Eur. Phys. J. C (2023) 83:485
https://doi.org/10.1140/epjc/s10052-023-11633-5

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Experimental Physics

Lorentz group equivariant autoencoders

Zichun Hao[1], **Raghav Kansal**[1,2,a], **Javier Duarte**[1], **Nadezda Chernyavskaya**[3]

[1] University of California San Diego, La Jolla, CA 92093, USA
[2] Fermi National Accelerator Laboratory, Batavia, IL 60510, USA
[3] European Organization for Nuclear Research (CERN), 1211 Geneva 23, Switzerland

Received: 24 December 2022 / Accepted: 17 May 2023

Abstract There has been significant work recently in developing machine learning (ML) models in high energy physics (HEP) for tasks such as classification, simulation, and anomaly detection. Often these models are adapted from those designed for datasets in computer vision or natural language processing, which lack inductive biases suited to HEP data, such as equivariance to its inherent symmetries. Such biases have been shown to make models more performant and interpretable, and reduce the amount of training data needed. To that end, we develop the Lorentz group autoencoder (LGAE), an autoencoder model equivariant with respect to the proper, orthochronous Lorentz group $\mathrm{SO}^+(3,1)$, with a latent space living in the representations of the group. We present our architecture and several experimental results on jets at the LHC and find it outperforms graph and convolutional neural network baseline models on several compression, reconstruction, and anomaly detection metrics. We also demonstrate the advantage of such an equivariant model in analyzing the latent space of the autoencoder, which can improve the explainability of potential anomalies discovered by such ML models.

1 Introduction

The increasingly large volume of data produced at the LHC and the new era of the High-Luminosity CERN Large Hadron Collider (LHC) poses a significant computational challenge in high energy physics (HEP). To face this, machine learning (ML) and deep neural networks (DNNs) are becoming powerful and ubiquitous tools for the analysis of particle collisions and their products, such as jets – collimated sprays of particles [1] produced in high energy collisions.

DNNs have been explored extensively for many tasks, such as classification [2–5], regression [6,7], track reconstruction [8–10], anomaly detection [11–17], and simulation [18–23].[1] In particular, there has been recent success using networks that incorporate key inductive biases of HEP data, such as infrared and colinear (IRC) safety via energy flow networks [28] or graph neural networks (GNNs) [29–31] and permutation symmetry and sparsity of jet constituents via GNNs [5,20,32].

Embedding such inductive biases and symmetries into DNNs can not only improve performance, as demonstrated in the references above, but also improve interpretability and reduce the amount of required training data. Hence, in this paper, we explore another fundamental symmetry of our data: equivariance to Lorentz transformations. Lorentz symmetry has been successfully exploited recently in HEP for jet classification [33–36], with competitive and even state-of-the-art (SOTA) results. We expand this work to the tasks of data compression and anomaly detection by incorporating the Lorentz symmetry into an autoencoder.

RK was partially supported by the LHC Physics Center at Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy (DOE). JD and RK were supported by the DOE, Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187, the DOE, Office of Advanced Scientific Computing Research under Award No. DE-SC0021396 (FAIR4HEP), and the NSF HDR Institute for Accelerating AI Algorithms for Data Driven Discovery (A3D3) under Cooperative Agreement OAC-2117997. NC was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369). This work was performed using the Pacific Research Platform Nautilus HyperCluster supported by NSF awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100 Gpbs networks.

[a] e-mail: rkansal@ucsd.edu (corresponding author)

[1] Interested readers can find comprehensive reviews in Ref. [24–26] and a living review in Ref. [27].

Published online: 09 June 2023

Autoencoders learn to encode and decode input data into a learned latent space, and thus have interesting applications in both data compression [37,38] and anomaly detection [11,13–17,39,40]. Both tasks are particularly relevant for HEP, the former to cope with the storage and processing of the ever-increasing data collected at the LHC, and the latter for model-independent searches for new physics. Incorporating Lorentz equivariance into an autoencoder has the potential to not only increase performance in both regards, but also provide a more interpretable latent space and reduce training data requirements. To this end, in this paper, we develop a Lorentz-group-equivariant autoencoder (LGAE) and explore its performance and interpretability. We also train alternative architectures, including GNNs and convolutional neural networks (CNNs), with different inherent symmetries and find the LGAE outperforms them on reconstruction and anomaly detection tasks.

The principal results of this work demonstrate (i) that the advantage of incorporating Lorentz equivariance extends beyond whole jet classification to applications with particle-level outputs and (ii) the interpretability of Lorentz-equiva riant models. The key challenges overcome in this work include: (i) training an equivariant autoencoder via particle-to-particle and permutation-invariant set-to-set losses (Sect. 4), (ii) defining a jet-level compression scheme for the latent space (Sect. 3), and (iii) optimizing the architecture for different tasks, such as reconstruction (Sect. 4.3) and anomaly detection (Sect. 4.4).

This paper is structured as follows. In Sect. 2, we discuss existing work, motivating the LGAE. We present the LGAE architecture in Sect. 3, and discuss experimental results on the reconstruction and anomaly detection of high energy jets in Sect. 4. We also demonstrate the interpretability of the model, by analyzing its latent space, and its data efficiency relative to baseline models. Finally, we conclude in Sect. 5.

2 Related work

In this section, we briefly review the large body of work on frameworks for equivariant neural networks in Sect. 2.1, recent progress in Lorentz-equivariant networks in Sect. 2.2, and finally, applications of autoencoders in HEP in Sect. 2.3.

2.1 Equivariant neural networks

A neural network $\mathrm{NN} : V \to W$ is said to be *equivariant* with respect to a group G if

$$\forall g \in G, v \in V : \mathrm{NN}(\rho_V(g) \cdot v) = \rho_W(g) \cdot \mathrm{NN}(v), \qquad (1)$$

where $\rho_V : G \to \mathrm{GL}(V)$ and $\rho_W : G \to \mathrm{GL}(W)$ are representations of G in spaces V and W respectively, where $\mathrm{GL}(X)$ is the general linear group of vector space X. The neural network is said to be *invariant* if ρ_W is a trivial representation, i.e. $\rho_W(g) = \mathbb{1}_W$ for all $g \in G$.

Equivariance has long been built into a number of successful DNN architectures, such as translation equivariance in CNNs, and permutation equivariance in GNNs [41]. Recently, equivariance in DNNs has been extended to a broader set of symmetries, such as those corresponding to the 2-dimensional special orthogonal SO(2) [42], the Euclidean E(2) [43], the 3-dimensional special orthogonal SO(3) [44], the 3-dimensional Euclidean E(3) [45,46] groups, and arbitrary matrix Lie groups [47].

Broadly, equivariance to a group G has been achieved either by extending the translation-equivariant convolutions in CNNs to more general symmetries with appropriately defined learnable filters [48–51], or by operating in the Fourier space of G, or a combination thereof. We employ the Fourier space approach, which uses the set of irreducible representations (irreps) of G as the basis for constructing equivariant maps [43,52,53].

2.2 Lorentz group equivariant neural networks

The Lorentz group O(3, 1) comprises the set of linear transformations between inertial frames with coincident origins. In this paper, we restrict ourselves to the special orthochronous Lorentz group $\mathrm{SO}^+(3, 1)$, which consists of all Lorentz transformations that preserve the orientation and direction of time. Lorentz symmetry, or invariance to transformations defined by the Lorentz group, is a fundamental symmetry of the data collected out of high-energy particle collisions.

There have been some recent advances in incorporating this symmetry into NNs. The Lorentz group network (LGN) [33] was the first DNN architecture developed to be equivariant to the $\mathrm{SO}^+(3, 1)$ group, with an architecture similar to that of a GNN, but operating entirely in Fourier space on objects in irreps of the Lorentz group, and using tensor products between irreps and Clebsch–Gordan decompositions to introduce non-linearities in the network. More recently, LorentzNet [34,35] uses a similar GNN framework for equivariance, with additional edge features – Minkowski inner products between node features – but restricting itself to only scalar and vector representations of the group. Both networks have been successful in jet classification, with LorentzNet achieving SOTA results in top quark and quark versus gluon classification, further demonstrating the benefit of incorporating physical inductive biases into network architectures. In this work, we build on top of the LGN framework to output not only scalars (e.g. jet class probabilities) but encode and reconstruct an input set of particles under the constraint of Lorentz group equivariance in an autoencoder-style architecture.

