Morphology for jet classification

Sung Hak Lim[1,*] and Mihoko M. Nojiri[2,3,4,†]

[1]*NHETC, Department of Physics and Astronomy, Rutgers University, Piscataway, New Jersey 08854, USA*
[2]*Theory Center, IPNS, KEK, 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan*
[3]*The Graduate University of Advanced Studies (Sokendai), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan*
[4]*Kavli IPMU (WPI), University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8583, Japan*

(Received 1 December 2020; accepted 2 December 2021; published 5 January 2022)

We introduce a jet tagger based on a neural network analyzing the Minkowski functionals (MFs) of pixelated jet images. The MFs are geometric measures of binary images, and they can be regarded as a generalization of the particle multiplicity, which is an important quantity in jet tagging. Their changes by dilation encode the jet constituents' geometric structures that appear at various angular scales. We explicitly show that this analysis using the MFs and dilation can be considered a constrained convolutional neural network (CNN). Conversely, CNN could model the MFs in the limit of a large network. We show an example that the CNN decision boundary correlates strongly with the value of MFs in semivisible jet tagging of a hidden valley scenario. The MFs are independent of the infrared and collinear (IRC)-safe observables commonly used in jet physics. We combine this morphological analysis with an IRC-safe relation network which models two-point energy correlations. While the resulting network uses constrained input parameters, it shows comparable dark jet and top jet tagging performances to the CNN. The architecture has significant computational advantages when the available data is limited. We show that its tagging performance is much better than that of the CNN with a small number of training samples. We also qualitatively discuss their parton shower model dependency. The results suggest that the MFs can be an efficient parametrization of the IRC-unsafe feature space of jets.

DOI: 10.1103/PhysRevD.105.014004

I. INTRODUCTION

The large hadron collider (LHC) has provided significant opportunities for new physics searches beyond the standard model. In the future extensions of the LHC, the sign of new physics may appear behind high p_T jets originating from the massive gauge bosons, top quarks, or Higgs bosons. Those boosted jets can be identified by examining jet substructures [1], and recently, there have been considerable efforts on using deep learning for tagging them [2–6].

The jet classification relies on substructures of jets from boosted massive particles. [1,7–12]. The quantification of those features may be performed with jet shape variables, such as n-subjettiness [13] or energy correlation functions [14]. In particular, these variables are often described by a set of n-point energy correlators [15,16], which is a basis of jet substructure variables with infrared and collinear (IRC) safety conditions.

On the other hand, counting variables, such as the number of charged tracks [17], are other discriminative variables in jet tagging. However, there is some subtlety in predicting them by QCD because they are not IRC safe. Those IRC unsafe features are often empirically modeled in event simulations. The predicted distribution often has a sizable deviation from the experimental data. We have to use them carefully so that classification models are not biased to particular simulators.

Meanwhile, these feature engineering that may be replaced with deep learning. For example, convolution-based networks [3,6] using (pixelated) particle distributions and recurrent neural networks [4,5] using a predefined sequence of particles are known for good jet tagging performance [18].

Those networks can represent a wide variety of functions, and they cover the high-dimensional phase space of inputs. However, some phase space of the training sample may be underrepresented by a finite number of samples, and the jet taggers based on them require high-quality samples to get the best performance. Because of that, it is often necessary to use dimensionality reductions, such as introducing bottlenecks in the middle of their architecture.

*sunghak.lim@rutgers.edu
†nojiri@post.kek.jp

However, those reduction techniques may not respect the physical constraints of the system, and explaining the outputs in domain-specific languages is less straightforward. Intensive postanalysis is often required in order to get an insight from the trained networks.

In this regard, starting from physics-inspired inputs and network architectures [19–23] has advantages over the general functional models in controllability and interpretability. For example, the energy flow network (EFN) [19,24] and the relation network (RN) [20,21,25,26] are known for their good tagging performance under the IRC-safe constraints [18,21]. If those constrained models cover all the relevant features for solving the given problem, the model will have equal performance compared to the general-purpose models [20,21]. So far, the networks covering IRC safe variables are well studied, but constrained models for IRC unsafe variables are not available yet. We need architectures bridging between general models and IRC unsafe variables.

Although deep learning models that systematically cover those IRC unsafe variables are not available, there are several frameworks based on multiplicities in coarse-graining [27], dilation and Minkowski functionals (MFs) [21], and Delaunay triangulation and its topology [28]. In this paper, we thoroughly reintroduce the approach in [21] in terms of the mathematical morphology and integral geometry, build a constrained model for the IRC unsafe variables, and show its analytic representation in the large network width limit.

This paper is organized as follows. In Sec. II, we introduce the morphological analysis on jet images using MFs, which is a generalization of counting variables by using its abstract algebraic features. We point out that the MFs can be represented by a chain of convolutions of the jet images and 2×2 filters, and therefore, convolutional neural networks (CNNs) can utilize it.

Section III reviews the two IRC-safe energy correlator-based networks which may provide complementary information to the MFs. In the case of jet image analysis, we show that the RN simplifies to a multilayer perceptron (MLP) taking a two-point energy correlation $S_2(R)$, which is an energy-weighted count of pairs of jet constituents at a given angular scale. On the other hand, the EFN is an MLP taking the jet image itself, where the jet image is an energy flow with a finite angular resolution.

In Sec. IV, we introduce a modular architecture combining morphological analysis and RN (or EFN). We simply combine outputs of each network using another MLP to get the final outputs. We compare the RN (or EFN) augmented with the morphological analysis against the baseline CNN.

Section V is devoted to the jet tagging performance between the combined setup using RN or EFN and the CNN. We consider two benchmark scenarios: tagging semi-visible jets [29] and top jets. Using the semivisible jet tagging example, we show that CNN can learn the distinctive feature of the MFs when the difference in the MF distributions between the signal and background is significant. Besides, our combined architectures and CNN augmented by the MFs show better performance than baseline CNN. This contradicts the observation that CNN can represent the MFs. These performance differences may be originated from the finite network size effects and regularization.

Section VI discusses the computational advantages of our constrained architecture compared to those of CNN. We show that the constrained architecture has better generalization performance when the number of training samples is small. We also point out that our setup is faster and memory-efficient because of lower computational complexity. In short, the MFs can efficiently represent IRC-unsafe information about the jet constituents.

Existing event simulation tools such as PYTHIA [30] and HERWIG [31,32] predict different soft particle distributions. Therefore, special care is needed to estimate the classification performance using simulated datasets. Section VII shows the generator dependence of jet constituent distributions in terms of MFs and describes the connection of qualitative features to the shower algorithms.

II. GENERALIZATION OF COUNTING VARIABLES IN JET PHYSICS

In order to generalize the counting variables, such as particle multiplicities, we need to introduce the mathematical concept called *valuation*. The particle multiplicities, which are essentially the number of elements in a set, have the following characteristic property for the union and intersection of two sets of particles, A and B,

$$n(A \cup B) = n(A) + n(B) - n(A \cap B). \tag{1}$$

This abstract mathematical feature is called valuation in measure theory. For example, the area of a region is a valuation. It would be worth exploring the space of valuations to generalize the counting variables, and MFs and Hadwiger's theorem are powerful tools for that.

A. Minkowski functionals and Hadwiger's theorem

The MFs of the jet constituents are the key characteristics for analyzing the space of valuations of jet substructures. Since we will analyze jet images on the pseudorapidity-polar coordinate plane, we will focus on discussing the MFs for two-dimensional Euclidean space $\mathbb{R}^2$. We also denote the coordinate vector as $\mathbf{R} = (\eta, \phi)$.

For a closed and bounded set S in $\mathbb{R}^2$, there are the three MFs: area A, boundary length L, and Euler characteristic χ. They can be expressed as the integral of local features of S as follows,

$$A = \int_S d^2\mathbf{R}, \qquad L = \int_{\partial S} d\mathbf{R}, \qquad \chi = \frac{1}{2\pi}\int_{\partial S} \kappa d\mathbf{R}, \quad (2)$$

where κ is the curvature of the boundary ∂S. The integral representation of the Euler characteristic is the Gauss-Bonnet theorem.

Hadwiger's theorem [33,34] states that these three functionals are the complete basis for the translation and rotation invariant valuations of convex bodies, where the convex body is a closed and bounded convex set with a nonempty interior. Let F be a function that satisfies the following properties,

(i) *Valuation*: for any two convex bodies B_i and B_j,

$$F(B_i \cup B_j) = F(B_i) + F(B_j) - F(B_i \cap B_j). \quad (3)$$

(ii) *Invariance*: for any translation or rotation g, the measure F is invariant, i.e., for any convex body B,

$$F(B) = F(gB). \quad (4)$$

(iii) *Continuity*: for any convergent sequence of convex bodies, $B_i \to B$,

$$\lim_{i\to\infty} F(B_i) = F(B). \quad (5)$$

Then for any F, there exist three constants c_0, c_1, and c_2 such that

$$F = \sum_{\nu=0,1,2} c_\nu \mathrm{MF}_\nu = c_0 A + c_1 L + c_2 \chi. \quad (6)$$

where MF_ν is (A, L, χ) for $\nu = (0, 1, 2)$, respectively.

Hadwiger's theorem also holds in the geometry of the square lattice and pixelated image, but the context should be modified accordingly [35]. The geometry of the square lattice has a different distance function called the L_1 distance, which is a sum of the absolute value of the difference in each component as follows.

$$\|\mathbf{R}_1 - \mathbf{R}_2\|_1 = |\eta_1 - \eta_1| + |\phi_1 - \phi_1|. \quad (7)$$

This distance is essentially identical to the length of the shortest path between two points on a square grid. The points within unit L_1 distance from the origin is different from those in Euclidean geometry. They form a square whose vertices are at (0,1), $(0,-1)$, (1,0), and $(-1,0)$.

The statements of Hadwiger's theorem still hold under this geometry, but there are two modifications. First, the invariance under translation and rotation is replaced by the isometry of the L_1 space. The convexity is replaced with L_1-convexity. A set B is L_1-convex if and only if there always exists a path connecting two points $\mathbf{R}_1$ and $\mathbf{R}_2$ in B, and the components of the path are monotonic along the path. L-shaped regions illustrate the difference between the two convexities. It is not convex but L_1-convex. After these modifications, we may safely use the MFs for the pixelated image analysis.

B. Morphological analysis on jet images

The morphological analysis on jet images is then performed on the filtered distribution of jet constituents projected on the pseudorapidity-polar coordinate (η, ϕ). We consider superlevel sets of the jet image, $P^{(0)}$, i.e., the set of pixels whose energy deposit $p_T^{(i,j)}$ is higher than the threshold value p_T [36],

$$P^{(0)}[p_T] = \{(i,j)|p_T^{(i,j)} > p_T\}, \quad (8)$$

where (i, j) is the integer coordinate of the given pixel.[1] The resulting binary images on a two-dimensional integer grid are used for morphological analysis. For the following discussion, we will omit the threshold argument $[p_T]$ unless explicitly required.

We then analyze the MFs of the images after dilation by a square called a structuring element to understand the geometric structure with the aid of mathematical morphology. The dilation helps to prove geometric features that are visible at the angular resolution of the size of the square. For our pixelated image analysis, the structuring element $B^{(k)}$ is a square with a side length $2k + 1$. The dilated image $P^{(k)}$ is defined as follows.

$$P^{(k)} = \{a + b|a \in P^{(0)}, b \in B^{(k)}\}, \quad (9)$$

$$B^{(k)} = \{(i,j)|i,j \in \{-k, -k+1, \ldots, k-1, k\}\}. \quad (10)$$

Sample binary images are in Fig. 1. The binary image $P^{(k)}$ is analogous to a coarse-graining or smearing of the original binary image $P^{(0)}$. We denote the three MFs of $P^{(k)}$ as $A^{(k)}$, $L^{(k)}$, and $\chi^{(k)}$. In [21], we have shown that the MFs $A^{(0)}$ and $A^{(1)}$ improve the top jet vs. QCD jet classification.

