

# Machine Learning in High Energy Physics: A review of heavy-flavor jet tagging at the LHC

Spandan Mondal \*

Luca Mastrolorenzo §

## Abstract

The application of machine learning (ML) in high energy physics (HEP), specifically in heavy-flavor jet tagging at Large Hadron Collider (LHC) experiments, has experienced remarkable growth and innovation in the past decade. This review provides a detailed examination of current and past ML techniques in this domain.

It starts by exploring various data representation methods and ML architectures, encompassing traditional ML algorithms and advanced deep learning techniques. Subsequent sections discuss specific instances of successful ML applications in jet flavor tagging in the ATLAS and CMS experiments at the LHC, ranging from basic fully-connected layers to graph neural networks employing attention mechanisms. To systematically categorize the advancements over the LHC's three runs, the paper classifies jet tagging algorithms into three generations, each characterized by specific data representation techniques and ML architectures. This classification aims to provide an overview of the chronological evolution in this field. Finally, a brief discussion about anticipated future developments and potential research directions in the field is presented.

## 1 Introduction

The task of jet identification in hadron collider experiments has seen steady development over the past decades. In particular, machine learning (ML) has revolutionized the way we view and

leverage various observables associated with a jet and its constituents. A jet refers to a collimated spray of particles that originate from a high-energy collision. They may originate either from a single quark/gluon generated in the hard scattering or from a hadronically-decaying Lorentz-boosted heavy particle. While viewing the jet as a single object gives us only limited information about the particle initiating the jet, looking at its constituents provides us with additional information that can be used to predict the type of the initiating particle. Several approaches have been developed for this purpose over the past decades and these approaches are generally referred to as jet tagging.

To analyze collision data in search for an interesting physics process (signal), one usually chooses a particular jet reconstruction strategy, along with a particular jet tagging algorithm (when necessary), depending on the expected features of the final state. For example, if the process involves the production of a hadronically-decaying heavy particle (such as W, Z, or Higgs boson, or a top quark) with a large transverse momentum ( $p_T \gtrsim 250$  GeV), one might use a so-called large-radius “fat” jet to reconstruct the entire decay within the jet radius. Such jets are usually characterized by a multi-pronged structure. Jet substructure techniques, and more recently ML-based classification algorithms, have been developed to distinguish fat jets arising from the aforementioned interesting heavy particle decays, from those arising from other physics processes (backgrounds), such as a gluon splitting into a pair of quarks. On the other hand, if the process generates one or more quarks/gluons that are spatially isolated, the event can be reconstructed using one or more small-radius “thin” jets. This is usually followed by application of techniques to identify the “fla-

\*Brown University, Providence, RI, USA.

§Ex-member, CMS Experiment, CERN.

vor” of the quark (up, down, strange, charm, bottom) or gluon that initiated the jet. This approach of identifying jet flavor is referred to as flavor tagging. The term flavor tagging can also be extended to specific use cases involving fat jets, for example, in the context of identifying heavy particles that decay into pairs of heavy-flavor quarks (bottom quark-antiquark pair,  $b\bar{b}$ , or charm quark-antiquark pair,  $c\bar{c}$ ). For the rest of this paper, up, down, strange, charm, bottom quarks, and gluons are denoted by  $u$ ,  $d$ ,  $s$ ,  $c$ ,  $b$ , and  $g$ , respectively, while  $b\bar{b}$  and  $c\bar{c}$  are denoted by  $bb$  and  $cc$ , respectively. The term “light-flavor” is used to denote  $u$ ,  $d$ ,  $s$ , and  $g$  combined.

The challenge of flavor tagging has been tackled using several approaches over the past three decades. These methods leverage certain observable characteristics of heavy-flavor quark ( $b$  and  $c$ ) hadronization and subsequent decays that are absent in case of light-flavor quarks ( $u$ ,  $d$ ,  $s$ ) and gluons ( $g$ ). As hadrons originating from  $b$  and  $c$  quarks (mostly  $B$  and  $D$  hadrons, respectively, among others) have a sizable lifetime ( $\sim 1.5$  ps) [1], they usually traverse a short distance within the detector systems ( $\sim$ few millimeters to centimeters) before further decaying. This gives rise to tracks that are displaced with respect to the primary point of collision (primary vertex, PV) and hence appear to originate from a displaced vertex (secondary vertex, SV). Methods leveraging the presence and properties of displaced tracks and/or reconstructed SVs were, therefore, developed to discriminate heavy-flavor jets from light-flavor ones. Additionally, gluon-initiated jets usually have a broader energy spectrum compared to quark-initiated jets, which makes quark-gluon discrimination possible. In case of fat jets, jet substructure methods leveraging predictions from the theory of quantum chromodynamics (QCD) have been used for flavor tagging. A discussion about early methods is presented in Sec. 4.

The advent of ML techniques, especially deep learning [2, 3], opened up a plethora of new possibilities of tackling the challenge of jet flavor tagging. Unlike using high-level physics-motivated observables as discussed above, ML-based methods allow us to build jet classification algorithms

by using information available from low-level observables associated with jets. The general idea behind these approaches is to train models to automatically leverage relevant information from observables (input features) and the correlations between them, that distinguish a heavy-flavor jet from a light-flavor one, using labeled simulated samples of jets (supervised learning) or from unlabeled collision data (unsupervised learning). However, these methods are optimal only when physically-meaningful representations of jet constituents are fed into corresponding artificial neural networks suited to process such representations.

This article is structured as follows. Section 2 discusses some general ML architectures and data representation methods that have been used in the field. Section 3 discusses some practical implementations of these in flavor tagging at Large Hadron Collider (LHC) [4] experiments. In particular, applications in the CMS [5] and ATLAS [6] experiments at the LHC have been covered in this review. Sections 4 and 5 present a discussion and a conclusion, respectively.

## 2 ML architectures and data representation

### 2.1 Boosted Decision Trees

Decision trees (DTs) [7–9] can be viewed as models resembling flowcharts. Each node in a DT represents a question and a decision based on a feature present in data. The decision of each node, and hence the “flow” along nodes, are optimized to lead to a correct prediction. Boosting [10], on the other hand, refers to an ensemble learning technique [11] that sequentially updates the model based on the errors made by the ensemble in the previous iteration. Boosted Decision Trees (BDTs), therefore, refer to models structured as DTs that leverage the technique of boosting to iteratively refine predictions based on the training dataset to perform classification and regression tasks [12].

BDTs are suitable for use with tabular data, represented as columns and rows of numbers and categorical variables. Each column of the table usually represents a feature and each row represents a unit (e.g., a jet). In supervised learning approaches, the correct (target) values are provided as additional columns in the training dataset.

Gradient Boosting Machines (GBMs) [13] introduce a more nuanced approach towards optimization of BDTs by making use of a technique called gradient descent. This introduces a *loss function* that quantifies the difference between the predictions and the target values in the training dataset using a predefined function. The gradient descent technique minimizes the loss function by updating the model parameters in the direction opposite to the gradient of the loss function. This allows fine-tuning of the model parameters and a more efficient convergence to the optimal solution. Gradient descent also allows the model to learn complex non-linear patterns in the input phase space.

Traditional BDTs as well as GBMs are widely used in a range of disciplines including Physics, in tasks involving simple, tabular data. Popular packages involve SCIKIT-LEARN [14] (Python [15]), XGBOOST [16] (Extreme Gradient Boosting), and TMVA [17] (ROOT [18]).

## 2.2 Dense Neural Networks

The simplest form of Neural Networks (NNs) [19, 20] consist of an input layer, a single hidden layer, and an output layer. Each node in a layer, referred to as a neuron, is densely connected to neurons in the subsequent layer, making a “fully-connected” structure. The flow of information is one-directional, from the input layer through the hidden layer to the output layer, which is why these architectures are also referred to as feed-forward NNs.

The connections between layers have associated weights, that are determined through the training process. Predefined activation functions [21] determine the output of each neuron and introduce non-linearity which helps the network capture complex patterns and relationships in data.

They determine the activation of a neuron based on the weights of its inputs.

Feed-forward Deep NNs (DNNs) [22] extend the architecture of simple feed-forward NNs by introducing additional hidden layers in between the input and output layers. This additional “depth” allows the network to automatically extract and leverage hierarchical features, intricate relationships, and non-linearities in the training data. Each layer learns progressively more abstract features and complex patterns which makes them suitable for large and higher-dimensional datasets.

Feed-forward NNs and DNNs are suitable for use with tabular data similar to BDTs. However, DNNs are usually capable of handling larger datasets and a larger number of input features as they can effectively capture complex, non-linear patterns in data. In this paper, the term “shallow NN” is used to describe dense NNs with one or two hidden layers, each with a small number ( $\sim 10$ ) of nodes. On the other hand, “DNN” is used to refer to NNs with a larger number of hidden layers and architectures that are not necessarily dense, feed-forward, or fully-connected.

The term Multilayer Perceptron (MLP) is also used to describe dense NNs. The simplest kind of MLP, a feed-forward MLP with one hidden layer, is essentially the same as a shallow NN. On the other hand, MLPs with additional hidden layers, which still have a feed-forward structure, resemble dense DNNs. In this paper, the terms “dense”, “fully-connected”, “feed-forward”, and “MLP” are used interchangeably to describe this class of NNs.

## 2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have revolutionized the field of image recognition and computer vision [23]. The architecture of CNNs is inspired by the visual processing hierarchy of the human brain, where initial layers in the hierarchy capture simple features like edges and textures, while deeper layers capture more complex patterns like composite objects and structures.

CNNs make use of convolutional layers, where filters (or kernels) slide across the input data and capture local patterns. In the context of a large image expressed as rectangular pixellated data, one can imagine the filter as a rectangle with a smaller area, systematically sliding across the image and producing an output corresponding to each spatial position. The output for each position depends on the local information of the image (pixels in the kernel’s receptive field) and the values of the kernel itself. The sliding process thus generates a new feature map typically smaller than the original image. The convolutional filters are adapted iteratively in the CNN training process such that they autonomously detect and learn representations that are most relevant to the task at hand. The associated weights are shared across all neurons in a particular feature map.

Convolutional layers are usually followed by pooling layers. The pooling operation involves sliding a rectangular pooling window in a similar way across the generated feature map. The main goal of the pooling layer is to reduce the spatial dimension by selecting representative values within local neighborhood of the feature maps. The representative values are usually evaluated through simple mathematical operations like taking the maximum of or averaging the pixel values of the feature map in the window’s receptive field. This not only helps in down-sampling the feature maps but also abstracts information by retaining only the relevant features from local regions of the image.

Unlike fully-connected NNs, neurons in each layer are connected to a neurons in a small spatial region of the previous layer. This lets the network capture local patterns and leverage the fact that adjoining pixels may contain related information without being sensitive to the exact spatial position of the feature relative to the full image. Again unlike fully-connected NNs, CNNs make use of weight sharing, which essentially means the same filter is applied to the entire image, and therefore use fewer trainable parameters.

CNNs are well-suited for tasks involving image recognition and classification. Images fed as input to CNNs are generally represented as pixels with one or multiple channels. In the context of

jet tagging, jets can be viewed as images on the detector surface with particle energy deposits acting as intensities of the pixels of the image.

## 2.4 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [20, 24, 25] are designed for sequential data and time series. RNNs process data sequentially such that the output at each step is influenced by a hidden state that encapsulates information from previous temporal steps. This lets the RNN maintain “memories” of previous inputs and learn temporal dependencies.

Long Short-Term Memory networks (LSTMs) [26] are specialized RNNs that make use of memory cells for a better handling of sequential data. These memory cells store and manage information over extended sequences, by selectively storing, discarding, and outputting information. This allows LSTMs to capture long-term dependencies in sequential data.

Gated Recurrent Units (GRUs) [27, 28], another variant of RNNs, offer a streamlined architecture without using memory cells but employ similar gating mechanisms to capture dependencies in sequential data. GRUs use update gates and reset gates to control the flow of information, making them computationally efficient and effective for tasks involving sequential information.

RNNs and LSTMs have been widely used in Natural Language Processing (NLP) [29], time-series analysis [30, 31], machine translation [27, 32], speech processing [33], and other domains where interpreting information across multiple steps is crucial. An interesting feature of LSTMs is their ability to handle sequences of varying lengths. This is particularly relevant in the field of jet tagging since jets naturally contain a varying number of clustered particles. Additionally, the ability of LSTMs to selectively store and forget information allows them to create meaningful summaries or averages over information (particles) available in a sequence (jet). This helps LSTMs identify, for example, particles in a jet that contribute the most to jet identification.

The utility of RNNs extends naturally to structured data like trees. Each node in a tree represented as a step in the sequence allows RNNs to capture dependencies and relationships within the hierarchical structure of the tree. As jets are clustered sequentially from individual particles, their clustering history can be represented as binary trees. Operating on trees, RNNs can take multiple inputs per level in the tree to add to the internal state at each step.

In the context of preprocessing data to be sequential, *tokenization* is a crucial step. It involves breaking down raw inputs into smaller units called tokens. Each token represents a discrete unit, facilitating the representation of the sequential information in a format suitable for RNNs and LSTMs.

## 2.5 Transformers

A crucial feature of transformer models is the self-attention mechanism [34, 35], that allows the model to assign importance (weights) to different parts of the input sequence differently. This allows them to capture long-range dependencies, extending the concept from LSTMs. Multi-head Attention (MHA) extends the concept by employing multiple sets of parameters (heads) to independently compute attention scores in parallel. The outputs from all the heads are then combined linearly with learnable weights. This allows the model to capture various patterns and dependencies within the input sequences simultaneously with different choice of attention parameters, making the model more versatile and robust.

Unlike RNNs, transformers can operate in parallel while computing attention weights. Furthermore, each layer in transformer models can operate independently of the others. Parallelization makes transformer models highly efficient, especially when training with large datasets. It also contributes to faster training and prediction times in general.

Transformers have revolutionized the field of NLP, achieving unprecedented performances in tasks

such as language understanding, translation, and question-answering. The concept of attention has also been applied to a wide range of tasks in ML as an effective tool for capturing important information selectively. Transformers also excel at classification tasks, which makes them relevant in jet flavor tagging. In classification tasks, a specialized token referred to as a class token is positioned at the start of an input sequence during tokenization. The hidden state associated with this class token is then updated throughout the sequence and utilized for making predictions through a dedicated classification head.

## 2.6 Deep Sets

In contrast to tabular data in dense NNs, grids in CNNs, and ordered sequences in RNNs, physical jets do not intrinsically have an ordered representation. Orders are usually imposed (for example, by sorting jet constituents in descending order of their  $p_T$ ) to make their representations suitable as inputs for the aforementioned networks. However, the choice of ordering is not unique and a more natural representation of a jet is as an unordered set of particles.

Deep Sets [36] provide a framework to handle data that can be represented as sets of unordered units, where the relationships between elements in the set matter while their orders are inconsequential. The key component in the Deep Set implementation is the approximation of a set function: a function that takes a set of elements as input and produces fixed-size representation of the entire set. The function is required to be permutation invariant to the order of objects in the set. The architecture acts on each instance in the input set and transforms each element, possibly through several layers, into some abstract representation. The representations from the entire set are added up and the output is processed using dense NNs through non-linear transformations. The set function is usually designed in a way to make use of weight sharing, which ensures that the network learns consistent transformers across all elements and results in fewer learnable parameters.

Thus Deep Sets can not only handle sequences of varying lengths but also provide consistent representations of sets with different ordering of elements, which is a more natural choice for representing jets. This improves the generalization capabilities of the model without relying on specific element orders. Permutation invariance also simplifies the training process and training optimization and leads to faster and stable convergence.

## 2.7 Graph Neural Networks

Graph Neural Networks (GNNs) [37–39] are tailored to handle data with well-defined and complex relationships. They are designed to process graphs: data structures that consist of nodes that represent entities, interconnected by edges that represent relationships and interactions between the different entities. They are, therefore, the natural choice for data structures characterized by interconnected and interdependent nodes.

A key feature of GNNs is the concept of *message passing*. Information is propagated between nodes connected by an edge. Each node aggregates information from its connected neighbors in a local as well as a global context. This allows GNNs to learn patterns of information flow, identify nodes important to the task at hand, and leverage abstract relationships between all nodes in the network.

In the field of HEP, GNNs are often used to model particles and/or physics objects represented as nodes of a graph, and pairwise features between particles as edges. Such representations are referred to as point clouds in general, and sometimes as particle clouds in HEP.

## 2.8 Dynamic Graph Convolutional Neural Networks

Dynamic Graph Convolutional Neural Networks (DGCNN) [40] represent an evolution in the realm of GNNs by extending their capabilities to characterize local features in the global structure of the

data. DGCNNs are equipped to handle graphs that are dynamically computed in each layer of the network starting with the input graph representation at the first layer, as opposed to GNNs that act on a static input graph. This is achieved using a stackable NN module referred to as edge convolution (EdgeConv). In the context of HEP, sequential EdgeConv operations, starting with original particle coordinates and features, lets one create several layers of graph representations of particle clouds. This is similar to processing images with CNNs where one can create hierarchical (increasingly abstract) representations of an image by sequentially applying convolutional filters starting with the original image.

The EdgeConv operation begins with a graph whose nodes represent particles, and edges for each node represent the difference in feature vectors of the  $k$  nearest neighboring nodes from the central node. An edge function with learnable parameters is then used to project each point into a latent feature space, where the point is effectively assigned new coordinates. Thus, a single EdgeConv operation transforms one point cloud into another point cloud with the same number of points, but with different feature vectors for each point. Repeated EdgeConv operations can be applied to generate hierarchical representations such that feature space structures in the deeper layers can group related points together regardless of their spatial separation in the original graph structure.

Similar to convolution operations in CNNs, EdgeConv operations incorporate local neighborhood information and can be stacked to learn global spatial properties. In the context of HEP, this is helpful in capturing characteristics and relationships between particles that are spatially well-separated in the original representation on the detector surface.

