Eur. Phys. J. C manuscript No.
(will be inserted by the editor)

Lorentz Group Equivariant Autoencoders

Zichun Hao[1], **Raghav Kansal**[a,1,2], **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), CH-1211 Geneva 23, Switzerland

Submitted: December 15, 2022

Abstract There has been significant work recently in developing machine learning models in high energy physics (HEP), for tasks such as classification, simulation, and anomaly detection. Typically, these models are adapted from those designed for datasets in computer vision or natural language processing without necessarily incorporating inductive biases suited to HEP data, such as respecting its inherent symmetries. Such inductive biases can make the model 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 significantly outperforms a non-Lorentz-equivariant graph neural network baseline on compression and reconstruction, and anomaly detection. We also demonstrate the advantage of such an equivariant model in analyzing the latent space of the autoencoder, which can have a significant impact on the explainability of anomalies found by such black-box machine learning 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 [2–7], regression [8, 9], track reconstruction [10–12], anomaly detection [13–18], and simulation [19–24]. [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 [29] or graph neural networks (GNNs) [30–32] and permutation symmetry and sparsity of jet constituents via GNNs [5, 21, 33].

Embedding such inductive biases and symmetries into DNNs can not only improve performance, as seen with GNNs in HEP, but also improve interpretability, reduce training data requirements, and reduce network complexity in terms of learnable parameters. Motivated by this, 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 [2, 34–36], with competitive and even state-of-the-art (SOTA) results. We expand this work to the tasks of data

⋆ R. K. 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). J. D. and R. K. 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. N. C. 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

[1] Interested readers can find comprehensive reviews in Ref. [25–27] and a living review in Ref. [28].

compression and anomaly detection by incorporating the Lorentz symmetry into an autoencoder.

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 [7, 13, 15–18, 39, 40]. Both tasks are extremely important for HEP, to cope with the storage and processing of the ever-increasing data at the LHC, and 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 compare the LGAE's performance in reconstruction and anomaly detection to alternative architectures that exhibit different symmetry properties.

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

2 Related Work

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

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

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. 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 graph neural networks (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 $\mathrm{SO}(2)$ [42], the Euclidean $\mathrm{E}(2)$ [43], the 3-dimensional special orthogonal $\mathrm{SO}(3)$ [44], the 3-dimensional Euclidean $\mathrm{E}(3)$ [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). The latter approach uses the set of irreducible representations (irreps) of G as the basis functions for constructing an equivariant map [43, 52, 53], and is what we employ for the LGAE.

2.2 Lorentz Group Equivariant Neural Networks

The Lorentz group $\mathrm{O}(3, 1)$ 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) [34] was the first DNN architecture developed to be covariant to the $\mathrm{SO}^+(3, 1)$ group, based on a GNN architecture 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 nonlinearities in the network. More recently, LorentzNet [2, 35] uses a similar GNN framework for equivariance, with additional edge features — Minkowski inner products between connected node features — but restricting itself to features living only in scalar and vector representations of the group. Both networks have been successful to 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 covariantly encode and reconstruct an input set of particles in an autoencoder-style architecture.

2.3 Autoencoders in HEP

An autoencoder is a 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 a huge advantage in searches for new physics, as a specific signal is therefore not needed and instead a broader search can be performed for data incompatible with the background. This approach has been successfully demonstrated in Refs. [14, 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 models [18, 65], there has not yet been an attempt at developing an autoencoder architecture for any of these tasks that is equivariant to the Lorentz group. In this work, we focus on data compression and anomaly detection but note that our model can be extended to a variety of applications.

3 LGAE Architecture

The LGAE is built out of Lorentz group-equivariant message passing (LMP) layers, described in Section 3.1, which constitute the two encoder and decoder networks, described in Section 3.2 and Section 3.3 respectively. The LMP layer and overall model architecture are depicted in Fig. 1. The LGAE code is available at Ref. [66].

3.1 LMP

LMP layers are a form of message-passing neural network [67] layers, but based on the LGN framework in Ref. [34]. The input to each LMP layer is a fully-connected graph 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 [68], including at least one 4-vector (transforming under the $(1/2, 1/2)$ representation) representing its 4-momentum.

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

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

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

where $\mathcal{F}_i^{(t)}$ are the node features 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:

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

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

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

for all $i \in \{1, \ldots, N_{\mathrm{particle}}\}$, where W is a node-wise operator with learnable parameters which linearly mixes features in the same representation space to the desired multiplicity. As in Ref [34], we use $\tau_{(m,n)}^{(t)}$ to denote the multiplicity of the (m,n) representation in each node at the t-th LMP layer. In practice, we truncate the output irreps to a maximum dimension to make computations more tractable.

3.2 Encoder

The encoder takes as input a point cloud of N particles, termed a "particle cloud", 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 then mixed via learned weights, to a chosen multiplicity of $\left(\tau_{(m,n)}^{(0)}\right)_{\mathrm{E}}$. 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), which thereby 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.

[2] Another approach directly examines the latent space [16, 17].