We also note that the dilation by a square is good enough for retrieving the topology of an underlying smooth body where the point clouds are sampled. The topology of the dilated image is generally sensitive to the structuring element, especially when we are using a finite number of samples. Still, the square is connected and sufficiently round so that the dilation by the square is a good topology estimation process without any glitches [37].

We can get some intuitive idea of how the sequences of the MFs encode the geometric information of a given image by considering its limiting behavior. For a scale k much

[1]The physical unit length of the grid is the hadronic calorimeter resolution $\Delta R = 0.1$ of our analysis. The physical coordinates (η, ϕ) are obtained by multiplying ΔR to those integer coordinates.

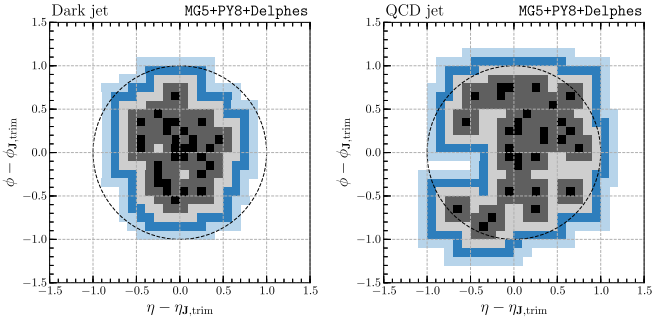


FIG. 1. Binary jet images of (a) a dark jet and (b) a QCD jet. Black dots are the active pixels in $P^{(0)}$ without any filtering. Dark gray, gray, blue, and light blue pixels are $P^{(i)} - P^{(i-1)}$ for $i = 1, 2, 3, 4$, respectively. Both of the binary images have $A^{(0)} = 30$. The dark jet model is described in Sec. V.

larger than the size of the image, $A^{(k)} \to (2k+1)^2$ because the details of the images are irrelevant to $P^{(k)}$. In the other extreme case that $P^{(k)}$ consists of N sufficiently isolated clusters, the asymptotic behavior changes to $A^{(k)} \to N(2k+1)^2$. Therefore, the sequence $A^{(k)}$ is sensitive to the number of clusters of active pixels in the jet image.

The intermediate behavior of the MF sequences ($A^{(k)}$, $L^{(k)}$, $\chi^{(k)}$) contains more details about the pixel distributions. When $P^{(k)}$ is a convex body, the MFs of $P^{(k)}$ and $P^{(k+1)}$ satisfy the following recurrence relation [35],[2]

$$A^{(k+1)} = A^{(k+1)}_{\mathrm{ext}} \equiv A^{(k)} + L^{(k)} + 4\chi^{(k)}, \tag{11}$$

$$L^{(k+1)} = L^{(k+1)}_{\mathrm{ext}} \equiv L^{(k)} + 8\chi^{(k)}, \tag{12}$$

$$\chi^{(k+1)} = \chi^{(k+1)}_{\mathrm{ext}} \equiv \chi^{(k)}. \tag{13}$$

The deviation from this relation signals that some change of the shape or topology occurs at the given angular scale. For example, the above recurrence relation does not hold if the dilation fills holes or dents. Therefore, the entire sequences of the MFs contain helpful information about the geometry of the binary image in general. This jet study using MFs is a persistent analysis of geometric features of jet substructures, similar to [28].

The recurrence relation also explains the asymptotic behavior of $A^{(k)}$. Suppose that the recurrence relations of the MFs hold after the given scale k_0. The solution for $A^{(k+k_0)}$ in terms of the MFs of $P^{(k_0)}$ is as follows.

$$A^{(k_0+k)}_{\mathrm{ext}} = A^{(k_0)} + kL^{(k_0)} + 4k^2\chi^{(k_0)}. \tag{14}$$

[2]The equation can be derived from the theorem 6.2 of [35], where the L_1-intrinsic volume $(V'_0, V'_1, V'_2) = (\chi, L/2, A)$ and the scale factor $\lambda = 2$

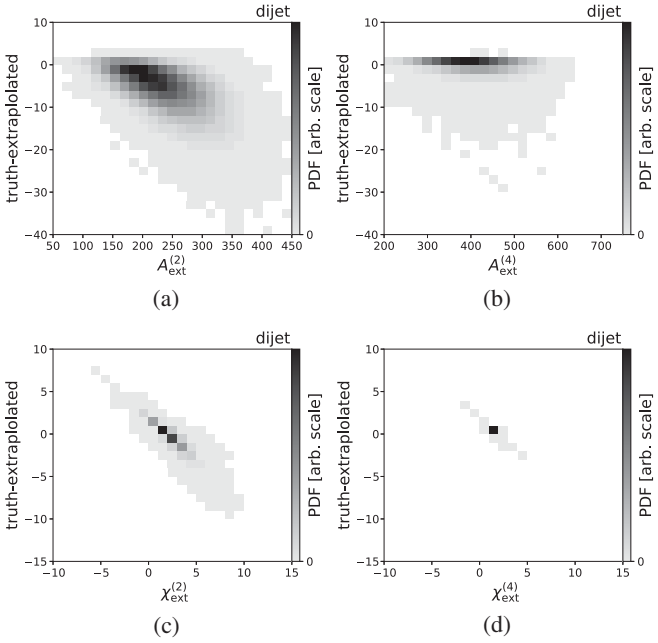


FIG. 2. The correlation between the MFs at a given scale k and its extrapolated values from the scale $k-1$. The horizontal axis is the extrapolated value, and the vertical axis is the difference between the true and extrapolated values. The upper plots (a) and (b) are for the area $A^{(k)}$, and the lower plots (c) and (d) are for the Euler characteristic $\chi^{(k)}$. The left plots (a) and (c) are for $k = 2$, and the right plots (b) and (d) are for $k = 4$. For $k = 4$, more samples have the difference zero since the dilation smooth out detailed features of jets, and its geometry and topology become more and more trivial.

For $k \gg k_0$, the area $A^{(k_0+k)}$ is approximately $4k^2\chi^{(k_0)}$, and the Euler characteristic $\chi^{(k_0)}$ can be interpreted as the number of clusters.

We now compare the area $A^{(k)}$ with the extrapolated area $A^{(k)}_{\mathrm{ext}}$ from the MFs of $P^{(k-1)}$ in order to check whether the dilation preserves the geometric features. The difference $\Delta A^{(k)}$ is a useful measure for checking the geometric persistence,

$$\Delta A^{(k)} = A^{(k)} - A^{(k)}_{\mathrm{ext}}. \tag{15}$$

Figure 2 shows 2D histograms of ($A^{(k)}_{\mathrm{ext}}$, $\Delta A^{(k)}$) of the leading p_T jets of QCD dijet events with $p_{T,\mathbf{J}} \in [500, 600]$ GeV. Figure 2(a) for $k = 2$ shows that typical jets have lots of vibrant activities at the short scale so that the condition $\Delta A^{(k)} = 0$ can be easily violated for a small k. Smeared images become more regular at a large scale so that many of the samples have $\Delta A^{(k)} = 0$, as shown in Fig. 2(b) for $k = 4$.

Similar behavior can be directly seen in the Euler characteristics. For a small k, the jets occasionally have subclusters, i.e., the intrinsic topology $\chi^{(k)}$ variate a lot.

Therefore, the extrapolation $\chi^{(k)}_{\rm ext} = \chi^{(k-1)}$ is also quite different from $\chi^{(k)}$, as shown in the 2D histogram of $(\chi^{(k-1)}, \chi^{(k)} - \chi^{(k-1)})$ in Fig. 2(c). For a large k, since we are analyzing a single jet, we expect most of the events to have $\chi^{(k-1)} \simeq \chi^{(k)} \simeq 1$, as in Fig. 2(d). Note that $\chi^{(k)} - \chi^{(k-1)}$ is positive for some events, indicating that there are holes at the scale $k-1$, and they are filled at the scale k.

Note that MFs are aggregated features, and their statistical fluctuations are smaller than the primitive inputs. For example, the number of active pixels $A^{(0)}$ has fluctuation $\delta A^{(0)}/A^{(0)} \sim 1/\sqrt{A^{(0)}}$, but its pixel-by-pixel fluctuation is of order 1. As a result, the training of RN with MFs is potentially more stable against the fluctuation of the energy deposit of pixels, while CNN is more susceptible to that.

Neural networks trained on these MFs utilize those geometric measures for solving the given task. The MFs do not use energy weighting in contrast to other energy-weighted IRC safe jet substructure observables so that all the jet constituents are treated with equal weight once they pass the p_T threshold.

C. Convolution representation of Minkowski functionals

The MFs are defined as an integral of local features in the continuum limit as in Eq. (2) so that they are a sum of all the local contributions from finite-sized patches. The feature tells us that MFs are embedded in the CNN using finite-size filters.

For example, the area of a two-dimensional region S can be written as a following double integral of an indicator function K of a square with side length ℓ and centered at (0,0).

$$A = \int_S d^2\mathbf{r} \int_{\mathbb{R}^2} d^2\mathbf{r}_0 \frac{1}{\ell^2} K_\ell(\mathbf{r} - \mathbf{r}_0) \tag{16}$$

$$K_\ell(x,y) = \begin{cases} 1 & x,y \in [-\frac{\ell}{2}, \frac{\ell}{2}] \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

By swapping the order of the integration, we obtain the expression in the form of the sum of the local contribution of finite patches,

$$A = \int_{\mathbb{R}^2} d^2\mathbf{r}_0 \left[\int_S d^2\mathbf{r} \frac{1}{\ell^2} K_\ell(\mathbf{r} - \mathbf{r}_0) \right]. \tag{18}$$

In order to discretize and evaluate this integral over the binary image on a square grid, the following marching square algorithm [38] is fast and useful.

The marching square algorithm for computing MFs first calculates and the local features of all 2×2 sub-binary images and collects them. The local features relevant for computing MFs are summarized in Table I. Note that these local features exclude those of the boundary of the subimages.

TABLE I. The list of local contributions to the Minkowski functionals. We also show the binary representation of the 2×2 subimages for each argument (Arg.) of $\vec{v}$ in Eq. (20). The first, second, third, and fourth digit of the four-digit binary representation of $P^{(k)}_{ij}$ is the value of upper left, upper right, lower left and lower right pixels, respectively.

		Local contributions					Local contributions		
Arg.	$P^{(k)}_{ij}$	A	L	χ	Arg.	$P^{(k)}_{ij}$	A	L	χ
0	0000	0	0	0	8	0001	1/4	1	1/4
1	1000	1/4	1	1/4	9	1001	1/2	2	−1/2
2	0100	1/4	1	1/4	10	0101	1/2	1	0
3	1100	1/2	1	0	11	1101	3/4	1	−1/4
4	0010	1/4	1	1/4	12	0011	1/2	1	0
5	1010	1/2	1	0	13	1011	3/4	1	−1/4
6	0110	1/2	2	−1/2	14	0111	3/4	1	−1/4
7	1110	3/4	1	−1/4	15	1111	1	0	0

(i) For the area A, a subimage contribution is $1/4$ of the number of its active pixels because a pixel belongs to four subimages.

