




# Foundations of automatic feature extraction at LHC–point clouds and graphs

Akanksha Bhardwaj<sup>1,a</sup>, Partha Konar<sup>2,b</sup>, and Vishal Ngairangbam<sup>3,c</sup> 

<sup>1</sup> Department of Physics, Oklahoma State University, Stillwater, OH 74078, USA

<sup>2</sup> Theoretical Physics Division, Physical Research Laboratory, Shree Pannalal Patel Marg, Ahmedabad 380009, Gujarat, India

<sup>3</sup> Institute for Particle Physics Phenomenology, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK

Received 26 April 2024 / Accepted 23 August 2024

© The Author(s) 2024

**Abstract** Deep learning algorithms will play a key role in the upcoming runs of the Large Hadron Collider (LHC), helping bolster various fronts ranging from fast and accurate detector simulations to physics analysis probing possible deviations from the Standard Model. The game-changing feature of these new algorithms is the ability to extract relevant information from high-dimensional input spaces, often regarded as “replacing the expert” in designing physics-intuitive variables. While this may seem true at first glance, it is far from reality. Existing research shows that physics-inspired feature extractors have many advantages beyond improving the qualitative understanding of the extracted features. In this review, we systematically explore automatic feature extraction from a phenomenological viewpoint and the motivation for physics-inspired architectures. We also discuss how prior knowledge from physics results in the naturalness of the point cloud representation and discuss graph-based applications to LHC phenomenology.

## 1 Introduction

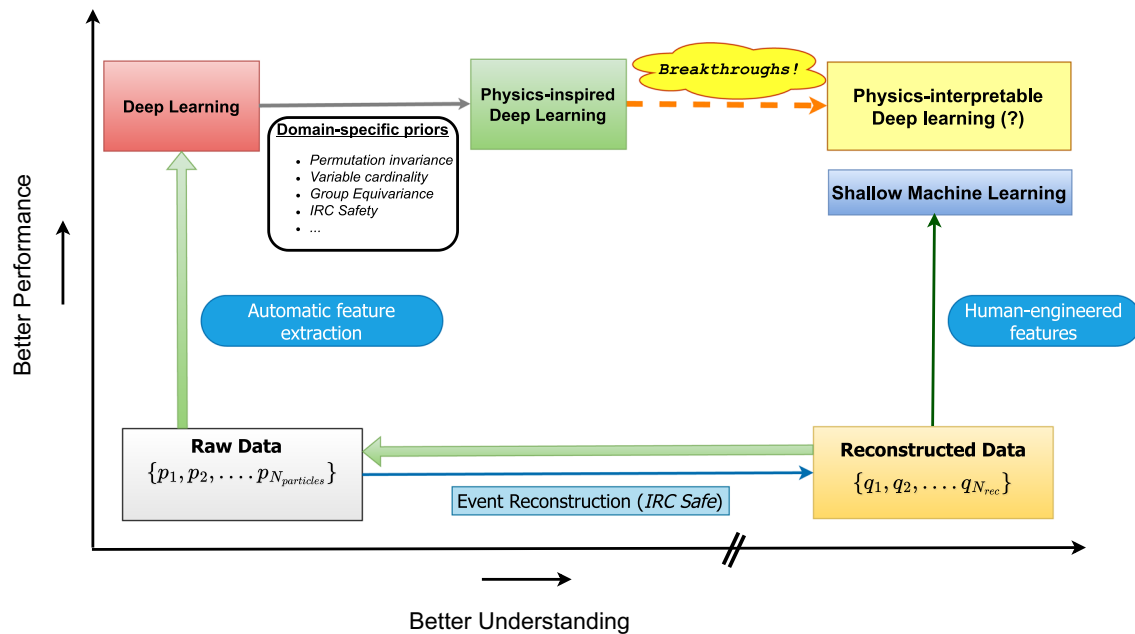
Modern machine learning (ML) techniques are quickly becoming ubiquitous in the natural sciences due to their excellent data processing capabilities and ability to find excellent interpolations from high-dimensional data. The situation is not too different in various branches of physics (see Ref. [1] for a recent review), where they are being employed to enhance traditional ways of studying physical systems ranging from finding the ground state of many-body quantum systems to predicting cosmological parameters. The situation is similar in collider phenomenology. Particularly relevant for the huge data that the Large Hadron Collider (LHC) will generate in the high-luminosity runs [2–5], the community is currently striving to work out the intricacies of these efficient data handling algorithms ranging from their applications to triggers [6–9]: the stage which decides which events to store and which to discard, the study of various detector modules [10–18], phenomenological applications [19–28]: which connects the recorded data to update our current understanding of the physical universe, and foundation models [29–34] capable of generalising across several applications.

The defining characteristic of current state-of-the-art ML algorithms is their ability to extract features from a high-dimensional input representation, which has undergone little to no processing (in terms of dimensionality reduction to a few variables through domain knowledge). These so-called deep-learning algorithms often need not rely on hand-engineered variables and can (with appropriate architecture design) outperform shallow methods relying on human-designed variables. The downside of such algorithms is the loss of understanding, in the traditional sense, of what the network is exploiting to outperform the human-engineered variables. Even though the increase, in principle, can be attributed to a higher degree of optimisation via numerical methods coupled with the neural network’s inherent power, the very high-dimensional parameter space prohibits an analytic understanding of the optimisation process.

<sup>a</sup> e-mail: [akanksha.bhardwaj@okstate.edu](mailto:akanksha.bhardwaj@okstate.edu)

<sup>b</sup> e-mail: [konar@prl.res.in](mailto:konar@prl.res.in)

<sup>c</sup> e-mail: [vishal.s.ngairangbam@durham.ac.uk](mailto:vishal.s.ngairangbam@durham.ac.uk) (corresponding author)



**Fig. 1** A schematic diagram on the plane of classifier performance and understanding of deep learning algorithms and its comparison to shallow machine learning methods

Taking the example of signal vs background classification as a proxy for performance, we can cast the interplay of performance and physics-understanding in a two-dimensional plane as shown in Fig. 1. In the bottom left, with virtually no separation and understanding reside the raw detector data<sup>1</sup> used as inputs to reconstruction algorithms. This process which include defining infrared and collinear safe jets, and other objects like leptons, photons, etc., based on isolation criteria on the measurements of the various components of the detector, give us a set of reconstructed objects on the bottom right. On the reconstructed data, one generally applies loose selection cuts to enrich the signal and then apply shallow machine learning after constructing observables based on physics insights of the specific signature in question. In deep-learning based analyses, one goes back to the raw data after pre-selection and uses the high dimensional data as inputs to a suitable model. These models generally reside higher on the performance axis compared to shallow methods but fare poorly on the understanding axis. With physics-inspired deep-learning methods, there is a small increase on the understanding axis with still comparable performance to out-of-the-box methods. However, this is still a long way from the traditional understanding of physics-based observable design (evident from the discontinuous scale on the understanding axis). It would be considered a breakthrough to reach an equal level of understanding to traditional approaches without potentially losing out on the performance gain. Although, what qualifies as equal, will of course be decided by the community at large, a pragmatic criterion could be the understanding of their systematic uncertainties [22, 35–39], especially those with theoretical origins [40, 41].

We systematically review the nature of automatic feature extraction for phenomenology at the Large Hadron Collider, taking a more expository approach to what is already known in mainstream machine learning literature and their relations/reinterpretations specific to LHC phenomenology. One major theme of current research in this context, is the suitability of the point cloud representation in analysing high-energy collision events. In the point cloud representation, events are described as an unordered set of particles, characterised by their measured properties like four-momentum, charge, etc., whose number is not constant on an event-by-event basis. This is due to the non-conservation of particle multiplicity in relativistic quantum mechanics and our innate interest in how the measured particles relate to the interactions at subnuclear length scales. As the details of these interactions reside in the correlations between the measured particles, relational structures (like a graph) between the set's elements are often employed to expose these correlations. While we touch upon graph-based representations and overview some applications, our coverage is essentially incomplete as our aim is to elucidate the interplay between the expressive power and generalisation capabilities of deep neural networks in the context of high-energy physics. For a more detailed literature review, we refer interested readers to Refs. [42–51] and the living review [52].

The rest of the review is organised as follows. In Sect. 2, we present a qualitative discussion of the expressivity of fully-connected feed-forward neural networks, and its interplay between generalisation performance, arguing

<sup>1</sup>Note that what we consider raw data here is still highly processed when one considers the intricacies of experimental measurements like track finding, pileup subtraction etc.

the need to curtail the theoretical expressivity to have good generalisation capabilities for specific purposes. We discuss these prior assumptions in the context of LHC phenomenology in Sect. 3 which motivates the set-based point cloud representation. Thereby, we present an introduction to graph-based architectures acting on the point cloud representation in Sect. 4. In Sect. 5, we overview some applications relevant to LHC phenomenology. We summarise in Sect. 6.

## 2 Universal approximation: theory and practice

Artificial Neural Networks (ANNs) are non-parametric models which learn an underlying target function based on observed data. By non-parametric, we mean that the free parameters do not have any inherent (physical) meaning and the analysis relies on approximating the target function as accurately as possible. The models are structured to capture a large set of possible functions. ANNs owe their origin to mathematical models of the biological neuron [53, 54]. As we will be discussing point cloud architectures which utilise dense feed-forward networks as building blocks, we will use the term Multilayer Layer Perceptron (MLP) to refer to such networks to avoid confusion with the composite architectures built out of such units.

This section presents a brief account of the mathematical structure of MLPs, their expressive power captured in terms of Universal Approximation Theorems (UAT). These UATs, generally phrased as existence theorems, deal with the ability to approximate any continuous function on a compact domain upto any arbitrary precision. Such generality at times impede a practical optimisation especially when the input dimensionality is very large. The expressivity is meticulously reduced based on the input representation and the training objective by biasing the network architecture to some smaller sets of functions. This preconditions are colloquially referred to as “*inductive biases*”, closely related to the paradigm of inductive learning. In such scenarios, we train a model on a specific training dataset to be able to generalise to samples not present during the training, assuming that the new samples follow the same relationship between the input and the target variable. The expected error on the unseen samples is called the generalisation error of the model and it is estimated with the help of the validation dataset. Highly expressive models generally result in very good training error with poor generalisation and frequently overfit the training data. When the architecture is suitable for the particular task at hand, it results in a better generalisation error and hence can reach lower levels of validation loss without overfitting to the training dataset. We end the section with a discussion on the need to curtail the expressivity via such biases.

### 2.1 Structure of multilayer perceptrons

A statistical model takes an input vector  $\mathbf{x} \in \mathbb{R}^n$  and learns a target function<sup>2</sup>  $\tilde{y} = f(\Theta, \mathbf{x})$  by tuning the set of parameters  $\Theta$  according to some known pair of tuples  $\{(\mathbf{x}, y)_i\}$ . Different models have different forms of functional dependence on the parameters and may involve hidden representations as in the case of neural networks. For MLPs, it is an affine map from the input space  $\mathcal{X} \ni \mathbf{x}$  to a sequence of spaces  $\mathcal{Z}_\alpha$ . Without any additional restrictions, these are simply  $\mathbb{R}^{n_\alpha}$ , with  $n_\alpha$  the dimensionality of the  $\alpha$ -th representation. Denoting the input vector as  $\mathbf{x} = \mathbf{z}_0$  and the output as  $\tilde{y} = \mathbf{z}_{n+1}$ , with  $n$  being the number of hidden layers, the mathematical form can be written as

$$\mathbf{z}_\alpha = A(\mathbf{w}_\alpha \cdot \mathbf{z}_{\alpha-1} + \mathbf{b}_\alpha), \quad (1)$$

where  $\mathbf{w}_\alpha$  is a  $n_\alpha \times n_{\alpha-1}$  weight matrix and  $\mathbf{b}_\alpha$  is a  $n_\alpha$  dimensional weight vector.  $A$  is an *activation function* applied to each element of the argument vector. As composition of linear functions is a linear function, activation functions need to be non-linear to be able to capture non linearities in data. On the other hand, not every non-linear activation usually leads to the universal approximation property, although the requirements are not very restrictive as we shall see in the following discussions.

The architecture as described above, are sequential, i.e., each layer feeds to the next and so on, to give the final predicted output, or the function  $\tilde{y}(\mathbf{x})$  is a composition of several functions. In cases where the quality of the approximation can be quantified with a differentiable metric between  $\tilde{y}$  and the true value  $y$ , one generally uses a gradient descent algorithm to optimise the parameters  $\mathbf{w}_\alpha$  and  $\mathbf{b}_\alpha$  which are initialised to some random value before the training starts.<sup>3</sup> Due to the compositional nature, the application of the chain rule, gives a handy way to update the parameters when utilising gradient descent by what is known as the *back-propagation* algorithm [58]. In the following, we will denote the parameters of an MLP by  $\Theta$ , and the space to which it belongs to with  $\mathcal{W} \ni \Theta$ , without explicitly talking about the sequential structure of the functional mappings. The optimisation algorithm finds a point  $\Theta_0 \in \mathcal{W}$  which best approximates a target function by reducing a metric between the approximated function  $\tilde{y}(\mathbf{x})$  and  $y$  for all  $(\mathbf{x}, y)$  in the training dataset.

<sup>2</sup>Without loss of generality, we will consider a scalar function.

<sup>3</sup>For more details of how the process of initialisation affects the network performance, see for instance Refs. [55–57].

## 2.2 Universal approximation theorems: a bird's eye view

The range of functions that neural networks can approximate are encapsulated within the so-called universal approximation theorems (UATs) (see [59–62] for instance). These theorems investigate the class of functions that MLPs with a certain activation can approximate with finite but arbitrary number of hidden nodes and/or layers. Most of them prove that the set of functions represented by a certain class of neural networks are *dense* in the set of bounded continuous functions in the  $n$ -dimensional hypercube.

