jets for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$ and (b) $p_{T,\text{jet}} > 45 \text{ GeV}$. Both particle and jet-level features from section 2.4 are included. The particles are numbered by decreasing p_T values. The kinks present in curves corresponding to particle identities are due to the binary nature of the variable. The AUC values are given in parentheses in the legends, where the features are ordered based on the absolute distance of their AUC from 0.5.

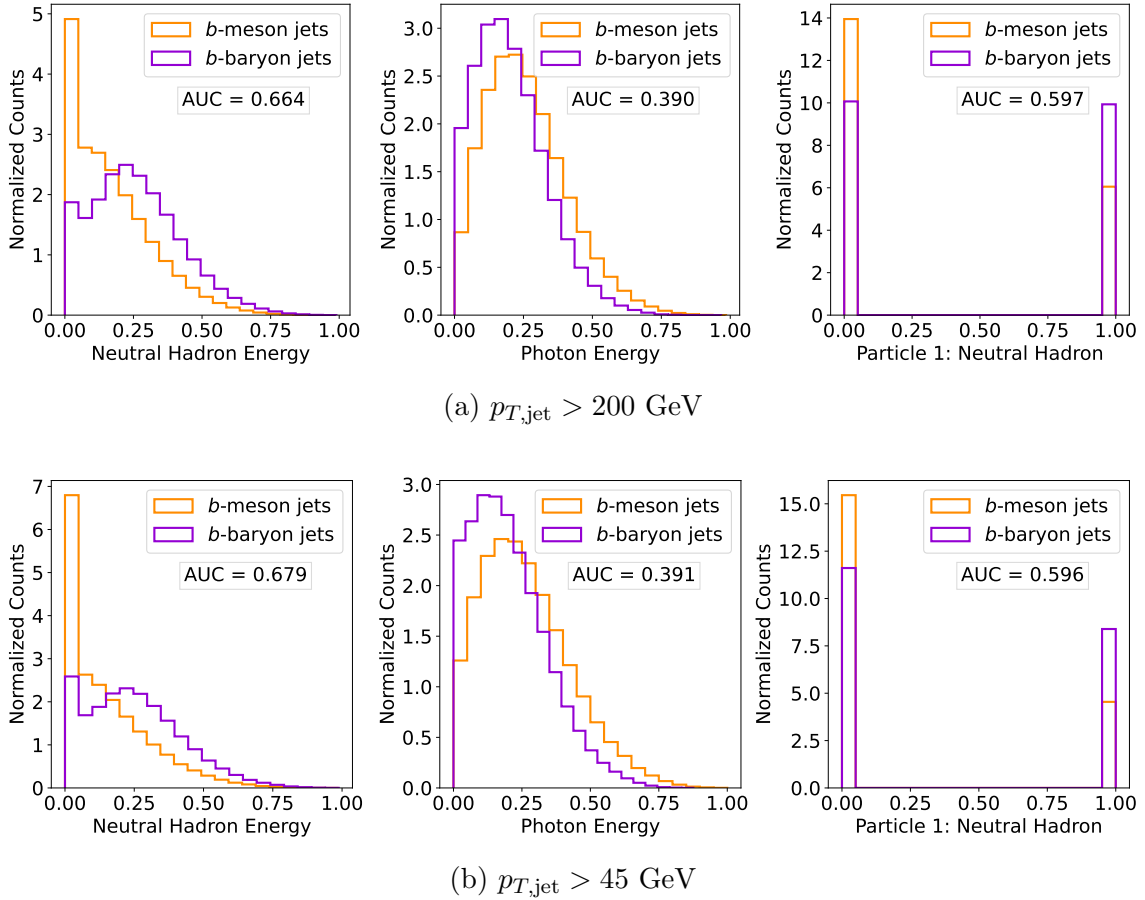


Figure 10: Distributions of the three most discriminative features from figure 9 in b -baryon and b -meson jets for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$, (b) $p_{T,\text{jet}} > 45 \text{ GeV}$.

4 ML-based taggers

We will now describe the format of the data that will be fed into the NNs and the architectures we will be using (implemented with PyTorch [59]), and then analyze the performance of the taggers.

4.1 NN inputs

Jet properties The whole-jet properties, as described in section 2.4, are the jet’s p_T , η ,² number of constituents N , and the energy fractions carried by the different types of constituents: photon energy E_γ , electron energy E_e , muon energy E_μ , charged hadron energy E_{CH} (in the b -jets case, we use two separate variables: charged hadron energy from the primary vertex $E_{\text{CH,PV}}$ and charged hadron energy from the secondary vertex $E_{\text{CH,SV}}$),

²Since we designed the samples to have similar jet p_T and η distributions, these two features are not useful for discrimination. We still provide them to the NNs since their values can be useful for interpreting some of the other features, whose distributions have some dependence on the jet p_T and η .

neutral hadron energy E_{NH} , reconstructed K_S energy E_{K_S} , and reconstructed Λ energy E_Λ (including $\bar{\Lambda}$). The jet’s p_T , η , and the number of constituents N are shifted and scaled to have a mean of 0 and a standard deviation of 1. This is done to ensure that all features are within a similar range, which is beneficial for the stability and convergence of the machine learning algorithms.

Constituent properties The constituent properties, as discussed in section 2.4, are the normalized transverse momentum ($p_{T,i}^{\text{norm}}$), the angular position relative to the jet axis in terms of r and α , and the identity of the constituent. The identity is represented by a set of discrete variables, corresponding to photons, electrons, muons, charged hadrons (in the case of the b -jets analysis, there are separate entries for charged hadrons from the primary and those from the secondary vertex), neutral hadrons, and reconstructed K_S/Λ particles. For each constituent, the corresponding entry is set to 1 if it is a positively charged particle, a neutral hadron, or a reconstructed Λ baryon, and -1 if it is a negatively charged particle or a reconstructed K_S meson, while the other entries are set to 0.

Graph representation We represent each jet as a graph, implemented with Deep Graph Library (DGL) [60]. The graph nodes represent the jet constituents. Each node’s features include the properties of both the jet and the constituent. We employ a fully-connected topology, where edges are formed between every pair of nodes. For an edge between node i and node j , a vector is initialized with a list of the jet properties, the constituent properties from node i , and the constituent properties from node j . We leave it to each of the NNs to construct useful edge features based on these physical inputs. Reverse edges (between nodes j and i), are included as well, with the order of the nodes in the vector swapped, to allow the independent flow of information in each direction. Each graph in our simulated dataset carries a label to denote the jet type: ‘0’ for d -quark jets or b -meson jets, and ‘1’ for s -quark jets or b -baryon jets.

Datasets Our strange-tagging datasets contain about one million jets, equally distributed between s -quark and d -quark jets. Our fragmentation-tagging datasets contain about one million b jets, with a distribution of 30% b -baryon and 70% b -meson jets (after we discarded a large fraction of the meson jets to avoid a bigger imbalance between the two classes due to the natural rarity of baryons). The datasets are split into training (72%), validation (18%), and testing (10%) samples.

4.2 NN architectures

One architecture we use is a variant of a Graph Attention Network (GAT) [22, 23], a type of Graph Neural Network [24, 25]. In this architecture, the node features are updated iteratively with aggregated features of all other nodes, with weights determined through an attention mechanism. In the first iteration, the aggregation weights are determined by an embedding of the physical features of the two nodes and the jet features, as described above. In subsequent iterations, the updated features of the node pairs are used to determine the weights. Finally, the features of all nodes are aggregated and processed to produce the

classifier output—a number between 0 and 1. The full structure of this NN is described in appendix A.1.