(ii) For the boundary length L, the contribution is local boundary length divided by two since every boundary belongs to two subimages.

(iii) For the Euler characteristics χ, we only need to count the number of outward corners, $N_{\rm out}$, and inward corners, $N_{\rm in}$. Since inward and outward corners have exterior angles of $\pi/2$ and $-\pi/2$, respectively, the total curvature is just proportion to the difference between $N_{\rm out}$ and $N_{\rm in}$. The Euler characteristic is then as follows,

$$\chi = \frac{1}{2\pi}\left[\frac{\pi}{2}(N_{\rm out} - N_{\rm in})\right] = \frac{1}{4}(N_{\rm out} - N_{\rm in}). \tag{19}$$

Each corners are considered only once during the marching, the local contributions are $1/4$ for the outward corners and $-1/4$ for the inward corners.

Note that the Euler characteristic of the binary images depends on the definition of the connectivity between two diagonally neighboring pixels. We define that pixels sharing the same vertex are connected, and the corresponding subimages have two inward corners.

For example, (A, L, χ) of an isolated pixel is the sum of 1, 2, 4, and 8 of the Table I, and the value is (1,4,1). This algorithm can be generalized for calculating MFs of an image on other types of lattice, such as hexagonal pixels,[3]

[3]Note that the hexagonal grids are essentially identical to the plane of $\mathbb{R}^3$ with L_1 distance, with constraints $x + y + z = 0$ [39,40]. The hexagonal pixel is rounder, and it has bigger symmetry groups than the square pixels, but the integral geometry of hexagonal grids is not trivial. Nevertheless, Hadwiger's theorem in the $\mathbb{R}^3$ space still holds, and the nontrivial L_1-intrinsic volumes V'_1 and V'_2 are proportional to the perimeter and area of the hexagonal pixels.

or to approximate MFs of raw images without pixelation [41].

Since there are only 16 unique configurations for the 2×2 subimages, we may use the look-up table $\mathbf{v}^k(a)$, where $a = 0, \ldots, 15$ and $k \in \{A, L, \chi\}$, in Table I for parametrizing the local contribution. The MFs are then the sum of look-up table values as follows,

$$(A^{(k)}, L^{(k)}, \chi^{(k)}) = \sum_{i,j} \sum_{n,m\in\{0,1\}} \mathbf{v}(P^{(k)}_{(i+n)(j+m)} f_{nm}), \tag{20}$$

where $f_{nm} = ((1,2),(4,8))$, and $P^{(k)}_{ij}$ is 1 or 0 if (i,j)th pixel of $P^{(k)}$ is active or not, respectively.

Note that all the steps for calculating MFs in this section can be written in terms of convolutions. Let $p_T^{(i,j)}$ be the energy deposit of (i,j)th pixel. The calculation method of MFs discussed in this section can be summarized as follows.

$$\begin{aligned} P^{(0)}_{ij}[p_T] &= \theta(p_T^{(i,j)} - p_T), \\ P^{(k)} &= \theta(P^{(0)} * B^{(k)}), \\ (A^{(k)}, L^{(k)}, \chi^{(k)}) &= \mathbf{v}(P^{(k)} * f), \end{aligned} \tag{21}$$

where all the binary images in the above equations are functions that gives 1 for active pixels and 0 for otherwise, and $*$ is the discrete convolution. The stacked convolution layers can simulate this algorithm, i.e., $B^{(k)}$ and f can be considered as the weights of convolution layers, and the functions θ and $\mathbf{v}$ can be modeled by 1×1 convolutions [42]. Therefore, $A^{(k)}$, $L^{(k)}$, and $\chi^{(k)}$ are in principle covered by a CNN trained on jet images.

One subtle point is that this closed expression contains a step function that has a discontinuity point. The CNN with a finite number of filters and smooth activation functions may have difficulty accessing this variable set since the network is a smooth function. A similar situation may happen on the CNN with L_2 regularizers. We will show an example that the tagging performance of the CNN is improved by adding MFs to the inputs.

III. ENERGY CORRELATOR BASED NEURAL NETWORKS FOR JET SUBSTRUCTURE

The energy dependence of MFs in Eq. (21) is nonlinear, while many theory-motivated jet substructure variables typically have a multilinear energy dependence; these types of variables are called IRC safe energy correlators [15,16]. Since the counting variables complement those variables, we may use a neural network model representing the IRC-safe energy correlators and provide the MFs as additional inputs. In this section, we briefly review two examples: the IRC-safe relation network [20,21,43], and the EFN [19].

A. Relation network

The relation network (RN) is mainly designed for capturing the common properties of relational reasoning. For example, if we use the momentum p_i of the ith constituents of the jet as a network input, we can build one simplest model of RN with two scalar functions f and g as follows,

$$f\left[\sum_{i\in a, j\in b} g(p_i, p_j)\right], \tag{22}$$

where a and b are labels for subsets of jet constituents. If we impose the IRC-safe constraints [15,16], the function g should be bilinear in the constituent p_T and the coefficients Φ_{ab} should depend only on the relative angular distance between the jet constituents, R_{ij}. The following is then the basic form of the IRC-safe RN for the jet substructure,

$$f\left[\sum_{i\in a, j\in b} p_{T,i} p_{T,j} \Phi_{ab}(R_{ij})\right]. \tag{23}$$

The summation in the above equation is a nested loop over the jet constituents. Nevertheless, this part can be simplified to a single summation as we describe below.

We introduce the following two-point energy correlation $S_{2,ab}$ that accumulates energy correlations at a given angular scale R.

$$S_{2,ab}(R) = \sum_{i\in a, j\in b} p_{T,i} p_{T,j} \delta(R - R_{ij}). \tag{24}$$

By using $S_{2,ab}$, the nested summation in Eq. (23) can be replaced with a single integral as follows,

$$\int dR S_{2,ab}(R) \Phi_{ab}(R). \tag{25}$$

This model covers various jet substructure variables. For example, the two-point energy correlation function EFP^n_2 [14,16] can be written in a linear combination of the S_2 as follows,

$$\mathrm{EFP}^n_{2,ab} = \int_0^\infty dR S_{2,ab}(R) R^n, \tag{26}$$

Therefore, this network covers all information encoded in EFP^n_2.

For the practical use of this RN with IRC-safe constraints, we discretize the integral in Eq. (25) by binning the integrand with bin size ΔR. The discrete version of $S_{2,ab}$ is then defined as follows.

$$S^{(k)}_{2,ab} = \int_{k\Delta R}^{(k+1)\Delta R} dR S_{2,ab}(R), \tag{27}$$

where k is the bin index. The integral in Eq. (25) can be expressed as an inner product between $S_{2,ab}^{(k)}$ and a weight vector $\Phi_{ab}^{(k)}$,

$$\int dR S_{2,ab}(R)\Phi(R) = \sum_k S_{2,ab}^{(k)}\Phi_{ab}^{(k)}. \tag{28}$$

For our numerical study, we take bin size $\Delta R = 0.1$, which is the hadronic calorimeter resolution. We calculate S_2's from the HCAL and ECAL outputs.

If we use an MLP to model the function f of the RN in Eq. (23), we can embed $\Phi^{(k)}$ to the first fully-connected layer. The fully-connected layer that maps one input $\sum_k S_{2,ab}^{(k)}\Phi_{ab}^{(k)}$ to the latent dimension is equivalent to a fully connected layer that maps $S_{2,ab}^{(k)}$'s to the latent dimension, i.e.,

$$W_l \sum_k S_{2,ab}^{(k)}\Phi_{ab}^{(k)} = \sum_k W_{lk} S_{2,ab}^{(k)}, \qquad W_{lk} = W_l \Phi_{ab}^{(k)}. \tag{29}$$

The outputs of the first layer are then considered as *trainable two-point energy correlations*.

B. Energy flow network

EFN [19] is also a graph neural network based on the energy correlators, but this network uses only pointwise features. This network is based on the deep set architecture [24], i.e.,

$$f\left[\sum_{i\in a} g(p_i)\right]. \tag{30}$$

As discussed before, this pointwise feature $g(p_i)$ should be a linear function of energy when the IRC-safe constraint is assumed, and we have the following model of the EFN.

$$f\left[\sum_{i\in a} p_{T,i}\Phi(\mathbf{R}_i)\right]. \tag{31}$$

For the pixelated image analysis, the p_T-weighted sum over the jet constituents is replaced with the energy-weighted sum over all pixels,

$$\sum_{i\in a} p_{T,i}\Phi(\mathbf{R}_i) \approx \sum_{i,j} P_T^{(ij)}\Phi_{ij}, \tag{32}$$

where P_T^{ij} is the energy deposit of the (i,j)-th pixel, and Φ_{ij} is the corresponding angular weights.

If we use an MLP as f of Eq. (31), the product between the weights W_ℓ of the first dense layer and Φ_{ij} can be regarded as effective weights $W_{\ell ij}$ of an MLP with $P_T^{(ij)}$ as inputs. The dense layer can be rewritten as follows.

$$W_\ell\left[\sum_{i,j} P_T^{(ij)}\Phi_{ij}\right] = \sum_{i,j} P_T^{(ij)} W_{\ell ij}, \quad W_{\ell ij} = W_\ell \Phi_{ij}. \tag{33}$$

Therefore, an MLP for the pixelated image analysis models the EFN for the pixelated jet image.

Note that using the standardized inputs results does not change the structure of the model since the standardization is a linear transformation. Let us consider the following transformation of the inputs and parameters of the dense layer transforms,

$$P_T^{(ij)} \to \frac{P_T^{(ij)} - \mu^{(ij)}}{\sigma^{(ij)}}, \tag{34}$$

$$W_{\ell ij} \to \sigma^{(ij)} W_{\ell ij}, \tag{35}$$

$$B_\ell \to \sum_{i,j} \mu^{(ij)} W_{\ell ij} + B_\ell, \tag{36}$$

where $\mu^{(ij)}$ and $\sigma^{(ij)}$ are the mean and standard deviation of the inputs.[4] The first dense layer, $\sum_{i,j} P_T^{(ij)} W_{\ell ij} + B_\ell$, is invariant under this transformation. we may safely use the MLP for the standardized image to model the EFN.

IV. COMBINED NETWORK SETUP

In this section, we describe a network that combines morphological analysis and the RN or EFN.

A. Network inputs

For the morphological analysis, we use the MFs up to $k = 6$ and denote them as x_{morph},

$$x_{\text{morph}} = \bigcup_{p_T \text{ threshold}} \{A^{(k)}, L^{(k)}, \chi^{(k)} | k = 0, \ldots, 6\}. \tag{37}$$

We use the following p_T thresholds: 8, 4, and 2 GeV, and the default threshold of the detector simulation.[5]

We used the two-point energy correlation $S_{2,ab}$ of the following subsets of jet constituents for the IRC-safe relation network,

(i) the trimmed jet $\mathbf{J}_{\text{trim}}$ [9], denoted by h,
(ii) the compliment set of $\mathbf{J}_{\text{trim}}$, denoted by s,
(iii) the leading p_T subjet $\mathbf{J}_1$, denoted by 1,
(iv) the compliment set of $\mathbf{J}_1$, denoted by c.

Using these subsets is effective in the top tagging [21]. We use the following sets of binned two-point correlations as inputs of the RN,

[4] For the pixels which do not have energy variations, we assign $\sigma^{(ij)} = 1$.