The notion of dense sets in a larger superset is a generalisation of the properties of the rational numbers  $\mathbb{Q}$  and real numbers  $\mathbb{R}$ , with  $\mathbb{Q}$  being dense in  $\mathbb{R}$ . This means that for any  $r \in \mathbb{R}$ , we can find  $q \in \mathbb{Q}$  such that  $|r - q| < \epsilon$  for any arbitrarily small  $\epsilon > 0$ . To generalise this notion to more intricate scenarios, one defines a metric, which measures a “distance” between two elements of the set. A dense set  $\mathcal{D}$  in a superset  $\mathcal{S}$ , is one where for all elements  $t \in \mathcal{S}$ , we can find an element  $a \in \mathcal{D}$  which is arbitrarily close with respect to a metric  $d(s, a) < \epsilon$ . For UATs, this is generally the metric induced by the supremum norm:  $\sup_{\mathbf{x} \in D} |\hat{y}(\mathbf{x})|$ , which is the least upper bound of the absolute value of the function  $\hat{y}$  in the domain  $D$ . For a parametrised function  $\tilde{y}(\mathbf{x}) = f(\Theta, \mathbf{x})$  and a target function  $\hat{y}$ , we have

$$d(\tilde{y}, \hat{y}) = \sup_{\mathbf{x} \in D} |\tilde{y}(\mathbf{x}) - \hat{y}(\mathbf{x})|. \quad (2)$$

Loosely speaking, it measures the largest difference between the two functions in the domain  $D$ . Therefore,  $\epsilon$  quantifies the highest absolute difference between the approximated function  $\tilde{y}$  and the target function  $\hat{y}$ .

The superset of interest for universal approximation is  $C(I_n)$ , the set of bounded continuous functions<sup>4</sup> from the closed unit hypercube  $I_n = [0, 1]^n$  to  $\mathbb{R}$ . One of the first universal approximation theorem [60], proves that the set of functions represented by an MLP of one hidden layer and arbitrary but finite number of nodes with sigmoid activation is dense in  $C(I_n)$  with respect to  $d(\tilde{y}, \hat{y})$ . This means that for any given target function  $\hat{y} \in C(I_n)$ , there exists a function  $\tilde{y}(\mathbf{x}) = f(\Theta_0, \mathbf{x})$  parametrised by an MLP of some finite number of nodes in the hidden layer and a point  $\Theta_0$  in the parameter space which is an arbitrarily close approximation of  $\hat{y}$ . As an existence theorem, it does not talk about how to find the particular point  $\Theta_0$ , nor does it restrict it to any subset of  $\mathbb{R}^N$ ,  $N$  being the dimension of  $\mathcal{W}$  which is the number of tunable parameters in the MLP. This dimension is controlled by the number of nodes in the hidden layer and the theorems do not provide a particular strategy to find  $N$  for a target level of accuracy  $\epsilon$ . However, given the generality of the function space which the theorem covers, the formal notion of existence of an arbitrarily close approximation to any bounded continuous function predicated the modern interest and revival of Artificial Neural Networks.

Practically, an MLP with a single layer is not a very efficient function approximator and falls within the broader class of shallow machine learning algorithms [63]. This is because for highly fluctuating functions, one may need an exponentially large number of nodes compared to deeper architectures. It was known intuitively that depth increases the effectiveness of function approximation [63] which follows from close analogy to the computational complexity for width vs depth [64, 65] in circuits for implementing boolean operations, as well as the known ability of biological brains to learn simpler concepts first and more abstract concepts later in life. Deeper networks are able to approximate topologically more complex functions [66, 67] much more efficiently than shallow networks.<sup>5</sup> Intuitively, this can be understood [68] from the ability of the component functions in the overall compositional chain to induce topologically discontinuous changes in each mapping.

As the target function's closed form expression is not known, the approximation can only be as good as the quality and the amount available data. In HEP where synthetic data can be simulated based on well-understood theoretical formulations, the amount and quality of data is seldom a problem. Yet the generality of the functions that dense architectures can approximate results in their handicap. The efficient optimisation via gradient descent algorithms only finds the best point  $\Theta_0$  accessible from the initial point  $\Theta_i$ . However, domain knowledge from particle physics, motivate additional restrictions on the functions beyond continuity. Such restrictions can be built into the architecture which biases the functions to (strongly) always follow a particular property, or to (weakly) partially have some desired properties. We present a qualitative discussion of why such restrictions are needed in the next subsection.

<sup>4</sup>Although, there are larger function spaces like the space of Lebesgue integrable functions  $L_1(I_n)$  and possibility of UATs in them, we will restrict ourselves to  $C(I_n)$  as finding effective training strategies for discontinuous functions are extremely difficult for the large dimensionality of modern neural networks.

<sup>5</sup>In Ref. [66], it is found that the sum of Betti numbers (a measure of topological complexity) of the subset of the domain of the function with positive outputs for a single output network scales at most as a polynomial in the number of nodes for several activation functions for a shallow network while it can scale exponentially for deeper networks.

### 2.3 Restrictions beyond continuity: no such thing as a free lunch

As we have seen hinted in the previous description, the existence of arbitrarily close approximation via a dense neural network to any given continuous function does not necessarily translate to finding the said approximation. Due to their non-constructive nature, one has to often revert to heuristics and brute force searches in the form of hyperparameter<sup>6</sup> optimisation coupled with some form of gradient descent optimisation to find a workable architecture and a particular set of parameters in the weight space. Although this is manageable for moderately large input dimensionality, it becomes intractable very fast with increasing number of dimensions. This is true even for deep fully-connected networks with more than two hidden layers. To counter this issue, additional informative priors are generally employed when one goes to very high dimensional input spaces which restricts the function beyond continuity. In this subsection, we discuss the motivation and the need to assume these priors.

It is often enticing to think that given the universal nature of dense neural networks, an equally universal algorithm could exist that would guarantee an efficient optimisation for all distributions. However, this is known to be untrue [69–73], where it was found that without any assumption about the underlying distribution, there is no such thing as an algorithm which performs the best in comparison to any other algorithm over the full range of possible functions. A particularly relevant exposition for the choice of learning algorithms and the inherent balance in its performance in all possible learning tasks can be found in Ref. [71],<sup>7</sup> where a conservation law for off-training generalisation error was proved over the full range of target functions for a fixed input distribution and, finite and fixed training size. *Simply put, an algorithm performing well for a set of target functions will lose its performance and be worse than a random guess in a different set of target functions so that the overall sum balances over the full space of target functions.* Therefore, if one knows some properties of the target function beforehand, it is possible (or mandatory perhaps) to devise algorithms or models which favour that particular form. This motivates building restrictions in the possible functions that a non-parametric model (neural networks for our case) can approximate.

The restriction of the possible functions beyond continuity can be done primarily in three ways: the data representation, the architecture of the network, and the loss function including the choice of regulariser and the training procedure. The first two are related although for a given data representation, there can be competing choices of architectures based on the complexity. The last strategy can be regarded as independent of the first two while depending more on the nature of finding the optimal weights for a given data representation and architecture like convex vs non-convex optimisation, or saddle point finding vs extrema finding of the loss function. Strictly speaking, *depth* is also one such prior which forces the learning of hierarchical abstractions of the input data relevant to the particular function that needs to be learnt at the output layer. However, with increasing input dimensionality, this alone is not enough to guarantee efficient search of the target function and one has to include domain-specific constraints into the architecture. For instance, the structure of convolutional neural networks with local connectivity, parameter sharing etc., is motivated from the fact that images tend to have local features which are relevant for (say) classification and they can be present anywhere in the image.

When the architecture is structured according to domain-specific priors, there is usually an additional fully-connected module which takes the features and processes them for the final goal of the training. Therefore, the actual function that is approximated (by the downstream network) may not follow the implemented priors. The initial domain-specific modules are colloquially known as feature extractors, meaning, they replace the feature-engineering stage which builds domain-specific features normally utilised as inputs to shallow machine learning methods. Although deep-learning is a subset of machine learning algorithms, their distinction from shallow methods resides in their ability to find better functions than those relying on traditional hand-engineered features. There is a long history of the usage of shallow machine learning methods [74, 75] in collider experiments and phenomenology (see Ref. [76] for a review). These methods utilise physics-motivated variable definitions constructed by processing the low-level high-dimensional data that one records at the detectors. Although, dense feed-forward neural networks with more than one hidden layer, i.e., with no additional domain-specific designs, qualify as deep-learning models, the introduction of domain-specific design is what sets apart the modern algorithms which do not need extensively processed data. They can utilise the very high-dimensional low-level data and mostly outperform (dependent on the suitability of the priors) shallow machine learning methods.

For LHC phenomenology, all three methods of restricting the target function are utilised and they have been generally successful. However, due to the inherent nature of scientific research where favourable outcomes get published, there is a selection bias which can sometimes lead one to the false belief of out-of-the-box deep learning algorithms always performing better than shallow methods acting on physics-intuitive observables devised by experts. For instance, see Refs. [77, 78] for a more pragmatic standpoint on Convolutional Neural Networks' comparison to substructure variables. Nevertheless, the wealth of prior physics knowledge also stipulates one to utilise it as much as possible while not being too restrictive to cause a performance degradation. Although the

<sup>6</sup>The different choices given a base architecture like the activation function, optimisation method, learning rate, the number of hidden layers, the number of nodes in these layers etc., are known as hyperparameters.

<sup>7</sup>Also, see Ref. [63] for a discussion on the quest for artificial intelligence and its interplay with priors and these *no-free-lunch* theorems.



performance is only estimated *a posteriori*, it has been found that physics-inspired networks generally help better the performance, interpretability, and training convergence. We will discuss these priors and their relation to the point cloud representation in the next section.

### 3 Broad priors at the energy frontier

The nature of collisions at high energies is described via the Standard Model (SM) of particle physics—a Quantum Field Theory of the fundamental particles based on the gauge group  $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$ . Consequently, investigating particle collisions at the LHC has an inextricable connection to the underlying formalism, especially for phenomenological searches of new physics and particles. The nature of high-energy collisions result in the suitability of the point cloud representation to analyse events at the LHC. However, the restriction of the possible set of functions that neural networks can approximate based on priors is not restricted to physics, and most architectures in modern deep-learning usage inherently assume some form of symmetries or biases which help in the effective approximation for the particular domain. The underlying theme that unites the different deep learning models is the assumption of some form of geometry of the input data. With deep connections to functional analysis, differential geometry, and invariant theory, the study of the interplay between geometric priors within the architecture design, which restricts the function beyond continuity, resulting in efficient learning for various domains, is collectively referred to as geometric deep learning [79, 80]. Consequently, other than building physics-inspired biases into the architecture, existing architectures with their own domain-specific priors have proved to be advantageous for the physics programme at the LHC. In this section, we will discuss domain-specific priors resulting in the suitability of using the point cloud representation, as well as the compatibility of existing priors in the context of LHC phenomenology.

#### 3.1 Physical symmetries

One of the most important priors that result from physics knowledge is the existence of symmetries. When training a neural network for a particular purpose, without any additional restriction, the network does not know of the already known symmetries like Lorentz invariance. These symmetries can be built into architecture using group equivariant neural networks, where the hidden representations follow the group multiplication property and the maps between them commute with the group actions on the underlying spaces.

Given a group  $\mathcal{G}$  and a set  $\mathbf{X}$ , the left-action of  $\mathcal{G}$  on  $\mathbf{X}$  is a map  $A : \mathcal{G} \times \mathbf{X} \rightarrow \mathbf{X}$  with the following properties:

1.  $A(e, \mathbf{x}) = \mathbf{x}$  for the identity element  $e \in \mathcal{G}$  and any  $\mathbf{x} \in \mathbf{X}$ , and
2.  $A(g_1, A(g_2, \mathbf{x})) = A(g_1 g_2, \mathbf{x})$  for all  $g_1, g_2 \in \mathcal{G}$ .

A map  $f : \mathbf{X} \rightarrow \mathbf{Y}$  from sets  $\mathbf{X}$  to  $\mathbf{Y}$  which both admit group actions, say  $\mathbf{A}_\mathbf{X}$  and  $\mathbf{A}_\mathbf{Y}$  respectively, for the group  $\mathcal{G}$  is said to be  $\mathcal{G}$ -equivariant if  $f$  commutes with the group actions, i.e.,

$$f(\mathbf{A}_\mathbf{X}(g, \mathbf{x})) = \mathbf{A}_\mathbf{Y}(g, f(\mathbf{x})), \quad (3)$$

for every  $g \in \mathcal{G}$  and  $\mathbf{x} \in \mathbf{X}$ . The map  $f$  is  $\mathcal{G}$ -invariant if the action  $\mathbf{A}_\mathbf{Y}$  is trivial, i.e.,  $\mathbf{A}(g, \mathbf{y}) = \mathbf{y}$  for all  $g \in \mathcal{G}$  and any  $\mathbf{y} \in \mathbf{Y}$ .

Looking at the form of the group action, it is evident that it has a very close connection to group representations. Given a representation of the group  $\mathcal{G}$  in terms of invertible matrices of dimension  $n \times n$ ,  $\rho : \mathcal{G} \rightarrow GL(n, \mathbb{R})$  which preserves the group multiplication in terms of the matrix multiplication (a group homomorphism), we can define an action on  $\mathbb{R}^n$  as

$$A(g, \mathbf{x}) = \rho(g) \mathbf{x},$$

where  $\rho(g)$  are  $n \times n$  invertible matrices. Consequently, the Lorentz transformation

$$p'_\mu = \Lambda^\nu_\mu p_\mu, \quad (4)$$

is a linear action of the Lorentz group on the set of physical four-vectors  $p_\mu$ . Similar considerations hold for the multitude of groups that one encounters in high-energy physics, although the underlying field may at times be complex.

Evidently the use of function approximators which already take into account the group structure reduces the search space of possible target functions immensely. This may,<sup>8</sup> result in an increase in sample efficiency (the number of training samples required to reach a particular level of performance), convergence (the number of epochs required to reach that particular accuracy), reduced number of trainable parameters, or increase in overall performance with respect to non-equivariant architectures. This gain is dependent on the suitability of the target function and the particular group equivariance that is implemented into the architecture that learns the target function. As a simplified example, take the binary classification problem of two resonant particles with different (but with some overlap due to their width) masses decaying into two particles. The mass of the resonance is obviously a very important quantity in the discrimination and a Lorentz invariant feature extraction from the sum of the decay products' four vectors would expose this feature directly, while, say Euclidean group invariance in eight dimensions would struggle to find the distinguishing feature. From a purely data driven perspective, the four vector of the two particles are just some feature vector living in  $\mathbb{R}^8$ , and it is our physics intuition which drives the additional algebraic structure of the Lorentz group.