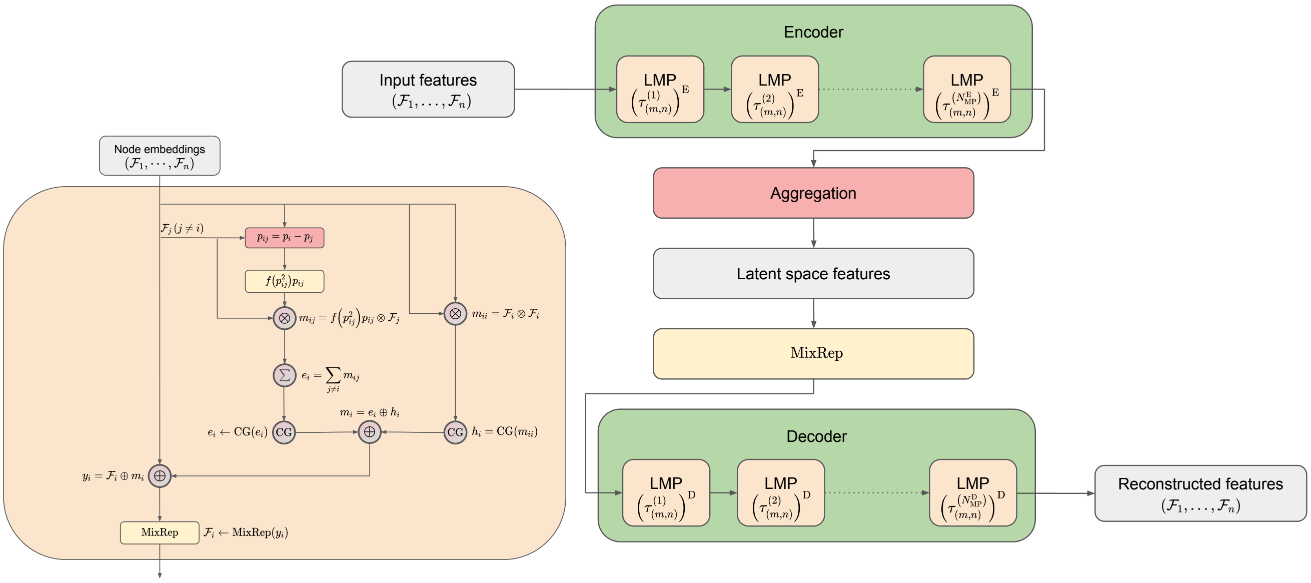

Input features
$(\mathcal{F}_1, \ldots, \mathcal{F}_n)$
Encoder
LMP $\left(\tau^{(1)}_{(m,n)}\right)^{\mathrm{E}}$
LMP $\left(\tau^{(2)}_{(m,n)}\right)^{\mathrm{E}}$
LMP $\left(\tau^{(N^{\mathrm{E}}_{\mathrm{MP}})}_{(m,n)}\right)^{\mathrm{E}}$
Aggregation
Latent space features
MixRep
Decoder
LMP $\left(\tau^{(1)}_{(m,n)}\right)^{\mathrm{D}}$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

This cloud passes through $N^{\mathrm{D}}_{\mathrm{MP}}$ LMP layers, specified by a sequence of multiplicities $\left\{\left(\tau^{(t)}_{(m,n)}\right)_{\mathrm{D}}\right\}_{t=1}^{N^{\mathrm{D}}_{\mathrm{MP}}}$, where $\left(\tau^{(t)}_{(m,n)}\right)_{\mathrm{D}}$ 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 on reconstruction and anomaly detection for simulated high-momentum jets. 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 a baseline GNN autoencoder (referred to as "GNNAE") model composed of fully-connected MPNNs adapted from Ref. [21]. We experiment with two types of encodings: (1) particle-level (GNNAE-PL), as for the PGAE [18] model, which compresses the features per node in the graph to form the latent space, and (2) jet-level (GNNAE-JL), which aggregates the features across the nodes as in the LGAE using the mean aggregation. 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. Hyperparameter and training details for both 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. We find the LGAE models to be equivariant to Lorentz boosts and rotations up to numerical error, with further details provided in Appendix C.

In this section, we briefly describe the dataset in Section 4.1, before discussing the reconstruction and anomaly detection results in Sections 4.2 and 4.3 respectively, and finally an interpretation of the latent space in Section 4.4.

4.1 Dataset

The model is trained to reconstruct 30-particle high transverse momentum jets from the JetNet [69] dataset, obtained using the associated library [70], zero-padding jets with fewer than 30, produced from gluons and light quarks. These are collectively referred to as quantum chromodynamics (QCD) jets. For particle features, we use the respective 3-momenta in absolute coordinates. 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. Further details about the dataset and its generation are available in Ref. [21].

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

Model	Aggregation	Name	Lorentz symmetry	Permutation symmetry
LGAE	Min-Max Mix	LGAE-Min-Max LGAE-Mix	✓(covariance) ✓(covariance)	✓(invariance) ✗
GNNAE	Jet-level Particle-level	GNNAE-JL GNNAE-PL	✗ ✗	✓(invariance) ✓(covariance)

4.2 Reconstruction

We evaluate the performance of the LGAE and GNNAE models, with the different aggregation schemes discussed, on the reconstruction of the particle and jet features of QCD jets. We consider relative transverse momentum $p_\mathrm{T}^\mathrm{rel} = p_\mathrm{T}^\mathrm{particle}/p_\mathrm{T}^\mathrm{jet}$ and relative angular coordinates $\eta^\mathrm{rel} = \eta^\mathrm{particle} - \eta^\mathrm{jet}$ and $\phi^\mathrm{rel} = \phi^\mathrm{particle} - \phi^\mathrm{jet} \pmod{2\pi}$ as each particle's features, and total jet mass, p_T and η as jet features. 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, while 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 permutation invariant models, particles are matched between the input and output clouds using the Jonker–Volgenant algorithm [71, 72] based on the L2 distance between particle features.

We can observe visually in Figure 2 that out of the two permutation invariant models, while neither is able to completely reconstruct the jet substructure, 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 Figure 3 and quantitatively from Tables 2 and 3, we can conclude that the LGAE-Mix has the best performance overall, significantly outperforming both GNN 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.

4.3 Anomaly Detection

We test the performance of these models as unsupervised anomaly detection algorithms by pre-training them solely on QCD and then using a reconstruction error for the QCD and new signal jets as a 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 [73], and MSE between input and output jets as reconstruction errors, and find the Chamfer distance most performant for all models.

Receiver operating characteristic (ROC) curves showing the signal efficiencies (ε_s) versus background efficiencies (ε_b) for individual and combined signals are shown in Figure 4, and ε_s values at particular background efficiencies are given in Table 4. We see that in general LGAE models have significantly higher signal efficiencies than GNNAEs 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.

4.4 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, and indeed we are able to understand the latent spaces of LGAE models to a considerable extent. For example, we learn how the LGAE-Mix model with $\tau_{(1/2,1/2)} = 2$ is encoding the target jet momentum from Fig. 5, which shows a significant correlation between the target jet momenta and the total momentum of the imaginary components of each latent vector.

We can also 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, explaining the high performance in anomaly detection. We also note that through the LGAE we have derived novel Lorentz-invariant jet observables in the form of these invariant masses, a detailed study of which we leave to future work.