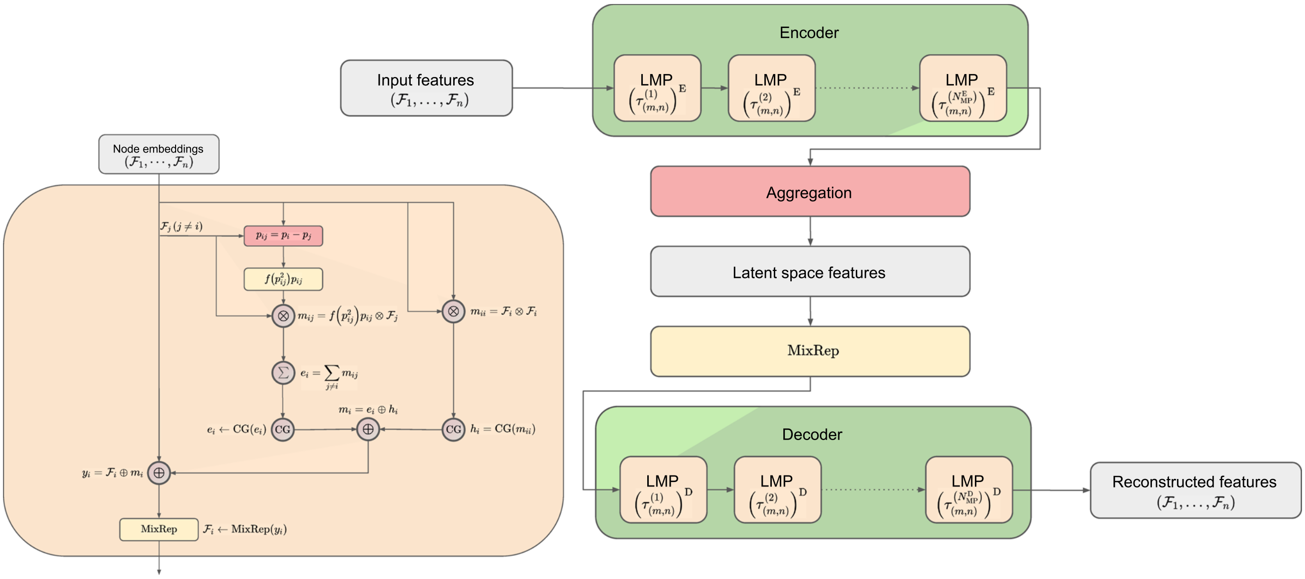

Input features
$(\mathcal{F}_1, \dots, \mathcal{F}_n)$
Encoder
LMP $\left(\tau^{(1)}_{(m,n)}\right)^{\mathrm{E}}$
LMP $\left(\tau^{(2)}_{(m,n)}\right)^{\mathrm{E}}$
LMP $\left(\tau^{(N^{\mathrm{E}}_{\mathrm{MP}})}_{(m,n)}\right)^{\mathrm{E}}$
Node embeddings
$(\mathcal{F}_1, \cdots, \mathcal{F}_n)$
Aggregation
$\mathcal{F}_j \, (j \neq i)$
$p_{ij} = p_i - p_j$
$f(p_{ij}^2) p_{ij}$
$m_{ij} = f\left(p_{ij}^2\right) p_{ij} \otimes \mathcal{F}_j$
$m_{ii} = \mathcal{F}_i \otimes \mathcal{F}_i$
Latent space features
$e_i = \sum_{j \neq i} m_{ij}$
MixRep
$m_i = e_i \oplus h_i$
$e_i \leftarrow \mathrm{CG}(e_i)$
CG
CG
$h_i = \mathrm{CG}(m_{ii})$
Decoder
$y_i = \mathcal{F}_i \oplus m_i$
LMP $\left(\tau^{(1)}_{(m,n)}\right)^{\mathrm{D}}$
LMP $\left(\tau^{(2)}_{(m,n)}\right)^{\mathrm{D}}$
LMP $\left(\tau^{(N^{\mathrm{D}}_{\mathrm{MP}})}_{(m,n)}\right)^{\mathrm{D}}$
Reconstructed features
$(\mathcal{F}_1, \dots, \mathcal{F}_n)$
MixRep
$\mathcal{F}_i \leftarrow \mathrm{MixRep}(y_i)$

Fig. 1 Individual Lorentz group equivariant message passing (LMP) layers are shown on the left, and the LGAE architecture is built out of LMPs on the right. Here, MixRep denotes the node-level operator that upsamples features in each (m, n) representation space to $\tau_{(m,n)}$ channels; it appears as W in Eq. (5)

2.3 Autoencoders in HEP

An autoencoder is an NN architecture comprised of an encoder, which maps the input into a, typically lower dimensional, latent space, and a decoder, which attempts to reconstruct the original input from the latent features. By using a lower dimensional latent space, an autoencoder can learn a smaller representation of data that captures salient properties [54], which can be valuable in HEP for compressing the significant volumes of data collected at the LHC [55].

This learned representation can also be exploited for later downstream tasks, such as anomaly detection, where an autoencoder is trained to reconstruct data considered "background" to our signal, with the expectation that it will reconstruct the signal poorly relative to the background. Thus, examining the reconstruction loss of a trained autoencoder may allow the identification of anomalous data.[2] This can be an advantage in searches for new physics, since instead of having to specify a particular signal hypothesis, a broader search can be performed for data incompatible with the background. This approach has been successfully demonstrated in Refs. [12,39,40,56–61].

Furthermore, there are many possible variations to the general autoencoder framework for alternative tasks [62,63], such as variational autoencoders (VAEs) [64], which are popular generative models. To our knowledge, while there have been some recent efforts at GNN-based autoencoder models [16,65], Lorentz equivariance has not yet been explored. In this work, we focus on data compression and anomaly detection but note that our model can be extended to further applications.

3 LGAE architecture

The LGAE is built out of Lorentz group-equivariant message passing (LMP) layers, which are identical to individual layers in the LGN [33]. We reinterpret them in the framework of message-passing neural networks [66], to highlight the connection to GNNs, and define them in Sect. 3.1. We then describe the encoder and decoder networks in Sects. 3.2 and 3.3, respectively. The LMP layers and LGAE architecture are depicted in Fig. 1. We provide the LGAE code, written in Python using the PyTorch ML framework [67] in Ref. [68].

3.1 LMP

LMP layers take as inputs fully-connected graphs with nodes representing particles and the Minkowski distance between respective node 4-vectors as edge features. Each node $\mathcal{F}_i$ is defined by its features, all transforming under a corresponding irrep of the Lorentz group in the canonical basis [69], including at least one 4-vector (transforming under the $(1/2, 1/2)$ representation) representing its 4-momentum. As in Ref. [33], we denote the number of features in each node transforming under the (m, n) irrep as $\tau_{(m,n)}$, referred to as the multiplicity of the (m, n) representation.

[2] Another approach directly examines the latent space [14,15].

The $(t + 1)$-th MP layer operation consists of message-passing between each pair of nodes, with a message $m_{ij}^{(t)}$ to node i from node j (where $j \neq i$) and a self-interaction term m_{ii} defined as

$$m_{ij}^{(t)} = f\left(\left(p_{ij}^{(t)}\right)^2\right) p_{ij}^{(t)} \otimes \mathcal{F}_j^{(t)} \tag{2}$$

$$m_{ii}^{(t)} = \mathcal{F}_i^{(t)} \otimes \mathcal{F}_i^{(t)} \tag{3}$$

where $\mathcal{F}_i^{(t)}$ are the node features of node i before the $(t+1)$-th layer, $p_{ij} = p_i - p_j$ is the difference between node four-vectors, p_{ij}^2 is the squared Minkowski norm of p_{ij}, and f is a learnable, differentiable function acting on Lorentz scalars. A Clebsch–Gordan (CG) decomposition, which reduces the features to direct sums of irreps of $\mathrm{SO}^+(3, 1)$, is performed on both terms before concatenating them to produce the message m_i for node i:

$$m_i^{(t)} = \mathrm{CG}\left[m_{ii}^{(t)}\right] \oplus \mathrm{CG}\left[\sum_{j\neq i} m_{ij}^{(t)}\right], \tag{4}$$

where the summation over the destination node j ensures permutation symmetry because it treats all other nodes equally.

Finally, this aggregated message is used to update each node's features, such that

$$\mathcal{F}_i^{(t+1)} = W^{(t+1)}\left(\mathcal{F}_i^{(t)} \oplus m_i^{(t)}\right) \tag{5}$$

for all $i \in \{1, \ldots, N_{\text{particle}}\}$, where $W^{(t+1)}$ is a learnable node-wise operator which acts as separate fully-connected linear layers $W_{(m,n)}^{(t+1)}$ on the set of components living within each separate (m, n) representation space, outputting a chosen $\tau_{(m,n)}^{(t+1)}$ number of components per representation. In practice, we then truncate the irreps to a maximum dimension to make computations more tractable.

3.2 Encoder

The encoder takes as input an N-particle cloud, where each particle is each associated with a 4-momentum vector and an arbitrary number of scalars representing physical features such as mass, charge, and spin. Each isotypic component is initially transformed to a chosen multiplicity of $\left(\tau_{(m,n)}^{(0)}\right)_{\mathrm{E}}$ via a node-wise operator $W^{(0)}$ identical conceptually to $W^{(t+1)}$ in Eq. (5). The resultant graph is then processed through $N_{\mathrm{MP}}^{\mathrm{E}}$ LMP layers, specified by a sequence of multiplicities $\left\{\left(\tau_{(m,n)}^{(t)}\right)_{\mathrm{E}}\right\}_{t=1}^{N_{\mathrm{MP}}^{\mathrm{E}}}$, where $\left(\tau_{(m,n)}^{(t)}\right)_{\mathrm{E}}$ is the multiplicity of the (m, n) representation at the t-th layer. Weights are shared across the nodes in a layer to ensure permutation equivariance.

After the final MP layer, node features are aggregated to the latent space by a component-wise minimum (min), maximum (max), or mean. The min and max operations are performed on the respective Lorentz invariants. We also find, empirically, interesting performance by simply concatenating isotypic components across each particle and linearly "mixing" them via a learned matrix as in Eq. (5). Crucially, unlike in Eq. (5), where this operation only happens per particle, the concatenation across the particles imposes an ordering and, hence, breaks the permutation symmetry.

3.3 Decoder

The decoder recovers the N-particle cloud by acting on the latent space with N independent, learned linear operators, which again mix components living in the same representations. This cloud passes through $N_{\mathrm{MP}}^{\mathrm{D}}$ LMP layers, specified by a sequence of multiplicities $\left\{\left(\tau_{(m,n)}^{(t)}\right)_{\mathrm{D}}\right\}_{t=1}^{N_{\mathrm{MP}}^{\mathrm{D}}}$, where $\left(\tau_{(m,n)}^{(t)}\right)_{\mathrm{D}}$ is the multiplicity of the (m, n) representation at the t-th LMP layer. After the LMP layers, node features are mixed back to the input representation space $\left(D^{(0,0)}\right)^{\oplus\tau_{(0,0)}^{(0)}} \oplus D^{(1/2,1/2)}$ by applying a linear mixing layer and then truncating other isotypic components.

4 Experiments

We experiment with and evaluate the performance of the LGAE and baseline models on reconstruction and anomaly detection for simulated high-momentum jets. We describe the dataset in Sect. 4.1, the different models we consider in Sect. 4.2, the reconstruction and anomaly detection results in Sects. 4.3 and 4.4 respectively, an interpretation of the LGAE latent space in Sect. 4.5, and finally experiments of the data efficiency of the different models in Sect. 4.6.

4.1 Dataset

The model is trained to reconstruct 30-particle high transverse momentum jets from the JetNet [70] dataset, obtained using the associated library [71], zero-padding jets with fewer than 30, produced from gluons and light quarks. These are collectively referred to as quantum chromodynamics (QCD) jets.

Jets in JetNet are first produced at leading-order using MADGRAPH5_aMCATNLO [72] and decayed and showered with PYTHIA 8.2 [73]. They are then discretized and smeared to take detector spatial and energy resolution respectively into account, with simulated tracking inefficiencies – emulating the effects of the CMS and ATLAS trackers and calorimeters – and finally clustered using the anti-k_{T} [74] algorithm with distance parameter $R = 0.8$. Further details on the generation and reconstruction process are available

Table 1 Summary of the relevant symmetries respected by each model discussed in Sect. 4

Model	Aggregation	Name	Lorentz symmetry	Permutation symmetry	Translation symmetry
LGAE	Min-Max	LGAE-Min-Max	✓ (equivariance)	✓ (invariance)	✓ (equivariance)
	Mix	LGAE-Mix	✓ (equivariance)	×	✓ (equivariance)
GNNAE	Jet-level	GNNAE-JL	×	✓ (invariance)	✓ (equivariance)
	Particle-level	GNNAE-PL	×	✓ (equivariance)	✓ (equivariance)
CNNAE		CNNAE	×	×	✓ (equivariance)

in Ref. [20]. The exact smearing parameters and calorimeter granularities used are reported in Table 2 of Ref. [75] and correspond to the "CMS-like" scenario.