Equivariant neural networks have been instrumental in several domains ranging from image processing [81, 82], to the prediction of protein structures [83]. With the rich structure of symmetries in particle physics, it is no wonder that the community is exploring novel ways to utilise equivariant architectures [84–87] for various applications in theory and phenomenology.

### 3.2 Permutation invariance and variable cardinality

Another important aspect of studying collisions at the LHC, is the inherent absence of order in the data. In event analysis, the reconstructed objects like jets, leptons, photons etc., have four vectors and additional information like flavour and charge which form an unordered collection. Similarly in jet substructure analysis, the ordering of the constituents are irrelevant while the discriminating information is found in the inter particle correlations. Due to the non-conservation of particle multiplicity in relativistic quantum field theory where highly energetic particles inevitably produce additional particles which share the momentum, the number of elements in the set for either case is not constant. Therefore, one needs to consider an input representation which acts on unordered sets of variable cardinality, i.e., to preserve the property of sets, a function needs to be permutation invariant with respect to the order of the inputs, and be well-defined for varying number of inputs.

Restricting events to an ordered and fixed sized representation may result in truncation and combinatorial ambiguities [88]. While we may be interested in a specific number of coloured objects at the parton level, due to their highly radiative nature, it is often advantageous to not have a hard restriction on the number of reconstructed jets with additional jets arising from additional radiations. In other words, truncating an event signature to a fixed sized representation introduces ambiguities in how one should select these fixed number of objects with the kinematics inextricably linked to those objects which are not utilised in the representation. Combinatorial ambiguities arise when a signature has more than one way of associating the reconstructed objects to an earlier parton. For instance, in the fully leptonic decay of a pair of top quarks, other than the inability to reconstruct the momentum of the two neutrinos, we also have to decide the allocation of two bottom jets (as we cannot determine the charge) to the two reconstructed lepton. The situation becomes more complicated if we have more than two identified bottom jets in a candidate event.

Truncation ambiguities are much more pronounced in jet-substructure studies, where analogues of multiplicity are well-defined with more stringent theoretical considerations [89–91]. This difficulty is due to the resolution of QCD radiation patterns at much smaller angular scales within the jet in the regimes with considerable collinear enhancement. Combinatorial ambiguities can also arise in signal decays into more than two particles, like the top quark, when one is interested in assigning a parton-level flavour to some of the constituent subjects.

### 3.3 Infrared and collinear safety: resilience to deformations

The presence of massless particles, demand an additional consideration for perturbative calculations in QFT, namely, infrared and collinear (IRC) safety. This is one of the cornerstones for fixed order calculability of any observable for theories containing massless gauge bosons, where there is an intricate cancellation of infrared divergences arising from virtual loops with the interference contribution of soft and collinear emissions. This cancellation is guaranteed to happen for IRC safe observables, i.e., when an observable defined for  $n + 1$  particles is equal (or as a limit) to the same for  $n$  particles when any of the  $n + 1$  particles become soft or any two of them become collinear. Therefore, IRC safety puts additional relations between the functions defined on sets with cardinality difference of one.

In domains like image processing or 3D point cloud processing, neural networks have some form of pooling or readout operation in a local region which propagates the summary statistic of the region forward. This results in

<sup>8</sup>As the dimensionality of the input space is still very large, it is hard to ascertain the improvements a priori.

the propagated information being approximately resilient to *local deformations* in the data, which mainly arise from noise. Due to the enhancement of radiation in the almost collinear or soft regimes, there is an inherent deformation of the data living in the rapidity-azimuth plane for the point cloud or the image representation. Near collinear emissions result in local distortions in a *jet-image*, while soft emissions are inherently less prominent since the harder hits will dominate the sum in the convolution operation. Therefore, Convolutional Neural Networks already take into account resilience to such soft and collinear emissions [92]. On the other hand, point cloud architectures which take the three dimensional features ( $p_T$ ,  $\Delta\eta$ ,  $\Delta\phi$ ) of each particle as inputs to an MLP, do not distinguish between the transverse momentum and the angular variables and as such, the extracted features are IRC unsafe. Nevertheless, the very general nature of point cloud representation allows defining IRC safe feature extractors acting on sets of particles as epitomised by Energy Flow Networks [93–97] along with its graph and hypergraph generalisations [98–102].

An important aspect of the local deformations induced by QCD radiation is in its inevitability even in the assumption of perfect measurements, since it results from the underlying physical process rather than being induced by measurement errors. Therefore, IRC safety is a crucial element of any phenomenological analysis at hadron colliders. While the IRC safe anti- $k_t$  [103] jet definition has been the main workhorse of physics analyses at LHC, jet definitions utilised at Tevatron were still IRC unsafe [104, 105]. One should note that theoretical predictions even between IRC safe jet definitions can result in systematic differences of 10% or larger as shown in Ref. [106], and without IRC safety, we are at the mercy of poorly understood non-perturbative effects. As deep learning approaches to event analysis will presumably expand in the future, it is important to consider their theoretical aspects within perturbative QCD. In this regard, the algorithm's IRC safety is an important requirement for its understanding, especially since IRC-safe deep learning algorithms do not restrict the expressivity to a very high degree for jet-tagging.<sup>9</sup> However, IRC safety alone does not equate to the perturbative physics being under control, or the absence of non-trivial non-perturbative effects, and additional considerations like the algorithm's all-order effects [96, 107–109] need to be properly addressed, with little loss (preferably without losing out) on their generalisation capabilities. On the other hand, there are also classes of observables [110, 111] that do not require fixed-order IRC safety but can be studied with more elaborate theoretical underpinnings. Notably, additional sources of deformation also arise from experimental measurements, as well as the presence of pileup interactions.

### 3.4 Priors from image processing

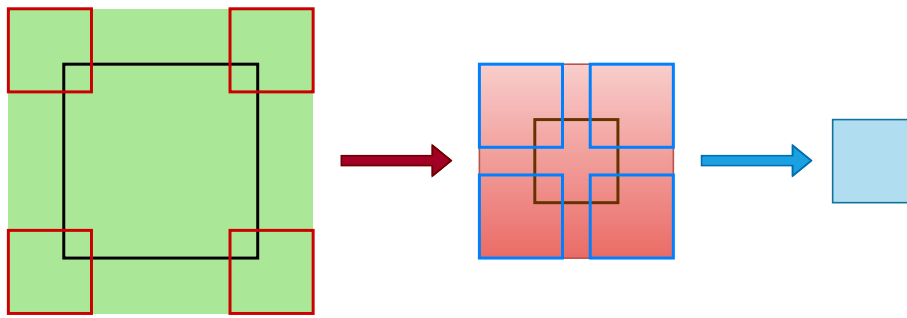
Convolutional Neural Networks (CNNs) [112, 113] remain the quintessential example of deep learning algorithms with very good generalisation abilities after curtailing the expressivity based on domain knowledge. On top of being approximately resilient to local deformations as explained in the previous subsection, they also assume local connectivity and parameter sharing, which, when combined with successive composition of layers result in scale separation. These biases are carried over in some sense into graph-based architectures depending on specific details of the architecture which we will touch upon in the next section. Here, we discuss the natural ways in which these three assumptions already account for the structure of QCD radiation patterns arising at the LHC.

Local connectivity means that for a given input pixel or a node, the information processed by the network concentrates in its local neighbourhood. Compared to the image size, the smaller filters in a CNN act on local regions determined by their relative sizes. The collinear splitting structure in parton showers which result in the formation of jets is an inherently local phenomenon in the rapidity-azimuth plane. Therefore, the local nature of feature extraction looks into the local substructure in jets or jets in events. Parameter sharing refers to reusing the same set of weights for different local regions in the input. In QCD, the universality of parton showers results in similar patterns of radiation over the full detector<sup>10</sup> in different positions which motivates reusing the same set of weights to pick up such features. Separation of scales occurs via successive composition of functions acting on local regions in the input representation. Being a more refined version of depth to learn hierarchical abstractions of the data, the locally acting functions create a separation of scales, where the filters in initial layers look at the immediate locality and deeper ones sequentially capture larger scales. This is diagrammatically explained in Fig. 2. Consequently, hard prongs within jets or jets within events are present at different scales in the rapidity-azimuth plane, and a separation of scales help look into these structures in a hierarchical manner. An example for a three prong top jet is shown in Fig. 3.

<sup>9</sup>If the information of interest for a particular training objective is in the high-energy patterns in an event, one would intuitively expect an IRC-safe algorithm to be relatively comparable to unsafe ones, as IRC-safe observables concentrate on the hard energy flow of the event.

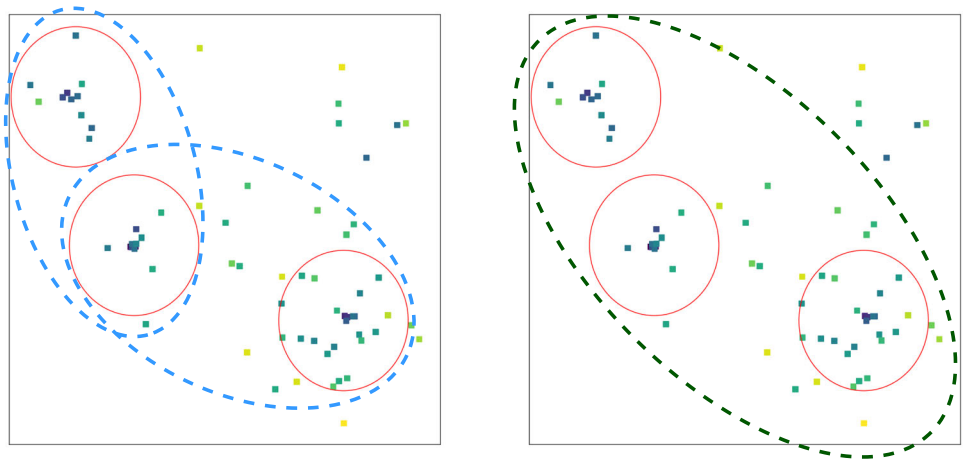
<sup>10</sup>Even though pile-up break this generality, this remains true after pileup removal after which phenomenological analyses are conducted on events following such patterns.





**Fig. 2** The figure shows a diagrammatic representation of the effective region (shown as a solid green box) in the input image for a second convolution on the first feature map. The result of the convolution on the green region of the image with the filters (shown as hollow red squares) produces a part of the feature map shown as a solid red box. Similarly, a second convolution on the region of the feature map with the different filters of the same size produces a region of a new feature map. Therefore, while the first filters look into the local region corresponding to their actual size, the second filters learn the features in the image corresponding to a much larger area (determined by the relative size of the filters) in the original image

**Fig. 3** The hierarchical separation of scales encountered in a top jet image with the little boxes denoting calorimeter hits. The red circles denote three subjects, which represent the first local features in the jet. On the left, the blue ovals denote the scale at which the two prong structure becomes visible, while on the right the green dotted oval is the scale at which all three subjects become relevant



## 4 Graph based architectures

In the previous section, we discussed some pertinent characteristics of high-energy collisions at LHC that motivate a set-based representation of events. In this section, we elaborate on the set-based representation and architectures that preserve the set property, concentrating on the graph representation that endows relational structures between pairs of constituents of the set.

### 4.1 Functions on sets

The set-based representation where the elements of the set represent some feature of objects in any underlying metric space is known as point clouds. It originates from the set-based representation of discrete points in 3D coordinate space for use in various computer graphics and imaging applications while being general enough to allow the elements to reside in more abstract spaces like those found in high-energy physics.

As we are interested in sets of finite but arbitrary size consisting of features living in an abstract space, say  $\mathcal{X}$ , we are generally interested in learning functions from the powerset<sup>11</sup> of  $\mathcal{X}$  to the output domain  $\mathcal{Y}$ . Such functions are required to be permutation invariant with respect to the input ordering to preserve the property of sets as unordered collection of objects. The universality of such functions was studied in the Deep Sets [114] framework, which defined set based feature extractors that possess the property of permutation invariance or equivariance. For

<sup>11</sup>If the cardinality of  $\mathcal{X}$  is infinity, we are interested in elements of the powerset which has bounded cardinality.

the invariant case, a Deep Sets model  $f$  is of the form

$$f(\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M\}) = \rho\left(\sum_{i=1}^M \phi(\mathbf{h}_i)\right), \quad (5)$$

where  $\mathbf{h}_i$  are the set elements,  $M$  the cardinality of the set, and  $\rho$  and  $\phi$  are MLPs. We can see that the function  $\phi$  learns a per-object map to a latent space, which undergoes a permutation-invariant summation over the set constituents. Going into the two-part segregation of deep learning algorithms, the MLP  $\phi$  acts as a feature extractor while  $\rho$  gives the downstream output. The extracted features are combined in a permutation invariant manner under the summation, which reflects the prior of set-based functions. Note that the summation can be replaced by any permutation invariant operation like taking the component-wise maximum or minimum over the latent representation.

Although the Deep Sets algorithm is a powerful feature extractor, the feature extraction stage looks only into the individual elements of the set. The downstream MLP, on the other hand, has access to the aggregated global information of the entire set. Such an architecture does not exploit the relational information between the set's constituents. To circumvent this issue, one endows additional structures on the set, like graphs, which expose pairwise relationships, or hypergraphs, which generalise graphs to account for any  $n$ -element relationships within the set.

## 4.2 Describing data with graphs

As we have describe above, graphs equip the input set with relational information between two elements. These elements in graph terminology are called nodes or vertices. The edge set  $\mathcal{E}$ , is a subset of the cartesian product  $\mathcal{S} \times \mathcal{S}$ . The presence or absence of an edge, say  $(i, j)$  in  $\mathcal{E}$  is determined by usefulness of the relation between the two elements  $i$  and  $j$ . Consequently, the way a graph is constructed determines the information highlighted amongst the nodes. In LHC physics, these may include forming edges between geometrically close components in a detector module, a Feynmann diagram based construction, or closeness in the rapidity-azimuth plane. If  $\mathcal{E} = \mathcal{S} \times \mathcal{S}$ , the corresponding graph is called a complete graph, the relation between every pair of elements is exposed in the graph. This is generally avoided as it leads to a quadratic scaling with the input set cardinality.