The most sophisticated NN architecture we consider is based on the idea of the Particle Transformer (ParT), introduced in ref. [29] and inspired by the famous transformer architectures [26, 27]. Its central element is the *scaled dot-product attention mechanism*, used in two ways. First, in *particle attention blocks*, learned linear projections are applied to each node’s features to produce three vectors: *query* (Q), *key* (K), and *value* (V). Each node sends its query to all other nodes. The other nodes respond with their values, with a weight that depends on the similarity (dot product) between the query and their key, as well as on the edge features. (Different from ref. [29], the edge features in our implementation are not hand-crafted physical quantities but are instead generated by the NN based on the physical properties of the two particles and the jet, as mentioned above.) The original node features are then updated based on these weighted values. This process is repeated several times. Each iteration enhances the information carried by each node as a result of its interactions with the other nodes in the graph. Moreover, several sets of queries, keys, and values, known as *heads*, operate in parallel, implementing *multihead attention*. After the particle attention blocks are completed, a *class token*—an additional node that does not represent any particle—is introduced. In *class attention blocks*, the class token sends queries to all nodes in the graph and, based on the returned weighted values, develops an understanding of the jet as a whole. This procedure is also repeated several times. The class token features are eventually processed to produce the ParT output. For many additional details, see appendix A.2.

We also implement the simplest possible NN architecture—a Multilayer Perceptron (MLP)—that is only given the whole-jet properties and the properties of the most energetic constituent. The purpose of including this architecture alongside more complex models like the GAT and ParT is to serve as a baseline for performance comparison. By evaluating how well these sophisticated NN architectures, which are given the properties of all constituents, perform against the MLP, we can see whether the tasks in question actually benefit from the architectural advantages of these more complicated models. The details of our MLP are given in appendix A.3.

4.3 Performance

We now present the results obtained with each of the tagger types for each of the classification tasks.

4.3.1 Strange tagging

Figure 11 presents the distributions of the NN outputs for the test datasets. The overlapping distributions show that it is challenging for all the models to clearly distinguish between s -quark and d -quark jets.

The ROC curves characterizing the performance of the different models are presented in figure 12. Figure 13 zooms in on the region of low (although still sizable and relevant) signal efficiencies. We can observe that if we choose, for example, to accept $\varepsilon_s = 10\%$ of

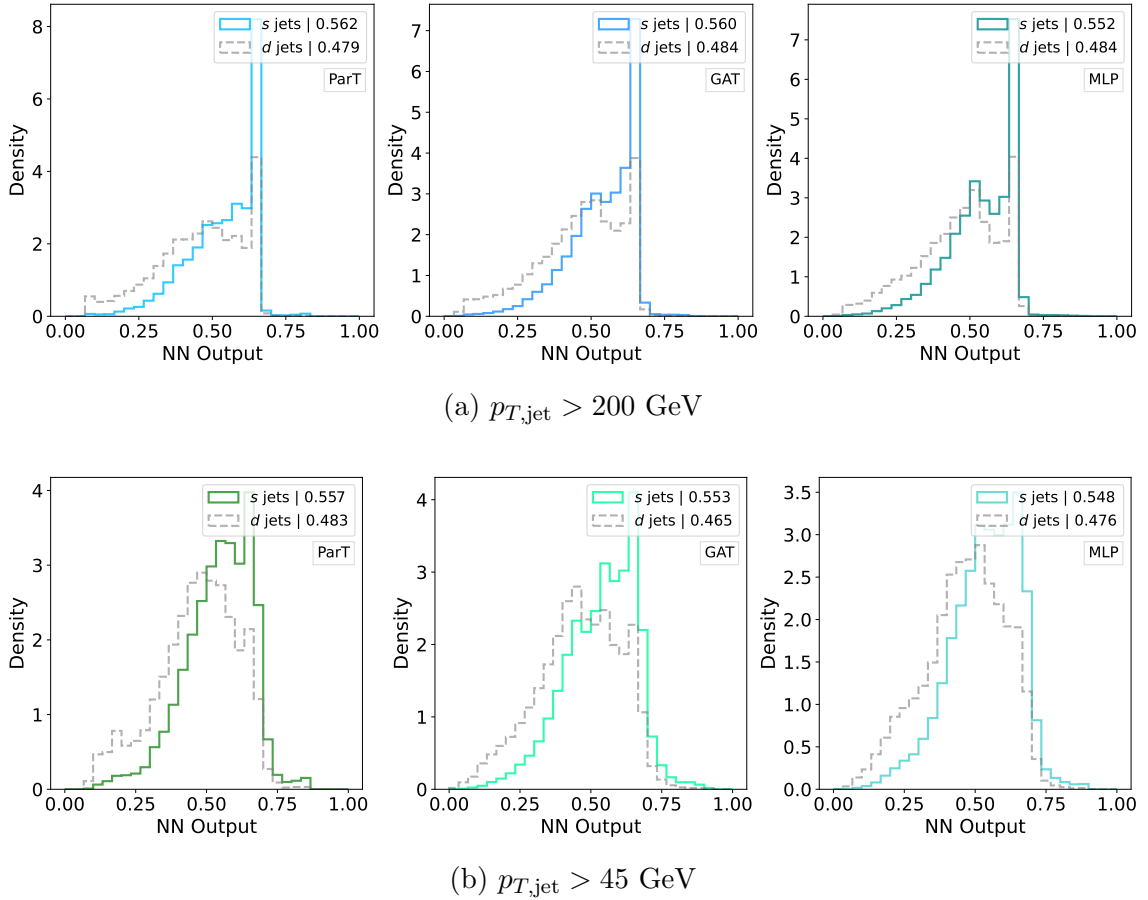


Figure 11: Distributions of the NN outputs for s -quark and d -quark jets for (a) $p_{T,\text{jet}} > 200 \text{ GeV}$, (b) $p_{T,\text{jet}} > 45 \text{ GeV}$, for ParT (left), GAT (middle) and MLP (right). The median value for each distribution is indicated in the legend.

the signal events, the background efficiency is $\varepsilon_b \approx 4\%$ for $p_{T,\text{jet}} > 45 \text{ GeV}$, and 5% for $p_{T,\text{jet}} > 200 \text{ GeV}$. In other words, the taggers improve the s/d ratio by a factor of ~ 2 .

Figure 12 shows that all the NNs outperform the most discriminative individual features (cf. figure 4). However, all the models have very similar performance, which is also quite similar to what has been obtained with the simpler architectures explored in the past: BDTs [19], CNNs [19], LSTM RNNs [20, 21] and FNNs [21]. The sophisticated GAT and ParT architectures do not bring any improvement in performance, suggesting that the jet data does not contain much useful information beyond what can be captured by a simple combination of hand-crafted variables. This leads us to believe that achieving significantly better strange-tagging performance with the ATLAS and CMS detectors is unlikely, at least with the physical inputs that we assumed to be available and potentially relevant.