We represent the jets as a point cloud of particles, termed a "particle cloud", with the respective 3-momenta, in absolute coordinates, as particle features. In the processing step, each 3-momentum is converted to a 4-momentum: $p^\mu = (|\mathbf{p}|, \mathbf{p})$, where we consider the mass of each particle to be negligible. We use a 60%/20%/20% training/testing/validation splitting for the total 177,000 jets. For evaluating performance in anomaly detection, we consider jets from JetNet produced by top quarks, W bosons, and Z bosons as our anomalous signals.

Finally, we note here that the detector and reconstruction effects in JetNet, and indeed in real data collected at the LHC, break the Lorentz symmetry; hence, Lorentz equivariance is generally an *approximate* rather than an exact symmetry of HEP data. We assume henceforth that the magnitude of the symmetry breaking is small enough that imposing exact Lorentz equivariance in the LGAE is still advantageous – and the high performance of the LGAE and classification models such as LorentzNet support this assumption. Nevertheless, important studies in future work may include quantifying this symmetry breaking and considering approximate, as well as exact, symmetries in neural networks.

4.2 Models

LGAE model results are presented using both the min-max (LGAE-Min-Max) and "mix" (LGAE-Mix) aggregation schemes for the latent space, which consists of varying numbers of complex Lorentz vectors – corresponding to different compression rates. We compare the LGAE to baseline GNN and CNN autoencoder models, referred to as "GNNAE" and "CNNAE" respectively.

The GNNAE model is composed of fully-connected MPNNs adapted from Ref. [20]. We experiment with two types of encodings: (1) particle-level (GNNAE-PL), as in the PGAE [16] model, which compresses the features per node in the graph but retains the graph structure in the latent space, and (2) jet-level (GNNAE-JL), which averages the features across each node to form the latent space, as in the LGAE. Particle-level encodings produce better performance overall for the GNNAE, but the jet-level provides a more fair comparison with the LGAE, which uses jet-level encoding to achieve a high level of compression of the features.

For the CNNAE, which is adapted from Ref. [76], the relative coordinates of each input jets' particle constituents are first discretized into a 40 × 40 grid. The particles are then represented as pixels in an image, with intensities corresponding to p_T^{rel}. Multiple particles per jet may correspond to the same pixel, in which case their p_T^{rel}'s are summed. The CNNAE has neither Lorentz nor permutation symmetry, however, it does have in-built translation equivariance in $\eta - \phi$ space.

Hyperparameter and training details for all models can be found in Appendix A and Appendix B respectively, and a summary of the relevant symmetries respected by each model is provided in Table 1. The LGAE models are verified to be equivariant to Lorentz boosts and rotations up to numerical error, with details provided in Appendix C.

4.3 Reconstruction

We evaluate the performance of the LGAE, GNNAE, and CNNAE models, with the different aggregation schemes discussed, on the reconstruction of the particle and jet features of QCD jets. We consider relative transverse momentum $p_T^{\text{rel}} = p_T^{\text{particle}} / p_T^{\text{jet}}$ and relative angular coordinates $\eta^{\text{rel}} = \eta^{\text{particle}} - \eta^{\text{jet}}$ and $\phi^{\text{rel}} = \phi^{\text{particle}} - \phi^{\text{jet}} \pmod{2\pi}$ as each particle's features, and total jet mass, p_T and η as jet features. We define the compression rate as the ratio between the total dimension of the latent space and the number of features in the input space: 30 particles × 3 features per particle = 90.

Figure 2 shows random samples of jets, represented as discrete images in the angular-coordinate plane, reconstructed by the models with similar levels of compression in comparison to the true jets. Figure 3 shows histograms of the reconstructed features compared to the true distributions. The differences between the two distributions are quantified in Table 2 by calculating the median and interquartile ranges (IQR) of the relative errors between the reconstructed and true features. To calculate the relative errors of particle features for the permutation invariant LGAE and GNNAE models, particles are matched between the input and output clouds

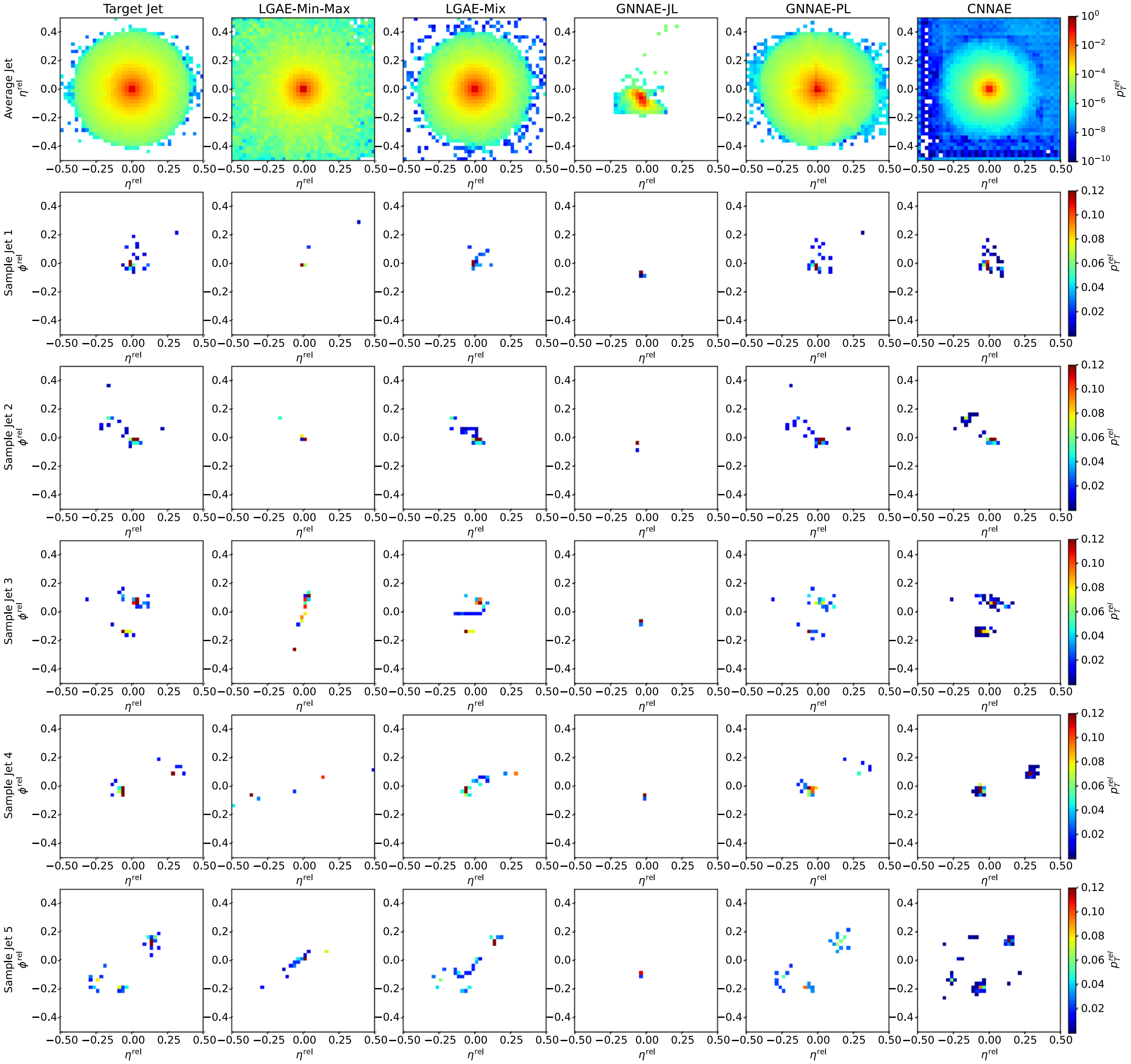


Fig. 2 Jet image reconstructions by LGAE-Min-Max ($\tau_{(1/2,1/2)} = 4$, 56.67% compression), LGAE-Mix ($\tau_{(1/2,1/2)} = 9$, 61.67% compression), GNNAE-JL ($\dim(L) = 55$, 61.11% compression), GNNAE-PL ($\dim(L) = 2 \times 30$, 66.67% compression), and CNNAE ($\dim(L) = 55$, 61.11% compression)

using the Jonker-Volgenant algorithm [77,78] based on the L2 distance between particle features. Due to the discretization of the inputs to the CNNAE, reconstructing individual particle features is not possible; instead, only jet features are shown.[3]

We can observe visually in Fig. 2 that out of the two permutation invariant models, while neither is able to reconstruct the jet substructure perfectly, the LGAE-Min-Max outperforms the GNNAE-JL. Perhaps surprisingly, the permutation-symmetry-breaking mix aggregation scheme improves the LGAE in this regard. Both visually in Fig. 3 and quantitatively from Tables 2 and 3, we conclude that the LGAE-Mix has the best performance overall, significantly outperforming the GNNAE and CNNAE models at similar compression rates. The LGAE-Min-Max model outperforms the GNNAE-JL in reconstructing all features and the GNNAE-PL in all but the IQR of the particle angular coordinates.

[3] These are calculated by summing each pixel's momentum "4-vector" – using the center of the pixel as angular coordinates and intensity as the $p_{\mathrm{T}}^{\mathrm{rel}}$.