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

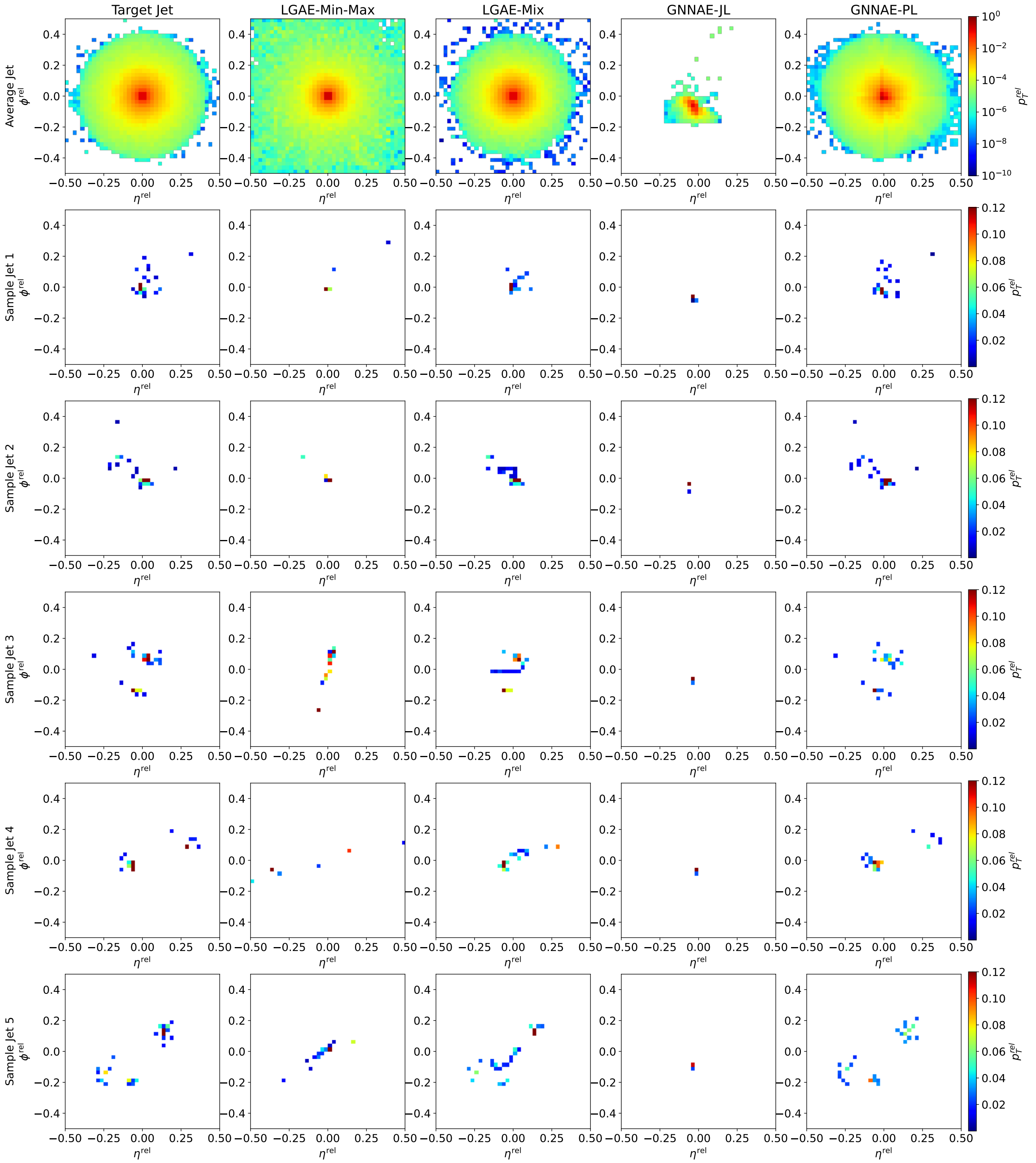


Fig. 2 Jet image reconstructions by LGAE-Min-Max ($\tau_{(1/2,1/2)} = 4$, 56.7% compression), LGAE-Mix ($\tau_{(1/2,1/2)} = 9$, 61.7% compression), GNNAE-JL ($\dim(L) = 55$, 61.11% compression), and GNNAE-PL ($\dim(L) = 2 \times 30$, 66.67% compression).

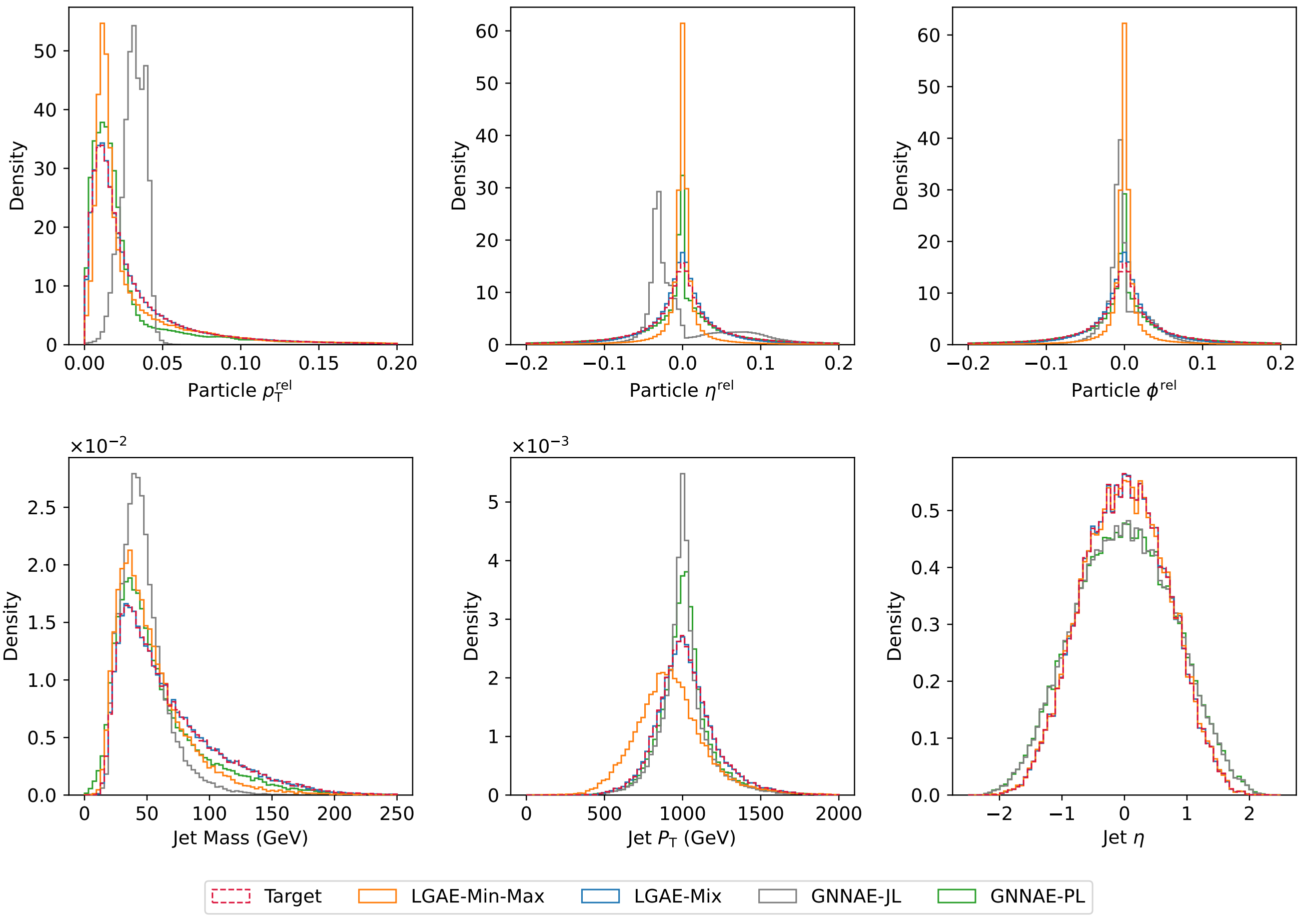


Fig. 3 **Top**: Particle momenta (p_T^{rel}, η^{rel}, ϕ^{rel}) reconstruction by LGAE-Min-Max ($\tau_{(1/2,1/2)} = 4$, resulting in 56.67% compression) and and LGAE-Mix ($\tau_{(1/2,1/2)} = 9$, resulting in 61.67% compression), and GNNAE-JL ($\dim(L) = 55$, resulting in 61.11% compression) and GNNAE-PL ($\dim(L) = 2 \times 30$, resulting in 66.67% compression). **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.