## 2.9 Summary

This section describes a few ML architectures that are widely used and especially relevant to the task of jet tagging in HEP. Different architectures are

| Architecture | Suitable for                    |
|--------------|---------------------------------|
| BDT          | Tabular/structured data         |
| Dense NN/MLP | Tabular/structured data         |
| CNN          | Images, grids                   |
| RNN/LSTM     | Sequences, trees, text          |
| Transformer  | Sequences, text, large datasets |
| Deep Set     | Unordered sets, point clouds    |
| GNN          | Graphs, point clouds            |
| DGCNN        | Graphs, point clouds            |

Table 1: A few common ML architectures (first column) and non-comprehensive lists of data representations (second column) they are suited for.

suitable for different kinds of data and their representations. A brief, non-comprehensive overview of architectures and corresponding datasets is presented in Table 1.

### 3 ML in heavy-flavor jet identification at hadron colliders

Hadron collider experiments have adopted several ML-based strategies for jet tagging over the years. These strategies can be broadly classified into single-pronged jet tagging typically used in conjunction with thin jets, and multi-pronged jet tagging used with fat jets. The following subsections discuss these two categories separately. In this paper, flavor tagging algorithms are further classified into three generations. Taggers developed with BDTs and shallow NNs during LHC Run-1 [41] and early Run-2 [42], are included in the first generation of taggers. In the second generation, taggers using early DNNs, CNNs, and RNNs, developed during Run-2 and implemented as standard taggers by the end of Run-2 data reconstruction, are included. Finally, more advanced taggers developed after Run-2 and commissioned with early Run-3 [43] data are included within the third generation of taggers. These include more complex algorithms implementing Deep Sets, GNNs, DGC-

NNs, and transformer architectures, relying on representing jets as unordered sets and particle clouds. A summary of the taggers is presented in Table 2 at the end of this section. Additionally, a plot depicting the evolution of single-pronged jet tagging performance over the last decade is presented in Fig. 1.

#### 3.1 Single-pronged jet tagging

Individual quarks and gluons usually give rise to small-radius “thin” jets. In both CMS and ATLAS experiments, these are clustered using the anti- $k_T$  (AK) [44, 45] algorithm. The distance parameter  $\Delta R$  used by the experiments for thin jets was 0.5 or 0.4 during Run-1, 0.4 during Run-2, and 0.4 in Run-3. These jets are referred to as AK5 or AK4 jets, depending on  $\Delta R$ .

##### 3.1.1 First generation: BDTs and shallow NNs

The Combined Secondary Vertex (CSV) algorithm [46] was developed in CMS during Run-1 of the LHC. This approach combines a small set of high-level physics-motivated variables, that are expected to have different distributions for b, c, and light-flavor jets, such as vertex kinematics, track kinematics, multiplicities, etc. The number of variables ranges between 2 to 8 depending on whether an SV is reconstructed [47–49] within the jet. Two likelihood ratios are built from these variables, to discriminate b from light-flavor, and b from c jets, respectively.

In Run-2 of the LHC, the CSV algorithm was updated by implementing a feed-forward MLP (cf. Sec. 2.2) with one hidden layer. The new NN-based approach was dubbed as CSVv2 [50, 51]. Even though a likelihood-based method with a small number of physics-motivated variables is a simple and physically-interpretable approach, the update from using likelihood ratios in the CSV algorithm in Run-1 to using an MLP in CSVv2 in Run-2 allowed the incorporation of additional input variables in the input layer of the MLP. Thus,

up to 22 variables, including information of up to 4 tracks, additional track and SV variables, and overall kinematic features of the jet, were used as inputs to the algorithm. The number of nodes in the hidden layer was set to twice the number of input variables for effective projection and extraction of features from the input data. As a result of additional input information and the updated architecture, the efficiency of selecting b jets at approximately 10% light-jet mistag rate went up from around 74% in CSV to around 82% in CSVv2. More detailed efficiency values at different mistag rates can be found in Refs. [52, 53].

In addition to track and SV variables, soft-leptons arising from semileptonic decays of b and c hadrons can also provide handles to discriminate heavy-flavor jets from light-flavor ones. Such jets, however, account for only about 20% (10%) of b (c) hadron decays. In Run-2 of CMS, BDT-based (cf. Sec. 2.1) soft-lepton (SL) taggers [51] were used as jet flavor taggers. These algorithms, called soft-electron (SE) and soft-muon (SM) taggers, make use of 2D and 3D impact parameter (IP) significances<sup>1</sup> of the lepton, the angular distance between the jet axis and the lepton, the ratio of the  $p_T$  of the lepton to that of the jet, the  $p_T$  of the lepton relative to the jet axis, and an MVA-based electron quality (in case of SE only) as inputs.

In Run-2, the CMS experiment also implemented a combined tagger named combined MultiVariate Analysis version 2 (cMVAv2) [51]. This tagger combined the outputs of six other heavy-flavor taggers, including both physics-motivated taggers and ML-based taggers like CSVv2, SE, and SM, using a GBM model (cf. Sec. 2.1) as a BDT. The cMVAv2 algorithm achieves a gain of about 3–4% in b jet identification compared to the CSVv2 algorithm, indicating the importance of additional input information.

Dedicated c jet tagging algorithms were also developed during Run-2 of CMS. This development was motivated by both the availability of more advanced ML technologies in Run-2, and the general switch of the focus of the physics program

<sup>1</sup>IP significance is defined as the ratio of the IP value to its uncertainty.

at the LHC experiments from discovery of the Higgs boson [54, 55] in Run-1 to characterization of the nature of the Higgs boson [56] in Run-2. As properties of c jets lie somewhere in between that of b and light-flavor jets, two different taggers to discriminate c from b (CvsB), and c from light-flavor (CvsL) jets were developed [51, 57]. These taggers combined the same kind of track, SV, and soft-lepton information used in CSVv2 and SL taggers. Two separate GBM models as BDTs were trained to implement the CvsB and CvsL taggers. Physics analyses in CMS involving c jets in the final state implemented selection criteria on both the CvsL and CvsB output scores to select c jets while rejecting both light-flavor and b jets at certain misidentification rates.

The ATLAS experiment used likelihood-based approaches exploiting IP and SV information separately to construct low-level b tagging algorithms in Run-1 [58, 59]. Additionally, the JetFitter algorithm [60, 61] based on a shallow NN was also developed. JetFitter was designed to exploit the topological structure of b and c hadron decays inside jets. It takes 8 variables as input, which include vertex and track multiplicities, the vertex mass, energy fraction carried by the tracks, vertex flight distance significance, jet  $p_T$ , and jet pseudorapidity. The JetFitter architecture consisted of two hidden layers, with 12 and 7 nodes, respectively. The output layer consisted of 3 nodes, representing the probabilities of the jet being a b, c, or light-flavor jet, respectively.

In addition to low-level tagging algorithms, the ATLAS experiment in Run-1 used a MLP-based combined tagging algorithm called MV1 [58, 62]. The MV1 algorithm was designed to combine the outputs of the two low-level likelihood-based taggers exploiting IP and SV information. As an NN, the MV1 algorithm had the advantage of automatically leveraging complex correlations between its inputs. The architecture consisted of 2 hidden layers with 3 and 2 nodes, respectively. The output layer consisted of only 1 node indicating the final discriminant value. The MV1 algorithm improved the light-flavor jet background rejection by a factor of  $\sim 2$  compared to the JetFitter algorithm, at a b jet tagging efficiency of 70%. The improvement



can be attributed to MV1’s ability to combine IP and SV information as opposed to the SV-based JetFitter algorithm.

In early Run-2 of the LHC, the ATLAS experiment implemented the MV2 algorithm [63–65]. As opposed to the NN-based MV1, the MV2 is built on a BDT-based architecture. The key distinction between MV1 and MV2 lies in the handling of input IP and SV information. MV1 utilizes the outputs of intermediate low-level tagging algorithms, whereas MV2 directly takes as input the variables employed as inputs to the low-level taggers. Additionally, the MV2 algorithm makes use of the inputs to the JetFitter algorithm. This results in a total of 24 input variables to the BDT-based architecture. This streamlined approach not only simplifies the overall flavor tagging process but also enables MV2 to directly leverage correlations among the lower-level inputs, avoiding the inevitable loss of information that occurs when condensing input information into just two or three numbers.

Several versions of the MV2 algorithm were trained, varying the proportions of light-flavor and  $c$  jets used as backgrounds in the training, with the aim of optimizing the respective networks to strike a balance between light-flavor jet and  $c$  jet discrimination. Algorithms dubbed MV2c00, MV2c10, and MV2c20 use only light-flavor jets, an admixture of 90% light-flavor and 10%  $c$  jets, and an admixture of 80% light-flavor and 20%  $c$  jets, respectively, as backgrounds. As expected, the MV2c20 tagger significantly improves  $c$  jet rejection ( $\sim 4$  times better at 50%  $b$  efficiency) with only a slight degradation (factor of  $\sim 1.5$  at 50%  $b$  efficiency) in light-flavor jet rejection, compared to the MV2c00 tagger. The MV2c10 and MV2c20 variants were, however, trained with altered  $c$  jet fractions in 2016 while retaining the nomenclatures. The details about the optimization can be found in Ref. [66]. The intermediate MV2c10 tagger was used as the standard tagger in ATLAS in Run-2 [67–69]. When comparing the Run-1 MV1 and Run-2 MV2 algorithms, MV2 improves light-jet rejection by a factor of  $\sim 4$  and  $c$  jet rejection by a factor of 1.5–2 for a  $b$  jet efficiency of 70%.

### 3.1.2 Second generation: Early DNNs

The advent of deep learning led to the development of the DeepCSV algorithm [51, 70] in the CMS experiment. This is an update to the CSVv2 algorithm and incorporates more hidden layers, more nodes per layer, and the kinematics of up to six tracks as inputs. The total number of inputs to DeepCSV can be up to 66 per jet. The architecture consists of four hidden layers and 100 nodes per layer, while maintaining the fully-connected and feed-forward structure. Furthermore, the architecture is implemented as a multiclassifier, which means the output layer has 5 nodes representing 5 types of jets, namely  $b$ ,  $bb$ ,  $c$ ,  $cc$  and light-flavor. As opposed to SCIKIT-LEARN used for BDT and shallow NN training, these trainings made use of the KERAS [71] and TENSORFLOW [72] libraries.

The DeepCSV algorithm achieves a 3–9% improvement in  $b$  jet efficiency compared to CSVv2 depending on the selection criterion used on the output score. The performance is similar to, or on par with  $c$ MVA $v_2$  at low purity (high light-jet mistag) regions, while the performance is significantly superior to  $c$ MVA $v_2$  at high purity selections. The improved performance of DeepCSV without using explicit soft-lepton information in jets can be attributed to the use of a deeper NN capable of taking a vastly larger number jet variables as inputs.

Owing to the multiclassifier nature of the DeepCSV algorithm, it is suitable for  $c$  jet tagging as well. The output scores can be transformed to define  $c$  tagging scores that perform the same task as dedicated CvsB and CvsL BDT trainings, as follows:

$$\begin{aligned} \text{DeepCSV CvsB} &= \frac{P(c) + P(cc)}{1 - P(\text{udsg})}, \\ \text{DeepCSV CvsL} &= \frac{P(c) + P(cc)}{1 - (P(b) + P(bb))}. \end{aligned} \quad (1)$$

Charm jet tagging with DeepCSV also achieves slightly better performance than with dedicated BDT trainings.

The first algorithm based on deep learning implemented in the ATLAS experiment was the DL1

algorithm [65, 73]. The variables used as inputs to DL1 are the same as the MV2 algorithm (cf. Sec. 3.1.1) with the addition of so-called “JetFitter c tagging variables”. The latter include properties of a tertiary vertex, consistent with the hypothesis of a b hadron further decaying to a c hadron, such as flight distance, vertex mass and number of tracks, energy, energy fraction, and rapidity of the tracks associated with the secondary and tertiary vertices. This results in a total of 28 input variables. The DL1 architecture consists of 8 fully-connected hidden layers, each with a decreasing number of nodes ranging from 78 to 6, and 3 Max-out layers [74]. Similar to DeepCSV, DL1 is also a multiclassifier with dedicated output nodes corresponding to probabilities of the jet being a b, c, or light-flavor jet. The b tagging discriminator used in the ATLAS experiment is defined as

$$D_{\text{DL1}}^{\text{b}} = \ln \left( \frac{p_{\text{b}}}{f_{\text{c}} \cdot p_{\text{c}} + (1 - f_{\text{c}}) \cdot p_{\text{light}}} \right), \quad (2)$$

where  $p_F$  represents the NN output score corresponding to flavor  $F$ , and  $f_c$  represents the effective c jet fraction in the background training sample. This allows flexibility in choosing the c jet fraction in the background *a posteriori* and allows optimization of the performance after training. This avoids multiple trainings of the algorithm with varying c jet fractions as was the case for MV2c00 and MV2c20 variants (cf. Sec. 3.1.1). The value of  $f_c$ , for example, may be set to 8% in physics analyses after optimization. The DL1 algorithm improves the light-flavor and c jet rejections by 30% and 10%, respectively, compared to MV2, at a b jet efficiency of 70%. This improvement can be attributed to the use of JetFitter c tagging variables and a deeper NN capable of effectively exploiting a larger number of inputs.

During Run-2, the ATLAS experiment also developed a track-based RNN tagger for flavor tagging. This tagger, named RNNIP [75], is designed to efficiently leverage intrinsic correlations between the IPs (and other observables) of the several tracks in a jet, as opposed to likelihood-based IP taggers, which assume properties of each track in a jet are independent of all other track. Owing to a variable number of tracks per jet, a representation in form of variable-length sequences of

particles and an RNN-like architecture resulted to be the natural choice for this kind of data. A set of basic  $p_{\text{T}}$  and IP selection criteria are applied while selecting tracks that are fed as input to the RNNIP tagger. As RNNs can handle sequences of arbitrary lengths, no limit on the maximum number of tracks per jet is necessary, but a limit of 15 tracks per jet was imposed for ease of training. The tagger takes as input the transverse and longitudinal IPs of each selected track, the  $p_{\text{T}}$  fraction of the jet carried by each selected track, the angular distances between the selected tracks and the jet-axis, and the hit multiplicities of the selected tracks. Tracks in each jet are ordered by their transverse IP in the input sequence. This arbitrary-length sequence is then fed into an LSTM which transforms it to a 50 dimensional vector in the latent space. This vector is then processed by a dense NN with four output nodes, corresponding to probabilities of the jet being a b, c, light-flavor, or hadronic  $\tau$  jet, respectively. The RNNIP algorithm improves light-flavor and c jet rejection by factor of 2.5 and 1.2 times, respectively, compared to likelihood-based IP taggers, at a b jet efficiency of 70%.

Again during Run-2, the ATLAS experiment introduced an additional variant of the DL1 algorithms by additionally incorporating the outputs of the RNNIP tagger as inputs to DL1. This variant of DL1, dubbed DL1r [76, 77], is thus able to leverage additional information that is not strongly correlated with the inputs of the baseline DL1. The number of inputs to DL1r increases to 31, and it retains the 8-hidden-layer architecture of DL1, but uses a larger number of nodes per layer, ranging from 256 in the first hidden layer to 6 in the final. The output layer similarly has three nodes corresponding to probabilities of the jet being a b, c, or light-flavor jet. Similar to  $D_{\text{DL1}}^{\text{b}}$ , a c tagging score for DL1r can be defined as

$$D_{\text{DL1r}}^{\text{c}} = \ln \left( \frac{p_{\text{c}}}{f_{\text{b}} \cdot p_{\text{b}} + (1 - f_{\text{b}}) \cdot p_{\text{light}}} \right), \quad (3)$$

where  $f_{\text{b}}$  represents the effective b fraction in the background training sample and can be tuned to optimize b jet rejection rate. The DL1r tagger improves the light-jet and c jet rejection rates by

about 20% and 12% respectively, when compared with the DL1 algorithm, at a b jet efficiency of 70%.

The DeepCSV, DL1(r), and RNNIP algorithms are limited by the fact that only a small number of charged tracks that pass certain physics-motivated quality criteria are used as inputs to the network. Information about neutral jet constituents and additional charged tracks that are deemed less important (such as ones with lower IP or IP significance), are essentially lost. This led to the development of DeepJet [78], a hybrid NN that utilizes the full information from a significantly larger number of charged particles, neutral particles, SVs, and global jet- and event-level variables, resulting in a total of up to 650 input variables. Sixteen features of up to 25 charged particles are used as input. These features include track kinematics, track fit quality, IP, and IP uncertainty. Additionally, 6 features of up to 25 neutral particles are provided as input. Twelve features of up to 4 reconstructed SVs in the jet are also used. These features include the kinematics of the SV, SV mass, fit quality, and flight distance. Finally, the global variables include the overall jet kinematics, particle and SV multiplicities in the jet, and number of PVs (*intime pileup*<sup>2</sup>) in the event. The DeepJet algorithm (also referred to as DeepFlavour during early developments [79]) was practically implemented in the CMS experiment during Run-2 [80].

The architecture of DeepJet is tailored to handle the large input dimension. For this purpose, separate convolutional filters [81] are trained for each of the three branches (charged, neutral particles, and SVs). For each branch, a  $1 \times 1$  filter size is chosen to ensure that all particles in the branch undergo the same feature transformation without interacting with other particles. This step is aimed at reducing the dimensionality of the input space, and is in contrast to using convolutional filters on image grids, where the main aim is to find clusters of neighboring pixels that share similar features relevant to object detection. To

---

<sup>2</sup>Pileup refers to overlapping events from additional parton-parton interactions occurring simultaneously within the same bunch crossing.

this end, several layers with decreasing number of filters are used so that features of each constituent are projected to a lower dimensional space. Outputs from each convolutional branches are fed into individual LSTMs. A jet is treated and processed by the LSTM as a sequence of constituents ordered by their importance. The outputs of the three LSTMs, as well as the global jet/event features are concatenated and passed as inputs to a dense NN consisting of 8 layers. The output layer consists of 6 nodes, representing b, bb, lepb, c, uds, and g. As opposed to DeepCSV, DeepJet splits the b node into b and lepb, the latter marking b jets containing soft leptons. It also splits the udsg node into uds and g, making quark-versus-gluon discrimination possible. On the other hand, the cc node present in DeepCSV is deprecated and merged with the c node in DeepJet.