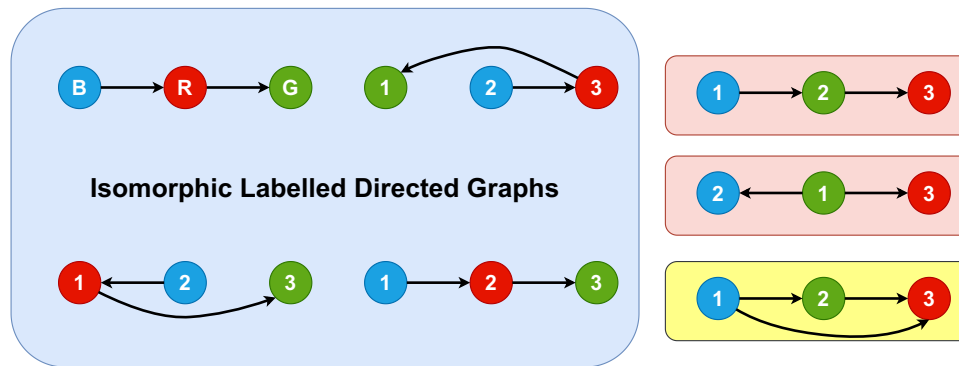
A graph can be represented as  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ , where  $\mathcal{S}$  represents the set of nodes and  $\mathcal{E}$  represents the set of edges. Each element in the set is some feature vector  $\mathbf{h}_i \in \mathbb{R}^d$ . However, the set  $\mathcal{S}$  may at times refer to the index set consisting of  $i \in \{1, 2, \dots, |\mathcal{S}|\}$  in the following discussions which will be clear from context. The edge set  $\mathcal{E}$  for computational purposes, can be suitably represented as an  $|\mathcal{S}| \times |\mathcal{S}|$  matrix  $\mathbf{A}$ , with components

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases},$$

which is called the adjacency matrix of the graph. Note that  $\mathbf{A}$  assumes a particular ordering of the nodes. This is the same when we input the graph quantities to a computer in the form of an array. Although there can be complexity studies of graphs where the nodes are considered indistinguishable, the nodes for our case are distinguished by their feature vectors  $\mathbf{h}_i$ . Moreover, the edges for our case will be directed as the message functions (to be explained later) will generally be asymmetric to interchange of the two constituent nodes of an edge.

The addition of edges as relational pairs require the conservation of the overall structure of the edge set for any permutation of the nodes. This is a stricter requirement than permutation invariance of the nodes, and formally covered within the notion of *graph isomorphisms*. Any two graphs  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$  and  $\mathcal{G}' = (\mathcal{S}', \mathcal{E}')$  are isomorphic to each other if we can find an invertible function  $N : \mathcal{S} \rightarrow \mathcal{S}'$  between the node sets which induces an invertible map  $E : \mathcal{E} \rightarrow \mathcal{E}'$  between the edge sets. As permutation of the elements of  $\mathcal{S}$  can be viewed as a bijection of the set to itself, graph isomorphisms additionally require the conservation of the edge structure for any permutation of the nodes. Although the notion of graph isomorphisms is often equated to being oblivious to relabelling of the nodes, our case is not as straightforward as assuming such a simple relabelling as the node features are immutable intrinsic labels of the nodes. On top of these node features, we assign an index to the the array containing each node features and construct the edges as a two-dimensional arrays of size  $2 \times |\mathcal{E}|$  in the sparse representation, or the adjacency matrix of size  $|\mathcal{S}| \times |\mathcal{S}|$ . Graph isomorphisms then practically relate to how the node features are arranged in this representation with the graph property intact as long as we work with quantities which are invariant for different ordering of the node features.

A representation of different isomorphic and non-isomorphic graphs with three coloured nodes is shown in Fig. 4. The colour in our case represents the node features which are intrinsic to that particular node. Therefore, one has to take into account this intrinsic character when defining graph isomorphisms. All graphs within the blue bounded box are isomorphic to each other as one can easily define invertible maps between the node labels which preserved



**Fig. 4** Examples of isomorphic and non-isomorphic graphs between directed graphs with nodes distinguished by their colours. The graphs in the bounded blue box are isomorphic to one another while the three on the right are not isomorphic to any other graph. The two graphs within the red boxes become isomorphic to the blue ones when one forgoes the difference in the colours and the direction of the edges

the underlying directed connections between the coloured nodes. We have intentionally ordered the nodes in a straight line and numbered them in increasing natural numbers (except for the one on the top-left) to make it evident the analogy to a matrix based representation where the numbers denote its index. On the right all three graphs are not isomorphic to each other or to any of the ones in the blue box when considering the colours and edge directions. As there can be no bijections between node sets or edge sets of different sizes, the graph on the yellow box at the bottom can never be isomorphic to any other graph in the figure when relaxing the additional constraints of colourings and directions. However, the graph on top right is isomorphic to the ones in the blue box if one forgoes the colourings but keeps the edge directions. This is not the case for the one in the middle since the green node has two outgoing edges while it is not the case in the other graphs. Finally, all graphs except for the one in the yellow box are isomorphic to each other if one forgoes both the colourings and the edge directions.

We now explain some terminologies which will be used in the following discussion of graph neural networks. The neighbourhood of a node  $i$ , written as  $\mathcal{N}(i)$  is the set of all connected nodes to  $i$ . For directed graphs, one usually regards either the incoming or outgoing edges (but not both) to define the neighbourhood. For instance, in the isomorphic graphs within the blue box in Fig. 4, the neighbourhood of the red node if one considers incoming edges is made up of the blue node, while for outgoing connections it is the green node alone. The  $l$ -hop neighbourhood of a node  $i$  consists of all nodes which can be reached from the node  $i$  by following at most  $l$  edges. Like before, the direction should be consistently defined for directed graphs and  $l = 1$  results in the usual neighbourhood definition. In Fig. 4, considering outgoing edges, the blue node's 2-hop neighbourhood consist of the green node on top of the red one. For our discussion of GNNs, we will consider the neighbourhood  $\mathcal{N}(i)$  to be defined as the set of nodes with incoming connections to  $i$ . Thereby, we will denote any edge attribute connecting two nodes of indices  $i$  and  $j$  with the subscript  $i \leftarrow j$  to make the direction self-evident.

### 4.3 Graph neural networks

In order to extract features from a graph for graph-level purposes, we need an architecture which extracts features which are invariant with respect to graph isomorphisms<sup>12</sup> where the extracted features are identical for any two isomorphic graphs. For node or edge level applications where one uses their features rather than the global graph representation, the output of the network should depend only on the intrinsic features and not on the index of the features, i.e., for any two isomorphic graphs we can obtain an invertible map between the node sets which preserve the updated features of the graph components.

One of the first proposals of Graph Neural Networks [115, 116] (GNNs) for any general graph compatible with back-propagation optimisation devised the concept of states of nodes and edges adapting their usage in recursive neural networks [117, 118] which acts on directed acyclic<sup>13</sup> graphs. The essential quality which conserved the graph properties consisted of recursively updating the states in successive timesteps with a shared parametric model like an MLP. More recent generalisations include the Message Passing Neural Network (MPNN) [119] approach and the more general Graph Networks [120]. Our discussions will concentrate on the more popular MPNNs.

<sup>12</sup>Technically, this means that the graph representation is defined for an equivalence class of isomorphic graphs rather than each individual graphs in the class.

<sup>13</sup>An acyclic graph is one where there are no paths with distinct nodes between any two nodes which form a closed loop.

An MPNN consists of several message-passing operations that takes in a graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$  with node features  $\mathbf{h}_i \in \mathcal{S}$ , and optionally, edge features for all  $(i, j) \in \mathcal{E}$  as  $\mathbf{e}_{i \leftarrow j}$  and updates the node features via the following steps:

1. **Message passing:** A learnable function (generally an MLP)  $\phi_e$ , called the message function takes the node features  $\mathbf{h}_i$  and  $\mathbf{h}_j$ , for each edge  $(i, j) \in \mathcal{E}$  and optionally the edge feature  $\mathbf{e}_{i \leftarrow j}$  and evaluates the message as

$$\mathbf{m}_{i \leftarrow j} = \phi_e(\mathbf{h}_i, \mathbf{h}_j, \mathbf{e}_{i \leftarrow j}). \quad (6)$$

2. **Message Aggregation:** For each node  $i$ , we aggregate the messages  $\mathbf{m}_{i \leftarrow j}$  for all nodes  $j$  in the neighbourhood  $\mathcal{N}(i)$  with a permutation invariant function  $\square_l$

$$\mathbf{m}_i = \square_l(\{\mathbf{m}_{i \leftarrow j} \mid j \in \mathcal{N}(i)\}). \quad (7)$$

3. **Node update:** The node features are updated to  $\mathbf{h}'_i$  as a function  $\phi_h$  (which also can be an MLP) of the aggregated message  $\mathbf{m}_i$  and the input node feature  $\mathbf{h}_i$  as

$$\mathbf{h}'_i = \phi_h(\mathbf{h}_i, \mathbf{m}_i). \quad (8)$$

The number of message passing operation is fixed for a particular model and one either uses the messages  $\mathbf{m}_{i \leftarrow j}$  and the node feature  $\mathbf{h}_i$  as inputs to an MLP for edge and node specific tasks, respectively. For graph level purposes, we constructs the graph representation  $\mathbf{G}$ , by a readout operation on the node features with an analogous global readout function  $\square_g$  as  $\mathbf{G} = \square_g(\{\mathbf{h}_i \mid i \in \mathcal{S}\})$ .

It is now evident that an MPNN's local connectivity is determined by the graph's structure with the first message-passing operation looking at the immediate neighbourhood, and successive ones looking at larger and larger  $l$ -hop neighbourhoods. The parameter sharing (the message function or the node update MLP is the same for all edges and nodes), and the message- aggregation stage, brings forward the biases discussed for CNNs to GNNs. However, while the pooling operations downsample the image representation thereby reducing redundant information from going deeper into the architecture, the readout operation does not reduce the cardinality of the node set and there is a much faster increase of redundant information amongst the node representations as one goes deeper. This hugely restricts the practical viability of deeper MPNNs for small graphs as the node features become increasingly similar with more message-passing operations.

The GN formalism additionally consists of edge updates and global feature updates, however, most common GNNs utilised in LHC phenomenology can be explained as a specific form of the MPNN approach. For instance, the message function in edge-convolutions [121] is of the form

$$\mathbf{m}_{i \leftarrow j} = \phi_e(\mathbf{h}_i \oplus \mathbf{h}_j - \mathbf{h}_i)$$

with the node updated via  $\mathbf{h}'_i = \mathbf{m}_i$  after a choice of permutation invariant aggregation. The operation  $\oplus$  denotes concatenation of the two vectors.

Another important operation is the attention mechanism which currently have state-of-the-art performance on the public top-tagging dataset. Although attention mechanisms originated [122] as a way to circumvent the loss of information in Recurrent Neural Networks for natural language processing (NLP), the Transformer model [123] which solved the bottleneck for efficient training of sequential language models by utilising attention alone while forgoing the recurrence, also found state-of-the-art performance for various NLP tasks and opened the proverbial ‘‘Pandora’s box’’ of Large Language Models [124–127]. While these exciting developments are outside the scope of the current review, the attention mechanism being a set operation is applicable to point clouds [128, 129]. The message function for any general attention mechanism can be written as

$$\mathbf{m}_{i \leftarrow j} = \mathbf{w}(\mathbf{h}_i, \mathbf{h}_j) \alpha(\mathbf{h}_i),$$

for some learnable function  $\alpha$  and attention mechanism  $\mathbf{w}$  whose output dimension should either be a scalar or a vector with the same dimensions as that of  $\alpha(\mathbf{h}_i)$  for component-wise multiplication. The learnable function  $\mathbf{w}(\mathbf{h}_i, \mathbf{h}_j)$  denote the attention of the node  $j$  with respect to the source node  $i$ , and can be cast in different ways according the particular attention mechanism. For the interpretation of  $\mathbf{w}(\mathbf{h}_i, \mathbf{h}_j)$  as weights in the summed aggregation  $\mathbf{h}'_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{i \leftarrow j}$ , the requirement  $\sum_{j \in \mathcal{N}(i)} \mathbf{w}(\mathbf{h}_i, \mathbf{h}_j) = 1$  is achieved by a softmax normalisation on the  $j$  index. Due to this learnable importance of the different nodes, attention-based networks often employ a complete graph structure where the job is left to the attention mechanism to learn the respective inter-node importance.

## 5 Overview of applications in high energy physics

Graph Neural Networks in their different guises have been explored in a varied range of applications for the physics program at the Large Hadron Collider. There can be several ways to categorise these applications based on the learning algorithm, the prediction task, and the physics application. The categories of learning methods broadly fall under unsupervised, semi-supervised and supervised methods, while the prediction task can be segregated into node, edge and graph level tasks. In this section, we give a brief overview of the applications of graph-based for some physics applications at the LHC.

**Jet Classification:** The prototypical example of deep learning applications is the classification of large radius QCD jets [130] from boosted hadronic decays of heavy particles and various point-cloud architectures have been studied for jet classification. Carrying forward the point cloud representation, ParticleNet [131], which utilized dynamic graph convolutions, presented the first application and showed high efficiency in classifying QCD/top and quark/gluon tagging. Interaction networks-based methods [132, 133] were also explored for jet tagging for 2-prong and 3-prong boosted jets. Attention-based GNNs [134] were explored for quark-gluon tagging applications. The point transformer [129] architecture has also been explored [135] for discriminating quark/gluon and top/qcd jets. Chebyshev graph convolutions [136] which enables the model to capture local dependencies intrinsic to jet formation, have also been studied for jet-tagging [137]. Haar pooling message passing networks [138, 139], have also shown to improve tagging performance compared to the usual readout operations. Till date Particle Transformer [140], based on a modified multi-head attention mechanism [123, 141], shows state-of-the-art performance in the public top tagging dataset [142].

