The Physics Behind ML-based Quark-Gluon Taggers

Sophia Vent[1,2], Ramon Winterhalder[3], and Tilman Plehn[1,4]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Dipartimento di Fisica e Astronomia, Universitá di Bologna, Italy
3 TIFLab, Università degli Studi di Milano & INFN Sezione di Milano, Italy
4 Interdisciplinary Center for Scientific Computing (IWR), Universität Heidelberg, Germany

December 23, 2025

Abstract

Jet taggers provide an ideal testbed for applying explainability techniques to powerful ML tools. For theoretically and experimentally challenging quark-gluon tagging, we first identify the leading latent features that correlate strongly with physics observables, both in a linear and a non-linear approach. Next, we show how Shapley values can assess feature importance, although the standard implementation assumes independent inputs and can lead to distorted attributions in the presence of correlations. Finally, we use symbolic regression to derive compact formulas to approximate the tagger output.

Contents

---

1 Introduction

Whenever we apply modern machine learning (ML) for fundamental physics [1], the same question arises: *Can we identify which physical observables and theoretical structures a trained neural network relies on?** Understanding the physical basis of the network's decisions gives us insights into their reasoning and strengthen confidence in their theoretical and experimental soundness. We demonstrate how this can be achieved using concepts from explainable AI (XAI), which involves probing the internal representations of trained, high-performance networks.

We take this XAI step by analyzing the inner workings of ParticleNet [2], a classic architecture for jet tagging. It represents modern ML tools operating on low-level detector inputs, such as jet constituent four-vectors or calorimeter images [3]. They have led to significant advances for jet tagging, including advanced transformer implementations [4–9]. Specific tasks include quark-gluon tagging [10–22], top tagging [23–27], W/Z tagging [28–33], and bottom/charm identification [34–37]. Complementing the classification performance, we look into the trained network and ask:

1. What features does the network rely on?
2. Do they align with known key observables?
3. Can formulas approximate the network?

As a testbed, we choose quark-gluon (QG) tagging [38–41]. While practically extremely promising for many LHC analyses, such as separating signal from background in weak boson fusion or mono-jet searches, QG tagging is theoretically and experimentally tricky. The question of whether a jet originates from a quark or gluon is ill-defined beyond leading order and sensitive to soft and collinear splittings. Furthermore, it strongly depends on the parton shower, hadronization, and detector effects [42, 43]. Very generally, gluon jets radiate more than quark jets because of the color charges, $C_F = 4/3 < C_A = 3$, so their increased particle multiplicity scales with the ratio of color factors, known as Casimir scaling [44, 45]. For additional discriminative power, we want to add more observables with different behavior.

The theoretical and experimental subtleties make QG tagging a particularly compelling case for XAI. Because it lacks a clear-cut ground truth and involves nuanced physics, interpretability is not just a bonus but a necessity. We ultimately envision applying such techniques to networks trained on data, where explainability can drive scientific discovery. Meanwhile, simulation-based studies like ours provide a controlled environment for developing and evaluating XAI tools. While not yet fully established, there is a growing number of physics applications employing promising XAI methods, including Shapley values [46–52], symbolic regression [53–61], and other techniques [62–68].

In this work, we study the internal representations learned by ParticleNet after training on quark–gluon tagging. Our goal is to investigate how the network uses and combines established physical observables, and whether it encodes additional observables that traditional high-level taggers typically do not exploit. First, we introduce the dataset and describe the ParticleNet architecture used for QG tagging in Sec. 2. In Sec. 3, we analyze the latent feature space of the network using linear and non-linear dimensionality reduction techniques and investigate how the learned features correlate and share mutual information with known jet observables. In Sec. 4, we perform a Shapley value analysis and discuss its benefits and limitations. Finally, in Sec. 5, we employ symbolic regression (SR) to derive an analytic expression in terms of the leading physical observables that approximates the decision boundary of the ML classifier.

*We refrain from using this question as the paper title because of Hinchliffe's rule.

2 Dataset and classifier network

Distinguishing quark-initiated from gluon-initiated jets is a long-standing challenge in LHC physics. It can enhance precision in Standard Model (SM) measurements and improve sensitivity in searches for Beyond Standard Model (BSM) physics, where signal and background processes often differ in jet flavor composition. While quark and gluon jets arise from massless QCD splittings, their internal structures differ because of the gluon's larger color charge. This results in higher particle multiplicities and broader radiation patterns. Beyond these qualitative properties, we investigate the performance of ML-based quark-gluon taggers using precision simulations.

Our primary dataset is generated using `Pythia` 8.2 [69–71] with default tunes and parton shower settings for the parton-level processes

$$q\bar{q} \to \mathrm{Z}(\to \nu\bar{\nu}) + \mathrm{g} \qquad \text{and} \qquad qg \to \mathrm{Z}(\to \nu\bar{\nu}) + (\mathrm{uds})\,. \tag{1}$$

As the neutrinos remain undetected, these processes provide a clean quark-gluon jet sample, allowing us to investigate any subtle differences in the jet substructure. In Fig. 1, we show some examples of LO Feynman diagrams for these processes.

Later in our analysis in Sec. 3, we compare results using a similar dataset generated with `Herwig` 7.1 [72, 73] to assess robustness across different generators. Each dataset consists of 2M jets, with up to 100 constituents per jet. We focus on light-flavor jets, and exclude events containing charm or bottom quarks. The jet reconstruction uses the anti-k_T algorithm [74] with $R = 0.4$, implemented in FastJet [75]. We select a subset of 600k jets with a training/-validation/test split of 400k/100k/100k, each with a 50:50 mixture of quark and gluon jets. For the the labeling of the two processes given in Eq.(1) we use

$$\text{jet label} = \begin{cases} 0 & \mathrm{Z}(\to \nu\bar{\nu}) + g \quad \text{(gluon-like)} \\ 1 & \mathrm{Z}(\to \nu\bar{\nu}) + (\mathrm{uds}) \quad \text{(quark-like)} \end{cases} \tag{2}$$

Our goal is not to define quark and gluon jets in a theoretically rigorous or generator-independent manner, but rather to analyze what structures a network learns when trained on a standard benchmark dataset. For this reason, we adopt the commonly used `Pythia` 8.2 quark–gluon dataset introduced above [70, 71], the references dataset for quark–gluon tagging studies.

While the labeling of quark and gluon jets is inherently ambiguous beyond leading order and subject to generator-specific modeling, this choice is sufficient for our purpose: to interpret the internal representations learned by the network under the same conditions used throughout the literature. Our results should therefore be understood as an analysis of what the network encodes given these standard labels, only approximately related to a proper definition of quark and gluon jets.

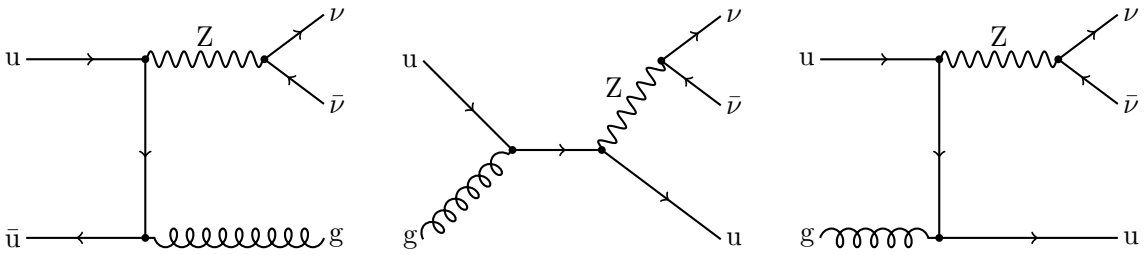


Figure 1: Examples of LO Feynman diagrams leading to gluon and quark jets.

Low-level classifier

Historically, quark-gluon taggers have relied on high-level observables motivated by QCD. Modern low-level ML taggers such as ParticleNet [2], operate on raw information about jet constituents, allowing the network to learn discriminative patterns without any bottleneck. This paradigm shift raises an important question: Are the features learned by such networks consistent with established high-level observables, or do they encode more intricate combinations of the jet constituents?

To address this, we examine the internal representations learned by ParticleNet. We aim to determine whether the network implicitly reconstructs established observables, identifies new combinations of known features, or encodes latent patterns that are difficult to interpret. In subsequent sections, we analyze the structure of the latent space, explore its correlation with physics-motivated observables, and investigate the minimal set of features necessary to preserve full classification performance.

ParticleNet

ParticleNet [2] is a graph convolutional network that processes unordered sets of jet constituents. Each constituent i is represented by a set of low-level features, such as angular distances from the jet axis, momentum, energy, and particle identification (PID). The full set of input features is given by

$$\left\{\Delta\eta_i, \Delta\phi_i, \Delta R_i,\ \log p_{T,i}, \log\frac{p_{T,i}}{p_{T,\text{jet}}},\ \log E_i, \log\frac{E_i}{E_{\text{jet}}},\ \text{PID}_i\right\}. \tag{3}$$

Following the ParticleNet convention, we only consider five different particle categories: electrons, muons, charged hadrons, neutral hadrons, and photons. The electric charge is included in the feature set, consistent with the original ParticleNet design, and we encode the PID by one-hot encoding.

The features in Eq.(3) are passed through a series of edge convolution (EdgeConv) layers. At each layer l, the network constructs a dynamic graph by connecting each particle i to its k-nearest neighbors $j \in \mathcal{N}(i)$ in the learned feature space. The per-particle feature vector $h_i^{(l)}$ is then updated using learned pairwise interactions:

$$h_i^{(l+1)} = \sum_{j\in\mathcal{N}(i)} f_\theta^{(l)}\left(h_i^{(l)}, h_j^{(l)} - h_i^{(l)}\right), \tag{4}$$

where $f_\theta^{(l)}$ denotes a sub-network at layer l with trainable parameters θ. This formulation allows the network to learn local patterns and update the particle features accordingly. At the end, per-particle features are aggregated using average pooling to produce a fixed-size jet representation, which is then passed through a multilayer perceptron (MLP) to output a binary classification probability.

Figure 2 illustrates the overall structure of the ParticleNet classifier alongside the inputs and outputs used in our explainability analysis. While the upper path corresponds to the standard inference pipeline described above, the lower path highlights how high-level observables derived from the point cloud can serve as inputs to methods such as symbolic regression or Shapley-based feature attribution. These techniques allow us to probe which physically motivated features the tagger may be implicitly relying on.

We use the compact ParticleNet-Lite variant. It utilizes a single, smaller edge convolution block and outputs a 64-dimensional pooled feature vector per jet. It simplifies the full ParticleNet architecture, which employs two edge convolution blocks and produces a 256-dimensional feature vector.

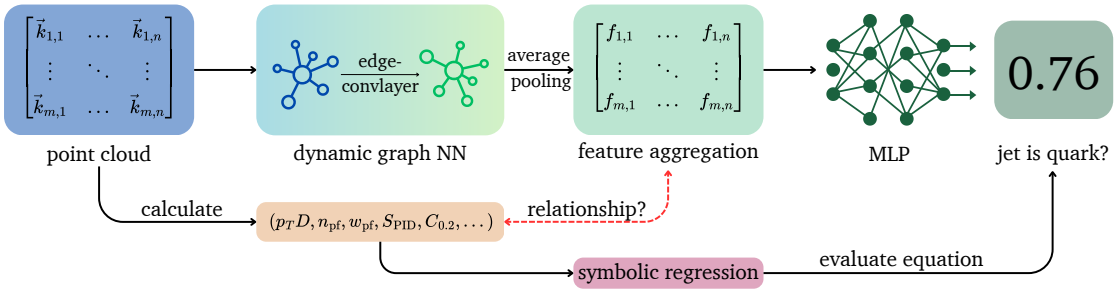


Figure 2: Overview of the ParticleNet architecture and its connection to the explainability techniques explored in our study.

3 From latent features to observables

ParticleNet-Lite learns a 64-dimensional representation of each jet, but it is not obvious that they are all needed for classification. Compressing the representation is the first step towards explainability. We extract the output of the average pooling layer from ParticleNet-Lite, a 64-dimensional vector summarizing each jet, and study its structure using a linear principal component analysis (PCA) and a latent representation from an autoencoder.

We consider a latent direction *interpretable* if it can be associated with a well-defined physical quantity (e.g. multiplicity, fragmentation, charge) rather than only with a complex and entangled mixture of observables. For images, a principal component (PC) might correspond to color or brightness of the jet images. We seek analogous, concise physics semantics and aim to show that the main discriminative directions can be described in these terms.

As a primary alignment score between a latent coordinate ℓ (e.g. a principal component (PC)) and a jet observable, we use the Pearson correlation coefficient ρ. Pearson is bounded within $[-1,1]$ and is scale-invariant, providing a direct and comparable measure of linear correlations across variables. For completeness, we also report mutual information as a non-linear check. However, because it is unbounded, its magnitude is less directly interpretable.

3.1 Linear dimensionality reduction

PCA [76] reduces the dimensionality of data by identifying directions that maximize variance. Let

$$X \in \mathbb{R}^{N\times d} \qquad \text{with} \qquad d = 64 \tag{5}$$

be the features of N jets. After zero-centering each feature, we compute the empirical covariance matrix

$$\Sigma = \frac{1}{N-1}(X-\mu_X)^\top (X-\mu_X)\,. \tag{6}$$

We then perform an eigen-decomposition

$$\Sigma = V\Sigma_0 V^\top\,, \tag{7}$$

where Σ_0 is diagonal, the eigenvalues are called explained variances, i.e. the leading principal components (PCs) capture directions of maximal variance in the latent space. The matrix V gives the principal directions as eigenvectors, so in the PC basis the jet data is given by

$$Z = (X-\mu)V\,. \tag{8}$$

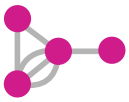
To evaluate the impact of the PCs for classification, we train a simple quark-gluon classifier on a set of leading k PCs and determine their AUCs. This tells us how much discriminative power can be retained in lower-dimensional representations. In Fig. 3, we see that the first five principal components are sufficient to recover the ParticleNet-Lite performance, AUC = 0.902. The leading three PCs already yield an AUC > 0.89.

To ensure resilience, we repeat the analysis using a `Herwig` dataset and show similar results also in Fig. 3. Even when the PCA transformation is learned on `Pythia` jets and applied to `Herwig` jets, the performance remains comparable. This suggests that the principal directions are relatively universal across generators. Altogether, the performance degrades when using `Herwig`, relative to `Pythia`, consistent with previous results [43].

To understand the compressed latent space learned by ParticleNet, we compare the leading PCs with standard substructure observables like the particle multiplicity n_{pf}, the first radial moment or girth w_{pf} [77, 78], the two-point energy correlation function C_β for $\beta = 0.2$ [79], and the width of the p_T-distribution of the constituents $p_T D$,

$$
\begin{aligned}
n_{\text{pf}} &= \sum_i 1 & w_{\text{pf}} &= \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{\sum_i p_{T,i}} \\
C_\beta &= \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2} & p_T D &= \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \,.
\end{aligned}
\tag{9}
$$

These observables are chosen as a minimal starting set, since they are commonly used in high-level jet substructure taggers. We then extend this set with additional standard jet observables, such as thrust and higher-order energy correlators, but only report those with the highest correlations. Furthermore, we consider a set of Energy Flow Polynomials (EFPs) [80], which form an infrared- and collinear-safe, complete linear basis of IRC-safe observables. EFPs summarize a jet by combining momentum fractions z_i with angular separations ΔR_{ij} in multigraph patterns. A given multigraph is evaluated by multiplying one z_i per vertex and one ΔR_{ij} per edge, summed over all jet constituents, for example



$$
= \sum_i \sum_j \sum_k \sum_l z_i z_j z_k z_l \Delta R_{ij} \Delta R_{ik} \Delta R_{jk}^3 \Delta R_{kl} \qquad \text{with} \qquad z_i = \frac{p_{T,i}}{\sum_i p_{T,i}} \,. \tag{10}
$$

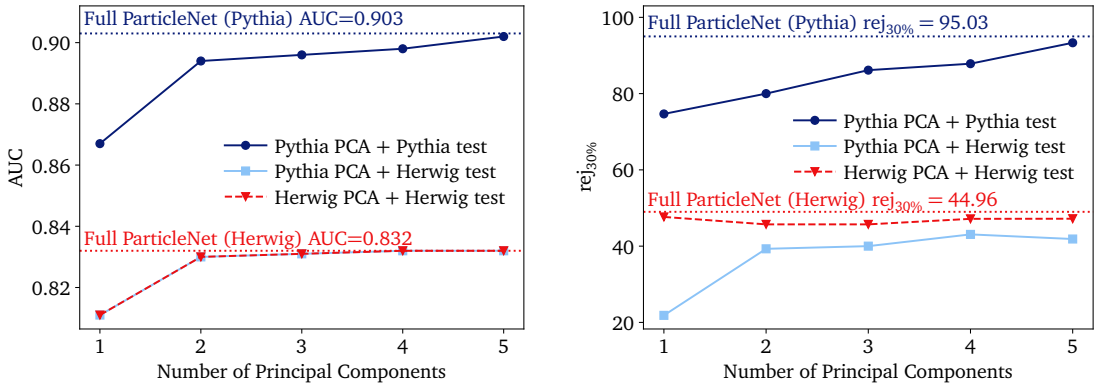


Figure 3: Performance of a NN-classifier on the best-performing PC combinations compared to the full ParticleNet-Lite for `Herwig` and `Pythia` data sets based on AUC and the rejection rate at 30% efficiency.

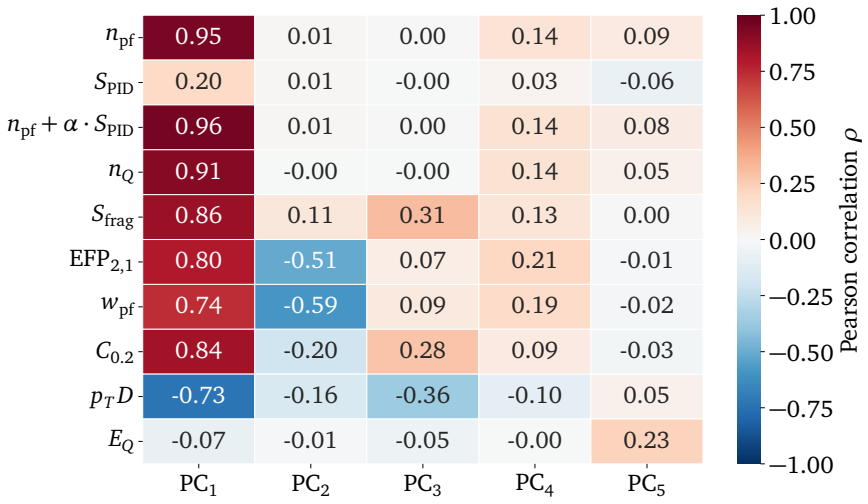


Figure 4: Pearson correlation ρ between the PCs and observables related to multiplicity and particle-type diversity as well as standard observables.