The DeepJet algorithm in CMS achieves a 20% improvement in b tagging efficiency at a light-jet mistag rate of 0.1%, when compared with the DeepCSV algorithm. The improved performance can be attributed to the use of up to an order of magnitude larger number of inputs, without applying explicit quality requirements on the input constituents, and the use of neutral candidates as inputs. This change marks a shift from using a handful of high-level physics-motivated variables to using a large number of low-level variables without explicit selections, and relying on the network to automatically leverage relevant information from inputs that may contain additional inconsequential and/or correlated information. More detailed performance comparisons for b and c tagging using CMS collision data can be found in Refs. [80, 82–87].

### 3.1.3 Third generation: Sets and clouds

The CNN- and RNN-based approaches discussed so far treat jets either as detector images or as ordered sequences of particles. Ordering jet constituents using a certain parameter, e.g. the IP significance or  $p_T$ , is a necessity in all these use cases. As final-state particles in jets do not have a natural order, any order that is imposed artificially may prove to be detrimental to the data rep-

resentation and hence the model’s performance. A more natural way of representing jets is in the form of point clouds without assigning any order to the particles. Deep Set-based approaches (cf. Sec. 2.6), that treat jet constituents as permutation invariant, are therefore a more natural choice to handle particle cloud-based representations of jets.

The ATLAS experiment introduced the Deep Impact Parameter Sets (DIPS) tagger [88] after the LHC Run-2 data-taking period. DIPS is based on the Deep Set architecture, and in particular, the application of the Deep Set formalism in HEP, referred to as Energy Flow Networks [89]. Similar to RNNIP (cf. Sec. 3.1.2), DIPS uses track information as input but treats them as an unordered, variable-sized set, instead of explicitly assigning an order. The DIPS architecture applies a NN to each track, sums over the tracks, and then processes the summed representation using a dense NN with 2 hidden layers and 100 nodes per layer. The output layer consists of 3 nodes, corresponding to the probabilities of the jet being a b, c, or light-flavor jet, respectively. The summation operation over all tracks in a jet being a permutation invariant operation, makes the network agnostic to the ordering of the tracks in the input. On the other hand, the final dense NN accounts for correlations between tracks.

The permutation invariant approach results in DIPS having fewer trainable parameters compared to RNNIP (for similar number of inputs and similar performance). The operation of processing the tracks can also be parallelized as opposed to the operation of iteratively processing sequence-like inputs in RNNs. As a result, DIPS exhibits a training speed per epoch approximately three times faster than RNNIP, and its inference (prediction) speed surpasses that of RNNIP by a factor of four, when handling the same number of input features. In terms of performance, the baseline version of DIPS having the same input features as RNNIP achieves about 15% higher light-flavor jet rejection and about 3% higher c jet rejection at a b jet efficiency of 70%. Additionally, the lower complexity and higher training speed of the DIPS tagger facilitates faster optimization studies

and allows the incorporation of a higher number of tracks and more features per track. When the track selection is relaxed to additionally incorporate low- $p_T$  and high IP tracks, and additional IP features per track are added as inputs, the DIPS tagger improves light-flavor jet rejection by a factor of  $\sim 1.9$  and c jet rejection by a factor of  $\sim 1.3$  at a b jet efficiency of 70%, when compared to the baseline version. These improvements highlight the importance of physically-motivated ML network architectures in the task of jet flavor tagging.

The introduction of the DIPS algorithm in ATLAS subsequently resulted in the introduction of a third variant of the DL1 algorithm, dubbed DL1d [90, 91], superseding the DL1r algorithm. Instead of using outputs of the RNN-based RNNIP algorithm as inputs, DL1d utilizes the outputs of Deep Set-based DIPS as input. At a b jet efficiency of 70%, DL1d achieves about 25% higher light-flavor jet and 20% higher c jet rejection compared to DL1r. This is a direct result of the improved performance of the optimized DIPS tagger compared to the RNNIP algorithm.

The next improvement in tagging performance in ATLAS was achieved with the use of GNNs (cf. Sec. 2.7). The GN1 algorithm [92], introduced during the beginning of the LHC Run-3 data-taking period, directly leverages information from tracks inside a jet to predict the flavor of the jet. This is in contrast to other high-level taggers (like the MV2 and DL1 series) used in ATLAS so far that utilized outputs of intermediate low-level taggers as inputs and performed the task in two steps. The GN1 algorithm takes as input the jet  $p_T$  and pseudorapidity, and a variable number of tracks contained inside the jet, with 21 tracking related variables for each track. The tracking related variables include several kinematic features of the track with uncertainties, IPs with their significances, and the number of hits in various tracker layers. For jets with more than 40 tracks, only 40 tracks with the highest transverse IP significance are chosen. An additional variant, named GN1 Lep, utilizes the lepton ID of tracks to indicate whether a track is reconstructed as an electron, or a muon, or neither.

In addition to flavor tagging, the GN1 is assigned two auxiliary objectives using training targets based on “truth” information available from simulation, similar to “truth” flavors obtained from simulation and used in flavor tagging. The first objective is to label each track by its origin, i.e. predict whether a track originates from pileup, the primary collision, a b hadron, a c hadron originating from a b hadron decay, a c hadron originating from the primary collision, other secondary interactions, or is an incorrectly-reconstructed track. The second objective is to predict whether any given pair of tracks originates from the same point in space (vertex) [93], with a binary prediction as an additional GN1 output. These auxiliary tasks help in the main task of flavor tagging by acting as a form of supervised attention [94], since detecting tracks from b and c hadrons and being able to group multiple tracks into a single SV lets the model pay more attention to these tracks to make flavor predictions. Thus these objectives guide the network to learn representations of the jet connected to the underlying physics. The second auxiliary objective also removes the need for dedicated SV inputs to the model, and also facilitates reconstructing SVs from the predictions of the network without using standalone vertexing algorithms.

The GN1 architecture is based on a previous implementation of GNN-based jet tagger [95]. Each input track with 21 track features (concatenated with the two global jet features) are fed into a per-track initialization network with three hidden layers, each containing 64 neurons, and an output layer with 64 nodes, similar to the DIPS approach. However, unlike DIPS, no summation operation on the output representation is applied. Instead, a fully-connected graph is built from the outputs of this network [96], where each node corresponds to a latent feature vector representing a track, and each node in the graph is connected to every other node. Each layer of the GNN aggregates the properties of each input node and its connections to compute a corresponding output node. This is achieved in each layer by first processing the input features using a fully-connected layer, then computing edge scores for each node pair in the output of the fully-connected layer using a second

fully-connected layer, then computing pairwise attention weights using the edge scores, and finally taking the weighted sum of each node from the output of the first fully-connected layer weighted by the respective attention weights. Three such layers are stacked to construct the GNN. A global representation of the jet is formed by combining the output representations of individual tracks through a weighted sum, the weights for which are learned during training. Subsequently, three distinct fully connected feedforward NNs operate independently to achieve the various classification objectives of GN1. A graph classification network using the entire jet representation predicts the jet flavor, a node classification network using features from individual nodes representing tracks predicts track origin, and an edge classification network utilizing features of edges representing pairs of tracks predict the vertex compatibility of every pair of tracks.

The GN1 tagger improves c jet rejection by a factor of  $\sim 2.1$  and light-flavor jet rejection by a factor of  $\sim 1.8$  compared to DL1r, at a b jet efficiency of 70%. For the GN1 Lep variant the respective improvements are by factors of  $\sim 2.8$  and  $\sim 2.5$  compared to DL1r, demonstrating the additional jet flavor discrimination information available from the leptonID track input. On the other hand, removing the auxiliary objectives from the GN1 tagger, results in a performance similar to that of the DL1r algorithm, suggesting that relevant physically-motivated auxiliary tasks help the network learn the underlying physics relevant to the task of flavor tagging.

The advent of the transformer architecture and attention mechanism led to the development of ParticleTransformer (ParT) [97]. Unlike typical implementations of the transformer architecture for NLP and related tasks, the ParT architecture does not assign positional encodings to particles in the jet; this aligns with the concept of particle clouds and keeps the jet constituents permutation invariant. Instead, the novelty of ParT lies in the use of two sets of inputs: the “particle” input comprising the features associated with all particles in a jet, and the “interaction” input which encodes pairwise features for every possible pair of particles in the jet. The latter is, therefore, represented

by a  $N \times N$  matrix of values for each pairwise feature, where  $N$  is the number of constituents in a jet. The network can also be viewed as a fully-connected GNN (cf. Sec. 2.7) where each node represents a particle and each edge represents a pairwise feature corresponding to the two nodes it connects.

The particle inputs and the interaction inputs are projected individually into latent embeddings using an MLP for each. The particle embedding is then operated on by a stack of layers implementing MHA (cf. Sec. 2.5), referred to as particle attention blocks (PABs). The introduction of attention mechanisms allows the model to focus on specific aspects of the particle embeddings, enhancing the network’s ability to capture relevant information from individual particles. At every PAB layer, the same embedded interaction matrix is used as a weight to augment the attention weights. This mechanism is crucial for incorporating information from pairwise interaction data into the particle embeddings, enabling the network to leverage the interplay between particles and interactions within a jet. The output from the last PAB, along with a global class token, is fed into two stacked Class Attention Blocks (CABs), which compute the attention score between the class token and all the particles. This allows the model to weigh the importance of different particles with respect to the class token, and selectively consider information from various particles when making a classification decision for the entire jet. The output from the second CAB is then passed to an MLP to produce the final classification score. This final step consolidates the information processed through the attention mechanisms and MLPs, in order to perform the classification.

In Ref. [97], the authors observe that the ParT model benefits from using a large dataset (Jet-Class) comprising  $\sim 100$ M jets, and that other transformer-based jet taggers fail to surpass the DGCNN-based ParticleNet (cf. Sec. 3.2.3) due to insufficient number of jets in the training samples. This indicates that models based on transformers are effective at harnessing larger training datasets by utilizing the attention mechanism.

A version of the ParT architecture was implemented in CMS [98] in early Run-3 using three PABs and one CAB, with 8 attention heads per block. The ParT-based b tagger achieves about 1% (12%) higher b jet efficiency at a light-flavor (c) jet mistag rate of 1%, when compared to DeepJet-based b tagger. This gain is about 15% (35%) for jets with  $p_T > 300$  GeV at same mistag values. Similarly, the ParT-based c tagger improves c jet efficiency by 14% (20%) compared to DeepJet at a light-flavor (b) jet mistag rate of 1% for an inclusive selection of jets, and almost 60% (60%) for jets with  $p_T > 300$  GeV. Detailed performance comparisons can be found in Ref. [98]. These indicate that the ParT architecture results in substantial gains for high- $p_T$  jets, which are usually underrepresented in tagger trainings.

In the ATLAS experiment, the flavor tagging performance is further improved using the transformer architecture to supersede the GN1 tagger with the GN2 tagger [100]. The attention type was been changed to *ScaledDotProduct* [34]. Although the change in attention type has no impact on physics performance, it reduces training time and memory usage. In GN2, the computation of attention weights is decoupled from the calculation of updated node representations, incorporating a dense layer between the attention layers. This introduces additional flexibility and complexity into the model and allows the model to gain the capacity to learn more intricate relationships and dependencies within the graph. The learning rate in GN2 is based on the *OneCycle* learning rate scheduler [101], that adjusts the learning rate during training, allowing for a faster convergence during the initial phase and a finer exploration of the loss landscape towards the end. The number of GNN layers is enhanced to 6, compared to 3 in GN1, enabling GN2 to capture more complex hierarchical patterns and dependencies within the graph. The number of learnable parameters in GN2 is 1.5M, which is almost twice of that in GN1. The number of attention heads is also enhanced to 8 and the training dataset size was expanded from  $\sim 30$  million jets to  $\sim 192$  million jets. These updates resulted in an improvement in the light-flavor (c) jet rejection by a factor of 2 (1.5) compared to GN1, at a b

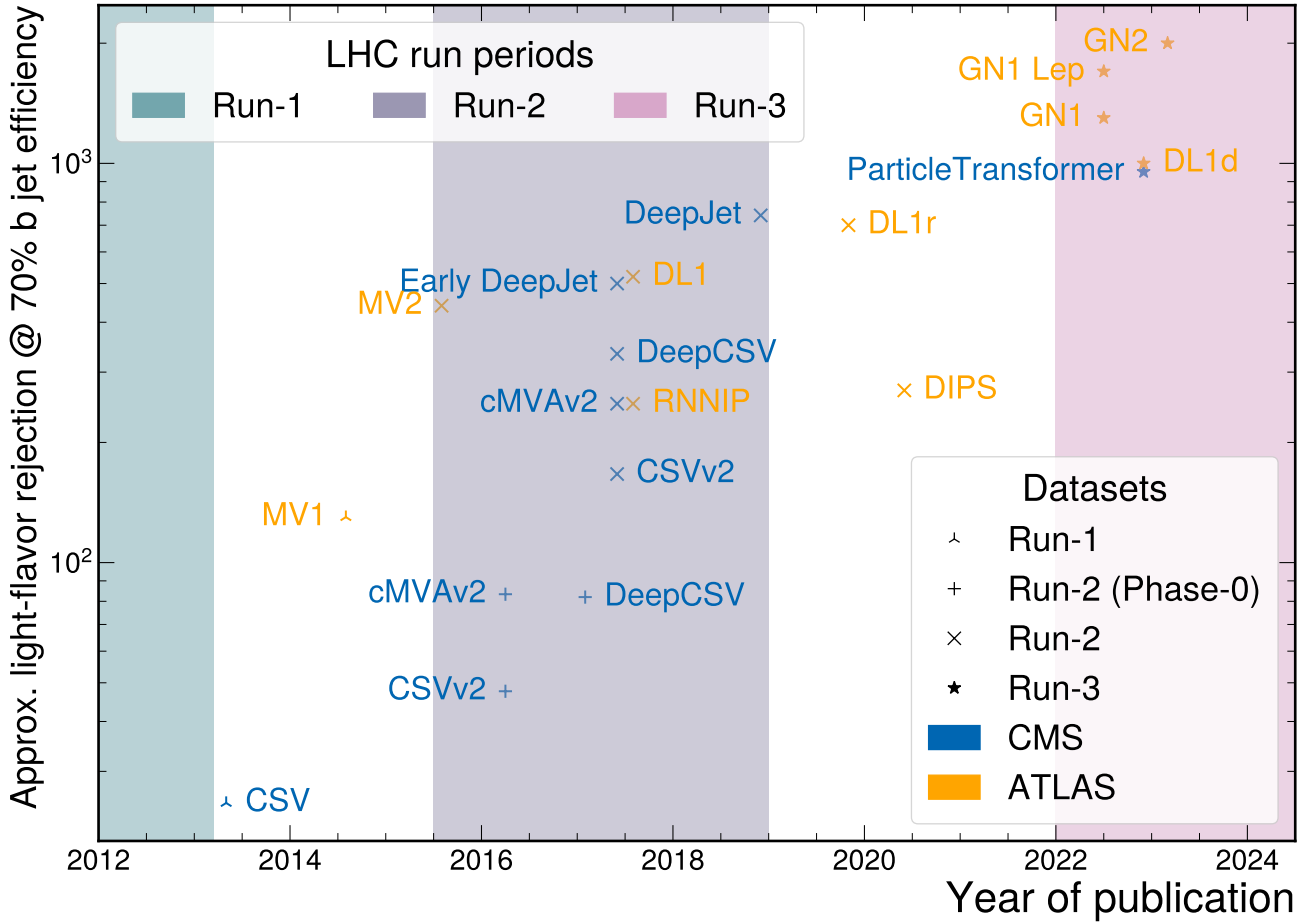


Figure 1: Approximate light-flavor jet rejection rates at a b jet efficiency of 70% (y-axis) achieved by different ML-based single-pronged jet flavor tagging algorithms used in the CMS and ATLAS experiments. The x-axis shows the approximate years the algorithms were published by the respective experiments. The vertical color bands indicate the timespan of the runs at the LHC, while the different markers indicate different datasets reconstructed with different run conditions on which the taggers were trained and/or evaluated. Run-2 data reconstruction in CMS is split into two parts corresponding to before (Phase-0) and after the CMS Phase-1 pixel detector upgrade [99]. It is important to note that (a) the CMS and ATLAS detectors, along with their subsystems, possess different detection and reconstruction capabilities, (b) there are differences in jet reconstruction techniques and quality criteria between the two experiments, (c) these techniques and criteria evolve over time, and (d) definitions of tagger scores, and hence the degree of discrimination specifically against light-flavor jets, vary between experiments (e.g. using tagger output scores versus employing transformations like that in eq. 2). Consequently, directly comparing tagger performance across experiments may not accurately reflect algorithm capabilities and must be interpreted carefully.

jet efficiency of 70% [91], once again demonstrating the impact of enhanced attention mechanisms and more complex network architectures capable of leveraging larger training datasets.

## 3.2 Multi-pronged jet tagging

Lorentz-boosted heavy particles decaying into heavy flavor quarks are usually reconstructed with one large-radius jet. The CMS experiment used AK jets with a distance parameter of 0.8 (AK8) and 1.5 (AK15) in Run-2, while the ATLAS experiment used a distance parameter of 1.0 (AK10). In Run-1, jets clustered with the Cambridge/Aachen jet clustering algorithm [102] with  $\Delta R = 0.8$  (CA8) and  $\Delta R = 1.5$  (CA15) were also used in CMS [103].

### 3.2.1 First generation: BDTs and shallow NNs

In early Run-2 (late Run-1), the CMS experiment demonstrated the effect of using the same CSVv2 (CSV) algorithm (cf. Sec. 3.1.1) developed for thin jets to run b tagging on AK8 jets [51] (CA8 and CA15 jets [103, 104]). This was achieved by using a set of looser track-to-jet and vertex-to-jet requirements for the constituents of an AK8 jet in Run-2, in order to account for all tracks and vertices within the effective radius of the jet [51]. A second approach is to apply the CSVv2 algorithm on the subjets of the AK8 jets, where the subjets are obtained by undoing the last step of the AK algorithm. A yet another approach is to match AK4 jets with AK8 jets by requiring the angular separation between the two jet axes to be small (e.g.  $< 0.4$ ). In this case, an AK8 jet is considered tagged based on the CSVv2 score of at least one of the AK4 jets matched to the AK8 jet.

