



City Research Online

City St George's, University of London

Citation: Araz, J. Y., Athanasakos, D., Ploskon, M. & Ringer, F. (2026). Graph theory inspired anomaly detection at the LHC. *Journal of High Energy Physics*, 2026(2), 254. doi: 10.1007/jhep02(2026)254

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/37203/>

Link to published version: [https://doi.org/10.1007/jhep02\(2026\)254](https://doi.org/10.1007/jhep02(2026)254)

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Graph theory inspired anomaly detection at the LHC

Jack Y. Araz ^{a,b,c} Dimitrios Athanasakos ^{c,d} Mateusz Ploskon ^e and Felix Ringer ^c

^a*Department of Physics and Astronomy, University College London, London, WC1E 6B, U.K.*

^b*School of Science & Technology, City St. George's, University of London, London, EC1V0HB, U.K.*

^c*Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794, U.S.A.*

^d*C.N. Yang Institute for Theoretical Physics, Stony Brook University, Stony Brook, NY 11794, U.S.A.*

^e*Nuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, U.S.A.*

E-mail: j.araz@ucl.ac.uk, dimitrios.athanasakos@stonybrook.edu, mploskon@lbl.gov, felix.ringer@stonybrook.edu

ABSTRACT: Designing model-independent anomaly detection algorithms for analyzing LHC data remains a central challenge in the search for new physics, due to the high dimensionality of collider events. In this work, we develop a graph autoencoder as an unsupervised, model-agnostic tool for anomaly detection, using the LHC Olympics dataset as a benchmark. By representing jet constituents as a graph, we introduce a method to systematically control the information available to the model through sparse graph constructions that serve as physically motivated inductive biases. Specifically, (1) we construct graph autoencoders based on locally rigid Laman graphs and globally rigid unique graphs, and (2) we explore the clustering of jet constituents into subjets to interpolate between high- and low-level input representations. We obtain the best performance, measured in terms of the Significance Improvement Characteristic curve for an intermediate level of subjet clustering and certain sparse unique graph constructions. We further investigate the role of graph connectivity in jet classification tasks. Our results demonstrate the potential of leveraging graph-theoretic insights to refine and increase the interpretability of machine learning tools for collider experiments.

KEYWORDS: Automation, Jets and Jet Substructure

ARXIV EPRINT: [2506.19920](https://arxiv.org/abs/2506.19920)

Contents

1	Introduction	1
2	Graph theory and jet representations	4
2.1	Subjet clustering and phase space considerations	4
2.2	Graph rigidity, Laman and unique graphs	5
2.3	Graph constructions for jets	8
3	Anomaly detection at the LHC	9
3.1	Benchmark LHC anomaly detection data set	10
3.2	Graph autoencoder	12
3.3	Numerical results	15
4	Conclusions and outlook	20
A	Applications to jet classification	21
A.1	Data sets	21
A.2	Graph neural network classifier	22
A.3	Numerical results and interpretability	22

1 Introduction

Decades of experimental validation, have confirmed Standard Model’s (SM) predictions with remarkable precision. However, the SM is widely understood to be an incomplete theory. It does not incorporate gravity, or account for the strong evidence of dark matter and dark energy. These and other issues serve as compelling motivations for the existence of physics beyond the Standard Model (BSM). Since the Higgs discovery [1, 2], the search for BSM signals is one of the central goals of the experimental program at the LHC. Historically, these searches have largely followed a “top-down” approach, targeting specific theoretical models proposed to address the SM’s shortcomings [3–8]. While this targeted search strategy has explored vast regions of potential new physics parameter space, it inherently carries the risk of being blind to signatures that do not align closely with the specific models being tested.

Given the absence of a definitive BSM discovery thus far, despite extensive searches, there is a growing interest in complementary search strategies that are essential to take advantage of the large data sets from the LHC. This has led to significant efforts in model-agnostic approaches using machine learning, as can be seen for example from the 2020 LHC Olympics anomaly detection challenge [9, 10]. These approaches aim to identify potential new physics signals by detecting anomalous events from the Standard Model background. While less sensitive to the signatures of any particular model, they enable simultaneous searches across a wide range of BSM scenarios in a theory-agnostic way. Moreover, they allow training directly on real, unlabeled data instead of relying only on simulations [11–16]. A

comprehensive overview of recent developments in anomaly detection methods at particle colliders can be found in ref. [17].

Most machine learning-based anomaly detection algorithms fall into two main categories: approaches based on unsupervised or weakly supervised training. Within the unsupervised paradigm, various studies have been performed using autoencoders [18–31]. A standard autoencoder consists of two neural networks: the encoder and the decoder, that are trained to reconstruct the input. Due to an intermediate bottleneck, the autoencoder is forced to learn the dominant features of the data distribution. In collider physics applications, an autoencoder can be trained on background-rich data and evaluated on a mixed data set, where events with high reconstruction loss are flagged as anomalous.

Recently, much work has focused on applying GNNs for autoencoder-based anomaly detection, moving beyond simple dense representations. Ref. [32] used graph convolutional networks with an EMD-based [33] reconstruction loss to model jets as particle clouds. To ensure theoretical consistency, [34] developed an IRC-safe architecture that uses energy-weighted aggregation to suppress soft radiation. More recently, Geometric Deep Learning [35] methods have been applied to enforce Lorentz equivariance in the latent space [36], while Transformer-based models have leveraged global attention mechanisms for improved performance [37]. Our work investigates the impact of using sparse graph structures motivated by graph theory.

Weakly supervised anomaly detection often employs the Classification Without Labels (CWoLa) approach [38–42], particularly for searches where a new signal is localized in a certain phase space region. A classifier is trained to distinguish events in this signal region from those in background-dominated sideband regions. To train classifiers on the low-level full event information, generative models have been developed to interpolate the sideband regions into the signal region. See for example refs. [43–50].

There are a few differences between unsupervised and weakly supervised anomaly detection techniques that lead to complementary sensitivity [51]. First, the weakly supervised approach relies on the signal fraction when training the classifier. As a result, the performance can improve significantly with an increasing signal fraction. In contrast, in an unsupervised setting such as the autoencoder used in this work, the machine is trained to reconstruct the input features without using labels, and its performance is independent of the signal-to-background ratio, provided that the training sample is dominated by background events. It is therefore particularly well suited for anomaly detection in the small signal fraction regime [51]. Second, unsupervised anomaly detection can be readily applied to both resonant and non-resonant anomaly detection [30, 37, 52–58], whereas weakly supervised methods have been developed primarily for resonant searches. That said, weakly supervised methods can also be extended to the non-resonant regime. See, for example, ref. [59]. Third, autoencoders tend to be sensitive only to particular topologies, while weakly supervised methods are generally more robust across a broader class of signals. In this regard, the two approaches are complementary.

Another axis to categorize anomaly detection algorithms is whether they operate on high-level or low-level features such as N -subjettiness observables [60, 61] or full particle four-momenta, respectively. In this work, we address this aspect by clustering jet constituents into subjects using the exclusive k_T algorithm [62]. This enables us to interpolate smoothly between using a small number of subjects as high-level inputs and using individual hadrons

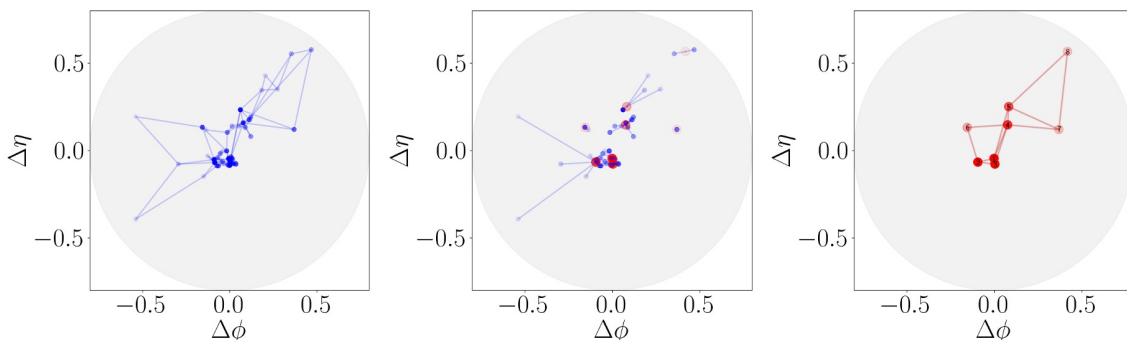


Figure 1. Representation of a two-pronged Z boson jet as a graph. Left panel: hadrons (constituents of a jet) shown as a Laman graph built using the Henneberg construction, described in section 2.3 and figure 4. Middle panel: hadrons grouped into 8 exclusive k_T subjects, which are shown as red nodes. The blue edges connect hadrons to the subject they belong to. Right panel: the Laman graph built from subjects according to the same procedure as the left panel. The intensity of particle and subject nodes is proportional to the fraction of the jet p_T they carry. The coordinates are in plane defined by the rapidity η and the azimuthal angle φ with respect to the jet axis where $\Delta\eta = \eta^{\text{node}} - \eta^{\text{jet}}$ and $\Delta\phi = \phi^{\text{node}} - \phi^{\text{jet}}$.

as low-level inputs to the learning algorithm. This approach removes the need to hand-select a few physically motivated observables. Indeed, we find that the performance of our anomaly detection algorithm peaks for an intermediate number of reclustered subjects, around $n_{\text{subjects}} \approx 30$.

In this work, we develop an unsupervised anomaly detection algorithm based on a graph autoencoder. Rather than reconstructing the absolute three-momenta of particles or subjects, our model focuses on their transverse momenta p_T and relative angular distances in the rapidity-azimuth plane. Using relative distances (either between particles or with respect to the jet axis) rather than absolute positions is more closely related to observables calculable in perturbative QCD. Moreover, high-energy jet structure is expected to be encoded primarily in relative features rather than absolute ones, which we confirm by additionally training classifiers on several representative jet physics tasks.

The primary focus of our work is on the use of different graph representations of jets. Instead of relying only on fully connected graphs that encode all pairwise distances, we explore the use of sparse graphs for anomaly detection algorithms. For the sparse graphs we consider, the number of edges scales linearly with the number of particles or subjects, whereas for fully connected graphs it scales quadratically. Specifically, we consider so-called locally rigid Laman graphs and different globally rigid unique graphs. The notion of rigidity is related to the embedding of a graph in Euclidean space, here specifically \mathbb{R}^2 , given only relative distances between nodes. The two types of sparse graphs we consider correspond to the minimal sets of constraints required to determine such an embedding, as discussed in more detail below. Due to their connection to the minimal amount of information needed to specify the graph structure in \mathbb{R}^2 or the azimuth-rapidity plane, they provide a natural choice for a physically motivated inductive bias in graph-based autoencoders for anomaly detection. Similar to the clustering of jet constituents into subjects, these graph constructions allow us to control the

amount of information accessible to the machine while ensuring that sufficient geometric structure is retained. We find that certain unique graph constructions achieve the best performance, as measured by the maximum Significance Improvement Characteristic (SIC) curve on the LHC Olympics data set, which we use to benchmark our approach.

The remainder of this work is organized as follows. In section 2, we introduce the representation of jets in terms of graphs based on particles or reclustered subjets. In addition, we review relevant concepts from graph theory, such as graph rigidity and the graph embedding problem. In particular, we describe both Laman and different unique graph construction techniques that will be used as input to the machine learning-based anomaly detection algorithm. In section 3, we describe the graph autoencoder developed in this work for unsupervised anomaly detection and present numerical results evaluating its performance on the LHC Olympics data set. We conclude in section 4 and provide an outlook. In appendix A, we additionally present results on jet classification using the different graph construction techniques.

2 Graph theory and jet representations

We start by discussing the representation of jets that will serve as input to the machine-learning algorithms described in the next section. First, we cluster jets into subjets to control the amount and the complexity of information that the machine has access to. Next, we will introduce relevant concepts in graph theory and describe different graph construction techniques that will be used as the basis of the graph autoencoder developed in the subsequent section.

2.1 Subjet clustering and phase space considerations

The set of massless particles contained in a high-energy jet can be described in terms of their transverse momenta, absolute positions in rapidity, and azimuthal angle (p_{Ti}, η_i, ϕ_i) . Due to rotational and translational invariance, we can also represent the particles inside a jet in terms of their transverse momenta p_{Ti} , and the relative distances between all particles i and j in the rapidity-azimuth plane $\theta_{ij} = (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{1/2}$. Jet clustering algorithms are designed to recursively cluster soft and collinear emissions to approximately reconstruct the parton branching mechanism that led to the observed jet. To systematically throttle the information used as input to machine learning algorithms, we recluster particles into a given number of subjets. We cluster particles using the exclusive k_T algorithm with the E recombination scheme [62]. Here, the recursive clustering algorithm is stopped when a given number of subjets is obtained. This results in a fixed number of subjets per jet but with varying subjet radii. Subjets have been explored in the context of jet classification tasks to assess the relevance of Infrared-Collinear (IRC) safe information in ref. [63]. In this work, we will use subjet reclustering to throttle the information that the anomaly detection algorithm has access to, as described in the next section. By changing the number of reclustered subjets, we can systematically dial in the amount of soft and collinear information that the machine learning algorithm has access to. This approach allows for a smooth interpolation to the hadron level where the full information content is accessible. The use of IRC safe subjets is closely related to the use of complete sets of jet substructure observables such as the

N -subjettiness [60, 61] basis [64–66] and Energy Flow Polynomials [67]. See figure 1 for an illustration of the particle content of a representative high-energy jet at the LHC in the rapidity-azimuth plane. The middle panel shows the assignment of hadrons grouped into 8 exclusive k_T subjets. The graph structures shown in the left and right panel of figure 1 will be discussed in more detail below.