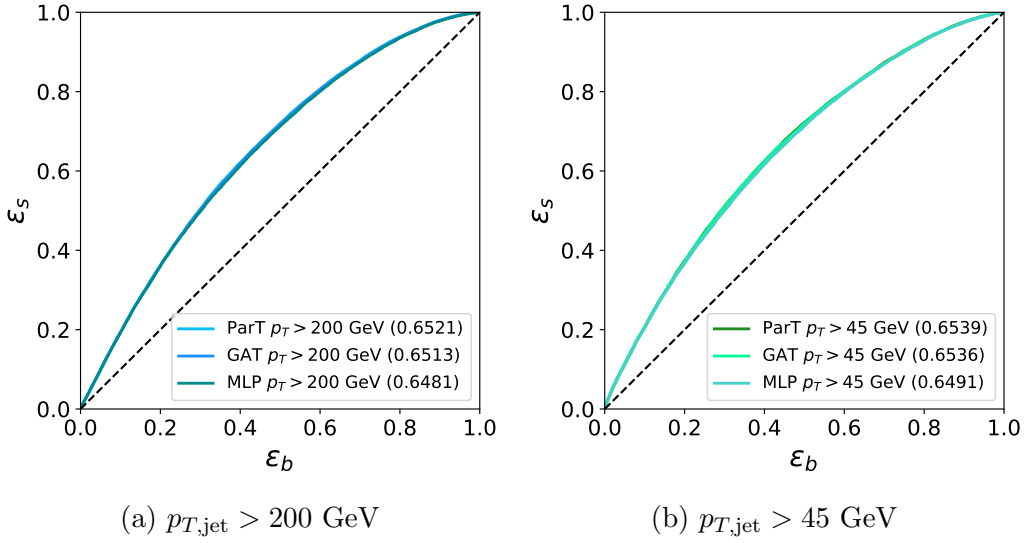


Figure 12: The strange-tagging ROC curves for the different architectures we have used: ParT, GAT, and MLP, for (a) $p_{T,jet} > 200$ GeV, and (b) $p_{T,jet} > 45$ GeV. The plots show the signal efficiency (ϵ_s), which is the fraction of s -quark jets passing the threshold on the NN output, as a function of the background efficiency (ϵ_b), indicating the fraction of d -quark jets incorrectly identified as s -quark jets by the model. The AUC values are given in parentheses in the legends.

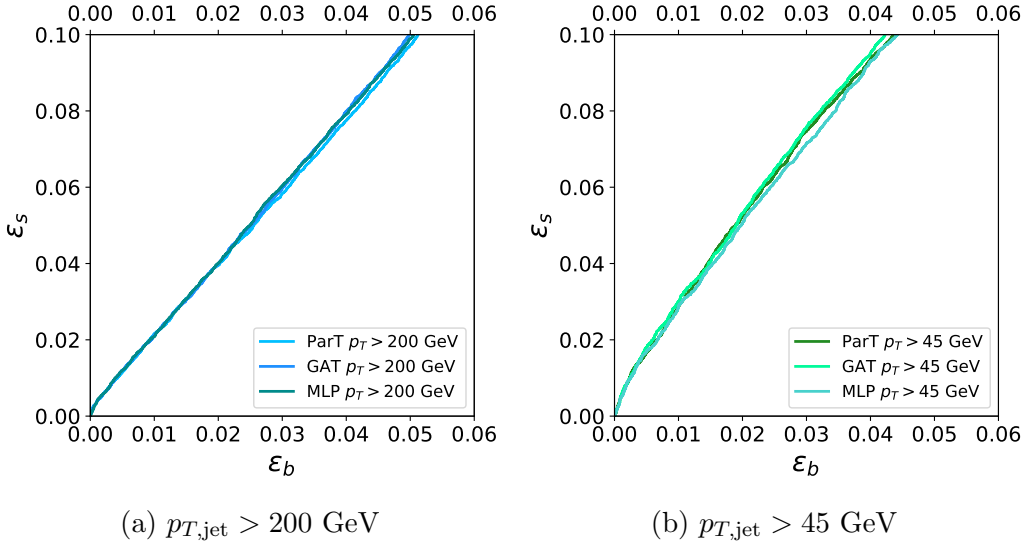


Figure 13: Zoomed-in ROC curves of figure 12 for s -quark vs. d -quark jet classification by the NN models for (a) $p_{T,jet} > 200$ GeV, (b) $p_{T,jet} > 45$ GeV.

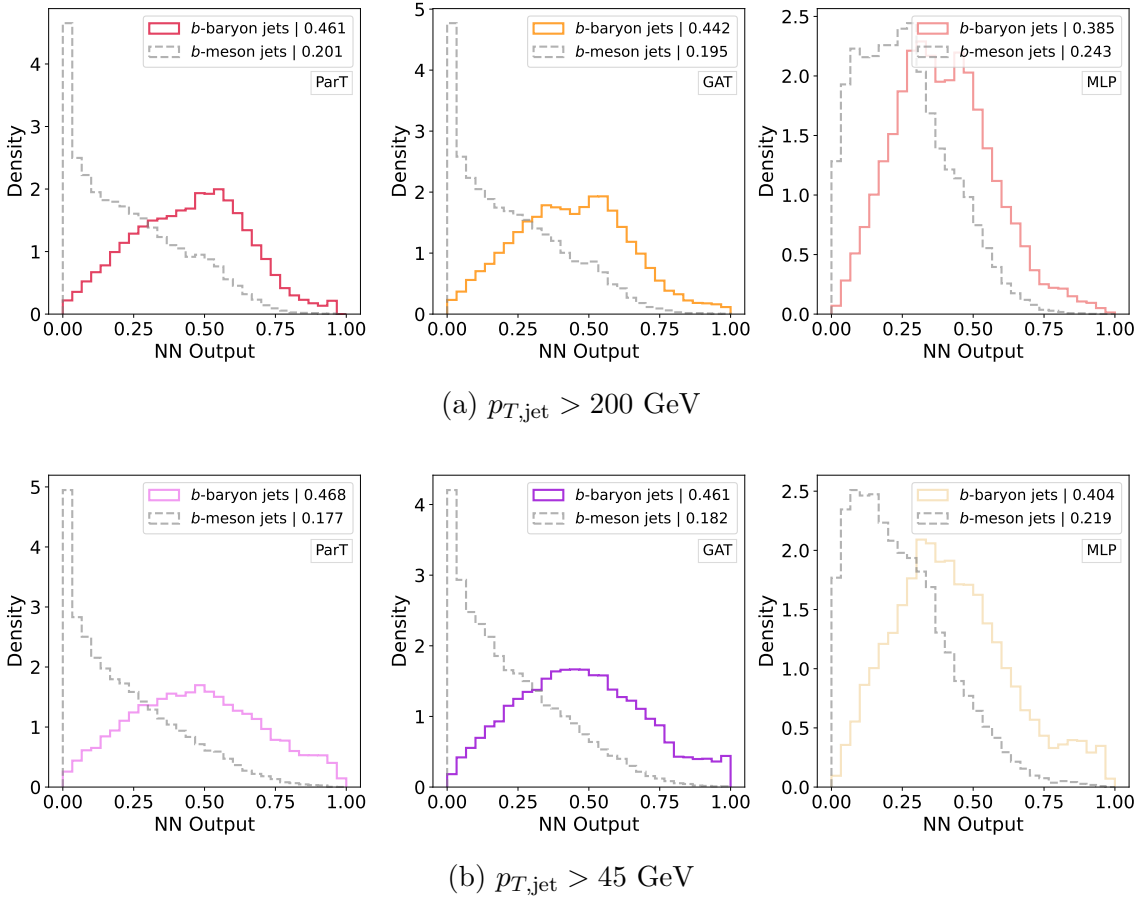


Figure 14: Distributions of the NN outputs for b -baryon and b -meson jets for (a) $p_{T,\text{jet}} > 200$ GeV, (b) $p_{T,\text{jet}} > 45$ GeV, for ParT (left), GAT (middle) and MLP (right). The median value for each distribution is indicated in the legend.

4.3.2 Fragmentation tagging

The distributions of the NN outputs from the b -baryon/ b -meson taggers are presented in figure 14. All the models show potential for a decent level of discrimination if one does not insist on having $\mathcal{O}(1)$ efficiencies.

The corresponding ROC curves and their AUC values are presented in figure 15, where b -baryon jets are taken to be the signal. The GAT and ParT models demonstrate similar performance, which is significantly better than that obtained with the MLP, which in turn is significantly better than that obtained with any individual feature (cf. figure 9).