The tagging performance of these strategies are dependent on the type of signal and backgrounds being considered and the Lorentz boost of the originating particle. In the case of tagging a Higgs (H) boson decaying into a pair of b quarks (H $\rightarrow$ bb) against a gluon decaying to a pair of

b quarks (g $\rightarrow$ b $\bar{b}$ ), the subjet tagging strategy achieves about 70% higher signal tagging efficiency ( $\sim 25\%$  against  $\sim 15\%$ ), compared to the strategy of tagging the entire AK8 jet, at jet  $p_T$  between 300 and 500 GeV and a background mistag rate of 10%. Tagging the AK8 jet outperforms tagging matched AK4 jets at these  $p_T$  ranges. On the other hand, at jet  $p_T$  above 1200, the subjet tagging method still outperforms the other two approaches, but the matched-AK4 tagging approach outperforms the AK8 tagging approach. Detailed performance comparisons can be found in Ref. [51].

The first dedicated tagger for di-pronged jets in CMS was also developed in Run-2 and was named *double-b* [51, 105]. The double-b algorithm was developed to exploit not only the presence of two b hadrons inside the AK8 jet but also the correlation between the directions of the momenta of the two b hadron. Besides using the same variables as CSVv2, the double-b algorithm used track and SV kinematics computed with respect to the *jet subtiness* ( $\tau$ ) axes [106, 107]. Furthermore, tracks from all the SVs associated with a given  $\tau$  axis are combined to define the SV mass and  $p_T$  corresponding to that axis. A total of 27 jet features are then combined using a BDT that is trained to discriminate H $\rightarrow$ b $\bar{b}$  jets from jets in multijet events. The double-b algorithm slightly exceeds the performance of subjet tagging at intermediate boosts but improves bb jet tagging efficiency by about a factor of 2 (30% against 15%) at high boosts ( $p_T > 1.2$  TeV), at a background mistag rate of 10%. Thus the double-b algorithm brings a substantial improvement to searches for heavy resonances that are expected to have high- $p_T$  jets in the final state.

Similar to CMS, the ATLAS experiment used a combination of AK10 jets and small-radius AK jets with  $\Delta R = 0.2$  or  $0.3$  to identify boosted heavy particles in Run-1 [108] and early Run-2 [109–111]. The AK10 jet reconstructed from calorimeter information was used to reconstruct the decay, while b tagging algorithms were run on the individual small-radius track jets<sup>3</sup> matched to

<sup>3</sup>Track jets are reconstructed by applying jet clustering on the tracks detected in the inner tracker. These jets pro-



the AK10 jet. The MV2 algorithm evaluated on each thin jet, along with requirements on jet sub-structure variables associated with the AK10 jet, was used to tag these decays.

In late Run-2, ATLAS made use of variable-radius (VR) track jets associated with AK10 jets [113] for di-pronged b tagging. VR jets are clustered using the VR AK algorithm [114, 115] that uses a varying value of  $\Delta R$  depending on the jet  $p_T$  and a constant parameter  $\rho$ , such that the radius scales as  $\rho/p_T$ . As this discussion involves the use of both BDT- and RNN-based taggers, it is again followed up in the next subsection.

### 3.2.2 Second generation: Early DNNs

The late Run-2 approach of the ATLAS experiment towards di-pronged b tagging evolved with the use of VR track jets associated with an AK10 jet. Reference [116] introduces three new approaches; the first two involve tagging the individual VR track jets with the BDT-based MV2 and the RNN-based DL1r taggers, respectively, the latter being optimized dedicatedly for VR jets [116]. The third approach [113, 117] involves using a feedforward DNN to combine the outputs of the single-pronged tagging algorithms along with AK10 jet kinematics to predict the probability of the AK10 jet containing the entire heavy particle decay. The inputs to the NN include kinematics of the fat jet, and output nodes of the DL1r (or MV2) algorithm (probability of the jet being b, c, or light-flavor) run on up to three VR jets associated with the AK10 jet. The network architecture consists of six fully-connected layers, each with 250 nodes. Since DNNs require fixed-length inputs, inputs corresponding to any missing subjets are replaced with the mean input values when fewer than 3 subjets are associated with the AK10 jet. In this paper, the third tagger is denoted by  $D_{Xbb}$ .

The baseline DL1r-based VR subjet tagging approach improved the rejection of multijet (top)

---

vide information complementary to that obtained from jets reconstructed with calorimeter information. More details can be found in Ref. [112].

background by a factor of  $\sim 1.8$  ( $\sim 1.2$ ) compared to the MV2-based VR tagging approach, at a Higgs efficiency of 70%. At the same signal efficiency, the  $D_{Xbb}$  tagger has a similar rejection for multijet background, but improves top rejection by a factor of  $\sim 1.9$ . On the other hand, at high efficiencies of the Higgs signal ( $\sim 93\%$ ), the  $D_{Xbb}$  tagger rejects about 9 (1.6) times more multijet (top) background as compared to the VR-based MV2 tagger, while the DL1r performance remains similar to that at low Higgs boson efficiencies. As neither the mass of the AK10 jet, nor any variables largely correlated with the mass, is used as input to  $D_{Xbb}$ , the multijet background jets mistagged by the  $D_{Xbb}$  algorithm do not exhibit an artificial mass peak. This is an important feature of the tagger as is discussed in the following paragraphs.

In Run-2, the CMS experiment developed the DeepAK8 algorithm [118]. This multiclass classifier was designed to identify hadronically-decaying heavy particles, with 5 main classes in the output node corresponding to W, Z, and H bosons, top (t) quark, and other decays. Furthermore, each main category is subdivided into minor categories depending on the decay of each particle into a pair of b, c, or light-flavor quarks. Thus the DeepAK8 algorithm is also able to perform di-pronged flavor tagging.

The DeepAK8 algorithm uses two lists of inputs corresponding to constituent particles and SVs, respectively. The “particle” list includes 42 variables per particle, including  $p_T$ , energy, charge, angular separation between the particle and the jet axis or the subjet axes, and displacement and quality of the tracks. The “SV” list includes 15 features of SVs including the kinematics, displacement, and quality criteria. The “particle” and “SV” lists are sorted (in a descending order) by the particle  $p_T$  and the two-dimensional IP significance, respectively, and the features of only the first 100 particles and first 7 SVs are used.

As is the case for DeepJet (cf. Sec. 3.1.2) the large input feature space is first processed with two separate one-dimensional CNNs corresponding to the “particle” and “SV” lists, respectively. The CNN structure is based on the ResNet model

[119] and is aimed at reducing the dimensionality and keeping only the relevant information in the intermediate latent space. The CNN for the “particle” (“SV”) list has 14 (10) layers. The output of the CNNs are fed into a single-layer fully-connected layer with 512 nodes. This is followed by the output layer with nodes corresponding to the several particle and decay categories.

Even though the mass of the jet is not explicitly used as input to DeepAK8, the mass can be inferred by the network from the kinematics of the particle inputs. Hence, the DeepAK8 network is able to leverage the difference in invariant masses of the W/Z/H/t particles to make its predictions. This is, however, an undesirable feature in several physics analyses as analyses often rely on leveraging mass peaks corresponding to heavy resonances in a continuum of background. The baseline DeepAK8 tagger preferentially selects jets with masses close to masses of the signal particles and thereby *sculpts* the background jet mass spectrum to artificially peak at these masses. This phenomenon, referred to as *mass sculpting*, is detrimental to the next steps of physics analyses that may involve using the mass of jets to differentiate between the signal and background templates. Therefore, a second version of the DeepAK8 algorithm, dubbed DeepAK8-MD (“mass-decorrelated”), is trained not to utilize the mass of the jet in performing classification. In addition to the baseline version, the DeepAK8-MD variant includes a mass prediction network, consisting of 3 fully-connected layers, each with 256 nodes. This network is trained to predict the mass of a background jet from the features extracted by the CNNs. The accuracy of the mass prediction for the background jets is then added as a penalty term to the loss function, thereby preventing the CNNs from learning features that are correlated with the mass.

The baseline version of the DeepAK8 algorithm improves the efficiency of selecting  $H \rightarrow b\bar{b}$  jets by a factor of  $\sim 1.7$  compared to the double-b tagger, at a background mistag rate of 0.1% and intermediate boosts of the jets. The DeepAK8-MD tagger has a performance intermediate to double-b and DeepAK8, as it is prevented from using the mass

of the jet to discriminate signal from background. The improvement brought by the baseline version is by a factor of  $\sim 2.2$  at high  $p_T$  ( $> 1000$  GeV), compared to double-b.

Run-2 of CMS also saw the development of the DeepDoubleX series of taggers [120, 121]. These are three different binary classifiers that aim to classify decays of the Higgs boson into a pair of b quarks, c quarks, or light-flavor quarks. The three versions called BvsL, CvsL, and CvsB, as the names suggest, perform binary classification of AK8 jets to distinguish b from light-flavor, c from light-flavor, and c from b jets, respectively. Similar to DeepJet, DeepDoubleX uses properties of charged particles, neutral particles, and SVs associated with the jet. After pruning, the first 21, 8, and 7 features, respectively, from each collection that provide the most relevant information, are kept. Additionally, the 5 most relevant features among the inputs to the double-b algorithm are also kept. For each of the charged, neutral, and SV collections, separate  $1 \times 1$  convolutions, each with 2 hidden layers with 32 filters each, are trained. Each of the outputs from the CNNs is then separately is then fed into a GRU with 50 output nodes. The outputs of the GRUs, along with the global jet properties, are then processed by a dense layer with 100 nodes. The output layer then consists of two nodes, which represent either BvsL, CvsL, or CvsB, depending on the variant.

While earlier versions of the DeepDoubleX models, labelled “v0” [120], used a similar mass decorrelation technique as DeepAK8-MD, later versions achieve mass-decorrelation more “natively” by using simulated samples of Higgs-like particles with variable masses. This makes the DeepDoubleX taggers suitable for tagging any heavy particle resonance decaying into a pair of b or c quarks. The BvsL tagger improves bb tagging efficiency by a factor of 2 compared to double-b at a multijet mistag rate of 1%. The CvsL (CvsB) tagger achieves a cc tagging efficiency of  $\sim 0.48$  ( $\sim 0.38$ ) at a multijet (bb) mistag rate of 1%.

### 3.2.3 Third generation: Sets and clouds

The ATLAS collaboration developed the DeXTer (Deep set  $X \rightarrow bb$  Tagger) [122] after the Run-2 data-taking period, inspired by the architecture of the Deep Set-based DIPS tagger for single-pronged jets. The DeXTer algorithm is tailored towards tagging low- $p_T$  ( $p_T \lesssim 200$  GeV) jets that may arise, e.g., from low-mass beyond-SM particles that decay into a pair of b quarks [123–125]. For this purpose, large-radius track jets are clustered using the AK algorithm with  $\Delta R = 0.8$ , using an extended list of tracks found around a small-radius particle-flow [126] jet (AK4) with  $p_T > 20$  GeV, with a similar approach as that adopted in Ref. [127]. This ensures that the particles that may escape the 0.4 radius of the AK4 jet are captured by the large-radius track jet.

The DeXTer algorithm uses two separate feedforward NNs to extract features from the tracks and SVs associated to the AK8 track jet. Each of these contain 2 hidden layers with 100 nodes each, and an output layer with 128 nodes. The inputs to the track (SV) NN are sorted by the IP (decay length) significance of the tracks (SVs) and the first 25 (12) tracks (SVs) are used as inputs. Even though an artificial ordering is imposed at this stage to select the inputs, the NNs are permutation invariant in the same way as in the DIPS architecture. The 256 output nodes from the two NNs, combined with features relating to the jet’s kinematics, are used as inputs to a global feedforward NN with 3 hidden layers. The output layer of this NN consists of three nodes, representing the probability of the jet containing two b quarks, one b quark, or only light-flavor quarks.

Deep Set-based implementations of jet taggers, however, process particles only in a global way. This falls short of the advantage that CNNs provide—capturing local information in a global context by using multiple hierarchical layers with increasing levels of abstraction. On the other hand, DGCNNs overcome this challenge with stackable EdgeConv operations, which achieve hierarchical representations similar to CNNs, starting with an initial particle cloud representation (cf. Sec. 2.8).

The ParticleNet tagger [128] implements the DGCNN architecture with jets representations as inputs in order to fully exploit the potentials of a particle cloud representation. It makes use of the EdgeConv operation starting with jet constituents expressed with their associated features and coordinates in the pseudorapidity–azimuth plane of the detector. For each particle, its features and those of the  $k$  (e.g.  $k = 16$ ) nearest neighboring particles are used to find the transformed “edge features” and EdgeConv operation for the particle. A total of 3 EdgeConv layers are used, such that the physical coordinates of the jet constituents are used as inputs to the first layer, while transformed feature vectors generated as outputs from each layer are used as inputs to the subsequent layer. Following the EdgeConv blocks, the outputs are processed by a global average pooling layer and a fully-connected layer to produce the outputs.

The ParticleNet tagger was implemented in CMS in conjunction with AK8 and AK15 jets after Run-2 [129]. Similar to DeepAK8, the CMS implementation of ParticleNet identifies W/Z/H bosons and classifies them by their decay modes (bb/cc/light-flavor). The CMS implementation of the ParticleNet tagger uses SV features as inputs, in addition to particle features.

A second variant of the ParticleNet tagger, named ParticleNet-MD [129], is trained with simulated samples containing Higgs-like particles with a flat mass spectrum between 15–250 GeV. This tagger contains 4 output nodes, corresponding to the probabilities  $P(X \rightarrow bb)$ ,  $P(X \rightarrow cc)$ ,  $P(X \rightarrow qq)$ , and  $P(\text{QCD})$ . Deriving from these, flavor tagging discriminants, defined as

$$D_{X \rightarrow bb} = \frac{P(X \rightarrow bb)}{P(X \rightarrow bb) + P(\text{QCD})},$$

$$D_{X \rightarrow cc} = \frac{P(X \rightarrow cc)}{P(X \rightarrow cc) + P(\text{QCD})},$$

are useful in tagging heavy particles decaying to a pair of b quarks and a pair of c quarks, respectively, against the massive, irreducible QCD background usually encountered in physics analyses. ParticleNet-MD improves  $H \rightarrow b\bar{b}$  tagging efficiency by  $\sim 30\%$  compared to DeepAK8-MD,

at a QCD mistag rate of 0.1%. The improvement is around  $\sim 50\%$  for ( $H \rightarrow c\bar{c}$ ) tagging. The ParticleNet-MD tagger was also trained for AK15 jets [130, 131] in CMS. Detailed performance comparison between double-b, DeepDoubleX, DeepAK8-MD, and ParticleNet-MD in CMS collision data can be found in Ref. [132]. Reference [132] also presents a novel use case of BDTs in calibration of fat jet flavor tagging algorithms.

Inspired by the transformer- and GNN-based GN2 algorithm for single-pronged b tagging, the ATLAS experiment developed the GN2X tagger [133] for di-pronged b tagging in Run-3. GN2X takes as input 3 variables ( $p_T$ , pseudorapidity, mass) of the jet, and 20 variables associated with each track contained in the jet. Up to 100 tracks (with the highest IP significance) per jet are used as inputs. The architecture is very similar to that of the GN2 tagger, with a per-track initialization network designed in the same way as Deep Set formalism without the reduction (summation) operation over track output features. The outputs of the initialization networks are fed into a transformer encoder [134]. A total of 6 encoder blocks with 4 attention heads are used. The outputs are combined to form a global representation by computing a weighted sum over the track representations, where the weights are attention weights learned in the training process. The global representation is used for classification. The output layer contains a node representing jets from hadronic top pair samples, besides the usual  $H \rightarrow b\bar{b}$ ,  $H \rightarrow c\bar{c}$ , and QCD multijet nodes. In addition, as is the case for GN1 and GN2, two auxiliary training objectives are also incorporated for GN2X. Track origin classification and computation of vertex compatibility of any pair of input tracks are achieved by extending the architecture to include three hidden layers containing 128, 64, and 32 nodes, respectively, for each task. Mass-sculpting of tagged background jets is avoided in the same way as in DeepDoubleX, i.e. by training the networks with signal samples containing artificial Higgs-like particles with a flat mass spectrum.

The GN2X tagger improves the rejection of top pair (QCD) jet backgrounds by a factor of  $\sim 1.7$

( $\sim 1.9$ ) compared to the  $D_{Xbb}$  algorithm at a  $H \rightarrow b\bar{b}$  signal efficiency of 70%. Similarly for  $H \rightarrow c\bar{c}$  tagging, the top pair (QCD) jet background rejection is improved by a factor of  $\sim 2.6$  ( $\sim 6.0$ ) at a  $H \rightarrow c\bar{c}$  signal efficiency of 70%. The  $H \rightarrow b\bar{b}$  rejection is also improved by a factor of  $\sim 3$  at the same  $H \rightarrow c\bar{c}$  efficiency, compared to  $D_{Xbb}$ .

Two additional variants of the GN2X tagger incorporating heterogenous inputs were also developed. They are referred to as GN2X+Subjets and GN2X+Flow. The former utilizes the kinematic and b tagging information of the VR subjets of the AK10 jet, where the subjets are tagged using the GN2 tagger. The latter variant uses Unified Flow Object (UFO) [135] constituents which includes the use of charged and neutral calorimeter information. GN2X+Subjets improves top jet rejection by a factor of 2 compared to baseline GN2X, at a  $H \rightarrow b\bar{b}$  signal efficiency of 70%, but reduces QCD rejection to almost half of the baseline. On the other hand, the GN2X+Flow architecture improves the top (QCD) jet rejection by a factor of  $\sim 1.6$  ( $\sim 1.3$ ) at the same efficiency.

