Neural embedding: learning the embedding of the manifold of physics data

Citation: Journal of High Energy Physics. 2023 Jul 12;2023(7):108

As Published: https://doi.org/10.1007/JHEP07(2023)108

Publisher: Springer Berlin Heidelberg

Persistent URL: https://hdl.handle.net/1721.1/151120

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Published for SISSA by Springer

Received: January 29, 2023
Revised: May 31, 2023
Accepted: July 5, 2023
Published: July 12, 2023

Neural embedding: learning the embedding of the manifold of physics data

Sang Eon Park,[a,b] **Philip Harris**[a,b] **and Bryan Ostdiek**[c,b]

[a] *Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
[b] *The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, U.S.A.*
[c] *Department of Physics, Harvard University, Cambridge, MA 02138, U.S.A.*

E-mail: sangeon@mit.edu, pcharris@mit.edu, bostdiek@gmail.com

Abstract: In this paper, we present a method of embedding physics data manifolds with metric structure into lower dimensional spaces with simpler metrics, such as Euclidean and Hyperbolic spaces. We then demonstrate that it can be a powerful step in the data analysis pipeline for many applications. Using progressively more realistic simulated collisions at the Large Hadron Collider, we show that this embedding approach learns the underlying latent structure. With the notion of volume in Euclidean spaces, we provide for the first time a viable solution to quantifying the true search capability of model agnostic search algorithms in collider physics (i.e. anomaly detection). Finally, we discuss how the ideas presented in this paper can be employed to solve many practical challenges that require the extraction of physically meaningful representations from information in complex high dimensional datasets.

Keywords: Jets and Jet Substructure, Dark Matter at Colliders

ArXiv ePrint: 2208.05484

https://doi.org/10.1007/JHEP07(2023)108

Contents

JHEP07(2023)108

1 Introduction

Despite being high dimensional, physics datasets are highly structured since physical laws strictly govern the data generating process. Although the data is complicated, it is not hard to imagine that physics data can exist within low-dimensional manifolds inside a high-dimensional ambient space.

There is a growing recent interest in endowing the space of collider events with a metric structure calculated directly in the space of its inputs. Metrics based on optimal transport, such as energy mover's distance (EMD) [1], Hellinger distance [2], and sliced Wasserstein distance [3] , allow us to compare raw inputs directly and quantify the global structural difference between any pair of collider events. Since the advent of these studies, a broad range of use cases has been emerging for these metrics. These include event tagging, anomaly tagging [4–6], representation learning of jets [7], and measurements of Quantum Chromo Dynamical (QCD) properties [8].

However, the input dimension is usually very large for collider data; thus, the induced manifold of the metric lives in a very high dimensional space, making it challenging to work with directly. With just 50 particles and 3 features per particle, the induced manifold lives in $\mathbb{R}^{150}$, a prohibitively large dimensional space subject to the curse of dimensionality.

After decades of searching at the LHC, no new physics beyond the Standard Model has been observed despite a large variety of targeted searches for new physics models. In light of this, we are starting to consider that maybe we are not looking in the right area of the collider data, and we should go beyond our existing new physics models. The shift from targeted searches to model agnostic searches is happening rapidly, and a diverse and rich variety of model agnostic search methods has been proposed by the community, based on a wide variety of different underlying principles [9–13, 13–17, 17–36].

However, ways to evaluate and quantify the performance of these algorithms are less studied. We can't systematically study different anomaly detection methods without a good method to quantify and study how each algorithm performs. Consequently, there is a strong need to come up with ways to quantify each method, especially to understand how far in the search space the algorithm can reach and how wide of a net is cast in the space of total possible physics events.

This paper introduces a flexible framework for embedding the manifold of collider events in lower-dimensional spaces. This framework allows physicists to get the most out of metric space properties of collider events and demonstrate that it can be used to quantify different anomaly detection algorithms for model agnostic searches. Moreover, we show this embedding space captures core physical features and self assembles events into physically meaningful categories.

Standard jet finding algorithms embed events into lower-dimensional manifolds taking the individual particle energies and angles and replacing them with a single jet energy and direction. However, jet finding is the result of a complex iterative computation and the ensuing embedded manifold structure makes it difficult to compare jets from different decays with limitations in how to interpret jets from a variety of physical processes. This paper aims to tackle the problem of metric embedding: when we have some well defined

JHEP07(2023)108

metric already defined on the original space of jets that captures jets from a variety of processes and we seek to construct a lower dimensional space that preserves our metric so as to extract maximal information. By preserving the metric, the embedding allows us to define the notion of volume in the manifold, which leads to a strategy to quantify the space of jets selected by some means including through the identification of anomalies with model agnostic search algorithms.