As already explained in Sect. 3.1, physical symmetry has been used in graph-based architectures. Lorentz Group Network (LGN) [86] utilise the isomorphism of proper orthochronous Lorentz group  $SO(1, 3)^+$  to the projective special complex linear group  $PSL(2, \mathbb{C})$  to build tensorial representations decomposed into irreducible representations in terms of the Clebsch-Gordan coefficients. While LorentzNet [143, 144] implemented a simplified scalar and vector representation in the hidden representations to extract the features. Later, PELICAN [145, 146] combined permutation and Lorentz equivariant aggregations. On the public top tagging data set, these architectures matched or outperformed ParticleNet with fewer trainable parameters. Meanwhile, LorentzNet showed a very high sample efficiency, reaching an AUC of 0.9839 with only 5% of the training data. LorentzNet was augmented with capsule network [147], showing improved performance for quark-gluon performance. The gain in performance comes because the feature extraction specifically utilises the physical symmetry of the dataset.

When utilising the jet's constituents when extracting features with deep-learning models, it is not necessary that automatic feature extraction on a jet is IRC safe, and the IRC safety of these features depend not only on the input features but also on how they are structured. Constructing infrared and collinear-safe feature extractors on point clouds has also received considerable effort. Energy Flow Networks [93] (EFNs) adapted the Deep Sets [114] framework to account for IRC safety by utilising per-particle maps of the angular coordinates and performing a summed readout over the particles after linearly weighting with energy. Extensions of this approach include a permutation equivariant EFN [94], building hierarchical EFNs by utilising a basis of Legendre polynomials [95], Lipschitz-EFNs [96] where the extracted features are made to follow the Lipschitz continuity condition motivated from the geometry of particle collisions [148], and utilising higher moments [97]. The per-particle nature of EFN feature extraction has been generalised to graphs in the framework of Energy-weighted Message Passing Networks [98] for supervised classification. Further extensions include hypergraph based feature extraction [100], and equivariance in the rapidity-azimuth plane [101, 102]. An architecture based on dynamic edge convolutions was also studied for tagging jets originating from dark showers [149].

Another theoretically motivated way to represent a jet as a graph is through the Lund tree declustering [150, 151], which maps emissions according to the jet clustering history. LundNet [152] employed Lund graphs as inputs for various binary jet classification tasks. Due to the theoretically transparent nature of the input construction, it allows more detailed semi-analytic studies [153] of its discriminative power than is possible with the other graph-based approaches.

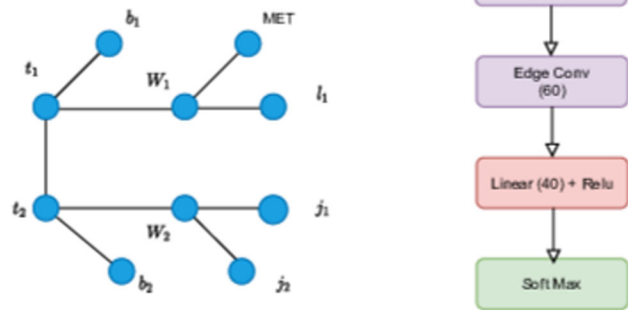
### **Discovering New Physics via Event Classification:**

Graph Neural Networks (GNNs) can categorize particle collision events into different classes, distinguishing between signal and background events. Leveraging the inherent graph structure of particle interactions, GNNs are adept at capturing nuanced characteristics that conventional approaches may overlook, thereby improving the accuracy of event classification. GNNs are advantageous for the event classification task for LHC data as they overcome the shortcomings of CNNs, which are limited to the Euclidean domain. An event at a collider can be naturally represented as graphs, where the final state particle represents the node and their interactions are encoded by edges.

In phenomenological studies, fully connected graphs and physics-based topology-inspired graphs are explored for event classification. Physics-inspired topology-based graphs are an efficient approach to capturing complex data relationships while minimizing NN parameters. In a multiclass set-up for the semileptonic  $t\bar{t}$  final states, GNN was used to discriminate the thirteen independent Wilson coefficients [154]. The decay-topology-inspired graph



**Fig. 5** Example of a decay topology inspired graph (figure taken from Ref. [154])



is shown in Fig. 5. To construct the graph, final state reconstructed particles are first added as the nodes, and additional nodes are added to the topology if the parent of the decay product can be reconstructed. On the other hand, fully connected graphs are also advantageous in case of unknown underlying physics topology or if fewer final state particles are involved in an event.

Graphs can have varying numbers of nodes and edges for each event, unlike fixed-size inputs in typical CNNs. Therefore, GNNs are adaptable to work on different event topologies and can generalize well across various types of particle collisions. In an early study, [155] collision events were represented as event graphs, and the message-passing neural network (MPNN) was used to search for the stop pair production at the LHC. Employing GNN for particle-flow (PF) reconstruction in a high-pileup environment has been developed for a general-purpose multilayered particle detector [156]. In Ref. [157], GNN with a fully connected graph is used to combat comprehensive backgrounds arising from other SM processes, which improves the sensitivity of di-Higgs analysis in the vector-boson fusion production channel. Similar architectures are used to improve the four top measurements for SM [158] and constrain the BSM parameter in the context of 2HDM. The fully connected graph and architecture used in the four top analysis are illustrated in Fig. 6. Self-attention to extract features from jet substructure and cross attention mechanism to combine these features for event level classification have also been studied [159] for BSM searches at LHC showing improved performance compared to simple concatenation of features. Accurate measurement of the trilinear and quartic Higgs self-couplings is essential to understanding the shape of the Higgs potential and the electroweak phase transition. While traditional searches at the LHC using Higgs pair production provide limited constraints, recent studies using Graph Neural Networks (GNNs) have shown promise in improving these bounds [160]. Topological graph algorithms were also developed to reconstruct intermediate particles and address combinatorial challenges in full hadronic decays  $t\bar{t}$  [161]. Recently, hypergraph representations [162] have been employed to demonstrate the reconstruction of parent particles for full hadronic decays of  $t\bar{t}$ . Events represented by weighted nodes with edges depicting the distance between events in kinematic space, were utilised [163] to separate the stop quark pairs from the top pair in the kinematic space. In Refs. [164, 165],  $t\bar{t}X$  process is scrutinized with Transformer and graph-based event classification. Using the point cloud representation for Higgs boson decays in tau leptons significantly improves compared to classical analysis techniques [88].

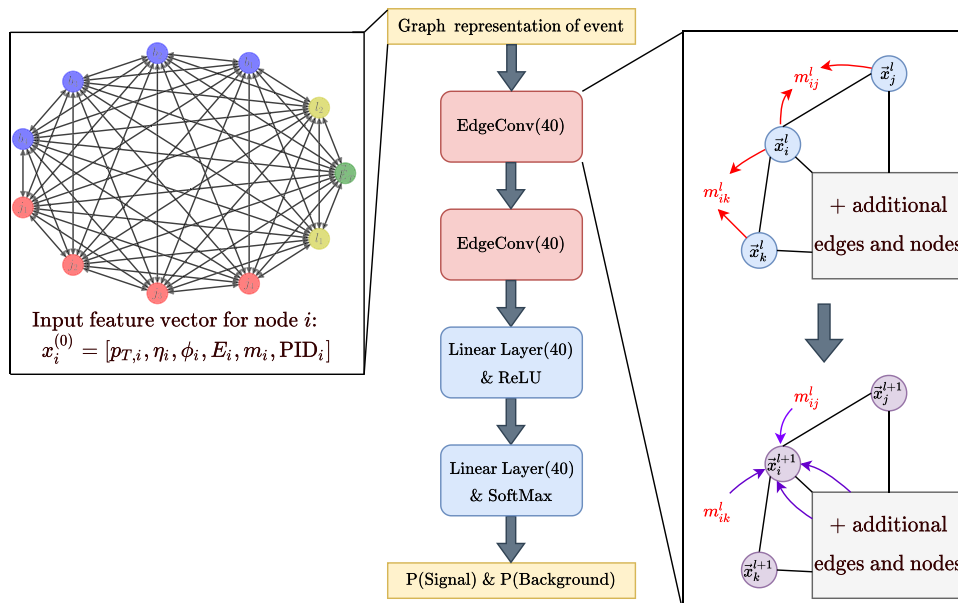
### **Anomaly Detection:**

Even though there are many theoretically motivated extensions of the Standard Model, the search for new physics can be regarded as an inherently model-independent endeavour, as nature need not fit into the constraints of our imagination. This motivates developing algorithms that do not rely on a specific BSM model but try to find deviations from the known background SM processes. General techniques exist in ML literature which are appropriate for model-independent searches called anomaly detection, wherein an algorithm focuses on learning the features of a well-understood class (the background in our case) and identifying anomalous data samples that do not have the same features.

A common way to achieve this unsupervised learning is via autoencoders. An autoencoder is an ANN used in unsupervised learning that aims to encode input data into a reduced-dimensional representation (called the latent representation) and then decode it back to its original form. The model learns by comparing the reconstructed output with the original input and tuning parameters to minimize the reconstruction error. Potential of autoencoder architectures have been demonstrated in simulating realistic and diverse aspects of high energy physics events. For instance, convolutional autoencoders were studied for detecting anomalous jet images in Refs. [166, 167].

GNNs rely on understanding the graph's topology and node connections to extract meaningful representations. Capturing this information accurately during the encoding and decoding phases to reconstruct the original graph structure is non-trivial. Particle Graph Autoencoder [22, 168] used an edge-convolution encoder block to map the node features to a two-dimensional latent node representation, which fed to the symmetrical decoder block to reconstruct the node features. For anomaly detection of events or jets, we are interested in the global features of the graph. Although a graph readout extracts this information in classification, the operation itself is destructive,

**Fig. 6** Example of a fully connected event graph (figure taken from Ref. [158])



and one cannot recover the graph structure in an autoencoder-based approach where the decoder needs to have the graph structure intact for its reconstruction. To address these challenges in graph autoencoders, Ref. [169] devised edge reconstruction networks, which were used to reconstruct weighted edge features, thereby learning the graph structure without undergoing a destructive graph readout operation. The Deep Sets autoencoder [170], on the other hand, relied solely on the encoded global latent space without using any decoder. Graph autoencoders with physics-motivated inductive biases like Lorentz group equivariance [171] and IRC-safety [99] have also been explored for anomaly detection.

## 6 Summary

In this review, we have laid down the rationale for designing automatic feature extractors from high-dimensional data in the context of high-energy physics. The main theme of such strategies is the balance between expressivity and generalisation power. While expressivity deals with the theoretical power of non-parametric models to capture a large set of functions accurately, generalisation power trades some of this expressivity for the practical gain of finding a good function for specific purposes based on prior knowledge of the use case.

The wealth of fundamental understanding of particle physics motivates their use in designing automatic feature extractors for various applications at the Large Hadron Collider. One recurrent theme from prior physics intuition is the suitability of the set-based point cloud and graph representation of collision events. Among different deep learning frameworks, graph neural networks possess a unique ability to learn from relational data. They are also adaptable to versatile data structures, capturing the essence of graphs characterised by node features and their relationships (features related to connected edges) along with graph features, if any. At the heart of a GNN lies a clever mechanism called message passing. Nodes exchange information with their neighbours, iteratively aggregating and updating their representations. Such operations allow a node to understand its self-attributes in light of the context provided by its connected neighbours, capturing complex relationships within the graph.

It is not an understatement to state that mainstream scientific inquiry is in a data-driven era, propelled by the development of powerful and data-hungry deep-learning algorithms. However, from the perspective of fundamental fields like particle physics, the main objective is to update our understanding of the universe based on empirical evidence. Focusing on their performance alone would be detrimental in the long term, and the common consensus in the community is to understand these algorithms in sufficient detail. Removing this obscurity is not just of academic interest but rather a practical requirement arising from the infamous mathematical complexity of such algorithms. The (potential) discovery of new physics at the Large Hadron Collider is an extreme statistical condition, and one needs to understand the algorithm's behaviour in such extreme conditions concretely.

**Acknowledgements** A.B. is supported by the U.S. Department of Energy under grant number DE-SC 0016013. V. S. N. is supported by the STFC under grant ST/P001246/1. The computational works are performed using the Param Vikram-1000 High-Performance Computing Cluster and the TDP project resources of the Physical Research Laboratory (PRL). Authors

would like to thank for the warm hospitality and support received at the school of Deep Machine Learning for particle and astroparticle physics (ML4HEP) at ICTS, Bengaluru (2023) and IOP, Bhubaneswar (2024).