We zoom in to low efficiencies in figure 16. We see that for a signal efficiency of $\varepsilon_s = 10\%$, for $p_{T,\text{jet}} > 200$ GeV, the ParT and GAT models have a background efficiency of only $\varepsilon_b \approx 1.25\%$, which is better than the MLP by a factor of 1.4. For $p_{T,\text{jet}} > 45$ GeV, the ParT model has a background efficiency as low as $\varepsilon_b \approx 0.67\%$, which is slightly better than the GAT, and a factor of 1.6 better than in the MLP case. Relative to the original samples, the best taggers improve the baryon-to-meson ratio at this signal efficiency by a

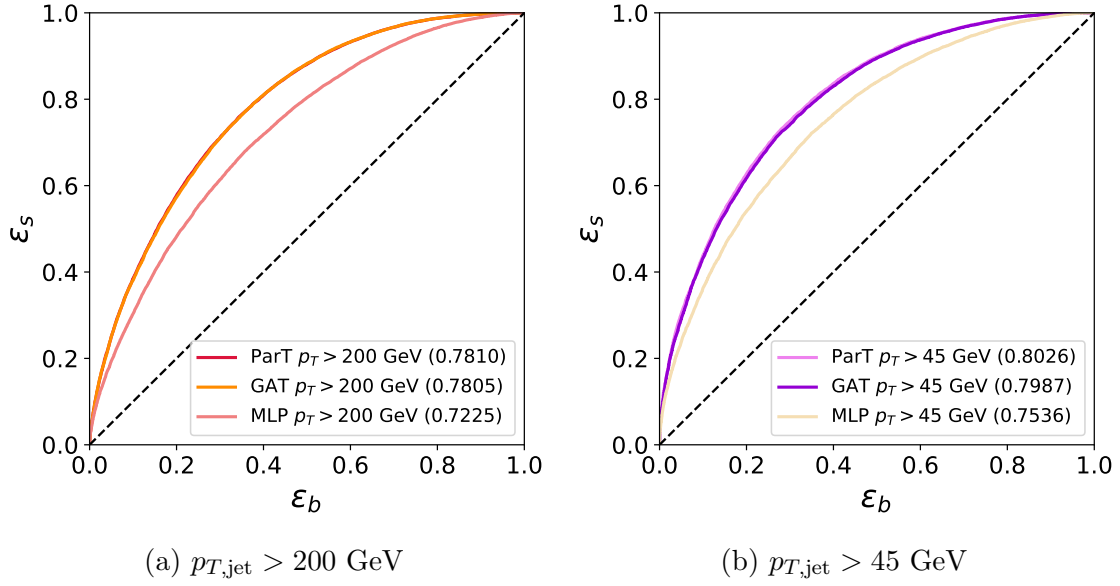


Figure 15: The fragmentation-tagging ROC curves for the different architectures we have used: ParT, GAT, and MLP, for (a) $p_{T,\text{jet}} > 200$ GeV, and (b) $p_{T,\text{jet}} > 45$ GeV. The plots show the signal efficiency (ϵ_s), which is the fraction of b -baryon jets passing the threshold on the NN output, as a function of the background efficiency (ϵ_b), indicating the fraction of b -meson jets incorrectly identified as b -baryon jets by the model. The AUC values are given in parentheses in the legends.

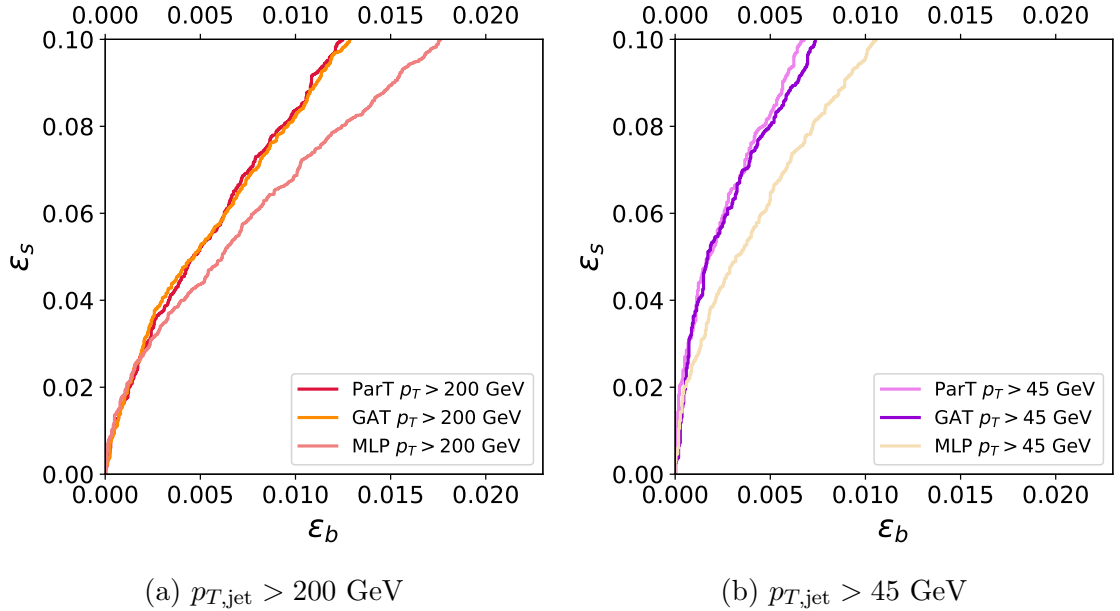


Figure 16: Zoomed-in ROC curves of figure 15 for b -baryon vs. b -meson jet classification by the NN models for (a) $p_{T,\text{jet}} > 200$ GeV, (b) $p_{T,\text{jet}} > 45$ GeV.

factor of 8 in the $p_{T,\text{jet}} > 200$ GeV case, and 15 in the $p_{T,\text{jet}} > 45$ GeV case.

4.4 Robustness to measurement errors

To assess the robustness of our results against detector resolution effects (which were not initially simulated), we conducted a test by introducing a random 5% measurement error to the transverse momenta of all jet constituents. We then trained our models on this modified data and subsequently tested them on similarly altered data. By comparing the ROC curves and AUC values with and without these induced errors, we observed no significant sensitivity to these effects.

5 Summary and discussion

Jet identification is a broad and imperfectly solved problem in collider physics. In this paper, we addressed two different cases of jet identification at the LHC using variables that are measurable in the ATLAS and CMS detectors.

The first case is *strange tagging*, where we considered the most challenging scenario, which is differentiating between jets originating from strange and down quarks. A strange-quark tagger can in principle be useful for measuring the CKM matrix elements V_{ts} and V_{td} in top-quark decays, and V_{cs} and V_{cd} in W decays. It can also improve the kinematic reconstruction of top-quark decays. Additionally, it can increase the sensitivity to new physics scenarios that involve strange-quark production. Building a strange tagger has been attempted several times in the past, including with machine learning techniques, with a moderate level of success [18–21]. In this work, we approached the same problem with different and more sophisticated neural network architectures.

The second case is *fragmentation tagging*, which in our example differentiates between bottom-baryon jets and bottom-meson jets, but would more generally identify the particular bottom hadron that was produced in the jet. This approach can also be extended to other quark flavors. A fragmentation tagger can be useful for more inclusive measurements of fragmentation functions and for reducing background in the proposed b -quark polarization and spin correlation measurements [41–43]. To our knowledge, this work is the first attempt at using machine learning for inclusive fragmentation tagging.

The problem of fragmentation tagging is similar to that of discriminating between s and d -quark jets. First, in both problems, the pattern of parton showering is the same in the two classes, and therefore cannot be used for discrimination. In addition, the basic properties of the displaced vertex are the same for the different b hadrons, which is analogous to the absence of such a typical displaced vertex in both s and d jets. On the other hand, a remaining handle that can be used in both problems is the different probabilities for the appearance of the various final-state particles in the jet and their kinematics. These stem from the differences in the hadronization processes in the different jets, followed by hadron decays.