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 out-of-training-distribution anomalous top quark, W boson, and Z boson jets. We report excellent performance in comparison to a baseline graph neural network autoencoder (GNNAE) model, with the LGAE outperforming the GNNAE on several key metrics. We also demonstrate the interpretability of the LGAE by analyzing the latent space of the LGAE models for both tasks. The LGAE opens up many promising avenues both in terms of performance and model interpretability, with the exploration of new datasets, higher-order Lorentz group representations, and characterization of the derived jet observables in the LGAE latent space 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.

References

1. A. J. Larkoski, I. Moult, and B. Nachman, “Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning”, *Phys. Rept.* **841** (2020) 1, doi:10.1016/j.physrep.2019.11.001, arXiv:1709.04464.
2. S. Gong et al., “An efficient Lorentz equivariant graph neural network for jet tagging”, *JHEP* **07** (2022) 030, doi:10.1007/JHEP07(2022)030, arXiv:2201.08187.
3. J. Pearkes, W. Fedorko, A. Lister, and C. Gay, “Jet Constituents for Deep Neural Network Based Top Quark Tagging”, arXiv:1704.02124.

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 $p_{\mathrm{T}}^{\mathrm{rel}}$		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
		$\tau_{(1/2,1/2)} = 7$ (56.67%)	0.002	0.640	−0.627	*1.7*	**$< 10^{-5}$**	*1.7*
	Mix	$\tau_{(1/2,1/2)} = 9$ (61.67%)	**$< 10^{-5}$**	0.011	**$< 10^{-5}$**	0.452	**$< 10^{-5}$**	0.451
		$\tau_{(1/2,1/2)} = 13$ (88.33%)	**$< 10^{-5}$**	**0.001**	**$< 10^{-5}$**	**0.022**	**$< 10^{-5}$**	**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. 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*	**$< 10^{-5}$**	*0.004*	**$< 10^{-5}$**	*0.001*
		$\tau_{(1/2,1/2)} = 7$ (96.67%)	−0.139	0.287	−0.221	0.609	**$< 10^{-5}$**	0.021	**$< 10^{-5}$**	0.007
	Mix	$\tau_{(1/2,1/2)} = 9$ (61.67%)	**$< 10^{-5}$**	0.008	**$< 10^{-5}$**	0.002	**$< 10^{-5}$**	0.0004	**$< 10^{-5}$**	**0.0001**
		$\tau_{(1/2,1/2)} = 13$ (88.33%)	0.001	**0.003**	**$< 10^{-5}$**	**0.001**	**$< 10^{-5}$**	**0.0002**	**$< 10^{-5}$**	**0.0001**
GNNAE	Jet-level	$\dim(L) = 45$ (50.00%)	0.326	*0.667*	*0.030*	*0.088*	*0.005*	*0.040*	*0.994*	*1.3*
		$\dim(L) = 90$ (100.00%)	3.7	2.6	*0.030*	0.089	0.292	0.433	0.996	1.4
	Particle-level	$\dim(L) = 2 \times 30$ (66.67%)	*0.277*	0.299	*0.037*	0.110	0.002	*0.010*	0.199	*1.3*
		$\dim(L) = 3 \times 30$ (100.00%)	0.339	*0.244*	0.050	*0.094*	*−0.001*	0.011	*0.150*	*1.3*

Table 4 Anomaly detection metrics by a 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	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*

4. S. Macaluso and D. Shih, "Pulling Out All the Tops with Computer Vision and Deep Learning", *JHEP* **10** (2018) 121, doi:10.1007/JHEP10(2018)121, arXiv:1803.00107.
5. H. Qu and L. Gouskos, "ParticleNet: Jet Tagging via Particle Clouds", *Phys. Rev. D* **101** (2020) 056019, doi:10.1103/PhysRevD.101.056019, arXiv:1902.08570.
6. P. W. Battagliaet al., "Interaction Networks for Learning about Objects, Relations and Physics", in *Advances in Neural Information Processing Systems*, D. D. Lee et al., eds., volume 29, p. 4502. Curran Associates, Inc., 2016. arXiv:1612.00222.
7. S. E. Park et al., "Quasi Anomalous Knowledge: Searching for new physics with embedded knowledge", *JHEP* **21** (2020) 030, doi:10.1007/JHEP06(2021)030, arXiv:2011.03550.
8. F. Bury and C. Delaere, "Matrix element regression with deep neural networks — Breaking the CPU barrier", *JHEP* **04** (2021) 020, doi:10.1007/JHEP04(2021)020, arXiv:2008.10949.
9. D. Belayneh et al., "Calorimetry with deep learning: particle simulation and reconstruction for collider physics", *Eur. Phys. J. C* **80** (2020) 688,