## 4 Discussions and future prospects

Section 3 highlights how ML-based flavor tagging algorithms have evolved over time in the CMS and ATLAS experiments. These architectures have not only been utilized for flavor tagging but also paved the way for identifying hadronically-decaying heavy objects like the W boson and the top quark. At times, these experimental innovations have utilized progress from parallel developments in phenomenological studies. Since this review focuses only on flavor tagging algorithms that have been successfully used at the LHC, neither generalized heavy-object tagging nor the parallel developments in phenomenology have been discussed. However, it is important to note that several of the methods discussed here are the consequence of earlier research into QCD-motivated jet substructure methods. These earlier methods proposed and improved definitions of jet substructure variables that can be used to tag jets.

| Generation  | Main architecture | Single-pronged (AK4)   |             | Multi-pronged  |                               | Novelty  |
|---|-------------------|------------------------|-------------|----------------|-------------------------------|--|
|   |                   | CMS                    | ATLAS       | CMS (AK8/AK15) | ATLAS (AK10)                  |  |
| <b>First</b> (Run-1 and early Run-2, shallow architectures)         | Shallow NN        | CSVv2                  | MV1         | CSVv2          | MV1 using subjets             | Switch from likelihood methods to first ML-based methods.  |
|   | BDT               | c-tagger, cMVAv2       | MV2         | double-b       | MV2 using subjets             | Incorporate larger number of inputs.   |
| <b>Second</b> (Late Run-2, early deep NNs)                          | Deep dense NN     | DeepCSV                | DL1         |                |                               | Use information from multiple selected tracks. Multiclassification.                              |
|   | CNN               |                        |             | DeepAK8        |                               | Use a large number of particles per jet.   |
|   | RNN               |                        | RNNIP, DL1r |                | DL1r using subjets, $D_{Xbb}$ | Accommodate an arbitrary number of tracks per jet.   |
|   | CNN+RNN           | DeepJet                |             | DeepDoubleX    |                               | Avoid applying selection criteria; use all jet constituents of jets including neutral particles. |
| <b>Third</b> (Early Run-3, sets and particle cloud representations) | Deep Set          |                        | DIPS, DL1d  |                | DeXTer                        | Avoid sorting of jet constituents: permutation invariant.  |
|   | GNN               |                        | GN1         |                |                               | Use large number of track features, including pixel hits.  |
|   | DGCNN             |                        |             | ParticleNet    |                               | Assign auxiliary track origin and vertex finding tasks.  |
|   | Transformer       | ParticleTransformerAK4 | GN2         |                | GN2X                          | Leverage local features of point clouds using convolutions.                                      |
|   |                   |                        |             |                |                               | Leverage pairwise features, attention mechanisms, and larger training datasets.                  |

Table 2: Flavor tagging algorithms developed in the CMS and ATLAS experiments, classified by generations and main ML architecture used.

These observables utilize the internal kinematic properties of a highly boosted jet to identify W [106, 136], Z/H [137–140], and top [107, 141–146] jets, as well as to distinguish quark- and gluon-initiated jets [147–152].

Similarly, ML-based tagging strategies first introduced in phenomenological studies have also been used in developing modern flavor tagging algorithms in experiments. As successors of substructure-based approaches, ML-based studies have interpreted jets as detector images [153–165] using CNNs, particle sequences and trees [75, 79, 80, 166–176] using RNNs, sets using energy flow networks [89, 177], interactions using interaction networks [178], and graphs and particle clouds [97, 128, 179–193] using GNNs, DGCNNs, autoencoders [194], and transformers [34]. More detailed reviews of some of these algorithms can be found in Refs. [3, 195–197]. While early developments with ML provided performances at par with those obtained with substructure observables [160, 198], more recent developments have significantly surpassed earlier approaches in terms of performance, often at the cost of interpretability [199, 200]. Other phenomenological studies have proposed using novel jet reconstruction techniques to reconstruct hadronically-decaying heavy particles and/or enhance tagging performance [201–207].

Trends in the evolution of tagging algorithms indicate a steady shift towards leveraging increasingly lower levels of information along with advanced networks are that capable of extracting relevant features from a large number of inputs. As leveraging low-level inputs preserves information that might be lost in the process of constructing physically-meaningful quantities, future tagging algorithms in the ATLAS and CMS experiments are expected to directly use raw information from the detectors. On the other hand, the examples of improvements observed by adding auxiliary tasks in GN1/GN2 (cf. Sec. 3.1.3) and heterogenous inputs in GN2X+Flow (cf. Sec. 3.2.3) algorithms, indicate that using physics-motivated high-level quantities to steer the learning priorities of the network can bring about additional improvements in tagging performance. Thus, future taggers are

expected to leverage both detector-level information and high-level targets simultaneously, with mechanisms to effectively guide the network to learn physically-meaningful representations. Using detector-level information, however, requires even more advanced architectures that can condense and interpret a large number of input variables, along with faster, parallelizable computing methods, which are challenges for the broader ML community to address. Moreover, leveraging larger number of input features usually requires larger number of jets in the training process, and hence future training datasets may need  $\mathcal{O}(10^9)$  jets which poses significant computing and storage challenges.

However, with the use of lower-level information and attempts to extract every last bit of information available in the training sample, there is a possibility that newer taggers simply learn to leverage simulation-specific artifacts that do not reflect real collision data. Simulations of jets are, in turn, limited by the jet showering algorithms’ precision and underlying assumptions, among other factors. Therefore, it is inevitable that at some point any additional gains observed from new tagging algorithms are largely due to the tagger backtracing the steps in the algorithms used to generate the training samples or relying exceedingly on mismodeled features in simulation [208–210]. Such gains in performance in simulation would not necessarily be reflected when the algorithm is used in conjunction with collision data. Data-aware and unsupervised training methods [211–214] and adversarial training approaches [215, 216] have been proposed to mitigate this problem, but may adversely affect the best achievable performance in simulations. Therefore, while improved tagging performance on simulations are interesting from an ML point of view, they may not necessarily be indicative of the achievable performance in collision data. Recent studies have also used QCD-motivated observables to design more interpretable networks with fewer learnable parameters [217–225]. These approaches provide an alternative avenue for developing more robust taggers without relying exceedingly on features in simulations that are prone to mismodeling.

The above paragraph motivates the necessity to calibrate flavor tagging algorithms using collision data. Corrections are usually derived as Scale Factors (SFs) that measure the ratio of the tagging efficiency in data to that in simulation [51, 68, 69, 73, 83, 85, 86, 103, 129, 132, 226–230]. Performance measurements of the most recent taggers [85, 86, 132, 228, 230] using collision data have yielded SFs that are reasonably close to 1. Significant departures of the SFs from 1 in future taggers, if observed, would indicate that improvements seen on simulated samples are a result of the tagger leveraging simulation-only artifacts. In such cases, understanding and mitigating the sources of mismodeling in simulations should take precedence over improving tagger performance on simulations.

Another potential challenge for future taggers is the high level of pileup expected at the High Luminosity LHC (HL-LHC) [231]. An increased number of particles unrelated to the fragmentation of b/c hadrons in a jet will inevitably result in poorer tagging performance at high pileup environments. Improved implementations of attention-mechanisms and novel pileup mitigation techniques [232–236] are expected to mitigate such phenomena.

## 5 Conclusion

Flavor tagging in the CMS and ATLAS experiments at the LHC has seen significant evolution over the past decade. Tagging algorithms used in the experiments have been classified into three generations in this paper. The first generation of taggers, developed and implemented during Run-1 and early Run-2 of the LHC, mark the first departure from physics-motivated likelihood methods to shallow ML techniques. The second generation of taggers, developed during Run-2 and implemented on legacy Run-2 datasets, are characterized by the use of deep learning methods and a use of a larger number of inputs. The third generation, developed shortly before and during early Run-3, marks the use of more natural and physically-motivated representations for jets, the

use of even larger number of features of jet constituents, and the use of attention mechanisms.

The accuracies of these taggers, in case of both single-pronged and multi-pronged jets, have improved by leaps and bounds over the three generations. Notably, they have enabled the first observation of the Higgs boson decaying to a pair of bottom quarks [237, 238] and the Z boson decaying to a pair of charm quarks [131, 239] at the LHC. They have also enabled development of methods to measure the Higgs-charm coupling [131, 240] to a degree of accuracy that was earlier predicted to be achievable only at the HL-LHC [241, 242].

Future tagging algorithms are expected to incorporate increasingly lower-level information from the detectors, possibly even raw detector-level information, in conjunction with new architectures that are capable of extracting information from a large number of inputs. However, these approaches may be limited by mismodelings that are inevitable in simulations. Data-aware training methods are expected to circumvent this issue. The upcoming decades will undoubtedly witness fascinating advancements in the field of flavor tagging and the broader field of machine learning.

## References

- [1] Particle Data Group Collaboration, “Review of Particle Physics”, *PTEP* **2022** (2022) 083C01, [doi:10.1093/ptep/ptac097](https://doi.org/10.1093/ptep/ptac097).
- [2] I. J. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”. MIT Press, Cambridge, MA, USA, 2016.
- [3] D. Guest, K. Cranmer, and D. Whiteson, “Deep Learning and its Application to LHC Physics”, *Ann. Rev. Nucl. Part. Sci.* **68** (2018) 161–181, [doi:10.1146/annurev-nucl-101917-021019](https://doi.org/10.1146/annurev-nucl-101917-021019), [arXiv:1806.11484](https://arxiv.org/abs/1806.11484).
- [4] “LHC Machine”, *JINST* **3** (2008) S08001, [doi:10.1088/1748-0221/3/08/S08001](https://doi.org/10.1088/1748-0221/3/08/S08001).

- [5] CMS Collaboration, “The CMS Experiment at the CERN LHC”, *JINST* **3** (2008) S08004, [doi:10.1088/1748-0221/3/08/S08004](https://doi.org/10.1088/1748-0221/3/08/S08004).
- [6] ATLAS Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider”, *JINST* **3** (2008) S08003, [doi:10.1088/1748-0221/3/08/S08003](https://doi.org/10.1088/1748-0221/3/08/S08003).
- [7] E. B. Hunt, J. Marin, and P. J. Stone, “Experiments in induction.”. Academic press, 1966.
- [8] J. R. Quinlan, “Learning efficient classification procedures and their application to chess end games”, in *Machine learning*. Elsevier, 1983.
- [9] J. R. Quinlan, “Induction of decision trees”, *Machine learning* **1** (1986) 81–106.
- [10] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, *Journal of Computer and System Sciences* **55** (1997), no. 1, 119, [doi:https://doi.org/10.1006/jcss.1997.1504](https://doi.org/10.1006/jcss.1997.1504).
- [11] T. G. Dietterich, “Ensemble methods in machine learning”, in *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pp. 1–15. Springer-Verlag, Berlin, Heidelberg, 2000.
- [12] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification and Regression Trees”. Chapman and Hall/CRC, 1984.
- [13] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.”, *The Annals of Statistics* **29** (2001), no. 5, 1189 – 1232, [doi:10.1214/aos/1013203451](https://doi.org/10.1214/aos/1013203451).
- [14] F. Pedregosa et al., “Scikit-learn: Machine learning in Python”, *Journal of Machine Learning Research* **12** (2011) 2825–2830.
- [15] G. Van Rossum and F. L. Drake, “Python 3 Reference Manual”. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.
- [16] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August, 2016. [doi:10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [17] A. Hocker et al., “TMVA - Toolkit for Multivariate Data Analysis with ROOT: Users guide. TMVA - Toolkit for Multivariate Data Analysis”, technical report, CERN, Geneva, 2007. TMVA-v4 Users Guide: 135 pages, 19 figures, numerous code examples and references.
- [18] R. Brun and F. Rademakers, “ROOT: An object oriented data analysis framework”, *Nucl. Instrum. Meth. A* **389** (1997) 81–86, [doi:10.1016/S0168-9002\(97\)00048-X](https://doi.org/10.1016/S0168-9002(97)00048-X).
- [19] F. Rosenblatt, “The Perceptron, a Perceiving and Recognizing Automaton Project Para”. Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.
- [20] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.”, *Proceedings of the national academy of sciences* **79** (1982), no. 8, 2554–2558.
- [21] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, “Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark”, *arXiv e-prints* (September, 2021) arXiv:2109.14545, [doi:10.48550/arXiv.2109.14545](https://doi.org/10.48550/arXiv.2109.14545), [arXiv:2109.14545](https://arxiv.org/abs/2109.14545).
- [22] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biological Cybernetics* **36** (1980) 193–202, [doi:10.1007/BF00344251](https://doi.org/10.1007/BF00344251).
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”,



- Communications of the ACM* **60** (2012) 84 – 90.
- [24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation”, technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [25] A. Graves, “Supervised Sequence Labelling”. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.  
[doi:10.1007/978-3-642-24797-2\\_2](https://doi.org/10.1007/978-3-642-24797-2_2).
- [26] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural computation* **9** (1997), no. 8, 1735–1780.
- [27] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, *arXiv e-prints* (June, 2014) arXiv:1406.1078,  
[doi:10.48550/arXiv.1406.1078](https://doi.org/10.48550/arXiv.1406.1078),  
[arXiv:1406.1078](https://arxiv.org/abs/1406.1078).
- [28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, *arXiv e-prints* (December, 2014) arXiv:1412.3555,  
[doi:10.48550/arXiv.1412.3555](https://doi.org/10.48550/arXiv.1412.3555),  
[arXiv:1412.3555](https://arxiv.org/abs/1412.3555).
- [29] T. Mikolov et al., “Recurrent neural network based language model”, in *Interspeech*. 2010.
- [30] J. Connor, R. Martin, and L. Atlas, “Recurrent neural networks and robust time series prediction”, *IEEE Transactions on Neural Networks* **5** (1994) 240–254, [doi:10.1109/72.279188](https://doi.org/10.1109/72.279188).
- [31] Z. Che et al., “Recurrent Neural Networks for Multivariate Time Series with Missing Values”, *arXiv e-prints* (June, 2016) arXiv:1606.01865,  
[doi:10.48550/arXiv.1606.01865](https://doi.org/10.48550/arXiv.1606.01865),  
[arXiv:1606.01865](https://arxiv.org/abs/1606.01865).
- [32] M.-T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-based Neural Machine Translation”, *arXiv e-prints* (August, 2015) arXiv:1508.04025,  
[doi:10.48550/arXiv.1508.04025](https://doi.org/10.48550/arXiv.1508.04025),  
[arXiv:1508.04025](https://arxiv.org/abs/1508.04025).
- [33] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks”, *arXiv e-prints* (March, 2013) arXiv:1303.5778,  
[doi:10.48550/arXiv.1303.5778](https://doi.org/10.48550/arXiv.1303.5778),  
[arXiv:1303.5778](https://arxiv.org/abs/1303.5778).
- [34] A. Vaswani et al., “Attention is All you Need”, in *Advances in Neural Information Processing Systems*, I. Guyon et al., eds., volume 30. Curran Associates, Inc., 2017.
- [35] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate”, *arXiv e-prints* (September, 2014) arXiv:1409.0473,  
[doi:10.48550/arXiv.1409.0473](https://doi.org/10.48550/arXiv.1409.0473),  
[arXiv:1409.0473](https://arxiv.org/abs/1409.0473).
- [36] M. Zaheer et al., “Deep Sets”, *arXiv e-prints* (March, 2017) arXiv:1703.06114,  
[doi:10.48550/arXiv.1703.06114](https://doi.org/10.48550/arXiv.1703.06114),  
[arXiv:1703.06114](https://arxiv.org/abs/1703.06114).
- [37] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”, *arXiv e-prints* (June, 2016) arXiv:1606.09375,  
[doi:10.48550/arXiv.1606.09375](https://doi.org/10.48550/arXiv.1606.09375),  
[arXiv:1606.09375](https://arxiv.org/abs/1606.09375).
- [38] M. M. Bronstein et al., “Geometric deep learning: Going beyond euclidean data”, *IEEE Signal Processing Magazine* **34** (jul, 2017) 18–42,  
[doi:10.1109/msp.2017.2693418](https://doi.org/10.1109/msp.2017.2693418).
- [39] P. W. Battaglia et al., “Relational inductive biases, deep learning, and graph networks”, *arXiv e-prints* (2018) arXiv:1806.01261,

- [doi:10.48550/arXiv.1806.01261](https://doi.org/10.48550/arXiv.1806.01261),  
[arXiv:1806.01261](https://arxiv.org/abs/1806.01261).
- [40] Y. Wang et al., “Dynamic Graph CNN for Learning on Point Clouds”, *ACM Trans. Graph.* **38** (2019) [doi:10.1145/3326362](https://doi.org/10.1145/3326362).
- [41] R. Alemany-Fernandez et al., “Operation and Configuration of the LHC in Run 1”,.
- [42] J. Wenninger, “Operation and Configuration of the LHC in Run 2”,.
- [43] S. Fartouk et al., “LHC Configuration and Operational Scenario for Run 3”, technical report, CERN, Geneva, 2021.
- [44] M. Cacciari, G. P. Salam, and G. Soyez, “The anti- $k_T$  jet clustering algorithm”, *Journal of High Energy Physics* (2008) 063, [doi:10.1088/1126-6708/2008/04/063](https://doi.org/10.1088/1126-6708/2008/04/063).
- [45] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet User Manual”, *Eur. Phys. J. C* **72** (2012) 1896, [doi:10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2), [arXiv:1111.6097](https://arxiv.org/abs/1111.6097).
- [46] CMS Collaboration, “Identification of b-Quark Jets with the CMS Experiment”, *JINST* **8** (2013) P04013, [doi:10.1088/1748-0221/8/04/P04013](https://doi.org/10.1088/1748-0221/8/04/P04013), [arXiv:1211.4462](https://arxiv.org/abs/1211.4462).
- [47] W. Waltenberger, R. Frühwirth, and P. Vanlaer, “Adaptive vertex fitting”, *Journal of Physics G: Nuclear and Particle Physics* **34** (2007) N343, [doi:10.1088/0954-3899/34/12/N01](https://doi.org/10.1088/0954-3899/34/12/N01).
- [48] W. Waltenberger, “Adaptive Vertex Reconstruction”, technical report, CERN, Geneva, 2008.
- [49] CMS Collaboration, “Measurement of  $B\bar{B}$  Angular Correlations based on Secondary Vertex Reconstruction at  $\sqrt{s} = 7$  TeV”, *JHEP* **03** (2011) 136, [doi:10.1007/JHEP03\(2011\)136](https://doi.org/10.1007/JHEP03(2011)136), [arXiv:1102.3194](https://arxiv.org/abs/1102.3194).
- [50] CMS Collaboration, “Identification of b quark jets at the CMS Experiment in the LHC Run 2”, CMS Physics Analysis Summary CMS-PAS-BTV-15-001, CERN, Geneva, 2016.
- [51] CMS Collaboration, “Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV”, *JINST* **13** (2018) P05011, [doi:10.1088/1748-0221/13/05/P05011](https://doi.org/10.1088/1748-0221/13/05/P05011), [arXiv:1712.07158](https://arxiv.org/abs/1712.07158).
- [52] CMS Collaboration, “Performance of b-Tagging Algorithms in 25ns Data at 13TeV”, CMS Detector Performance Note CMS-DP-2015-056, 2015.
- [53] CMS Collaboration, “Performance of heavy flavour identification algorithms in proton-proton collisions at 13 TeV at the CMS experiment”, CMS Detector Performance Note CMS-DP-2017-012, 2017.
- [54] ATLAS Collaboration, “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”, *Phys. Lett. B* **716** (2012) 1–29, [doi:10.1016/j.physletb.2012.08.020](https://doi.org/10.1016/j.physletb.2012.08.020), [arXiv:1207.7214](https://arxiv.org/abs/1207.7214).
- [55] CMS Collaboration, “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC”, *Phys. Lett. B* **716** (2012) 30–61, [doi:10.1016/j.physletb.2012.08.021](https://doi.org/10.1016/j.physletb.2012.08.021), [arXiv:1207.7235](https://arxiv.org/abs/1207.7235).
- [56] CMS Collaboration, “A portrait of the Higgs boson by the CMS experiment ten years after the discovery.”, *Nature* **607** (2022), no. 7917, 60–68, [doi:10.1038/s41586-022-04892-x](https://doi.org/10.1038/s41586-022-04892-x), [arXiv:2207.00043](https://arxiv.org/abs/2207.00043).
- [57] CMS Collaboration, “Identification of c-quark jets at the CMS experiment”, CMS Physics Analysis Summary CMS-PAS-BTV-16-001, CERN, Geneva, 2016.