In both cases, we structure the jet into a graph format, which represents jet constituents by nodes and uses the properties of each two constituents, combined with the properties of the jet as a whole, as input for edges connecting the corresponding pair of nodes. Such a

representation enables the employment of more advanced architectures like the GAT and ParT. These neural networks attempt to identify complex patterns distinguishing between the jets to achieve the desired classification.

In strange tagging, we found that the GAT and ParT architectures provided with the full jet constituent data did not perform significantly better than a simple MLP that combined the features of the jet and the leading constituent. This extends the result of ref. [19], where it was shown that CNNs applied to jet images did not significantly outperform BDTs that used a small number of key whole-jet variables or even a single hand-crafted variable.

Fragmentation tagging, on the other hand, shows promise, especially with the more advanced NN architectures. Our results call for a variety of further studies. These include refining the classification to specific b hadrons, applying the classification to subsets of jets (e.g., those with semileptonic decays), as motivated by particular applications, and extending the framework to other quark flavors. This will likely motivate additional types of input features. Another important question to address is the systematic uncertainties associated with reliance on simulation (in our case, Pythia). It would also be beneficial to develop a scheme to train, or at least calibrate, the classifiers on experimental data. We hope to address some of these questions in future work.

Acknowledgments

This research was supported by the Israel Science Foundation (grant no. 1666/22) and the United States—Israel Binational Science Foundation (grant no. 2018257).

A NN details

This appendix provides the detailed descriptions of the NNs implemented in this work.

A.1 Graph Attention Network (GAT)

Our Graph Attention Network (GAT) architecture, inspired by refs. [22, 23], consists of node and edge embeddings, a graph attention block that iterates n times, and a linear block.

Node and edge embeddings To transform the physical input features of each node into a more useful representation, we start with Batch Normalization (BN) [61], and then employ three MLP layers with (128, 512, 128) neurons, with each of the layers preceded by Layer Normalization (LN) [62] and followed by the GELU activation function [63]. Similarly, for the edge features, we start with BN and then employ three MLP layers with (32, 64, 16) neurons, applying GELU and LN between the layers, and concluding with BN followed by the sigmoid activation function, which is applied to each of the final layer neurons to produce 16 edge weights within the range $(0, 1)$.

Graph attention block This block updates each node’s features by aggregating the features of all other nodes. The 128 features of each node are divided into 16 groups of 8 features, each corresponding to one of the 16 edge weights. Each weight is applied to aggregate the features within its respective group. The node’s own feature vector is also included in the sum, which is then divided by the total number of nodes. The resulting vector is transformed through an MLP block comprising two layers, where the first layer doubles the feature dimension, and the second layer restores it to the original dimension of 128, with GELU and LN applied after the first layer. Subsequently, a residual connection [64] followed by a dropout [65] is applied.

The edge weights are then recalculated by attending to the updated node features. First, the features of each pair of nodes i and j are concatenated to create new edge features $w_{ij} = [h_i, h_j]$ (where h_i and h_j are the updated node features). These edge features are then transformed through two MLP layers with (64, 16) neurons, with a GELU activation function and LN between the layers, and concluding with the sigmoid activation function applied to each neuron to produce the new weights (attention coefficients).

The entire graph attention block iterates $n = 10$ times.

Linear block After the graph attention blocks, the features are averaged across all nodes to form a single vector representing the entire graph, h_{mean} . The features are also summed across all nodes to form another single vector, h_{sum} . We then concatenate these vectors into $h = [h_{\text{mean}}, h_{\text{sum}}]$. This concatenated vector h is processed through a linear layer comprising 64 neurons, followed by a GELU activation function. Subsequently, a second linear layer with a single neuron and the sigmoid activation function produces a number within the range (0, 1) as the NN output.

A.2 Particle Transformer (ParT)

In this section, we review the *scaled dot-product attention mechanism* from ref. [26] and the Particle Transformer (ParT) model of ref. [29].

Attention mechanism The scaled dot-product attention mechanism starts by transforming each element (*token*) of the input data through learned linear projections into three vectors: a *query* Q , a *key* K , and a *value* V . Conceptually, the query is like a ‘search term’ for identifying relevant information, the key acts as a ‘label’ for each data point, and the value contains the actual data that the mechanism ultimately focuses on based on the computed relevance. Attention scores are computed by calculating the dot product of the query of one token and the key of another, scaling it by $\sqrt{d_k}$,³ where d_k is the dimensionality of each key and query, and applying the softmax function to obtain the weights that will eventually multiply the values:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (\text{A.1})$$

³The scaling by $\sqrt{d_k}$ is done to prevent the dot product from becoming too large in magnitude (which would force the softmax function to give a number very close to 1), which can cause vanishing gradients during backpropagation [26].

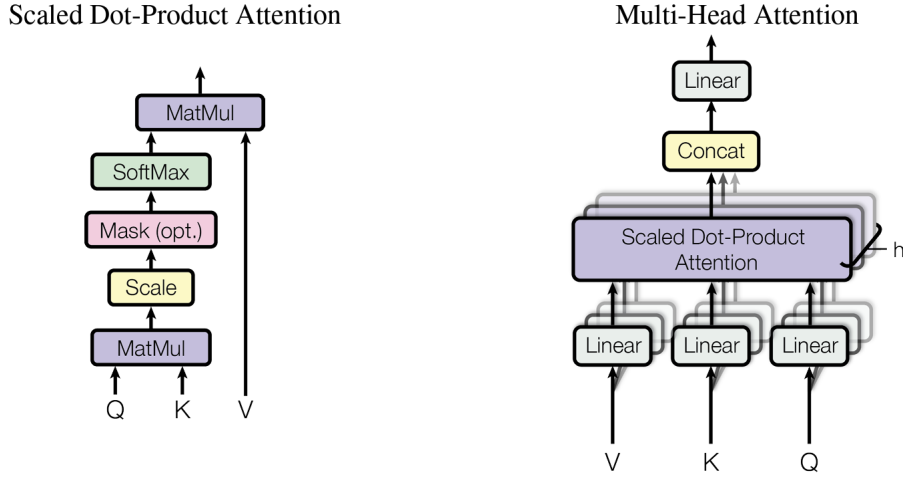


Figure 17: The Scaled Dot-Product Attention (left) and Multi-Head Attention mechanisms (right) from ref. [26].

as illustrated on the left side of figure 17.

The right side of figure 17 also depicts how the basic attention mechanism is expanded in *Multi-Head Attention (MHA)*. Rather than processing a single set of Q , K , and V , MHA enhances this approach by employing multiple instances of the attention mechanism (“heads”) in parallel enabling the simultaneous processing of the data with several different sets of weights. This design allows the model to focus on various aspects of the input simultaneously. The results from the different heads are then concatenated and linearly transformed to produce the final output.

The variables Q , K , and V in eq. (A.1) are tensors of dimension (H, N, d_k) ,⁴ where H is the number of heads and N is the number of tokens (which is the number of nodes in the context of GNNs). The number of features, d_k , is taken to be the input dimension divided by the number of heads, to facilitate incorporating residual connections. The dot product (in the feature space for each pair of nodes and each head) QK^T produces a tensor of dimensions (H, N, N) . The softmax function is applied over the last dimension of this tensor. The resulting attention scores are multiplied by the value tensor, producing a tensor with dimensions (H, N, d_k) . To make this more explicit, we can write eq. (A.1) in components as follows:

$$\text{Attention}(Q, K, V)_{hni} = \sum_m \text{softmax}_m \left(\frac{1}{\sqrt{d_k}} \sum_j Q_{hnj} K_{hmj} \right) V_{hmi}. \quad (\text{A.2})$$

Particle Transformer (ParT) architecture As illustrated in figure 18, the architecture is segmented into four parts: node and edge embeddings, particle attention blocks, class attention blocks, and MLP. Firstly, the node and edge features are processed in the

⁴In general, the feature dimensionality in V can be different from those of Q and K , but we will not be using this freedom.