Simple EFPs that are sufficiently described only by the number of vertices v and the number of edges d, we denote as $\text{EFP}_{v,d}$. We also consider disconnected multigraphs, defining composite EFPs as

$$\text{EFP}_G = \prod_{g \in C(G)} \text{EFP}_g \,, \tag{11}$$

where $C(G)$ denotes the set of connected components of the multigraph G. We include all multigraphs with up to 7 edges, resulting in a overcomplete set of about 1000 EFPs. We then investigate how these observables correlate with the three leading principal components of our trained quark–gluon tagger by evaluating the Pearson correlation coefficient.

PC_1: Constituent number and diversity

In Figure 4, we see that the first principal component PC_1 is dominated by observables related to the number of particles and their particle nature. Two strongly correlated observables with PC_1 are n_{pf} and the charged multiplicity n_Q, defined as the number of charged particles within a jet. In addition, PC_1 is correlated with the PID entropy

$$S_{\text{PID}} = -\sum_{\text{type}\, j} f_j \log f_j \,, \tag{12}$$

where f_j is the fraction of particles of type j. It captures the diversity of particle types in the jet. Gluon jets, which radiate more and produce a broader mix of particles, have larger S_{PID} and multiplicity. Even if the correlation is small, it is relevant since the linear combination

$$n_{\text{pf}} + \alpha \cdot S_{\text{PID}} = n_{\text{pf}} + 12.3 \cdot S_{\text{PID}} \,, \tag{13}$$

achieves a slightly higher correlation with PC_1. The factor 12.3 was tuned to maximize the correlation with PC_1. This indicates that S_{PID} adds information and is not just correlated with PC_1 through n_{pf}. This means PC_1 reflects both the quantity and diversity of jet constituents.

PC_2: Radial energy profile

Also in Fig. 4, we see that PC_2 captures how energy is distributed radially around the jet axis, independently of multiplicity. It is correlated with several observables sensitive to jet width and shape. An observable only correlated with PC_2 is the ellipticity, defined in terms of the jet inertia tensor [81] in the transverse plane,

$$I^{ij} = \sum_{k\in\text{jet}} p_{T,k} \frac{r_k^i r_k^j}{r_k^2}\,, \tag{14}$$

Here, r_k^i are the components of the transverse position vector of the constituent k relative to the jet axis. The ellipticity is given in terms of χ_{min} and χ_{max} as the eigenvalues of this tensor,

$$\epsilon = \frac{\chi_{\text{min}}}{\chi_{\text{max}}}\,. \tag{15}$$

Lower ellipticity corresponds to more elongated (non-circular) jets. Moreover, PC_2 is strongly correlated with w_{pf}, which is in turn correlated with n_{pf} and PC_1. To exploit this additional direction, we construct the de-correlated combination

$$w_{\text{pf}}^{\perp} = \alpha \cdot n_{\text{pf}} - w_{\text{pf}}\,, \tag{16}$$

where $\alpha = 0.0016$ minimizes the linear correlation with n_{pf}. It remains sensitive to the jet width while removing the dependence on the multiplicity and therefore removing the correlation to PC_1. The minus sign is chosen since w_{pf} is negatively correlated with PC_2 and we chose to obtain a positive correlation. In addition, we introduce the generalized angularities [42]

$$\lambda_k^\beta = \sum_i z_i^\beta \Delta R_{i,\text{jet}}^k\,. \tag{17}$$

Among those, λ_1^2 is strongly correlated with PC_2. In the spirit of energy correlation functions, we can define a ratio between the Les Houches angularity $\lambda_{0.5}^1$ [42, 82] and λ_1^2

$$r_\lambda = \frac{\lambda_{0.5}^1}{\lambda_1^2}\,. \tag{18}$$

The numerator $\lambda_{0.5}^1$ gives weight to soft emissions at moderate angular scales, typical for gluon jets; the denominator normalizes the broader radial energy profile. This construction has several advantages: (i) it captures the core and the periphery of the jet; (ii) it is dimensionless and robust against global energy rescaling; and (iii) it is naturally decorrelated from the multiplicity as the numerator and denominator share the same structure. Here, r_λ serves as an interpretable data-driven proxy for PC_2, not as a perturbative prediction for the full distribution. The left panel of Fig. 5 shows that PC_2 captures the genuine jet shape and radial flow, distinct from PC_1.

PC_3: Fragmentation and energy dispersion

Finally, PC_3 is associated with the way energy is shared among jet constituents, corresponding to the fragmentation pattern. The fragmentation entropy is given by

$$S_{\text{frag}} = -\sum_i z_i \log z_i\,, \tag{19}$$

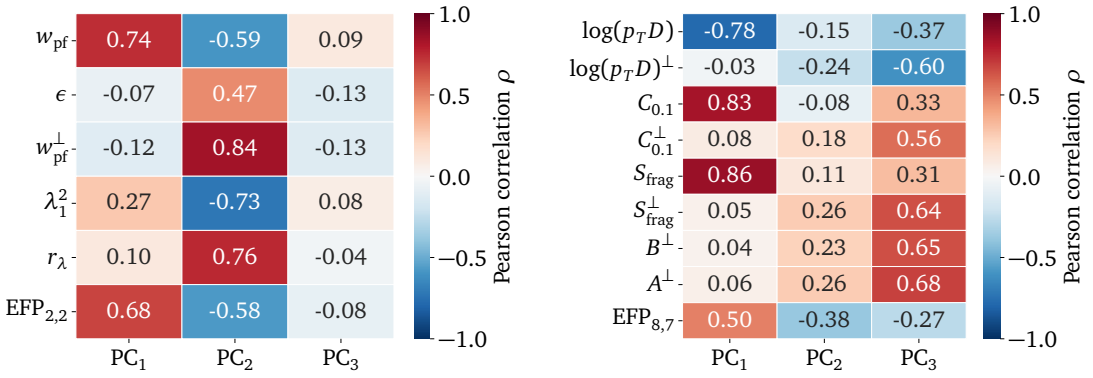


Figure 5: Correlation between the first three principal components and radial jet shape observables(left) and fragmentation and energy dispersion observables (right).

and measures how evenly the transverse momentum is distributed. Quark jets tend to have a lower fragmentation entropy because of their harder fragmentation.

The PCA components are linearly uncorrelated, so PC_3 captures variation in the latent space orthogonal to $\text{PC}_1 \approx n_{\text{pf}}$ and $\text{PC}_2 \approx w_{\text{pf}}^{\perp}$. This suggests that PC_3 encodes physical information that is independent of these two observables and is instead related to fragmentation and energy dispersion. To test this, we construct candidate observables that may align with PC_3, but remove any linear correlation with n_{pf} and $w_{\text{pf}}^{\perp}$,

$$O^{\perp} = O - \beta n_{\text{pf}} - \gamma w_{\text{pf}}^{\perp} . \tag{20}$$

where β and γ are chosen to minimize the linear correlation with n_{pf} and $w_{\text{pf}}^{\perp}$. This isolates the part of O that corresponds to the latent direction of PC_3. After testing various combinations, the following observables are strongly correlated with PC_3:

$$\begin{aligned}
A^{\perp} &= S_{\text{frag}} \frac{C_{0.1}}{C_{0.05}} - 0.03 \cdot n_{\text{pf}} + 1.95 w_{\text{pf}}^{\perp} \\
B^{\perp} &= -C_{0.1} \cdot \log(p_T D) \cdot C_{0.05} - 0.014 n_{\text{pf}} + 21.32 w_{\text{pf}}^{\perp} \\
S_{\text{frag}}^{\perp} &= S_{\text{frag}} - 0.03 n_{\text{pf}} + 0.45 w_{\text{pf}}^{\perp} \\
C_{0.1}^{\perp} &= C_{0.1} - 0.0046 n_{\text{pf}} + 0.7701 w_{\text{pf}}^{\perp} \\
\log(p_T D)^{\perp} &= \log(p_T D) + 0.0143 \cdot n_{\text{pf}} - 0.065 \cdot w_{\text{pf}}^{\perp} .
\end{aligned} \tag{21}$$

All these observables are sensitive to the distribution of transverse momentum within the jet, characterizing the extent to which the fragmentation pattern is hard or diffuse. In the right panel of Fig. 5 we see that PC_3 is significantly correlated with these fragmentation-sensitive observables, which means PC_3 capturing aspects of fragmentation dynamics and energy dispersion.

PC_4 and PC_5

Beyond the first three PCs, a clear physical interpretation becomes increasingly challenging. This difficulty arises because many jet observables are highly correlated and tend to span similar directions in feature space. Consequently, the leading PCs capture most of the variance associated with well-understood QCD observables, while the subleading components reflect more subtle structures that may represent combinations of multiple physical effects.

In our analysis, we did not find an individual observable that is strongly or uniquely correlated with PC_4. However, PC_5 shows a notable correlation with the charged energy fraction

$$E_Q = \frac{E_{\text{charged}}}{E_{\text{jet}}} \,. \tag{22}$$

This suggests that the ParticleNet learns charge-related information in a non-trivial and decorrelated way. Unlike PC_{1-3}, which align closely with standard observables, PC_5 does not map directly onto a single feature but captures a more subtle charge structure of the jet.

Throughout this analysis, we find that standard jet observables outperform individual EFPs when used as single variables. In principle, whenever a principal component corresponds to an IRC-safe observable, it can be written as a linear combination of EFPs, since they span the entire space of IRC-safe observables. In our setup, however, we do not impose IRC safety as ParticleNet is not constrained to ignore IRC-unsafe features. In practice, most of the variance of PC_1 is captured by the multiplicity n_{pf}, which is IRC-unsafe and therefore does not lie in the EFP vector span. The same applies to other IRC-unsafe quantities such as S_{PID}, so we cannot exactly express principal components dominated by these observables in terms of EFPs.

A further limitation is interpretability. Individual EFPs correspond to graphs, but their detailed physical meaning is not yet associated with clear semantic concepts. Roughly speaking, an EFP with many vertices probes higher-order multi-particle correlations, while a graph with many edges places more weight on angular separations ΔR and the fine structure of the radiation pattern. For high-degree or composite observables built from many EFPs, this picture becomes increasingly opaque. In general, EFPs are well-suited for probing IRC-safe subspaces of jet observables. In this study, we use EFPs mainly as an explicitly IRC-safe reference, not as our main physics observables.

Mutual information

The Pearson correlation ρ only captures the linear relationship between two observables. To quantify non-linear effects, we compute the mutual information (MI). The MI is a measure of the shared information between two random variables. It is usually measured in (Shannon) bits or nats. For continuous variables the mutual information I(X;Y) of jointly continuous (X, Y) variables is given by

$$\mathrm{I}(X;Y) = \int \mathrm{d}x\mathrm{d}y\, p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \,, \tag{23}$$

where $p(x,y)$ is the joint probability density function and $p(x)$ and $p(y)$ are the marginal probability density functions. The MI has no upper bound, making it intrinsically hard to interpret. For continuous variables, the exact mutual information is practically intractable, so we compute it using a k-nearest-neighbor method as implented in `scikit-learn`. As a reference, we also consider the case where (X, Y) has a purely Gaussian dependence with Pearson correlation ρ, for which the mutual information is simply given by

$$\mathrm{I}(X;Y)_{\text{Gauss}} = -\frac{1}{2}\log(1-\rho^2) \,. \tag{24}$$

By comparing the k-nearest-neighbor estimate $I_{\text{kNN}}(X;Y)$ to this Gaussian baseline, we can test whether the observable exhibits additional non-Gaussian structure beyond what is encoded in the linear correlation.

We evaluate the MI for a number of nearest neighbors $k \in \{3, 5, 7, 12\}$ and average the results, using the standard deviation as an uncertainty estimate. Table 1 summarizes the mutual information between various jet observables and the principal components. Alongside the

Observable	PC_1 k-NN	PC_1 Gauss	PC_2 k-NN	PC_2 Gauss	PC_3 k-NN	PC_3 Gauss
$n_{\text{pf}} + \alpha \cdot S_{\text{PID}}$	1.2964(4)	1.273	0.1140(10)	0.002	0.1000(6)	0.005
n_{pf}	1.1860(6)	1.160	0.1083(7)	0.000	0.0979(9)	0.000
$\text{EFP}_{2,1}$	0.7578(8)	0.511	0.2683(3)	0.154	0.1019(6)	0.003
$\text{EFP}_{3,3}$	0.7804(4)	0.288	0.1974(5)	0.113	0.0830(5)	0.008
r_λ	0.017(2)	0.005	0.4772(5)	0.423	0.0114(6)	0.001
w_{pf}	0.6412(8)	0.398	0.3195(7)	0.210	0.1015(6)	0.005
$w_{\text{pf}}^{\perp}$	0.2123(4)	0.007	0.6520(7)	0.614	0.0355(8)	0.009
$\text{EFP}_{2,2}$	0.5937(9)	0.307	0.2482(4)	0.228	0.0720(6)	0.003
$\prod_{i=1}^{3} \text{EFP}_{2,1}^{(i)}$	0.755(3)	0.204	0.2709(13)	0.148	0.0991(10)	0.003
$S_{\text{frag}}^{\perp}$	0.1077(9)	0.001	0.0663(2)	0.034	0.3156(10)	0.267
S_{frag}	0.7481(12)	0.681	0.1414(11)	0.006	0.1353(2)	0.051
$C_{0.1}^{\perp}$	0.1705(5)	0.003	0.06345(2)	0.017	0.3037(13)	0.190
$C_{0.1}$	0.7363(6)	0.594	0.1890(9)	0.003	0.1648(7)	0.058
$\log(p_T D)^{\perp}$	0.048(2)	0.000	0.0475(2)	0.033	0.2522(5)	0.244
$\log(p_T D)$	0.5094(7)	0.001	0.1152(3)	0.029	0.1369(12)	0.226
$A^{\perp}$	0.0905(3)	0.002	0.0589(12)	0.034	0.3502(7)	0.311
$B^{\perp}$	0.0340(10)	0.001	0.039(2)	0.027	0.2991(14)	0.268
$\text{EFP}_{8,7}$	0.2913(4)	0.145	0.0990(7)	0.076	0.0699(2)	0.038
$\text{EFP}_{2,1}$	0.7578(9)	0.511	0.2692(3)	0.154	0.1019(6)	0.003

Table 1: Mutual information in nats between jet observables and principal components using an averaged k-nearest-neighbor estimator over various k values and the Gaussian baseline of Eq.(24).

k-NN estimate, we show the Gaussian baseline of Eq.(24), which represents the information content of a purely Gaussian copula with the same Pearson coefficient. When both values agree, the dependence between an observable and a given principal component is adequately described by a Gaussian copula, and we do not resolve additional non-Gaussian structure beyond what is captured by the linear correlation. This is the case for n_{pf} and PC_1, whose k-NN and Gaussian mutual information values are very close. In contrast, several observables, such as S_{frag}, exhibit a mutual information with PC_1 that exceeds the Gaussian reference, indicating additional non-Gaussian features in their joint distribution (e.g. non-linear or tail effects) despite already large Pearson correlations.

The Gaussian assumption concerns only the copula of the two variables. Even if the individual marginal distributions are non-Gaussian, their copula can still be Gaussian and lead to the MI in Eq.(24). A more detailed derivation of this is discussed in App. A. It is also important to stress that absolute MI values across different principal components are hard to compare: each PC_i has a different entropy, so the maximum achievable MI varies between components. A value that appears small for PC_1 may already saturate the available information for PC_2.

The resulting pattern provides an observable hierarchy that is consistent with the Pearson analysis, and additionally indicates whether a given quantity is effectively encoded through a Gaussian dependence or requires a non-Gaussian structure. A representative example is w_{pf}: its mutual information with PC_1 significantly exceeds the corresponding Gaussian baseline, and a copula fit favors a Student-t distribution over the Gaussian ansatz, indicating enhanced tail dependence between w_{pf} and PC_1. Once the dependence on n_{pf} is removed, the remaining dependence is well described by a Gaussian copula, and the residual mutual information becomes small. This shows that apparent non-linear relationships can largely be absorbed by leading observables such as n_{pf}.

3.2 Non-linear dimensionality reduction

Using PCA we have analyzed the latent space of ParticleNet-Lite in a linear approximation. To probe non-linear structures we introduce a Disentangled Latent Classifier (DLC), a network that compresses the 64-dimensional X into a lower-dimensional latent representation, while simultaneously learning to classify quark and gluon jets. The architecture is visualized in Fig. 6. The DLC consists of three components: (i) an encoder that maps each input $x_i \in \mathbb{R}^{64}$, drawn from the dataset $X \in \mathbb{R}^{N\times 64}$, to a latent vector $\ell_i \in \mathbb{R}^d$; (ii) a decoder that reconstructs the original input $\hat{x}_i$ from ℓ_i; and (iii) a classification head that predicts the jet label $y_i \in \{0,1\}$ from the latent representation. The corresponding loss for N jets is

$$\mathcal{L} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}|x_i-\hat{x}_i|^2}_{\mathcal{L}_{\text{reco}}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\Big[y_i\log\sigma(\ell_i)+(1-y_i)\log(1-\sigma(\ell_i))\Big]}_{\mathcal{L}_{\text{class}}} + \underbrace{\sum_{j\neq k}\Big[\text{Cov}(\ell_j,\ell_k)\Big]^2}_{\mathcal{L}_{\text{disentangle}}} . \tag{25}$$

The first term is the MSE between the input vector x_i and its reconstruction $\hat{x}_i$. The second term is the binary cross-entropy, where $\sigma(z_i)$ denotes the predicted probability for jet i in its latent representation. The third term penalizes linear correlations between components of the latent vector by summing the squared off-diagonal elements of the covariance matrix. This loss encourages the network to encode linearly independent features in each latent direction. The covariance penalty is not intended to enforce full (non-linear) independence. Instead, we aim to avoid redundant latent directions that all align with the same dominant factor (e.g. n_{pf}) or, more generally, with the same reference observable. Since Pearson correlation measures linear dependence, an approximately decorrelated latent space allows us to attribute and rank high-level observables based on their Pearson correlations with individual latent directions.