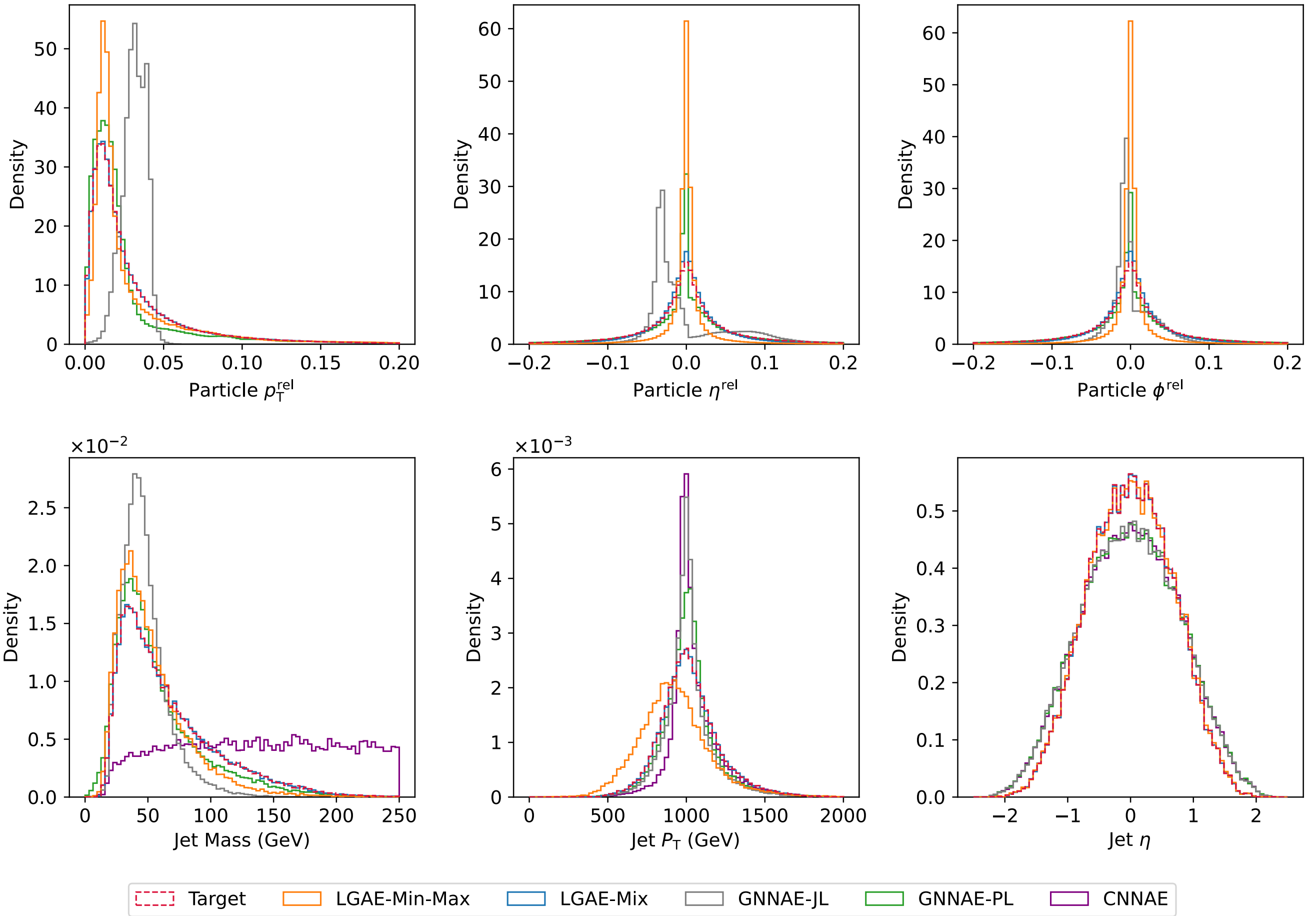


Fig. 3 Top: particle momenta ($p_T^{rel}, \eta^{rel}, \phi^{rel}$) reconstruction by LGAE-Min-Max ($\tau_{(1/2,1/2)} = 4$, resulting in 56.67% compression) and and LGAE-Mix ($\tau_{(1/2,1/2)} = 9$, resulting in 61.67% compression), and GNNAE-JL ($\dim(L) = 55$, resulting in 61.11% compression) and GNNAE-PL ($\dim(L) = 2 \times 30$, resulting in 66.67% compression). The reconstructions by the CNNAE are not included due to the discrete values of η^{rel} and ϕ^{rel}, as discussed in the text. Bottom: jet feature (M, p_T, η) reconstruction by the four models. For the jet feature reconstruction by the GNNAEs, the particle features in relative coordinates were transformed back to absolute coordinates before plotting. The jet ϕ is not shown because it follows a uniform distribution in $(-\pi, \pi]$ and is reconstructed well

Table 2 Median and IQR of relative errors in particle feature reconstruction of selected LGAE and GNNAE models. In each column, the best-performing latent space per model is italicized, and the best model overall is highlighted in bold

Model	Aggregation	Latent space	Particle p_T^{rel}		Particle η^{rel}		Particle ϕ^{rel}	
			Median	IQR	Median	IQR	Median	IQR
LGAE	Min-max	$\tau_{(1/2,1/2)} = 4$ (56.67%)	*0.006*	*0.562*	*0.002*	1.8	0.003	1.8
		$\tau_{(1/2,1/2)} = 7$ (96.67%)	0.002	0.640	−0.627	*1.7*	$< \mathbf{10^{-3}}$	*1.7*
	Mix	$\tau_{(1/2,1/2)} = 9$ (61.67%)	$< \mathbf{10^{-3}}$	0.011	$< \mathbf{10^{-3}}$	0.452	$< \mathbf{10^{-3}}$	0.451
		$\tau_{(1/2,1/2)} = 13$ (88.33%)	$< \mathbf{10^{-3}}$	**0.001**	$< \mathbf{10^{-3}}$	**0.022**	$< \mathbf{10^{-3}}$	**0.022**
GNNAE	Jet-level	$\dim(L) = 45$ (50.00%)	−0.983	3.8	*0.363*	*3.1*	*0.146*	*2.1*
		$\dim(L) = 90$ (100.00%)	*−0.627*	*3.5*	4.4	14.7	*0.146*	2.6
	Particle-level	$\dim(L) = 2 \times 30$ (66.67%)	−0.053	0.906	*0.009*	0.191	0.013	*0.139*
		$\dim(L) = 3 \times 30$ (100.00%)	*−0.040*	*0.892*	−0.037	*0.177*	*0.005*	0.243

Table 3 Median and IQR of relative errors in jet feature reconstruction by selected LGAE and GNNAE models, along with the CNNAE model. In each column, the best performing latent space per model is italicised, and the best model overall is highlighted in bold

Model	Aggregation	Latent space	Jet mass		Jet p_{T}		Jet η		Jet ϕ	
			Median	IQR	Median	IQR	Median	IQR	Median	IQR
LGAE	Min-max	$\tau_{(1/2,1/2)} = 4$ (56.67%)	*0.096*	*0.134*	*0.097*	*0.109*	$< \mathbf{10^{-3}}$	*0.004*	$< \mathbf{10^{-3}}$	*0.002*
		$\tau_{(1/2,1/2)} = 7$ (96.67%)	−0.139	0.287	−0.221	0.609	$< \mathbf{10^{-3}}$	0.021	$< \mathbf{10^{-3}}$	0.007
	Mix	$\tau_{(1/2,1/2)} = 9$ (61.67%)	$< \mathbf{10^{-3}}$	**0.003**	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$
		$\tau_{(1/2,1/2)} = 13$ (88.33%)	$< \mathbf{10^{-3}}$	**0.003**	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$	$< \mathbf{10^{-3}}$
GNNAE	Jet-level	$\dim(L) = 45$ (50.00%)	0.326	*0.667*	*0.030*	*0.088*	*0.005*	*0.040*	*0.001*	0.021
		$\dim(L) = 90$ (100.00%)	3.7	2.6	*0.030*	0.089	0.292	0.433	0.006	0.021
	Particle-level	$\dim(L) = 2 \times 30$ (66.67%)	*0.277*	0.299	*0.037*	0.110	0.002	*0.010*	−0.001	0.005
		$\dim(L) = 3 \times 30$ (100.00%)	0.339	*0.244*	0.050	*0.094*	*−0.001*	0.011	$< \mathbf{10^{-3}}$	0.005
CNNAE	linear layer	$\dim(L) = 55$ (61.67%)	−0.030	0.042	−0.021	0.017	$< \mathbf{10^{-3}}$	0.017	$< \mathbf{10^{-3}}$	0.003

4.4 Anomaly detection

We test the performance of all models as unsupervised anomaly detection algorithms by pre-training them solely on QCD and then using the reconstruction error for the QCD and new signal jets as the discriminating variable. We consider top quark, W boson, and Z boson jets as potential signals and QCD as the "background". We test the Chamfer distance, energy mover's distance [79] – the earth mover's distance applied to particle clouds, and MSE between input and output jets as reconstruction errors, and find the Chamfer distance most performant for all graph-based models. For the CNNAE, we use the MSE between the input and reconstructed image as the anomaly score.

Receiver operating characteristic (ROC) curves showing the signal efficiencies (ε_s) versus background efficiencies (ε_b) for individual and combined signals are shown in Fig. 4,[4] and ε_s values at particular background efficiencies are given in Table 4. We see that in general the permutation equivariant LGAE and GNNAE models outperform the CNNAE, strengthening the case for considering equivariance in neural networks. Furthermore, LGAE models have significantly higher signal efficiencies than GNNAEs and CNNAEs for all signals when rejecting $> 90\%$ of the background (which is the minimum level we typically require in HEP), and LGAE-Mix consistently performs better than LGAE-Min-Max.

4.5 Latent space interpretation

The outputs of the LGAE encoder are irreducible representations of the Lorentz groups; they consist of a pre-specified number of Lorentz scalars, vectors, and potentially higher-order representations. This implies a significantly more interpretable latent representation of the jets than traditional autoencoders, as the information distributed across the latent space is now disentangled between the different irreps of the Lorentz group. For example, scalar quantities like the jet mass will necessarily be encoded in the scalars of the latent space, and jet and particle 4-momenta in the vectors.

We demonstrate the latter empirically on the LGAE-Mix model ($\tau_{(1/2,1/2)} = 2$) by looking at correlations between jet 4-momenta and the components of different combinations of latent vector components. Figure 5 shows that, in fact, the jet momenta is encoded in the imaginary component of the sum of the latent vecotrs.

We can also attempt to understand the anomaly detection performance by looking at the encodings of the training data compared to the anomalous signal. Figure 6 shows the individual and total invariant mass of the latent vectors of sample LGAE models for QCD and top quark, W boson, and Z boson inputs. We observe that despite the overall similar kinematic properties of the different jet classes, the distributions for the QCD background are significantly different from the signals, indicating that the LGAE learns and encodes the difference in jet substructure – despite substructure observables such as jet mass not being direct inputs to the network – explaining the high performance in anomaly detection.

Finally, while in this section we showcased simple "brute-force" techniques for interpretability by looking directly at the distributions and correlations of latent features, we hypothesize that such an equivariant latent space would also lend itself effectively to the vast array of existing explainable AI algorithms [80,81], which generically evaluate the contribution of different input and intermediate neuron features to network outputs. We leave a detailed study of this to future work.