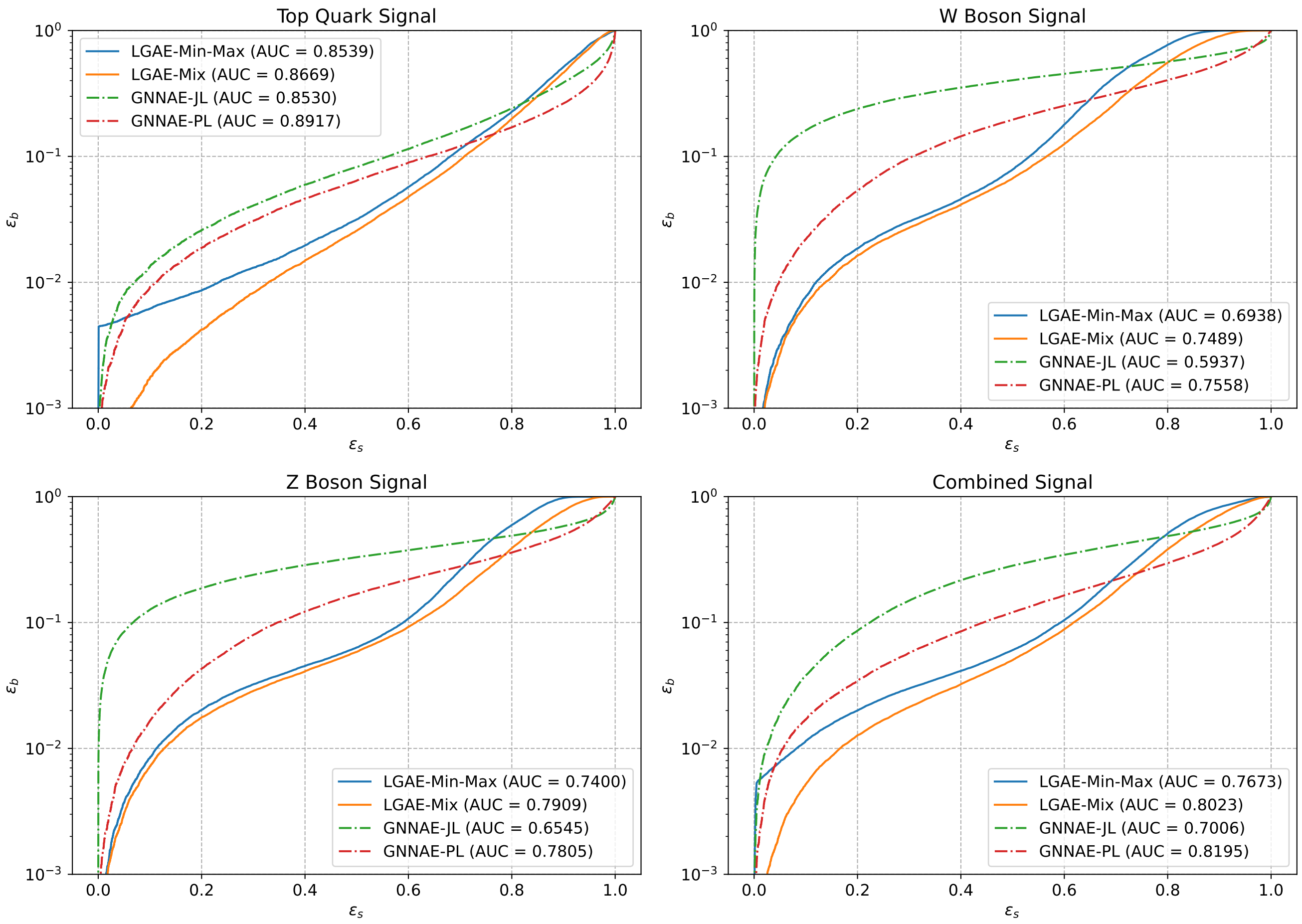


Fig. 4 Anomaly detections for the top quark signal (upper left), W boson signal (upper right), Z boson signal (lower left), and the combined signal (lower right) by the selected LGAE-Min-Max ($\tau_{(1/2,1/2)} = 7$), LGAE-Mix ($\tau_{(1/2,1/2)} = 2$), GNNAE-JL ($\dim(L) = 30$), and GNNAE-PL ($\dim(L)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

doi:10.1140/epjc/s10052-020-8251-9, arXiv:1912.06794.

10. J. Duarte and J.-R. Vlimant, "Graph neural networks for particle tracking and reconstruction", in *Artificial Intelligence for Particle Physics*. World Scientific Publishing, 2020. arXiv:2012.01249. Submitted to *Int. J. Mod. Phys. A*. doi:10.1142/12200.
11. S. Farrell et al., "Novel deep learning methods for track reconstruction", in *4th International Workshop Connecting The Dots 2018*. 2018. arXiv:1810.06111.
12. G. DeZoort et al., "Charged Particle Tracking via Edge-Classifying Interaction Networks", *Comput. Softw. Big Sci.* **5** (2021) 26, doi:10.1007/s41781-021-00073-z, arXiv:2103.16701.
13. O. Atkinson et al., "Anomaly detection with convolutional Graph Neural Networks", *JHEP* **08** (2021) 080, doi:10.1007/JHEP08(2021)080, arXiv:2105.07988.
14. T. Heimel, G. Kasieczka, T. Plehn, and J. M. Thompson, "QCD or What?", *SciPost Phys.* **6** (2019), no. 3, 030, doi:10.21468/SciPostPhys.6.3.030, arXiv:1808.08979.
15. F. Canelli et al., "Autoencoders for semivisible jet detection", *JHEP* **02** (2022) 074, doi:10.1007/JHEP02(2022)074, arXiv:2112.02864.
16. T. Cheng et al., "Variational Autoencoders for Anomalous Jet Tagging", 2020. arXiv:2007.01850. Accepted by *Phys. Rev. D*.
17. B. Bortolato, A. Smolkovič, B. M. Dillon, and J. F. Kamenik, "Bump hunting in latent space", *Phys. Rev. D* **105** (2022) 115009, doi:10.1103/PhysRevD.105.115009, arXiv:2103.06595.
18. S. Tsanet al., "Particle Graph Autoencoders and Differentiable, Learned Energy Mover's Distance", in *4th Machine Learning and the Physical Sciences Workshop at the 35th Conference on Neural Information Processing Systems*. 2021. arXiv:2111.12849.
19. L. de Oliveira, M. Paganini, and B. Nachman, "Learning Particle Physics by Example: Location-Aware

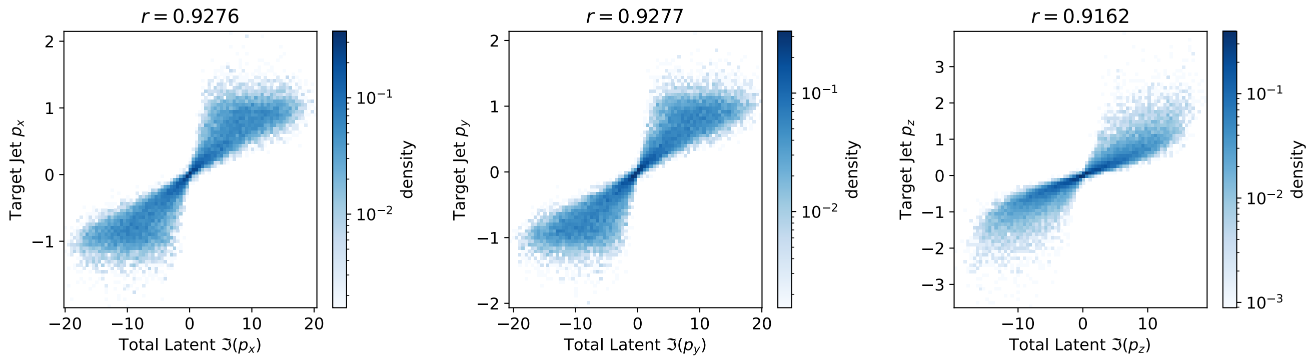


Fig. 5 The correlations between the total momentum of the imaginary components in the $\tau_{(1/2,1/2)} = 2$ LGAE-Mix model and the target jet momenta. The Pearson correlation coefficient r is listed above.