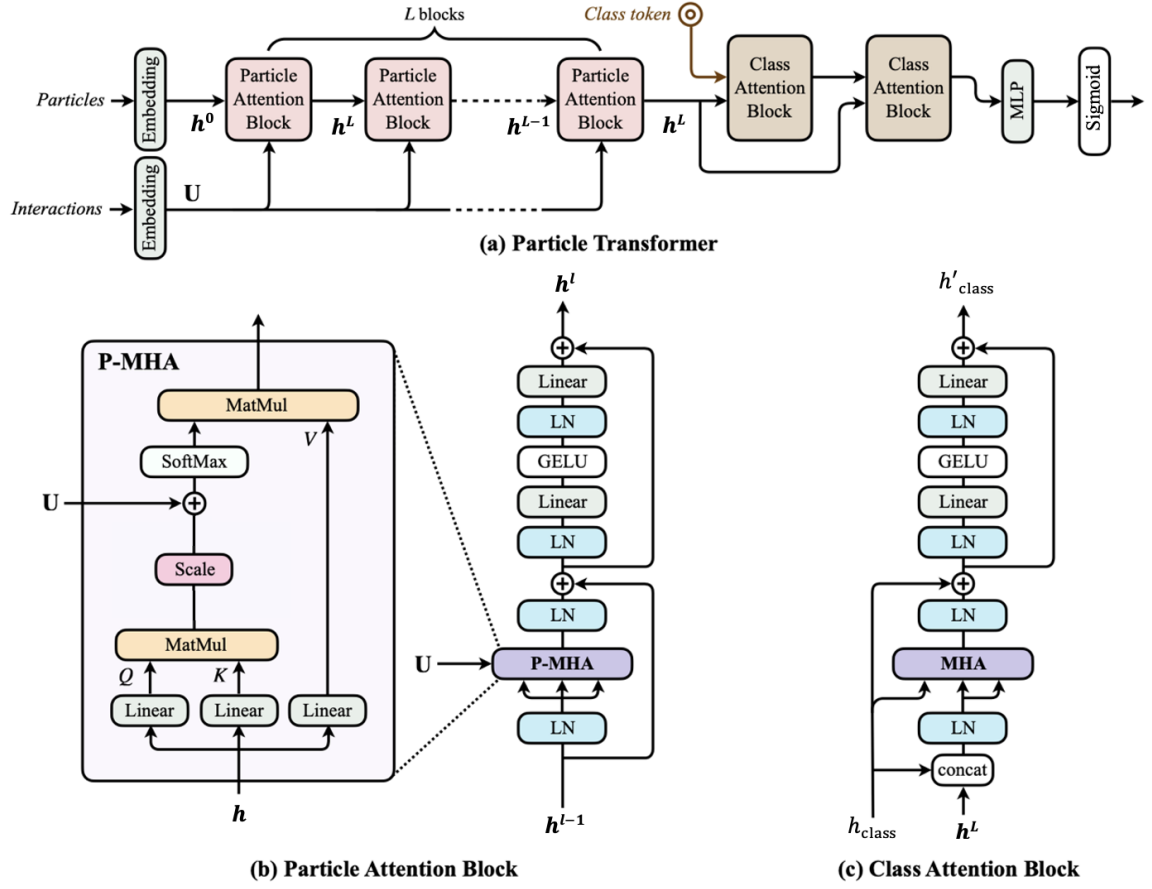


Figure 18: The Particle Transformer (ParT) architecture adapted from ref. [29]. Figure (a) illustrates the overall architecture of ParT, showing how the data is processed through sequential blocks. Figure (b) details the particle attention block, which includes the pairwise multi-head attention (P-MHA) mechanism and linear transformations for feature processing. Figure (c) describes the class attention block, where a class token is integrated to extract global information from the particle nodes via the MHA mechanism and linear transformations.

node/edge embedding stage. The output subsequently passes through the particle attention block L times. Following that, the output progresses through the class attention block M times. Ultimately, it is processed by the MLP, and after the final linear layer, the sigmoid function is activated to produce a value ranging from 0 to 1.

Node and edge embeddings The embedding procedure maps the physical data into a new vector space, using an MLP for the node features and a convolutional approach for the edge features. The node features, after Batch Normalization (BN) [61], are passed through three linear layers with 128, 512, and 128 neurons, with each of the layers preceded by Layer Normalization (LN) [62] and followed by a GELU activation function [63]. The edge

features, after BN, are processed through four layers of pointwise 1D convolution⁵ with 64, 64, 64, and 8 channels, with BN and GELU after each layer, except for the last layer, where GELU is not applied. The resulting edge features for each pair of nodes⁶ form the so-called *interaction matrix* U .

Particle attention block The particle attention block comprises two main parts, as described in figure 18(b). The first part consists of the *pairwise multi-head attention* (P-MHA) with LN layers before and after the P-MHA. The P-MHA mechanism is similar to the MHA mechanism described above, except that the pairwise interaction matrix U is incorporated as a bias,

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} + U \right) V, \quad (\text{A.3})$$

to integrate information about the relationships between nodes. The dimension compatibility between U and QK^T is achieved by ensuring that the number of edge features (after the embedding) aligns with the number of heads. Our P-MHA is configured with 8 heads. The same U is used across all particle attention blocks. A dropout [65] is applied after the softmax.

The second part of the particle attention block is constructed from an MLP with two linear layers, each preceded by LN, with a GELU activation function and a dropout between the layers. The first linear layer projects the input into a dimensionality of 512, while the second transforms it back to the original input dimension.

Residual connections [64], preceded by dropouts, are included after each of these two parts.

The particle attention block is repeated $L = 8$ times to deepen the network’s ability to learn complex patterns.

Class attention block The core concept of the class attention blocks is to generate a global graph-level representation. This is achieved by introducing a *class token* h_{class} [27], which is a single node, not corresponding to any particular jet constituent, with the same number of features as the constituent nodes \mathbf{h} . The features of h_{class} are initialized before the first class attention block by learnable parameters. Subsequently, the class token computes graph-level features by sending queries to the nodes \mathbf{h} (and to itself).

The structure of the class attention block, which is described in figure 18(c), is similar to that of the particle attention block, with the same hyperparameters, except that dropouts are not included. The main difference is that instead of the P-MHA, the standard MHA described at the beginning of this section is employed, with Q , K , and V computed as

$$Q = W_q h_{\text{class}} + b_q, \quad (\text{A.4})$$

$$K = W_k \mathbf{z} + b_k, \quad (\text{A.5})$$

$$V = W_v \mathbf{z} + b_v. \quad (\text{A.6})$$

⁵While the MLP and the pointwise convolution act similarly, we stick to the methodology presented in [29].

⁶Edges from a node to itself are not included.

Here, $\mathbf{z} = [h_{\text{class}}, \mathbf{h}]$ is the concatenation of h_{class} and \mathbf{h} , where \mathbf{h} is the output of the last particle attention block, and W_i and b_i with $i \in \{q, k, v\}$ are learnable weights and biases, respectively.

The class attention block is iterated $M = 2$ times. In each iteration, the particle embeddings \mathbf{h} (the output from the last particle attention block) remain unchanged, while h_{class} is updated. This process allows the class token to iteratively extract information from the particle embeddings through multiple class attention blocks.

Final MLP layers The final stages of the model comprise two linear layers to process the output from the class attention blocks and generate the final prediction. The representation obtained from the class attention blocks is passed through a linear layer, followed by a GELU activation function, to transform it into a lower-dimensional space with 64 neurons. This is followed by a second linear transformation, reducing the dimensions to a single number. Finally, the sigmoid activation function is applied, which brings the output to the range $(0, 1)$, forming the model prediction.

A.3 Multilayer Perceptron (MLP)

The MLP we use comprises five layers with (128, 256, 512, 64, 1) neurons, with a GELU activation function and LN after each layer, except for the final one, where the sigmoid activation function is used.