By contrast, a mutual information–based disentangling loss is poorly aligned with our objectives. Since our set of observables is intrinsically correlated, minimizing mutual information would favor representations that compress information into a small number of non-linear latent directions rather than distribute it across interpretable components. We therefore rely solely on a linear covariance penalty.

To determine the minimal latent dimensionality for successful classification, we train the DLC with different latent space sizes and evaluate the AUC and the calibration. Calibration

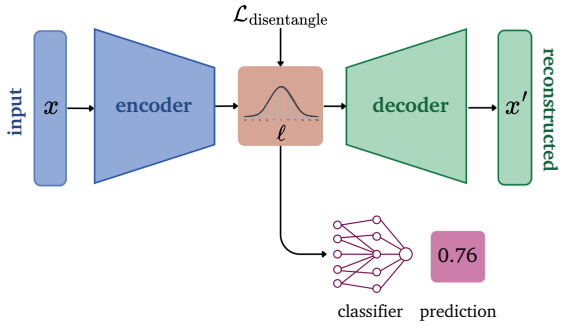


Figure 6: Setup of the DLC. The input is compressed through the encoder to a latent space ℓ, which is enforced to be disentangled. The disentangled latent space is then passed to a classifier as well as a decoder for reconstruction

Latent Dim	1	2	3	4
AUC	0.893(2)	0.9001(4)	0.9024(4)	0.9034(2)
$\text{rej}_{30\%}$	72(3)	77(3)	95(5)	95(3)
ΔC	1.8(3)	0.93(5)	1.0(16)	0.9(15)

Table 2: AUC and calibration for different latent dimensions, averaged over five runs.

curves test how well predicted probabilities match observed frequencies. A perfectly calibrated network produces a diagonal curve. To quantify the deviation from the diagonal we use the expected calibration error over K bins of the calibration curve

$$\Delta C = 100 \cdot \sum_{k=1}^{K} \frac{N_k}{K} |p_k - f_k| \ , \tag{26}$$

where N_k are the number of events in the k^{th} bin, p_k is the average predicted probability in that bin, and f_k is the observed frequency of positive labels. A lower ΔC means better calibration. For readability we include a factor of 100.

Table 2 shows that a latent space with just three dimensions achieves nearly the same AUC as the full 64-dimensional ParticleNet-Lite output, as well as a decent calibration. In principle, any classifier maps to a scalar output and constructs a one-dimensional representation for discrimination. However, including a reconstruction loss constrains the network to preserve a compressed version of the full feature space, yielding a structured and interpretable latent representation.

In principle, there is no guarantee that the latent directions are the same across different training runs. They can be permuted or have their signs flipped. To test robustness, we train the DLC 10 times with different random seeds. We use the first run as a reference and align the remaining runs to the reference latent directions using the Pearson correlation. If a latent direction ℓ_1 in the reference run correlates strongly (close to ± 1) with a direction ℓ_2 in another run, we permute the latent axes accordingly and flip the sign if needed.

After this alignment, we compute Pearson correlations between the jet observables and the latent directions and determine the mean and standard deviation across runs. For 7 out of 9 non-reference runs, a permutation was required, and in total, we applied 13 individual sign flips. This confirms that the order and sign of the latent directions are arbitrary, while their

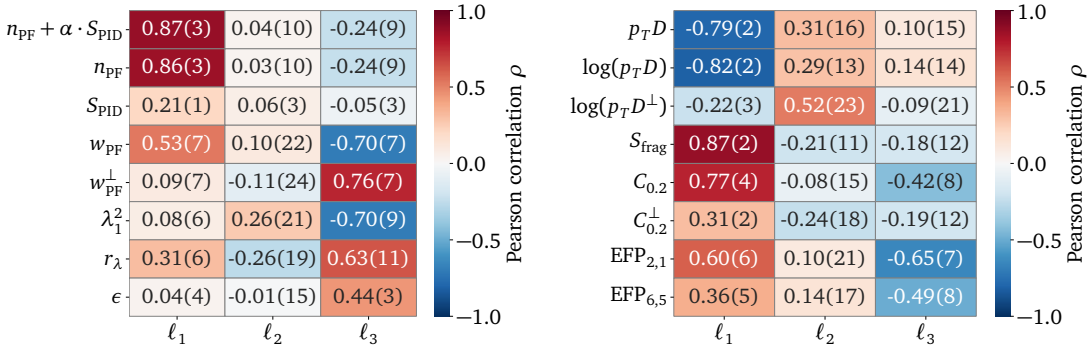


Figure 7: Pearson correlations ρ between physics observables and the linearly disentangled latent spaces ℓ_i. The uncertainties are computed from 10 independent runs.

Hyperparameter	Value
Latent dimension d_ℓ	3
Encoder architecture	$64 \to 128 \to 64 \to d_\ell$
Decoder architecture	$d_\ell \to 64 \to 128 \to 64$
Classifier architecture	$d_\ell \to 64 \to 1$
Activation function	ReLU
Batch normalization	Encoder (128, 64)
Dropout (encoder, classifier)	0.3
Optimizer	Adam
Learning rate	10^{-4}
Batch size	256
Number of epochs	100
Feature scaling	StandardScaler (zero mean, unit variance)

Table 3: Hyperparameters of the DLC.

physics interpretation remains stable, as shown in Fig. 7. We find a structure very similar to the PCA analysis. The first latent dimension is dominated by n_{pf}, the linear combination of n_{pf} and S_{PID} and fragmentation observables, the third by shape observables, and the second by more moderate correlations with fragmentation-related quantities. Overall, the correlations are stable across runs after alignment. Only the correlations with ℓ_2 vary more, but their correlations with physics observables are also moderate.

While the latent dimensions in DLC are learned to be linearly uncorrelated, their separation is less clean than in PCA. This is because the non-linear transformations can cause overlap in the physical interpretation of different latent directions. For instance, $\log(p_T D)$ is strongly correlated with the first latent direction ℓ_1, but this correlation disappears once the linear dependence on n_{pf} is removed. This suggests that the correlation is mainly due to the strong dependence of $\log(p_T D)$ on multiplicity, rather than an intrinsic feature of ℓ_1. While the DLC structure resembles the PCA, the mapping between latent dimensions and physical observables is more general but less direct.

4 Feature importance from Shapley values

Rather than learning and analyzing latent representations, we can train a simple NN-classifier and analyze the feature or observable importance using the SHapley Additive exPlanations (SHAP) framework [83]. Shapley values assign a contribution to each input feature for the classifier output $f(x)$, based on cooperative game theory [84]. For a given feature i, the Shapley value $\mathcal{V}_i$ is defined as the average marginal contribution of i across all subsets of the remaining features:

$$\mathcal{V}_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \Big[f(S \cup \{i\}) - f(S) \Big] . \tag{27}$$

Here, F is the full set of input features (e.g. jet observables), and S is a subset of F that does not contain i. The term $|S|$ ($|F|$) denotes the number of features in S (F), and the sum averages the contribution of feature i over all such subsets. The model output $f(S)$ represents the expected classifier prediction when only the features in S are known. Computing $f(S)$ requires marginalizing over the remaining features $B = F \setminus S$, which is generally intractable. To make this feasible, the model-agnostic kernel SHAP makes a simplifying assumption and replaces the conditional distribution $p(x_B|x_S)$ with the marginal distribution $p(x_B)$:

$$f(S) = \big\langle f(x_S, x_B) \big\rangle_{x_B \sim p(x_B|x_S)} \approx \big\langle f(x_S, x_B) \big\rangle_{x_B \sim p(x_B)} . \tag{28}$$

It renders the Shapley analysis numerically feasible, but it can lead to misleading attributions when features are strongly correlated. In such cases, SHAP may undervalue features that are informative but share mutual information with others.

Table 4 shows AUC scores for various combinations of high-level observables, to guide our choice of input sets for the SHAP analysis. The left panel of Fig. 8 shows the SHAP summary for the standard tagging observables defined in Eq.(9). Positive SHAP values indicate that a feature increases the confidence of the network in the quark label, negative values push the prediction towards the gluon label. The features n_{pf} and $C_{0.2}$ behave as expected: low

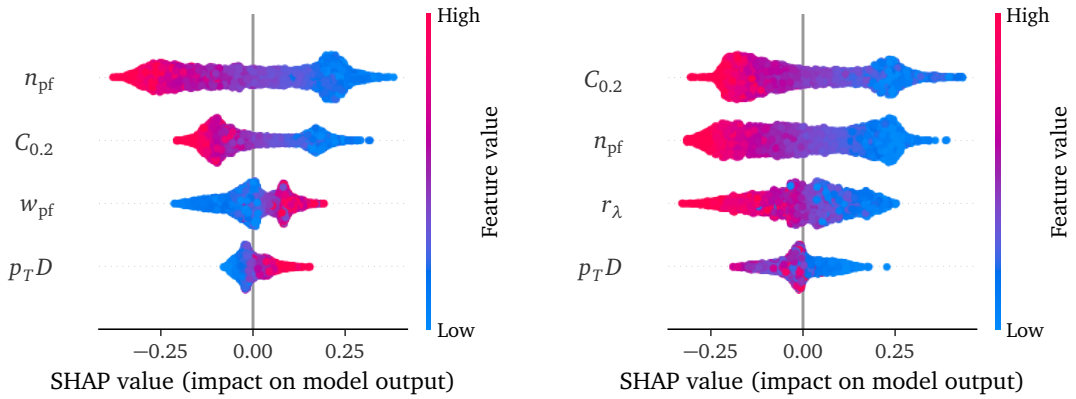


Figure 8: SHAP summary plots for the observables in Eq.(9): baseline set (left) and with w_{pf} replaced by r_λ (right). Observables (features) are ordered by mean absolute SHAP value across events. Each dot is one event; color encodes the observable value for that event (blue = low, red = high). A positive SHAP value indicates stronger contribution towards the quark score.

interpretation	Observables	AUC	$\text{rej}_{30\%}$	$\text{rej}_{50\%}$
standard set	$n_{\text{pf}}, w_{\text{pf}}, p_T D, C_{0.2}$	0.8626	71.11	21.53
decorrelated girth	$n_{\text{pf}}, r_\lambda, p_T D, C_{0.2}$	0.8720	70.00	23.33
PC1 approx.	n_{pf}	0.8406	53.33	16.65
	$n_{\text{pf}}, S_{\text{PID}}$	0.8489	64.35	20.32
PC1-2 approx.	$n_{\text{pf}}, S_{\text{PID}}, w_{\text{pf}}$	0.8691	70.60	21.75
	$n_{\text{pf}}, S_{\text{PID}}, w_{\text{pf}}^\perp$	0.8690	72.20	21.66
	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda$	0.8733	67.99	21.20
PC1-3 approx.	$n_{\text{pf}}, r_\lambda, A^\perp$	0.8645	67.88	19.31
	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda, A^\perp$	0.8774	73.10	23.45
	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda, C_{0.2}$	0.8778	77.48	24.78
	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda, C_{0.2}, p_T D$	0.8777	77.24	24.34
	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda, C_{0.2}, p_T D, S_{\text{frag}}$	0.8825	79.42	26.80
PC1-5 approx.	$n_{\text{pf}}, S_{\text{PID}}, r_\lambda, S_{\text{frag}}, C_{0.2}, p_T D, E_Q$	0.8841	80.89	27.52

Table 4: AUC scores for different combinations of three to seven high-level jet observables.

particle multiplicity and a small energy correlation suggest a quark jet. We also see that $p_T D$ contributes little to the classification.

The feature w_{pf} in the same panel displays a counter-intuitive pattern: jets with low w_{pf}, typically indicative of narrow, quark-like jets, receive negative SHAP values related to a gluon classification. This is not a failure of the classifier, but a limitation of the SHAP attribution mechanism. Low w_{pf} occurs in both, quark jets (with low n_{pf}) and some gluon jets (with high n_{pf}), due to their correlation. The classifier correctly learns that low w_{pf} combined with high n_{pf} is characteristic of gluon jets. However, SHAP evaluates the contribution of w_{pf} by marginalizing over n_{pf} and other features, assuming independence and thereby ignoring their joint structure. As a result, SHAP assigns a negative contribution to w_{pf} even when, conditional on n_{pf}, it should favor a quark classification.

To address this mis-attribution, we replace w_{pf} with r_λ, the decorrelated alternative introduced in Eq.(18), as part of a minimal input set,

$$\left\{ n_{\text{pf}}, w_{\text{pf}}, p_T D, C_{0.2} \right\} \quad \longrightarrow \quad \left\{ n_{\text{pf}}, r_\lambda, p_T D, C_{0.2} \right\} . \tag{29}$$

In the right panel of Fig. 8 the features r_λ, n_{pf} and $C_{0.2}$, now show a straightforward interpretation. The remaining issue is again methodological: While replacing w_{pf} by r_λ removes the strongest width–multiplicity, strong correlations among the remaining observables persist. In particular, $p_T D$ is strongly anti-correlated with n_{pf}. Consequently, the marginalization used by SHAP breaks this dependence and the resulting $p_T D$ attributions again reflect a counterfactual variation of one observable while the correlated one is effectively unconstrained. This is the same mechanism that produced the misleading w_{pf} attribution in the left panel of Fig. 8.

The problem becomes more visible once we increase the number of correlated inputs. The left panel of Fig. 9 shows SHAP values for a classifier trained on observables intended to approximate the first three principal components from Sec. 3. Here $C_{0.2}$ displays the opposite qualitative behaviour compared to Fig. 8. This is not a change in the physical information carried by $C_{0.2}$. It is a consequence of correlated inputs: SHAP distributes shared information among features in a basis-dependent way, which can change both the apparent ranking and the apparent sign structure. This also clarifies why n_{pf} and $C_{0.2}$ swap roles between Fig. 8 and Fig. 9. Our PCA analysis, together with long-established expectations from quark–gluon tag-

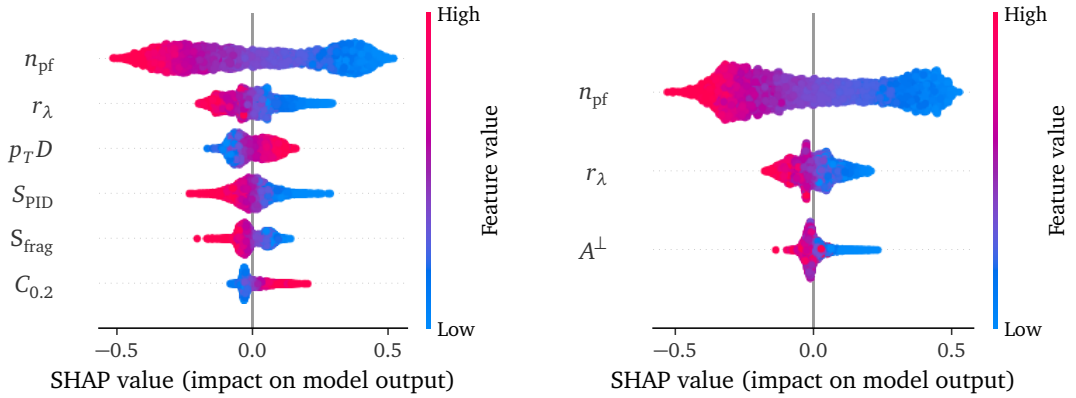


Figure 9: SHAP summaries for set of observables approximating the first three PCA components. Left: strong correlations among features distort SHAP attributions, particularly for $C_{0.2}$ where the importance direction is counterintuitive. Right: SHAP values for a decorrelated feature set approximating the first three PCA components. The importance of each feature aligns with the ranking of principal components.

ging studies, indicates that the multiplicity direction is dominant. The fact that $C_{0.2}$ appears first in Fig. 8 should therefore be interpreted as a correlation artifact, i.e. , multiplicity information is partially attributed to a correlated proxy. When additional correlated observables are introduced, this bookkeeping changes and more of the shared contribution is assigned back to n_{pf}, so that the ordering becomes closer to the physically expected one. The absolute magnitudes of the SHAP values, however, remain basis-dependent in the correlated setting and should not be overinterpreted. To obtain stable and physically meaningful attributions, we therefore work in an approximately decorrelated basis guided by the PCA interpretation and use

$$\{n_{\mathrm{pf}}, r_\lambda, A^\perp\}. \tag{30}$$

We could have picked other observables for the third component, but we use $A^\perp$ since it combines energy correlations and fragmentation entropy. The SHAP plot for this set is shown in the right panel of Fig. 9, and it aligns with our earlier PCA findings. n_{pf} is clearly the leading feature, with r_λ and $A^\perp$ adding extra discrimination.

By default, SHAP ranks features by their average absolute contribution. When the inputs are properly decorrelated, this ranking generally matches both intuition and our PCA results, making it easier to interpret. However, with correlated features, SHAP gives misleading attributions, even if the overall classifier still works fine. This means we need to carefully prepare inputs, using decorrelated features or a PCA basis, to get SHAP explanations that actually reflect the physics. SHAP is still a powerful tool, but we have to be mindful of these subtleties when applying it to jet observables.

5 Symbolic regression

Having analyzed the internal structure of the trained network using principal components and Shapley values, we now ask directly: Can the classifier output be approximated by a formula built from high-level observables? This question leads directly to the language of theoretical physics, i.e. formulas and equations. In principle, neural networks can be approximated by formulas, and the extremely strong regularization of formulas can be helpful in cases of (too) little training data [61]. Instead of reasoning about latent vectors or weight matrices, we aim to represent the trained ParticleNet as a formula, capturing its dependencies on subjet observables. Our goal is to express the machine-learned decision boundaries in terms of known physical quantities.