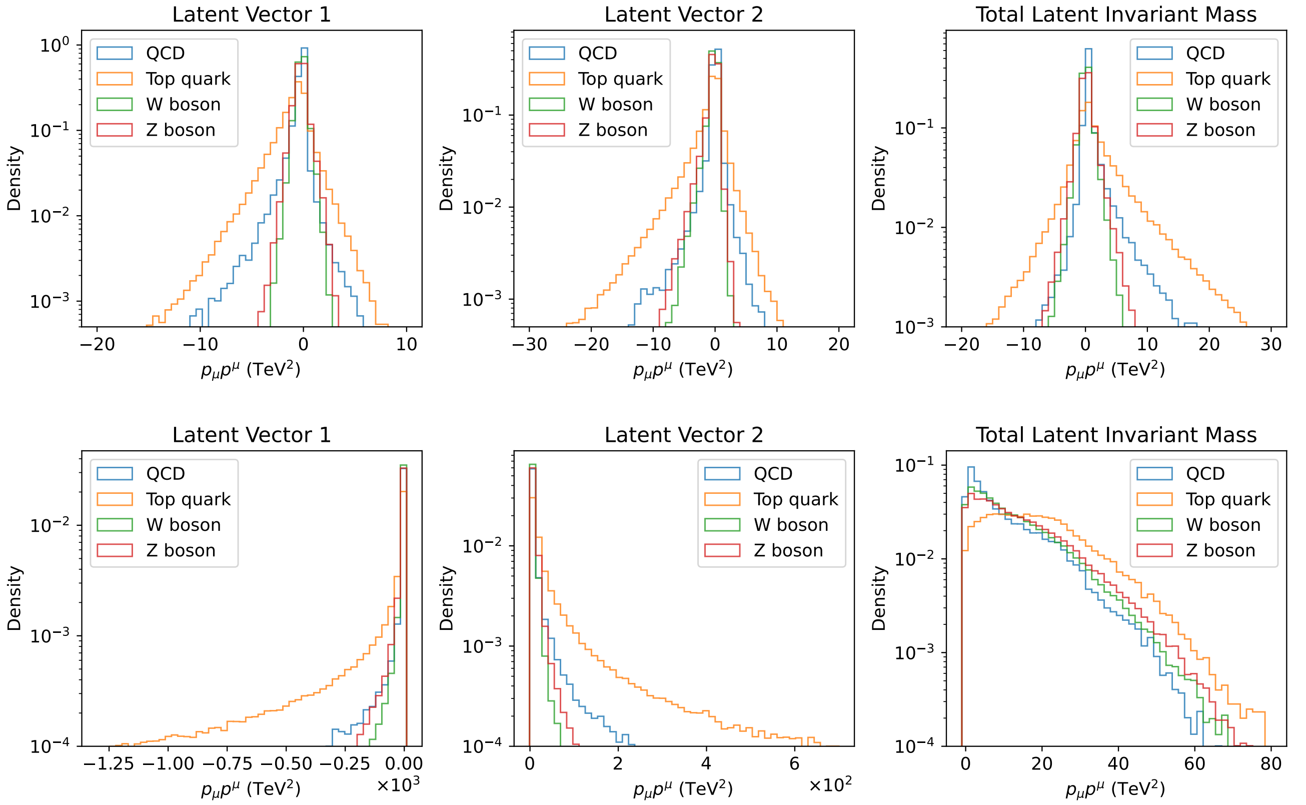


Fig. 6 **Top**: Distributions of the invariant mass squared of the latent 4-vectors and jet momenta of the LGAE-Mix with $\tau_{(1/2,1/2)} = 2$ latent 4-vectors. **Bottom**: Distributions of the invariant mass squared of two latent 4-vectors and jet momenta of the LGAE-Min-Max with $\tau_{(1/2,1/2)}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

Generative Adversarial Networks for Physics Synthesis", *Comput. Softw. Big Sci.* **1** (2017), no. 1, 4, doi:10.1007/s41781-017-0004-6, arXiv:1701.05927.

20. M. Paganini, L. de Oliveira, and B. Nachman, "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters", *Phys. Rev. Lett.* **120** (2018) 042003, doi:10.1103/PhysRevLett.120.042003, arXiv:1705.02355.
21. R. Kansalet al., "Graph generative adversarial networks for sparse data generation in high energy physics", in *3rd Machine Learning and the Physical Sciences Workshop at the 34th Conference on Neural Information Processing Systems*. 2020. arXiv:2012.00173.
22. K. Dohi, "Variational Autoencoders for Jet Simulation", 2020. arXiv:2009.04842.
23. M. Touranakou et al., "Particle-based fast jet simulation at the LHC with variational autoencoders", *Mach. Learn.: Sci. Technol.* **3** (2022) 035003,

doi:10.1088/2632-2153/ac7c56, arXiv:2203.00520.