We primarily focus on learning embedding functions into lower-dimensional spaces with the goal of approximating the given original metric on the space of collider events. We will show that low distortion and robust embedding can be achieved in very low dimensions, down to two dimensions. Different choices of space where we can embed the physics event are also explored. We discuss the advantage of learning the embedding by training the embedding function to approximate the metric distance in the original space over out-of-the-box manifold learning methods such as t-SNE [37] and UMAP [38].

The strength of the proposed method is presented with emphasis on quantifying anomaly detection algorithms. The embedding is a useful method of anomaly detection itself, but more importantly, it can address the bigger problem of quantifying the effectiveness of each technique. Using the notion of volume in the embedded space, we propose the volume-adjusted ROC curve, which in two dimensions becomes the area-adjusted ROC curve that tries to measure the "volume" of the total search space encompassed by an algorithm. We then quantify the performance of two different anomaly detection algorithms on a fixed dataset. We additionally show that low distortion embedding is useful for many different aspects of physics analysis by presenting a visualization and exploration of what is learned by the embedding.

Outside the realm of anomaly detection, we demonstrate that with embedding, we can tackle many problems. Mapping complicated metrics to simpler metrics gives access to a powerful algorithmic toolkit that allows us to do approximation, online analysis, data compression, and classification. Furthermore, mapping the original space to a lower-dimensional space makes many tasks, such as visualization, much easier. Embedding can also be used for data compression [7] by compressing the information about jets down to a few numbers corresponding to the dimension of the embedded space and the metrics learned in the embedding process.

Embedding is particularly computationally tractable and scales better than the pairwise computation of the distance between events. Since embedding is embarrassingly parallel and can be calculated relatively cheaply through a single forward pass of the neural network, embedding can be computed in real-time, leading to further possibilities for low latency event classification and online decision-making within a trigger system.

Different embedding techniques have led to successes in many fields, such as dealing with biological sequences and phylogenetic analysis [39], graph and network analysis [40–43], natural language processing [44–47], computer vision [48, 49], amongst others. This technique is quickly gaining popularity and falls within a large space of machine learning aimed at effectively, scientifically motivated data representation.

For this paper, we demonstrate the embedding of hadronically decaying final states consisting of resonances that subsequently decay to intermediate resonances resolved as

JHEP07(2023)108

jets with as many as 4-prongs over a large variety of masses. With this diverse set of events, we apply embedding and show that even with two dimensions, we can capture the core physical features. Furthermore, we developed a simplified "toy jet" generator to create complex objects under strict kinematic restrictions and a minimal set of parameters.We use this toy dataset as a testing ground for our method to check whether the embedding learns the proper latent structure and self-organizes the space into distinct features.

Embedding into a lower-dimensional space can also be seen as an alternate way of building a simpler space to perform physics measurements, searches, and classification. In papers such as [50, 51], mapping to lower dimensional space was performed using contrastive loss metric, and the study of the anomaly detection using this mapping with the contrastive loss metric was performed. We perform the metric distance embedding, starting with the given distance between events, by directly building the space through embedding with optimal transport distances on the original space, yielding a new handle on how to organize and classify data.

2 Neural embedding

In this paper, we develop a neural embedding to take collider events and embed them into a manifold governed by physically motivated principles. The key to performing this analysis is to show that this embedding is robust across a variety of datasets. In the following section, we will outline the core idea of the neural embedding, and motivate the choice and design of the simulated collider datasets. Our goal is to progressively build more complicated datasets towards an intuitive understanding for how this approach can be applied on fully realistic collider data.

2.1 Problem setting

Suppose we have a metric space $(\mathcal{X}, d)$ of collider events, where $\mathcal{X}$ denotes the space of collider events and d is some metric defined on the space. For collider events, this space is often represented by a list of all the particles in the event with their subsequent features. With collisions comprising hundreds of particles in the final state, the resulting space is high dimensional and metrics on the space are computationally difficult. Starting with the energy mover's distance (EMD) [1], many metrics based on ideas from optimal transport have been proposed on the collider events where the space $\mathcal{X}$ is represented as a subset of high dimensional input space $\mathbb{R}^D$ [2]. These metrics are capable of effectively interpreting collider events, but are often computationally intractable and hard to interpret within $\mathbb{R}^D$.