We perform symbolic regression using the PySR framework [55], which searches for formulas by evolving a population of candidate formulas through a genetic algorithm. Each candidate is represented as a tree, constructed from a predefined set of mathematical operations. Each node in the tree contributes to the complexity. For example, the equation

$$3x + a \tag{31}$$

has a complexity of five, three for the parameters and two for the operations. The algorithm balances two competing objectives, accuracy and simplicity. This makes PySR particularly well suited for discovering compact formulas that approximate the network output.

Setup and method

We first select a set of observables based on their performance and interpretability, as discussed in the previous sections. These include particle multiplicity, radial energy distribution, fragmentation entropy, momentum balance, the two-point correlation function, the charged energy fraction, and the PID entropy,

$$\left\{n_{\text{pf}}, S_{\text{PID}}, r_\lambda, S_{\text{frag}}, p_T D, C_{0.2}, E_Q\right\} . \tag{32}$$

For each input observable (or combination), we first train a simple neural network classifier that uses only those observables. Its output defines the target for the symbolic regression. This isolates the contribution of the chosen observables and ensures that the formulas approximate a realistic, learnable decision function.

For symbolic regression, we use PySR with a fixed set of operators including addition, multiplication, division, powers, and a small number of non-linear activation functions such as $\tanh(x)$. For single-observable fits, we allow a maximum complexity of 10; for two-observable combinations, we increase the limit to 22. The formulas are evaluated based on three criteria that balance precision and interpretability:

1. the area under the ROC curve (AUC);
2. the background rejection at 30% quark efficiency; and
3. the calibration metric ΔC defined in Eq.(26).

5.1 One-dimensional regression

We begin by applying symbolic regression to individual high-level observables, to see whether the tagger's decision surface, restricted to a single input observable, can be captured by a simple formula. For each observable $\mathcal{O}$, we train a neural network using only $\mathcal{O}$ as input and record its predicted quark probability $y_{\mathcal{O}}$. Symbolic regression is then used to approximate

$$f(\mathcal{O}) \approx y_{\mathcal{O}} . \tag{33}$$

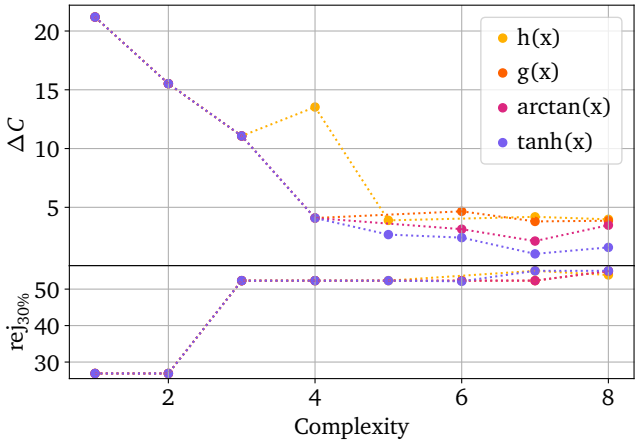


Figure 10: Comparison of activation functions in symbolic regression using n_{pf}. The calibration error ΔC and background rejection at 30% quark efficiency are shown as a function of formula complexity.

To understand the role of functional choices, we first focus on the particle multiplicity n_{pf} as the most discriminative observable for quark-gluon tagging.

Activation functions

We first study how the choice of non-linear activation functions affects the learned formulas. Since the tagger outputs are bounded between 0 and 1, it is natural to include bounded non-linearities that can model this range effectively. At the same time, we want to keep the formulas as compact and interpretable as possible. We test standard options like $\tanh(x)$ and $\arctan(x)$, as well as two custom choices

$$h(x) = \frac{-x}{\sqrt{1+x^2}} \qquad \text{and} \qquad g(x) = \frac{1}{\pi}\arctan(x) + 0.5\,. \tag{34}$$

In principle, PySR can build complex activation functions such as the sigmoid from elementary functions, but this would require a complexity of 19. Hence, explicitly providing more complex activation functions significantly boosts the performance. However, there are two reasons for

Hyperparameter	Value
iterations	5000
cycles per iteration	800
binary operators	$+, -, \times, \div$
unary operators	$x^2, x^3, \sqrt{x}, \tanh x$
populations	70
population size	40
procs	32
batching	True
maxsize	10 (1D), 22 (2D), 40 (7D)
precision	32
turbo	True
warmup maxsize by	0.05

Table 5: Hyperparameter settings for PySR.

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.083	0.5	3.33	-
3	$\frac{17.7}{n_{\text{pf}}}$	0.029	0.839	52.32	11.34
4	$\tanh \frac{22.3}{n_{\text{pf}}}$	0.0169	0.839	52.32	12.23
5	$\tanh \frac{850.9}{n_{\text{pf}}^2}$	0.0009	0.839	52.32	3.43
6	$\tanh^6 \frac{57.447636}{n_{\text{pf}}}$	0.0006	0.839	52.32	2.04
7	$\tanh \frac{44.19}{(0.084 \cdot n_{\text{pf}}+1)^3}$	0.00026	0.839	52.32	1.64
8	$\tanh^3\left(681.83 \cdot \left(0.014 + \frac{1}{n_{\text{pf}}}\right)^2\right)$	0.00025	0.839	52.32	1.58
9	$0.94 \cdot \tanh\left(21036 \cdot \left(0.005 + \frac{1}{n_{\text{pf}}}\right)^3\right)$	0.00004	0.839	52.32	1.08

Table 6: 1D symbolic regression results for n_{pf} only.

limiting the symbolic algorithm to a single non-linear function. First, PySR develops a bias toward an activation function appearing early in its formula search. Once a function like $\tanh(x)$ appears, the evolutionary search typically continues to build on it and ignores the better performance of alternative functions. Second, fixing the activation function reduces the complexity and prevents overly convoluted structures.

To explore this in a controlled setting, we perform a scan using only n_{pf} as input, and vary the allowed complexity from 1 to 10. For each function, we track the calibration error ΔC and the background rejection at 30% signal efficiency. Figure 10 summarizes the results.

The different activation functions perform similarly in terms of AUC, but $\tanh(x)$ consistently produces better-calibrated outputs and leads to shorter, cleaner formulas. Based on this, we adopt $\tanh(x)$ as the default non-linearity for all remaining regression tasks. Note that we also considered the sigmoid function in earlier tests, but it was excluded from the final analysis due to its inferior performance.

Monotonicity and constant AUC

Looking at Table 6, we see how the formula evolves with complexity. At low complexity, the network starts with a simple inverse scaling, $\sim 1/n_{\text{pf}}$, capturing the trend that higher multiplicities are associated with gluon jets. As complexity increases, PySR sharpens this behavior by adding non-linear functions like tanh and raising them to higher powers. These refinements do not change the overall monotonic trend, but they improve the match to the classifier output. From complexity 7, additional gains come mainly from fine-tuning the shape. The formulas remain compact and interpretable, with increasing agreement with the network output.

A subtle point arises when comparing symbolic regressions based on the AUC. Because the order of the ROC curve is invariant under monotonic transformations [85], formulas that differ substantially in sharpness or calibration will give identical AUC scores. This is evident in most of the one-dimensional regressions, where all formulas are monotonic transformations with identical AUCs. In Fig. 11, we see that visually the formulas differ for n_{pf} even if the AUC remains the same. Additionally, we observe that higher complexities match the calibrated tagger prediction more closely, and we indeed require higher complexities for a good calibration.

observable	Complexity	Formula	AUC	$\text{Rej}_{30\%}$	ΔC
n_{pf}	9	$0.94 \cdot \tanh\left(21036 \cdot \left(0.005 + \frac{1}{n_{\text{pf}}}\right)^3\right)$	0.839	52.32	1.08
S_{frag}	6	$\tanh^2 \frac{18.08}{S_{\text{frag}}^3}$	0.828	38.97	1.80
$p_T D$	7	$0.92 \cdot \tanh\left(19.49 \cdot p_T D^3\right)$	0.807	26.87	1.04
$C_{0.2}$	9	$\tanh\left(343.13 \cdot (0.72 \cdot C_{0.2} - 1)^{18} + 0.22\right)$	0.793	58.41	1.54
r_λ	10	$\left((0.59 - \tanh(0.0038 \cdot r_\lambda))^2\right)^{0.5} + 0.22$	0.637	6.46	1.81
S_{PID}	7	$\tanh\left((S_{\text{PID}} - 1.63)^2\right) + 0.42$	0.599	6.19	1.94

Table 7: Best equations for each observable based on simplicity, performance and calibration. All numbers are rounded to 2 digits for readability.

Formulas for each observable

Having validated our strategy on n_{pf}, we now extend it to the full set of high-level observables

$$\left\{n_{\text{pf}}, S_{\text{PID}}, r_\lambda, S_{\text{frag}}, p_T D, C_{0.2}, E_Q\right\} . \tag{35}$$

Following the previous sections, they span different aspects of jet substructure, including multiplicity, angular spread, fragmentation, and charge.

For each observable, we first train a classifier and then use PySR to approximate it. Our maximum complexity is 10, to ensure the equations are interpretable. Table 7 summarizes the best formulas for each observable, along with the AUC, background rejection at 30% signal efficiency, and calibration error ΔC. For each case, we select the simplest formula that achieves good performance across all three metrics. Full complexity scans for each observable are provided in the Appendix.

We can observe patterns in these equations. The fragmentation entropy S_{frag}, behaves similarly to n_{pf} and yields a relatively simple formula. The inverse behavior of n_{pf} resembles Casimir scaling, higher particle diversity tends to favor the gluon label, which is assigned to 0. On the other end of the spectrum, r_λ shows limited discriminative power, and the corresponding formula is noticeably more complex. In general, more informative observables tend to produce simpler formulas, often involving only a few transformations to capture the relevant trends.

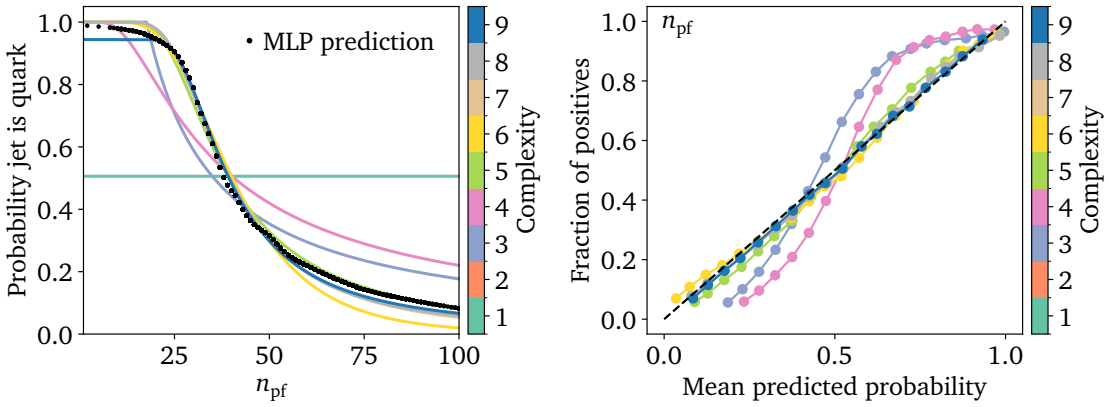


Figure 11: The left panel shows the symbolic regression curves compared to the target MLP prediction. The right panel shows the different calibration curves. A perfect calibration is shown by a diagonal (dashed black line).

5.2 Two-dimensional regression

Next, we apply symbolic regression to pairs of observables. We allow for a complexity of 22 to accurately describe the 2-dimensional dependence. Rather than testing all possible combinations, we focus on pairs including n_{pf} as a standard observable. To improve numerical stability and keep coefficient scales comparable, we multiply n_{pf} by a factor of 0.01 before regression. We deliberately avoid automated rescaling or ML-based normalization, as preserving the native physical scales of the observables supports direct interpretability of the resulting formulas.

At low complexities, the formulas remain effectively one-dimensional. This is expected, because even a basic linear combination of two observables can already have a complexity of 7 or more. For low complexity thresholds, the second observable is too expensive to be included in the formula. In fact, up to a complexity of 4, the symbolic regression arrives at the same formulas as for n_{pf} only. The complete set of formulas, ranging from simple to complex, is provided in the Appendix. In Fig. 12, we mark by vertical lines the point at which the second observable appears in the formula. Towards higher complexity, the second observable clearly improves the performance. Table 8 summarizes the best formulas for each n_{pf}-based pair, selected by balancing complexity, AUC, rejection rate, and calibration quality.

It is not always obvious which combinations of observables will yield the highest performance gain. Interestingly, r_λ performs poorly on its own, but complements n_{pf} best and leads to an AUC of 0.860. This shows that adding information decorrelated from n_{pf} improves the AUC the most. Similarly, $C_{0.2}$ improves the rejection rate, despite its modest individual AUC. In contrast, observables like $p_T D$ and E_Q produce simpler, more readable formulas that still reach competitive performance, especially in terms of calibration. This shows that combining n_{pf} with a second observable enables symbolic regression to access richer structures and yields better discrimination and equally interpretable formulas.

5.3 Towards all-observable regression

Finally, we turn to the question if the full ParticleNet can be approximated by a formula in terms of all seven observables from Eq.(32). We already know that adding an additional observable typically increases the formula complexity by at least five. Covering all observables should require a complexity of 40 or more. This scales poorly in terms of computational cost and

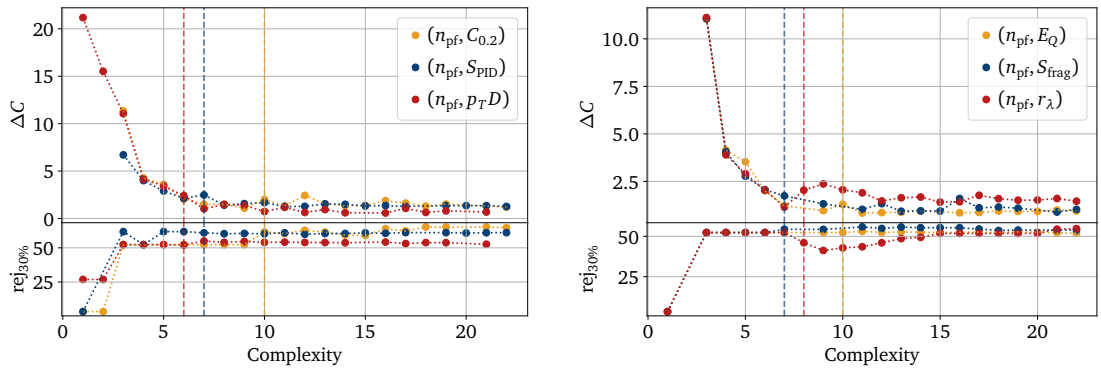


Figure 12: Rejection rates at 30% efficiency and calibration error dependent on the complexity. The vertical lines show when the equation depends for the first time on two observables.

Obs. pair	Complexity	Formula	AUC	$\text{Rej}_{30\%}$	ΔC
$(n_{\text{pf}}, r_\lambda)$	15	$\left(1.1 - n_{\text{pf}}\right) \cdot \tanh\left(181.1 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.003}{n_{\text{pf}}^3}\right)^3\right)$	0.860	51.81	1.41
$(n_{\text{pf}}, S_{\text{PID}})$	9	$\tanh\left(\frac{0.28}{n_{\text{pf}}^2 \cdot \left(n_{\text{pf}} + S_{\text{PID}}\right)^2}\right)$	0.849	64.43	1.01
$(n_{\text{pf}}, C_{0.2})$	18	$\left(0.03 - \tanh^3\left(\frac{82.47 \cdot \left((C_{0.2} - 0.68)^2 - 0.04\right)^2 + 0.54}{n_{\text{pf}}}\right)\right)^2$	0.848	65.27	1.30
$(n_{\text{pf}}, S_{\text{frag}})$	13	$0.94 \cdot \tanh^2\left(\frac{0.57}{\left(n_{\text{pf}} - 0.13\right) \cdot \left(n_{\text{pf}} - S_{\text{frag}}\right)}\right)$	0.844	55.74	0.93
$(n_{\text{pf}}, p_T D)$	7	$\tanh \frac{0.26 \cdot p_T D}{n_{\text{pf}}^2}$	0.844	55.01	1.02
(n_{pf}, E_Q)	11	$-0.098 \cdot E_Q + \tanh\left(0.11 + \frac{0.03}{n_{\text{pf}}^3}\right)$	0.840	52.36	0.84

Table 8: Symbolic regression results for pairs of observables including n_{pf}. Each formula is selected based on a balance of complexity, AUC, rejection rate, and calibration.

formula interpretability. Again, to preserve interpretability we only rescale n_{pf} by a factor of 0.01 to prevent large numerical coefficients from dominating the regression. To ensure a fair comparison, we train a new network on the same unscaled inputs. In Table 9 we compare the performance of the learned formula to this network.

In contrast to an estimated complexity around 40, we find that a complexity of 26 yields the best trade-off between performance and interpretability. Here, the learned formula matches the performance of the network. The best-performing formula at complexity 26 is

$$0.89 \cdot \tanh^3\left(\frac{0.008 \cdot C_{0.2}^2}{p_T D^2 \cdot \left(C_{0.2} - 0.29\right)^2} - 0.008 \cdot S_{\text{PID}} \cdot r_\lambda + \frac{0.56}{n_{\text{pf}}}\right) + 0.061\,. \tag{36}$$

It involves five of the seven observables, S_{frag} and E_Q are absent for the given complexity. The classifier output is a scaled $\tanh^3$ with a small offset, which provides a bounded non-linear mapping of a single effective score. The dominant contributions inside the activation come from three terms. The first term combines $C_{0.2}$ and $p_T D$ quadratically, showing sensitivity to jet fragmentation and momentum sharing. The second term is proportional to $S_{\text{PID}} \cdot r_\lambda$, showing that correlations between particle diversity and the radiation pattern contribute to the discrimination. The final term scales as $\frac{1}{n_{\text{pf}}}$, consistent with the strong dependence of quark–gluon separation on constituent multiplicity. Overall, the formula combines multiplicity, fragmentation, and radiation information in a compact and interpretable form.