24. M. Paganini, L. de Oliveira, and B. Nachman, "CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks", *Phys. Rev. D* **97** (2018) 014021, doi:10.1103/PhysRevD.97.014021, arXiv:1712.10321.
25. D. Guest, K. Cranmer, and D. Whiteson, "Deep Learning and its Application to LHC Physics", *Ann. Rev. Nucl. Part. Sci.* **68** (2018) 161, doi:10.1146/annurev-nucl-101917-021019, arXiv:1806.11484.
26. A. Radovic et al., "Machine learning at the energy and intensity frontiers of particle physics", *Nature* **560** (2018), no. 7716, 41, doi:10.1038/s41586-018-0361-2.
27. G. Carleo et al., "Machine learning and the physical sciences", *Rev. Mod. Phys.* **91** (2019), no. 4, 045002, doi:10.1103/RevModPhys.91.045002, arXiv:1903.10563.
28. HEP ML Community, "A living review of machine learning for particle physics", 2021. https://iml-wg.github.io/HEPML-LivingReview/.
29. P. T. Komiske, E. M. Metodiev, and J. Thaler, "Energy Flow Networks: Deep Sets for Particle Jets", *JHEP* **01** (2019) 121, doi:10.1007/JHEP01(2019)121, arXiv:1810.05165.
30. 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, doi:10.1007/JHEP02(2022)060, arXiv:2109.14636.
31. O. Atkinson et al., "IRC-Safe Graph Autoencoder for Unsupervised Anomaly Detection", *Front. AI* **5** (2022) 943135, doi:10.3389/frai.2022.943135, arXiv:2204.12231.
32. J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph Neural Networks in Particle Physics", doi:10.1088/2632-2153/abbf9a, arXiv:2007.13681.
33. S. Thais et al., "Graph Neural Networks in Particle Physics: Implementations, Innovations, and Challenges", in *2022 Snowmass Summer Study*. 3, 2022. arXiv:2203.12852.
34. A. Bogatskiyet al., "Lorentz group equivariant neural network for particle physics", arXiv:2006.04780.
35. C. Li et al., "Does Lorentz-symmetric design boost network performance in jet physics?", 2022. arXiv:2208.07814.
36. A. Butter, G. Kasieczka, T. Plehn, and M. Russell, "Deep-learned Top Tagging with a Lorentz Layer", *SciPost Phys.* **5** (2018) 028, doi:10.21468/SciPostPhys.5.3.028, arXiv:1707.08966.
37. J. H. Collins, "An Exploration of Learnt Representations of W Jets", in *ICLR workshop Deep Generative Models for Highly Structured Data*. 2021. arXiv:2109.10919.
38. J. W. Monk, "Deep Learning as a Parton Shower", *JHEP* **12** (2018) 021, doi:10.1007/JHEP12(2018)021, arXiv:1807.03685.
39. M. Farina, Y. Nakai, and D. Shih, "Searching for New Physics with Deep Autoencoders", *Phys. Rev. D* **101** (2020) 075021, doi:10.1103/PhysRevD.101.075021, arXiv:1808.08992.
40. T. Finke et al., "Autoencoders for unsupervised anomaly detection in high energy physics", *JHEP* **06** (2021) 161, doi:10.1007/JHEP06(2021)161, arXiv:2104.09051.
41. M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges", 2021. arXiv:2104.13478.
42. R. Walters, J. Li, and R. Yu, "Trajectory prediction using equivariant continuous convolution", in *International Conference on Learning Representations*. 2021. arXiv:2010.11344.
43. M. Weiler and G. Cesa, "General E(2)-equivariant steerable CNNs", in *Advances in Neural Information Processing Systems*, H. Wallach et al., eds., volume 32. Curran Associates, Inc., 2019.
44. C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) equivariant representations with spherical CNNs", *Int. J. Comput. Vis.* **128** (2020), no. 3, 588, doi:10.1007/s11263-019-01220-1.
45. N. Thomas et al., "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds", 2018. arXiv:1802.08219.
46. S. Batzner et al., "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials", *Nature Communications* **13** (2022), no. 1, 2453, doi:10.1038/s41467-022-29939-5.
47. M. Finzi, M. Welling, and A. G. Wilson, "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups", in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, eds., volume 139 of *Proceedings of Machine Learning Research*, p. 3318. PMLR, 2021. arXiv:2104.09459.
48. T. Cohen, M. Geiger, and M. Weiler, "A general theory of equivariant CNNs on homogeneous spaces", doi:10.48550/ARXIV.1811.02017.

49. M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson, "Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data", 2020. doi:10.48550/ARXIV.2002.12880, https://arxiv.org/abs/2002.12880.
50. T. Cohen and M. Welling, "Group equivariant convolutional networks", in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, eds., volume 48 of *Proceedings of Machine Learning Research*, p. 2990. PMLR, New York, New York, USA, 2016. arXiv:1602.07576.
51. M. Geiger and T. Smidt, "e3nn: Euclidean neural networks", 2022. doi:10.48550/ARXIV.2207.09453, https://arxiv.org/abs/2207.09453.
52. R. Kondor, Z. Lin, and S. Trivedi, "Clebsch-Gordan Nets: a fully Fourier space spherical convolutional neural network", 2018. doi:10.48550/ARXIV.1806.09231, https://arxiv.org/abs/1806.09231.
53. B. Anderson, T.-S. Hy, and R. Kondor, "Cormorant: Covariant molecular neural networks", 2019. doi:10.48550/ARXIV.1906.04015, https://arxiv.org/abs/1906.04015.
54. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", *Science* **313** (2006), no. 5786, 504, doi:10.1126/science.1127647.
55. G. Di Guglielmo et al., "A Reconfigurable Neural Network ASIC for Detector Front-End Data Compression at the HL-LHC", *IEEE Trans. Nucl. Sci.* **68** (2021), no. 8, 2179, doi:10.1109/TNS.2021.3087100, arXiv:2105.01683.
56. O. Cerri et al., "Variational Autoencoders for New Physics Mining at the Large Hadron Collider", *JHEP* **05** (2019) 036, doi:10.1007/JHEP05(2019)036, arXiv:1811.10276.
57. G. Kasieczka et al., "The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics", *Rept. Prog. Phys.* **84** (2021), no. 12, 124201, doi:10.1088/1361-6633/ac36b9, arXiv:2101.08320.
58. E. Govorkova et al., "Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider", *Nat. Mach. Intell.* **4** (2022) 154, doi:10.1038/s42256-022-00441-3, arXiv:2108.03986.
59. A. A. Pol et al., "Anomaly Detection With Conditional Variational Autoencoders", in *18th International Conference on Machine Learning and Applications*. 2020. arXiv:2010.05531.
60. V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi, "Anomaly detection in high-energy physics using a quantum autoencoder", *Phys. Rev. D* **105** (2022) 095004, doi:10.1103/PhysRevD.105.095004, arXiv:2112.04958.
61. B. M. Dillon et al., "Symmetries, safety, and self-supervision", *SciPost Phys.* **12** (2022), no. 6, 188, doi:10.21468/SciPostPhys.12.6.188, arXiv:2108.04253.
62. D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders", 2020. arXiv:2003.05991.
63. M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning", 2018. arXiv:1812.05069.
64. D. P. Kingma and M. Welling, "Auto-encoding variational bayes", in *2nd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds. 2014. arXiv:1312.6114.
65. O. Atkinson et al., "Anomaly detection with convolutional Graph Neural Networks", *JHEP* **08** (2021) 080, doi:10.1007/JHEP08(2021)080, arXiv:2105.07988.
66. Z. Hao, "zichunhao/lgn-autoencoder: V1.0.0", 2022. doi:10.5281/zenodo.7434799, https://github.com/zichunhao/lgn-autoencoder.
67. J. Gilmeret al., "Neural message passing for quantum chemistry", in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, eds., volume 70 of *Proceedings of Machine Learning Research*, p. 1263. PMLR, 2017. arXiv:1704.01212.
68. G. C. I. M. Gelfand, R. A. Minlos, "Representations of the Rotation and Lorentz Groups and Their Applications". Pergamon Press, Oxford, England, 1963.
69. R. Kansal et al., "Jetnet", 2022. doi:10.5281/zenodo.6975118.
70. R. Kansal, C. Pareja, and J. Duarte, "jet-net/JetNet: v0.2.1.post2", 2022. doi:10.5281/zenodo.7067466, https://github.com/jet-net/JetNet.
71. R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems", *Computing* **38** (1987) 325, doi:10.1007/BF02278710.
72. P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nat. Methods* **17** (2020) 261, doi:10.1038/s41592-019-0686-2.
73. P. T. Komiske, E. M. Metodiev, and J. Thaler, "Metric Space of Collider Events", *Phys. Rev. Lett.* **123** (2019) 041801, doi:10.1103/PhysRevLett.123.041801, arXiv:1902.02346.