[5] 0.5 GeV for the electronic calorimeters and 1.0 GeV for the hadronic calorimeters. This filtering is performed before the pixelation.

$$x_{\rm trim} = \{S^{(k)}_{2,hh}, S^{(k)}_{2,\rm soft} \equiv 2S^{(k)}_{2,hs} + S^{(k)}_{2,ss} | k = 0, \ldots, 14\},$$
$$x_{\mathbf{J}_1} = \{S^{(k)}_{2,11} | k = 0, 1, 2\} \cup \{S^{(k)}_{2,1c} | k = 0, \ldots, 9\}$$
$$\cup \{S^{(k)}_{2,cc} | k = 0, \ldots, 14\}. \quad (38)$$

In addition to those MFs and two-point energy correlations, we provide p_T and mass for each jet, trimmed jet, and leading p_T subjets as additional inputs to give information regarding jet kinematics, and we denote them as $x_{\rm kin}$.

$$x_{\rm kin} = \{p_{T,\mathbf{J}}, m_{\mathbf{J}}, p_{T,\mathbf{J}_{\rm trim}}, m_{\mathbf{J}_{\rm trim}}, p_{T,\mathbf{J}_1}, m_{\mathbf{J}_1}\}. \quad (39)$$

B. Network architecture

We use the following setup to transform the given inputs to the desired outputs for the binary classification. We first use MLPs to encode each of the primitive inputs $x_{\rm morph}$, $x_{\rm trim}$, and $x_{\mathbf{J}_1}$ into latent spaces of dimension 5,

$$h_{\rm morph} = {\rm MLP}_{\rm morph}(x_{\rm morph}, x_{\rm kin}), \quad (40)$$

$$h_{\rm trim} = {\rm MLP}_{\rm trim}(x_{\rm trim}, x_{\rm kin}), \quad (41)$$

$$h_{\mathbf{J}_1} = {\rm MLP}_{\mathbf{J}_1}(x_{\mathbf{J}_1}, x_{\rm kin}). \quad (42)$$

All the MLPs used in this section take the kinematic inputs $x_{\rm kin}$ as additional inputs. Those latent space features are mapped into the classifier outputs $\hat{y}$ by another MLP,

$${\rm logit}(\hat{y}) = {\rm MLP}_{\rm out}(h_{\rm morph}, h_{\rm trim}, h_{\mathbf{J}_1}, x_{\rm kin}), \quad (43)$$

where ${\rm logit}(\hat{y})$ is the inverse of the standard logistic function, $\log(\hat{y}) - \log(1-\hat{y})$. For the analysis using only the subset of the inputs, we take only the relevant latent space features. We denote this setup as RN + MF and the pure RN setup without morphological analysis as RN.

We will use this network for binary classifications, trained by minimizing the binary cross-entropy loss function.

$$\mathcal{L}_{\rm CE} = -\frac{1}{2}{\rm E}(\log\hat{y}|y=1) - \frac{1}{2}{\rm E}(\log(1-\hat{y})|y=0), \quad (44)$$

where $y = 1$ indicates the signal samples, and $y = 0$ indicates the background samples. The priors for each class are $1/2$. All the hidden layer weights are regularized by L2 regularization with a coefficient of 0.001. The network is trained by the ADAM optimizer [44] with default parameters, and we adopt the temporal exponential moving average on trainable parameters after ignoring the early 50 epochs. The ratios of training, validation, and test datasets are 9:1:10. We stop training when the validation loss does not improve for 50 epochs. We iterate this procedure for different numbers of mini-batches of 20, 50, 100, and 200, then choose the results with the largest validation AUC. All of these setups are implemented using KERAS [45] with TensorFlow backend [46]. Finally, all inputs are standardized, and we also reweight events to make the p_T distribution flat in order to marginalize learning from the $p_{T,\mathbf{J}}$ distribution.

We also remark that in a limit of large width of the MLPs and small bin size for S_2 and MFs, this network setup corresponds to the following smooth model,

$$h_{\rm MA} = \Psi_{\rm MA}\left[\int_0^\infty dp_T \int_0^\infty dR {\rm MF}_j(R; p_T)\Phi_j(R; p_T); x_{\rm kin}\right]$$
$$h_{\rm RN} = \Psi_{\rm RN}\left[\sum_{a,b}\int_0^\infty dR S_{2,ab}(R)\Phi(R); x_{\rm kin}\right]$$
$$\hat{y} = \Psi_{\rm out}[h_{\rm MA}, h_{\rm RN}; x_{\rm kin}], \quad (45)$$

where all the Φ and Ψ are some scalar functions. This expression can help discuss the relationship between the morphological analysis and other networks working on the momenta of jet constituents without pixelation, such as ParticleNet [6]. However, the discussion is beyond the scope of this paper.

C. Convolutional neural network and energy flow network

We compare this RN + MF to the following CNN and EFN.

Our baseline CNN is trained on the preprocessed jet images, as described in [21].

(1) The jet constituents are reclustered by k_T algorithm [47,48] with radius parameter 0.2.
(2) Set the center of the (η, ϕ) coordinate to be the leading p_T subjet axis.
(3) Rotate (η, ϕ) plane about the origin so that the subleading p_T subjet is on the positive y axis.
(4) If the third leading p_T subjet exists and has a negative x value, flip the x axis so that the third subjet is always on the right side of the image.
(5) Pixelate the jet constituents to get the jet image.

The preprocessed jet image is a two-dimensional p_T weighted histogram of jet constituents on a range $[-1.5, 1.5] \times [-1.5, 1.5]$ with bin size 0.1×0.1. We denote the set of energy deposits for each pixel as follows,

$$x_{\rm image} = \{P_T^{(ij)} | i, j = -15, \ldots, 14\}. \quad (46)$$

The image input $x_{\rm image}$ is provided to networks after standardization. In summary, the preprocessed images are aware of the most energetic subjet locations and the relative position of the two subleading p_T subjets.

The CNN consists of six convolutional layers. The filter size is 3×3, and a pooling layer with pool size 2×2 is inserted for every three convolutional layers. After then, all the spatial dimensions are flattened, and a 1×1 convolution maps the intermediate outputs to latent space with

TABLE II. The number of inputs N_{input}, the number of trainable parameters N_{param}. The number of inputs includes dummy inputs since each S_2's are saved on length 20 vectors. For EFNs, the number of parameters in parenthesis is the number for reduced setup with 50 energy correlators, while the nominal setup has 200 energy correlators.

	Inputs	N_{input}	N_{param}
MF	x_{morph}, x_{kin}	90	102,407
RN	x_{trim}, $x_{\mathbf{J}_1}$, x_{kin}	106	149,212
RN + MF	x_{morph}, x_{trim}, $x_{\mathbf{J}_1}$, x_{kin}	190	209,617
CNN	x_{image}, x_{kin}	906	131,740
CNN + MF	x_{image}, x_{morph}, x_{kin}	990	228,235
EFN	x_{image}, x_{kin}	906	202,167 (141,762)
EFN + MF	x_{image}, x_{morph}, x_{kin}	990	408,417 (348,012)

dimension 10.[6] These latent space features are then concatenated to the kinematic inputs x_{kin}, and we use an MLP to transform them into the desired classifier output. The training setups are the same as RN + MF, but we scan by minibatch numbers 100, 200, and 500.

Although CNN can represent MFs, we may explicitly provide the MFs to the CNN. As discussed earlier, CNN may experience technical difficulty expressing MFs through the training because the MFs are not smooth functions of the jet image. We additionally consider a CNN whose MLP at the end takes h_{morph} as additional latent space inputs. We denote this setup as CNN + MF.

We model the pointwise correlation of the EFN by an MLP with three hidden layers and ten outputs. The first hidden layer has 50 (200) outputs, while the others have 200 outputs. The input image is concatenated with x_{kin}. The outputs are then provided to another MLP that converts those inputs to the classifier, similar to that of the CNN.

Table II lists the combination of inputs studied in this paper and training costs for the classification problems discussed in Sec. VI. Some notable differences between the inputs to the CNNs and the RN + MFs are as follows.

The baseline CNN takes a large number of inputs since they are taking the whole image. However, the detector hits are sparsely distributed over the images since the center of the images contains more information while the outer region of the jet image has sparse soft activities. The CNN has to distill the useful information from this sparse dataset. On the other hand, RN only takes the basis for the two-point energy correlators. The soft activities are collected to each bin of S_2, and the resulting number of inputs is only $O[100]$.

The number of MF inputs is 3×7 for each binary image given energy thresholds. This is also a relatively small number compared to the dimension of the image inputs. We also note that as k increases, the change in geometry of the dilated image $P^{(k)}$ becomes more regular, and the MFs are getting dependent on their previous values in the sequences. The cutoff for k may be fine-tuned further, but we use 7, which effectively smoothes out geometric features below the angular scale of 1.5. The latter terms in the sequence merely validate the regularity in dilation, and dropping some of them may not affect the performance significantly.

[6]We have checked the classification performance of the CNNs with the latent dimensions 5, 10, 20, and 100, and 10 was the best.

V. JET TAGGING PERFORMANCE COMPARISON

A. Semi-visible jet tagging

As an application of our network, we consider a toy Hidden Valley model [49,50], in which a semivisible jet [29,51] is a characteristic signature. The hidden sector may include a fermion q_v charged under the secluded gauge group and a massive leptophobic gauge boson Z' that mediates the interaction between the SM particles and the hidden sector. At the hadron collider, q_v may be produced through the process $q\bar{q} \to Z' \to q_v\bar{q}_v$. The secluded gauge interaction confines q_v and $\bar{q}_v$ and forms pions π_v and rho mesons ρ_v after the hidden sector parton shower and hadronization. We consider a scenario that only ρ_v leaves visible signatures via the decay $\rho_v \to q\bar{q}$ while the other mesons are not visible at the detectors. The resulting semivisible jet, which we call a dark jet, contains many color-singlet quark pairs fragmenting into hadrons and missing particles. Therefore, the dark jets have different geometric structures and hard substructures compared to the QCD jets.

To simulate dark jets, we use PYTHIA 8 [30] and its Hidden Valley model implementation [50]. The mass spectrum is assigned as follows: $m_{Z'} = 1400$ GeV, $m_{q_v} = 10$ GeV, and $m_{\pi_v} = m_{\rho_v} = 20$ GeV. The fraction of π_v and ρ_v during the hadronization is 1:3, as the spin counting suggests. The QCD jet samples are the leading p_T jets of the process $pp \to 2j$, and they are generated using MadGraph5 2.6.6 [52] and PYTHIA 8. The detector effect is modeled by DELPHES 3.4.1 [53] with the default ATLAS detector card.

The training and test samples are the leading p_T jets with $p_{T,\mathbf{J}} \in [150, 300]$ GeV and $m_{\mathbf{J}} \in [30, 70]$ GeV. The number of selected events is 6.0×10^5 for the dark jet samples and 1.9×10^6 for the QCD jet samples.

Figure 3(a) shows the $A^{(k)}$ distributions of dark jets and QCD jets. The most left curve is the $A^{(0)}$ distributions, and they are close to each other. On the other hand, the average of $A^{(i)}$ $(i > 0)$ of the QCD jets is much larger, and the $A^{(i)}$ distribution extends far beyond the endpoint of the dark jet $A^{(i)}$ distribution. The RN + MF model can explicitly use the feature in the classification.