**Data availability statement** No data associated in the manuscript.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019). <https://doi.org/10.1103/RevModPhys.91.045002>
2. J. Elmsheuser, Evolution of the ATLAS analysis model for Run-3 and prospects for HL-LHC. *EPJ Web Conf.* **245**, 06014 (2020). <https://doi.org/10.1051/epjconf/202024506014>
3. T. Aarrestad, HL-LHC Computing Review: Common Tools and Community Software. In: Canal, P., (eds.) *Snowmass 2021* (2020). <https://doi.org/10.5281/zenodo.4009114>
4. A. Held, E. Kauffman, O. Shadura, A. Wightman, Physics analysis for the HL-LHC: concepts and pipelines in practice with the Analysis Grand Challenge. In: *26th International Conference on Computing in High Energy & Nuclear Physics* (2024)
5. J. Motta, Overview of the HL-LHC Upgrade for the CMS Level-1 Trigger. *PoS EPS-HEP2023*, 534 (2024) <https://doi.org/10.22323/1.449.0534>
6. T.Q. Nguyen, D. Weitekamp, D. Anderson, R. Castello, O. Cerri, M. Pierini, M. Spiropulu, J.-R. Vlimant, Topology classification with deep learning to improve real-time event selection at the LHC. *Comput. Softw. Big Sci.* **3**(1), 12 (2019). <https://doi.org/10.1007/s41781-019-0028-1>. [arXiv:1807.00083](https://arxiv.org/abs/1807.00083) [hep-ex]
7. J. Duarte, Fast inference of deep neural networks in FPGAs for particle physics. *JINST* **13**(07), 07027 (2018). <https://doi.org/10.1088/1748-0221/13/07/P07027>. [arXiv:1804.06913](https://arxiv.org/abs/1804.06913) [physics.ins-det]
8. A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman, T. Plehn, D. Shih, R. Winterhalder, Ephemeral Learning - Augmenting Triggers with Online-Trained Normalizing Flows. *SciPost Phys.* **13**(4), 087 (2022). <https://doi.org/10.21468/SciPostPhys.13.4.087>. [arXiv:2202.09375](https://arxiv.org/abs/2202.09375) [hep-ph]
9. G. Bortolato, M. Cepeda, J. Heikkilä, B. Huber, E. Leutgeb, D. Rabady, H. Sakulin, Design and implementation of neural network based conditions for the CMS Level-1 Global Trigger upgrade for the HL-LHC. *JINST* **19**(03), 03019 (2024). <https://doi.org/10.1088/1748-0221/19/03/C03019>
10. M. Paganini, L. Oliveira, B. Nachman, Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters. *Phys. Rev. Lett.* **120**(4), 042003 (2018). <https://doi.org/10.1103/PhysRevLett.120.042003>. [arXiv:1705.02355](https://arxiv.org/abs/1705.02355) [hep-ex]
11. S. Farrell, Novel deep learning methods for track reconstruction. In: *4th International Workshop Connecting The Dots 2018* (2018)
12. S.R. Qasim, J. Kieseler, Y. Iiyama, M. Pierini, Learning representations of irregular particle-detector geometry with distance-weighted graph networks. *Eur. Phys. J. C* **79**(7), 608 (2019). <https://doi.org/10.1140/epjc/s10052-019-7113-9>. [arXiv:1902.07987](https://arxiv.org/abs/1902.07987) [physics.data-an]
13. Fast simulation of the ATLAS calorimeter system with Generative Adversarial Networks. Technical Report ATL-SOFT-PUB-2020-006, CERN, Geneva (Nov 2020). All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2020-006>. <http://cds.cern.ch/record/2746032>
14. E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger, Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed. *Comput. Softw. Big Sci.* **5**(1), 13 (2021). <https://doi.org/10.1007/s41781-021-00056-0>. [arXiv:2005.05334](https://arxiv.org/abs/2005.05334) [physics.ins-det]
15. C. Biscarat, S. Caillou, C. Rougier, J. Stark, J. Zahreddine, Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC. *EPJ Web Conf.* **251**, 03047 (2021). <https://doi.org/10.1051/epjconf/202125103047>. [arXiv:2103.00916](https://arxiv.org/abs/2103.00916) [physics.ins-det]
16. M. Fucci Giannelli, G. Kasieczka, B. Nachman, D. Salamani, D. Shih, A. Zaborowska, Fast Calorimeter Simulation Challenge 2022. <https://calochallenge.github.io/homepage/> (2022). <https://calochallenge.github.io/homepage/>
17. A. Adelmann, New directions for surrogate models and differentiable programming for High Energy Physics detector simulation. In: *Snowmass 2021* (2022)
18. R. Liu, P. Calafiura, S. Farrell, X. Ju, D.T. Murnane, T.M. Pham, Hierarchical Graph Neural Networks for Particle Track Reconstruction. In: *21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI Meets Reality* (2023)

19. P. Baldi, P. Sadowski, D. Whiteson, Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.* **5**, 4308 (2014). <https://doi.org/10.1038/ncomms5308>. arXiv:1402.4735 [hep-ph]
20. L.M. Dery, B. Nachman, F. Rubbo, A. Schwartzman, Weakly Supervised Classification in High Energy Physics. *JHEP* **05**, 145 (2017). [https://doi.org/10.1007/JHEP05\(2017\)145](https://doi.org/10.1007/JHEP05(2017)145). arXiv:1702.00414 [hep-ph]
21. E.M. Metodiev, B. Nachman, J. Thaler, Classification without labels: Learning from mixed samples in high energy physics. *JHEP* **10**, 174 (2017). [https://doi.org/10.1007/JHEP10\(2017\)174](https://doi.org/10.1007/JHEP10(2017)174). arXiv:1708.02949 [hep-ph]
22. G. Kasieczka, The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics. *Rept. Prog. Phys.* **84**(12), 124201 (2021). <https://doi.org/10.1088/1361-6633/ac36b9>. arXiv:2101.08320 [hep-ph]
23. G. Aad, Dijet resonance search with weak supervision using  $\sqrt{s} = 13$  TeV  $pp$  collisions in the ATLAS detector. *Phys. Rev. Lett.* **125**(13), 131801 (2020). <https://doi.org/10.1103/PhysRevLett.125.131801>. arXiv:2005.02983 [hep-ex]
24. T. Aarrestad, The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider. *SciPost Phys.* **12**(1), 043 (2022). <https://doi.org/10.21468/SciPostPhys.12.1.043>. arXiv:2105.14027 [hep-ph]
25. A. Hallin, G. Kasieczka, T. Quadfasel, D. Shih, M. Sommerhalder, Resonant anomaly detection without background sculpting. *Phys. Rev. D* **107**(11), 114012 (2023). <https://doi.org/10.1103/PhysRevD.107.114012>. arXiv:2210.14924 [hep-ph]
26. G. Aad, Search for New Phenomena in Two-Body Invariant Mass Distributions Using Unsupervised Machine Learning for Anomaly Detection at  $s=13$  TeV with the ATLAS Detector. *Phys. Rev. Lett.* **132**(8), 081801 (2024). <https://doi.org/10.1103/PhysRevLett.132.081801>. arXiv:2307.01612 [hep-ex]
27. V.S. Ngairangbam, M. Spannowsky, Interpretable deep learning models for the inference and classification of LHC data (2023) arXiv:2312.12330 [hep-ph]
28. Model-agnostic search for dijet resonances with anomalous jet substructure in proton-proton collisions at  $\sqrt{s} = 13$  TeV (2024)
29. T. Finke, M. Krämer, A. Mück, J. Tönshoff, Learning the language of QCD jets with transformers. *JHEP* **06**, 184 (2023). [https://doi.org/10.1007/JHEP06\(2023\)184](https://doi.org/10.1007/JHEP06(2023)184). arXiv:2303.07364 [hep-ph]
30. A. Butter, N. Huetsch, S. Palacios Schweitzer, T. Plehn, P. Sorrenson, J. Spinner, Jet Diffusion versus JetGPT – Modern Networks for the LHC (2023) arXiv:2305.10475 [hep-ph]
31. M. Vigl, N. Hartman, L. Heinrich, Finetuning Foundation Models for Joint Analysis Optimization (2024) arXiv:2401.13536 [hep-ex]
32. L. Heinrich, T. Golling, M. Kagan, S. Klein, M. Leigh, M. Osadchy, J.A. Raine, Masked Particle Modeling on Sets: Towards Self-Supervised High Energy Physics Foundation Models (2024) arXiv:2401.13537 [hep-ph]
33. J. Birk et al., OmniJet- $\alpha$ : the first cross-task foundation model for particle physics. *Mach. Learn.: Sci. Technol.* **5**, 035031 (2024). <https://doi.org/10.1088/2632-2153/ad66ad>
34. P. Harris, M. Kagan, J. Krupa, B. Maier, N. Woodward, Re-Simulation-based Self-Supervised Learning for Pre-Training Foundation Models (2024) arXiv:2403.07066 [hep-ph]
35. G. Louppe, M. Kagan, K. Cranmer, Learning to pivot with adversarial networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., ??? (2017). [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/48ab2f9b45957ab574cf005eb8a76760-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/48ab2f9b45957ab574cf005eb8a76760-Paper.pdf)
36. J.Y. Araz, M. Spannowsky, Combine and Conquer: Event Reconstruction with Bayesian Ensemble Neural Networks. *JHEP* **04**, 296 (2021). [https://doi.org/10.1007/JHEP04\(2021\)296](https://doi.org/10.1007/JHEP04(2021)296). arXiv:2102.01078 [hep-ph]
37. A. Ghosh, B. Nachman, D. Whiteson, Uncertainty-aware machine learning for high energy physics. *Phys. Rev. D* **104**(5), 056026 (2021). <https://doi.org/10.1103/PhysRevD.104.056026>. arXiv:2105.08742 [physics.data-an]
38. T.Y. Chen, B. Dey, A. Ghosh, M. Kagan, B. Nord, N. Ramachandra, Interpretable Uncertainty Quantification in AI for HEP. In: *Snowmass 2021* (2022). <https://doi.org/10.2172/1886020>
39. A. Golutvin, A. Iniukhin, A. Mauri, P. Owen, N. Serra, A. Ustyuzhanin, The DL Advocate: playing the devil's advocate with hidden systematic uncertainties. *Eur. Phys. J. C* **83**(9), 779 (2023). <https://doi.org/10.1140/epjc/s10052-023-11925-w>. arXiv:2303.15956 [hep-ex]
40. A. Ghosh, B. Nachman, T. Plehn, L. Shire, T.M.P. Tait, D. Whiteson, Statistical patterns of theory uncertainties. *SciPost Phys. Core* **6**, 045 (2023). <https://doi.org/10.21468/SciPostPhysCore.6.2.045>. arXiv:2210.15167 [hep-ph]
41. A. Ghosh, B. Nachman, A cautionary tale of decorrelating theory uncertainties. *Eur. Phys. J. C* **82**(1), 46 (2022). <https://doi.org/10.1140/epjc/s10052-022-10012-w>. arXiv:2109.08159 [hep-ph]
42. A.J. Larkoski, I. Moulton, B. Nachman, Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning. *Phys. Rept.* **841**, 1–63 (2020). <https://doi.org/10.1016/j.physrep.2019.11.001>. arXiv:1709.04464 [hep-ph]
43. A. Butter, T. Plehn, Chapter 7. Generative Networks for LHC Events, pp. 191–240. [https://doi.org/10.1142/9789811234033\\_0007](https://doi.org/10.1142/9789811234033_0007). [https://worldscientific.com/doi/abs/10.1142/9789811234033\\_0007](https://worldscientific.com/doi/abs/10.1142/9789811234033_0007)
44. J. Duarte, J.-R. Vlimant, Graph Neural Networks for Particle Tracking and Reconstruction (2020) [https://doi.org/10.1142/9789811234033\\_0012](https://doi.org/10.1142/9789811234033_0012) arXiv:2012.01249 [hep-ph]
45. J. Shlomi, P. Battaglia, J.-R. Vlimant, Graph Neural Networks in Particle Physics (2020) <https://doi.org/10.1088/2632-2153/abbf9a> arXiv:2007.13681 [hep-ex]



46. S. Thais, P. Calafiura, G. Chachamis, G. DeZoort, J. Duarte, S. Ganguly, M. Kagan, D. Murnane, M.S. Neubauer, K. Terao, Graph Neural Networks in Particle Physics: Implementations, Innovations, and Challenges. In: Snowmass 2021 (2022)
47. V. Belis, P. Odagiu, T.K. Aarrestad, Machine learning for anomaly detection in particle physics. *Rev. Phys.* **12**, 100091 (2024). <https://doi.org/10.1016/j.revip.2024.100091>. arXiv:2312.14190 [physics.data-an]
48. R.L. Workman, Others: Review of Particle Physics. *PTEP* **2022**, 083–01 (2022). <https://doi.org/10.1093/ptep/ptac097>
49. G. DeZoort, P.W. Battaglia, C. Biscarat, J.-R. Vlimant, Graph neural networks at the large hadron collider. *Nature Reviews Physics* **5**(5), 281–303 (2023). <https://doi.org/10.1038/s42254-023-00569-0>
50. B. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn, T. Kuhr, Author Correction: Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning [<https://doi.org/10.1038/s41467-024-49104-4>]. *Nature Commun.* **15**(1), 5825 (2024) <https://doi.org/10.1038/s41467-024-49971-xarXiv:2303.08046> [physics.ins-det]
51. B. Hashemi, C. Krause, Deep generative models for detector signature simulation: A taxonomic review. *Rev. Phys.* **12**, 100092 (2024). <https://doi.org/10.1016/j.revip.2024.100092>. arXiv:2312.09597 [physics.ins-det]
52. M. Feickert, B. Nachman, A Living Review of Machine Learning for Particle Physics (2021) arXiv:2102.02770 [hep-ph]
53. W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943). <https://doi.org/10.1007/BF02478259>
54. F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**(6), 386–408 (1958). <https://doi.org/10.1037/h0042519>
55. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterton, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (2010). <https://proceedings.mlr.press/v9/glorot10a.html>
56. K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015). <https://doi.org/10.1109/ICCV.2015.123>
57. M.V. Narkhede, P.P. Bartakke, M.S. Sutaone, A review on weight initialization strategies for neural networks. *Artif. Intell. Rev.* **55**(1), 291–322 (2022). <https://doi.org/10.1007/s10462-021-10033-z>
58. P.J. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard University, ??? (1975). <https://books.google.co.in/books?id=r3uljgEACAAJ>
59. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**(5), 359–366 (1989). [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
60. G. Cybenko, Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems* **2**(4), 303–314 (1989). <https://doi.org/10.1007/BF02551274>
61. M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **6**(6), 861–867 (1993). [https://doi.org/10.1016/S0893-6080\(05\)80131-5](https://doi.org/10.1016/S0893-6080(05)80131-5)
62. F. Voigtlaender, The universal approximation theorem for complex-valued neural networks. *Appl. Comput. Harmon. Anal.* **64**, 33–61 (2023). <https://doi.org/10.1016/j.acha.2022.12.002>
63. Y. Bengio, Lecun, Y., in *Scaling learning algorithms towards AI*. ed. by L. Bottou, O. Chapelle, D. DeCoste, J. Weston (Neural Information Processing series. MIT Press, London, England, 2007)
64. J. Håstad, *Computational Limitations of Small-depth Circuits* (MIT Press, Cambridge, MA, USA, 1987)
65. E. Allender, Circuit complexity before the dawn of the new millennium, in *Foundations of Software Technology and Theoretical Computer Science*. ed. by V. Chandru, V. Vinay (Springer, Berlin, Heidelberg, 1996), pp.1–18
66. M. Bianchini, F. Scarselli, On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems* **25**(8), 1553–1565 (2014). <https://doi.org/10.1109/TNNLS.2013.2293637>
67. R. Eldan, O. Shamir, The power of depth for feedforward neural networks. In: Feldman, V., Rakhlin, A., Shamir, O. (eds.) 29th Annual Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 49, pp. 907–940. PMLR, Columbia University, New York, New York, USA (2016). <https://proceedings.mlr.press/v49/eldan16.html>
68. G. Naitzat, A. Zhitnikov, L.-H. Lim, Topology of deep neural networks. *J. Mach. Learn. Res.* **21**(184), 1–40 (2020)
69. D.H. Wolpert, On the connection between in-sample testing and generalization error. *Complex Syst.* **6** (1992)
70. D.H. Wolpert, The lack of a priori distinctions between learning algorithms. *Neural Comput.* **8**(7), 1341–1390 (1996). <https://doi.org/10.1162/neco.1996.8.7.1341>
71. C. Schaffer, A conservation law for generalization performance, in *Machine Learning Proceedings 1994*. ed. by W.W. Cohen, H. Hirsh (Morgan Kaufmann, San Francisco (CA), 1994), pp.259–265. <https://doi.org/10.1016/B978-1-55860-335-6.50039-8>
72. D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997). <https://doi.org/10.1109/4235.585893>
73. D.H. Wolpert, In: Pardalos, P.M., Rasskazova, V., Vrahatis, M.N. (eds.) What Is Important About the No Free Lunch Theorems?, pp. 373–388. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-66515-9\\_13](https://doi.org/10.1007/978-3-030-66515-9_13)
74. B. Denby, Neural networks and cellular automata in experimental high energy physics. *Comput. Phys. Commun.* **49**(3), 429–448 (1988). [https://doi.org/10.1016/0010-4655\(88\)90004-5](https://doi.org/10.1016/0010-4655(88)90004-5)