In general, a machine learning algorithm for classification or anomaly detection tasks can take as input the four momenta of all N particles inside the jet, which corresponds to $4N$ variables per jet. For massless particles using momentum conservation, this is reduced to $3N - 4$ independent variables. This $(3N - 4)$ -dimensional phase space manifold was found to be isomorphic to $\Delta_{N-1} \times S^{2N-3}$ [68, 69]. Here, Δ_{N-1} corresponds to an $N - 1$ dimensional simplex, which means that the N transverse momenta add to the total jet momentum, and S^{2N-3} denotes the $2N - 3$ dimensional sphere. This manifold can be parametrized in terms of $N - 1$ transverse momenta p_{Ti} , and a suitable basis of $2N - 3$ relative angles, θ_{ij} , between particle pairs [64]. This manifold is related to so-called Laman graphs but not unique graphs, as discussed in the next section.

In principle, a machine learning algorithm has access to the same information content using the absolute positions encoded in the full particle four momenta or in terms of transverse momenta and relative distances. However, in practice, the performance of machine learning algorithms for various tasks at collider experiments can differ significantly depending on the type and amount of input information that is provided and the architecture using the provided information. For example, machine learning-based jet classification algorithms typically achieve the best performance the more low-level information is provided [64, 70]. However, it has been found that unsupervised anomaly detection algorithms at collider experiments can perform well with a limited number of high-level observables. See for example ref. [71]. In this work, instead of hand selecting a set of high-level observables, we will employ reclustered subjets, which allows us to smoothly interpolate between high-level information analogous to observables or summary statistics and low-level particle information. We find that the performance of our graph autoencoder for anomaly detection peaks at intermediate values of the provided information. In addition, from our numerical studies, we find that we achieve a better performance by providing only relative distances of particles instead of their absolute positions.

2.2 Graph rigidity, Laman and unique graphs

Unsupervised anomaly detection at collider experiments may benefit from enforcing physically motivated inductive biases in the machine learning architecture. We will explore these concepts numerically using a graph neural network (GNN)-based autoencoder in the next section. First, since tasks like jet classification and anomaly detection at collider experiments are invariant under translations and/or rotations, it is natural to work with relative distances between particles or subjets, rather than absolute positions in the rapidity-azimuth plane as discussed above. When representing jets as graphs, the relative distances are naturally encoded in the edge features of the graph. Second, the choice between fully connected and sparse graphs impacts the expressiveness and generalization of the downstream machine learning architecture. For anomaly detection, we aim to achieve a balance: the graph should

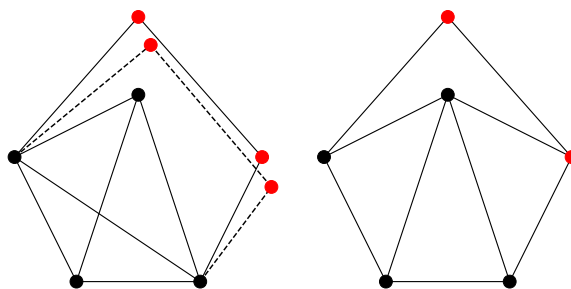


Figure 2. Representative examples of flexible (left) and locally rigid (right) graphs with $N = 5$ nodes and $|E| = 2N - 3 = 7$ edges embedded in \mathbb{R}^2 . The flexible graph admits continuous deformations of the two red vertices that preserve pairwise edge lengths while changing distances between non-adjacent nodes. Two distinct embeddings of the same flexible graph are shown (solid and dashed lines). In contrast, by moving one of the edges, the right graph is locally rigid.

be sparse enough to avoid redundancies, yet sufficiently connected to preserve essential geometric structures. These considerations motivate a graph-theoretic framework for jet representations. To formalize this, we are going to review several fundamental concepts from graph theory, which will motivate the construction of certain sparse graph structures, which we will explore in the context of anomaly detection in jet physics.

Let $G = (E, V)$ denote an undirected graph, where V is a set of $N = |V|$ vertices, and E is a set of $m = |E|$ edges, encoding pairwise geometric relations. Each particle or subject $i \in V$ is associated with scalar features, such as the normalized transverse momentum p_{Ti} , and each edge $(i, j) \in E$ corresponds to a measured relative distance or angle of particles θ_{ij} and/or similar relational quantities. There are various ways to choose the connectivity structure of the graph. For example, choosing only a specific subset of $2N - 3$ edges yields a graph that minimally captures geometric constraints. Alternatively, a fully connected graph includes all $\binom{N}{2} = N(N - 1)/2$ pairwise distances, as used in transformer-based models. While fully connected graphs are more expressive, we find that anomaly detection performance can improve when using certain sparse graphs. To motivate how to choose such graphs, we are now going to review the concept of rigid vs. flexible graphs, which is tied to how graphs can be embedded in Euclidean space.

A central question in graph theory and graph reconstruction is whether a graph with given edge lengths can be embedded in an Euclidean space. Formally, given

- an undirected graph $G = (V, E)$,
- a set of edge lengths $\{e_{ij}\}$ with $(i, j) \in E$,
- the dimension d of the Euclidean space \mathbb{R}^d ,

find a mapping $p : V \rightarrow \mathbb{R}^d$ such that

$$\|p(i) - p(j)\| = e_{ij}, \quad \forall (i, j) \in E. \tag{2.1}$$

Here $p(i)$ denotes the position of node i and $\|\cdot\|$ denotes the Euclidean distance metric in \mathbb{R}^d . That is, we are looking for the positions of nodes that are consistent with the given edge

lengths. In the context of jet physics, this corresponds to embedding particles or subjects in the rapidity-azimuth plane, such that their given relative distances or angles are preserved. A key question is whether a given graph, with its set of pairwise distances, admits a unique realization (up to global isometries such as translations and rotations), or whether multiple, potentially infinitely many, configurations exist that satisfy the same set of edge constraints. This is described by the notion of graph rigidity. A graph is called

- *Flexible* if it can be continuously deformed while preserving pairwise edge constraints, resulting in an infinite family of realizations. This flexibility corresponds to degrees of freedom in the relative configuration of non-adjacent particles that are not connected by edges. For an illustration, see figure 2.
- *Locally rigid* if there are finitely many discrete realizations. See the left and middle graph in figure 3, where the reflection of one node leads to two distinct realizations of a graph embedded in \mathbb{R}^2 . Rigid graphs are locally unique: small perturbations of the vertex positions that preserve the edge lengths result only in isometric configurations.
- *Globally rigid* if there is only one unique embedding up to isometries. For an illustrative example comparing locally and globally rigid graphs, see figure 3.

In the context of both jet classification and anomaly detection, we expect that machine learning algorithms based on flexible graphs generally underperform since they do not contain sufficient geometric information. Instead, we will focus on rigid graphs in this work. Since our application lies in two spatial dimensions, embedding a jet in the rapidity-azimuth plane, we now restrict our discussion to graph embeddings in \mathbb{R}^2 . In this case, the minimal number of edges required to achieve local rigidity is $|E| = 2N - 3$. Minimally rigid graphs are referred to as *Laman graphs* [72]. Minimal rigidity means that removing one edge always results in a flexible graph. Since Laman graphs ensure local rigidity, they are a natural candidate for a machine-learning architecture based on sparse graphs.

While Laman graphs ensure local rigidity, they may still admit multiple discrete realizations in \mathbb{R}^2 . Global rigidity is obtained for *unique graphs*. A graph has a unique realization in \mathbb{R}^2 if and only if it satisfies Hendrickson’s conditions [73–75]. First, the graph must be redundantly rigid, meaning it remains rigid after the removal of any single edge. Second, it must be 3-connected, i.e. for every pair of nodes there exist three vertex-disjoint paths between them. As a result, unique graphs may be ideally suited for machine learning tasks at collider experiments as they encode enough relational geometry to fix all positions unambiguously up to isometries.

In contrast to fully connected graphs, finding an embedding of a sparse graph in \mathbb{R}^2 that satisfies given edge length constraints is, in general, an NP-hard problem in the worst case [73, 76, 77]. Designing a machine learning architecture for anomaly detection that is tasked to solve this embedding problem could provide new insights into the performance of the corresponding algorithm. However, for practical applications, the relevant computational complexity is determined more by the average case than by worst-case hardness. A detailed investigation of this aspect is beyond the scope of this work. Instead, we will focus on anomaly detection algorithms based on Laman and unique graph constructions as discussed in the next section.

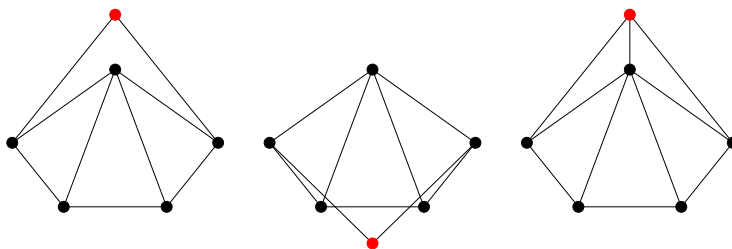


Figure 3. Illustration of locally vs. globally rigid graphs embedded in \mathbb{R}^2 . The two graphs on the left correspond to two distinct realizations of the same locally rigid Laman graph. Reflecting the red node about the horizontal axis preserves all pairwise edge lengths but alters the distances between non-adjacent nodes. Instead, the graph on the right is globally rigid, which allows for only one unique realization up to the isometries.

2.3 Graph constructions for jets

In this section, we discuss in detail graph construction techniques for jets, focusing on Laman and unique graphs. When constructing graphs, we generally rely on an ordering of the particles or subjects. Throughout this section, we order particles/subjects by their transverse momentum p_T , starting with the particles with the highest p_T . We also explored alternative orderings but consistently observed worse performance for the machine learning tasks described below. Note that this ordering breaks permutation invariance for the sparse graphs considered here. Permutation invariance is only recovered in the limit of fully connected graphs.

We start by discussing the construction of Laman graphs, which are minimally rigid graphs with $2N - 3$ edges. All Laman graphs can be obtained by the Henneberg construction, which provides an inductive recipe using two types of operations [78, 79]. Starting from a single edge between two nodes:

- Type I: Add a new node and connect it to two existing nodes with new edges.
- Type II: Subdivide an existing edge, then connect the new node to a third existing node.

For simplicity, we restrict ourselves to Type I operations, which are sufficient to construct Laman graphs. See figure 4 for an illustration. We begin with a triangle formed by the three hardest particles, i.e., those with the highest p_T . We then iterate through the list of remaining particles, adding them one by one to the existing graph. Each new particle is connected to the graph with two edges linking it to existing nodes. There are multiple ways to choose the two existing nodes to connect to each new particle. For instance, one could always connect to the two hardest particles. However, we find improved performance for the machine learning task discussed below when each new particle is connected to its two nearest neighbors in angular distance in the η - ϕ plane. This procedure is similar to the Cambridge/Aachen jet clustering algorithm [80, 81]. Using a Voronoi tessellation, this can be achieved in $\mathcal{O}(N \ln N)$ steps. See figure 1 for a representative Laman graph using the Henneberg construction for the particles (left panel) and subjects (right panel) inside a jet. The resulting graph has $2N - 3$ edges and is locally rigid.

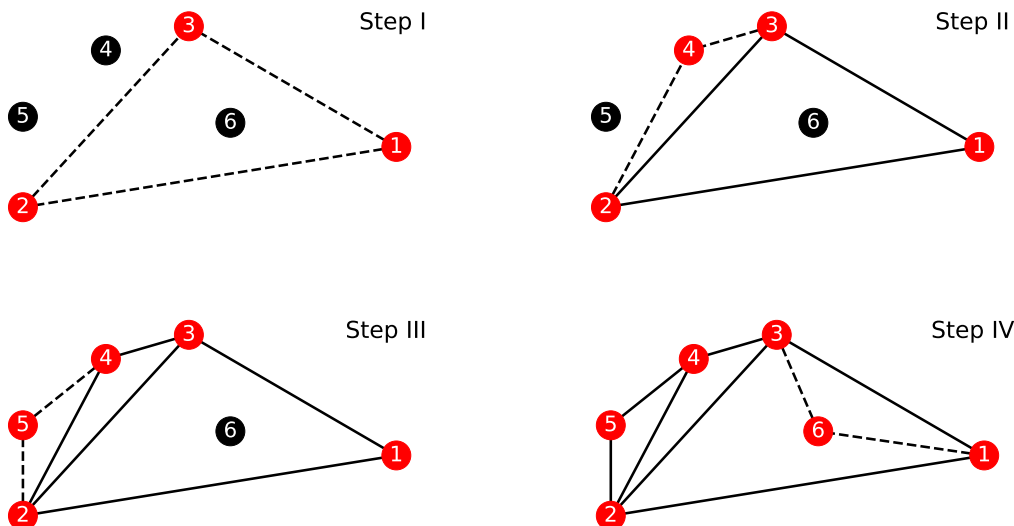


Figure 4. Illustration of the Laman graph construction for a jet with six particles (nodes), ordered by descending transverse momentum p_T . The three hardest particles form an initial triangle. Each subsequent particle is added to the graph by connecting it to its two nearest neighbors, based on angular proximity in the η - ϕ plane, among the previously included nodes. This sequential procedure ensures that the resulting graph is locally rigid, with $2N - 3$ edges.