Given the apparent difference of $A^{(i)}$ distributions, the CNN is also capable of learning this phase space where only QCD jets exist. The classifier reasoning appears in the dijet distribution in Fig. 3(b). The distributions are after

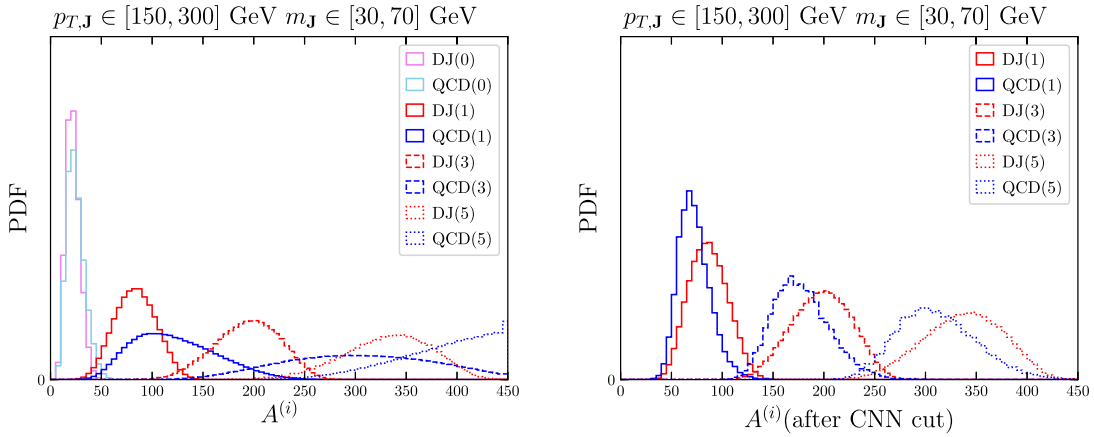


FIG. 3. Left: distributions of MFs: $A^{(0)}$ (light color), $A^{(1)}$ (solid), $A^{(3)}$ (dashed), and $A^{(5)}$ (dotted) of dark jets (red) and QCD jets (blue). We select leading p_T jets with criteria $p_{T,\mathbf{J}} \in [150, 300]$ GeV and $m_{\mathbf{J}} \in [30, 70]$ GeV. Right: distributions of MFs after rejecting 10% signal events by the CNN. 1.5% of QCD events remain after the selection.

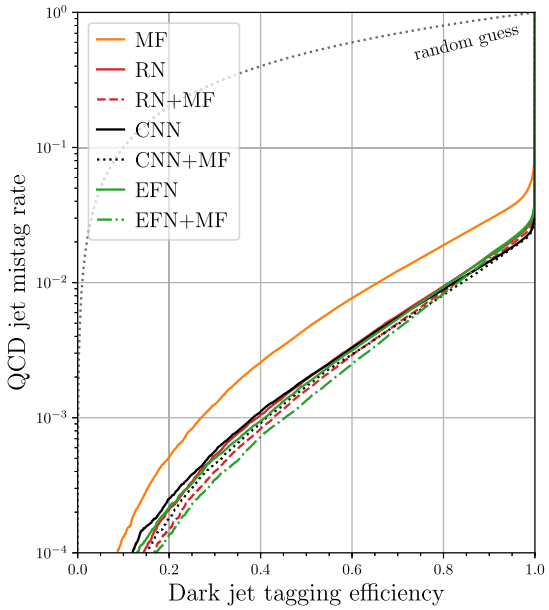


FIG. 4. ROC curves of various classification models for dark jets vs. QCD jets.

applying a loose selection of 90% dark jet signal efficiency using the CNN. The selection significantly suppresses events beyond the endpoint of the dark jet distribution.

We show the receiver operator characteristic (ROC) curves of RN and CNN[7] with and without the MFs on Fig. 4. The corresponding area under the ROC curve (AUC) in Table III. Both RN and CNN models reject more than 90% QCD jets on the phase space of large MFs without losing any dark jet events, as illustrated in Fig. 3. Even a simple classifier using only the MFs and kinematical variables rejects most of the QCD jet samples, as seen by the orange curve. This shows that the MFs describe the boundary of the phase space of the dark jet events quite efficiently. As you can see in Table III, the model with MFs consistently outperforms the model without MFs. The AUC of RN + MF is slightly better than CNN, and the AUC of CNN + MF is the best among the CNN and RN models.

The ROC curves show some crossovers in the region of small dark jet tagging efficiency below $\epsilon_{\text{dark}} = 0.6$, and RN + MF rejection efficiency looks better than CNN + MF in such regions. However, the rejection rate is high, and only O(1000) events support the learned features. Therefore, slight differences in the rejection efficiencies are not statistically significant.

We can estimate the difference between the CNN and RN + MF models by calculating the correlation coefficient of the logit outputs $\text{logit}(\hat{y}_{\text{CNN}})$ and $\text{logit}(\hat{y}_{\text{RN+MF}})$ for the same testing event set. We list the values in Table IV. Here $\hat{y}$ is the output vector of each model, and its logit is $\text{logit}(\hat{y}) = \log(\hat{y}) - \log(1 - \hat{y})$. The correlation coefficient

TABLE III. AUCs of various dark jet taggers. The EFN models have 200 hidden features at the first dense layer. We also show the training time t_{train} and the number of epochs at the end of the training, N_{train} for minibatch numbers $N_{\text{batch}} = 20$ and 200.

		$t_{\text{train}}/N_{\text{epoch}}$	
	AUC	$N_{\text{batch}} = 20$	$N_{\text{batch}} = 200$
MF	0.9897	793 s/564 epochs	954 s/363 epochs
RN	0.9950	929 s/434 epochs	2468 s/560 epochs
RN + MF	0.9955	1128 s/429 epochs	2288 s/556 epochs
CNN	0.9953		11401 s/327 epochs
CNN + MF	0.9956		19610 s/543 epochs
EFN	0.9950	2222 s/220 epochs	2141 s/163 epochs
EFN + MF	0.9955	1988 s/190 epochs	2270 s/172 epochs

[7]EFN results are explained in Sec. V C.

TABLE IV. The correlation coefficients of the logits of the model output, $\mathrm{logit}(\hat{y})$, between the trained models for the dark jet samples. The coefficients of the same models are the correlation coefficients of the outputs between the same networks trained with different random number seeds.

	MF	RN	RN + MF	CNN	CNN + MF	EFN	EFN + MF
MF	0.976	0.681	0.801	0.736	0.780	0.609	0.712
RN		0.942	0.868	0.745	0.732	0.705	0.723
RN + MF			0.973	0.793	0.839	0.679	0.777
CNN				0.958	0.924	0.763	0.809
CNN + MF					0.967	0.727	0.822
EFN						0.902	0.873
EFN + MF							0.933

ρ between CNN and RN + MF is relatively small, and $\rho = 0.793$ for the dark jet dataset. Nevertheless, once we give the MF information to the CNN model, the correlation improves, and $\rho = 0.893$ between CNN + MF and RN + MF. The improvement of correlation and classification performance indicates that the CNN is not fully utilizing those MFs unless explicitly provided as inputs.

The correlation coefficient between the network outputs trained with different random number seeds is significantly larger than the correlation between the different models. This indicates that the difference between the network outputs is primarily due to the systematic difference in the network architectures.

B. Top jet tagging

For the top jet study, we use the samples described in [21]. We use the events with $p_{T,\mathbf{J}} \in [500, 600]$ GeV and $m_{\mathbf{J}} \in [150, 200]$ GeV. The number of selected events is 9.5×10^5 for top jets and 3.5×10^5 for QCD jets. The training method and the ratio between training, validation, and test data samples are the same as the dark jet case.

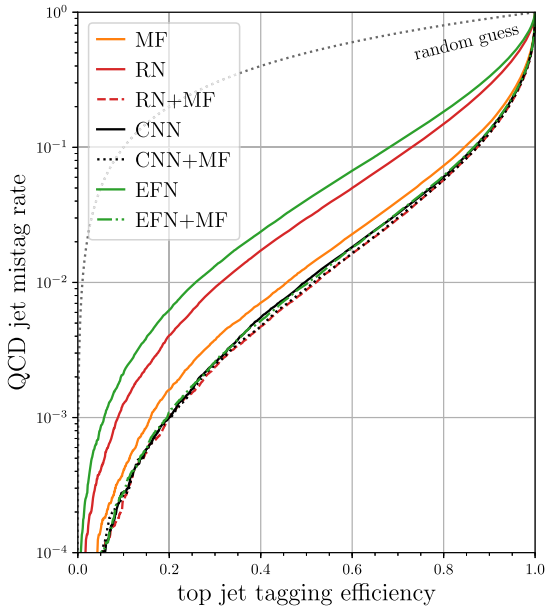


FIG. 5. ROC curves of various classification models for top jets vs. QCD jets.

We show the ROC curves in Fig. 5. The model MF, which uses only the MFs as inputs (without any IRC safe correlators), performs better than the RN model. This indicates that the geometric and topological information is the primary information for the top jet classification. As can be seen in Table V, the model using IRC safe variable and MFs is better than the one without MFs as the dark jet case. The MFs are enhancing the performance of the RN much more than the dark jet tagging case. The CNN + MF shows a similar tagging performance to the RN + MF, but the baseline CNN does not. As discussed earlier, the convolutional representation of the MFs contains a discontinuous step function. However, the step function is hard to be modeled by convolutional layers with a finite number of filters and L2 regularizers. This setup effectively penalizes functions with discontinuity because it requires large weights or a large number of filters with small weights.

The correlation coefficient ρ of the logit of outputs among the training of the same model with different random number seeds is 0.986 for RN + MF. On the other hand, the ρ of CNN is 0.933. The difference shows that the training of the CNN model suffers the local minimum problem relative to RN + MF. In gradient-based training methods, easily classifiable samples dominate the early

TABLE V. AUC of various top jet taggers. The EFN models have 50 hidden features at the first dense layer. We also show the training time t_{train} and the number of epochs at the end of the training, N_{train} for minibatch numbers $N_{\mathrm{batch}} = 20$ and 200.

		$t_{\mathrm{train}}/N_{\mathrm{epoch}}$	
	AUC	$N_{\mathrm{batch}} = 20$	$N_{\mathrm{batch}} = 200$
MF	0.9467	793 s/564 epochs	954 s/363 epochs
RN	0.9038	288 s/186 epochs	619 s/214 epochs
RN + MF	0.9552	418 s/255 epochs	1057 s/288 epochs
CNN	0.9529		31020 s/1483 epochs
CNN + MF	0.9547		12319 s/530 epochs
EFN	0.8900	535 s/120 epochs	723 s/108 epochs
EFN + MF	0.9521	725 s/149 epochs	813 s/111 epochs

TABLE VI. The correlation coefficients of the logit of outputs between the trained models for top jet samples. Diagonal elements are the correlation coefficients between the same networks trained with different random number seeds.

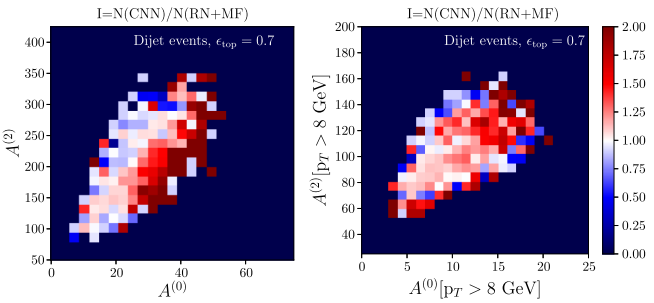
	MF	RN	RN + MF	CNN	CNN + MF	EFN	EFN + MF
MF	0.990	0.670	0.922	0.808	0.924	0.635	0.911
RN		0.978	0.778	0.738	0.730	0.847	0.714
RN + MF			0.986	0.847	0.941	0.711	0.931
CNN				0.933	0.866	0.739	0.849
CNN + MF					0.979	0.723	0.945
EFN						0.913	0.727
EFN + MF							0.960

phase of the training. The different training may show us different local minima that mainly describe the classification boundary for the dominant samples. In such cases, confusing events are underrepresented, and the training results will have some variance. This variance is larger for the more generic function model, such as the CNN model. Therefore, the CNNs have a smaller correlation coefficient than the RN + MFs. The other correlation coefficients are listed in Table VI.