[4] Discontinuities in the top quark and combined signal LGAE-Min-Max ROCs indicate that at background efficiencies of $\lessapprox 5 \times 10^{-3}$, there are no signal events remaining in the validation dataset.

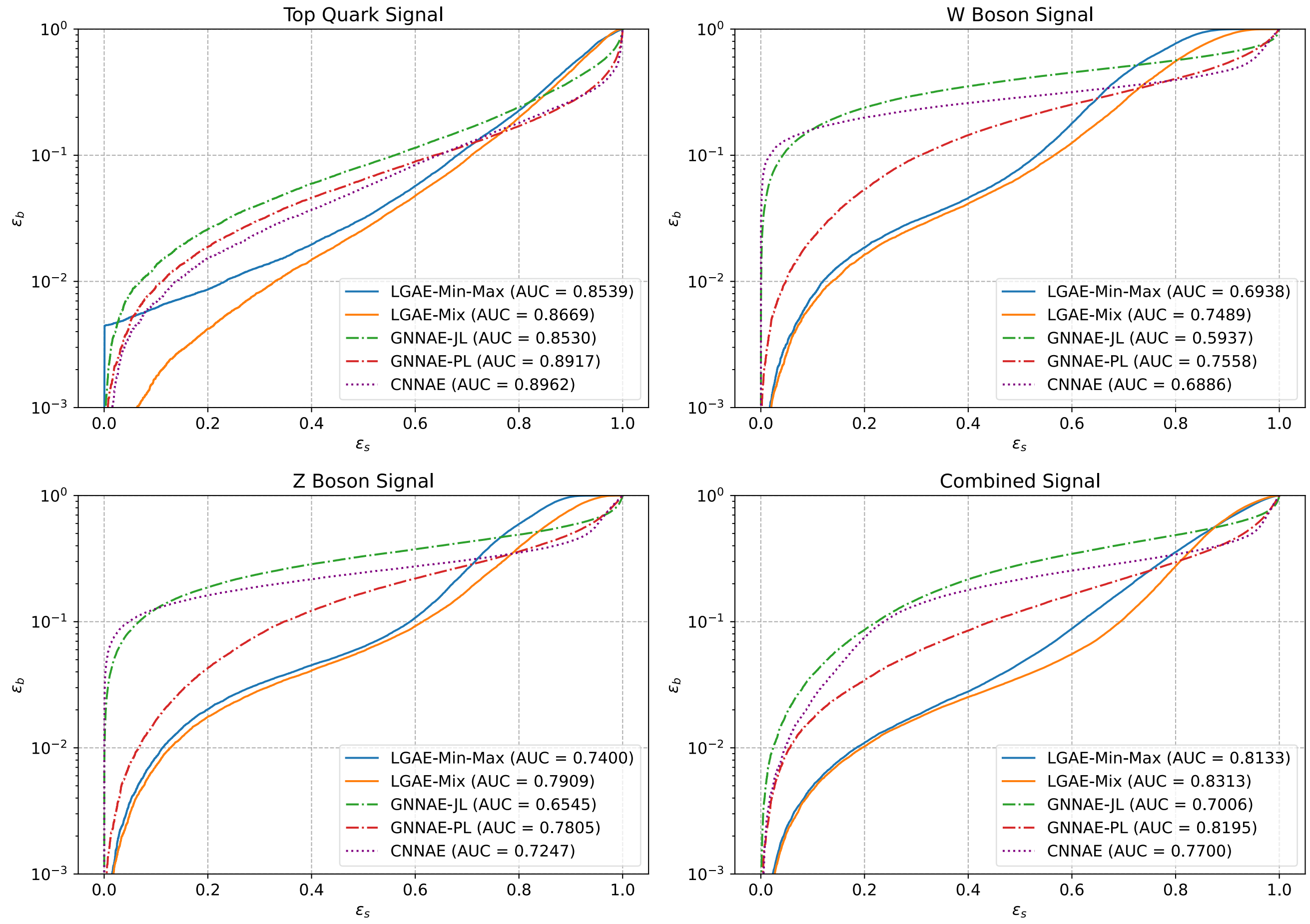


Fig. 4 Anomaly detections for the top quark signal (upper left), W boson signal (upper right), Z boson signal (lower left), and the combined signal (lower right) by the selected LGAE-Min-Max ($\tau_{(1/2,1/2)} = 7$), LGAE-Mix ($\tau_{(1/2,1/2)} = 2$), GNNAE-JL ($\dim(L) = 30$), GNNAE-PL ($\dim(L) = 2 \times 30$), and CNNAE ($\dim(L) = 55$) models

Table 4 Anomaly detection metrics by a selected LGAE and GNNAE models, along with the CNNAE model. In each column, the best performing latent space per model is italicized, and the best model overall is highlighted in bold

Model	Aggregation	Latent space	AUC	ε_s at given ε_b		
				$\varepsilon_s\,(10^{-1})$	$\varepsilon_s\,(10^{-2})$	$\varepsilon_s\,(10^{-3})$
LGAE	Min-Max	$\tau_{(1/2,1/2)} = 2$ (30.00%)	0.7253	0.5706	0.1130	*0.0011*
		$\tau_{(1/2,1/2)} = 4$ (56.67%)	0.7627	0.5832	*0.1305*	0.0007
		$\tau_{(1/2,1/2)} = 7$ (96.67%)	*0.7673*	*0.5932*	0.0820	0.0009
	Mix	$\tau_{(1/2,1/2)} = 2$ (15.00%)	*0.8023*	0.6178	**0.1662**	**0.0250**
		$\tau_{(1/2,1/2)} = 4$ (28.33%)	*0.8023*	0.6257	0.1592	0.0229
		$\tau_{(1/2,1/2)} = 7$ (48.33%)	0.7967	**0.6290**	0.1562	0.0225
GNNAE	JL	$\dim(L) = 10$ (11.11%)	0.5891	0.1576	0.0161	*0.0014*
		$\dim(L) = 40$ (44.44%)	0.6636	*0.2293*	*0.0262*	0.0013
		$\dim(L) = 80$ (88.89%)	*0.7006*	0.2240	0.0239	0.0010
	PL	$\dim(L) = 2 \times 30$ (66.67%)	**0.8195**	*0.4435*	0.0564	0.0042
		$\dim(L) = 3 \times 30$ (100.00%)	0.8095	0.4306	*0.0762*	*0.0044*
CNNAE	Linear layer	$\dim(L) = 55$ (61.67%)	0.7700	0.2473	0.0469	0.0053

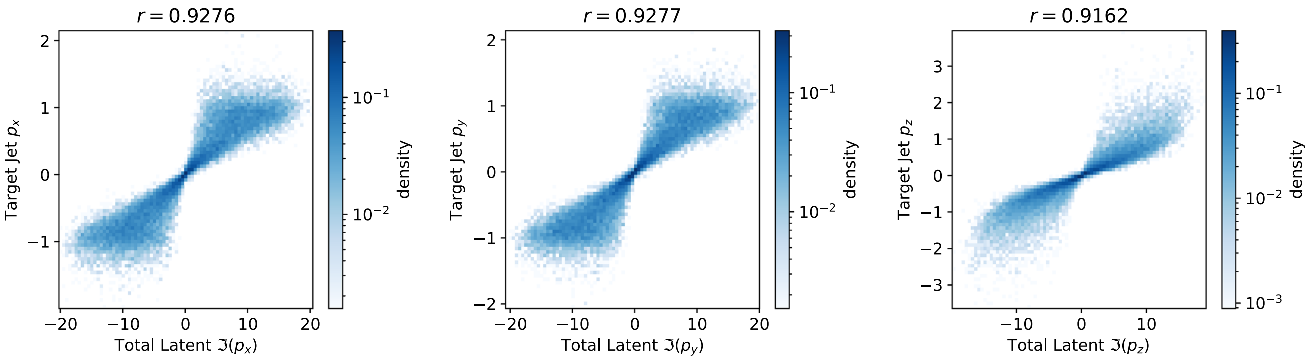


Fig. 5 The correlations between the total momentum of the imaginary components in the $\tau_{(1/2,1/2)} = 2$ LGAE-Mix model and the target jet momenta. The Pearson correlation coefficient r is listed above

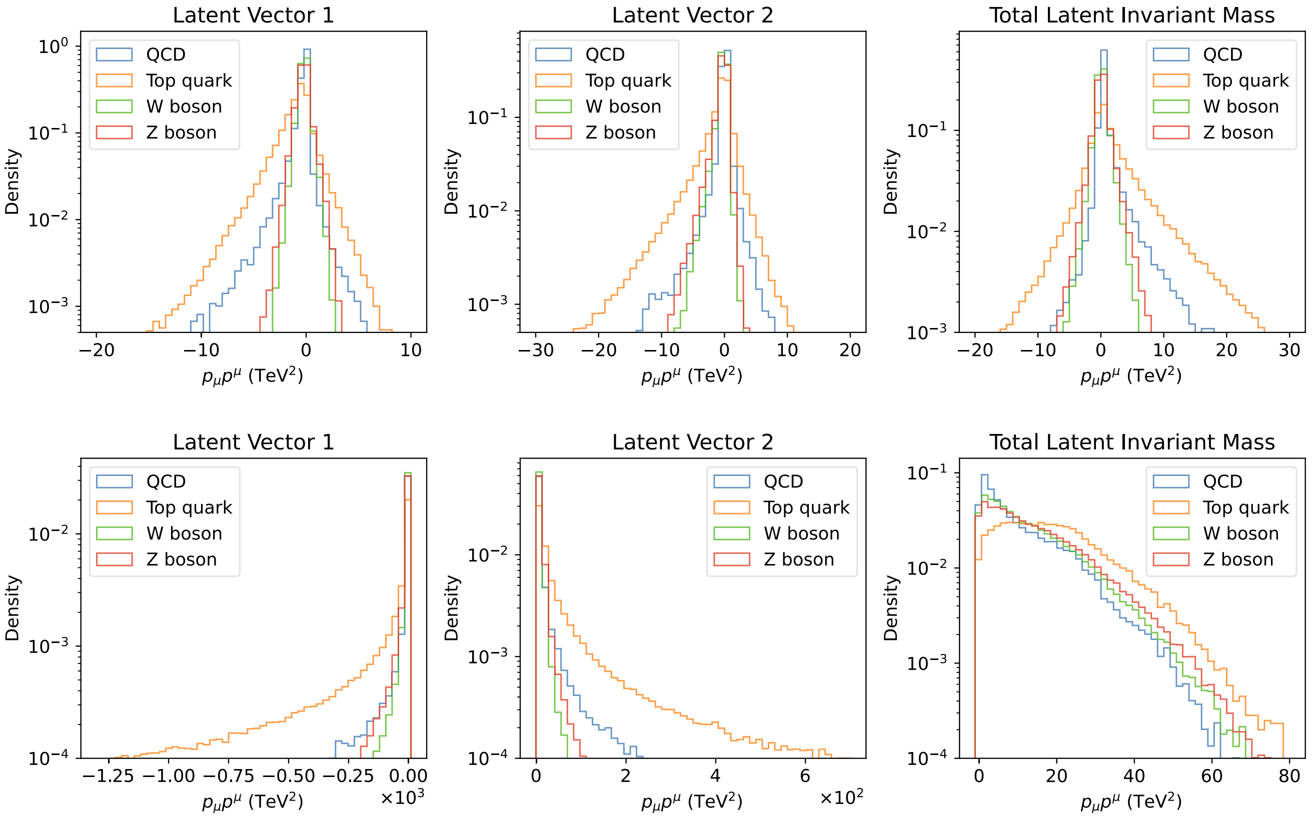


Fig. 6 Top: distributions of the invariant mass squared of the latent 4-vectors and jet momenta of the LGAE-Mix with $\tau_{(1/2,1/2)} = 2$ latent 4-vectors. Bottom: distributions of the invariant mass squared of two latent 4-vectors and jet momenta of the LGAE-Min-Max with $\tau_{(1/2,1/2)} = 2$ latent 4-vectors