Analogously, we can construct unique graphs that are globally rigid with $|E| = \mathcal{O}(3N)$ edges. We again sort particles by decreasing p_T , and start with a fully connected graph of the four hardest particles. Each subsequent particle is then connected to its three nearest neighbors in angular distance, adding three edges per new node. Repeating this process yields a graph that satisfies the Hendrickson conditions for global rigidity. See ref. [73] for more details. In the following, we refer to this construction as a *unique-3* graph. More generally, we define *unique- k* graphs by starting from a fully connected subgraph of the $k + 1$ hardest particles. For each new particle, we add k edges to its k nearest neighbors. The resulting graph has $|E| = kN - k(k + 1)/2$ edges. See figure 5 for an illustration of the *unique-3* graph construction. For $k \geq 3$ the resulting graph is globally rigid, whereas for $k = 2$, we recover the locally rigid (not unique) Laman graph. Note that these unique graphs are not minimal constructions of globally rigid graphs, which are referred to as M -circuits [82]. They can be obtained using a modification of the Type II Henneberg operations. However, in this work, our primary concern is whether the graph admits a unique realization, rather than its minimality. While increasing k yields denser graphs, the number of edges still scales linearly with N , in contrast to the $\mathcal{O}(N^2)$ scaling of fully connected graphs. In the next section, we are going to use the graph constructions introduced here for anomaly detection tasks at the LHC using a graph-based autoencoder.

3 Anomaly detection at the LHC

We begin by briefly reviewing the LHC Olympics data set [9, 10], which serves as the benchmark for our numerical studies in the subsequent sections. Next, we describe the machine learning architecture of the graph autoencoder used throughout this work. We

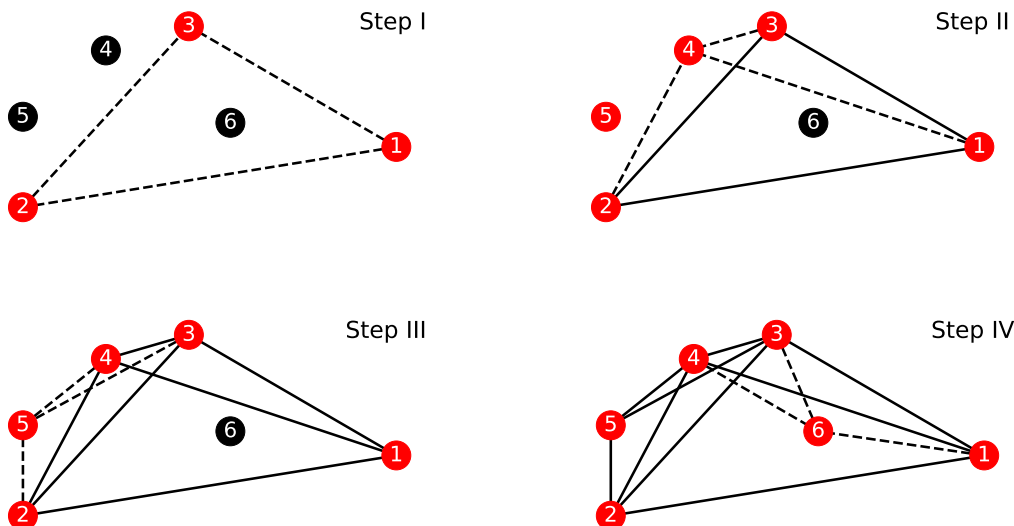


Figure 5. Construction of a *unique-3* graph analogous to figure 4. The resulting graph is globally rigid allowing for a unique realization in \mathbb{R}^2 up to isometries.

then present the results of our anomaly detection analysis using different subset clusterings and graph constructions.

3.1 Benchmark LHC anomaly detection data set

As the benchmark data set for anomaly detection studies of BSM signals at the LHC, we use the LHC Olympics challenge 2020 data set [9, 10], which enables direct comparison with a wide range of recent methods [42–50, 71, 83, 84]. It contains up to 1M background events generated from Standard Model processes, resulting in dijet final states. The anomalous signal events are created using the resonant production of a new heavy boson Z' , that decays as $Z' \rightarrow X(\rightarrow q\bar{q}), Y(\rightarrow q\bar{q})$, with masses $m_{Z'} = 3.5$ TeV, $m_X = 0.5$ TeV, and $m_Y = 0.1$ TeV.

The BSM particle Z' is produced on-shell and decays into two intermediate particles X, Y that subsequently decay into quarks. This produces a signal characterized by a peak in the dijet invariant mass spectrum. For sufficiently large signal fractions, the Z' resonance can be discovered using traditional bump-hunting techniques. Figure 6 provides an illustrative representation of a bump-hunt search. The signal region is identified as the area with the highest concentration of signal events, while the sidebands serve as control regions where the signal contribution is negligible. The blue, black, and dashed red lines represent the signal, observed counts, and background yields, respectively. Dashed vertical lines delineate the signal and sideband regions. For smaller signal fractions, the bump’s significance is reduced and detection may instead require machine learning techniques that can exploit the full event information. This type of data set naturally fits within the weakly-supervised anomaly detection paradigm, where the sideband regions are dominated by background events, while the signal region is localized in a central window of dijet invariant mass values that can in practice be varied to carry out a full search [44, 45, 83]. Instead, the unsupervised anomaly detection based on autoencoders employed in this work can be trained on the entire range of dijet invariant mass values as long as the signal fraction is sufficiently small.

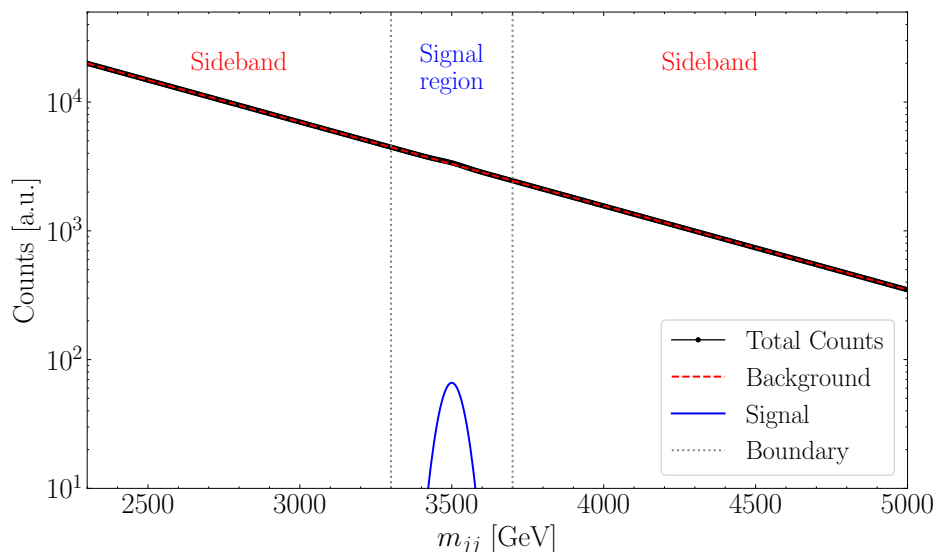


Figure 6. Schematic depiction of the dijet mass distribution of background (red) and signal (blue) in the LHC Olympics data set. The sideband regions are dominated by background events, while the signal region contains a finite fraction of signal events. Here, it corresponds to $S/B \sim 3 \cdot 10^{-3}$ in the signal region and a significance $S/\sqrt{B} \sim \mathcal{O}(1)$.

The autoencoder is trained to minimize the reconstruction loss resulting in a large anomaly score for BSM signal events. Ref. [51] found that the autoencoder performance remains stable for signal-to-background ratios up to $S/B \lesssim 1\%$, a result we were able to confirm in our numerical studies.

The event generation is performed using PYTHIA8 [85], and detector effects are simulated with DELPHES 3.4.1 [86]. In both cases, default configurations are used, without pileup or multiple parton interactions (MPI). The final-state jets are reconstructed using the anti- k_T clustering algorithm [87], as implemented in FASTJET [88]. We refer the reader to section A.1 on the details of the preprocessing of the events. A jet radius of $R = 1.0$ is used with a transverse momentum cut of $p_T > 1.2$ TeV. The signal events feature two high-energy jets containing the decay products of the boosted X and Y particles. These jets form a dijet system with large invariant mass, originating from the heavy Z' boson. In addition, they exhibit a non-trivial two-pronged jet substructure. Background events are simulated using Standard Model processes that contain two jets satisfying the same transverse momentum cuts. For training our graph autoencoder, we use 10^5 events with a tunable ratio of signal and background events. For validating the reconstruction loss, we use 10^4 events with the same S/B ratio as in training. Finally, to evaluate the model's ability to distinguish between unseen background and anomalous events using a reconstruction loss threshold, we use $5 \cdot 10^4$ events that are not part of the training or validation sets. We verified that increasing the training sample size beyond the chosen value leads only to marginal performance improvements, while significantly increasing the computational cost, particularly when scanning over multiple graph architectures.

3.2 Graph autoencoder

We develop a GNN-based autoencoder for the anomaly detection task described above. The model takes a graph representation of individual jets as input, with different information assigned to both the graph nodes and edges. Each particle or subjet inside the main jet corresponds to a node. Instead of using the absolute positional information of the particles or subjets in the rapidity-azimuth plane, we only use the transverse momentum $p_{T,i}$ of each particle or subjet i as a node input feature. The edges encode relationships between particles, in particular, their relative distances in the rapidity-azimuth plane. For each edge connecting particles i and j , we include the following features e_{ij} [89]:

$$\begin{aligned} \theta_{ij} &= (\Delta\eta_{ij}^2 + \Delta\phi_{ij}^2)^{1/2}, \\ k_{T,ij} &= \min(p_{T,i}, p_{T,j}) \theta_{ij}, \\ z_{ij} &= \min(p_{T,i}, p_{T,j}) / (p_{T,i} + p_{T,j}). \end{aligned} \tag{3.1}$$

Since jets have a variable number of constituents, each jet corresponds to a graph with a different number of nodes and edges. By contrast, exclusive k_T subjet reclustering always produces a graph with a fixed number of nodes and edges. The graph autoencoder developed in this work is compatible with both fixed and variable length input.

A GNN-based autoencoder is a machine learning architecture designed to learn a low-dimensional representation of graph-structured data in an unsupervised way. It consists of two parts: an encoder, which processes the input graph through multiple message-passing layers (described in more detail below) to produce a compressed latent embedding; and a decoder, which reconstructs aspects of the original graph from this embedding. The model is trained by minimizing a reconstruction loss that encourages it to capture the essential features of the input data. A key feature of autoencoder-based anomaly detection is that its performance does not depend on the fraction of anomalous events in the signal region, as long as the model is trained on a background-rich sample. During training, the autoencoder learns to compress and reconstruct background events. When applied to test data, the reconstruction loss — or a related scoring function — serves as an anomaly score, flagging events that deviate significantly from the learned background distribution. A higher score indicates a larger deviation, which has been found to be largely independent of the overall signal fraction [51].

Figure 7 shows a schematic illustration of the GNN-based autoencoder developed in this work. The encoder consists of a stack of custom message-passing blocks built upon the edge convolution (EdgeConv) layer introduced in ref. [90]. Each encoder block is designed to update the node features x_i by incorporating information from the neighboring node features x_j and the corresponding edge features e_{ij} . These blocks do not update the edge features. Instead, they reuse the same three relative features from the input layer as listed in eq. (3.1). Each block is composed of two linear layers, both using a 64-dimensional auxiliary space for the node features. In the final block, the second layer outputs a lower-dimensional representation x_{latent} forming the bottleneck of the autoencoder. For each node i , the message-passing operation of layer ℓ is given by

$$x_i^{\ell+1} = \text{ReLU} \sum_j \left(W_{e,1}^\ell \cdot e_{ij} + W_{e,2}^\ell \cdot x_j^\ell \right), \tag{3.2}$$

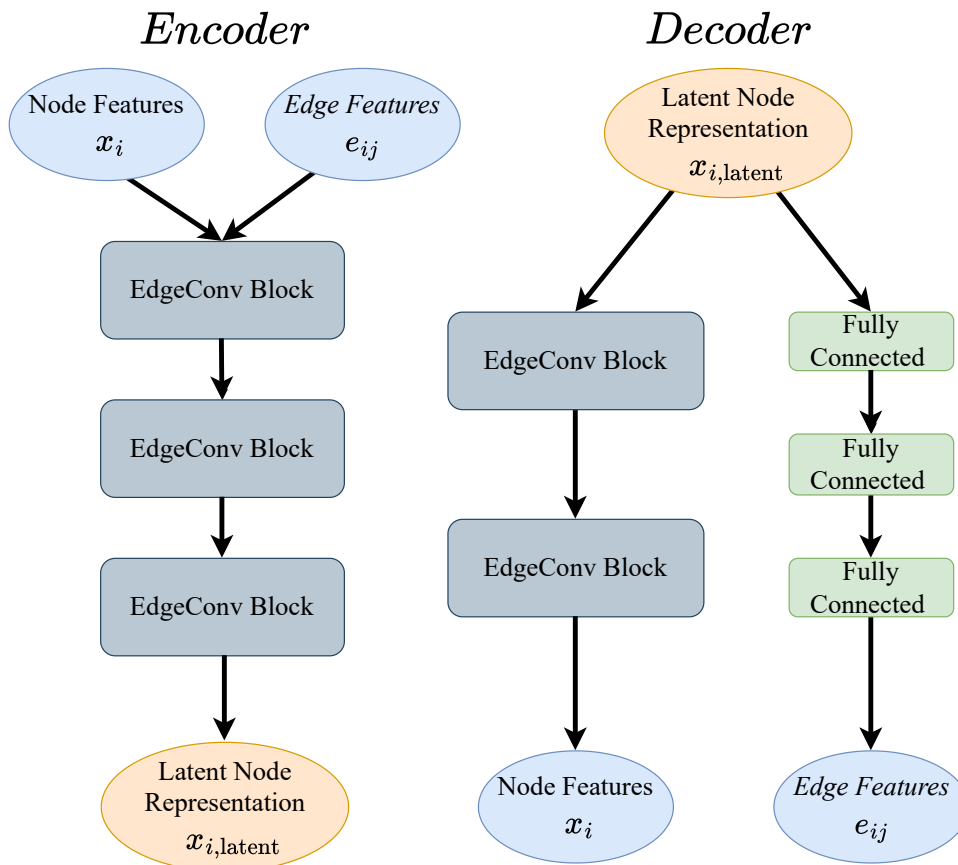


Figure 7. Schematic illustration of the GNN-based autoencoder developed in this work. For visualization purposes, we do not include the residual connections that exist within the encoder and the decoder. See text for more details.