- [58] ATLAS Collaboration, “Performance of  $b$ -jet Identification in the ATLAS Experiment”, *JINST* **11** (2016) P04008, [doi:10.1088/1748-0221/11/04/P04008](https://doi.org/10.1088/1748-0221/11/04/P04008), [arXiv:1512.01094](https://arxiv.org/abs/1512.01094).
- [59] ATLAS Collaboration, “Secondary vertex finding for jet flavour identification with the ATLAS detector”, ATLAS Note ATL-PHYS-PUB-2017-011, CERN, Geneva, 2017.
- [60] ATLAS Collaboration, “Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics”, [arXiv:0901.0512](https://arxiv.org/abs/0901.0512).
- [61] ATLAS Collaboration, “Topological  $b$ -hadron decay reconstruction and identification of  $b$ -jets with the JetFitter package in the ATLAS experiment at the LHC”, ATLAS Note ATL-PHYS-PUB-2018-025, CERN, Geneva, 2018.
- [62] “Calibration of the performance of  $b$ -tagging for  $c$  and light-flavour jets in the 2012 ATLAS data”, ATLAS Note ATL-COEF-2014-046, CERN, Geneva, 2014.
- [63] “Expected performance of the ATLAS  $b$ -tagging algorithms in Run-2”, ATLAS Note ATL-PHYS-PUB-2015-022, CERN, Geneva, 2015.
- [64] ATLAS Collaboration, “Commissioning of the ATLAS  $b$ -tagging algorithms using  $t\bar{t}$  events in early Run-2 data”, ATLAS Note ATL-PHYS-PUB-2015-039, CERN, Geneva, 2015.
- [65] ATLAS Collaboration, “Optimisation and performance studies of the ATLAS  $b$ -tagging algorithms for the 2017-18 LHC run”, ATLAS Note ATL-PHYS-PUB-2017-013, CERN, Geneva, 2017.
- [66] ATLAS Collaboration, “Optimisation of the ATLAS  $b$ -tagging performance for the 2016 LHC Run”, ATLAS Note ATL-PHYS-PUB-2016-012, CERN, Geneva, 2016.
- [67] ATLAS Collaboration, “Measurements of  $b$ -jet tagging efficiency with the ATLAS detector using  $t\bar{t}$  events at  $\sqrt{s} = 13$  TeV”, *JHEP* **08** (2018) 089, [doi:10.1007/JHEP08\(2018\)089](https://doi.org/10.1007/JHEP08(2018)089), [arXiv:1805.01845](https://arxiv.org/abs/1805.01845).
- [68] ATLAS Collaboration, “Calibration of the ATLAS  $b$ -tagging algorithm in  $t\bar{t}$  semi-leptonic events”, ATLAS Note, CERN, Geneva, 2018.
- [69] ATLAS Collaboration, “Calibration of light-flavour  $b$ -jet mistagging rates using ATLAS proton-proton collision data at  $\sqrt{s} = 13$  TeV”, ATLAS Note ATL-COEF-2018-006, CERN, Geneva, 2018.
- [70] CMS Collaboration, “Heavy flavor identification at CMS with deep neural networks”, CMS Detector Performance Note CMS-DP-2017-005, 2017.
- [71] F. Chollet et al., “Keras”. <https://keras.io>, 2015.
- [72] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems”, 2015. Software available from [tensorflow.org](https://tensorflow.org). <https://www.tensorflow.org/>.
- [73] ATLAS Collaboration, “ATLAS  $b$ -jet identification performance and efficiency measurement with  $t\bar{t}$  events in pp collisions at  $\sqrt{s} = 13$  TeV”, *Eur. Phys. J. C* **79** (2019) 970, [doi:10.1140/epjc/s10052-019-7450-8](https://doi.org/10.1140/epjc/s10052-019-7450-8), [arXiv:1907.05120](https://arxiv.org/abs/1907.05120).
- [74] I. J. Goodfellow et al., “Maxout Networks”, *arXiv e-prints* (February, 2013) [arXiv:1302.4389](https://arxiv.org/abs/1302.4389), [doi:10.48550/arXiv.1302.4389](https://doi.org/10.48550/arXiv.1302.4389), [arXiv:1302.4389](https://arxiv.org/abs/1302.4389).
- [75] ATLAS Collaboration, “Identification of Jets Containing  $b$ -Hadrons with Recurrent

- Neural Networks at the ATLAS Experiment”, ATLAS Note ATL-PHYS-PUB-2017-003, 2017.
- [76] ATLAS Collaboration, “Search for new resonances in mass distributions of jet pairs using 139 fb<sup>-1</sup> of  $pp$  collisions at  $\sqrt{s} = 13$  TeV with the ATLAS detector”, *JHEP* **03** (2020) 145, [doi:10.1007/JHEP03\(2020\)145](https://doi.org/10.1007/JHEP03(2020)145), [arXiv:1910.08447](https://arxiv.org/abs/1910.08447).
- [77] ATLAS Collaboration, “ATLAS flavour-tagging algorithms for the LHC Run 2 pp collision dataset”, *Eur. Phys. J. C* **83** (2023) 681, [doi:10.1140/epjc/s10052-023-11699-1](https://doi.org/10.1140/epjc/s10052-023-11699-1), [arXiv:2211.16345](https://arxiv.org/abs/2211.16345).
- [78] E. Bols et al., “Jet Flavour Classification Using DeepJet”, *JINST* **15** (2020) P12012, [doi:10.1088/1748-0221/15/12/P12012](https://doi.org/10.1088/1748-0221/15/12/P12012), [arXiv:2008.10519](https://arxiv.org/abs/2008.10519).
- [79] CMS Collaboration, “CMS Phase 1 heavy flavour identification performance and developments”, CMS Detector Performance Note CMS-DP-2017-013, 2017.
- [80] CMS Collaboration, “Performance of the DeepJet b tagging algorithm using 41.9/fb of data from proton-proton collisions at 13TeV with Phase 1 CMS detector”, CMS Detector Performance Note CMS-DP-2018-058, 2018.
- [81] S. Haykin and B. Kosko, “GradientBased Learning Applied to Document Recognition”, pp. 306–351. 2001. [doi:10.1109/9780470544976.ch9](https://doi.org/10.1109/9780470544976.ch9).
- [82] CMS Collaboration, “Performance of b tagging algorithms in proton-proton collisions at 13 TeV with Phase 1 CMS detector”, CMS Detector Performance Note CMS-DP-2018-033, 2018.
- [83] CMS Collaboration, “A new calibration method for charm jet identification validated with proton-proton collision events at  $\sqrt{s} = 13$  TeV”, *JINST* **17** (2022) P03014, [doi:10.1088/1748-0221/17/03/P03014](https://doi.org/10.1088/1748-0221/17/03/P03014), [arXiv:2111.03027](https://arxiv.org/abs/2111.03027).
- [84] CMS Collaboration, “B-tagging performance of the CMS Legacy dataset 2018.”, CMS Detector Performance Note CMS-DP-2021-004, 2021.
- [85] CMS Collaboration, “Performance summary of AK4 jet b tagging with data from proton-proton collisions at 13 TeV with the CMS detector”, CMS Detector Performance Note CMS-DP-2023-005, 2023.
- [86] CMS Collaboration, “Performance summary of AK4 jet charm tagging with the CMS Run2 Legacy dataset”, CMS Detector Performance Note CMS-DP-2023-006, 2023.
- [87] CMS Collaboration, “A first look at early 2022 proton-proton collisions at  $\sqrt{s} = 13.6$  TeV for heavy-flavor jet tagging”, CMS Detector Performance Note CMS-DP-2023-012, 2023.
- [88] ATLAS Collaboration, “Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS”, ATLAS Note ATL-PHYS-PUB-2020-014, CERN, Geneva, 2020.
- [89] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Energy Flow Networks: Deep Sets for Particle Jets”, *JHEP* **01** (2019) 121, [doi:10.1007/JHEP01\(2019\)121](https://doi.org/10.1007/JHEP01(2019)121), [arXiv:1810.05165](https://arxiv.org/abs/1810.05165).
- [90] ATLAS Collaboration, “Neural Network Jet Flavour Tagging with the Upgraded ATLAS Inner Tracker Detector at the High-Luminosity LHC”, ATLAS Note ATL-PHYS-PUB-2022-047, CERN, Geneva, 2022.
- [91] ATLAS Collaboration, “Jet Flavour Tagging With GN1 and DL1d. Generator dependence, Run 2 and Run 3 data agreement studies”, ATLAS Plots FTAG-2023-01.

- [92] ATLAS Collaboration, “Graph Neural Network Jet Flavour Tagging with the ATLAS Detector”, ATLAS Note ATL-PHYS-PUB-2022-027, 2022.
- [93] J. Shlomi et al., “Secondary vertex finding in jets with neural networks”, *Eur. Phys. J. C* **81** (2021), no. 6, 540, doi:10.1140/epjc/s10052-021-09342-y, arXiv:2008.02831.
- [94] D. Hwang et al., “Self-supervised Auxiliary Learning with Meta-paths for Heterogeneous Graphs”, *arXiv e-prints* (July, 2020) arXiv:2007.08294, doi:10.48550/arXiv.2007.08294, arXiv:2007.08294.
- [95] H. Serviansky et al., “Set2Graph: Learning Graphs From Sets”, *arXiv e-prints* (February, 2020) arXiv:2002.08772, doi:10.48550/arXiv.2002.08772, arXiv:2002.08772.
- [96] S. Brody, U. Alon, and E. Yahav, “How Attentive are Graph Attention Networks?”, *arXiv e-prints* (May, 2021) arXiv:2105.14491, doi:10.48550/arXiv.2105.14491, arXiv:2105.14491.
- [97] H. Qu, C. Li, and S. Qian, “Particle Transformer for jet tagging”, in *Proceedings of the 39th International Conference on Machine Learning*, pp. 18281–18292. 2022. arXiv:2202.03772.
- [98] CMS Collaboration, “Transformer models for heavy flavor jet identification”, CMS Detector Performance Note CMS-DP-2022-050, 2022.
- [99] CMS Tracker Group Collaboration, “The CMS Phase-1 Pixel Detector Upgrade”, *JINST* **16** (2021) P02027, doi:10.1088/1748-0221/16/02/P02027, arXiv:2012.14304.
- [100] 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*. 6, 2023. arXiv:2306.04415.
- [101] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay”, *arXiv e-prints* (2018) arXiv:1803.09820, doi:10.48550/arXiv.1803.09820, arXiv:1803.09820.
- [102] Y. L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber, “Better jet clustering algorithms”, *JHEP* **08** (1997) 001, doi:10.1088/1126-6708/1997/08/001, arXiv:hep-ph/9707323.
- [103] CMS Collaboration, “Performance of b tagging at sqrt(s)=8 TeV in multijet, ttbar and boosted topology events”, CMS Physics Analysis Summary CMS-PAS-BTV-13-001, CERN, Geneva, 2013.
- [104] CMS Collaboration, “Performance of b tagging in boosted topology events”, CMS Detector Performance Note CMS-DP-2014-031, 2014.
- [105] CMS Collaboration, “Identification of double-b quark jets in boosted event topologies”, CMS Physics Analysis Summary CMS-PAS-BTV-15-002, CERN, Geneva, 2016.
- [106] J. Thaler and K. Van Tilburg, “Identifying Boosted Objects with N-subjettiness”, *JHEP* **03** (2011) 015, doi:10.1007/JHEP03(2011)015, arXiv:1011.2268.
- [107] J. Thaler and K. Van Tilburg, “Maximizing Boosted Top Identification by Minimizing N-subjettiness”, *JHEP* **02** (2012) 093, doi:10.1007/JHEP02(2012)093, arXiv:1108.2701.
- [108] ATLAS Collaboration, “Search for Higgs boson pair production in the  $b\bar{b}b\bar{b}$  final state from pp collisions at  $\sqrt{s} = 8$  TeV with

- the ATLAS detector”, *Eur. Phys. J. C* **75** (2015), no. 9, 412,  
doi:10.1140/epjc/s10052-015-3628-x,  
arXiv:1506.00285.
- [109] ATLAS Collaboration, “Expected Performance of Boosted Higgs ( $\rightarrow b\bar{b}$ ) Boson Identification with the ATLAS Detector at  $\sqrt{s} = 13$  TeV”, ATLAS Note ATL-PHYS-PUB-2015-035, CERN, Geneva, 2015.
- [110] ATLAS Collaboration, “Boosted Higgs ( $\rightarrow b\bar{b}$ ) Boson Identification with the ATLAS Detector at  $\sqrt{s} = 13$  TeV”, ATLAS Note ATLAS-CONF-2016-039, CERN, Geneva, 2016.
- [111] ATLAS Collaboration, “Identification of boosted Higgs bosons decaying into  $b$ -quark pairs with the ATLAS detector at 13 TeV”, *Eur. Phys. J. C* **79** (2019) 836,  
doi:10.1140/epjc/s10052-019-7335-x,  
arXiv:1906.11005.
- [112] “Flavor Tagging with Track Jets in Boosted Topologies with the ATLAS Detector”, ATLAS Note ATL-PHYS-PUB-2014-013, CERN, Geneva, 2014.
- [113] ATLAS Collaboration, “Identification of Boosted Higgs Bosons Decaying Into  $b\bar{b}$  With Neural Networks and Variable Radius Subjets in ATLAS”, ATLAS Note ATL-PHYS-PUB-2020-019, CERN, Geneva, 2020.
- [114] D. Krohn, J. Thaler, and L.-T. Wang, “Jets with variable R”, *Journal of High Energy Physics* (2009) 059,  
doi:10.1088/1126-6708/2009/06/059.
- [115] ATLAS Collaboration, “Variable Radius, Exclusive- $k_T$ , and Center-of-Mass Subjet Reconstruction for Higgs( $\rightarrow b\bar{b}$ ) Tagging in ATLAS”, ATLAS Note ATL-PHYS-PUB-2017-010, CERN, Geneva, 2017.
- [116] ATLAS Collaboration, “Performance of 2019 recommendations of atlas flavor tagging algorithms with variable radius track jets”, 2019,  
https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/FTAG-2019-006/.
- [117] ATLAS Collaboration, “Efficiency corrections for a tagger for boosted  $H \rightarrow b\bar{b}$  decays in  $pp$  collisions at  $\sqrt{s} = 13$  TeV with the ATLAS detector”, ATLAS Note ATL-PHYS-PUB-2021-035, CERN, Geneva, 2021.
- [118] CMS Collaboration, “Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques”, *JINST* **15** (2020) P06005,  
doi:10.1088/1748-0221/15/06/P06005,  
arXiv:2004.08262.
- [119] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, *arXiv e-prints* (December, 2015) arXiv:1512.03385,  
doi:10.48550/arXiv.1512.03385,  
arXiv:1512.03385.
- [120] CMS Collaboration, “Performance of Deep Tagging Algorithms for Boosted Double Quark Jet Topology in Proton-Proton Collisions at 13 TeV with the Phase-0 CMS Detector”, CMS Detector Performance Note CMS-DP-2018-046, 2018.
- [121] CMS Collaboration, “Performance of the mass-decorrelated DeepDoubleX classifier for double- $b$  and double- $c$  large-radius jets with the CMS detector”, CMS Detector Performance Note CMS-DP-2022-041, 2022.
- [122] ATLAS Collaboration, “DeXTer: Deep Sets based Neural Networks for Low- $p_T$   $X \rightarrow b\bar{b}$  Identification in ATLAS”, ATLAS Note ATL-PHYS-PUB-2022-042, CERN, Geneva, 2022.
- [123] D. Curtin et al., “Exotic decays of the 125 GeV Higgs boson”, *Phys. Rev. D* **90** (2014) 075004,  
doi:10.1103/PhysRevD.90.075004,  
arXiv:1312.4992.