To assess the robustness of the learned structure, we repeat the regression with PySR five times at fixed complexity 26. We bin the classifier score into 50 equally spaced bins and compute the mean and standard deviation of the predicted score over the resulting formulas in each bin. To illustrate how the distributions change, we plot the score distributions for quark and gluon jets and compare them with the MLP and ParticleNet-Lite baselines. Although the

Observables	Model	AUC	$\text{Rej}_{30\%}$
$(n_{\text{pf}}, S_{\text{PID}}, r_\lambda, S_{\text{frag}}, p_T D, C_{0.2}, E_Q)$	MLP	0.874	71.75
	PySR	0.874	68.12

Table 9: Performance of the full model using all observables. The symbolic regression formula in Eq.(36) has a complexity of 26.

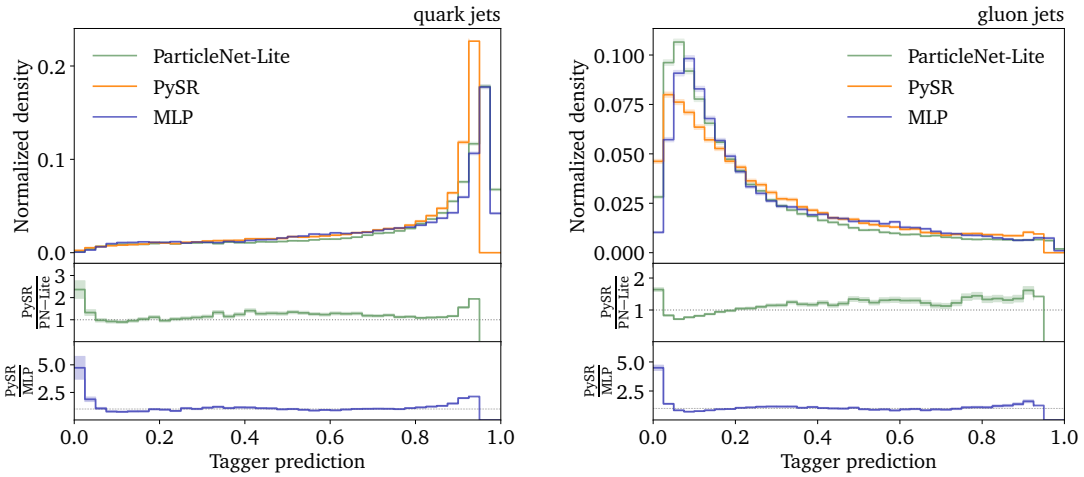


Figure 13: Score distribution plot for the MLP, PySR and the ParticleNet-Lite predictions. The bottom two panels show the ratio between a baseline (MLP or ParticleNet Lite) and the SR formula averaged over 6 runs with error band.

resulting formulas differ algebraically (see Tab. 10), their performance is numerically indistinguishable: the AUC values agree up to negligible differences, and the bin-by-bin variation of the score is too small to be visible in Fig. 13. Compared to the MLP and ParticleNet-Lite baselines, the PySR tagger produces sharper score distributions for quark jets, with more jets assigned values close to 1 and fewer close to 0.

This formula is an approximation of a tagger, meaning that it is only as robust as the tagger on which it was trained. Using a neural network, we obtain a classification prediction without explicit knowledge of how each input feature contributes. By contrast, the resulting expression provides a transparent, closed analytic form. If we evaluate this formula on a different dataset (e.g. different `Pythia` tunes or `Herwig`), its performance will reflect the robustness of the original NN classifier on which PySR was trained. We therefore take Eq.(36) and evaluate it on the `Herwig` dataset. To provide a fair AUC comparison, we also evaluate our MLP and ParticleNet-Lite on the same dataset. The results are shown in Tab. 11.

When evaluated on the `Herwig` dataset, all models exhibit noticeable performance degradation relative to the `Pythia` dataset. This is expected given intrinsic differences between the two generators, in particular their hadronization models: `Pythia` employs the Lund string model, while `Herwig` relies on the cluster model, leading to systematic shifts in observables

Trial	Equation	AUC
1	$0.89 \cdot \tanh^3\left(\frac{0.008 \cdot C_{0.2}^2}{p_T D^2 \cdot (C_{0.2}-0.29)^2} - 0.008 \cdot S_{\text{PID}} \cdot r_\lambda + \frac{0.56}{n_{\text{pf}}}\right) + 0.061$	0.874
2	$\left(0.7 - 0.29 \cdot \tanh\left(0.04 \cdot p_T D \cdot r_\lambda \cdot \left(S_{\text{PID}} + (C_{0.2}-1.36)^9\right) - \frac{0.74}{n_{\text{pf}}}\right)\right)^6$	0.874
3	$\tanh\left(0.27 \cdot \left(C_{0.2} + n_{\text{pf}}^4 \cdot \left(-S_{\text{PID}} + \frac{1.08 \cdot 10^{-5} \cdot r_\lambda^2}{(-C_{0.2}-0.03)^2} - 0.02 \cdot r_\lambda + \frac{0.37}{p_T D}\right)^4\right)^{-2}\right)$	0.874
4	$1.1 - \tanh^2\left(C_{0.2} + r_\lambda^2 \cdot \left(n_{\text{pf}} - 0.12\right)^2 \cdot \left(0.098 \cdot p_T D - (C_{0.2}-0.89)^6\right)^2 \cdot (S_{\text{PID}} - 0.14)^2\right)$	0.874
5	$\left[\tanh\left(\frac{219 \cdot \left(\frac{1}{r_\lambda}\right)^{3/2}}{n_{\text{pf}}^3 \cdot (p_T D + S_{\text{PID}})^3}\right) + 0.072 \cdot \tanh\left(\frac{0.127}{0.41 - C_{0.2}}\right)\right] \cdot (2.0 \cdot C_{0.2} + p_T D)^{-1}$	0.874

Table 10: Variability of formulas at a fixed complexity of 26.

observables	model	AUC	$\text{Rej}_{30\%}$
$(n_{\text{pf}}, S_{\text{PID}}, r_\lambda, S_{\text{frag}}, p_T D, C_{0.2}, E_Q)$	MLP	0.796	36.18
	PySR Eq.(36)	0.801	36.23
	ParticleNet-Lite	0.804	37.41

Table 11: Performance of the symbolic regression for a network trained on `Pythia` and evaluated on `Herwig`, comparing symbolic regression and a MLP and ParticleNet-Lite

such as particle multiplicity and fragmentation patterns, as also discussed in Ref. [43]. The `PySR` formula performs on par with both neural-network baselines on `Herwig`, both in terms of AUC and background rejection. Its cross-generator behavior closely tracks that of the underlying network it approximates.

Importantly, the formula makes the sources of this generator dependence transparent. The dominant contributions arise from observables known to be sensitive to hadronization modeling, in particular the constituent multiplicity and fragmentation-related features. As a result, the observed performance degradation can be directly linked to shifts in these inputs, rather than to opaque correlations learned by the network. In this sense, symbolic regression does not remove generator dependence but instead explicitly exposes it in an interpretable way.

6 Outlook

We have shown that quark-gluon taggers trained on low-level jet constituents, despite their complexity, rely on a small set of physically meaningful features. By examining the latent representations of a trained ParticleNet-Lite, we found that much of its performance can be attributed to a few directions, closely related to (i) jet multiplicity, (ii) radial energy flow, and (iii) fragmentation. These directions are not hard-coded into the network but learned, i.e. the training re-discovers the relevant physics and combines it with subtle additional structures.

Beyond confirming the established observables, our analysis suggests new, refined combinations of features that are not immediately obvious from theory. An example is r_λ to isolate radial jet structure while remaining decorrelated from multiplicity. Combinations involving fragmentation entropy or charge-related observables show how ParticleNet uses information not captured by the leading substructure observables. We address the question of how to utilize particle identification, a challenge for constituent-based taggers, through its entropy S_{PID}. It quantifies the diversity of particle types and is strongly correlated with one of the leading latent directions. This shows that the network leverages not just the presence of charged particles, but also the variety of particle types. Another interesting point is that the jet observables are almost exclusively linearly encoded, as suggested by our mutual information and copula study.

Our Shapley value analysis with the `SHAP` framework highlights both the potential and the limitations of feature attribution in jet tagging. While `SHAP` successfully identifies physically meaningful observables, its assumption of independent inputs leads to distorted attributions in the presence of correlations. Using decorrelated input features restores consistency with physics expectations. This underscores the importance of careful input preparation when applying `SHAP` to strongly correlated jet observables.

Finally, we employed symbolic regression using `PySR` to derive simple formulas for these observables that can accurately reproduce the network output. While formulas in terms of

only one observable follow the expected pattern, adding a second observable gives us valuable information about additional uncorrelated distinctive power. When allowed to use the seven leading observables, the learned formula only uses five and finds a good compromise between complexity and power. It almost matches the performance of the corresponding trained network and reflects the robustness across generators of the underlying network.

In the longer term, the compact formulas obtained through symbolic regression could be explored as fast surrogates for full ML taggers in experimental analyses. Their simple analytic form enables rapid evaluation on large-scale datasets and within environments where computational speed is a critical constraint. While they may not capture the full complexity of a network, such formulas could provide a practical compromise between performance and computational efficiency.

Our set of XAI tools allows us to understand trained precision networks without compromising the training objective or performance. The results show how ML classifiers or other trained networks can inspire theory by highlighting patterns that might otherwise be missed. For networks trained on data, this XAI methodology can connect data-driven methods to physically meaningful information. For applications, understanding the network output and the strong regularization of formulas will allow us to improve resilience and reduce systematic errors.

Acknowledgements

We would like to thank Björn Malte Schäfer, Rebecca Maria Kuntz and Benedikt Schosser for the helpful discussions about mutual information. We would like to thank the Baden-Württemberg-Stiftung for financing through the program *Internationale Spitzenforschung*, project *Uncertainties — Teaching AI its Limits* (BWST_IF2020-010). This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 – TRR 257 *Particle Physics Phenomenology after the Higgs Discovery*. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG (bwForCluster NEMO). SV is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research.

A Mutual information and copula diagnostics

We use mutual information and copulas to quantify how the latent principal components (PCs) relate to standard jet observables. Throughout we work with the cumulative distributions

$$U = F_X(X)\,, \qquad V = F_Y(Y)\,, \tag{37}$$

which are uniform on $[0,1]$. This is justified by the invariance of mutual information under strictly monotone reparameterization,

$$I(X;Y) = I\big(f(X);g(Y)\big) \tag{38}$$

for strictly increasing f and g. We flatten the one-dimensional spectra and analyze only the dependence structure between (U,V).

Sklar's theorem and Gaussian copulas

By Sklar's theorem any joint cumulative distribution function (CDF) $F_{X,Y}$ can be written as

$$F_{X,Y}(x,y) = C\big(F_X(x), F_Y(y)\big)\,, \tag{39}$$

where F_X and F_Y are the marginal CDFs and C is a copula, i.e. a bivariate CDF on $[0,1]^2$ with uniform marginals. The copula C contains all information about the dependence between X and Y, independent of the marginal shapes. As a baseline we use the Gaussian copula. Let (Z_1,Z_2) be jointly Gaussian with zero means, unit variances, and correlation matrix

$$\mathrm{P} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \tag{40}$$

and define $U = \Phi(Z_1)$ and $V = \Phi(Z_2)$, with Φ the standard normal CDF. Then (U,V) has a Gaussian copula with parameter $\rho \in (-1,1)$. We say that (X,Y) has a Gaussian copula if its uniformised marginals (U,V) can be obtained in this way. For a bivariate normal (X,Y) with correlation coefficient ρ the mutual information is

$$I(X;Y)_{\mathrm{Gauss}} = -\frac{1}{2}\log\big(1-\rho^2\big)\,. \tag{41}$$

For a Gaussian copula the mutual information is therefore completely fixed once ρ is known; any excess mutual information beyond this expression must come from non-Gaussian features (non-linear structure, heavy tails, asymmetries).

Copula families and tail dependence

To probe non-Gaussian structure we fit standard parametric copulas to pairs $(X,Y) = (\mathrm{PC}_i, \mathcal{O}_j)$, where PC_i are the first five principal components of the pooled latent space and $\mathcal{O}_j$ denotes a high-level jet observable such as n_{PF}, w_{PF}, S_{PID}, etc. Given a sample $\{(x_k,y_k)\}_{k=1}^n$, we first construct rank-based pseudo–observations

$$u_k = \frac{R_X(k)}{n+1}\,, \qquad v_k = \frac{R_Y(k)}{n+1}\,, \tag{42}$$

where $R_X(k)$ and $R_Y(k)$ are the empirical ranks of x_k and y_k. This yields approximately uniform marginals and isolates the copula. We fit the following bivariate copula families to $\{(u_k,v_k)\}$:

- Gaussian copula $C_{\text{Gauss}}(u, v; \rho)$;
- Student-t copula $C_t(u, v; \rho, \nu)$ (elliptical with power-law tails);
- Clayton copula (pronounced lower-tail dependence);
- Gumbel copula (pronounced upper-tail dependence);
- Frank copula (no asymptotic tail dependence, flexible in the bulk).

Parameters θ are obtained by maximising the copula log-likelihood

$$\ell(\theta) = \sum_{k=1}^{n} \log c_\theta(u_k, v_k)\,, \tag{43}$$

where c_θ is the copula density. We initialise from Kendall's rank correlation,

$$\tau = \text{Kendall}(U, V)\,, \tag{44}$$

which for elliptical copulas fixes the linear correlation ρ; for the Gaussian copula

$$\rho = \sin\left(\frac{\pi}{2}\tau\right)\,. \tag{45}$$

Model selection is based on the Akaike information criterion

$$\text{AIC} = -2\ell + 2k\,, \tag{46}$$

with the Bayesian information criterion

$$\text{BIC} = -2\ell + k \log n \tag{47}$$

used as a cross-check, where k is the number of free parameters. The copula family with the lowest AIC is taken as preferred. To quantify extreme correlations we extract the upper and lower tail-dependence coefficients

$$\lambda_U = \lim_{q\to 1^-} \mathbb{P}\big[V > q \,\big|\, U > q\big]\,, \tag{48}$$

$$\lambda_L = \lim_{q\to 0^+} \mathbb{P}\big[V \le q \,\big|\, U \le q\big]\,, \tag{49}$$

whenever the limits exist. Nonzero $\lambda_{U,L}$ means that very large (or very small) values of the two variables occur together more often than for weakly correlated Gaussian variables. For the bivariate Student-t copula with correlation ρ and ν degrees of freedom one has

$$\lambda_U = \lambda_L = 2\,T_{\nu+1}\left(-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}}\right), \tag{50}$$

where $T_{\nu+1}$ is the CDF of a univariate t-distribution with $\nu+1$ degrees of freedom. In contrast, for Gaussian and Frank copulas one finds $\lambda_L = \lambda_U = 0$.

Fit setup and observables

We apply this procedure to $n = 10^5$ jets for all pairs $(\text{PC}_i, \mathcal{O}_j)$ with $i = 1, \ldots, 5$ and

$$\{\mathcal{O}_j\} = \{n_{\text{PF}}, S_{\text{PID}}, w_{\text{PF}}, r_\lambda, S^{\perp}_{\text{frag}}, S_{\text{frag}}, \log p_T^D, C_{0.2}, C^{\perp}_{0.2}, E_Q\}. \tag{51}$$

For each pair we record:

- the best-fit copula family (AIC and BIC);

- Kendall's τ and Spearman's ρ_S;
- the copula parameters (e.g. ρ, ν, θ);
- the implied tail coefficients (λ_L, λ_U);
- and a non-parametric estimate of $I(X;Y)$ from (U,V), compared to the Gaussian baseline $I_{\text{Gauss}}(X;Y)$.

In the main text we use the mutual information only as a qualitative check: agreement with I_{Gauss} means that the dependence is effectively Gaussian at the copula level.

Results for latent PCs and observables

For the leading component PC_1 we find:

Multiplicity axis. The strongest correlation is with multiplicity For $(\text{PC}_1, n_{\text{PF}})$ the best copula is Gaussian, with

$$\tau(\text{PC}_1, n_{\text{PF}}) \simeq 0.82\,, \tag{52}$$

and $\lambda_L = \lambda_U = 0$. The mutual information is well described by the Gaussian prediction. This shows that PC_1 is essentially a monotonic multiplicity axis with an almost purely Gaussian dependence structure.

Fragmentation / PID observables. Observables such as S_{PID}, S_{frag}, and $C_{0.2}$ are also strongly correlated with PC_1:

$$\tau(\text{PC}_1, S_{\text{PID}}) \simeq 0.76, \quad \tau(\text{PC}_1, S) \simeq 0.70, \quad \tau(\text{PC}_1, C_{0.2}) \simeq 0.69\,. \tag{53}$$

The preferred copulas are Frank or Gaussian with $\lambda_L = \lambda_U = 0$, i.e. smooth, essentially Gaussian-like dependence without asymptotic tail enhancement.

Width and heavy tails. The jet width w_{PF} is special. For $(\text{PC}_1, w_{\text{PF}})$ the preferred family is a Student-t copula with

$$\tau(\text{PC}_1, w_{\text{PF}}) \simeq 0.63, \qquad \lambda_L \simeq \lambda_U \simeq 0.39, \qquad \nu \simeq 8\,. \tag{54}$$

The empirical mutual information significantly exceeds the Gaussian baseline, indicating a strong monotonic dependence with sizeable, symmetric tail dependence: very extreme values of PC_1 and w_{PF} co-occur much more often than in the Gaussian case. At first sight this looks like a genuine "width tail" direction in the latent space.

Small lower-tail effects. A few weakly correlated observables show Clayton behaviour with small but nonzero lower-tail coefficients, e.g.

$$\tau(\text{PC}_1, S_{\text{frag}}^{\perp}) \simeq 0.02, \quad \tau(\text{PC}_1, C_{0.2}^{\perp}) \simeq 0.01, \quad \tau(\text{PC}_1, \text{neg. energy frac}) \simeq 0.08, \tag{55}$$

with $\lambda_L \sim 0.03$–0.07 and $\lambda_U \simeq 0$. These encode a mild preference for extreme low-end configurations (e.g. unusually large negative energy contributions) but do not affect the bulk.

For the subleading components $\text{PC}_{2\text{–}5}$ the picture is much milder:

- A few pairs show moderate correlations, notably (PC_2, Λ) with $\tau \simeq 0.58$ (Gaussian copula, no tails) and $(\text{PC}_3, S_{\text{frag}}^{\perp})$ and $(\text{PC}_3, C_{0.2}^{\perp})$ with $\tau \simeq 0.48$–0.47 (Gaussian/Frank, no tails). These are again essentially Gaussian at the copula level.