where we sum over the neighbors j of node i . The matrices $W_{e,k}^\ell$ of the encoder with $k = 1, 2$ contain trainable weights, which are shared across all nodes and edges but vary from layer to layer. In addition, a residual connection [91] is applied after each block by adding the original node features x_i^ℓ to the updated representation. This helps to preserve important node-level information, ensures better gradient flow, and enables deeper network architectures. The sequence of encoder blocks maps the input graph to a lower-dimensional latent node representation x_{latent} with dimension d_{latent} . This latent representation focuses solely on node-specific information and does not retain explicit edge features. A hyperparameter scan showed that $d_{\text{latent}} = 2$ provides the best performance, creating a tight bottleneck that ensures only a compressed summary of the original information is passed to the decoder. Although the latent space is defined per node, the input information is distributed between node features and edge features. For a graph with N nodes and $2N - 3$ edges, this corresponds to approximately $3N$ input degrees of freedom, even for the sparsest Laman graph structures considered here. Since the latent representation is restricted to node variables, a three-dimensional latent space would not constitute a bottleneck. A one-dimensional latent space was found to be insufficient to reconstruct the jet structure, while a two-dimensional latent space provides the minimal capacity required to capture the relevant features, corresponding to a compression

factor of roughly $2/3$. The compression factor is more significant for unique- k graphs. We therefore refer to this choice as a tight bottleneck.

The decoder architecture consists of two independent branches that operate on the latent node features x_{node} . The first branch reconstructs the node features and consists of a stack of decoder blocks that are structurally analogous to those in the encoder. The second branch reconstructs the edge features. A key difference in the decoder is the absence of edge inputs: the decoder blocks form messages by combining the features x_i of node i with the features of the neighboring nodes x_j by concatenating them element-wise as $x_i - x_j$, as in the original EdgeConv proposal [90]. The message-passing operation for the decoder block is given by

$$x_i^{\ell+1} = \text{ReLU} \sum_j \left(W_{d,1}^\ell \cdot (x_i^\ell - x_j^\ell) + W_{d,2}^\ell \cdot x_i^\ell \right), \quad (3.3)$$

with trainable weights matrices $W_{d,k}^\ell$ analogous to the encoder operation. Each decoder block uses two linear layers of dimension 64. This branch ultimately maps the latent node features back to the original 1D node inputs. As in the encoder, residual connections are added to help preserve information and support deeper architectures. The second decoder branch is responsible for reconstructing the edge features. It takes the latent representations of each pair of connected nodes (x_i, x_j) and processes them through a symmetric function. Specifically, we use the concatenation of element-wise minima and maxima. The output is passed through two fully connected layers (each of dimension 64) followed by a final linear layer that produces a 3D output, matching the dimensionality of the input edge features. A residual connection is also added in this branch to ensure stable propagation of x_{latent} through the network.

The model is trained end-to-end by minimizing a combined loss function, for both node and edge features. We use Mean Squared Error (MSE) as the base loss function, following prior studies showing its suitability for anomaly detection [32]. The total loss that the model is trained on is the sum of the MSE loss for the node and edge features

$$\mathcal{L} = \text{MSE}_{\text{nodes}} + \text{MSE}_{\text{edges}}. \quad (3.4)$$

Here, the MSE loss is defined by summing over all vertices and edges, respectively,

$$\text{MSE}_{\text{nodes}} = \sum_{v \in V} |x_v - \hat{x}_v|^2, \quad \text{MSE}_{\text{edges}} = \sum_{ij \in E} |e_{ij} - \hat{e}_{ij}|^2. \quad (3.5)$$

The hatted quantities denote the original inputs the autoencoder is trained to reconstruct. Since the input node features consist only of the transverse momenta p_{Ti} of the particles or subjects, while the latent node dimension is two-dimensional, the autoencoder can relatively easily reconstruct the transverse momenta. As a result, the reconstruction loss is dominated by the edge features. Potentially, a relative scaling factor between the two contributions to the total loss in eq. (3.4) could be used to further improve the performance but we do not explore this option here. Since the LHC Olympics anomaly detection data set contains dijets, we independently process the two jets through the autoencoder. Their respective reconstruction loss is summed to obtain the total event-level loss. An alternative approach, which is left for future work, would be to train separate models for the leading and subleading jets, or to apply thresholds to each jet's reconstruction loss individually rather than summing them.

After training, we quantify the performance of the autoencoder by evaluating its ability to distinguish between background and signal in terms of differences in the reconstruction loss. Specifically, we consider the Area Under the Curve (AUC) of the Receiver Operator Characteristic (ROC) curve. The ROC curve is the cumulative distribution function of the true positive rate as a function of the false positive rate of a binary classification task as the decision threshold is varied. In addition, we consider a more relevant metric for anomaly detection: the maximum value of the Significance Improvement Characteristic (SIC) curve [92]. The SIC quantifies the gain in statistical significance achieved by applying a cut on the classifier output score. For a given threshold, the signal efficiency (true positive rate), ϵ_S , is the fraction of signal events that pass the selection, while the background efficiency (false positive rate), ϵ_B , is the fraction of background events that also pass the cut. The significance improvement for the cut is then given by $\text{SIC} = \epsilon_S / \sqrt{\epsilon_B}$. In our case, the anomaly score is the reconstruction loss; see figure 8. By varying the decision threshold, we determine which events are classified as background or signal. To estimate the statistical uncertainty in the performance metrics, we train the neural networks four times for each task and use the standard deviation of the resulting scores as a proxy for model uncertainty.

We note that we do *not* use any batch normalization layers in either the encoder or the decoder as we find that they degrade the anomaly detection performance by up to $\approx 20\%$ in terms of the background rejection rate, or similarly, $1 - \text{AUC}$. Batch normalization is typically used to mitigate internal covariate shift [93] by computing the batch-wise mean μ and variance σ^2 , and normalizing the data via $x \rightarrow (x - \mu) / \sigma$. However, this normalization can inadvertently smooth out the tail regions of the data distribution. In anomaly detection, where the test set may include both in-distribution (background) and out-of-distribution (signal) samples, this smoothing reduces the contrast between signal and background, weakening the model's ability to detect anomalies. While the overall reconstruction loss may be reduced, the discriminatory power is diminished.

We implement the model in PyTorch [94] using the AdamW optimizer [95]. We train the model for 50 epochs using a batch size of 512 and a linearly annealed learning rate. The learning rate starts at a peak value of $3 \cdot 10^{-3}$ and decreases to $2 \cdot 10^{-4}$ by the end of the training. During training, the model minimizes the reconstruction loss on the background-rich data set. During testing, it distinguishes signal from background by applying a threshold to the total reconstruction loss, as shown in figure 8.

3.3 Numerical results

As described in section 3.1, the LHC Olympics data set contains dijet background events simulated using Standard Model processes, along with signal events corresponding to the resonant production of a new heavy boson [9]. The new particle has a mass of 3.5 TeV, producing a localized bump in the dijet invariant mass spectrum m_{jj} . When the signal fraction is small relative to the background, the new particle is difficult to identify using traditional bump-hunting techniques. However, machine learning-based anomaly detection methods that exploit full event-level information can successfully identify such signals. As described in the previous section, the anomaly detection strategy employed in this work relies on an autoencoder trained in an unsupervised manner, using sparse graph representations

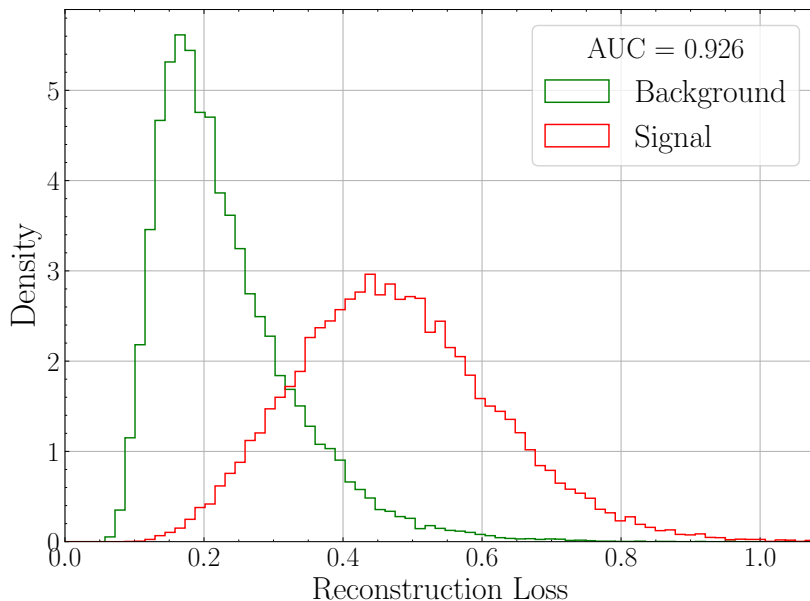


Figure 8. The reconstruction loss of the graph autoencoder for a representative run based on a sparse *unique-6* graph and $n_{\text{subjets}} = 30$.

of jets. The training data set for the autoencoder may include both signal and background events, provided the signal fraction remains sufficiently small. We now present numerical results obtained by applying the autoencoder described above to the LHC Olympics data set, on the full range of the dijet mass m_{jj} , using a variable number of reconstructed k_T subjets and exploring a range of graph construction strategies. For the results shown here, we use a signal-to-background ratio of $S/B = 3\%$. However, we note that comparable performance is observed across a range of values with $S/B \leq 3\%$, consistent with expectations [51]. An illustrative result for the reconstruction loss at test time is shown in figure 8. In this example, we use $n_{\text{subjets}} = 30$ exclusive subjets and employ the *unique-6* sparse graph construction, as described in section 2.3 above. By scanning over different decision thresholds, we obtain the corresponding ROC curve which determines the AUC and SIC, which are our chosen performance metrics for evaluating the various algorithms considered in this section.

The results for the performance of our graph-based autoencoder are shown in figure 9, which represent the central findings of our study. Each numerical value corresponds to the average over four independent runs, and the error bars indicate the standard deviation across these runs. To ensure the robustness of our results, we monitored the training convergence. In a small fraction of instances (approximately 5% on average), the training process did not converge to a satisfactory minimum for the reconstruction loss. These runs, identified by a significantly higher training loss, i.e. not within 75% of the best loss achieved by the other runs of that data point, were discarded and repeated until four successfully converged runs were obtained for each reported data point. In figure 9, we show the max SIC (left panel) and the AUC (right panel) for fully connected (blue), Laman (orange), *unique-6* (red), and *unique-3* (green) graphs. For completeness, we show the SIC for a representative run in figure 10.

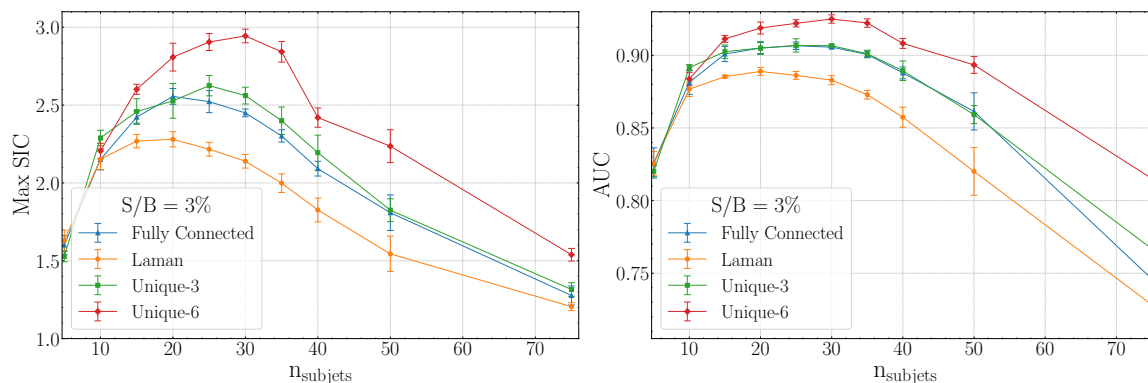


Figure 9. Anomaly detection performance (left: Maximum SIC, right: AUC) of our graph autoencoder as a function of the number of exclusive subjects for different graph constructions. The two curves exhibit similar behavior, with both performance metrics peaking for the *unique-6* graph at $n_{\text{subjets}} = 30$, resulting in a max SIC of 2.94 and an AUC of 0.925, respectively.