75. L. Lönnblad, C. Peterson, T. Rognvaldsson, Finding gluon jets with a neural trigger. *Phys. Rev. Lett.* **65**, 1321–1324 (1990). <https://doi.org/10.1103/PhysRevLett.65.1321>
76. A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, Machine learning at the energy and intensity frontiers of particle physics. *Nature* **560**(7716), 41–48 (2018). <https://doi.org/10.1038/s41586-018-0361-2>
77. L. Moore, K. Nordström, S. Varma, M. Fairbairn, Reports of My Demise Are Greatly Exaggerated:  $N$ -subjettiness Taggers Take On Jet Images. *SciPost Phys.* **7**(3), 036 (2019). <https://doi.org/10.21468/SciPostPhys.7.3.036>. [arXiv:1807.04769](https://arxiv.org/abs/1807.04769) [hep-ph]
78. T. Fausett, J. Thaler, D. Whiteson, Mapping Machine-Learned Physics into a Human-Readable Space. *Phys. Rev. D* **103**(3), 036020 (2021). <https://doi.org/10.1103/PhysRevD.103.036020>. [arXiv:2010.11998](https://arxiv.org/abs/2010.11998) [hep-ph]
79. M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.* **34**(4), 18–42 (2017). <https://doi.org/10.1109/MSP.2017.2693418>
80. M.M. Bronstein, J. Bruna, T. Cohen, P. Veličković, *Geometric Deep Learning: Grids, Groups* (Geodesics, and Gauges, Graphs, 2021)
81. T. Cohen, M. Welling, Group equivariant convolutional networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) *Proceedings of The 33rd International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 48, pp. 2990–2999. PMLR, New York, New York, USA (2016). <https://proceedings.mlr.press/v48/cohenc16.html>
82. T. Cohen, M. Weiler, B. Kicanaoglu, M. Welling, Gauge equivariant convolutional networks and the icosahedral CNN. In: Chaudhuri, K., Salakhutdinov, R. (eds.) *Proceedings of the 36th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 97, pp. 1321–1330. PMLR, ??? (2019). <https://proceedings.mlr.press/v97/cohen19d.html>
83. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S.A.A. Kohl, A.J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A.W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with alphafold. *Nature* **596**(7873), 583–589 (2021). <https://doi.org/10.1038/s41586-021-03819-2>
84. G. Kanwar, M.S. Albergo, D. Boyda, K. Cranmer, D.C. Hackett, S. Racanière, D.J. Rezende, P.E. Shanahan, Equivariant flow-based sampling for lattice gauge theory. *Phys. Rev. Lett.* **125**(12), 121601 (2020). <https://doi.org/10.1103/PhysRevLett.125.121601>. [arXiv:2003.06413](https://arxiv.org/abs/2003.06413) [hep-lat]
85. M. Favoni, A. Ipp, D.I. Müller, D. Schuh, Lattice Gauge Equivariant Convolutional Neural Networks. *Phys. Rev. Lett.* **128**(3), 032003 (2022). <https://doi.org/10.1103/PhysRevLett.128.032003>. [arXiv:2012.12901](https://arxiv.org/abs/2012.12901) [hep-lat]
86. A. Bogatskiy, B. Anderson, J.T. Offermann, M. Roussi, D.W. Miller, R. Kondor, Lorentz Group Equivariant Neural Network for Particle Physics (2020) [arXiv:2006.04780](https://arxiv.org/abs/2006.04780) [hep-ph]
87. E. Buhmann, G. Kasieczka, J. Thaler, EPiC-GAN: Equivariant point cloud generation for particle jets. *SciPost Phys.* **15**(4), 130 (2023). <https://doi.org/10.21468/SciPostPhys.15.4.130>. [arXiv:2301.08128](https://arxiv.org/abs/2301.08128) [hep-ph]
88. P. Onyisi, D. Shen, J. Thaler, Comparing point cloud strategies for collider event classification. *Phys. Rev. D* **108**(1), 012001 (2023). <https://doi.org/10.1103/PhysRevD.108.012001>. [arXiv:2212.10659](https://arxiv.org/abs/2212.10659) [hep-ph]
89. C. Frye, A.J. Larkoski, J. Thaler, K. Zhou, Casimir Meets Poisson: Improved Quark/Gluon Discrimination with Counting Observables. *JHEP* **09**, 083 (2017). [https://doi.org/10.1007/JHEP09\(2017\)083](https://doi.org/10.1007/JHEP09(2017)083). [arXiv:1704.06266](https://arxiv.org/abs/1704.06266) [hep-ph]
90. R. Medves, A. Soto-Ontoso, G. Soyez, Lund and Cambridge multiplicities for precision physics. *JHEP* **10**, 156 (2022). [https://doi.org/10.1007/JHEP10\(2022\)156](https://doi.org/10.1007/JHEP10(2022)156). [arXiv:2205.02861](https://arxiv.org/abs/2205.02861) [hep-ph]
91. R. Medves, A. Soto-Ontoso, G. Soyez, Lund multiplicity in QCD jets. *JHEP* **04**, 104 (2023). [https://doi.org/10.1007/JHEP04\(2023\)104](https://doi.org/10.1007/JHEP04(2023)104). [arXiv:2212.05076](https://arxiv.org/abs/2212.05076) [hep-ph]
92. S. Choi, S.J. Lee, M. Perelstein, Infrared Safety of a Neural-Net Top Tagging Algorithm. *JHEP* **02**, 132 (2019). [https://doi.org/10.1007/JHEP02\(2019\)132](https://doi.org/10.1007/JHEP02(2019)132). [arXiv:1806.01263](https://arxiv.org/abs/1806.01263) [hep-ph]
93. P.T. Komiske, E.M. Metodiev, J. Thaler, Energy Flow Networks: Deep Sets for Particle Jets. *JHEP* **01**, 121 (2019). [https://doi.org/10.1007/JHEP01\(2019\)121](https://doi.org/10.1007/JHEP01(2019)121). [arXiv:1810.05165](https://arxiv.org/abs/1810.05165) [hep-ph]
94. M.J. Dolan, A. Ore, Equivariant Energy Flow Networks for Jet Tagging. *Phys. Rev. D* **103**(7), 074022 (2021). <https://doi.org/10.1103/PhysRevD.103.074022>. [arXiv:2012.00964](https://arxiv.org/abs/2012.00964) [hep-ph]
95. W. Shen, D. Wang, J.M. Yang, Hierarchical high-point Energy Flow Network for jet tagging. *JHEP* **09**, 135 (2023). [https://doi.org/10.1007/JHEP09\(2023\)135](https://doi.org/10.1007/JHEP09(2023)135). [arXiv:2308.08300](https://arxiv.org/abs/2308.08300) [hep-ph]
96. S. Bright-Thonney, B. Nachman, J. Thaler, Safe but Incalculable: Energy-weighting is not all you need (2023) [arXiv:2311.07652](https://arxiv.org/abs/2311.07652) [hep-ph]
97. R. Gambhir, A. Osathapan, J. Thaler, Moments of Clarity: Streamlining Latent Spaces in Machine Learning using Moment Pooling (2024) [arXiv:2403.08854](https://arxiv.org/abs/2403.08854) [hep-ph]
98. P. Konar, V.S. Ngairangbam, M. Spannowsky, Energy-weighted message passing: an infra-red and collinear safe graph neural network algorithm. *JHEP* **02**, 060 (2022). [https://doi.org/10.1007/JHEP02\(2022\)060](https://doi.org/10.1007/JHEP02(2022)060). [arXiv:2109.14636](https://arxiv.org/abs/2109.14636) [hep-ph]
99. O. Atkinson, A. Bhardwaj, C. Englert, P. Konar, V.S. Ngairangbam, M. Spannowsky, IRC-Safe Graph Autoencoder for Unsupervised Anomaly Detection. *Front. Artif. Intell.* **5**, 943135 (2022). <https://doi.org/10.3389/frai.2022.943135>. [arXiv:2204.12231](https://arxiv.org/abs/2204.12231) [hep-ph]
100. P. Konar, V.S. Ngairangbam, M. Spannowsky, Hypergraphs in LHC phenomenology – the next frontier of IRC-safe feature extraction. *JHEP* **01**, 113 (2024). [https://doi.org/10.1007/JHEP01\(2024\)113](https://doi.org/10.1007/JHEP01(2024)113). [arXiv:2309.17351](https://arxiv.org/abs/2309.17351) [hep-ph]

101. S. Chatterjee, S.S. Cruz, R. Schöfbeck, D. Schwarz, A rotation-equivariant graph neural network for learning hadronic SMEFT effects (2024) [arXiv:2401.10323](https://arxiv.org/abs/2401.10323) [hep-ph]
102. A. Bhardwaj, C. Englert, W. Naskar, V.S. Ngairangbam, M. Spannowsky, Equivariant, Safe and Sensitive – Graph Networks for New Physics (2024) [arXiv:2402.12449](https://arxiv.org/abs/2402.12449) [hep-ph]
103. M. Cacciari, G.P. Salam, G. Soyez, The anti- $k_t$  jet clustering algorithm. *JHEP* **04**, 063 (2008). <https://doi.org/10.1088/1126-6708/2008/04/063>. [arXiv:0802.1189](https://arxiv.org/abs/0802.1189) [hep-ph]
104. B. Andrieu, Jet finding algorithms at Tevatron. *Acta Phys. Polon. B* **36**, 409–415 (2005)
105. G.P. Salam, Towards Jetography. *Eur. Phys. J. C* **67**, 637–686 (2010). <https://doi.org/10.1140/epjc/s10052-010-1314-6>. [arXiv:0906.1833](https://arxiv.org/abs/0906.1833) [hep-ph]
106. R.K. Ellis, K. Melnikov, G. Zanderighi, W+3 jet production at the Tevatron. *Phys. Rev. D* **80**, 094002 (2009). <https://doi.org/10.1103/PhysRevD.80.094002>. [arXiv:0906.1445](https://arxiv.org/abs/0906.1445) [hep-ph]
107. A. Banfi, G.P. Salam, G. Zanderighi, Principles of general final-state resummation and automated implementation. *JHEP* **03**, 073 (2005). <https://doi.org/10.1088/1126-6708/2005/03/073>. [arXiv:hep-ph/0407286](https://arxiv.org/abs/hep-ph/0407286)
108. M. Dasgupta, A. Fregoso, S. Marzani, G.P. Salam, Towards an understanding of jet substructure. *JHEP* **09**, 029 (2013). [https://doi.org/10.1007/JHEP09\(2013\)029](https://doi.org/10.1007/JHEP09(2013)029). [arXiv:1307.0007](https://arxiv.org/abs/1307.0007) [hep-ph]
109. G. Kasieczka, S. Marzani, G. Soyez, G. Stagnitto, Towards Machine Learning Analytics for Jet Substructure. *JHEP* **09**, 195 (2020). [https://doi.org/10.1007/JHEP09\(2020\)195](https://doi.org/10.1007/JHEP09(2020)195). [arXiv:2007.04319](https://arxiv.org/abs/2007.04319) [hep-ph]
110. H.-M. Chang, M. Procura, J. Thaler, W.J. Waalewijn, Calculating Track-Based Observables for the LHC. *Phys. Rev. Lett.* **111**, 102002 (2013). <https://doi.org/10.1103/PhysRevLett.111.102002>. [arXiv:1303.6637](https://arxiv.org/abs/1303.6637) [hep-ph]
111. A.J. Larkoski, S. Marzani, J. Thaler, Sudakov Safety in Perturbative QCD. *Phys. Rev. D* **91**(11), 111501 (2015). <https://doi.org/10.1103/PhysRevD.91.111501>. [arXiv:1502.01719](https://arxiv.org/abs/1502.01719) [hep-ph]
112. K. Fukushima, S. Miyake, Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recogn.* **15**(6), 455–469 (1982). [https://doi.org/10.1016/0031-3203\(82\)90024-3](https://doi.org/10.1016/0031-3203(82)90024-3)
113. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**(4), 541–551 (1989). <https://doi.org/10.1162/neco.1989.1.4.541>
114. M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R.R. Salakhutdinov, A.J. Smola, Deep sets. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., ??? (2017). [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf)
115. M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 729–7342 (2005). <https://doi.org/10.1109/IJCNN.2005.1555942>
116. F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model. *IEEE Trans. Neural Networks* **20**(1), 61–80 (2009). <https://doi.org/10.1109/TNN.2008.2005605>
117. A. Sperduti, A. Starita, Supervised neural networks for the classification of structures. *IEEE Trans. Neural Networks* **8**(3), 714–735 (1997). <https://doi.org/10.1109/72.572108>
118. P. Frasconi, M. Gori, A. Sperduti, A general framework for adaptive processing of data structures. *IEEE Trans. Neural Networks* **9**(5), 768–786 (1998). <https://doi.org/10.1109/72.712151>
119. J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry. In: Precup, D., Teh, Y.W. (eds.) *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 70, pp. 1263–1272. PMLR, ??? (2017). <https://proceedings.mlr.press/v70/gilmer17a.html>
120. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V.F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Song, H.F., Ballard, A.J., Gilmer, J., Dahl, G.E., Vaswani, A., Allen, K.R., Nash, C., Langston, V., Dyer, C., Heess, N.M.O., Wierstra, D., Kohli, P., Botvinick, M.M., Vinyals, O., Li, Y., Pascanu, R.: Relational inductive biases, deep learning, and graph networks. *ArXiv abs/1806.01261* (2018)
121. Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **38**(5) (2019) <https://doi.org/10.1145/3326362>
122. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate. In: Bengio, Y., LeCun, Y. (eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings* (2015). <http://arxiv.org/abs/1409.0473>
123. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., ??? (2017). [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
124. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). <https://doi.org/10.18653/v1/N19-1423>
125. T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D.