- Most other pairs involving PC_{2-5} have $|\tau| \lesssim 0.2$. The best-fit copulas are predominantly Gaussian or Frank with $\lambda_{L,U} = 0$. In the few cases where AIC prefers Student-t or Clayton families, the tail coefficients are at the level $|\lambda_{L,U}| \lesssim \mathcal{O}(10^{-2})$.

Overall, PC_1 is cleanly identified as a monotonic multiplicity axis with mostly Gaussian dependence on standard observables. Higher PCs carry weaker and more specialised structures, largely without significant tail dependence.

Decorrelated width versus multiplicity

The heavy tails in $(\mathrm{PC}_1, w_{\mathrm{PF}})$ can be largely explained by the trivial statement that wider jets tend to have larger multiplicity. To isolate genuine width information we use a decorrelated width observable $w^{\perp}_{\mathrm{PF}}$, defined in the main text as a linear combination

$$w^{\perp}_{\mathrm{PF}} = \alpha\, n_{\mathrm{PF}} - w_{\mathrm{PF}}\,, \tag{56}$$

with α chosen such that $w^{\perp}_{\mathrm{PF}}$ is (approximately) uncorrelated with n_{PF}. This removes the leading multiplicity dependence while keeping sensitivity to the radial energy profile.

Repeating the copula analysis with $w^{\perp}_{\mathrm{PF}}$ (or, equivalently, a residualised width obtained from regressing w_{PF} on n_{PF}) we find that the best-fit copula is still Student-t, but the dependence on PC_1 is now very weak:

$$\tau(\mathrm{PC}_1, w^{\perp}_{\mathrm{PF}}) \simeq 0.06, \qquad \lambda_L \simeq \lambda_U \simeq 0.02\,. \tag{57}$$

The mutual information drops by about an order of magnitude and becomes close to the Gaussian prediction based on the same ρ.

We conclude that essentially all of the strong, heavy-tailed correlation between PC_1 and jet width is mediated by multiplicity. Once w_{PF} is decorrelated from n_{PF} (via $w^{\perp}_{\mathrm{PF}}$), the remaining dependence of PC_1 on width is small and only mildly non-Gaussian. In other words, the apparent "width tail" of PC_1 is really just the multiplicity tail seen through w_{PF}.

B Supplementary tables

PCA Combination	AUC	$Rej_{30\%}$	PCA Combination	AUC	$Rej_{30\%}$
PC_1, PC_2, PC_3, PC_4, PC_5	0.901	93.33	PC_2, PC_3, PC_4, PC_5	0.774	17.36
PC_1, PC_2, PC_4, PC_5	0.898	78.60	PC_2, PC_4, PC_5	0.733	10.67
PC_1, PC_2, PC_3, PC_5	0.898	87.84	PC_3, PC_4, PC_5	0.728	12.08
PC_1, PC_2, PC_5	0.896	86.15	PC_2, PC_3, PC_4	0.727	9.80
PC_1, PC_2, PC_3, PC_4	0.893	78.60	PC_2, PC_3, PC_5	0.718	10.87
PC_1, PC_3, PC_4, PC_5	0.893	81.45	PC_2, PC_4	0.686	7.49
PC_1, PC_2, PC_3	0.891	74.67	PC_2, PC_5	0.681	7.13
PC_1, PC_4, PC_5	0.891	73.44	PC_3, PC_4	0.676	7.79
PC_1, PC_2, PC_4	0.890	75.93	PC_2, PC_3	0.668	7.31
PC_1, PC_3, PC_5	0.889	78.60	PC_4, PC_5	0.665	7.70
PC_1, PC_2	0.888	80.00	PC_2	0.634	5.57
PC_1, PC_5	0.887	72.26	PC_3, PC_5	0.627	7.32
PC_1, PC_3, PC_4	0.886	75.93	PC_4	0.611	5.45
PC_1, PC_3	0.883	74.67	PC_3	0.575	5.42
PC_1, PC_4	0.882	73.44	PC_5	0.549	4.18
PC_1	0.879	74.67			

Table 12: Feedforward classifier trained on different PC combinations using the `Pythia` dataset, showing AUC and 30% quark rejection rate.

PCA Combination	AUC	$Rej_{30\%}$	PCA Combination	AUC	$Rej_{30\%}$
PC_1, PC_2, PC_3, PC_4, PC_5	0.831	47.19	PC_2, PC_3, PC_4, PC_5	0.715	11.49
PC_1, PC_2, PC_3, PC_5	0.831	47.16	PC_2, PC_3, PC_4	0.692	9.89
PC_1, PC_2, PC_3, PC_4	0.831	46.19	PC_2, PC_3, PC_5	0.692	9.91
PC_1, PC_2, PC_4, PC_5	0.831	44.36	PC_2, PC_3	0.667	8.89
PC_1, PC_2, PC_3	0.831	45.71	PC_2, PC_4, PC_5	0.654	7.87
PC_1, PC_2, PC_4	0.830	44.36	PC_2, PC_4	0.646	7.47
PC_1, PC_2, PC_5	0.830	45.71	PC_3, PC_4, PC_5	0.637	6.54
PC_1, PC_2	0.830	45.71	PC_3, PC_5	0.629	6.25
PC_1, PC_3, PC_4, PC_5	0.814	43.92	PC_3, PC_4	0.620	5.93
PC_1, PC_3, PC_4	0.814	45.71	PC_2, PC_5	0.615	6.35
PC_1, PC_3, PC_5	0.814	44.80	PC_3	0.611	5.43
PC_1, PC_3	0.813	44.36	PC_2	0.609	6.22
PC_1, PC_4, PC_5	0.812	49.23	PC_4, PC_5	0.553	4.44
PC_1, PC_5	0.811	47.66	PC_4	0.537	3.76
PC_1, PC_4	0.811	48.17	PC_5	0.530	3.90
PC_1	0.811	47.66			

Table 13: Feedforward classifier trained on different PC combinations using the `Herwig` dataset, showing AUC and 30% quark rejection rate

PCA Combination	AUC	$\text{Rej}_{30\%}$	PCA Combination	AUC	$\text{Rej}_{30\%}$
PC_1, PC_2, PC_3, PC_4, PC_5	0.829	41.87	PC_3, PC_4	0.805	43.08
PC_1, PC_2, PC_3, PC_4	0.828	43.08	PC_1, PC_2, PC_4	0.805	23.33
PC_1, PC_2, PC_3, PC_5	0.828	38.62	PC_1, PC_2	0.803	22.97
PC_2, PC_3, PC_4, PC_5	0.828	43.08	PC_2, PC_4	0.803	22.40
PC_2, PC_3, PC_5	0.828	40.00	PC_2	0.801	21.85
PC_1, PC_2, PC_3	0.827	39.65	PC_3, PC_5	0.801	34.20
PC_2, PC_3, PC_4	0.827	41.87	PC_1, PC_3	0.783	32.46
PC_2, PC_3	0.826	39.30	PC_3	0.780	32.70
PC_1, PC_3, PC_4, PC_5	0.816	42.26	PC_1, PC_4, PC_5	0.767	22.51
PC_3, PC_4, PC_5	0.816	42.67	PC_4, PC_5	0.764	20.55
PC_1, PC_2, PC_4, PC_5	0.816	33.43	PC_1, PC_4	0.759	17.78
PC_1, PC_3, PC_4	0.813	42.67	PC_4	0.756	17.85
PC_1, PC_2, PC_5	0.811	27.48	PC_1, PC_5	0.679	9.45
PC_2, PC_4, PC_5	0.810	28.00	PC_1	0.641	7.09
PC_2, PC_5	0.809	24.75	PC_5	0.563	4.93
PC_1, PC_3, PC_5	0.807	36.72			

Table 14: Feedforward classifier trained on different PC combinations using the `Pythia` PCA directions applied to the `Herwig` dataset, showing AUC and 30% quark rejection rate

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.083	0.500	3.33	-
4	$\tanh\left(\frac{0.78}{S_{\text{PID}}}\right)$	0.0043	0.594	6.19	12.75
5	$1.3-\tanh^2(S_{\text{PID}})$	0.0036	0.594	6.19	7.11
6	$(S_{\text{PID}}-1.6)^2+0.43$	0.00089	0.599	6.20	2.30
7	$\tanh\left((S_{\text{PID}}-1.63)^2\right)+0.42$	0.00056	0.599	6.19	1.94
8	$\tanh\left(\left((S_{\text{PID}}-1.61)^2+0.67\right)^2\right)$	0.00009	0.599	6.19	1.64
10	$0.99\cdot\tanh\left(\left((S_{\text{PID}}-1.61)^2+0.68\right)^2\right)$	0.00009	0.599	6.19	2.00

Table 15: 1D symbolic regression tables for S_{PID}

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.014	0.5	3.33	-
4	$4.3\cdot\left(\frac{1}{r_\lambda}\right)^{0.5}$	0.003	0.637	6.46	7.39
5	$0.78-0.0032\cdot r_\lambda$	0.0003	0.637	6.46	4.52
6	$0.85\cdot(1.0-0.0019\cdot r_\lambda)^3$	0.0001	0.637	6.46	3.00
7	$(0.92-\tanh(0.002\cdot r_\lambda))^2$	0.00011	0.637	6.46	2.80
8	$\tanh\left(\frac{114.82}{r_\lambda+36.44}\right)-0.23$	0.00009	0.637	6.46	2.31
9	$0.78\cdot\left((1.0-0.0051\cdot r_\lambda)^2+0.099\right)^{0.5}$	0.00005	0.637	6.46	2.03
10	$\left((0.59-\tanh(0.0038\cdot r_\lambda))^2\right)^{0.5}+0.22$	0.00005	0.637	6.46	1.81

Table 16: 1D symbolic regression tables for r_λ

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	$p_T D$	0.058	0.807	26.87	21.19
2	$p_T D^{0.5}$	0.038	0.807	26.87	15.52
3	$1.6 \cdot p_T D$	0.015	0.807	26.87	10.78
5	$\tanh\left(17.17 \cdot p_T D^3\right)$	0.0009	0.807	26.87	2.15
6	$\tanh^9\left(5.25 \cdot p_T D\right)$	0.0005	0.807	26.87	1.81
7	$0.92 \cdot \tanh\left(19.49 \cdot p_T D^3\right)$	0.00014	0.807	26.87	1.04
9	$0.92 \cdot \tanh\left(22.1 \cdot (p_T D - 0.01)^3\right)$	0.00011	0.807	26.87	0.87
10	$\tanh^2\left(21.33 \cdot p_T D^3 + 0.35\right) - 0.087$	0.00010	0.807	26.87	0.80

Table 17: 1D symbolic regression tables for $p_T D$

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.076	0.5	3.33	–
3	$0.97 - C_{0.2}$	0.033	0.793	58.41	15.35
4	$2.0 \cdot (1.0 - 0.79 \cdot C_{0.2})^3$	0.012	0.793	58.41	10.26
5	$2.8 \cdot (0.77 \cdot C_{0.2} - 1.0)^4$	0.010	0.793	58.41	8.09
6	$\tanh\left(19.66 \cdot (1 - 0.72 \cdot C_{0.2})^9\right)$	0.0042	0.793	58.41	5.78
7	$\tanh\left(\frac{0.026}{(0.23 - C_{0.2})^2}\right)$	0.0014	0.793	58.41	4.28
9	$\tanh\left(343.13 \cdot (0.72 \cdot C_{0.2} - 1)^{18} + 0.22\right)$	0.00040	0.793	58.41	1.54
10	$\tanh\left(88.98 \cdot \sqrt{(1 - 0.72 \cdot C_{0.2})^{27} + 6.72 \cdot 10^{-6}}\right)$	0.00038	0.793	58.41	1.72

Table 18: 1D symbolic regression tables for $C_{0.2}$

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.49	0.01183	0.5	3.33	-
5	$0.43 \cdot \left(\frac{1}{E_Q}\right)^{0.25}$	0.01108	0.483	4.15	10.45
6	$E_Q^3 - E_Q + 0.82$	0.00206	0.617	7.37	8.39
7	$E_Q^4 - E_Q + 0.9$	0.00060	0.621	7.66	6.01
8	$\tanh\left(E_Q^3 + 1.75 \cdot \left(1 - 0.83 \cdot E_Q\right)^3\right)$	0.00046	0.621	7.66	6.01
9	$1.2 \cdot E_Q^4 - 1.2 \cdot E_Q + 0.99$	0.00017	0.62	7.66	2.17
10	$\left(E_Q - 0.031\right)^3 + \tanh\left(1.76 \cdot \left(1 - 0.84 \cdot E_Q\right)^3\right)$	0.00005	0.62	7.65	2.57

Table 19: 1D symbolic regression tables for E_Q

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.081	0.5	3.33	7.34
4	$-\tanh\left(S_{\text{frag}} - 3.5\right)$	0.017	0.928	38.97	3.59
5	$-\tanh^3\left(S_{\text{frag}} - 3.9\right)$	0.0027	0.928	38.97	3.59
6	$\tanh^2\left(\frac{18.08}{S_{\text{frag}}^3}\right)$	0.0005	0.928	38.97	1.80
7	$\tanh\left(19.94 \cdot \left(0.2 - \frac{1}{S_{\text{frag}}}\right)^2\right)$	0.0005	0.928	38.97	1.50
8	$\left(\tanh\left(\frac{19.38}{S_{\text{frag}}^3}\right) - 0.026\right)^2$	0.00015	0.928	38.97	3.02
10	$\tanh\left(\frac{5.18}{1.71 \cdot \left(0.87 \cdot S_{\text{frag}} - 1\right)^4 + 2.8}\right)$	0.00009	0.928	38.88	0.93

Table 20: 1D symbolic regression tables for S_{frag}

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.51	0.08625	0.50	3.33	-
3	$0.93 - n_{\text{pf}}$	0.00832	0.840	52.32	11.04
4	$\left(1.2 - n_{\text{pf}}\right)^3$	0.00656	0.840	52.32	4.16
5	$\tanh\left(\frac{0.08}{n_{\text{pf}}^2}\right)$	0.00330	0.840	52.32	3.45
6	$\tanh^6\left(\frac{0.6}{n_{\text{pf}}}\right)$	0.00279	0.840	52.32	2.00
7	$\left(-n_{\text{pf}}^2 \cdot S_{\text{PID}} + 1.0\right)^3$	0.00047	0.848	62.94	4.11
8	$\tanh\left(0.06 \cdot \left(\frac{1}{S_{\text{PID}}}\right)^{3/2} n_{\text{pf}}^{-3}\right)$	0.00037	0.848	62.94	2.42
9	$\tanh\left(\frac{0.28}{n_{\text{pf}}^2 \cdot \left(n_{\text{pf}} + S_{\text{PID}}\right)^2}\right)$	0.00035	0.849	64.43	1.01
10	$\tanh\left(\frac{0.1 \cdot \left(\frac{1}{S_{\text{PID}}}\right)^{3/2}}{\left(n_{\text{pf}} + 0.09\right)^3}\right)$	0.00020	0.849	64.28	0.96
11	$0.96 \times \tanh\left(\frac{0.3}{n_{\text{pf}}^2 \times \left(n_{\text{pf}} + S_{\text{PID}}\right)^2}\right)$	0.00019	0.849	64.43	1.11
12	$0.96 \cdot \tanh\left(\frac{0.095 \cdot \left(\frac{1}{S_{\text{PID}}}\right)^{3/2}}{\left(n_{\text{pf}} + 0.06\right)^3}\right)$	0.00012	0.849	63.84	0.73
13	$0.96 \cdot \tanh^{1.5}\left(\frac{0.67}{n_{\text{pf}}^2 \cdot \left(S_{\text{PID}} + 0.94\right)^2}\right)$	0.00011	0.849	64.02	0.93
14	$0.96 \cdot \tanh^3\left(0.3 + \frac{0.65}{n_{\text{pf}}^2 \cdot \left(S_{\text{PID}} + 0.85\right)^2}\right)$	0.00009	0.849	64.29	0.83
17	$\left(-0.032 \cdot \sqrt{n_{\text{pf}}} + \tanh\left(0.32 + \frac{0.64}{n_{\text{pf}}^2 \cdot \left(S_{\text{PID}} + 0.82\right)^2}\right)\right)^3$	0.00008	0.849	63.94	0.85
18	$\left(-0.032 \times \tanh^{0.5}\left(n_{\text{pf}}\right) + \tanh\left(0.32 + \frac{0.64}{n_{\text{pf}}^2 \times \left(S_{\text{PID}} + 0.81\right)^2}\right)\right)^3$	0.00008	0.849	63.94	0.82
20	$\left(-0.032 \cdot \tanh^{0.5}\left(1.06 \cdot n_{\text{pf}}\right) + \tanh\left(0.32 + \frac{0.64}{n_{\text{pf}}^2 \times \left(S_{\text{PID}} + 0.82\right)^2}\right)\right)^3$	0.00007	0.849	60.85	0.86