First, we find that the performance of the graph-based autoencoder peaks at intermediate values of n_{subjets} across all tested graph constructions. This behavior contrasts with typical jet classification tasks, where the performance peaks at large values of n_{subjets} or directly at the hadron level, i.e., without any subjet clustering. For further comparison, see appendix A, where we reproduce this trend using various graph-based jet classification methods. This is consistent with the findings of ref. [63], which used a deep sets-based classifier based on subjects. These findings highlight a difference between supervised classification and unsupervised anomaly detection: while classification generally benefits from richer representations, the latter may suffer from excessive input complexity, which can lead to overfitting. Subjet clustering, therefore, acts as a useful physics-informed bottleneck that can guide the model toward physically meaningful structures, resulting in improved performance. Rather than hand-selecting a few high-level observables, we can smoothly interpolate between high- and low-level information by adjusting the number of subjects n_{subjet} , achieving optimal performance in the range $n_{\text{subjet}} \approx 25 - 30$.

Second, we observe that autoencoders based on fully connected graphs, which encode the maximal amount of information, do not yield the best performance. While Laman graphs tend to underperform for $n_{\text{subjet}} \gtrsim 10$, the *unique-3* graph performs comparably or better, within the shown error bars, than the fully connected graph across all values of n_{subjet} . The sparse *unique-6* graphs consistently achieve the best performance among all *unique-k* graphs, which peaks for $n_{\text{subjet}} \sim 25 - 30$, in figure 9. A similar behavior, when comparing against the fully connected graph, was observed for higher-point unique graph constructions as well. The relatively low performance of Laman graphs may be due to the fact that edge information does not uniquely determine the graph structure, in contrast to the case of unique graphs. This suggests that, in addition to subjet reconstruction, sparse but structured graphs can serve as effective inductive biases that retain sufficient geometric structure allowing for an efficient learning procedure.

Next, in figure 11, we plot the maximum SIC value for different graph constructions for $n_{\text{subjets}} = 30$ that corresponds to the best performance for most of the unique graphs,

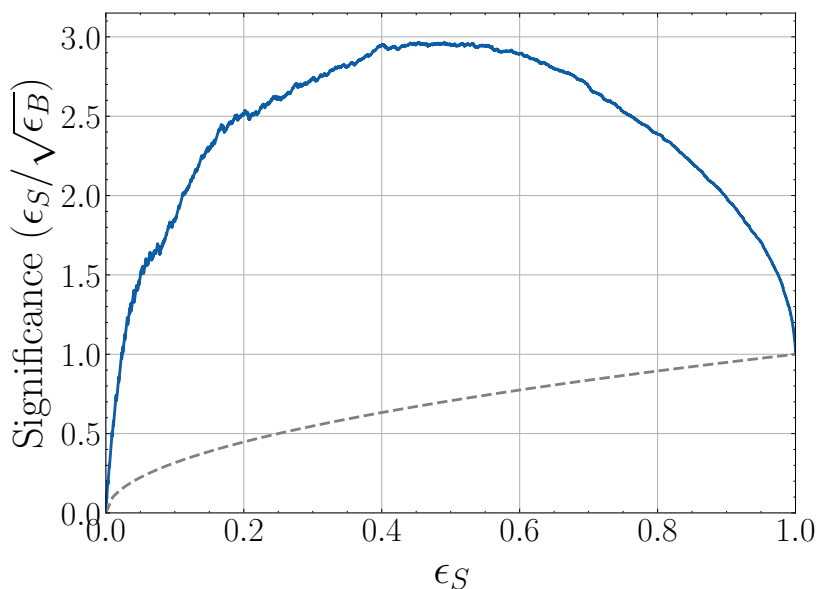


Figure 10. The SIC curve based on a sparse *unique-6* graph for $n_{\text{subjects}} = 30$. The dashed line indicates the behavior of a random classifier with $\epsilon_S = \epsilon_B$. The SIC corresponds to the same run as shown in figure 8.

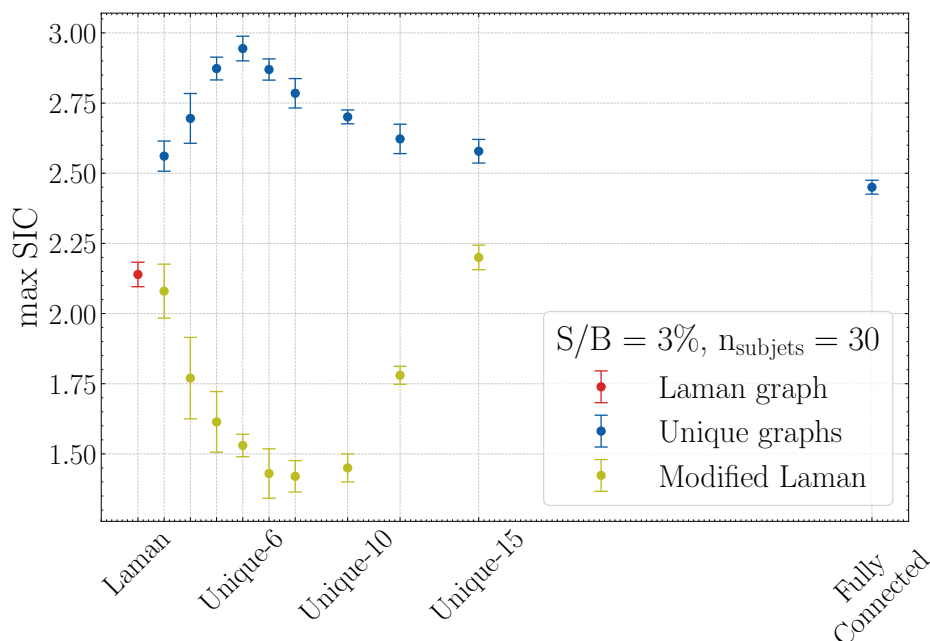


Figure 11. Maximum value of the SIC curve as a function of the connectivity for different graph constructions, for a fixed number of subjects $n_{\text{subjects}} = 30$. Here the *unique-30* graph is fully connected. The highest max SIC value is achieved for *unique-k* graphs with an intermediate level of sparsity.

see figure 9. We show the result for the Laman graph (red), several unique graphs and the fully connected graph (blue). We observe a peak at intermediate levels of sparsity with max SIC ~ 3 around the *unique-6* graph. Adding significantly more edges ultimately leads to a drop in performance. Similar to figure 9, this result indicates that systematically increasing the sparsity of the graph-based autoencoder can lead to improved performance.

To further investigate the impact of graph connectivity and to disentangle the effects of sparsity from those of graph topology, we introduce an alternative graph construction algorithm. The yellow markers in figure 11 show the performance of this alternative algorithm, which interpolates between Laman and fully connected graphs analogous to the *unique-k* graphs. This construction is referred to as “modified Laman” in figure 11. Starting from a Laman graph, we successively add edges by computing the angular distances in the η - ϕ plane between all pairs of subjects that are not yet connected. Edges are then added one-by-one between the closest unconnected pairs until the graph becomes fully connected. The yellow markers in figure 11 show the performance of these modified Laman graphs as a function of the number of edges, which is directly compared to the corresponding *unique-k* graphs shown by the blue markers. As both constructions eventually converge to the fully connected graph, their performance also becomes comparable for a large number of edges. However, while the unique graphs display a peak in performance at intermediate levels of sparsity, as discussed above, the performance of the modified Laman graphs decreases significantly for the same number of edges. This behavior suggests that it is not merely the degree of sparsity that explains the improved performance of *unique-k* graphs, but rather their specific connectivity structure. We conclude that the graph-theoretic design of *unique-k* graphs more effectively captures the topology of the underlying phase space. In fact, to the best of our knowledge, the performance of the *unique-6* graph represents the current state-of-the-art among unsupervised autoencoder-based methods for this anomaly detection benchmark data set. In addition, we compared our results to sparse k -nearest-neighbor graphs. Although these constructions also yield sparse graphs, they generally do not satisfy the rigidity conditions, such as Laman or globally rigid graphs. As expected, we find that their performance is generally worse compared to that of fully connected graphs.

We also tested a graph autoencoder based solely on absolute positional (node) information [32], without edge features, but it did not achieve competitive performance. Similarly, retaining the edge prediction while adding absolute node information and reconstructing the three momenta (p_{Ti}, η_i, ϕ_i) , similarly to ref. [34], led to a maximum value of the SIC of ~ 2 . These findings are consistent with the expectation that the relevant physics information is primarily encoded in the relative relationships between particles or subjects. Potential improvements of our results could be obtained by modifying the latent space representation that acts as a bottleneck of the graph autoencoder. Currently, the latent representation only focuses on the node information, outputting a 2D vector for every particle or subject, but any graph-related structural information is dropped. It has been noted in previous studies [96, 97] that the topological properties of both the data and the latent space can obstruct the reconstruction properties and the generalizability of an autoencoder. One way to improve this effect would be to project the starting graph, down to a sparse graph in the latent space, since this could match the manifold of the data. An exploration of this will be left for future work.

Reconstruction-based autoencoders are known to exhibit a complexity bias, where topologically simple objects such as QCD jets are reconstructed with high fidelity even if they are underrepresented or entirely absent from the training data [27]. This behavior is also observed in our setup and appears to be largely independent of the specific graph construction employed. Several recent works have proposed architectural modifications to mitigate this effect, including the use of normalized autoencoders [30]. Exploring such approaches in combination with sparse and rigidity-motivated graph structures represents an interesting direction for future studies.

While the maximum SIC values achieved by autoencoder-based methods are often modest, this behavior is intrinsic to the paradigm and has been discussed in the literature. We note that here, we focus on relative performance improvements related to the graph structure, rather than on absolute sensitivity or discovery potential. A more detailed quantitative comparison with weakly supervised methods, particularly in the transition region of signal fractions, is left for future work.

4 Conclusions and outlook

Anomaly detection plays an increasingly important role in searches for new physics at collider experiments, where the goal is to identify rare events without relying on detailed signal models. In this work, we explored how graph-theoretic insights can inform the design of machine learning architectures for this task, with a particular focus on graph autoencoders constructed from sparse graphs. Motivated by the use of relative information instead of absolute particle positions, we investigated different sparse graph construction techniques and their impact on performance for anomaly detection and jet classification tasks, in comparison to fully connected graphs.

We found that the choice of graph used to represent a jet of particles can enhance the performance of machine learning models by embedding physically motivated inductive biases into the architecture. In particular, we focused on (1) locally rigid Laman graphs, and (2) different unique graph constructions that are globally rigid and admit only a single embedding in the rapidity-azimuth plane. As an exemplary application, we trained a graph-based autoencoder to reconstruct both particle features and pairwise geometric information using the data set from the LHC Olympics anomaly detection challenge. We observed that certain *unique- k* graphs outperform fully connected graphs, resulting in a higher maximum Significance Improvement Characteristic (SIC) curve. Our results show that not only the sparsity of the graph, but also its specific connectivity structure, is critical for improved performance. Additionally, we explored the effect of clustering particles into subjets, allowing for a smooth interpolation between low- and high-level representations. Similar to the use of relatively sparse graphs, we found that the autoencoder performance peaks at intermediate levels of clustering with $n_{\text{subjets}} \approx 30$.

These results motivate further exploration of how graph-theoretic insights can enhance machine learning approaches in collider physics. In particular, the connection between graph theory and computational complexity may enhance the interpretability of machine learning algorithms and inform the development of architectures with improved performance. Further investigations are also needed to study the dependence of the anomaly detection performance

on the type of signal. In addition, it would be interesting to investigate whether similar graph-based methods can benefit other anomaly detection strategies.

Acknowledgments

We would like to thank Andrew Larkoski and James Mulligan for collaborating during the early stages of this work. In addition, we would like to thank Joseph Mitchell, Vasilis Belis, and Nobuo Sato for helpful discussions. DA is supported by the NSF Grant PHY2210533 and in part by the Onassis Foundation. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility using NERSC awards NP-ERCAP0033891 and NP-ERCAP0031584. MP is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under the contract DE-AC02-05CH11231. FR is supported by the DOE, Office of Science, Office of Nuclear Physics, Early Career Program under contract No. DE-SC0024358/DE-SC0025881. We thank the Institute for Nuclear Theory at the University of Washington for its kind hospitality and stimulating research environment. This research was supported in part by the INT's U.S. Department of Energy grant No. DE-FG02-00ER41132.

A Applications to jet classification

In this section, we apply the sparse graph constructions introduced in section 2.3 to different jet classification tasks. Unlike in the anomaly detection task discussed above, we do not observe an improvement in classification performance when using sparse graphs compared to fully connected graphs. Nevertheless, exploring different graph construction techniques can provide useful insights. Here, we focus only on graph constructions based on the particles inside the jet, since jet classification using subjects has already been discussed in ref. [63] in the context of deep sets.