A.4 Hyperparameters

For all models presented, including the ParT, the GAT, and the MLP, we have standardized the initialization of hyperparameters. The batch size is set to 128. The loss function is Binary Cross-Entropy (BCELoss). All the dropout rates are 0.1. The Adam optimizer [66] is employed with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} . A learning rate scheduler is utilized to reduce the learning rate by a factor of 0.1 if there is no improvement in the validation loss for a duration of 10 epochs. The training is ended when there is no improvement in the validation loss over 20 epochs. The inference is done with weights from the point with the lowest validation loss. For strange tagging with $p_{T,\text{jet}} > 200$ GeV, this point for the ParT, GAT, and MLP models was at 27, 28, and 36 epochs, respectively, while for $p_{T,\text{jet}} > 45$ GeV, it was at 13, 36, and 29 epochs, respectively. For fragmentation tagging with $p_{T,\text{jet}} > 200$ GeV, the lowest validation loss for the ParT, GAT, and MLP models was attained at 26, 40, and 41 epochs, respectively, while for $p_{T,\text{jet}} > 45$ GeV, it was at 31, 29, and 31 epochs, respectively.

References

- [1] S. Mondal and L. Mastrolorenzo, “Machine Learning in High Energy Physics: A review of heavy-flavor jet tagging at the LHC,” *Eur. Phys. J. Spec. Top.* (2024), [arXiv:2404.01071 \[hep-ex\]](#).
- [2] A. Butter *et al.*, “The Machine Learning landscape of top taggers,” *SciPost Phys.* **7** (2019) 014, [arXiv:1902.09914 \[hep-ph\]](#).

- [3] ATLAS Collaboration, G. Aad *et al.*, “ATLAS flavour-tagging algorithms for the LHC Run 2 pp collision dataset,” *Eur. Phys. J. C* **83** (2023) 681, [arXiv:2211.16345](#) [[physics.data-an](#)].
- [4] CMS Collaboration, A. M. Sirunyan *et al.*, “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV,” *JINST* **13** (2018) P05011, [arXiv:1712.07158](#) [[physics.ins-det](#)].
- [5] R. Kogler, B. Nachman, A. Schmidt, *et al.*, “Jet Substructure at the Large Hadron Collider: Experimental Review,” *Rev. Mod. Phys.* **91** (2019) 045003, [arXiv:1803.06991](#) [[hep-ex](#)].
- [6] D. Krohn, M. D. Schwartz, T. Lin, and W. J. Waalewijn, “Jet Charge at the LHC,” *Phys. Rev. Lett.* **110** (2013) 212001, [arXiv:1209.2421](#) [[hep-ph](#)].
- [7] J. Thaler and K. Van Tilburg, “Identifying Boosted Objects with N -subjettiness,” *JHEP* **03** (2011) 015, [arXiv:1011.2268](#) [[hep-ph](#)].
- [8] T. Plehn and M. Spannowsky, “Top Tagging,” *J. Phys. G* **39** (2012) 083001, [arXiv:1112.4441](#) [[hep-ph](#)].
- [9] DELPHI Collaboration, P. Abreu *et al.*, “First measurement of the strange quark asymmetry at the Z^0 peak,” *Z. Phys. C* **67** (1995) 1–13.
- [10] DELPHI Collaboration, P. Abreu *et al.*, “Measurement of the strange quark forward-backward asymmetry around the Z^0 peak,” *Eur. Phys. J. C* **14** (2000) 613–631.
- [11] OPAL Collaboration, K. Ackerstaff *et al.*, “Measurement of the branching fractions and forward-backward asymmetries of the Z^0 into light quarks,” *Z. Phys. C* **76** (1997) 387–400, [arXiv:hep-ex/9707019](#).
- [12] SLD Collaboration, K. Abe *et al.*, “First direct measurement of the parity violating coupling of the Z^0 to the s quark,” *Phys. Rev. Lett.* **85** (2000) 5059–5063, [arXiv:hep-ex/0006019](#).
- [13] J. Duarte-Campderros, G. Perez, M. Schlaffer, and A. Soffer, “Probing the Higgs – strange-quark coupling at e^+e^- colliders using light-jet flavor tagging,” *Phys. Rev. D* **101** (2020) 115005, [arXiv:1811.09636](#) [[hep-ph](#)].
- [14] A. Albert *et al.*, “Strange quark as a probe for new physics in the Higgs sector,” in *Proceedings of Snowmass 2021*. [arXiv:2203.07535](#) [[hep-ex](#)].
- [15] F. Blekman, F. Canelli, A. De Moor, K. Gautam, A. Ilg, A. Macchiolo, and E. Ploerer, “Jet Flavour Tagging at FCC-ee with a Transformer-based Neural Network: DeepJetTransformer,” [arXiv:2406.08590](#) [[hep-ex](#)].
- [16] ATLAS Collaboration, G. Aad *et al.*, “The ATLAS Experiment at the CERN Large Hadron Collider,” *JINST* **3** (2008) S08003.
- [17] CMS Collaboration, S. Chatrchyan *et al.*, “The CMS Experiment at the CERN LHC,” *JINST* **3** (2008) S08004.
- [18] J. Erdmann, “A tagger for strange jets based on tracking information using long short-term memory,” *JINST* **15** (2020) P01021, [arXiv:1907.07505](#) [[physics.ins-det](#)].
- [19] Y. Nakai, D. Shih, and S. Thomas, “Strange Jet Tagging,” [arXiv:2003.09517](#) [[hep-ph](#)].
- [20] J. Erdmann, O. Nackenhorst, and S. V. Zeißner, “Maximum performance of strange-jet tagging at hadron colliders,” *JINST* **16** (2021) P08039, [arXiv:2011.10736](#) [[hep-ex](#)].

- [21] S. V. Zeiner, *Development and calibration of an s-tagging algorithm and its application to constrain the CKM matrix elements $|V_{ts}|$ and $|V_{td}|$ in top-quark decays using ATLAS Run-2 Data*. PhD thesis, Dortmund U., 2021.
- [22] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Li, and Y. Bengio, “Graph Attention Networks,” in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. 2018. [arXiv:1710.10903 \[stat.ML\]](#).
- [23] S. Brody, U. Alon, and E. Yahav, “How Attentive are Graph Attention Networks?,” in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*. 2022. [arXiv:2105.14491 \[cs.LG\]](#).
- [24] J. Shlomi, P. Battaglia, and J.-R. Vlimant, “Graph neural networks in particle physics,” *Mach. Learn.: Sci. Technol.* **2** (2021) 021001, [arXiv:2007.13681 \[hep-ex\]](#).
- [25] G. DeZoort, P. W. Battaglia, C. Biscarat, and J.-R. Vlimant, “Graph neural networks at the Large Hadron Collider,” *Nature Rev. Phys.* **5** (2023) 281–303.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30. 2017. [arXiv:1706.03762 \[cs.CL\]](#).
- [27] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jgou, “Going deeper with image transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42. 2021. [arXiv:2103.17239 \[cs.CV\]](#).
- [28] OpenAI, “ChatGPT (GPT-4),” 2024. <https://chat.openai.com/>.
- [29] H. Qu, C. Li, and S. Qian, “Particle Transformer for Jet Tagging,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *PMLR*, pp. 18281–18292. 2022. [arXiv:2202.03772 \[hep-ph\]](#).
- [30] ATLAS Collaboration, “Constituent-Based Quark Gluon Tagging using Transformers with the ATLAS detector,” Tech. Rep. ATL-PHYS-PUB-2023-032, CERN, Geneva, 2023. <https://cds.cern.ch/record/2878932>.
- [31] M. Usman, M. H. Shahid, M. Ejaz, U. Hani, N. Fatima, A. R. Khan, A. Khan, and N. M. Mirza, “Particle Multi-Axis Transformer for Jet Tagging,” [arXiv:2406.06638 \[hep-ph\]](#).
- [32] Y. Wu, K. Wang, and J. Zhu, “Jet Tagging with More-Interaction Particle Transformer,” [arXiv:2407.08682 \[hep-ph\]](#).
- [33] CMS Collaboration, “Search for highly energetic double Higgs boson production in the two bottom quark and two vector boson all-hadronic final state,” Tech. Rep. CMS-PAS-HIG-23-012, CERN, Geneva, 2024. <https://cds.cern.ch/record/2904879>.
- [34] ATLAS Collaboration, A. Duperrin, “Flavour tagging with graph neural networks with the ATLAS detector,” in *30th International Workshop on Deep-Inelastic Scattering and Related Subjects*. 2023. [arXiv:2306.04415 \[hep-ex\]](#).
- [35] ATLAS Collaboration, “Transformer Neural Networks for Identifying Boosted Higgs Bosons decaying into $b\bar{b}$ and $c\bar{c}$ in ATLAS,” Tech. Rep. ATL-PHYS-PUB-2023-021, CERN, Geneva, 2023. <https://cds.cern.ch/record/2866601>.
- [36] CMS Collaboration, “Transformer models for heavy flavor jet identification,” Tech. Rep. CMS-DP-2022-050, CERN, Geneva, 2022. <https://cds.cern.ch/record/2839920>.