- [124] M. Casolino et al., “Probing a light CP-odd scalar in di-top-associated production at the LHC”, *Eur. Phys. J. C* **75** (2015) 498, [doi:10.1140/epjc/s10052-015-3708-y](https://doi.org/10.1140/epjc/s10052-015-3708-y), [arXiv:1507.07004](https://arxiv.org/abs/1507.07004).
- [125] M. Cepeda, S. Gori, V. M. Outchoorn, and J. Shelton, “Exotic Higgs Decays”, [doi:10.1146/annurev-nucl-102319-024147](https://doi.org/10.1146/annurev-nucl-102319-024147), [arXiv:2111.12751](https://arxiv.org/abs/2111.12751).
- [126] ATLAS Collaboration, “Jet reconstruction and performance using particle flow with the ATLAS Detector”, *Eur. Phys. J. C* **77** (2017) 466, [doi:10.1140/epjc/s10052-017-5031-2](https://doi.org/10.1140/epjc/s10052-017-5031-2), [arXiv:1703.10485](https://arxiv.org/abs/1703.10485).
- [127] ATLAS Collaboration, “Search for Higgs boson decays into two new low-mass spin-0 particles in the  $4b$  channel with the ATLAS detector using  $pp$  collisions at  $\sqrt{s} = 13$  TeV”, *Phys. Rev. D* **102** (2020) 112006, [doi:10.1103/PhysRevD.102.112006](https://doi.org/10.1103/PhysRevD.102.112006), [arXiv:2005.12236](https://arxiv.org/abs/2005.12236).
- [128] H. Qu and L. Gouskos, “ParticleNet: Jet Tagging via Particle Clouds”, *Phys. Rev. D* **101** (2020) 056019, [doi:10.1103/PhysRevD.101.056019](https://doi.org/10.1103/PhysRevD.101.056019), [arXiv:1902.08570](https://arxiv.org/abs/1902.08570).
- [129] CMS Collaboration, “Calibration of the mass-decorrelated ParticleNet tagger for boosted  $b\bar{b}$  and  $c\bar{c}$  jets using LHC Run 2 data”, CMS Detector Performance Note CMS-DP-2022-005, 2022.
- [130] CMS Collaboration, “A search for the standard model Higgs boson decaying to charm quarks”, *JHEP* **03** (2020) 131, [doi:10.1007/JHEP03\(2020\)131](https://doi.org/10.1007/JHEP03(2020)131), [arXiv:1912.01662](https://arxiv.org/abs/1912.01662).
- [131] CMS Collaboration, “Search for Higgs boson decay to a charm quark-antiquark pair in proton-proton collisions at  $\sqrt{s} = 13$  TeV”, *Phys. Rev. Lett.* **131** (2023) 061801, [doi:10.1103/PhysRevLett.131.061801](https://doi.org/10.1103/PhysRevLett.131.061801), [arXiv:2205.05550](https://arxiv.org/abs/2205.05550).
- [132] CMS Collaboration, “Performance of heavy-flavour jet identification in boosted topologies in proton-proton collisions at  $\sqrt{s} = 13$  TeV”, CMS Physics Analysis Summary CMS-PAS-BTV-22-001, CERN, Geneva, 2023.
- [133] ATLAS Collaboration, “Transformer Neural Networks for Identifying Boosted Higgs Bosons decaying into  $b\bar{b}$  and  $c\bar{c}$  in ATLAS”, ATLAS Note ATL-PHYS-PUB-2023-021, CERN, Geneva, 2023.
- [134] S. Shleifer, J. Weston, and M. Ott, “NormFormer: Improved Transformer Pretraining with Extra Normalization”, *arXiv e-prints* (October, 2021) [arXiv:2110.09456](https://arxiv.org/abs/2110.09456), [doi:10.48550/arXiv.2110.09456](https://doi.org/10.48550/arXiv.2110.09456), [arXiv:2110.09456](https://arxiv.org/abs/2110.09456).
- [135] ATLAS Collaboration, “Optimisation of large-radius jet reconstruction for the ATLAS detector in 13 TeV proton-proton collisions”, *Eur. Phys. J. C* **81** (2021) 334, [doi:10.1140/epjc/s10052-021-09054-3](https://doi.org/10.1140/epjc/s10052-021-09054-3), [arXiv:2009.04986](https://arxiv.org/abs/2009.04986).
- [136] Y. Cui, Z. Han, and M. D. Schwartz, “W-jet Tagging: Optimizing the Identification of Boosted Hadronically-Decaying W Bosons”, *Phys. Rev. D* **83** (2011) 074023, [doi:10.1103/PhysRevD.83.074023](https://doi.org/10.1103/PhysRevD.83.074023), [arXiv:1012.2077](https://arxiv.org/abs/1012.2077).
- [137] J. M. Butterworth, A. R. Davison, M. Rubin, and G. P. Salam, “Jet substructure as a new Higgs search channel at the LHC”, *Phys. Rev. Lett.* **100** (2008) 242001, [doi:10.1103/PhysRevLett.100.242001](https://doi.org/10.1103/PhysRevLett.100.242001), [arXiv:0802.2470](https://arxiv.org/abs/0802.2470).
- [138] T. Plehn, G. P. Salam, and M. Spannowsky, “Fat Jets for a Light

- Higgs”, *Phys. Rev. Lett.* **104** (2010) 111801,  
[doi:10.1103/PhysRevLett.104.111801](https://doi.org/10.1103/PhysRevLett.104.111801),  
[arXiv:0910.5472](https://arxiv.org/abs/0910.5472).
- [139] A. J. Larkoski, G. P. Salam, and J. Thaler, “Energy Correlation Functions for Jet Substructure”, *JHEP* **06** (2013) 108,  
[doi:10.1007/JHEP06\(2013\)108](https://doi.org/10.1007/JHEP06(2013)108),  
[arXiv:1305.0007](https://arxiv.org/abs/1305.0007).
- [140] I. Moult, L. Necib, and J. Thaler, “New Angles on Energy Correlation Functions”, *JHEP* **12** (2016) 153,  
[doi:10.1007/JHEP12\(2016\)153](https://doi.org/10.1007/JHEP12(2016)153),  
[arXiv:1609.07483](https://arxiv.org/abs/1609.07483).
- [141] D. E. Kaplan, K. Rehermann, M. D. Schwartz, and B. Tweedie, “Top Tagging: A Method for Identifying Boosted Hadronically Decaying Top Quarks”, *Phys. Rev. Lett.* **101** (2008) 142001,  
[doi:10.1103/PhysRevLett.101.142001](https://doi.org/10.1103/PhysRevLett.101.142001),  
[arXiv:0806.0848](https://arxiv.org/abs/0806.0848).
- [142] T. Plehn and M. Spannowsky, “Top Tagging”, *J. Phys. G* **39** (2012) 083001,  
[doi:10.1088/0954-3899/39/8/083001](https://doi.org/10.1088/0954-3899/39/8/083001),  
[arXiv:1112.4441](https://arxiv.org/abs/1112.4441).
- [143] T. Plehn, M. Spannowsky, and M. Takeuchi, “How to Improve Top Tagging”, *Phys. Rev. D* **85** (2012) 034029,  
[doi:10.1103/PhysRevD.85.034029](https://doi.org/10.1103/PhysRevD.85.034029),  
[arXiv:1111.5034](https://arxiv.org/abs/1111.5034).
- [144] D. E. Soper and M. Spannowsky, “Finding top quarks with shower deconstruction”, *Phys. Rev. D* **87** (2013) 054012,  
[doi:10.1103/PhysRevD.87.054012](https://doi.org/10.1103/PhysRevD.87.054012),  
[arXiv:1211.3140](https://arxiv.org/abs/1211.3140).
- [145] C. Anders et al., “Benchmarking an even better top tagger algorithm”, *Phys. Rev. D* **89** (2014) 074047,  
[doi:10.1103/PhysRevD.89.074047](https://doi.org/10.1103/PhysRevD.89.074047),  
[arXiv:1312.1504](https://arxiv.org/abs/1312.1504).
- [146] G. Kasieczka et al., “Resonance Searches with an Updated Top Tagger”, *JHEP* **06** (2015) 203,  
[doi:10.1007/JHEP06\(2015\)203](https://doi.org/10.1007/JHEP06(2015)203),  
[arXiv:1503.05921](https://arxiv.org/abs/1503.05921).
- [147] J. Gallicchio and M. D. Schwartz, “Quark and Gluon Tagging at the LHC”, *Phys. Rev. Lett.* **107** (2011) 172001,  
[doi:10.1103/PhysRevLett.107.172001](https://doi.org/10.1103/PhysRevLett.107.172001),  
[arXiv:1106.3076](https://arxiv.org/abs/1106.3076).
- [148] J. Gallicchio and M. D. Schwartz, “Quark and Gluon Jet Substructure”, *JHEP* **04** (2013) 090,  
[doi:10.1007/JHEP04\(2013\)090](https://doi.org/10.1007/JHEP04(2013)090),  
[arXiv:1211.7038](https://arxiv.org/abs/1211.7038).
- [149] A. J. Larkoski, J. Thaler, and W. J. Waalewijn, “Gaining (Mutual) Information about Quark/Gluon Discrimination”, *JHEP* **11** (2014) 129,  
[doi:10.1007/JHEP11\(2014\)129](https://doi.org/10.1007/JHEP11(2014)129),  
[arXiv:1408.3122](https://arxiv.org/abs/1408.3122).
- [150] B. Bhattacharjee et al., “Associated jet and subjet rates in light-quark and gluon jet discrimination”, *JHEP* **04** (2015) 131,  
[doi:10.1007/JHEP04\(2015\)131](https://doi.org/10.1007/JHEP04(2015)131),  
[arXiv:1501.04794](https://arxiv.org/abs/1501.04794).
- [151] D. Ferreira de Lima, P. Petrov, D. Soper, and M. Spannowsky, “Quark-Gluon tagging with Shower Deconstruction: Unearthing dark matter and Higgs couplings”, *Phys. Rev. D* **95** (2017) 034001,  
[doi:10.1103/PhysRevD.95.034001](https://doi.org/10.1103/PhysRevD.95.034001),  
[arXiv:1607.06031](https://arxiv.org/abs/1607.06031).
- [152] C. Frye, A. J. Larkoski, J. Thaler, and K. Zhou, “Casimir Meets Poisson: Improved Quark/Gluon Discrimination with Counting Observables”, *JHEP* **09** (2017) 083,  
[doi:10.1007/JHEP09\(2017\)083](https://doi.org/10.1007/JHEP09(2017)083),  
[arXiv:1704.06266](https://arxiv.org/abs/1704.06266).
- [153] J. Cogan, M. Kagan, E. Strauss, and A. Schwartzman, “Jet-Images: Computer Vision Inspired Techniques for Jet Tagging”, *JHEP* **02** (2015) 118,  
[doi:10.1007/JHEP02\(2015\)118](https://doi.org/10.1007/JHEP02(2015)118),  
[arXiv:1407.5675](https://arxiv.org/abs/1407.5675).



- [154] L. G. Almeida et al., “Playing Tag with ANN: Boosted Top Identification with Pattern Recognition”, *JHEP* **07** (2015) 086, doi:10.1007/JHEP07(2015)086, arXiv:1501.05968.
- [155] L. de Oliveira et al., “Jet-images — deep learning edition”, *JHEP* **07** (2016) 069, doi:10.1007/JHEP07(2016)069, arXiv:1511.05190.
- [156] P. Baldi et al., “Jet Substructure Classification in High-Energy Physics with Deep Neural Networks”, *Phys. Rev. D* **93** (2016) 094034, doi:10.1103/PhysRevD.93.094034, arXiv:1603.09349.
- [157] A. Schwartzman et al., “Image Processing, Computer Vision, and Deep Learning: new approaches to the analysis and physics interpretation of LHC events”, *J. Phys. Conf. Ser.* **762** (2016) 012035, doi:10.1088/1742-6596/762/1/012035.
- [158] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, “Deep learning in color: towards automated quark/gluon jet discrimination”, *JHEP* **01** (2017) 110, doi:10.1007/JHEP01(2017)110, arXiv:1612.01551.
- [159] ATLAS Collaboration, “Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector”, ATLAS Note ATL-PHYS-PUB-2017-017, 2017.
- [160] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, “Deep-learning Top Taggers or The End of QCD?”, *JHEP* **05** (2017) 006, doi:10.1007/JHEP05(2017)006, arXiv:1701.08784.
- [161] S. Macaluso and D. Shih, “Pulling Out All the Tops with Computer Vision and Deep Learning”, *JHEP* **10** (2018) 121, doi:10.1007/JHEP10(2018)121, arXiv:1803.00107.
- [162] M. Andrews et al., “End-to-end jet classification of quarks and gluons with the CMS Open Data”, *Nucl. Instrum. Meth. A* **977** (2020) 164304, doi:10.1016/j.nima.2020.164304, arXiv:1902.08276.
- [163] S. Diefenbacher et al., “CapsNets Continuing the Convolutional Quest”, *SciPost Phys.* **8** (2020) 023, doi:10.21468/SciPostPhys.8.2.023, arXiv:1906.11265.
- [164] S. Bhattacharya, M. Guchait, and A. H. Vijay, “Boosted top quark tagging and polarization measurement using machine learning”, *Phys. Rev. D* **105** (2022) 042005, doi:10.1103/PhysRevD.105.042005, arXiv:2010.11778.
- [165] M. Andrews et al., “End-to-end jet classification of boosted top quarks with the CMS open data”, *EPJ Web Conf.* **251** (2021) 04030, doi:10.1051/epjconf/202125104030, arXiv:2104.14659.
- [166] D. Guest et al., “Jet Flavor Classification in High-Energy Physics with Deep Neural Networks”, *Phys. Rev. D* **94** (2016) 112002, doi:10.1103/PhysRevD.94.112002, arXiv:1607.08633.
- [167] J. Pearkes, W. Fedorko, A. Lister, and C. Gay, “Jet Constituents for Deep Neural Network Based Top Quark Tagging”, arXiv:1704.02124.
- [168] S. Egan et al., “Long Short-Term Memory (LSTM) networks with jet constituents for boosted top tagging at the LHC”, arXiv:1711.09059.
- [169] K. Fraser and M. D. Schwartz, “Jet Charge and Machine Learning”, *JHEP* **10** (2018) 093, doi:10.1007/JHEP10(2018)093, arXiv:1803.08066.
- [170] CMS Collaboration, “Boosted jet identification using particle candidates and deep neural networks”, CMS Detector Performance Note CMS-DP-2017-049, 2017.

- [171] A. Butter, G. Kasieczka, T. Plehn, and M. Russell, “Deep-learned Top Tagging with a Lorentz Layer”, *SciPost Phys.* **5** (2018) 028, [doi:10.21468/SciPostPhys.5.3.028](https://doi.org/10.21468/SciPostPhys.5.3.028), [arXiv:1707.08966](https://arxiv.org/abs/1707.08966).
- [172] G. Louppe, K. Cho, C. Becot, and K. Cranmer, “QCD-Aware Recursive Neural Networks for Jet Physics”, *JHEP* **01** (2019) 057, [doi:10.1007/JHEP01\(2019\)057](https://doi.org/10.1007/JHEP01(2019)057), [arXiv:1702.00748](https://arxiv.org/abs/1702.00748).
- [173] CMS Collaboration, “Machine learning-based identification of highly Lorentz-boosted hadronically decaying particles at the CMS experiment”, CMS Physics Analysis Summary CMS-PAS-JME-18-002, CERN, Geneva, 2019.
- [174] G. Kasieczka, N. Kiefer, T. Plehn, and J. M. Thompson, “Quark-Gluon Tagging: Machine Learning vs Detector”, *SciPost Phys.* **6** (2019) 069, [doi:10.21468/SciPostPhys.6.6.069](https://doi.org/10.21468/SciPostPhys.6.6.069), [arXiv:1812.09223](https://arxiv.org/abs/1812.09223).
- [175] M. Erdmann, E. Geiser, Y. Rath, and M. Rieger, “Lorentz Boost Networks: Autonomous Physics-Inspired Feature Engineering”, *JINST* **14** (2019) P06006, [doi:10.1088/1748-0221/14/06/P06006](https://doi.org/10.1088/1748-0221/14/06/P06006), [arXiv:1812.09722](https://arxiv.org/abs/1812.09722).
- [176] T. Cheng, “Recursive Neural Networks in Quark/Gluon Tagging”, *Comput. Softw. Big Sci.* **2** (2018) 3, [doi:10.1007/s41781-018-0007-y](https://doi.org/10.1007/s41781-018-0007-y), [arXiv:1711.02633](https://arxiv.org/abs/1711.02633).
- [177] M. J. Dolan and A. Ore, “Equivariant Energy Flow Networks for Jet Tagging”, *Phys. Rev. D* **103** (2021) 074022, [doi:10.1103/PhysRevD.103.074022](https://doi.org/10.1103/PhysRevD.103.074022), [arXiv:2012.00964](https://arxiv.org/abs/2012.00964).
- [178] E. A. Moreno et al., “JEDI-net: a jet identification algorithm based on interaction networks”, *Eur. Phys. J. C* **80** (2020) 58, [doi:10.1140/epjc/s10052-020-7608-4](https://doi.org/10.1140/epjc/s10052-020-7608-4), [arXiv:1908.05318](https://arxiv.org/abs/1908.05318).
- [179] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe, and G. Rochette, “Neural Message Passing for Jet Physics”, *Deep Learning for Physical Sciences Workshop at the 31st Conference on Neural Information Processing Systems (NIPS)* (2017).
- [180] M. Abdughani, J. Ren, L. Wu, and J. M. Yang, “Probing stop pair production at the LHC with graph neural networks”, *JHEP* **08** (2019) 055, [doi:10.1007/JHEP08\(2019\)055](https://doi.org/10.1007/JHEP08(2019)055), [arXiv:1807.09088](https://arxiv.org/abs/1807.09088).
- [181] T. S. Roy and A. H. Vijay, “A robust anomaly finder based on autoencoders”, [arXiv:1903.02032](https://arxiv.org/abs/1903.02032).
- [182] J. Ren, L. Wu, and J. M. Yang, “Unveiling CP property of top-Higgs coupling with graph neural networks at the LHC”, *Phys. Lett. B* **802** (2020) 135198, [doi:10.1016/j.physletb.2020.135198](https://doi.org/10.1016/j.physletb.2020.135198), [arXiv:1901.05627](https://arxiv.org/abs/1901.05627).
- [183] V. Mikuni and F. Canelli, “ABCNet: An attention-based method for particle tagging”, *Eur. Phys. J. Plus* **135** (2020) 463, [doi:10.1140/epjp/s13360-020-00497-3](https://doi.org/10.1140/epjp/s13360-020-00497-3), [arXiv:2001.05311](https://arxiv.org/abs/2001.05311).
- [184] A. Chakraborty, S. H. Lim, M. M. Nojiri, and M. Takeuchi, “Neural Network-based Top Tagger with Two-Point Energy Correlations and Geometry of Soft Emissions”, *JHEP* **07** (2020) 111, [doi:10.1007/JHEP07\(2020\)111](https://doi.org/10.1007/JHEP07(2020)111), [arXiv:2003.11787](https://arxiv.org/abs/2003.11787).
- [185] F. A. Dreyer and H. Qu, “Jet tagging in the Lund plane with graph networks”, *JHEP* **03** (2021) 052, [doi:10.1007/JHEP03\(2021\)052](https://doi.org/10.1007/JHEP03(2021)052), [arXiv:2012.08526](https://arxiv.org/abs/2012.08526).