The local minima problem of the CNN can be relaxed by explicitly providing MFs. Adding the MFs to CNN inputs improves the classification performance, and CNN + MF has a correlation coefficient of 0.979. Furthermore, the correlation between CNN + MF and RN + MF is 0.941, which is much higher than that between CNN and RN + MF. That is, the two models are now quite correlated to each other.

To visualize the subtle difference between the RN + MF and CNN, we compare the $(A^{(0)}, A^{(2)})$ distribution of dijet samples, conditioned on the classifier outputs. We select the dijet samples with classifier outputs $\hat{y}_{\mathrm{CNN}}$ and $\hat{y}_{\mathrm{RN+MF}}$ of CNN and RN + MF models less than its value at the 70% of top jet selection efficiency, respectively.

By taking the ratio of the histograms of the MFs, we can visualize the difference in classification boundaries of RN + MF and CNN. In Fig. 6, we consider the ratio



(a) (b)

FIG. 6. The PDF of the dijet event in the CNN model divided by the one in the RN + MF model with the same signal efficiency at $\epsilon_{\mathrm{top}} = 0.7$.

$$\mathcal{I} = \frac{N(\mathrm{CNN})}{N(\mathrm{RN} + \mathrm{MF}) + \epsilon}, \tag{47}$$

where N is the density at a given bin of the histogram of the samples selected by the CNN or RN + MF, and $\epsilon = 0.1$ is the regularization to avoid dividing by zero. Figure 6(a) is the distribution of $\mathcal{I}$ in $(A^{(0)}, A^{(2)})$ plane, and Fig. 6(b) is the same plot but for the MFs obtained from the pixels above the 8 GeV threshold, $(A^{(0)}[8\ \mathrm{GeV}], A^{(2)}[8\ \mathrm{GeV}])$.

Because the RN + MF model easily rejects dijet events, the ratios tend to be bigger than 1 for most of the bins. In the figure, the red bins represent $\mathcal{I} > 1$, while the blue bins correspond to $\mathcal{I} < 1$. For Fig. 6(a), the bins with large $A^{(0)}$ and small $A^{(2)}$ are red, indicating the RN + MF improves the classification by selecting more samples on this region. For Fig. 6(b), the region with large $A^{(0)}$ and large $A^{(2)}$ tend to have larger values, but the red region is less prominent. It may indicate that the CNN model utilized the geometric features of the pixels with energy above 8 GeV. On the other hand, the model has difficulty fully utilizing the geometric information of soft energy deposits.

C. Comments on EFN and EFN + MF

In addition to CNN, we study the classification using EFN and EFN + MF models. The EFN model uses the same jet images as inputs, but the model itself is constrained to be IRC safe. Because of the constraint, the EFN cannot fully use the geometric information of the soft activities encoded in the MFs. As a result, the classification performance of EFN is worse than that of the networks taking MFs as inputs and the CNN, which implicitly cover the MFs. Nevertheless, the EFN + MF works nearly as equal as the CNN + MF and RN + MF, and it covers sufficiently useful IRC safe information for both dark jet tagging and top jet tagging.

In dark jet tagging, the IRC safe variables are the critical information for jet tagging, and EFN performs well in the classification as illustrated in Fig. 4. Additional MFs inputs improve the performance slightly. At the low signal efficiency, the EFN + MF model has the best among all models in Fig. 3. As discussed already, due to the extensive

background rejection in the region, the number of the training sample is not enough, and we suspect that the difference is within the statistical fluctuations.

In the top tagging, the geometric and topological information is important. Sole EFN performs comparable to the RN, but it is significantly improved when MFs are additionally provided. Our RN model uses the two-point correlation to the leading p_T subjet and two-point correlation after removing the leading subjet to capture the three-point correlation inside the top jet. The inputs for the EFN are also sensitive to this topological three-prong structure of the top jet because we preprocess the jet images, and those three subjets always appear at particular points on the jet image. The EFN + MF covers more geometric information than the EFN, and its performance is comparable to the CNN. The improved performance mostly comes from the MFs, because the EFN + MF works nearly equal to the CNN + MF and RN + MF.

VI. COMPUTATIONAL ADVANTAGES OF MORPHOLOGICAL ANALYSIS AND RELATION NETWORK

A. Overcoming a small dataset

As discussed in the previous section, the RN + MF model has merits over the CNN model on better training performance. Models with broader coverage, such as CNN, are capable of modeling generic functions. The price of the high expressive power is often the high variance in the trained outputs and the high sensitivity to the statistical noise. These errors may degrade the generalization performance of the network. In this respect, using a simpler model helps to maintain the performance for some cases.

Figure 7 shows the AUCs of RN + MF and CNN as the functions of the number of training samples. We can see from the figure that the AUC of RN + MF is significantly larger than that of CNN for a small dataset, although their gap decreases as the size of the training dataset increases.

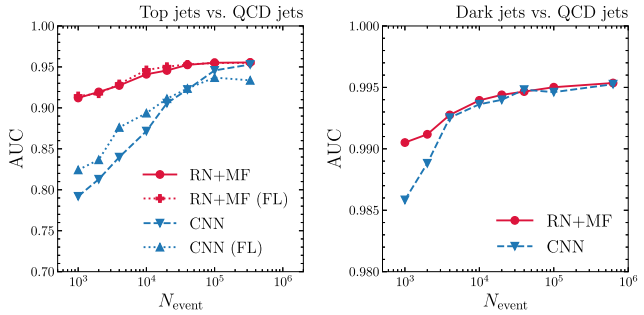


FIG. 7. The AUCs of RN + MF and CNN trained with a given number of training samples. N_{event} on the x-axis denotes the number of samples in each class. The rightmost entries are the AUCs of the networks trained on the full training dataset. Since the number of the signal and background samples are not identical in this case, we put their average value on the x-axis.

For the top jet classification, RN + MF achieves the AUC higher than 0.9 already at 1000 training samples, and the AUC is only 4% smaller at most than the best AUC. Meanwhile, CNN needs $\mathcal{O}[10,000]$ samples to achieve the same performance as RN + MF.

We find similar behavior of the AUC curves in the dark jet classification. The curves for RN + MF and CNN meet at 4000 events, which is much smaller than the meeting point of the curves in the top jet tagging case. It is because there are no dark jet samples at the tail of the MF distributions of QCD jets, as shown in Fig. 3. The training of the CNN could easily find this difference with a small number of samples, and the curves will meet much earlier.

Since the CNN model has comparable performance to RN + MF, we may consider optimizing learning steps to improve the performance when the dataset is small. For example, we may adjust learning dynamics by replacing the cross-entropy loss $\mathcal{L}_{\text{CE}}$ with a focal loss $\mathcal{L}_{\text{FL}}$ [54],

$$\mathcal{L}_{\text{FL}} = -\frac{1}{2}\mathrm{E}((1-\hat{y})^2 \log \hat{y}|y=1) - \frac{1}{2}\mathrm{E}((\hat{y})^2 \log(1-\hat{y})|y=0). \tag{48}$$

The results are shown in dotted lines in Fig. 7. The focal loss penalizes the contribution from easily classifiable examples by extra factors $(1-\hat{y})^2$ and $(\hat{y})^2$, and it helps training when the dataset is sparse. The jet image dataset is sparse so that we can see the improvement in the low statistics. However, there are no improvements to RN + MF since MF and S_2 distributions are mostly dense and smooth. Note that the training using focal loss does not converge to the maximum likelihood estimation of the binary classifier, i.e., $\hat{y} \nrightarrow p(y=1|x)$ in the asymptotic limit. Therefore, the performance is generally less than the one using the cross-entropy loss when enough data is available.

B. Less computational complexity and training time

Another advantage of the RN + MF is its low computation complexity. Networks with less computational complexity can be evaluated much faster and take less memory.

Tables III and V show that the training time of RN + MF is about ten times shorter than that of CNN. We also note that RN + MF takes about 300 MB GPU memory during the training with 200 mini-batches, while CNN takes about 6000 MB GPU memory in our setup.

We can estimate the computational complexity difference between CNN and RN + MF from the complexities of network evaluations and the input calculations. Because input calculations can be cached, the network evaluation complexity is the dominant factor in the training. The evaluation complexity is proportional to the number of multiplications since the networks mostly consist of tensor multiplications. One of the most expensive layers of our

CNN is a convolution layer with 3×3 filters mapping images with 30×30 pixels and 16 channels to the images of the same size. This layer has the following number of multiplications,

$$(3 \times 3) \times (16 \times 16) \times (30 \times 30) = 2,073,600. \quad (49)$$

Our CNN has two convolutional layers with this configuration, so that those two layers used about 4,000,000 multiplications.

Meanwhile, our RN + MF has only fully connected layers, and the most expensive one has 200 incoming and 200 outgoing features. This layer has $200 \times 200 = 40,000$ multiplications. We use three dense layers for each of the MLPs of RN + MF, which have four MLPs. Then the number of multiplications is at most

$$3 \times 4 \times 40,000 = 480,000. \quad (50)$$

The estimated computational complexity is factor 10 less than the convolutional layers. This qualitatively explains the difference in training time. More operations demand more GPU memory during the training since the back-propagation algorithm records the entire operations.

On the other hand, the complexity of input calculations only matters when the network inputs are not cached. The computational complexity of evaluating the inputs of RN + MF is as follows. The calculation of MFs has two convolutions with filter sizes $(2k+1) \times (2k+1)$ and 2×2 for the dilation and local feature identification, respectively. Those two convolutions have the number of multiplications,

$$(2k+1) \times (2k+1) \times (30 \times 30) + (2 \times 2) \times (30 \times 30), \quad (51)$$

which is 4,500 for $k = 0$ and 155,700 for $k = 6$. Note that the complexity of dilation, $(2k+1) \times (2k+1) \times (30 \times 30)$, can be further reduced by using optimized algorithms. We may consider this number as the upper bound of the complexity.

The calculation complexity of the two-point correlation $S_{2,ab}$ is a function of the number of jet constituents, N. The jet reclustering has $N \log N$ complexity [55], and the two-point correlation calculation has N^2 complexity in general. In the case of $N = 50$, which is approximately the largest number of jet constituents in our sample according to Fig. 3, the total complexity is $\approx 2,700$. The second N^2 factor can be reduced to $N^2/2$ if a and b of $S_{2,ab}$ are the same.

Those two complexities of evaluating the inputs of RN + MF, 155,700 and 2,700, are still much smaller than the complexity of the two convolutions layers. We conclude that the RN + MF setup is computationally efficient than the CNN.

VII. PARTON SHOWER MODELING AND MINKOWSKI FUNCTIONALS

So far, we have been discussing jets generated by PYTHIA8, but the simulated jet substructures have a simulator dependency in general because of different parton shower schemes. PYTHIA8 adopts p_T-ordered showering [56,57] while HERWIG7 adopts angular-ordered showering. The distributions of MFs with energy thresholds can capture the geometric differences between those two shower schemes, and the two simulated distributions may be different from each other. We quickly identify the difference in $A^{(k)}[p_T]$ distributions and discuss the origin of the difference in terms of the shower schemes.