74. H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, “Parametric correspondence and Chamfer matching: Two new techniques for image matching”, in *Proceedings of the 5th International Joint Conference on Artificial Intelligence (KJCAI)*, volume 2, p. 659. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977.
75. H. Fan, H. Su, and L. J. Guibas, “A point set generation network for 3D object reconstruction from a single image”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 2463. 6, 2017. arXiv:1612.00603. doi:10.1109/CVPR.2017.264.
76. Y. Zhang, J. Hare, and A. Prügel-Bennett, “FSPool: Learning set representations with featurewise sort pooling”, in *8th International Conference on Learning Representations*. 2020. arXiv:1906.02795.
77. H. W. Kuhn, “The Hungarian method for the assignment problem”, *Naval Research Logistics Quarterly* **2** (1955) 83, doi:10.1002/nav.3800020109.
78. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, in *3rd International Conference on Learning Representations (ICLR)*, Y. Be
Y. LeCun, eds. 2015. arXiv:1412.698
79. A. Paszkeet al., “PyTorch: An imperativ
high-performance deep learning library
in Neural Information Processing Syste
p. 8024. Curran Associates, Inc., 2019. arXiv:1912.01703.

Appendix A: Model Details

Appendix A.1: LGAE

For both encoder and decoder, we choose *N*
LMP layers. The multiplicity per node in eac
been optimized to be

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

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. After each CG decomposition, we truncate irreps of dimensions higher than $(1/2, 1/2)$ for tractable computations. Empirically, we did not find such a truncation to affect the performance of the model.

The differentiable mapping $f(d_{ij})$ in Eq.(2) is chosen to be the Lorentzian bell function as in Ref. [34]. For all models, the latent space contains only $\tau_{(0,0)} = 1$ complex Lorentz scalar, as increasing this does not improve the performance in either reconstruction or anomaly detection.

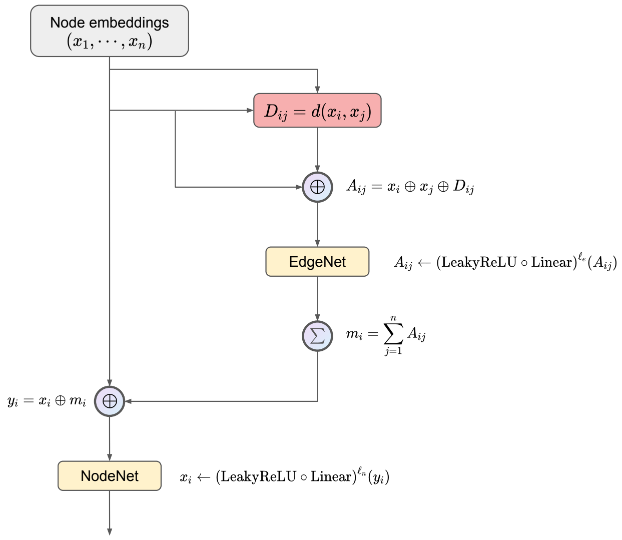
Appendix A.2: GNNAE

The GNNAE is constructed from fully-connected MPNNs. The update rule in the $(t+1)$-th MPNN layer is based on Ref. [21], 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. 7. 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. [66].



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
$A_{ij} \leftarrow (\mathrm{LeakyReLU} \circ \mathrm{Linear})^{\ell_e}(A_{ij})$
$m_i = \sum_{j=1}^{n} A_{ij}$
$y_i = x_i \oplus m_i$
NodeNet
$x_i \leftarrow (\mathrm{LeakyReLU} \circ \mathrm{Linear})^{\ell_n}(y_i)$

Fig. 7 An MPNN layer in the GNNAE. Here, EdgeNet and NodeNet are feed-forward neural networks.

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

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

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

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

where $\mathrm{LeakyRelu}_{0.2}(x) = \max(0.2x, x)$ 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 B: Training Details

We use the Chamfer loss function [74–76] for the LGAE-Min-Max and GNNAE-JL models, and MSE for LGAE-Mix and GNNAE-PL. We tested the Hungarian loss [72, 77] and differentiable energy mover's distance (EMD) [73], calculated using the JetNet library [70], as well but found the Chamfer and MSE losses more performant.

All models are optimized using the Adam optimizer [78] implemented in PyTorch [79] with a learning rate $\gamma = 10^{-3}$, coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay $\lambda = 0$. They are each trained on single NVIDIA RTX 2080 Ti GPUs each for a maximum of 2000 epochs using early stopping with a patience of 200 epochs. The total training time for LGAE models is typically 35 hours, and at most 100 hours, while GNNAE-PL and GNNAE-JL train for 50 and 120 hours on average, respectively.

Appendix C: Covariance 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 [34], so we use double precision (64-bit) throughout the model. In addition, we scale down the data by a factor of 1,000 (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

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

Figure 8 shows the mean relative deviation, averaged over each particle in each jet, over 3000 jets from our test dataset in rotation and boost equivariance test.

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 $< 10^{12}$.

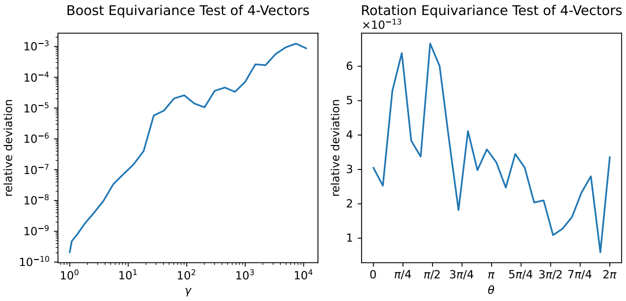


Fig. 8 The relative deviations, as defined in Eq. (C.8), of the output 4-momenta p^μ to boosts along the z-axis (left) and rotations around the z-axis (right).