- [37] A. Metz and A. Vossen, “Parton Fragmentation Functions,” *Prog. Part. Nucl. Phys.* **91** (2016) 136–202, [arXiv:1607.02521 \[hep-ex\]](#).
- [38] S. Albino, B. A. Kniehl, and G. Kramer, “AKK Update: Improvements from New Theoretical Input and Experimental Data,” *Nucl. Phys. B* **803** (2008) 42–104, [arXiv:0803.2768 \[hep-ph\]](#).
- [39] Belle Collaboration, R. Seuster *et al.*, “Charm hadrons from fragmentation and B decays in e^+e^- annihilation at $\sqrt{s} = 10.6$ GeV,” *Phys. Rev. D* **73** (2006) 032002, [arXiv:hep-ex/0506068](#).
- [40] CMS Collaboration, “Measurement of the shape of the b quark fragmentation function using charmed mesons produced inside b jets from $t\bar{t}$ pair decays,” Tech. Rep. CMS-PAS-TOP-18-012, CERN, Geneva, 2021. <https://cds.cern.ch/record/2771694>.
- [41] M. Galanti, A. Giammanco, Y. Grossman, Y. Kats, E. Stamou, and J. Zupan, “Heavy baryons as polarimeters at colliders,” *JHEP* **11** (2015) 067, [arXiv:1505.02771 \[hep-ph\]](#).
- [42] Y. Kats and D. Uzan, “Prospects for measuring quark polarization and spin correlations in $b\bar{b}$ and $c\bar{c}$ samples at the LHC,” *JHEP* **03** (2024) 063, [arXiv:2311.08226 \[hep-ph\]](#).
- [43] Y. Afik, Y. Kats, J. R. Muñoz de Nova, A. Soffer, and D. Uzan, “Entanglement and Bell nonlocality with bottom-quark pairs at hadron colliders,” [arXiv:2406.04402 \[hep-ph\]](#).
- [44] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations,” *JHEP* **07** (2014) 079, [arXiv:1405.0301 \[hep-ph\]](#).
- [45] C. Bierlich *et al.*, “A comprehensive guide to the physics and usage of PYTHIA 8.3,” *SciPost Phys. Codeb.* (2022) 8, [arXiv:2203.11601 \[hep-ph\]](#).
- [46] ATLAS Collaboration, “Expected performance of the ATLAS detector at the High-Luminosity LHC,” Tech. Rep. ATL-PHYS-PUB-2019-005, CERN, Geneva, 2019. <https://cds.cern.ch/record/2655304>.
- [47] CMS Collaboration, “Expected performance of the physics objects with the upgraded CMS detector at the HL-LHC,” Tech. Rep. CMS-NOTE-2018-006, CERN, Geneva, 2018. <https://cds.cern.ch/record/2650976>.
- [48] ATLAS Collaboration, T. Strebler, “Expected tracking performance of the ATLAS Phase-II Inner Tracker Upgrade,” *PoS ICHEP2022* (2022) 665.
- [49] Particle Data Group Collaboration, R. L. Workman *et al.*, “Review of Particle Physics,” *PTEP* **2022** (2022) 083C01.
- [50] M. Gronau, J. L. Rosner, and C. G. Wohl, “Overview of Λ_c decays,” *Phys. Rev. D* **97** (2018) 116015, [arXiv:1808.03720 \[hep-ph\]](#). [Addendum: *Phys. Rev. D* **98** (2018) 073003].
- [51] ATLAS Collaboration, M. Aaboud *et al.*, “Search for long-lived charginos based on a disappearing-track signature in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *JHEP* **06** (2018) 022, [arXiv:1712.02118 \[hep-ex\]](#).
- [52] ATLAS Collaboration, G. Aad *et al.*, “Search for long-lived charginos based on a disappearing-track signature using 136 fb^{-1} of pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Eur. Phys. J. C* **82** (2022) 606, [arXiv:2201.02472 \[hep-ex\]](#).

- [53] CMS Collaboration, A. M. Sirunyan *et al.*, “Searches for physics beyond the standard model with the M_{T2} variable in hadronic final states with and without disappearing tracks in proton-proton collisions at $\sqrt{s} = 13$ TeV,” *Eur. Phys. J. C* **80** (2020) 3, [arXiv:1909.03460 \[hep-ex\]](#).
- [54] CMS Collaboration, A. M. Sirunyan *et al.*, “Search for disappearing tracks in proton-proton collisions at $\sqrt{s} = 13$ TeV,” *Phys. Lett. B* **806** (2020) 135502, [arXiv:2004.05153 \[hep-ex\]](#).
- [55] CMS Collaboration, A. Hayrapetyan *et al.*, “Search for supersymmetry in final states with disappearing tracks in proton-proton collisions at $\sqrt{s} = 13$ TeV,” *Phys. Rev. D* **109** (2024) 072007, [arXiv:2309.16823 \[hep-ex\]](#).
- [56] Y. Kats, “Kinked tracks from Σ^+ baryons as a probe of light quark polarizations,” *JHEP* **07** (2023) 018, [arXiv:2301.06188 \[hep-ph\]](#).
- [57] M. Cacciari, G. P. Salam, and G. Soyez, “The anti- k_t jet clustering algorithm,” *JHEP* **04** (2008) 063, [arXiv:0802.1189 \[hep-ph\]](#).
- [58] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet User Manual,” *Eur. Phys. J. C* **72** (2012) 1896, [arXiv:1111.6097 \[hep-ph\]](#).
- [59] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, vol. 32, p. 8024–8035. 2019. [arXiv:1912.01703 \[cs.LG\]](#).
- [60] M. Wang *et al.*, “Deep graph library: Towards efficient and scalable deep learning on graphs,” [arXiv:1909.01315 \[cs.LG\]](#).
- [61] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *PMLR*, pp. 448–456. 2015. [arXiv:1502.03167 \[cs.LG\]](#).
- [62] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” [arXiv:1607.06450 \[stat.ML\]](#).
- [63] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” [arXiv:1606.08415 \[cs.LG\]](#).
- [64] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 770–778. 2016. [arXiv:1512.03385 \[cs.CV\]](#).
- [65] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Machine Learning Res.* **15** (2014) 1929–1958. <http://jmlr.org/papers/v15/srivastava14a.html>.
- [66] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2015. [arXiv:1412.6980 \[cs.LG\]](#).