A.1 Data sets

We consider three representative binary jet classification tasks in high-energy physics: quark vs. gluon jets, Z vs. QCD jets, and top vs. QCD jets. See refs. for more details [67, 98–125]. For the quark vs. gluon classification task, we use the data set provided in ref. [126], which consists of 2M jets with transverse momenta $p_T \in [500, 550]$ GeV, rapidity $|\eta| < 1.7$. Jets are clustered using the anti- k_T algorithm [87] with radius parameter $R = 0.4$. The jets were generated with PYTHIA8 [127] at a center-of-mass energy of $\sqrt{s} = 14$ TeV. For the Z vs. QCD and top vs. QCD classification tasks, we use the JetClass data set of refs. [89, 128]. The signal processes for these tasks correspond to boosted $Z \rightarrow q\bar{q}$ and $t \rightarrow bqq'$ decays, respectively. The background QCD jets are simulated using processes such as $q\bar{q} \rightarrow Z(\rightarrow \nu\bar{\nu}) + g$ and $q\bar{q} \rightarrow Z(\rightarrow \nu\bar{\nu}) + (uds)$. The simulation was carried out with PYTHIA8 [85], while detector effects are modelled using DELPHES [129] with the CMS detector card. The jet reconstruction is performed using the anti- k_T algorithm with $R = 0.8$, transverse momentum $p_T \in [500, 1000]$ GeV, and rapidity $|\eta| < 2$. For all three tasks, we use 2M jets. Each data set is split into training, validation, and test sets using an 80%, 10%, 10% split.

<i>AUC</i>	<i>Laman</i>	<i>unique-3</i>	<i>ParT</i> ($p_T, \eta, \phi, \theta_{ij}$)	<i>ParT</i> (p_T, θ_{ij})
<i>q vs. g</i>	0.8989 ± 0.0007	0.9007 ± 0.0005	0.9032 ± 0.0009	0.9041 ± 0.0006
<i>Z vs. QCD</i>	0.9439 ± 0.0006	0.9460 ± 0.0003	0.9483 ± 0.0006	0.9480 ± 0.0007
<i>t vs. QCD</i>	0.9829 ± 0.0004	0.9841 ± 0.0003	0.9865 ± 0.0005	0.9862 ± 0.0005

Table 1. The performance of the different classifiers quantified in terms of the AUC for the three discrimination tasks: quark vs. gluon, Z vs. QCD, and top vs. QCD jets.

A.2 Graph neural network classifier

To embed the different graph structures into a classifier and compare with current state-of-the-art results, we use a modified version of the Particle Transformer (**ParT**) classifier [89]. In the original **ParT**, there are two inputs: the *particle* input, which includes features for each particle, and the *interaction* input, which captures features of particle pairs. As in the anomaly detection task, we aim to restrict the machine’s access to relative distances rather than absolute positions in the rapidity-azimuth plane. Therefore, we limit the *particle* input of the **ParT** architecture to the transverse momenta of all particles inside the jet. In addition, we include non-zero *interaction* features in **ParT** only for particle pairs connected by an edge in the corresponding graph construction [130]. In contrast to the GNN used in the graph autoencoder discussed in section 3.2, every particle pair in the **ParT** architecture is connected via an attention layer that uses node features, but does not access their relative distances unless it is included in the input graph. The pairwise features include logarithmic versions of all the variables listed in eq. (3.1), as well as

$$m^2 = (E_i + E_j)^2 - |\vec{p}_i + \vec{p}_j|^2. \quad (\text{A.1})$$

In the limit of highly boosted jets, m^2 depends only on the pairwise angles θ_{ij} and the transverse momenta p_{Ti} . These modifications allow us to use the **ParT** architecture while incorporating the modified graph structure.

The initial particle embedding is processed sequentially through a stack of particle attention blocks, which refine the embeddings via a multi-head self-attention mechanism. The interaction matrix acts as a bias term for the pre-softmax attention weights and is applied identically across all particle attention blocks, effectively serving as a residual connection [91]. The final embedding is then passed to two class attention blocks. The resulting class token is processed through a single-layer MLP and a softmax layer to generate the final classification scores. We implement the architectures in PyTorch [94], using 8 particle attention blocks and 2 class attention blocks. The training is performed for 20 epochs, with a batch size of 512, using the AdamW optimizer [95] and a linear annealing scheduler with a warm-up phase. The initial learning rate is $2 \cdot 10^{-4}$, which linearly increases over the first 2 epochs to a peak of $2 \cdot 10^{-3}$, and then it linearly decreases to $4 \cdot 10^{-5}$.

A.3 Numerical results and interpretability

We assess the performance of the classifier for the three binary classification tasks using the obtained AUC as the metric. Statistical uncertainties are estimated by training the model four times for each task and computing the standard deviation across these runs. The

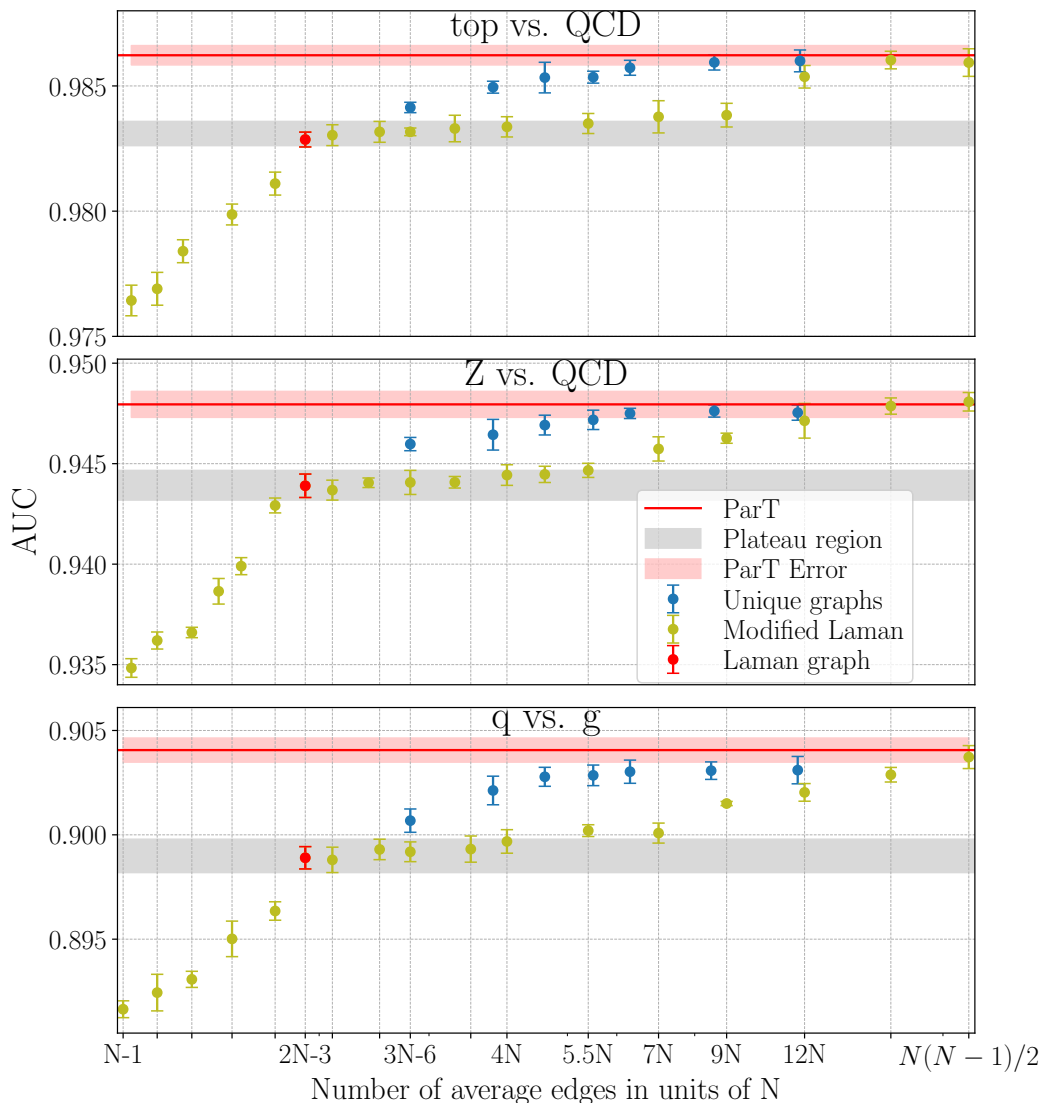


Figure 12. Performance of different graph-based jet classifiers. The obtained AUC is shown as a function of the average number of edges that the input graph has in units of the particle number N inside a jet (log scale). From top to bottom, the results are shown for: top vs QCD, Z vs QCD, and quark vs. gluon jets.

AUC results for all three classification tasks are summarized in table 1. In the first two columns, we report the results of the modified ParT classifier using Laman and *unique-3* graph constructions, respectively. The third column shows the performance of the original ParT classifier, where each node includes full particle information (p_T, η, ϕ) . In the fourth column, the node information is restricted to only the transverse momentum p_T , while the classifier is provided with relative angular distances θ_{ij} between all particle pairs. First, we observe that the performance of ParT using either absolute positions or only relative distances agrees within the estimated uncertainties for all three classification tasks. This observation is in line with the expectation that the relevant physics of jet classification is encoded in the relative information of particles instead of absolute positions. Next, while

the performance gap between the Laman graph-based classifier and the original **ParT** is statistically significant, the AUC values differ by only about 0.3 – 0.4%. This is a remarkably small difference, given that the Laman graph-based classifier has access to roughly an order of magnitude fewer pairwise distances θ_{ij} . For an average jet particle multiplicity of $\langle N \rangle \approx 50$ at the LHC, the average ratio of pairwise angles in a fully connected graph to those in a Laman graph is approximately $(N(N-1)/2)/(2N-3) \approx 12$. The classifier based on the *unique-3* graph outperforms the one using Laman graphs, though a small but statistically significant difference remains compared to the original **ParT**. However, as we increase k in the *unique-k* graph constructions, this performance gap closes, as illustrated in figure 12.

To illustrate that the specific choice of graph construction impacts how quickly the performance gap closes as the number of edges increases, we show in figure 12 the AUC for all three classification tasks as a function of the average number of edges in the input graph. Similar to figure 11, we compare the *unique-k* graphs to the modified Laman graphs. Here, the starting point is the Laman graph, indicated by the red marker. The yellow markers represent the AUC performance when edges are either added to or removed from the Laman graph. To add edges, we calculate the pairwise distances of all particles in a jet that are not yet connected, and we progressively connect the nearest particles with an edge. This process continues until the graph becomes fully connected. To remove edges, we randomly select a node and delete the edge connecting it to the hardest particle. We ensure that the entire graph is still connected. Therefore, if removing an edge would disconnect the graph, a different node is chosen. This allows us to remove up to $N-2$ edges, ultimately resulting in a tree graph with $N-1$ edges. In each panel of figure 12, the leftmost yellow marker corresponds to the tree graph, and the rightmost yellow marker shows the result for the fully connected graph. As expected, removing edges from the Laman graph leads to a steep drop in performance, consistent with the fact that the Laman graph represents the minimal configuration required for local rigidity. A further reduction in connectivity leads to a loss of information. Interestingly, randomly adding edges to the Laman graph does not immediately improve performance. Instead, we observe a plateau, indicated by the gray shaded band in each panel of figure 12. The AUC starts to increase only after adding approximately an order of magnitude more edges. Eventually, the performance matches that of the fully connected graph. Instead, the blue markers show the AUC achieved using various *unique-k* graph constructions for $k = 3, 4, 5, 6, 7, 10, 15$. Across all classification tasks, we find that peak performance is reached with significantly fewer edges compared to the modified Laman graphs. This indicates that insights from graph theory may aid in both optimizing machine learning-based classifiers and interpreting their performance.

Data Availability Statement. This article has no associated data or the data will not be deposited.

Code Availability Statement. The code developed for this work is available on GitHub.

- The Graph Autoencoder for anomaly detection can be found at: <https://github.com/DimAthanasakos/Graph-AutoEncoder-for-Model-Agnostic-Anomaly-Detection>.
- The code for jet classification, as described in the appendix, is available at: <https://github.com/DimAthanasakos/Graphs-and-Jet-Classification>.

Open Access. This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

References

- [1] CMS collaboration, *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, *Phys. Lett. B* **716** (2012) 30 [[arXiv:1207.7235](https://arxiv.org/abs/1207.7235)] [[INSPIRE](https://inspirehep.net/literature/1207723)].
- [2] ATLAS collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett. B* **716** (2012) 1 [[arXiv:1207.7214](https://arxiv.org/abs/1207.7214)] [[INSPIRE](https://inspirehep.net/literature/1207714)].
- [3] CMS collaboration, *Search for supersymmetry in proton-proton collisions at 13 TeV in final states with jets and missing transverse momentum*, *JHEP* **10** (2019) 244 [[arXiv:1908.04722](https://arxiv.org/abs/1908.04722)] [[INSPIRE](https://inspirehep.net/literature/1908047)].
- [4] CMS collaboration, *Search for Supersymmetry with a Compressed Mass Spectrum in Events with a Soft τ Lepton, a Highly Energetic Jet, and Large Missing Transverse Momentum in Proton-Proton Collisions at $\sqrt{s} = 13$ TeV*, *Phys. Rev. Lett.* **124** (2020) 041803 [[arXiv:1910.01185](https://arxiv.org/abs/1910.01185)] [[INSPIRE](https://inspirehep.net/literature/1910011)].
- [5] ATLAS collaboration, *Search for pair production of gluinos decaying via stop and sbottom in events with b-jets and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Phys. Rev. D* **94** (2016) 032003 [[arXiv:1605.09318](https://arxiv.org/abs/1605.09318)] [[INSPIRE](https://inspirehep.net/literature/1605093)].
- [6] CMS collaboration, *Search for supersymmetry in the multijet and missing transverse momentum final state in pp collisions at 13 TeV*, *Phys. Lett. B* **758** (2016) 152 [[arXiv:1602.06581](https://arxiv.org/abs/1602.06581)] [[INSPIRE](https://inspirehep.net/literature/1602065)].
- [7] ATLAS collaboration, *Search for dark matter at $\sqrt{s} = 13$ TeV in final states containing an energetic photon and large missing transverse momentum with the ATLAS detector*, *Eur. Phys. J. C* **77** (2017) 393 [[arXiv:1704.03848](https://arxiv.org/abs/1704.03848)] [[INSPIRE](https://inspirehep.net/literature/1704038)].
- [8] ATLAS collaboration, *Search for new phenomena in final states with an energetic jet and large missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector*, *Phys. Rev. D* **94** (2016) 032005 [[arXiv:1604.07773](https://arxiv.org/abs/1604.07773)] [[INSPIRE](https://inspirehep.net/literature/1604077)].
- [9] G. Kasieczka et al., *The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics*, *Rept. Prog. Phys.* **84** (2021) 124201 [[arXiv:2101.08320](https://arxiv.org/abs/2101.08320)] [[INSPIRE](https://inspirehep.net/literature/2101083)].
- [10] G. Kasieczka, B. Nachman and D. Shih, *Official datasets for lhc olympics 2020 anomaly detection challenge*, November (2019) [[DOI:10.5281/zenodo.4536624](https://doi.org/10.5281/zenodo.4536624)].