Table 21: 2D symbolic regression tables for n_{pf} and S_{PID}

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	$p_T D$	0.07802	0.807	26.87	21.19
2	$p_T D^{0.5}$	0.05645	0.807	26.87	15.52
3	$0.93 - n_{\text{pf}}$	0.02289	0.840	52.32	11.05
4	$1.7 \cdot \left(1.0 - 0.84 \cdot n_{\text{pf}}\right)^3$	0.00592	0.840	52.32	4.13
5	$\tanh\left(\frac{0.08}{n_{\text{pf}}^2}\right)$	0.00406	0.840	52.32	3.42
6	$-\tanh\left(0.22 - \frac{p_T D}{n_{\text{pf}}}\right)$	0.00317	0.842	52.14	2.41
7	$\tanh\left(\frac{0.26 \cdot p_T D}{n_{\text{pf}}^2}\right)$	0.00136	0.844	55.01	1.02
8	$\tanh^3\left(\frac{p_T D^2 + 0.32}{n_{\text{pf}}}\right)$	0.00111	0.844	54.00	1.48
9	$\tanh\left(\left(p_T D - 0.27 + \frac{0.27}{n_{\text{pf}}}\right)^2\right)$	0.00093	0.845	54.70	1.39
10	$0.94 \cdot \tanh\left(\left(p_T D + \frac{0.068}{n_{\text{pf}}^2}\right)^2\right)$	0.00076	0.845	54.00	0.73
11	$\tanh^2\left(\left(p_T D + 0.35\right)^3 + \frac{0.08}{n_{\text{pf}}^2}\right)$	0.00070	0.845	54.17	1.18
12	$0.95 \cdot \tanh\left(4.65 \cdot p_T D^3 + \frac{0.03}{n_{\text{pf}}^3}\right)$	0.00044	0.845	53.94	0.64
13	$0.95 \cdot \tanh\left(\left(p_T D + 0.41\right)^6 + \frac{0.025}{n_{\text{pf}}^3}\right)$	0.00044	0.845	53.88	0.94
14	$0.95 \cdot \tanh\left(4.41 \cdot p_T D^3 + 0.01 + \frac{0.02}{n_{\text{pf}}^3}\right)$	0.00042	0.845	53.71	0.60
16	$0.95 \cdot \tanh\left(6.16 \cdot (0.21 - p_T D)^2 + 0.02 \cdot \left(0.49 + \frac{1}{n_{\text{pf}}}\right)^3\right)$	0.00037	0.846	54.23	0.58
17	$0.95 \cdot \tanh\left(5.28 \cdot p_T D^3 + \frac{0.02}{\left(n_{\text{pf}} - \frac{0.0005}{p_T D^3}\right)^3}\right)$	0.00031	0.846	53.13	1.07
18	$0.95 \cdot \tanh\left(0.019 \cdot \left(0.41 + \frac{1}{n_{\text{pf}}}\right)^3 + \left(p_T D \cdot \left(n_{\text{pf}} + 2.30\right) - 0.63\right)^2\right)$	0.00031	0.846	53.82	0.64
19	$\tanh\left(5.59 \cdot p_T D^3 + 4836.5 \cdot \left(\left(0.37 - p_T D\right)^3 + \frac{0.017}{n_{\text{pf}}}\right)^3\right) - 0.047$	0.00025	0.846	53.76	0.80
21	$0.95 \cdot \tanh\left(5.79 \cdot p_T D^3 + 0.03 \cdot \left(-0.23 + \frac{1}{n_{\text{pf}} + 9.96 \cdot (p_T D - 0.39)^3}\right)^3\right)$	0.00022	0.846	52.63	0.68

Table 22: 2D symbolic regression tables for n_{pf} and $p_T D$

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.50	0.10417	0.5	3.33	-
3	$0.94 - n_{\text{pf}}$	0.03430	0.840	52.32	11.12
4	$1.7 \cdot \left(1.0 - 0.84 \cdot n_{\text{pf}}\right)^3$	0.01592	0.840	52.32	3.90
5	$\tanh\left(\frac{0.03}{n_{\text{pf}}^3}\right)$	0.01307	0.840	52.32	2.90
6	$\tanh^6\left(\frac{0.57}{n_{\text{pf}}}\right)$	0.01262	0.840	52.32	2.05
7	$\tanh\left(\frac{0.04}{n_{\text{pf}}^3 + 0.01}\right)$	0.01226	0.840	52.32	1.15
8	$\tanh\left(\frac{0.31 \cdot \sqrt{\frac{1}{r_\lambda}}}{n_{\text{pf}}^3}\right)$	0.00612	0.852	46.04	2.04
9	$\tanh^2\left(\frac{1.28 \cdot \sqrt{\frac{1}{r_\lambda}}}{n_{\text{pf}}^2}\right)$	0.00470	0.852	41.25	2.37
10	$\tanh^9\left(\frac{189.7}{n_{\text{pf}} \cdot (r_\lambda + 198.6)}\right)$	0.00391	0.855	42.84	2.06
11	$\tanh^3\left(9.27 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.091}{n_{\text{pf}}}\right)^2\right)$	0.00337	0.856	43.52	1.90
12	$\tanh^9\left(7.32 \cdot \sqrt{\frac{1}{r_\lambda}} + \frac{0.13}{n_{\text{pf}}^2}\right)$	0.00238	0.858	46.00	1.47
13	$\tanh^9\left(\frac{123.8}{r_\lambda + 68.67} + \frac{0.13}{n_{\text{pf}}^2}\right)$	0.00190	0.859	48.59	1.64
14	$\tanh^{18}\left(\frac{236.17}{r_\lambda + 120.8} + \frac{0.13}{n_{\text{pf}}^2}\right)$	0.00184	0.859	49.36	1.68
15	$\left(1.1 - n_{\text{pf}}\right) \cdot \tanh\left(181.1 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.003}{n_{\text{pf}}^3}\right)^3\right)$	0.00115	0.860	51.81	1.41
16	$\left(1.1 - n_{\text{pf}}\right) \cdot \tanh^3\left(54.53 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.003}{n_{\text{pf}}^3}\right)^2\right)$	0.00083	0.860	51.81	1.43
17	$\left(1.1 - n_{\text{pf}}\right) \cdot \tanh^{1.5}\left(268.4 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.003}{n_{\text{pf}}^3}\right)^3\right)$	0.00082	0.860	51.81	1.77
18	$\tanh\left(\frac{0.36}{n_{\text{pf}}}\right) \cdot \tanh\left(464.46 \cdot \left(\sqrt{\frac{1}{r_\lambda}} - 0.05 + \frac{0.01}{n_{\text{pf}}^2}\right)^3\right)$	0.00078	0.860	51.81	1.59
19	$\tanh\left(\frac{0.36}{n_{\text{pf}}}\right) \cdot \tanh\left(1273.42 \cdot \left(0.01 \cdot \left(-0.60 + \frac{1}{n_{\text{pf}}}\right)^2 + \sqrt{\frac{1}{r_\lambda}}\right)^4\right)$	0.00076	0.860	51.81	1.51
20	$\tanh\left(\frac{0.36}{n_{\text{pf}}}\right) \cdot \tanh\left(464.25 \cdot \left(\sqrt{\frac{1}{r_\lambda}} - 0.04 + \frac{0.007}{\left(n_{\text{pf}} - 0.05\right)^2}\right)^3\right)$	0.00075	0.860	51.81	1.52
21	$\left(-n_{\text{pf}} - 0.59 \cdot \left(\frac{1}{r_\lambda}\right)^{0.5} + 1.2\right) \cdot \tanh\left(1472.86 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.0026}{n_{\text{pf}}^3}\right)^4\right)$	0.00069	0.860	54.41	1.60
22	$\left(-n_{\text{pf}} - 0.57 \cdot \left(\frac{1}{r_\lambda}\right)^{0.5} + 1.2\right) \cdot \left(\tanh^3\left(272.2 \cdot \left(\sqrt{\frac{1}{r_\lambda}} + \frac{0.003}{n_{\text{pf}}^3}\right)^3\right)\right)^{0.5}$	0.00065	0.860	54.76	1.46

Table 23: 2D symbolic regression tables for n_{pf} and r_λ

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.08389	0.500	3.33	-
3	$0.93 - n_{\text{pf}}$	0.01909	0.840	52.32	11.04
4	$1.7 \cdot \left(1.0 - 0.84 \cdot n_{\text{pf}}\right)^3$	0.00358	0.840	52.32	4.14
5	$\tanh\left(\frac{0.08}{n_{\text{pf}}^2}\right)$	0.00156	0.840	52.32	3.53
6	$\tanh^6\left(\frac{0.57}{n_{\text{pf}}}\right)$	0.00113	0.840	52.32	2.01
7	$\tanh\left(\frac{0.04}{n_{\text{pf}}^3+0.01}\right)$	0.00069	0.840	52.32	1.22
9	$0.93 \cdot \tanh\left(\frac{0.055}{\left(n_{\text{pf}}-0.086\right)^2}\right)$	0.00053	0.830	52.32	0.96
10	$\tanh\left(\frac{0.04}{n_{\text{pf}}^3+0.02\cdot\sqrt{E_Q}}\right)$	0.00047	0.840	52.36	1.29
11	$-0.098 \cdot E_Q + \tanh\left(0.11 + \frac{0.03}{n_{\text{pf}}^3}\right)$	0.00040	0.840	52.36	0.84
12	$0.94 \cdot \tanh\left(\frac{0.06}{\left(n_{\text{pf}}+0.08\cdot\sqrt{E_Q}\right)^3}\right)$	0.00032	0.840	52.41	0.86
13	$0.9 \cdot \tanh\left(\frac{0.04}{\left(n_{\text{pf}}+0.05\cdot E_Q\right)^3}\right) + 0.04$	0.00026	0.840	52.69	0.88
14	$0.92 \cdot \tanh\left(\frac{0.05}{\left(n_{\text{pf}}+0.06\cdot\sqrt{E_Q}\right)^3}\right) + 0.027$	0.00025	0.840	52.52	0.94
16	$0.18 - 0.76 \cdot \tanh\left(0.19 - \frac{0.06}{\left(n_{\text{pf}}+0.07\cdot\sqrt{E_Q}\right)^3}\right)$	0.00023	0.840	52.36	0.85
17	$0.32 - 0.62 \cdot \tanh\left(0.49 - \frac{0.098}{\left(n_{\text{pf}}+0.13\cdot\sqrt[4]{E_Q}\right)^3}\right)$	0.00022	0.840	52.52	0.87
18	$0.42 - 0.52 \cdot \tanh\left(0.92 - \frac{0.24}{\left(n_{\text{pf}}+0.23\cdot\sqrt[8]{E_Q}\right)^3}\right)$	0.00021	0.840	52.36	0.94
19	$0.18 - 0.76 \cdot \tanh\left(0.19 - \frac{0.06}{\left(n_{\text{pf}}+E_Q\cdot\left(0.18-0.11\cdot E_Q\right)\right)^3}\right)$	0.00021	0.840	52.69	0.92
20	$0.36 - 0.58 \cdot \tanh\left(0.62 - \frac{0.13}{\left(n_{\text{pf}}+0.32\cdot\sqrt{E_Q}-0.17\cdot E_Q\right)^3}\right)$	0.00020	0.840	52.52	0.92
21	$0.91 \cdot \tanh\left(\frac{0.05}{\left(n_{\text{pf}}+\frac{0.09\cdot E_Q}{\sqrt{\left(n_{\text{pf}}-E_Q\right)^2+0.58}}\right)^3}\right) + 0.034$	0.00017	0.840	52.08	0.99
22	$0.91 \cdot \tanh\left(\frac{0.05}{\left(n_{\text{pf}}+\frac{0.07\cdot E_Q}{\sqrt{\sqrt{\left(n_{\text{pf}}-E_Q\right)^2+0.10}}}\right)^3}\right) + 0.034$	0.00017	0.840	51.92	0.95

Table 24: 2D symbolic regression tables for n_{pf} and E_Q

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.49	0.09075	0.5	3.33	-
3	$0.92 - n_{\text{pf}}$	0.02469	0.840	52.32	11.35
4	$1.7 \cdot \left(1.0 - 0.84 \cdot n_{\text{pf}}\right)^3$	0.00795	0.840	52.32	4.25
5	$\tanh\left(\frac{0.082}{n_{\text{pf}}^2}\right)$	0.00612	0.840	52.32	3.59
6	$\tanh^6\left(\frac{0.56}{n_{\text{pf}}}\right)$	0.00532	0.840	52.32	2.05
7	$\tanh\left(\frac{0.06}{\left(n_{\text{pf}}+0.10\right)^3}\right)$	0.00510	0.840	52.32	1.49
8	$\tanh\left(\frac{0.10}{\left(n_{\text{pf}}+0.26\right)^4}\right)$	0.00506	0.840	52.32	1.49
9	$0.94 \cdot \tanh\left(0.02 \cdot \left(0.37 + \frac{1}{n_{\text{pf}}}\right)^3\right)$	0.00484	0.840	52.32	1.08
10	$\tanh\left(\frac{0.07}{\left(n_{\text{pf}}-(C_{0.2}-0.58)^2\right)^2}\right)$	0.00444	0.841	61.58	2.00
11	$\tanh^4\left(\frac{0.46}{-n_{\text{pf}}+(C_{0.2}-0.57)^2}\right)$	0.00417	0.841	60.46	1.32
12	$-\tanh\left(C_{0.2} - \frac{\sqrt{(C_{0.2}-0.46)^2}+0.35}{n_{\text{pf}}}\right)$	0.00265	0.846	62.81	2.43
13	$\tanh^2\left(C_{0.2} - \frac{\sqrt{(C_{0.2}-0.447)^2}+0.48}{n_{\text{pf}}}\right)$	0.00233	0.846	61.5	1.55
14	$\tanh^3\left(-C_{0.2} + \sqrt[4]{(C_{0.2}-0.45)^2} + \frac{0.53}{n_{\text{pf}}}\right)$	0.00199	0.846	59.03	1.34
15	$\tanh^2\left(-\sqrt{C_{0.2}} + \sqrt[4]{(C_{0.2}-0.45)^2} + \frac{0.53}{n_{\text{pf}}}\right)$	0.00181	0.846	59.03	1.36
16	$\tanh^6\left(131.67 \cdot \left((C_{0.2}-0.68)^2 - 0.05\right)^2 + \frac{0.52}{n_{\text{pf}}}\right)$	0.00139	0.847	64.02	1.88
17	$\tanh^3\left(-C_{0.2} + 3.88 \cdot \sqrt{\left((0.71 \cdot C_{0.2}-1)^4 - 0.21\right)^2} + \frac{0.54}{n_{\text{pf}}}\right)$	0.00122	0.848	62.34	1.61
18	$\left(0.03 - \tanh^3\left(\frac{82.47 \cdot \left((C_{0.2}-0.68)^2-0.04\right)^2+0.54}{n_{\text{pf}}}\right)\right)^2$	0.00089	0.848	65.27	1.30
19	$\left(\tanh\left(-C_{0.2} + 287 \cdot \left((0.70 \cdot C_{0.2}-1)^8 - 0.05\right)^2 + \frac{0.58}{n_{\text{pf}}}\right) - 0.019\right)^3$	0.00078	0.848	65.10	1.51
21	$\tanh^3\left(-C_{0.2} + 288.3 \cdot \left((0.70 \cdot C_{0.2}-1)^8 - 0.05\right)^2 + \frac{0.64}{n_{\text{pf}}+0.03}\right) - 0.05$	0.00074	0.848	65.19	1.33
22	$\left(\tanh^3\left(-C_{0.2} + 300.9 \cdot \left((0.70 \cdot C_{0.2}-1)^8 - 0.05\right)^2 + \frac{0.89}{n_{\text{pf}}+0.09}\right) - 0.027\right)^2$	0.00073	0.848	64.77	1.18

Table 25: 2D symbolic regression tables for n_{pf} and $C_{0.2}$

Complexity	Formula	Loss	AUC	$\text{Rej}_{30\%}$	ΔC
1	0.5	0.09448	0.5	3.33	
3	$0.93 - n_{\text{pf}}$	0.02494	0.840	52.32	11.07
4	$1.7 \cdot \left(1.0 - 0.84 \cdot n_{\text{pf}}\right)^3$	0.00669	0.840	52.32	4.06
5	$\tanh\left(\frac{0.03}{n_{\text{pf}}^3}\right)$	0.00372	0.840	52.32	2.78
6	$\tanh^6\left(\frac{0.57}{n_{\text{pf}}}\right)$	0.00327	0.840	52.32	2.06
7	$\tanh^3\left(\frac{1.18}{n_{\text{pf}} \cdot S_{\text{frag}}}\right)$	0.00151	0.844	54.35	1.74
9	$0.94 \cdot \tanh\left(\frac{0.69}{n_{\text{pf}}^2 \cdot S_{\text{frag}}^2}\right)$	0.00109	0.844	54.29	1.32
11	$0.95 \cdot \tanh^2\left(\frac{0.73}{S_{\text{frag}} \cdot \left(n_{\text{pf}} - 0.11\right)}\right)$	0.00099	0.844	55.80	1.03
12	$0.93 \cdot \tanh\left(1.89 \cdot \left(-\frac{1}{S_{\text{frag}}} - \frac{0.15}{n_{\text{pf}}}\right)^4\right)$	0.00096	0.844	55.80	1.33
13	$0.94 \cdot \tanh^2\left(\frac{0.57}{\left(n_{\text{pf}} - 0.13\right) \cdot \left(n_{\text{pf}} - S_{\text{frag}}\right)}\right)$	0.00092	0.844	55.74	0.93
14	$0.94 \cdot \tanh\left(0.96 \cdot \left(-0.06 - \frac{1}{S_{\text{frag}}} - \frac{0.18}{n_{\text{pf}}}\right)^4\right)$	0.00089	0.844	55.31	0.95
15	$0.95 \cdot \tanh^2\left(0.28 \cdot \left(-0.52 - \frac{1}{S_{\text{frag}}} - \frac{0.18}{n_{\text{pf}}}\right)^4\right)$	0.00086	0.844	55.37	0.94
16	$0.93 \cdot \tanh\left(\frac{\left(\left(n_{\text{pf}} - 0.17 \cdot S_{\text{frag}}\right)^2 + \frac{0.81}{S_{\text{frag}}}\right)^2}{n_{\text{pf}}^2}\right)$	0.00082	0.844	55.31	1.60
17	$0.94 \cdot \tanh\left(\frac{\left(\left(-n_{\text{pf}} + 0.059 \cdot S_{\text{frag}}^2\right)^2 + \frac{0.81}{S_{\text{frag}}}\right)^2}{n_{\text{pf}}^2}\right)$	0.00045	0.845	54.53	1.11
18	$0.94 \cdot \tanh\left(\frac{\left(\sqrt{\left(-n_{\text{pf}} + 0.06 \cdot S_{\text{frag}}^2\right)^2} + 0.77\right)^2}{n_{\text{pf}}^2 \cdot S_{\text{frag}}^2}\right)$	0.00040	0.845	53.59	1.15
19	$0.94 \cdot \tanh\left(\frac{4 \cdot \left(\left(n_{\text{pf}} - 0.06 \cdot S_{\text{frag}}^2\right)^2 + \frac{0.40}{S_{\text{frag}}}\right)^2}{n_{\text{pf}}^2}\right)$	0.00030	0.845	53.88	1.09
21	$0.93 \cdot \tanh\left(\frac{3.90 \cdot \left(\left(-n_{\text{pf}} + 0.06 \cdot S_{\text{frag}}^2\right)^2 + \frac{0.40}{S_{\text{frag}}}\right)^2}{n_{\text{pf}}^2}\right) + 0.0092$	0.00029	0.845	53.88	0.89
22	$0.94 \cdot \tanh^2\left(2.74 \cdot \left(0.19 + \frac{\left(-n_{\text{pf}} + 0.05 \cdot S_{\text{frag}}^2\right)^2 + \frac{0.40}{S_{\text{frag}}}}{n_{\text{pf}}}\right)^2\right)$	0.00027	0.845	53.88	1.03

Table 26: 2D symbolic regression tables for n_{pf} and S_{frag}

References

[1] T. Plehn, A. Butter, B. Dillon, T. Heimel, C. Krause, and R. Winterhalder, *Modern Machine Learning for LHC Physicists*, arXiv:2211.01421 [hep-ph].