4.6 Data efficiency

In principle, equivariant neural networks should require less training data for high performance, since critical biases of the data, which would otherwise have to be learnt by non-equivariant networks, are already built in. We test this claim by measuring the performances of the best-performing LGAE and CNNAE architectures from Sect. 4.3 trained on varying fractions of the training data.

The median magnitude of the relative errors between the reconstructed and true jet masses of the different models and fractions is shown in Fig. 7. Each model is trained five times per training fraction, with different random seeds, and evaluated on the same-sized validation dataset; the median of the five models is plotted. We observe that, in agreement with our hypothesis, the LGAE models both maintain their high performance all the way down to training on 1% of the data, while the CNNAE's performance steadily degrades down to 2% and then experiences a further sharp drop.

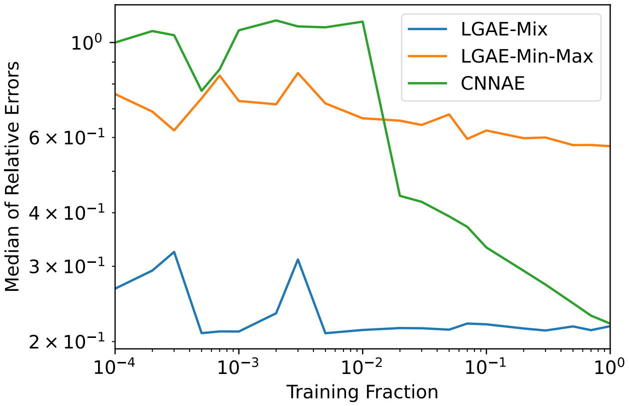


Fig. 7 Median magnitude of relative errors of jet mass reconstruction by LGAE and CNNAE models at trained on different fractions of the training data

5 Conclusion

We develop the Lorentz group autoencoder (LGAE), an autoencoder model equivariant to Lorentz transformations. We argue that incorporating this key inductive bias of high energy physics (HEP) data can have a significant impact on the performance, efficiency, and interpretability of machine learning models in HEP. We apply the LGAE to tasks of compression and reconstruction of input quantum chromodynamics (QCD) jets, and of identifying anomalous top quark, W boson, and Z boson jets. We report excellent performance in comparison to baseline graph and convolutional neural network autoencoder models, with the LGAE outperforming them on several key metrics. We also demonstrate the LGAE's interpretability, by analyzing the latent spaces of LGAE models for both tasks, and data efficiency relative to baseline models. The LGAE opens many promising avenues in terms of both performance and model interpretability, with the exploration of new datasets, the magnitude of Lorentz and permutation symmetry breaking due to detector effects, higher-order Lorentz group representations, and challenges with real-life compression and anomaly detection applications all exciting possibilities for future work.

Acknowledgements We would like to thank Dr. Rose Yu for discussions on equivariant neural networks, and Dr. Dylan Rankin for suggestions on the anomaly detection performance and latent space analysis of the LGAE. ZH thanks the UC San Diego Faculty Mentor Program for supporting this research. RK was partially supported by the LHC Physics Center at Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy (DOE). JD is supported by the DOE, Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187, the DOE, Office of Advanced Scientific Computing Research under Award No. DE-SC0021396 (FAIR4HEP), and the NSF HDR Institute for Accelerating AI Algorithms for Data Driven Discovery (A3D3) under Cooperative Agreement OAC-2117997. NC was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369). This work was performed using the Pacific Research Platform Nautilus HyperCluster supported by NSF awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100 Gpbs networks.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The datasets used in this manuscript are publicly available on [70] and the code for all models used in this paper can be found in a public repository [68].]

Appendix A: Model details

Appendix A.1: LGAE

For both encoder and decoder, we choose $N_{\mathrm{MP}}^{\mathrm{E}} = N_{\mathrm{MP}}^{\mathrm{D}} = 4$ LMP layers. The multiplicity per node in each LMP layer has been optimized to be

$$\left\{\left(\tau_{(m,n)}^{(t)}\right)^{\mathrm{E}}\right\}_{t=1}^{4} = (3, 3, 4, 4) \tag{A.1}$$

for the encoder and

$$\left\{\left(\tau_{(m,n)}^{(t)}\right)^{\mathrm{D}}\right\}_{t=1}^{4} = (4, 4, 3, 3) \tag{A.2}$$

for the decoder, the components in the vector on the right-hand side are the multiplicity in each of the four LMP layers per network, and the multiplicity per layer is the same for all representations. After each CG decomposition, we truncate irreps of dimensions higher than $(1/2, 1/2)$ for tractable computations, i.e., after each LMP operation we are left with only scalar and vector representations per node. Empirically, we did not find such a truncation to affect the performance of the model. This means that the LMP layers in the LGAE are similar in practice to those of LorentzNet, which uses only scalar and vector representations throughout, but are more general as higher dimensional representations are involved in the intermediate steps before truncation.

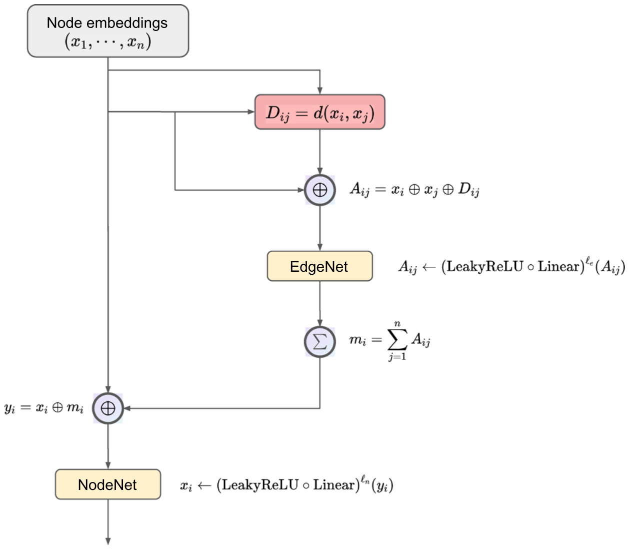

Node embeddings $(x_1, \cdots, x_n)$
$D_{ij} = d(x_i, x_j)$
$A_{ij} = x_i \oplus x_j \oplus D_{ij}$
EdgeNet
$A_{ij} \leftarrow (\text{LeakyReLU} \circ \text{Linear})^{\ell_e}(A_{ij})$
$m_i = \sum_{j=1}^{n} A_{ij}$
$y_i = x_i \oplus m_i$
NodeNet
$x_i \leftarrow (\text{LeakyReLU} \circ \text{Linear})^{\ell_n}(y_i)$

Fig. 8 An MPNN layer in the GNNAE. Here, EdgeNet and NodeNet are feed-forward neural networks

The differentiable mapping $f(d_{ij})$ in Eq.(2) is chosen to be the Lorentzian bell function as in Ref. [33]. For all models, the latent space contains only $\tau_{(0,0)} = 1$ complex Lorentz scalar, as we found increasing the number of scalars beyond one did not improve the performance in either reconstruction or anomaly detection. Empirically, the reconstruction performance increased with more latent vectors, as one might expect, while anomaly detection performance generally worsened from adding more than two latent vectors.

5.1 GNNAE

The GNNAE is constructed from fully-connected MPNNs. The update rule in the $(t+1)$-th MPNN layer is based on Ref. [20], and given by

$$m_i^{(t)} = \sum_{j=1}^{n} f_e^{(t)}\left(x_i^{(t)} \oplus x_j^{(t)} \oplus d\left(x_i^{(t)}, x_j^{(t)}\right)\right), \tag{A.3}$$

$$x_i^{(t+1)} = f_n^{(t)}\left(x_i^{(t)} \oplus m_i^{(t)}\right), \tag{A.4}$$

where $x_i^{(t)}$ is the node embedding of node i at t-th iteration, d is any distance function (Euclidean norm in our case), $m_i^{(t)}$ is the message for updating node embedding in node i, $f_e^{(t+1)}$ and $f_n^{(t+1)}$ are any learnable mapping at the current MP layer. A diagram for an MPNN layer is shown in Fig. 8. The overall architecture is similar to that in Fig. 1, with the LMP replaced by the MPNN. The code for the GNNAE model can be found in the Ref. [82].

For both the encoder and decoder, there are 3 MPNN layers. The learnable functions in each layer are optimized to be

$$\begin{aligned} f_n^{(1)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{30\to 15}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{60\to 30}) \\ f_e^{(1)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{40\to 30}), \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{50\to 40}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{61\to 50}), \end{aligned} \tag{A.5}$$

$$\begin{aligned} f_n^{(2)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{15\to 8}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{45\to 15}) \\ f_e^{(2)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{31\to 30}), \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{30\to 30}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{30\to 30}), \end{aligned} \tag{A.6}$$

$$\begin{aligned} f_n^{(3)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{8\to \delta}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{38\to 8}) \\ f_e^{(3)} &= (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{20\to 30}), \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{16\to 20}) \\ &\quad \circ (\text{LeakyReLU}_{0.2} \circ \text{Linear}_{17\to 16}), \end{aligned} \tag{A.7}$$

where $\text{LeakyRelu}_{0.2}(x) = \max(0.2x, x)$ is the LeakyReLu function.