- [186] O. Atkinson et al., “Anomaly detection with convolutional Graph Neural Networks”, *JHEP* **08** (2021) 080, [doi:10.1007/JHEP08\(2021\)080](https://doi.org/10.1007/JHEP08(2021)080), [arXiv:2105.07988](https://arxiv.org/abs/2105.07988).
- [187] O. Atkinson et al., “IRC-Safe Graph Autoencoder for Unsupervised Anomaly Detection”, *Front. Artif. Intell.* **5** (2022) 943135, [doi:10.3389/frai.2022.943135](https://doi.org/10.3389/frai.2022.943135), [arXiv:2204.12231](https://arxiv.org/abs/2204.12231).
- [188] F. A. Dreyer, G. Soyez, and A. Takacs, “Quarks and gluons in the Lund plane”, *JHEP* **08** (2022) 177, [doi:10.1007/JHEP08\(2022\)177](https://doi.org/10.1007/JHEP08(2022)177), [arXiv:2112.09140](https://arxiv.org/abs/2112.09140).
- [189] C. Shimmin, “Particle Convolution for High Energy Physics”, 7, 2021. [arXiv:2107.02908](https://arxiv.org/abs/2107.02908).
- [190] F. A. Dreyer, R. Grabarczyk, and P. F. Monni, “Leveraging universality of jet taggers through transfer learning”, *Eur. Phys. J. C* **82** (2022) 564, [doi:10.1140/epjc/s10052-022-10469-9](https://doi.org/10.1140/epjc/s10052-022-10469-9), [arXiv:2203.06210](https://arxiv.org/abs/2203.06210).
- [191] C. Li et al., “Does Lorentz-symmetric design boost network performance in jet physics?”, [arXiv:2208.07814](https://arxiv.org/abs/2208.07814).
- [192] S. Gong et al., “An efficient Lorentz equivariant graph neural network for jet tagging”, *JHEP* **07** (2022) 030, [doi:10.1007/JHEP07\(2022\)030](https://doi.org/10.1007/JHEP07(2022)030), [arXiv:2201.08187](https://arxiv.org/abs/2201.08187).
- [193] F. Ma, F. Liu, and W. Li, “A jet tagging algorithm of graph network with HaarPooling message passing”, [arXiv:2210.13869](https://arxiv.org/abs/2210.13869).
- [194] P. Vincent et al., “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”, *Journal of Machine Learning Research* **11** (2010) 3371–3408.
- [195] R. Kogler et al., “Jet Substructure at the Large Hadron Collider: Experimental Review”, *Rev. Mod. Phys.* **91** (2019) 045003, [doi:10.1103/RevModPhys.91.045003](https://doi.org/10.1103/RevModPhys.91.045003), [arXiv:1803.06991](https://arxiv.org/abs/1803.06991).
- [196] A. Butter et al., “The Machine Learning landscape of top taggers”, *SciPost Phys.* **7** (2019) 014, [doi:10.21468/SciPostPhys.7.1.014](https://doi.org/10.21468/SciPostPhys.7.1.014), [arXiv:1902.09914](https://arxiv.org/abs/1902.09914).
- [197] A. J. Larkoski, I. Moutl, and B. Nachman, “Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning”, *Phys. Rept.* **841** (2020) 1–63, [doi:10.1016/j.physrep.2019.11.001](https://doi.org/10.1016/j.physrep.2019.11.001), [arXiv:1709.04464](https://arxiv.org/abs/1709.04464).
- [198] L. Moore, K. Nordström, S. Varma, and M. Fairbairn, “Reports of My Demise Are Greatly Exaggerated:  $N$ -subjettiness Taggers Take On Jet Images”, *SciPost Phys.* **7** (2019) 036, [doi:10.21468/SciPostPhys.7.3.036](https://doi.org/10.21468/SciPostPhys.7.3.036), [arXiv:1807.04769](https://arxiv.org/abs/1807.04769).
- [199] A. Romero et al., “Safety of Quark/Gluon Jet Classification”, [arXiv:2103.09103](https://arxiv.org/abs/2103.09103).
- [200] A. Khot, M. S. Neubauer, and A. Roy, “A Detailed Study of Interpretability of Deep Neural Network based Top Taggers”, [arXiv:2210.04371](https://arxiv.org/abs/2210.04371).
- [201] I. W. Stewart et al., “XCone:  $N$ -jettiness as an Exclusive Cone Jet Algorithm”, *JHEP* **11** (2015) 072, [doi:10.1007/JHEP11\(2015\)072](https://doi.org/10.1007/JHEP11(2015)072), [arXiv:1508.01516](https://arxiv.org/abs/1508.01516).
- [202] J. Thaler and T. F. Wilkason, “Resolving Boosted Jets with XCone”, *JHEP* **12** (2015) 051, [doi:10.1007/JHEP12\(2015\)051](https://doi.org/10.1007/JHEP12(2015)051), [arXiv:1508.01518](https://arxiv.org/abs/1508.01518).
- [203] T. Lapsien, R. Kogler, and J. Haller, “A new tagger for hadronically decaying heavy particles at the LHC”, *Eur. Phys. J. C*

- 76** (2016) 600,  
[doi:10.1140/epjc/s10052-016-4443-8](https://doi.org/10.1140/epjc/s10052-016-4443-8),  
[arXiv:1606.04961](https://arxiv.org/abs/1606.04961).
- [204] B. Mukhopadhyaya, T. Samui, and R. K. Singh, “Dynamic radius jet clustering algorithm”, *JHEP* **04** (2023) 019,  
[doi:10.1007/JHEP04\(2023\)019](https://doi.org/10.1007/JHEP04(2023)019),  
[arXiv:2301.13074](https://arxiv.org/abs/2301.13074).
- [205] A. J. Larkoski, D. Rathjens, J. Veatch, and J. W. Walker, “Jet SIFT-ing: a new scale-invariant jet clustering algorithm for the substructure era”, [arXiv:2302.08609](https://arxiv.org/abs/2302.08609).
- [206] X. Ju and B. Nachman, “Supervised Jet Clustering with Graph Neural Networks for Lorentz Boosted Bosons”, *Phys. Rev. D* **102** (2020) 075014,  
[doi:10.1103/PhysRevD.102.075014](https://doi.org/10.1103/PhysRevD.102.075014),  
[arXiv:2008.06064](https://arxiv.org/abs/2008.06064).
- [207] S. Mondal, G. Barone, and A. Schmidt, “PAIReD jet: A multi-pronged resonance tagging strategy across all Lorentz boosts”, [arXiv:2311.11011](https://arxiv.org/abs/2311.11011).
- [208] B. Nachman and C. Shimmin, “AI Safety for High Energy Physics”, [arXiv:1910.08606](https://arxiv.org/abs/1910.08606).
- [209] D. Yallup and W. Handley, “Hunting for bumps in the margins”, *JINST* **18** (2023) P05014,  
[doi:10.1088/1748-0221/18/05/P05014](https://doi.org/10.1088/1748-0221/18/05/P05014),  
[arXiv:2211.10391](https://arxiv.org/abs/2211.10391).
- [210] A. Butter, B. M. Dillon, T. Plehn, and L. Vogel, “Performance versus resilience in modern quark-gluon tagging”, *SciPost Phys. Core* **6** (2023) 085, [doi:10.21468/SciPostPhysCore.6.4.085](https://doi.org/10.21468/SciPostPhysCore.6.4.085),  
[arXiv:2212.10493](https://arxiv.org/abs/2212.10493).
- [211] E. M. Metodiev, B. Nachman, and J. Thaler, “Classification without labels: Learning from mixed samples in high energy physics”, *JHEP* **10** (2017) 174,  
[doi:10.1007/JHEP10\(2017\)174](https://doi.org/10.1007/JHEP10(2017)174),  
[arXiv:1708.02949](https://arxiv.org/abs/1708.02949).
- [212] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, “JUNIPR: a Framework for Unsupervised Machine Learning in Particle Physics”, *Eur. Phys. J. C* **79** (2019) 102,  
[doi:10.1140/epjc/s10052-019-6607-9](https://doi.org/10.1140/epjc/s10052-019-6607-9),  
[arXiv:1804.09720](https://arxiv.org/abs/1804.09720).
- [213] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, “Learning to classify from impure samples with high-dimensional data”, *Phys. Rev. D* **98** (2018) 011502,  
[doi:10.1103/PhysRevD.98.011502](https://doi.org/10.1103/PhysRevD.98.011502),  
[arXiv:1801.10158](https://arxiv.org/abs/1801.10158).
- [214] E. Alvarez et al., “Exploring unsupervised top tagging using Bayesian inference”, *SciPost Phys. Core* **6** (2023) 046, [doi:10.21468/SciPostPhysCore.6.2.046](https://doi.org/10.21468/SciPostPhysCore.6.2.046),  
[arXiv:2212.13583](https://arxiv.org/abs/2212.13583).
- [215] A. Stein et al., “Improving Robustness of Jet Tagging Algorithms with Adversarial Training”, *Comput. Softw. Big Sci.* **6** (2022) 15,  
[doi:10.1007/s41781-022-00087-1](https://doi.org/10.1007/s41781-022-00087-1),  
[arXiv:2203.13890](https://arxiv.org/abs/2203.13890).
- [216] CMS Collaboration, “Adversarial training for b-tagging algorithms in CMS”, CMS Detector Performance Note CMS-DP-2022-049, 2022.
- [217] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Energy flow polynomials: A complete linear basis for jet substructure”, *JHEP* **04** (2018) 013,  
[doi:10.1007/JHEP04\(2018\)013](https://doi.org/10.1007/JHEP04(2018)013),  
[arXiv:1712.07124](https://arxiv.org/abs/1712.07124).
- [218] C. Grojean, A. Paul, and Z. Qian, “Resurrecting  $b\bar{b}h$  with kinematic shapes”, *JHEP* **04** (2021) 139,  
[doi:10.1007/JHEP04\(2021\)139](https://doi.org/10.1007/JHEP04(2021)139),  
[arXiv:2011.13945](https://arxiv.org/abs/2011.13945).
- [219] A. Bogatskiy et al., “Lorentz Group Equivariant Neural Network for Particle Physics”, [arXiv:2006.04780](https://arxiv.org/abs/2006.04780).
- [220] J. M. Munoz, I. Batatia, and C. Ortner, “Boost invariant polynomials for efficient

- jet tagging”, *Mach. Learn. Sci. Tech.* **3** (2022) 04LT05, [doi:10.1088/2632-2153/aca9ca](https://doi.org/10.1088/2632-2153/aca9ca), [arXiv:2207.08272](https://arxiv.org/abs/2207.08272).
- [221] O. Fedkevych, C. K. Khosa, S. Marzani, and F. Sforza, “Identification of b jets using QCD-inspired observables”, *Phys. Rev. D* **107** (2023) 034032, [doi:10.1103/PhysRevD.107.034032](https://doi.org/10.1103/PhysRevD.107.034032), [arXiv:2202.05082](https://arxiv.org/abs/2202.05082).
- [222] B. Bhattacharjee, C. Bose, A. Chakraborty, and R. Sengupta, “Boosted top tagging and its interpretation using Shapley values”, [arXiv:2212.11606](https://arxiv.org/abs/2212.11606).
- [223] L. Bradshaw, S. Chang, and B. Ostdiek, “Creating simple, interpretable anomaly detectors for new physics in jet substructure”, *Phys. Rev. D* **106** (2022) 035014, [doi:10.1103/PhysRevD.106.035014](https://doi.org/10.1103/PhysRevD.106.035014), [arXiv:2203.01343](https://arxiv.org/abs/2203.01343).
- [224] R. Das, G. Kasieczka, and D. Shih, “Feature Selection with Distance Correlation”, [arXiv:2212.00046](https://arxiv.org/abs/2212.00046).
- [225] P. Cal, J. Thaler, and W. J. Waalewijn, “Power counting energy flow polynomials”, *JHEP* **09** (2022) 021, [doi:10.1007/JHEP09\(2022\)021](https://doi.org/10.1007/JHEP09(2022)021), [arXiv:2205.06818](https://arxiv.org/abs/2205.06818).
- [226] ATLAS Collaboration, “Calibration of the b-tagging efficiency on charm jets using a sample of  $W+c$  events with  $\sqrt{s} = 13$  TeV ATLAS data”, ATLAS Note ATLAS-CONF-2018-055, CERN, Geneva, 2018.
- [227] ATLAS Collaboration, “Measurement of the b-jet identification efficiency for high transverse momentum jets in  $t\bar{t}$  events in the lepton + jets channel with the ATLAS detector using Run 2 data”, ATLAS Note ATLAS-PHYS-PUB-2021-004, CERN, Geneva, 2021.
- [228] ATLAS Collaboration, “Measurement of the c-jet mistagging efficiency in  $t\bar{t}$  events using pp collision data at  $\sqrt{s} = 13$  TeV collected with the ATLAS detector”, *Eur. Phys. J. C* **82** (2022), no. 1, 95, [doi:10.1140/epjc/s10052-021-09843-w](https://doi.org/10.1140/epjc/s10052-021-09843-w), [arXiv:2109.10627](https://arxiv.org/abs/2109.10627).
- [229] ATLAS Collaboration, “Measurement of the b-jet identification efficiency with the  $p_T^{\text{rel}}$  method in multi-jet events using pp collisions at  $\sqrt{s} = 13$  TeV with the ATLAS Detector”, ATLAS Note ATLAS-PHYS-PUB-2022-025, CERN, Geneva, 2022.
- [230] ATLAS Collaboration, “Calibration of the light-flavour jet mistagging efficiency of the b-tagging algorithms with Z+jets events using 139 fb $^{-1}$  of ATLAS proton–proton collision data at  $\sqrt{s} = 13$  TeV”, *Eur. Phys. J. C* **83** (2023) 728, [doi:10.1140/epjc/s10052-023-11736-z](https://doi.org/10.1140/epjc/s10052-023-11736-z), [arXiv:2301.06319](https://arxiv.org/abs/2301.06319).
- [231] I. Zurbano Fernandez et al., “High-Luminosity Large Hadron Collider (HL-LHC): Technical design report”, [doi:10.23731/CYRM-2020-0010](https://doi.org/10.23731/CYRM-2020-0010).
- [232] D. Bertolini, P. Harris, M. Low, and N. Tran, “Pileup Per Particle Identification”, *JHEP* **10** (2014) 059, [doi:10.1007/JHEP10\(2014\)059](https://doi.org/10.1007/JHEP10(2014)059), [arXiv:1407.6013](https://arxiv.org/abs/1407.6013).
- [233] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, “Pileup Mitigation with Machine Learning (PUMML)”, *JHEP* **12** (2017) 051, [doi:10.1007/JHEP12\(2017\)051](https://doi.org/10.1007/JHEP12(2017)051), [arXiv:1707.08600](https://arxiv.org/abs/1707.08600).
- [234] P. Hansen, J. W. Monk, and C. Wigglesworth, “A Wavelet Based Pile-Up Mitigation Method for the LHC Upgrade”, [arXiv:1812.07412](https://arxiv.org/abs/1812.07412).
- [235] J. Arjona Martínez et al., “Pileup mitigation at the Large Hadron Collider with graph neural networks”, *Eur. Phys. J. Plus* **134** (2019) 333, [doi:10.1140/epjp/i2019-12710-3](https://doi.org/10.1140/epjp/i2019-12710-3), [arXiv:1810.07988](https://arxiv.org/abs/1810.07988).

- [236] S. Alipour-Fard, P. T. Komiske, E. M. Metodiev, and J. Thaler, “Pileup and Infrared Radiation Annihilation (PIRANHA): a paradigm for continuous jet grooming”, *JHEP* **09** (2023) 157, [doi:10.1007/JHEP09\(2023\)157](https://doi.org/10.1007/JHEP09(2023)157), [arXiv:2305.00989](https://arxiv.org/abs/2305.00989).
- [237] ATLAS Collaboration, “Observation of  $H \rightarrow b\bar{b}$  decays and  $VH$  production with the ATLAS detector”, *Phys. Lett. B* **786** (2018) 59–86, [doi:10.1016/j.physletb.2018.09.013](https://doi.org/10.1016/j.physletb.2018.09.013), [arXiv:1808.08238](https://arxiv.org/abs/1808.08238).
- [238] CMS Collaboration, “Observation of Higgs boson decay to bottom quarks”, *Phys. Rev. Lett.* **121** (2018) 121801, [doi:10.1103/PhysRevLett.121.121801](https://doi.org/10.1103/PhysRevLett.121.121801), [arXiv:1808.08242](https://arxiv.org/abs/1808.08242).
- [239] CMS Collaboration, “Search for Higgs boson and observation of Z boson through their decay into a charm quark-antiquark pair in boosted topologies in proton-proton collisions at  $\sqrt{s} = 13$  TeV”, *Phys. Rev. Lett.* **131** (2023) 041801, [doi:10.1103/PhysRevLett.131.041801](https://doi.org/10.1103/PhysRevLett.131.041801), [arXiv:2211.14181](https://arxiv.org/abs/2211.14181).
- [240] ATLAS Collaboration, “Direct constraint on the Higgs-charm coupling from a search for Higgs boson decays into charm quarks with the ATLAS detector”, *Eur. Phys. J. C* **82** (2022) 717, [doi:10.1140/epjc/s10052-022-10588-3](https://doi.org/10.1140/epjc/s10052-022-10588-3), [arXiv:2201.11428](https://arxiv.org/abs/2201.11428).
- [241] ATLAS Collaboration, “Prospects for  $H \rightarrow c\bar{c}$  using Charm Tagging with the ATLAS Experiment at the HL-LHC”, ATLAS PUB Note ATL-PHYS-PUB-2018-016, 2018.
- [242] ATLAS Collaboration, “Extrapolation of ATLAS sensitivity to  $H \rightarrow b\bar{b}$  and  $H \rightarrow c\bar{c}$  decays in  $VH$  production at the HL-LHC”, ATLAS PUB Note ATL-PHYS-PUB-2021-039, 2021.