[2] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, Phys. Rev. D **101** (2020) 5, 056019, arXiv:1902.08570 [hep-ph].

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

[4] H. Qu, C. Li, and S. Qian, *Particle Transformer for Jet Tagging*, arXiv:2202.03772 [hep-ph].

[5] M. He and D. Wang, *Quark/gluon discrimination and top tagging with dual attention transformer*, Eur. Phys. J. C **83** (2023) 12, 1116, arXiv:2307.04723 [hep-ph].

[6] Y. Wu, K. Wang, C. Li, H. Qu, and J. Zhu, *Jet tagging with more-interaction particle transformer**, Chin. Phys. C **49** (2025) 1, 013110, arXiv:2407.08682 [hep-ph].

[7] J. Brehmer, V. Bresó, P. de Haan, T. Plehn, H. Qu, J. Spinner, and J. Thaler, *A Lorentz-Equivariant Transformer for All of the LHC*, arXiv:2411.00446 [hep-ph].

[8] J. Spinner, L. Favaro, P. Lippmann, S. Pitz, G. Gerhartz, T. Plehn, and F. A. Hamprecht, *Lorentz Local Canonicalization: How to Make Any Network Lorentz-Equivariant*, arXiv:2505.20280 [stat.ML].

[9] W. Esmail, A. Hammad, and M. Nojiri, *IAFormer: Interaction-Aware Transformer network for collider data analysis*, arXiv:2505.03258 [hep-ph].

[10] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, JHEP **01** (2017) 110, arXiv:1612.01551 [hep-ph].

[11] T. Cheng, *Recursive Neural Networks in Quark/Gluon Tagging*, Comput. Softw. Big Sci. **2** (2018) 1, 3, arXiv:1711.02633 [hep-ph].

[12] G. Kasieczka, N. Kiefer, T. Plehn, and J. M. Thompson, *Quark-Gluon Tagging: Machine Learning vs Detector*, SciPost Phys. **6** (2019) 6, 069, arXiv:1812.09223 [hep-ph].

[13] J. S. H. Lee, S. M. Lee, Y. Lee, I. Park, I. J. Watson, and S. Yang, *Quark Gluon Jet Discrimination with Weakly Supervised Learning*, J. Korean Phys. Soc. **75** (2019) 9, 652, arXiv:2012.02540 [hep-ph].

[14] J. S. H. Lee, I. Park, I. J. Watson, and S. Yang, *Quark-Gluon Jet Discrimination Using Convolutional Neural Networks*, J. Korean Phys. Soc. **74** (2019) 3, 219, arXiv:2012.02531 [hep-ex].

[15] G. Kasieczka, S. Marzani, G. Soyez, and G. Stagnitto, *Towards Machine Learning Analytics for Jet Substructure*, JHEP **09** (2020) 195, arXiv:2007.04319 [hep-ph].

[16] A. Romero, D. Whiteson, M. Fenton, J. Collado, and P. Baldi, *Safety of Quark/Gluon Jet Classification*, arXiv:2103.09103 [hep-ph].

[17] F. A. Dreyer, G. Soyez, and A. Takacs, *Quarks and gluons in the Lund plane*, JHEP **08** (2022) 177, arXiv:2112.09140 [hep-ph].

[18] S. Bright-Thonney, I. Moult, B. Nachman, and S. Prestel, *Systematic quark/gluon identification with ratios of likelihoods*, JHEP **12** (2022) 021, arXiv:2207.12411 [hep-ph].

[19] A. Bogatskiy, T. Hoffman, D. W. Miller, J. T. Offermann, and X. Liu, *Explainable equivariant neural networks for particle physics: PELICAN*, JHEP **03** (2024) 113, arXiv:2307.16506 [hep-ph].

[20] D. Athanasakos, A. J. Larkoski, J. Mulligan, M. Płoskoń, and F. Ringer, *Is infrared-collinear safe information all you need for jet classification?*, JHEP **07** (2024) 257, arXiv:2305.08979 [hep-ph].

[21] W. Shen, D. Wang, and J. M. Yang, *Hierarchical high-point Energy Flow Network for jet tagging*, JHEP **09** (2023) 135, arXiv:2308.08300 [hep-ph].

[22] M. J. Dolan, J. Gargalionis, and A. Ore, *Quark-versus-gluon tagging in CMS Open Data with CWoLa and TopicFlow*, arXiv:2312.03434 [hep-ph].

[23] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, JHEP **05** (2017) 006, arXiv:1701.08784 [hep-ph].

[24] S. Macaluso and D. Shih, *Pulling Out All the Tops with Computer Vision and Deep Learning*, JHEP **10** (2018) 121, arXiv:1803.00107 [hep-ph].

[25] R. Sahu, S. Ashanujjaman, and K. Ghosh, *Unveiling the secrets of new physics through top quark tagging*, Eur. Phys. J. ST **233** (2024) 15-16, 2465, arXiv:2409.12085 [hep-ph].

[26] A. J. Larkoski, *Systematic Interpretability and the Likelihood for Boosted Top Quark Identification*, arXiv:2411.00104 [hep-ph].

[27] N. S. Woodward, S. E. Park, G. Grosso, J. Krupa, and P. Harris, *Product Manifold Machine Learning for Physics*, arXiv:2412.07033 [hep-ph].

[28] Y.-C. J. Chen, C.-W. Chiang, G. Cottin, and D. Shih, *Boosted W and Z tagging with jet charge and deep learning*, Phys. Rev. D **101** (2020) 5, 053001, arXiv:1908.08256 [hep-ph].

[29] CMS Collaboration, *Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques*, JINST **15** (2020) 06, P06005, arXiv:2004.08262 [hep-ex].

[30] T. Kim and A. Martin, *A $W^{\pm}$ polarization analyzer from Deep Neural Networks*, arXiv:2102.05124 [hep-ph].

[31] P. Baroň, J. Kvita, R. Přívara, J. Tomeček, and R. Vodák, *Application of Machine Learning Based Top Quark and W Jet Tagging to Hadronic Four-Top Final States Induced by SM as well as BSM Processes*, in *16th International Workshop on Top Quark Physics*. 10, 2023. arXiv:2310.13009 [hep-ex].

[32] R. Colyer and D. Duda, *Proposed measurement of longitudinally polarised vector bosons in WH and ZH production at Hadron colliders*, arXiv:2506.13002 [hep-ph].

[33] J. Li, P. Li, B. Long, and R. Zhang, *Jet Reconstruction with Mamba Networks in Collider Events*, arXiv:2506.18336 [hep-ph].

[34] S. Van Stroud, N. Pond, M. Hart, J. Barr, S. Rettie, G. Facini, and T. Scanlon, *Secondary vertex reconstruction with MaskFormers*, Eur. Phys. J. C **84** (2024) 10, 1020, arXiv:2312.12272 [hep-ex].

[35] H. Hassan, N. Mallick, and D. J. Kim, *Machine Learning-Based b-Jet Tagging in pp Collisions at* $\sqrt{s} = 13$ *TeV*, arXiv:2504.18291 [hep-ph].

[36] ATLAS Collaboration, *A continuous calibration of the ATLAS flavour-tagging classifiers via optimal transportation maps*, arXiv:2505.13063 [hep-ex].

[37] ATLAS Collaboration, *Transforming jet flavour tagging at ATLAS*, arXiv:2505.19689 [hep-ex].

[38] H. P. Nilles and K. H. Streng, *Quark - Gluon Separation in Three Jet Events*, Phys. Rev. D **23** (1981) 1944.

[39] J. Gallicchio and M. D. Schwartz, *Quark and Gluon Jet Substructure*, JHEP **04** (2013) 090, arXiv:1211.7038 [hep-ph].

[40] P. T. Komiske, E. M. Metodiev, and J. Thaler, *An operational definition of quark and gluon jets*, JHEP **11** (2018) 059, arXiv:1809.01140 [hep-ph].

[41] A. J. Larkoski and E. M. Metodiev, *A Theory of Quark vs. Gluon Discrimination*, JHEP **10** (2019) 014, arXiv:1906.01639 [hep-ph].

[42] P. Gras, S. Höche, D. Kar, A. Larkoski, L. Lönnblad, S. Plätzer, A. Siódmok, P. Skands, G. Soyez, and J. Thaler, *Systematics of quark/gluon tagging*, JHEP **07** (2017) 091, arXiv:1704.03878 [hep-ph].

[43] A. Butter, B. M. Dillon, T. Plehn, and L. Vogel, *Performance versus resilience in modern quark-gluon tagging*, SciPost Phys. Core **6** (2023) 085, arXiv:2212.10493 [hep-ph].

[44] C. Frye, A. J. Larkoski, J. Thaler, and K. Zhou, *Casimir Meets Poisson: Improved Quark/Gluon Discrimination with Counting Observables*, JHEP **09** (2017) 083, arXiv:1704.06266 [hep-ph].

[45] A. J. Larkoski, J. Thaler, and W. J. Waalewijn, *Gaining (Mutual) Information about Quark/Gluon Discrimination*, JHEP **11** (2014) 129, arXiv:1408.3122 [hep-ph].

[46] A. S. Cornell, W. Doorsamy, B. Fuks, G. Harmsen, and L. Mason, *Boosted decision trees in the era of new physics: a smuon analysis case study*, JHEP **04** (2022) 015, arXiv:2109.11815 [hep-ph].

[47] C. Grojean, A. Paul, Z. Qian, and I. Strümke, *Lessons on interpretable machine learning from particle physics*, Nature Rev. Phys. **4** (2022) 5, 284, arXiv:2203.08021 [hep-ph].

[48] B. Bhattacherjee, C. Bose, A. Chakraborty, and R. Sengupta, *Boosted top tagging and its interpretation using Shapley values*, Eur. Phys. J. Plus **139** (2024) 12, 1131, arXiv:2212.11606 [hep-ph].

[49] J. M. Munoz, I. Batatia, C. Ortner, and F. Romeo, *Retrieval of Boost Invariant Symbolic Observables via Feature Importance*, arXiv:2306.13496 [physics.comp-ph].

[50] A. Choudhury, A. Mondal, and S. Sarkar, *Searches for the BSM scenarios at the LHC using decision tree-based machine learning algorithms: a comparative study and review of random forest, AdaBoost, XGBoost and LightGBM frameworks*, Eur. Phys. J. ST **233** (2024) 15-16, 2425, arXiv:2405.06040 [hep-ph].

[51] S. Vilardi, S. Capozziello, and M. Brescia, *Discriminating between cosmological models using data-driven methods*, Astron. Astrophys. **695** (2025) A166, arXiv:2408.01563 [astro-ph.CO].

[52] R. Pezoa, L. Salinas, and C. Torres, *Explainability of High Energy Physics events classification using SHAP*, J. Phys. Conf. Ser. **2438** (2023) 1, 012082.

[53] A. Butter, T. Plehn, N. Soybelman, and J. Brehmer, *Back to the formula - LHC edition*, SciPost Phys. **16** (2024) 1, 037, arXiv:2109.10414 [hep-ph].

[54] Z. Zhang, R. Ma, J. Hu, and Q. Wang, *Approach the Gell-Mann-Okubo Formula with Machine Learning*, Chin. Phys. Lett. **39** (2022) 11, 111201, arXiv:2208.03165 [hep-ph].

[55] M. Cranmer, *Interpretable machine learning for science with pysr and symbolicregression.jl*, arXiv:2305.01582 [astro-ph.IM].

[56] N. Soybelman, C. Schiavi, F. A. Di Bello, and E. Gross, *Accelerating graph-based tracking tasks with symbolic regression*, Mach. Learn. Sci. Tech. **5** (2024) 4, 045042, arXiv:2406.16752 [hep-ex].

[57] M. Morales-Alvarado, D. Conde, J. Bendavid, V. Sanz, and M. Ubiali, *Symbolic regression for precision LHC physics*, in *38th conference on Neural Information Processing Systems*. 12, 2024. arXiv:2412.07839 [hep-ph].

[58] S. AbdusSalam, S. Abel, and M. Crispim Romão, *Symbolic regression for beyond the standard model physics*, Phys. Rev. D **111** (2025) 1, 015022, arXiv:2405.18471 [hep-ph].

[59] H. F. Tsoi, D. Rankin, C. Caillol, M. Cranmer, S. Dasu, J. Duarte, P. Harris, E. Lipeles, and V. Loncar, *SymbolFit: Automatic Parametric Modeling with Symbolic Regression*, Comput. Softw. Big Sci. **9** (2025) 1, 12, arXiv:2411.09851 [hep-ex].

[60] N. Makke and S. Chawla, *Inferring interpretable models of fragmentation functions using symbolic regression*, Mach. Learn. Sci. Tech. **6** (2025) 2, 025003, arXiv:2501.07123 [hep-ph].

[61] H. Bahl, E. Fuchs, M. Menen, and T. Plehn, *$\mathcal{CP}$-Analyses with Symbolic Regression*, arXiv:2507.05858 [hep-ph].

[62] G. Agarwal, L. Hay, I. Iashvili, B. Mannix, C. McLean, M. Morris, S. Rappoccio, and U. Schubert, *Explainable AI for ML jet taggers using expert variables and layerwise relevance propagation*, JHEP **05** (2021) 208, arXiv:2011.13466 [physics.data-an].

[63] M. S. Neubauer and A. Roy, *Explainable AI for High Energy Physics*, in *Snowmass 2021*. 6, 2022. arXiv:2206.06632 [hep-ex].

[64] A. Khot, M. S. Neubauer, and A. Roy, *A detailed study of interpretability of deep neural network based top taggers*, Mach. Learn. Sci. Tech. **4** (2023) 3, 035003, arXiv:2210.04371 [hep-ex].

[65] B. Kriesten, J. Gomprecht, and T. J. Hobbs, *Explainable AI classification for parton density theory*, JHEP **11** (2024) 007, arXiv:2407.03411 [hep-ph].

[66] K. Dimitrova, V. Kozhuharov, and P. Petkov, *Applicability evaluation of selected xAI methods for machine learning algorithms for signal parameters extraction*, J. Phys. Conf. Ser. **3002** (2025) 1, 012005, arXiv:2504.15670 [physics.comp-ph].

[67] M. Erdmann, N. Langner, J. Schulte, and D. Wirtz, *What exactly did the Transformer learn from our physics data?*, arXiv:2505.21042 [astro-ph.IM].

[68] Z.-Y. Song, T.-Z. Yang, Q.-H. Cao, M.-x. Luo, and H. X. Zhu, *Explainable AI-assisted Optimization for Feynman Integral Reduction*, arXiv:2502.09544 [hep-ph].

[69] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, *An introduction to PYTHIA 8.2*, Comput. Phys. Commun. **191** (2015) 159, arXiv:1410.3012 [hep-ph].

[70] P. T. Komiske, E. M. Metodiev, and J. Thaler, *Energy Flow Networks: Deep Sets for Particle Jets*, JHEP **01** (2019) 121, arXiv:1810.05165 [hep-ph].

[71] P. Komiske, E. Metodiev, and J. Thaler, *Pythia8 quark and gluon jets for energy flow*, Zenodo (5, 2019) .

[72] J. Bellm *et al.*, *Herwig 7.1 Release Note*, arXiv:1705.06919 [hep-ph].

[73] A. Pathak, P. Komiske, E. Metodiev, and M. Schwartz, *Herwig7.1 quark and gluon jets*, Zenodo (5, 2019) .

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

[75] M. Cacciari, G. P. Salam, and G. Soyez, *FastJet User Manual*, Eur. Phys. J. C **72** (2012) 1896, arXiv:1111.6097 [hep-ph].

[76] K. Pearson *et al.*, *Liii. on lines and planes of closest fit to systems of points in space*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2** (1901) 11, 559.

[77] J. Gallicchio, J. Huth, M. Kagan, M. D. Schwartz, K. Black, and B. Tweedie, *Multivariate discrimination and the Higgs + W/Z search*, JHEP **04** (2011) 069, arXiv:1010.3698 [hep-ph].

[78] J. Gallicchio and M. D. Schwartz, *Quark and Gluon Tagging at the LHC*, Phys. Rev. Lett. **107** (2011) 172001, arXiv:1106.3076 [hep-ph].

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

[80] P. T. Komiske, E. M. Metodiev, and J. Thaler, *Energy flow polynomials: A complete linear basis for jet substructure*, JHEP **04** (2018) 013, arXiv:1712.07124 [hep-ph].

[81] S. Brandt, C. Peyrou, R. Sosnowski, and A. Wroblewski, *The Principal axis of jets. An Attempt to analyze high-energy collisions as two-body processes*, Phys. Lett. **12** (1964) 57.

[82] J. R. Andersen *et al.*, *Les Houches 2015: Physics at TeV Colliders Standard Model Working Group Report*, in *9th Les Houches Workshop on Physics at TeV Colliders*. 5, 2016. arXiv:1605.04692 [hep-ph].

[83] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. Curran Associates Inc., 2017. arXiv:1705.07874 [cs.AI].

[84] L. S. Shapley, *Notes on the N-Person Game – II: The Value of an N-Person Game*. RAND Corporation, Santa Monica, CA, 1951.

[85] C. Calì and M. Longobardi, *Some mathematical properties of the roc curve and their applications*, Ricerche di Matematica **64** (2015) 2, 391.