Depending on the aggregation layer, the value of δ in $f_n^{(3)}$ and the final aggregation layer is different. For GNNAE-JL encoders, $\delta = N \times \dim(L)$, where L is the latent space, and N is the number of nodes in the graph. Then, mean aggregation is done across the graph. For GNNAE-PL encoders, $\delta = d$, where d is the node dimension in the latent space. In the GNNAE-JL decoder, the input layer is a linear layer that recovers the particle cloud structure similar to that in the LGAE.

Appendix A.3: CNNAE

The encoder is composed of two convolutional layers with kernel size (3, 3), stride size (2, 2), "same" padding, and 128 output channels, each followed by a ReLU activation function. The aggregation layer into the latent space is a fully-connected linear layer. The decoder is composed of transposed convolution layers (also known as deconvolutional layers) with the same settings as the encoder. A softmax function is applied at the end so that the sum of all pixel values in an image is 1, as a property of the jet image representation. A 55-dimensional latent space is chosen so that the compression rate is $55/90 \approx 60\%$ for even comparisons with the LGAE and GNNAE models.

Appendix B: Training details

We use the Chamfer loss function [83–85] for the LGAE-Min-Max and GNNAE-JL models, and MSE for LGAE-Mix and GNNAE-PL. We tested the Hungarian loss [78,86] and differentiable energy mover's distance (EMD) [79], calculated using the JetNet library [71], as well but found the Chamfer and MSE losses more performant.

The graph-based models are optimized using the Adam optimizer [87] implemented in PyTorch [67] with a learning rate $\gamma = 10^{-3}$, coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay $\lambda = 0$. The CNNAE is optimized using the same optimizer implemented in TensorFlow [88]. They are all trained on single NVIDIA RTX 2080 Ti GPUs each for a maximum of 20,000 epochs using early stopping with the patience of 200 epochs. The total training time for LGAE models is typically 35 h, and at most 100 h, while GNNAE-PL and GNNAE-JL train for 50 and 120 h on average, respectively. By contrast, the CNNAE model, due to its simplicity, can typically converge within 3 h.

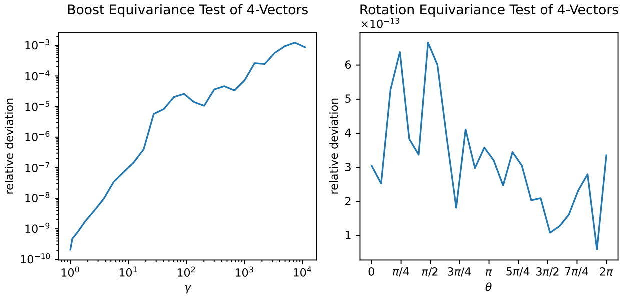
Appendix C: Equivariance tests

We test the covariance of the LGAE models to Lorentz transformations and find they are indeed equivariant up to numerical errors. Bogatskiy et al. point out that equivariance to boosts in particular is sensitive to numerical precision [33], so we use double precision (64-bit) throughout the model. In addition, we scale down the data by a factor of 1000 (i.e. working in the units of PeV) for better numerical precision at high boosts.

For a given transformation $\Lambda \in \mathrm{SO}^+(3,1)$ we compare $\Lambda \cdot \mathrm{LGAE}(p)$ and $\mathrm{LGAE}(\Lambda \cdot p)$ are compared, where p is the particle-level 4-momentum. The relative deviation is defined as

$$\delta_p(\Lambda) = \left| \frac{\mathrm{mean}(\mathrm{LGAE}(\Lambda \cdot p)) - \mathrm{mean}(\Lambda \cdot \mathrm{LGAE}(p))}{\mathrm{mean}(\Lambda \cdot \mathrm{LGAE}(p))} \right| \tag{A.8}$$

Figure 9 shows the mean relative deviation, averaged over each particle in each jet, over 3000 jets from our test dataset from boosts along and rotations around the z-axis. We find the relative deviation from boosts to be within $\mathcal{O}\left(10^{-3}\right)$ in the interval $\gamma \in [0, \cosh(10)]$ (equivalent to $\beta \in [0, 1 - 4 \times 10^{-9}]$) and from rotations to be $< 10^{12}$.



Boost Equivariance Test of 4-Vectors

Rotation Equivariance Test of 4-Vectors

Fig. 9 The relative deviations, as defined in Eq. (A.8), of the output 4-momenta p^{μ} to boosts along the z-axis (left) and rotations around the z-axis (right)

References

1. A.J. Larkoski, I. Moult, B. Nachman, Phys. Rep. **841**, 1 (2020). https://doi.org/10.1016/j.physrep.2019.11.001. arXiv:1709.04464
2. P. Baldi, P. Sadowski, D. Whiteson, Nat. Commun. **5**, 4308 (2014). https://doi.org/10.1038/ncomms5308. arXiv:1402.4735
3. P. Baldi, P. Sadowski, D. Whiteson, Phys. Rev. Lett. **114**, 111801 (2015). https://doi.org/10.1103/PhysRevLett.114.111801. arXiv:1410.3469
4. J. Pearkes, W. Fedorko, A. Lister, C. Gay. arXiv:1704.02124 (2017)
5. H. Qu, L. Gouskos, Phys. Rev. D **101**, 056019 (2020). https://doi.org/10.1103/PhysRevD.101.056019. arXiv:1902.08570
6. F. Bury, C. Delaere, JHEP **04**, 020 (2021). https://doi.org/10.1007/JHEP04(2021)020. arXiv:2008.10949
7. D. Belayneh et al., Eur. Phys. J. C **80**, 688 (2020). https://doi.org/10.1140/epjc/s10052-020-8251-9. arXiv:1912.06794
8. J. Duarte, J.-R. Vlimant, in *Artificial Intelligence for Particle Physics* (World Scientific Publishing, 2020). Submitted to Int. J. Mod. Phys. A. https://doi.org/10.1142/12200. arXiv:2012.01249
9. S. Farrell et al., in *4th International Workshop Connecting The Dots 2018* (2018). arXiv:1810.06111
10. G. DeZoort, S. Thais, J. Duarte, V. Razavimaleki, M. Atkinson, I. Ojalvo, M. Neubauer, P. Elmer, Comput. Softw. Big Sci. **5**, 26 (2021). https://doi.org/10.1007/s41781-021-00073-z. arXiv:2103.16701
11. O. Atkinson, A. Bhardwaj, C. Englert, V.S. Ngairangbam, M. Spannowsky, JHEP **08**, 080 (2021). https://doi.org/10.1007/JHEP08(2021)080. arXiv:2105.07988
12. T. Heimel, G. Kasieczka, T. Plehn, J.M. Thompson, SciPost Phys. **6**, 030 (2019). https://doi.org/10.21468/SciPostPhys.6.3.030. arXiv:1808.08979
13. F. Canelli, A. de Cosa, L.L. Pottier, J. Niedziela, K. Pedro, M. Pierini, JHEP **02**, 074 (2022). https://doi.org/10.1007/JHEP02(2022)074. arXiv:2112.02864
14. T. Cheng, J.-F. Arguin, J. Leissner-Martin, J. Pilette, T. Golling, Variational autoencoders for anomalous jet tagging. Accepted by Phys. Rev. D (2020). arXiv:2007.01850
15. B. Bortolato, A. Smolkovič, B.M. Dillon, J.F. Kamenik, Phys. Rev. D **105**, 115009 (2022). https://doi.org/10.1103/PhysRevD.105.115009. arXiv:2103.06595
16. S. Tsan, R. Kansal, A. Aportela, D. Diaz, J. Duarte, S. Krishna, F. Mokhtar, J.-R. Vlimant, M. Pierini, in *4th Machine Learning and the Physical Sciences Workshop at the 35th Conference on Neural Information Processing Systems* (2021). https://ml4physicalsciences.github.io/2021/files/NeurIPS_ML4PS_2021_98.pdf. arXiv:2111.12849
17. S.E. Park, D. Rankin, S.-M. Udrescu, M. Yunus, P. Harris, JHEP **21**, 030 (2020). https://doi.org/10.1007/JHEP06(2021)030. arXiv:2011.03550
18. L. de Oliveira, M. Paganini, B. Nachman, Comput. Softw. Big Sci. **1**, 4 (2017). https://doi.org/10.1007/s41781-017-0004-6. arXiv:1701.05927
19. M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. Lett. **120**, 042003 (2018). https://doi.org/10.1103/PhysRevLett.120.042003. arXiv:1705.02355
20. R. Kansal, J. Duarte, B. Orzari, T. Tomei, M. Pierini, M. Touranakou, J.-R. Vlimant, D. Gunopoulos, in *3rd Machine Learning and the Physical Sciences Workshop at the 34th Conference on Neural Information Processing*