In Fig. 8, we show the following asymmetry ratio $\mathcal{D}$ of the distribution of two selected $A^{(k)}[p_T]$.

$$\mathcal{D}(i) = \frac{f_P(i) - f_H(i)}{f_P(i) + f_H(i)}, \qquad f_A(i) = \frac{N_A(i)}{\sum_i N_A(i)} \quad \text{for } A \in \{P, H\}, \quad (52)$$

where $N_P(i)$ and $N_H(i)$ are the numbers of PYTHIA8 and HERWIG7 events in the ith bin, and $f_P(i)$ and $f_H(i)$ are their fractions with respect to the total number of events, respectively. Here, the samples are the QCD jets of

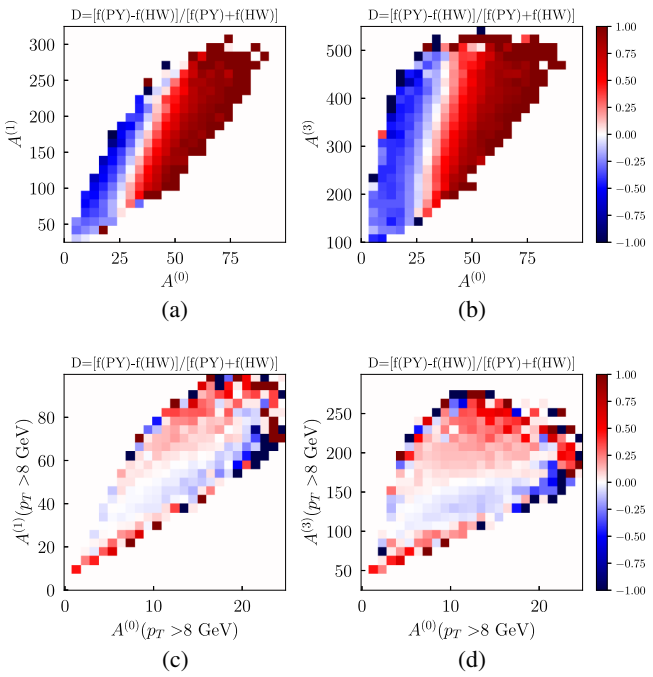


FIG. 8. The asymmetry $\mathcal{D}$ of the $(A^{(0)}, A^{(k)})$ distributions simulated by PYTHIA8 and HERWIG7. Figures (a) and (c) show the asymmetry of $(A^{(0)}, A^{(1)})$ distributions. Figures (b) and (d) show the asymmetry of $(A^{(0)}, A^{(3)})$ distributions. No p_T filter is applied to (a) and (b), while $p_T > 8$ GeV filter is applied for (c) and (d).

the top jet classification, with $p_{T,\mathbf{J}} \in [500, 600]$ GeV and $m_{\mathbf{J}} \in [150, 200]$ GeV.

In Figs. 8(a) and 8(b), we show the asymmetry ratio of $(A^{(0)}, A^{(1)})$ without p_T filters. The darkest red bins have $\mathcal{D} = 1$, where no HERWIG7 events are observed. The darkest blue region corresponds to $\mathcal{D} = -1$, and no PYTHIA8 samples are in there. The dark red pixels tend to be in the large $A^{(0)}$ region because PYTHIA8 predicts higher $A^{(0)}$. For the same $A^{(0)}$ value, PYTHIA8 predicts smaller values of $A^{(1)}$ than HERWIG7. This means the jet constituents are more clustered in PYTHIA8. The trend is common for all $k > 1$ (See Fig. 8(b) for $k = 3$.)

The situation is different for $A^{(k)}$ with p_T filter. As illustrated in Figs. 8(c) and 8(d), the $A^{(k)}[8\text{ GeV}]$ of PYTHIA8 tend to be higher than that of HERWIG7 for given $A^{(0)}[8\text{ GeV}]$. This means high p_T pixels are more sparsely distributed in PYTHIA8 generated samples.

Recall that PYTHIA8 adopts a transverse-momentum-ordered evolution scheme. High $p_\perp$ radiation in PYTHIA8 tends to be emitted at a larger angle. For the case of HERWIG7, the first emission in the evolution is typically a large angle soft radiation. The asymmetry $\mathcal{D}$ for $A^{(k)}[p_T]$ distributions is consistent with the expectation of the shower modeling. HERWIG7 QCD jet has soft particles at a large angle, while PYTHIA8 QCD jet has higher p_T objects at a large angle.

The distribution of inputs, especially the MFs, has to be tuned carefully to the real experimental data for the best classification performance with less simulator bias in the application stage, The calibration of MF distributions will be helpful to reduce the simulator dependency in the prediction of more general models, such as the CNN, because the MFs are important features in the jet classifications, as shown in Sec. V.

VIII. SUMMARY

In this paper, we introduce a neural network covering the space of "valuations" of jet constituents. The valuations introduced in this paper can be considered as a generalization of particle multiplicities which is a useful variable in quark vs. gluon jet tagging, but it is not IRC safe in general. The space of IRC unsafe variables is less explored than IRC safe variables because of its theoretical difficulties. Nevertheless, Hadwiger's theorem in integral geometry tells us some structure of the valuation space, which is an interest to this paper. The dimension of the valuation space is finite, and its basis is called the MFs. In the two-dimensional Euclidean space, the MFs are Euler characteristic, perimeter, and area. We utilized these geometric features to build a neural network covering the space of valuations, and the resulting network is a MLP taking the MFs as inputs.

We showed that a chain of convolutional layers could represent the MFs of dilated jet images. Therefore, CNNs can explicitly utilize this information. Indeed, in the semivisible jet tagging example, we showed that the CNN finds out the phase-space region of MFs where no signal exists. However, the MFs are not a smooth function of jet images, and the CNN with L_2 regularization had a problem accessing the complete information of the MFs. The classification performance improves by explicitly adding the MFs as inputs to the CNN.

We further build up a neural network architecture combining these valuations to the IRC safe information. We process the IRC safe information using energy correlator based networks: the relation network and the EFN. We combine the outputs from the IRC safe neural networks and the network using IRC unsafe MFs. The combined setup has comparable performance to the CNN.

The combined model is constrained compared to the CNN, but its classification performance is similar; moreover, it has computational advantages. First, it has a smaller computational complexity than the CNN, so that its evaluation is fast and less memory-demanding. Second, the constrained model requires a fewer number of training samples in order to reach its best performance. This network is beneficial when data is expensive.

Deep neural networks are a highly expressive function model, but their prediction is not explainable [58,59] in general. Suppose we are aware of potentially important features for modeling. In that case, we may distill the features [58,60] by using interpretable models built from the essential features in order to get an insight. It will also allow us to control the network predictions systematically by using domain-specific knowledge. Since MFs can be embedded in the CNN, they could potentially be interpreting variables of the CNN. In this paper, we built a network based on MFs, which have clear geometric interpretations. This type of network combined with interpretable IRC-safe neural networks [19,20] can improve the jet tagging performance and reduce the systematics further.

For example, the distributions of IRC unsafe variables, including the MFs, must be appropriately tuned to reduce the simulation bias. Tuning the distribution of jet constituents themselves for that purpose is not trivial because parton shower simulations are an approximation, and they do not fully cover the phase space of radiated particles. The use of the MFs provides a significantly compressed expression of the valuation space, which possesses information of both counting variables and the geometry in jet constituents. Therefore, tuning the distribution of MFs by reweighting [21,61] can be a more feasible method for controlling systematical errors of modeling the space of IRC unsafe features.

Although we limit our discussion to the pixelated image analysis, it would also be interesting to develop a continuum version of this morphological analysis in order to compare it with graph convolutional neural networks [6]. We will leave these exciting possibilities to future studies.

ACKNOWLEDGMENTS

The authors thank Benjamin Nachman, David Shih, Iftah Galon, Kyoungchul Kong, Mengchao Zhang, Myeonghun Park, and Takeshi Tsuboi for useful discussions. This work is supported by the Grant-in-Aid for Scientific Research on Scientific Research B (No. 16H03991, No. 17H02878) and Innovative Areas (No. 16H06492); World Premier International Research Center Initiative (WPI Initiative), MEXT, Japan. The work of S. H. L. was also supported by the US Department of Energy under Grant No. DE-SC0010008.

APPENDIX A: NETWORK CONFIGURATIONS

In the following, we show the hidden layer configurations of the networks studied in this paper. The activations of all the layers are ReLU, except the last dense layer, whose activation is linear.

1. Valuation model and relation network

We model the morphological analysis and the relation network (RN) by MLPs taking MFs and $S_{2,ab}$ as inputs, respectively. The configuration of the MLP is as follows,

(i) Concatenate inputs and $x_{\rm kin}$.
(ii) `Dense`: output size: 200.
(iii) `Dense`: output size: 200.
(iv) `Dense`: output size: 5.

`Dense` is a fully connected layer of given output size. Note that the first dense layer is essentially the model for the valuation or two-point energy correlations.

2. Convolutional Neural Network

The baseline CNN is modeled as follows.

(i) `Conv2D`: filter size: 3×3, 16 filters.
(ii) `Conv2D`: filter size: 3×3, 16 filters.
(iii) `Conv2D`: filter size: 3×3, 16 filters.
(iv) `MaxPooling2D`: pool size: 2×2.
(v) `Conv2D`: filter size: 3×3, 8 filters.
(vi) `Conv2D`: filter size: 3×3, 8 filters.
(vii) `Conv2D`: filter size: 3×3, 8 filters.
(viii) `MaxPooling2D`: pool size: 2×2.
(ix) `Dense`: output size: 200.
(x) `Dense`: output size: 10.

`Conv2D` is a convolutional layers and `MaxPooling2D` is a max pooling layer for two-dimensional pixelated images. Zero padding is used for calculating convolutions at the pixels near the boundary. We also showed that this configuration has a similar classification performance to the `ResNet` and `ResNeXt` within our setup and training samples [21].

3. Energy flow network

The EFN presented in this paper is essentially the MLP of jet images. However, 900 inputs are much larger than that of the MFs and $S_{2,ab}$, we compress the inputs to 50 (or 200) latent dimensions first.

(i) Concatenate inputs and $x_{\rm kin}$.
(ii) `Dense`: output size: 50 (or 200).
(iii) `Dense`: output size: 200.
(iv) `Dense`: output size: 200.
(v) `Dense`: output size: 10.

Again, the first dense layer is essentially the model for the linear energy correlators.

4. Multilayer perceptron classifier and logistic regression

The selected network outputs are then combined to the binary classifier, i.e., MLP followed by logistic regression.

(i) Concatenate all the inputs and $x_{\rm kin}$.
(ii) `Dense`: output size: 200.
(iii) `Dense`: output size: 200.
(iv) `Dense`: output size: 1.

The final output is trained by minimizing the cross-entropy loss in Eq. (44) or the focal loss in Eq. (48).