- Amodei, Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc., ??? (2020). [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf)
126. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P.F. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744. Curran Associates, Inc., ??? (2022). [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf)
  127. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training (2018)
  128. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks. In: *International Conference on Learning Representations* (2018). <https://openreview.net/forum?id=rJXMpikCZ>
  129. H. Zhao, L. Jiang, J. Jia, P. Torr, V. Koltun, Point transformer. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16239–16248 (2021). <https://doi.org/10.1109/ICCV48922.2021.01595>
  130. L. Oliveira, M. Kagan, L. Mackey, B. Nachman, A. Schwartzman, Jet-images — deep learning edition. *JHEP* 07, 069 (2016) [https://doi.org/10.1007/JHEP07\(2016\)069](https://doi.org/10.1007/JHEP07(2016)069) arXiv:1511.05190 [hep-ph]
  131. H. Qu, L. Gouskos, ParticleNet: Jet Tagging via Particle Clouds. *Phys. Rev. D* **101**(5), 056019 (2020). <https://doi.org/10.1103/PhysRevD.101.056019>. arXiv:1902.08570 [hep-ph]
  132. E.A. Moreno, O. Cerri, J.M. Duarte, H.B. Newman, T.Q. Nguyen, A. Periwai, M. Pierini, A. Serikova, M. Spiropulu, J.-R. Vlimant, JEDI-net: a jet identification algorithm based on interaction networks. *Eur. Phys. J. C* **80**(1), 58 (2020). <https://doi.org/10.1140/epjc/s10052-020-7608-4>. arXiv:1908.05318 [hep-ex]
  133. E.A. Moreno, T.Q. Nguyen, J.-R. Vlimant, O. Cerri, H.B. Newman, A. Periwai, M. Spiropulu, J.M. Duarte, M. Pierini, Interaction networks for the identification of boosted  $H \rightarrow b\bar{b}$  decays. *Phys. Rev. D* **102**(1), 012010 (2020). <https://doi.org/10.1103/PhysRevD.102.012010>. arXiv:1909.12285 [hep-ex]
  134. V. Mikuni, F. Canelli, ABCNet: An attention-based method for particle tagging. *Eur. Phys. J. Plus* **135**(6), 463 (2020). <https://doi.org/10.1140/epjp/s13360-020-00497-3>. arXiv:2001.05311 [physics.data-an]
  135. M. He, D. Wang, Quark/gluon discrimination and top tagging with dual attention transformer. *Eur. Phys. J. C* **83**(12), 1116 (2023). <https://doi.org/10.1140/epjc/s10052-023-12293-1>. arXiv:2307.04723 [hep-ph]
  136. M. He, Z. Wei, J.-R. Wen, Convolutional neural networks on graphs with chebyshev approximation, revisited. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 35, pp. 7264–7276. Curran Associates, Inc., ??? (2022). [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/2f9b3ee2bcea04b327c09d7e3145bd1e-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/2f9b3ee2bcea04b327c09d7e3145bd1e-Paper-Conference.pdf)
  137. Y. Semrani, M. Relan, K. Ramesh, PCN: A Deep Learning Approach to Jet Tagging Utilizing Novel Graph Construction Methods and Chebyshev Graph Convolutions (2023) arXiv:2309.08630 [hep-ph]
  138. Y.G. Wang, M. Li, Z. Ma, G. Montufar, X. Zhuang, Y. Fan, Haar graph pooling. In: III, H.D., Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 119, pp. 9952–9962. PMLR, ??? (2020). <https://proceedings.mlr.press/v119/wang20m.html>
  139. F. Ma, F. Liu, W. Li, Jet tagging algorithm of graph network with Haar pooling message passing. *Phys. Rev. D* **108**(7), 072007 (2023). <https://doi.org/10.1103/PhysRevD.108.072007>. arXiv:2210.13869 [hep-ex]
  140. H. Qu, C. Li, S. Qian, Particle transformer for jet tagging. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) *Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 162, pp. 18281–18292. PMLR, ??? (2022). <https://proceedings.mlr.press/v162/qu22b.html>
  141. S. Shleifer, M. Ott, NormFormer: Improved Transformer Pretraining with Extra Normalization (2022). <https://openreview.net/forum?id=GMYWzWztDx5>
  142. V. Mikuni, F. Canelli, Point cloud transformers applied to collider physics. *Mach. Learn. Sci. Tech.* **2**(3), 035027 (2021). <https://doi.org/10.1088/2632-2153/ac07f6>. arXiv:2102.05073 [physics.data-an]
  143. S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian, W. Du, Z.-M. Ma, T.-Y. Liu, An efficient Lorentz equivariant graph neural network for jet tagging. *JHEP* **07**, 030 (2022). [https://doi.org/10.1007/JHEP07\(2022\)030](https://doi.org/10.1007/JHEP07(2022)030). arXiv:2201.08187 [hep-ph]
  144. C. Li, H. Qu, S. Qian, Q. Meng, S. Gong, J. Zhang, T.-Y. Liu, Q. Li, Does Lorentz-symmetric design boost network performance in jet physics? *Phys. Rev. D* **109**(5), 056003 (2024). <https://doi.org/10.1103/PhysRevD.109.056003>. arXiv:2208.07814 [hep-ph]
  145. A. Bogatskiy, T. Hoffman, D.W. Miller, J.T. Offermann, PELICAN: Permutation Equivariant and Lorentz Invariant or Covariant Aggregator Network for Particle Physics (2022) arXiv:2211.00454 [hep-ph]
  146. A. Bogatskiy, T. Hoffman, D.W. Miller, J.T. Offermann, X. Liu, Explainable equivariant neural networks for particle physics: PELICAN. *JHEP* **03**, 113 (2024). [https://doi.org/10.1007/JHEP03\(2024\)113](https://doi.org/10.1007/JHEP03(2024)113). arXiv:2307.16506 [hep-ph]
  147. R. Sahu, CapsLorentzNet: Integrating Physics Inspired Features with Graph Convolution (2024) arXiv:2403.11826 [hep-ph]
  148. P.T. Komiske, E.M. Metodiev, J. Thaler, The Hidden Geometry of Particle Collisions. *JHEP* **07**, 006 (2020). [https://doi.org/10.1007/JHEP07\(2020\)006](https://doi.org/10.1007/JHEP07(2020)006). arXiv:2004.04159 [hep-ph]
  149. E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer, A. Mück, Casting a graph net to catch dark showers. *SciPost Phys.* **10**(2), 046 (2021). <https://doi.org/10.21468/SciPostPhys.10.2.046>. arXiv:2006.08639 [hep-ph]

150. B. Andersson, G. Gustafson, L. Lönnblad, U. Petterson, Coherence effects in deep inelastic scattering. *Zeitschrift für Physik C Particles and Fields* **43**(4), 625–632 (1989). <https://doi.org/10.1007/BF01550942>
151. F.A. Dreyer, G.P. Salam, G. Soyez, The Lund Jet Plane. *JHEP* **12**, 064 (2018). [https://doi.org/10.1007/JHEP12\(2018\)064](https://doi.org/10.1007/JHEP12(2018)064). arXiv:1807.04758 [hep-ph]
152. F.A. Dreyer, H. Qu, Jet tagging in the Lund plane with graph networks. *JHEP* **03**, 052 (2021). [https://doi.org/10.1007/JHEP03\(2021\)052](https://doi.org/10.1007/JHEP03(2021)052). arXiv:2012.08526 [hep-ph]
153. F.A. Dreyer, G. Soyez, A. Takacs, Quarks and gluons in the Lund plane. *JHEP* **08**, 177 (2022). [https://doi.org/10.1007/JHEP08\(2022\)177](https://doi.org/10.1007/JHEP08(2022)177). arXiv:2112.09140 [hep-ph]
154. O. Atkinson, A. Bhardwaj, S. Brown, C. Englert, D.J. Miller, P. Stylianou, Improved constraints on effective top quark interactions using edge convolution networks. *JHEP* **04**, 137 (2022). [https://doi.org/10.1007/JHEP04\(2022\)137](https://doi.org/10.1007/JHEP04(2022)137). arXiv:2111.01838 [hep-ph]
155. M. Abdughani, J. Ren, L. Wu, J.M. Yang, Probing stop pair production at the LHC with graph neural networks. *JHEP* **08**, 055 (2019). [https://doi.org/10.1007/JHEP08\(2019\)055](https://doi.org/10.1007/JHEP08(2019)055). arXiv:1807.09088 [hep-ph]
156. J. Pata, J. Duarte, J.-R. Vlimant, M. Pierini, M. Spiropulu, MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks. *Eur. Phys. J. C* **81**(5), 381 (2021). <https://doi.org/10.1140/epjc/s10052-021-09158-w>. arXiv:2101.08578 [physics.data-an]
157. Atkinson, O. Anisha, A. Bhardwaj, C. Englert, P. Stylianou, Quartic Gauge-Higgs couplings: constraints and future directions. *JHEP* **10**, 172 (2022) [https://doi.org/10.1007/JHEP10\(2022\)172](https://doi.org/10.1007/JHEP10(2022)172) arXiv:2208.09334 [hep-ph]
158. Atkinson O. Anisha A. Bhardwaj, C. Englert, W. Naskar, P. Stylianou, BSM reach of four-top production at the LHC. *Phys. Rev. D* **108**(3), 035001 (2023) <https://doi.org/10.1103/PhysRevD.108.035001> arXiv:2302.08281 [hep-ph]
159. A. Hammad, S. Moretti, M. Nojiri, Multi-scale cross-attention transformer encoder for event classification. *JHEP* **03**, 144 (2024). [https://doi.org/10.1007/JHEP03\(2024\)144](https://doi.org/10.1007/JHEP03(2024)144). arXiv:2401.00452 [hep-ph]
160. P. Stylianou, G. Weiglein, Constraints on the trilinear and quartic Higgs couplings from triple Higgs production at the LHC and beyond (2023) arXiv:2312.04646 [hep-ph]
161. L. Ehrke, J.A. Raine, K. Zoch, M. Guth, T. Golling, Topological reconstruction of particle physics processes using graph neural networks. *Phys. Rev. D* **107**(11), 116019 (2023). <https://doi.org/10.1103/PhysRevD.107.116019>. arXiv:2303.13937 [hep-ph]
162. C. Birch-Sykes, B. Le, Y. Peters, E. Simpson, Z. Zhang, Reconstruction of Short-Lived Particles using Graph-Hypergraph Representation Learning (2024) arXiv:2402.10149 [hep-ph]
163. A. Mullin, S. Nicholls, H. Pacey, M. Parker, M. White, S. Williams, Does SUSY have friends? A new approach for LHC event analysis. *JHEP* **02**, 160 (2021). [https://doi.org/10.1007/JHEP02\(2021\)160](https://doi.org/10.1007/JHEP02(2021)160). arXiv:1912.10625 [hep-ph]
164. E. Pfeffer, M. Waßmer, Y.-Y. Cung, R. Wolf, U. Husemann, A case study of sending graph neural networks back to the test bench for applications in high-energy particle physics (2024) arXiv:2402.17386 [hep-ph]
165. L. Builtjes, S. Caron, P. Moskvitina, C. Nellist, R.R. Austri, R. Verheyen, Z. Zhang, Attention to the strengths of physical interactions: Transformer and graph-based event classification for particle physics experiments (2022) arXiv:2211.05143 [hep-ph]
166. T. Heimel, G. Kasieczka, T. Plehn, J.M. Thompson, QCD or What? *SciPost Phys.* **6**(3), 030 (2019). <https://doi.org/10.21468/SciPostPhys.6.3.030>. arXiv:1808.08979 [hep-ph]
167. M. Farina, Y. Nakai, D. Shih, Searching for New Physics with Deep Autoencoders. *Phys. Rev. D* **101**(7), 075021 (2020). <https://doi.org/10.1103/PhysRevD.101.075021>. arXiv:1808.08992 [hep-ph]
168. S. Tsan, R. Kansal, A. Aportela, D. Diaz, J. Duarte, S. Krishna, F. Mokhtar, J.-R. Vlimant, M. Pierini, Particle Graph Autoencoders and Differentiable, Learned Energy Mover’s Distance. In: 35th Conference on Neural Information Processing Systems (2021)
169. O. Atkinson, A. Bhardwaj, C. Englert, V.S. Ngairangbam, M. Spannowsky, Anomaly detection with convolutional Graph Neural Networks. *JHEP* **08**, 080 (2021). [https://doi.org/10.1007/JHEP08\(2021\)080](https://doi.org/10.1007/JHEP08(2021)080). arXiv:2105.07988 [hep-ph]
170. B. Ostdiek, Deep Set Auto Encoders for Anomaly Detection in Particle Physics. *SciPost Phys.* **12**(1), 045 (2022). <https://doi.org/10.21468/SciPostPhys.12.1.045>. arXiv:2109.01695 [hep-ph]
171. Z. Hao, R. Kansal, J. Duarte, N. Chernyavskaya, Lorentz group equivariant autoencoders. *Eur. Phys. J. C* **83**(6), 485 (2023). <https://doi.org/10.1140/epjc/s10052-023-11633-5>. arXiv:2212.07347 [hep-ex]