Systems (2020). https://ml4physicalsciences.github.io/2020/files/NeurIPS_ML4PS_2020_104.pdf. arXiv:2012.00173
21. K. Dohi, Variational autoencoders for jet simulation (2020). arXiv:2009.04842
22. M. Touranakou, N. Chernyavskaya, J. Duarte, D. Gunopulos, R. Kansal, B. Orzari, M. Pierini, T. Tomei, J.-R. Vlimant, Mach. Learn. Sci. Technol. **3**, 035003 (2022). https://doi.org/10.1088/2632-2153/ac7c56. arXiv:2203.00520
23. M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. D **97**, 014021 (2018). https://doi.org/10.1103/PhysRevD.97.014021. arXiv:1712.10321
24. D. Guest, K. Cranmer, D. Whiteson, Ann. Rev. Nucl. Part. Sci. **68**, 161 (2018). https://doi.org/10.1146/annurev-nucl-101917-021019. arXiv:1806.11484
25. A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, Nature **560**, 41 (2018). https://doi.org/10.1038/s41586-018-0361-2
26. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Rev. Mod. Phys. **91**, 045002 (2019). https://doi.org/10.1103/RevModPhys.91.045002. arXiv:1903.10563
27. HEP ML Community, A living review of machine learning for particle physics (2021). https://iml-wg.github.io/HEPML-LivingReview/. arXiv:2102.02770
28. P.T. Komiske, E.M. Metodiev, J. Thaler, JHEP **01**, 121 (2019). https://doi.org/10.1007/JHEP01(2019)121. arXiv:1810.05165
29. P. Konar, V.S. Ngairangbam, M. Spannowsky, JHEP **02**, 060 (2022). https://doi.org/10.1007/JHEP02(2022)060. arXiv:2109.14636
30. O. Atkinson, A. Bhardwaj, C. Englert, P. Konar, V.S. Ngairangbam, M. Spannowsky, Front. AI **5**, 943135 (2022). https://doi.org/10.3389/frai.2022.943135. arXiv:2204.12231
31. J. Shlomi, P. Battaglia, J.-R. Vlimant, https://doi.org/10.1088/2632-2153/abbf9a. arXiv:2007.13681 (2020)
32. S. Thais, P. Calafiura, G. Chachamis, G. DeZoort, J. Duarte, S. Ganguly, M. Kagan, D. Murnane, M.S. Neubauer, K. Terao, in *2022 Snowmass Summer Study* (2022). arXiv:2203.12852
33. A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller, R. Kondor, *Proceedings of the 37th International Conference on Machine Learning*. Proceedings of Machine Learning Research, vol. 119, p. 992 (2020). https://proceedings.mlr.press/v119/bogatskiy20a.html. arXiv:2006.04780
34. S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian, W. Du, Z.-M. Ma, T.-Y. Liu, JHEP **07**, 030 (2022). https://doi.org/10.1007/JHEP07(2022)030. arXiv:2201.08187
35. C. Li, H. Qu, S. Qian, Q. Meng, S. Gong, J. Zhang, T.-Y. Liu, Q. Li, Does Lorentz-symmetric design boost network performance in jet physics? (2022). arXiv:2208.07814
36. A. Butter, G. Kasieczka, T. Plehn, M. Russell, SciPost Phys. **5**, 028 (2018). https://doi.org/10.21468/SciPostPhys.5.3.028. arXiv:1707.08966
37. J.H. Collins, in *ICLR workshop Deep Generative Models for Highly Structured Data* (2021). arXiv:2109.10919
38. J.W. Monk, JHEP **12**, 021 (2018). https://doi.org/10.1007/JHEP12(2018)021. arXiv:1807.03685
39. M. Farina, Y. Nakai, D. Shih, Phys. Rev. D **101**, 075021 (2020). https://doi.org/10.1103/PhysRevD.101.075021. arXiv:1808.08992
40. T. Finke, M. Krämer, A. Morandini, A. Mück, I. Oleksiyuk, JHEP **06**, 161 (2021). https://doi.org/10.1007/JHEP06. arXiv:2104.09051
41. M.M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: grids, groups, graphs, geodesics, and gauges (2021). arXiv:2104.13478
42. R. Walters, J. Li, R. Yu, in *International Conference on Learning Representations* (2021). https://openreview.net/forum?id=J8_GttYLFgr. arXiv:2010.11344
43. M. Weiler, G. Cesa, in *Advances in Neural Information Processing Systems*, vol. 32, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Curran Associates, Inc., 2019). https://proceedings.neurips.cc/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf
44. C. Esteves, C. Allen-Blanchette, A. Makadia, K. Daniilidis, Int. J. Comput. Vis. **128**, 588 (2020). https://doi.org/10.1007/s11263-019-01220-1
45. N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds (2018). arXiv:1802.08219
46. S. Batzner, A. Musaelian, L. Sun, M. Geiger, J.P. Mailoa, M. Kornbluth, N. Molinari, T.E. Smidt, B. Kozinsky, Nat. Commun. **13**, 2453 (2022). https://doi.org/10.1038/s41467-022-29939-5
47. M. Finzi, M. Welling, A.G. Wilson, in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, vol. 139, ed. by M. Meila, T. Zhang (PMLR, 2021), p. 3318. https://proceedings.mlr.press/v139/finzi21a.html. arXiv:2104.09459
48. T. Cohen, M. Geiger, M. Weiler. https://doi.org/10.48550/ARXIV.1811.02017 (2018)
49. M. Finzi, S. Stanton, P. Izmailov, A.G. Wilson, Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data (2020). https://doi.org/10.48550/ARXIV.2002.12880
50. T. Cohen, M. Welling, in *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, vol. 48, ed. by M.F. Balcan, K.Q. Weinberger (PMLR, New York, 2016), p. 2990. http://proceedings.mlr.press/v48/cohenc16.html. arXiv:1602.07576
51. M. Geiger, T. Smidt, e3nn: Euclidean neural networks (2022). https://doi.org/10.48550/ARXIV.2207.09453
52. R. Kondor, Z. Lin, S. Trivedi, Clebsch–Gordan Nets: a fully Fourier space spherical convolutional neural network (2018). https://doi.org/10.48550/ARXIV.1806.09231
53. B. Anderson, T.-S. Hy, R. Kondor, Cormorant: covariant molecular neural networks (2019). https://doi.org/10.48550/ARXIV.1906.04015
54. G.E. Hinton, R.R. Salakhutdinov, Science **313**, 504 (2006). https://doi.org/10.1126/science.1127647
55. G. Di Guglielmo et al., IEEE Trans. Nucl. Sci. **68**, 2179 (2021). arXiv:2105.01683 10.1109/TNS.2021.3087100
56. O. Cerri, T.Q. Nguyen, M. Pierini, M. Spiropulu, J.-R. Vlimant, JHEP **05**, 036 (2019). https://doi.org/10.1007/JHEP05(2019)036. arXiv:1811.10276
57. G. Kasieczka et al., Rep. Prog. Phys. **84**, 124201 (2021). https://doi.org/10.1088/1361-6633/ac36b9. arXiv:2101.08320
58. E. Govorkova et al., Nat. Mach. Intell. **4**, 154 (2022). https://doi.org/10.1038/s42256-022-00441-3. arXiv:2108.03986
59. A.A. Pol, V. Berger, G. Cerminara, C. Germain, M. Pierini, in *18th International Conference on Machine Learning and Applications* (2020). arXiv:2010.05531
60. V.S. Ngairangbam, M. Spannowsky, M. Takeuchi, Phys. Rev. D **105**, 095004 (2022). https://doi.org/10.1103/PhysRevD.105.095004. arXiv:2112.04958
61. B.M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson, L. Vogel, SciPost Phys. **12**, 188 (2022). https://doi.org/10.21468/SciPostPhys.12.6.188. arXiv:2108.04253
62. D. Bank, N. Koenigstein, R. Giryes, Autoencoders (2020). arXiv:2003.05991

63. M. Tschannen, O. Bachem, M. Lucic, Recent advances in autoencoder-based representation learning (2018). arXiv:1812.05069
64. D.P. Kingma, M. Welling, in *2nd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, ed. by Y. Bengio, Y. LeCun (2014). arXiv: 1312.6114
65. O. Atkinson, A. Bhardwaj, C. Englert, V.S. Ngairangbam, M. Spannowsky, JHEP **08**, 080 (2021). https://doi.org/10.1007/JHEP08. arXiv:2105.07988
66. J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, vol. 70, ed. by D. Precup, Y.W. Teh (PMLR, 2017), p. 1263. http://proceedings.mlr.press/v70/gilmer17a.html. arXiv:1704.01212
67. A. Paszke et al., in *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc, 2019), p. 8024. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. arXiv:1912.01703
68. Z. Hao, R. Kansal, J. Duarte, N. Chernyavskaya. https://doi.org/10.5281/zenodo.7434838 (2022)
69. G.C.I.M. Gelfand, R.A. Minlos, *Representations of the Rotation and Lorentz Groups and Their Applications* (Pergamon Press, Oxford, 1963)
70. R. Kansal, J. Duarte, H. Su, B. Orzari, T. Tomei, M. Pierini, M. Touranakou, J.-R. Vlimant, D. Gunopulos, Jetnet (2022). https://doi.org/10.5281/zenodo.6975118
71. R. Kansal, C. Pareja, J. Duarte, jet-net/JetNet: v0.2.1.post2 (2022). https://doi.org/10.5281/zenodo.7067466
72. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, JHEP **07**, 079 (2014). https://doi.org/10.1007/JHEP07(2014)079. arXiv:1405.0301
73. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, Comput. Phys. Commun. **191**, 159 (2015). https://doi.org/10.1016/j.cpc.2015.01.024. arXiv:1410.3012
74. M. Cacciari, G.P. Salam, G. Soyez, JHEP **04**, 063 (2008). https://doi.org/10.1088/1126-6708/2008/04/063. arXiv:0802.1189
75. E. Coleman, M. Freytsis, A. Hinzmann, M. Narain, J. Thaler, N. Tran, C. Vernieri, JINST **13**, T01003 (2018). https://doi.org/10.1088/1748-0221/13/01/T01003. arXiv:1709.08705 [hep-ph]
76. M. Farina, Y. Nakai, D. Shih, Phys. Rev. D (2020). https://doi.org/10.1103/PhysRevD.101.075021
77. R. Jonker, A. Volgenant, Computing **38**, 325 (1987). https://doi.org/10.1007/BF02278710
78. P. Virtanen et al., Nat. Methods **17**, 261 (2020). https://doi.org/10.1038/s41592-019-0686-2
79. P.T. Komiske, E.M. Metodiev, J. Thaler, Phys. Rev. Lett. **123**, 041801 (2019). https://doi.org/10.1103/PhysRevLett.123.041801. arXiv:1902.02346
80. G. Vilone, L. Longo, Explainable artificial intelligence: a systematic review (2020). arXiv:2006.00093
81. D. Minh, H.X. Wang, Y.F. Li, T.N. Nguyen, Artif. Intell. Rev. **55**, 3503–3568 (2022). https://doi.org/10.1007/s10462-021-10088-y
82. Z. Hao, R. Kansal, J. Duarte, N. Chernyavskaya (2022). https://doi.org/10.5281/zenodo.7453769
83. H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, H.C. Wolf, in *Proceedings of the 5th International Joint Conference on Artificial Intelligence (KJCAI)*, vol. 2 (Morgan Kaufmann Publishers Inc, San Francisco, 1977), p. 659. https://www.ijcai.org/Proceedings/77-2/Papers/024.pdf
84. H. Fan, H. Su, L.J. Guibas, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), p. 2463. https://doi.org/10.1109/CVPR.2017.264. arXiv:1612.00603
85. Y. Zhang, J. Hare, A. Prügel-Bennett, in *8th International Conference on Learning Representations* (2020). https://openreview.net/forum?id=HJgBA2VYwH. arXiv:1906.02795
86. H.W. Kuhn, Naval Res. Logist. Q. **2**, 83 (1955). https://doi.org/10.1002/nav.3800020109
87. D.P. Kingma, J. Ba, in *3rd International Conference on Learning Representations (ICLR)*, ed. by Y. Bengio, Y. LeCun (2015). arXiv:1412.6980
88. M. Abadi et al., TensorFlow: large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015). https://www.tensorflow.org/