APPENDIX B: COMMENT ON SMOOTH ACTIVATION FUNCTIONS

In the previous paper [21], we compared the RN with $A^{(0)}$ and $A^{(1)}$ with CNN with ELU activation function and found that the performance is comparable, but this is accidental. As shown in Fig. 9, the performance of the

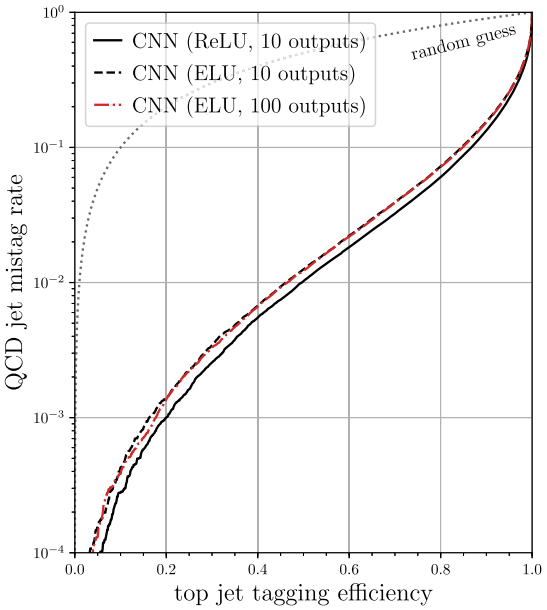


FIG. 9. ROC curves of CNNs for top jets vs. QCD jets. The solid black line is the baseline CNN with ReLU activation in this paper. Other CNNs use ELU activations. The red dot-dashed line is the ROC curve of the CNN in [21]. We also show the number of hidden outputs at the last dense layer of the CNN.

CNN with ReLU is better than the CNN with ELU [62] because ReLU is not a smooth function and can model the step function better. Nevertheless, the performance of RN + MF also improves after fully considering the MFs, and the performance is comparable with CNN with ReLU activation, as shown in the main text.

[1] Jonathan M. Butterworth, Adam R. Davison, Mathieu Rubin, and Gavin P. Salam, Jet Substructure as a New Higgs Search Channel at the LHC, Phys. Rev. Lett. **100**, 242001 (2008).

[2] Leandro G. Almeida, Mihailo Backović, Mathieu Cliche, Seung J. Lee, and Maxim Perelstein, Playing tag with ANN: Boosted top identification with pattern recognition, J. High Energy Phys. 07 (2015) 086.

[3] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman, Jet-images—deep learning edition, J. High Energy Phys. 07 (2016) 069.

[4] Gilles Louppe, Kyunghyun Cho, Cyril Becot, and Kyle Cranmer, QCD-aware recursive neural networks for jet physics, J. High Energy Phys. 01 (2019) 057.

[5] Taoli Cheng, Recursive neural networks in quark/gluon tagging, Comput. Softw. Big Sci. **2**, 3 (2018).

[6] Huilin Qu and Loukas Gouskos, Jet tagging via particle clouds, Phys. Rev. D **101**, 056019 (2020).

[7] Mrinal Dasgupta, Alessandro Fregoso, Simone Marzani, and Gavin P. Salam, Towards an understanding of jet substructure, J. High Energy Phys. 09 (2013) 029.

[8] Andrew J. Larkoski, Simone Marzani, Gregory Soyez, and Jesse Thaler, Soft drop, J. High Energy Phys. 05 (2014) 146.

[9] David Krohn, Jesse Thaler, and Lian-Tao Wang, Jet trimming, J. High Energy Phys. 02 (2010) 084.

[10] Stephen D. Ellis, Christopher K. Vermilion, and Jonathan R. Walsh, Techniques for improved heavy particle searches with jet substructure, Phys. Rev. D **80**, 051501 (2009).

[11] Stephen D. Ellis, Christopher K. Vermilion, and Jonathan R. Walsh, Recombination algorithms and jet substructure: Pruning as a tool for heavy particle searches, Phys. Rev. D **81**, 094023 (2010).

[12] Frédéric A. Dreyer, Lina Necib, Gregory Soyez, and Jesse Thaler, Recursive soft drop, J. High Energy Phys. 06 (2018) 093.

[13] Jesse Thaler and Ken Van Tilburg, Identifying boosted objects with N-subjettiness, J. High Energy Phys. 03 (2011) 015.

[14] Andrew J. Larkoski, Gavin P. Salam, and Jesse Thaler, Energy correlation functions for jet substructure, J. High Energy Phys. 06 (2013) 108.

[15] Fyodor V. Tkachov, Measuring multi-jet structure of hadronic energy flow or what is a jet?, Int. J. Mod. Phys. A **12**, 5411 (1997).

[16] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler, Energy flow polynomials: A complete linear basis for jet substructure, J. High Energy Phys. 04 (2018) 013.

[17] Jason Gallicchio and Matthew D. Schwartz, Quark and Gluon Tagging at the LHC, Phys. Rev. Lett. **107**, 172001 (2011).

[18] Gregor Kasieczka *et al.*, The machine learning landscape of top taggers, SciPost Phys. **7**, 014 (2019).

[19] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler, Energy flow networks: Deep sets for particle jets, J. High Energy Phys. 01 (2019) 121.

[20] Amit Chakraborty, Sung Hak Lim, and Mihoko M. Nojiri, Interpretable deep learning for two-prong jet classification with jet spectra, J. High Energy Phys. 07 (2019) 135.

[21] Amit Chakraborty, Sung Hak Lim, Mihoko M. Nojiri, and Michihisa Takeuchi, Neural network-based top tagger with two-point energy correlations and geometry of soft emissions, J. High Energy Phys. 07 (2020) 111.

[22] Anders Andreassen, Ilya Feige, Christopher Frye, and Matthew D. Schwartz, JUNIPR: A framework for unsupervised machine learning in particle physics, Eur. Phys. J. C **79**, 102 (2019).

[23] Anders Andreassen, Ilya Feige, Christopher Frye, and Matthew D. Schwartz, Binary JUNIPR: An Interpretable Probabilistic Model for Discrimination, Phys. Rev. Lett. **123**, 182001 (2019).

[24] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander J. Smola, Deep sets, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., New York, 2017), Vol. 30.

[25] David Raposo, Adam Santoro, David Barrett, Razvan Pascanu, Timothy Lillicrap, and Peter Battaglia, Discovering objects and their relations from entangled scene representations, arXiv:1702.05068.

[26] Adam Santoro, David Raposo, David G. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap, A simple neural network module for relational reasoning, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., New York, 2017), pp. 4967–4976.

[27] Joe Davighi and Philip Harris, Fractal based observables to probe jet substructure of quarks and gluons, Eur. Phys. J. C **78**, 334 (2018).

[28] Lingfeng Li, Tao Liu, and Si-Jun Xu, Jet topology, arXiv:2006.12446.

[29] Timothy Cohen, Mariangela Lisanti, and Hou Keong Lou, Semivisible Jets: Dark Matter Undercover at the LHC, Phys. Rev. Lett. **115**, 171804 (2015).

[30] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands,

An introduction to PYTHIA 8.2, Comput. Phys. Commun. **191**, 159 (2015).
[31] Johannes Bellm *et al.*, Herwig 7.0/Herwig + + 3.0 release note, Eur. Phys. J. C **76**, 196 (2016).
[32] M. Bahr *et al.*, Herwig + + physics and manual, Eur. Phys. J. C **58**, 639 (2008).
[33] H. Hadwiger, Integralsätze im konvexring, Abh. Math. Semin. Univ. Hambg. **20**, 136 (1956).
[34] Daniel A. Klain, A short proof of Hadwiger's characterization theorem, Mathematika **42**, 329 (1995).
[35] Tom Leinster, Integral geometry for the 1-norm, Adv. Appl. Math. **49**, 81 (2012).
[36] K. R. Mecke, Morphological characterization of patterns in reaction-diffusion systems, Phys. Rev. E **53**, 4794 (1996).
[37] Dominique Attali and André Lieutier, Reconstructing shapes with guarantees by unions of convex sets, in *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, SoCG '10 (Association for Computing Machinery, New York, NY, USA, 2010), p. 344–353.
[38] D. Göring, M. A. Klatt, C. Stegmann, and K. Mecke, Morphometric analysis in gamma-ray astronomy using Minkowski functionals—Source detection via structure quantification, Astron. Astrophys. **555**, A38 (2013).
[39] Innchyn Her, Geometric transformations on the hexagonal grid, IEEE Trans. Image Process. **4**, 1213 (1995).
[40] Amit Patel, Hexagonal grids, in *Red Blob Games* (2013).
[41] Hubert Mantz, Karin Jacobs, and Klaus Mecke, Utilizing Minkowski functionals for image analysis: A marching square algorithm, J. Stat. Mech. (2008) P12015.
[42] Min Lin, Qiang Chen, and Shuicheng Yan, Network in network, in *Proceeding of the International Conference on Learning Representations (ICLR), Banff, Canada* (2014), https://sites.google.com/site/representationlearning2014/home.
[43] Sung Hak Lim and Mihoko M. Nojiri, Spectral analysis of jet substructure with neural networks: Boosted Higgs case, J. High Energy Phys. 10 (2018) 181.
[44] Diederik P. Kingma and Jimmy Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA* (2015), https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html.
[45] François Chollet *et al.*, Keras, https://keras.io (2015).
[46] Martín Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, https://www.tensorflow.org/ (2015), software available from tensorflow.org.
[47] S. Catani, Yuri L. Dokshitzer, M. H. Seymour, and B. R. Webber, Longitudinally invariant K_t clustering algorithms for hadron hadron collisions, Nucl. Phys. **B406**, 187 (1993).
[48] Stephen D. Ellis and Davison E. Soper, Successive combination jet algorithm for hadron collisions, Phys. Rev. D **48**, 3160 (1993).
[49] Matthew J. Strassler and Kathryn M. Zurek, Echoes of a hidden valley at hadron colliders, Phys. Lett. B **651**, 374 (2007).
[50] Lisa Carloni, Johan Rathsman, and Torbjorn Sjostrand, Discerning secluded sector gauge structures, J. High Energy Phys. 04 (2011) 091.
[51] Elias Bernreuther, Thorben Finke, Felix Kahlhoefer, Michael Krämer, and Alexander Mück, Casting a graph net to catch dark showers, SciPost Phys. **10**, 046 (2021).
[52] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations, J. High Energy Phys. 07 (2014) 079.
[53] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, and V. Lemaître, A. Mertens, and M. Selvaggi (DELPHES 3 Collaboration), DELPHES 3, A modular framework for fast simulation of a generic collider experiment, J. High Energy Phys. 02 (2014) 057.
[54] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, Focal loss for dense object detection, in *Proceeding of the IEEE International Conference on Computer Vision (ICCV)* (IEEE Computer Society, Los Alamitos, 2017), https://www.computer.org/csdl/proceedings-article/iccv/2017/1032c999/12OmNApu5iv.
[55] Matteo Cacciari and Gavin P. Salam, Dispelling the N^3 myth for the k_t jet-finder, Phys. Lett. B **641**, 57 (2006).
[56] Christine O. Rasmussen and Torbjörn Sjöstrand, Hard diffraction with dynamic gap survival, J. High Energy Phys. 02 (2016) 142.
[57] Richard Corke and Torbjorn Sjostrand, Interleaved parton showers and tuning prospects, J. High Energy Phys. 03 (2011) 032.
[58] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran, Explainable deep learning: A field guide for the uninitiated, arXiv:2004.14545.
[59] Jesús Angulo, Some open questions on morphological operators and representations in the deep learning era, in *Discrete Geometry and Mathematical Morphology*, edited by Joakim Lindblad, Filip Malmberg, and Nataša Sladoje (Springer International Publishing, Cham, 2021), pp. 3–19.
[60] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, Distilling the knowledge in a neural network, arXiv:1503.02531.
[61] Sascha Diefenbacher, Engin Eren, Gregor Kasieczka, Anatolii Korol, Benjamin Nachman, and David Shih, DCTRGAN: Improving the precision of generative models with reweighting, J. Instrum. **15**, P11004 (2020).
[62] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), in *Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico,* (2016), https://iclr.cc/archive/www/2016.html.