- [11] R. Gambhir, R. Mastandrea, B. Nachman and J. Thaler, *Isolating Unisolated Upsilon's with Anomaly Detection in CMS Open Data*, *Phys. Rev. Lett.* **135** (2025) 021902 [[arXiv:2502.14036](#)] [[INSPIRE](#)].
- [12] T. Aarrestad et al., *The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider*, *SciPost Phys.* **12** (2022) 043 [[arXiv:2105.14027](#)] [[INSPIRE](#)].
- [13] ATLAS collaboration, *Weakly supervised anomaly detection for resonant new physics in the dijet final state using proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Phys. Rev. D* **112** (2025) 072009 [[arXiv:2502.09770](#)] [[INSPIRE](#)].
- [14] O. Knapp et al., *Adversarially Learned Anomaly Detection on CMS Open Data: re-discovering the top quark*, *Eur. Phys. J. Plus* **136** (2021) 236 [[arXiv:2005.01598](#)] [[INSPIRE](#)].
- [15] ATLAS collaboration, *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** (2024) 081801 [[arXiv:2307.01612](#)] [[INSPIRE](#)].
- [16] A. Bhardwaj et al., *Equivariant, safe and sensitive — graph networks for new physics*, *JHEP* **07** (2024) 245 [[arXiv:2402.12449](#)] [[INSPIRE](#)].
- [17] V. Belis, P. Odagiu and T.K. Aarrestad, *Machine learning for anomaly detection in particle physics*, *Rev. Phys.* **12** (2024) 100091 [[arXiv:2312.14190](#)] [[INSPIRE](#)].
- [18] D.P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, [arXiv:1312.6114](#) [[INSPIRE](#)].
- [19] O. Cerri et al., *Variational Autoencoders for New Physics Mining at the Large Hadron Collider*, *JHEP* **05** (2019) 036 [[arXiv:1811.10276](#)] [[INSPIRE](#)].
- [20] T. Cheng et al., *Variational autoencoders for anomalous jet tagging*, *Phys. Rev. D* **107** (2023) 016002 [[arXiv:2007.01850](#)] [[INSPIRE](#)].
- [21] B.M. Dillon, T. Plehn, C. Sauer and P. Sorrenson, *Better Latent Spaces for Better Autoencoders*, *SciPost Phys.* **11** (2021) 061 [[arXiv:2104.08291](#)] [[INSPIRE](#)].
- [22] T. Heimel, G. Kasieczka, T. Plehn and J.M. Thompson, *QCD or What?*, *SciPost Phys.* **6** (2019) 030 [[arXiv:1808.08979](#)] [[INSPIRE](#)].
- [23] M. Farina, Y. Nakai and D. Shih, *Searching for New Physics with Deep Autoencoders*, *Phys. Rev. D* **101** (2020) 075021 [[arXiv:1808.08992](#)] [[INSPIRE](#)].
- [24] B. Bortolato, A. Smolkovič, B.M. Dillon and J.F. Kamenik, *Bump hunting in latent space*, *Phys. Rev. D* **105** (2022) 115009 [[arXiv:2103.06595](#)] [[INSPIRE](#)].
- [25] L. Vaslin, V. Barra and J. Donini, *GAN-AE: an anomaly detection algorithm for New Physics search in LHC data*, *Eur. Phys. J. C* **83** (2023) 1008 [[arXiv:2305.15179](#)] [[INSPIRE](#)].
- [26] P. Jawahar et al., *Improving Variational Autoencoders for New Physics Detection at the LHC With Normalizing Flows*, *Front. Big Data* **5** (2022) 803685 [[arXiv:2110.08508](#)] [[INSPIRE](#)].
- [27] T. Finke et al., *Autoencoders for unsupervised anomaly detection in high energy physics*, *JHEP* **06** (2021) 161 [[arXiv:2104.09051](#)] [[INSPIRE](#)].
- [28] P. Laguarda et al., *Detection of anomalies amongst LIGO's glitch populations with autoencoders*, *Class. Quant. Grav.* **41** (2024) 055004 [[arXiv:2310.03453](#)] [[INSPIRE](#)].
- [29] A. Blance, M. Spannowsky and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, *JHEP* **10** (2019) 047 [[arXiv:1905.10384](#)] [[INSPIRE](#)].
- [30] B.M. Dillon et al., *A normalized autoencoder for LHC triggers*, *SciPost Phys. Core* **6** (2023) 074 [[arXiv:2206.14225](#)] [[INSPIRE](#)].

- [31] J. Schuhmacher et al., *Unravelling physics beyond the standard model with classical and quantum anomaly detection*, *Mach. Learn. Sci. Tech.* **4** (2023) 045031 [[arXiv:2301.10787](#)] [[INSPIRE](#)].
- [32] S. Tsan et al., *Particle Graph Autoencoders and Differentiable, Learned Energy Mover's Distance*, in the proceedings of the *35th Conference on Neural Information Processing Systems*, Online Conference, Canada, December 06–14 (2021) [[arXiv:2111.12849](#)] [[INSPIRE](#)].
- [33] P.T. Komiske, E.M. Metodiev and J. Thaler, *Metric Space of Collider Events*, *Phys. Rev. Lett.* **123** (2019) 041801 [[arXiv:1902.02346](#)] [[INSPIRE](#)].
- [34] O. Atkinson et al., *Anomaly detection with convolutional Graph Neural Networks*, *JHEP* **08** (2021) 080 [[arXiv:2105.07988](#)] [[INSPIRE](#)].
- [35] M.M. Bronstein, J. Bruna, T. Cohen and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, [arXiv:2104.13478](#) [[INSPIRE](#)].
- [36] Z. Hao, R. Kansal, J. Duarte and N. Chernyavskaya, *Lorentz group equivariant autoencoders*, *Eur. Phys. J. C* **83** (2023) 485 [[arXiv:2212.07347](#)] [[INSPIRE](#)].
- [37] R. Liu et al., *Fast Particle-based Anomaly Detection Algorithm with Variational Autoencoder*, in the proceedings of the *37th Conference on Neural Information Processing Systems*, New Orleans, U.S.A., December 10–16 (2023) [[arXiv:2311.17162](#)] [[INSPIRE](#)].
- [38] E.M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: Learning from mixed samples in high energy physics*, *JHEP* **10** (2017) 174 [[arXiv:1708.02949](#)] [[INSPIRE](#)].
- [39] L.M. Dery, B. Nachman, F. Rubbo and A. Schwartzman, *Weakly Supervised Classification in High Energy Physics*, *JHEP* **05** (2017) 145 [[arXiv:1702.00414](#)] [[INSPIRE](#)].
- [40] T. Finke, M. Krämer, M. Lipp and A. Mück, *Boosting mono-jet searches with model-agnostic machine learning*, *JHEP* **08** (2022) 015 [[arXiv:2204.11889](#)] [[INSPIRE](#)].
- [41] J.H. Collins, K. Howe and B. Nachman, *Extending the search for new resonances with machine learning*, *Phys. Rev. D* **99** (2019) 014038 [[arXiv:1902.02634](#)] [[INSPIRE](#)].
- [42] O. Amram and C.M. Suarez, *Tag N' Train: a technique to train improved classifiers on unlabeled data*, *JHEP* **01** (2021) 153 [[arXiv:2002.12376](#)] [[INSPIRE](#)].
- [43] A. Andreassen, B. Nachman and D. Shih, *Simulation Assisted Likelihood-free Anomaly Detection*, *Phys. Rev. D* **101** (2020) 095004 [[arXiv:2001.05001](#)] [[INSPIRE](#)].
- [44] A. Hallin et al., *Classifying anomalies through outer density estimation*, *Phys. Rev. D* **106** (2022) 055006 [[arXiv:2109.00546](#)] [[INSPIRE](#)].
- [45] E. Buhmann et al., *Full phase space resonant anomaly detection*, *Phys. Rev. D* **109** (2024) 055015 [[arXiv:2310.06897](#)] [[INSPIRE](#)].
- [46] A. Hallin et al., *Resonant anomaly detection without background sculpting*, *Phys. Rev. D* **107** (2023) 114012 [[arXiv:2210.14924](#)] [[INSPIRE](#)].
- [47] C.L. Cheng, G. Singh and B. Nachman, *Incorporating Physical Priors into Weakly Supervised Anomaly Detection*, *Phys. Rev. Lett.* **135** (2025) 021801 [[arXiv:2405.08889](#)] [[INSPIRE](#)].
- [48] T. Golling et al., *The interplay of machine learning-based resonant anomaly detection methods*, *Eur. Phys. J. C* **84** (2024) 241 [[arXiv:2307.11157](#)] [[INSPIRE](#)].
- [49] T. Golling, S. Klein, R. Mastandrea and B. Nachman, *Flow-enhanced transportation for anomaly detection*, *Phys. Rev. D* **107** (2023) 096025 [[arXiv:2212.11285](#)] [[INSPIRE](#)].
- [50] R.T. d'Agnolo et al., *Learning new physics from an imperfect machine*, *Eur. Phys. J. C* **82** (2022) 275 [[arXiv:2111.13633](#)] [[INSPIRE](#)].

- [51] J.H. Collins, P. Martín-Ramiro, B. Nachman and D. Shih, *Comparing weak- and unsupervised methods for resonant anomaly detection*, *Eur. Phys. J. C* **81** (2021) 617 [[arXiv:2104.02092](#)] [[INSPIRE](#)].
- [52] K. Fraser et al., *Challenges for unsupervised anomaly detection in particle physics*, *JHEP* **03** (2022) 066 [[arXiv:2110.06948](#)] [[INSPIRE](#)].
- [53] M. van Beekveld et al., *Combining outlier analysis algorithms to identify new physics at the LHC*, *JHEP* **09** (2021) 024 [[arXiv:2010.07940](#)] [[INSPIRE](#)].
- [54] S. Caron, L. Hendriks and R. Verheyen, *Rare and Different: Anomaly Scores from a combination of likelihood and out-of-distribution models to detect new physics at the LHC*, *SciPost Phys.* **12** (2022) 077 [[arXiv:2106.10164](#)] [[INSPIRE](#)].
- [55] J.H. Collins, *An Exploration of Learnt Representations of W Jets*, [arXiv:2109.10919](#) [[INSPIRE](#)].
- [56] V. Mikuni, B. Nachman and D. Shih, *Online-compatible unsupervised nonresonant anomaly detection*, *Phys. Rev. D* **105** (2022) 055006 [[arXiv:2111.06417](#)] [[INSPIRE](#)].
- [57] L. Bradshaw, S. Chang and B. Ostdiek, *Creating simple, interpretable anomaly detectors for new physics in jet substructure*, *Phys. Rev. D* **106** (2022) 035014 [[arXiv:2203.01343](#)] [[INSPIRE](#)].
- [58] S. Roche et al., *Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays*, *Nature Commun.* **15** (2024) 3527 [[arXiv:2304.03836](#)] [[INSPIRE](#)].
- [59] K. Bai, R. Mastandrea and B. Nachman, *Non-resonant anomaly detection with background extrapolation*, *JHEP* **04** (2024) 059 [[arXiv:2311.12924](#)] [[INSPIRE](#)].
- [60] J. Thaler and K. Van Tilburg, *Identifying Boosted Objects with N-subjettiness*, *JHEP* **03** (2011) 015 [[arXiv:1011.2268](#)] [[INSPIRE](#)].
- [61] I.W. Stewart, F.J. Tackmann and W.J. Waalewijn, *N-Jettiness: An Inclusive Event Shape to Veto Jets*, *Phys. Rev. Lett.* **105** (2010) 092002 [[arXiv:1004.2489](#)] [[INSPIRE](#)].
- [62] S. Catani, Y.L. Dokshitzer, M.H. Seymour and B.R. Webber, *Longitudinally invariant K_t clustering algorithms for hadron hadron collisions*, *Nucl. Phys. B* **406** (1993) 187 [[INSPIRE](#)].
- [63] D. Athanasakos et al., *Is infrared-collinear safe information all you need for jet classification?*, *JHEP* **07** (2024) 257 [[arXiv:2305.08979](#)] [[INSPIRE](#)].
- [64] K. Datta and A. Larkoski, *How Much Information is in a Jet?*, *JHEP* **06** (2017) 073 [[arXiv:1704.08249](#)] [[INSPIRE](#)].
- [65] K. Datta and A.J. Larkoski, *Novel Jet Observables from Machine Learning*, *JHEP* **03** (2018) 086 [[arXiv:1710.01305](#)] [[INSPIRE](#)].
- [66] K. Datta, A. Larkoski and B. Nachman, *Automating the Construction of Jet Observables with Machine Learning*, *Phys. Rev. D* **100** (2019) 095016 [[arXiv:1902.07180](#)] [[INSPIRE](#)].
- [67] P.T. Komiske, E.M. Metodiev and J. Thaler, *Energy flow polynomials: A complete linear basis for jet substructure*, *JHEP* **04** (2018) 013 [[arXiv:1712.07124](#)] [[INSPIRE](#)].
- [68] T. Cai et al., *The phase space distance between collider events*, *JHEP* **09** (2024) 054 [[arXiv:2405.16698](#)] [[INSPIRE](#)].
- [69] A.J. Larkoski and T. Melia, *Covariantizing phase space*, *Phys. Rev. D* **102** (2020) 094014 [[arXiv:2008.06508](#)] [[INSPIRE](#)].
- [70] Y.S. Lai, J. Mulligan, M. Płoskoń and F. Ringer, *The information content of jet quenching and machine learning assisted observable design*, *JHEP* **10** (2022) 011 [[arXiv:2111.14589](#)] [[INSPIRE](#)].

- [71] G. Stein, U. Seljak and B. Dai, *Unsupervised in-distribution anomaly detection of new physics through conditional density estimation*, in the proceedings of the *34th Conference on Neural Information Processing Systems*, Online Conference, Canada, December 06–12 (2020) [[arXiv:2012.11638](#)] [[INSPIRE](#)].
- [72] G. Laman, *On graphs and rigidity of plane skeletal structures*, *J. Eng. Math.* **4** (1970) 331.
- [73] B. Hendrickson, *Conditions for Unique Graph Realizations*, *SIAM J. Comput.* **21** (1992) 65.
- [74] R. Connelly, *Generic Global Rigidity*, *Discrete Comput. Geom.* **33** (2004) 549.
- [75] S.J. Gortler, A.D. Healy and D.P. Thurston, *Characterizing Generic Global Rigidity*, [arXiv:0710.0926](#) [[DOI:10.1016/j.adhoc.2011.06.016](#)].
- [76] J.B. Saxe *Embeddability of weighted graphs in k-space is strongly NP-hard*, in *17th Allerton Conf. Commun. Control Comput.*, (1979), pp. 480–489.
- [77] P. Biswas, H. Aghajan and Y. Ye, *Semidefinite Programming Algorithms for Sensor Network Localization using Angle Information*, in the proceedings of the *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005*, Pacific Grove, CA, U.S.A., 30 October – 2 November (2005) [[DOI:10.1109/acssc.2005.1599736](#)].
- [78] L. Henneberg, *Die graphische Statik der starren Systeme*, in *Sammlung von Lehrbüchern auf dem Gebiete der mathematischen Wissenschaften*, B.G. Teubner (1911).
- [79] R. Haas et al., *Planar minimally rigid graphs and pseudo-triangulations*, *Comput. Geom.* **31** (2005) 31.
- [80] Y.L. Dokshitzer, G.D. Leder, S. Moretti and B.R. Webber, *Better jet clustering algorithms*, *JHEP* **08** (1997) 001 [[hep-ph/9707323](#)] [[INSPIRE](#)].
- [81] M. Wobisch and T. Wengler, *Hadronization corrections to jet cross-sections in deep inelastic scattering*, in the proceedings of the *Workshop on Monte Carlo Generators for HERA Physics (Plenary Starting Meeting)*, Hamburg, Germany, April 27–30 (1998) [[hep-ph/9907280](#)] [[INSPIRE](#)].
- [82] B. Jackson and T. Jordán, *Connected rigidity matroids and unique realizations of graphs*, *J. Combin. Theor. B* **94** (2005) 1.
- [83] B. Nachman and D. Shih, *Anomaly Detection with Density Estimation*, *Phys. Rev. D* **101** (2020) 075042 [[arXiv:2001.04990](#)] [[INSPIRE](#)].
- [84] E.M. Metodiev, J. Thaler and R. Wynne, *Anomaly detection in collider physics via factorized observables*, *Phys. Rev. D* **110** (2024) 055012 [[arXiv:2312.00119](#)] [[INSPIRE](#)].
- [85] T. Sjöstrand, S. Mrenna and P.Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852 [[arXiv:0710.3820](#)] [[INSPIRE](#)].
- [86] M. Selvaggi, *DELPHES 3: A modular framework for fast-simulation of generic collider experiments*, *J. Phys. Conf. Ser.* **523** (2014) 012033 [[INSPIRE](#)].
- [87] M. Cacciari, G.P. Salam and G. Soyez, *The anti- k_t jet clustering algorithm*, *JHEP* **04** (2008) 063 [[arXiv:0802.1189](#)] [[INSPIRE](#)].
- [88] M. Cacciari, G.P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896 [[arXiv:1111.6097](#)] [[INSPIRE](#)].
- [89] H. Qu, C. Li and S. Qian, *Particle Transformer for Jet Tagging*, [arXiv:2202.03772](#) [[INSPIRE](#)].
- [90] Y. Wang et al., *Dynamic Graph CNN for Learning on Point Clouds*, [arXiv:1801.07829](#) [[INSPIRE](#)].
- [91] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, [arXiv:1512.03385](#) [[DOI:10.1109/CVPR.2016.90](#)] [[INSPIRE](#)].

- [92] J. Gallicchio et al., *Multivariate discrimination and the Higgs + W/Z search*, *JHEP* **04** (2011) 069 [[arXiv:1010.3698](#)] [[INSPIRE](#)].
- [93] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, [arXiv:1502.03167](#) [[INSPIRE](#)].
- [94] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, [arXiv:1912.01703](#) [[INSPIRE](#)].
- [95] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, [arXiv:1412.6980](#) [[INSPIRE](#)].
- [96] J. Batson, C.G. Haaf, Y. Kahn and D.A. Roberts, *Topological Obstructions to Autoencoding*, *JHEP* **04** (2021) 280 [[arXiv:2102.08380](#)] [[INSPIRE](#)].
- [97] V.S. Ngairangbam, B. Rozwoda, K. Sakurai and M. Spannowsky, *Enhancing anomaly detection with topology-aware autoencoders*, *Mach. Learn. Sci. Tech.* **6** (2025) 045051 [[arXiv:2502.10163](#)] [[INSPIRE](#)].
- [98] L. Lonnblad, C. Peterson and T. Rognvaldsson, *Finding Gluon Jets With a Neural Trigger*, *Phys. Rev. Lett.* **65** (1990) 1321 [[INSPIRE](#)].
- [99] L. de Oliveira et al., *Jet-images — deep learning edition*, *JHEP* **07** (2016) 069 [[arXiv:1511.05190](#)] [[INSPIRE](#)].
- [100] P.T. Komiske, E.M. Metodiev and M.D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, *JHEP* **01** (2017) 110 [[arXiv:1612.01551](#)] [[INSPIRE](#)].
- [101] G. Kasieczka, T. Plehn, M. Russell and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, *JHEP* **05** (2017) 006 [[arXiv:1701.08784](#)] [[INSPIRE](#)].
- [102] G. Louppe, K. Cho, C. Becot and K. Cranmer, *QCD-Aware Recursive Neural Networks for Jet Physics*, *JHEP* **01** (2019) 057 [[arXiv:1702.00748](#)] [[INSPIRE](#)].
- [103] P.T. Komiske, E.M. Metodiev and J. Thaler, *Energy Flow Networks: Deep Sets for Particle Jets*, *JHEP* **01** (2019) 121 [[arXiv:1810.05165](#)] [[INSPIRE](#)].
- [104] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, *Phys. Rev. D* **101** (2020) 056019 [[arXiv:1902.08570](#)] [[INSPIRE](#)].
- [105] G. Kasieczka, S. Marzani, G. Soyez and G. Stagnitto, *Towards Machine Learning Analytics for Jet Substructure*, *JHEP* **09** (2020) 195 [[arXiv:2007.04319](#)] [[INSPIRE](#)].
- [106] F.A. Dreyer and H. Qu, *Jet tagging in the Lund plane with graph networks*, *JHEP* **03** (2021) 052 [[arXiv:2012.08526](#)] [[INSPIRE](#)].
- [107] A. Butter et al., *The Machine Learning landscape of top taggers*, *SciPost Phys.* **7** (2019) 014 [[arXiv:1902.09914](#)] [[INSPIRE](#)].
- [108] A. Butter, G. Kasieczka, T. Plehn and M. Russell, *Deep-learned Top Tagging with a Lorentz Layer*, *SciPost Phys.* **5** (2018) 028 [[arXiv:1707.08966](#)] [[INSPIRE](#)].
- [109] Y.-C.J. Chen, C.-W. Chiang, G. Cottin and D. Shih, *Boosted W and Z tagging with jet charge and deep learning*, *Phys. Rev. D* **101** (2020) 053001 [[arXiv:1908.08256](#)] [[INSPIRE](#)].
- [110] J.Y. Araz and M. Spannowsky, *Combine and Conquer: Event Reconstruction with Bayesian Ensemble Neural Networks*, *JHEP* **04** (2021) 296 [[arXiv:2102.01078](#)] [[INSPIRE](#)].
- [111] S. Gong et al., *An efficient Lorentz equivariant graph neural network for jet tagging*, *JHEP* **07** (2022) 030 [[arXiv:2201.08187](#)] [[INSPIRE](#)].
- [112] M.D. Schwartz, *Modern Machine Learning and Particle Physics*, [arXiv:2103.12226](#) [[DOI:10.1162/99608f92.beeb1183](#)] [[INSPIRE](#)].
- [113] A. Khot, M.S. Neubauer and A. Roy, *A detailed study of interpretability of deep neural network based top taggers*, *Mach. Learn. Sci. Tech.* **4** (2023) 035003 [[arXiv:2210.04371](#)] [[INSPIRE](#)].

- [114] J. Lin, M. Freytsis, I. Moutl and B. Nachman, *Boosting $H \rightarrow b\bar{b}$ with Machine Learning*, *JHEP* **10** (2018) 101 [[arXiv:1807.10768](#)] [[INSPIRE](#)].
- [115] C.K. Khosa and S. Marzani, *Higgs boson tagging with the Lund jet plane*, *Phys. Rev. D* **104** (2021) 055043 [[arXiv:2105.03989](#)] [[INSPIRE](#)].
- [116] Y. Nakai, D. Shih and S. Thomas, *Strange Jet Tagging*, [arXiv:2003.09517](#) [[INSPIRE](#)].
- [117] A. Bogatskiy et al., *Explainable equivariant neural networks for particle physics: PELICAN*, *JHEP* **03** (2024) 113 [[arXiv:2307.16506](#)] [[INSPIRE](#)].
- [118] CMS collaboration, *Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques*, *2020 JINST* **15** P06005 [[arXiv:2004.08262](#)] [[INSPIRE](#)].
- [119] A. Romero et al., *Safety of Quark/Gluon Jet Classification*, [arXiv:2103.09103](#) [[INSPIRE](#)].
- [120] P. Konar, V.S. Ngairangbam and M. Spannowsky, *Energy-weighted message passing: an infra-red and collinear safe graph neural network algorithm*, *JHEP* **02** (2022) 060 [[arXiv:2109.14636](#)] [[INSPIRE](#)].
- [121] A.J. Larkoski and E.M. Metodiev, *A Theory of Quark vs. Gluon Discrimination*, *JHEP* **10** (2019) 014 [[arXiv:1906.01639](#)] [[INSPIRE](#)].
- [122] S. Choi, S.J. Lee and M. Perelstein, *Infrared Safety of a Neural-Net Top Tagging Algorithm*, *JHEP* **02** (2019) 132 [[arXiv:1806.01263](#)] [[INSPIRE](#)].
- [123] A. Bogatskiy et al., *Lorentz Group Equivariant Neural Network for Particle Physics*, [arXiv:2006.04780](#) [[INSPIRE](#)].
- [124] W. Esmail, A. Hammad and M. Nojiri, *IAFormer: Interaction-Aware Transformer network for collider data analysis*, [arXiv:2505.03258](#) [[INSPIRE](#)].
- [125] C. Krause, D. Wang and R. Winterhalder, *BitHEP — The Limits of Low-Precision ML in HEP*, *SciPost Phys.* **20** (2026) 038 [[arXiv:2504.03387](#)] [[INSPIRE](#)].
- [126] P. Komiske, E. Metodiev and J. Thaler, *Pythia8 quark and gluon jets for energy flow*, (2019) [[DOI:10.5281/zenodo.3164691](#)].
- [127] T. Sjöstrand et al., *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159 [[arXiv:1410.3012](#)] [[INSPIRE](#)].
- [128] H. Qu, C. Li and S. Qian, *Jetclass: A large-scale dataset for deep learning in jet physics*, (2022) [[DOI:10.5281/zenodo.6619768](#)].
- [129] DELPHES 3 collaboration, *Delphes 3: a modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [[arXiv:1307.6346](#)] [[INSPIRE](#)].
- [130] C. Ying et al., *Do transformers really perform badly for graph representation?*, *Adv. Neural Inf. Process. Syst.* **34** (2021) 28877.