Our goal in this paper is to simplify the interpretation of $\mathbb{R}^D$ by learning an embedding map to a low dimensional space that preserves our metrics d on the higher dimensional space. In other words, we aim to create a low dimensional space $\mathcal{Y}$ through $\phi : (\mathcal{X}, d_{\mathcal{X}}) \to (\mathcal{Y}, d_{\mathcal{Y}})$. While a perfect embedding where the distance between any two events is perfectly preserved is not guaranteed, we aim to learn a mapping that satisfies a low distortion, namely it satisfies the relation

$$\forall u, v \in \mathcal{X},\ L \cdot d_{\mathcal{Y}}(\phi(u), \phi(v)) < d_{\mathcal{X}}(u, v) < C \cdot d_{\mathcal{Y}}(\phi(u), \phi(v)) \tag{2.1}$$

JHEP07(2023)108

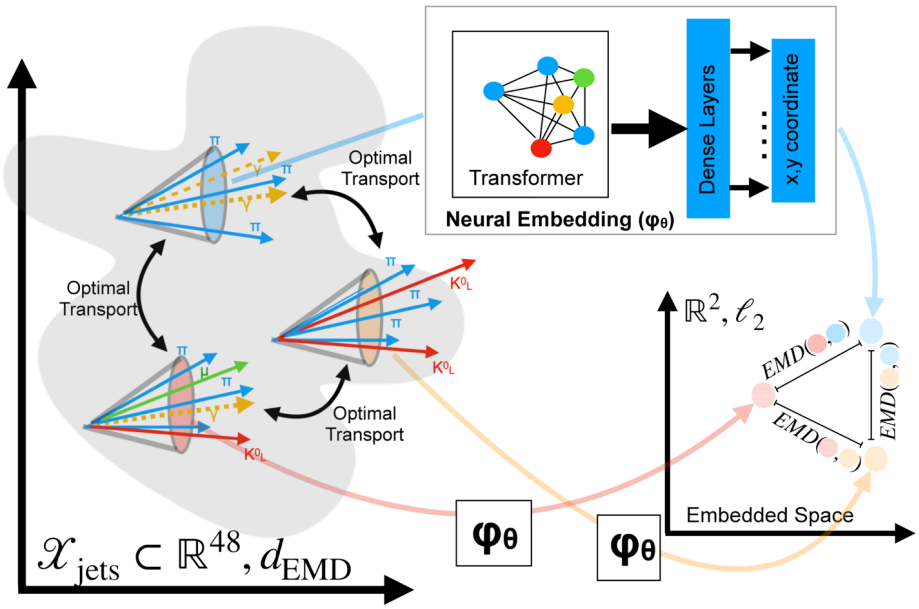

Figure 1. Diagrammatic representation of the distance preserving embedding. Grey region represents the data manifold, three different types of jets represent three points on the manifold which gets mapped to $(\mathbb{R}^2, l_2)$ by the learned embedding. The energy mover's distance in the original space is preserved in the embedded space.

For some $0 < L < 1$ and $C \geq 1$, where smaller C indicates a smaller overall distortion in the space and the metric on the space $d_{\mathcal{Y}}$ is a simpler metric than the original metric $d_{\mathcal{X}}$. The constant L is the inverse of the Lipschitz constant for the mapping ϕ, and it guarantees that the metric distance doesn't blow up in the embedded space.

Therefore our learning objective can be formulated as eq. 2.2. For a family of functions parameterized by θ, the goal is to minimize the empirical risk for the distortion for N pairs $(u_i, v_i), i \in 1, \ldots, N$ from our training dataset, and we can do this with standard gradient descent algorithms such as PyTorch [52] on θ given as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \frac{|d_{\mathcal{Y}}(\phi_\theta(u_i), \phi_\theta(v_i)) - d_{\mathcal{X}}(u_i, v_i)|}{d_{\mathcal{X}}(u_i, v_i)}, \phi_\theta \in \mathcal{F} \tag{2.2}$$

We denote this empirical risk minimization procedure as neural embedding (NE).

In this learning framework, we have to make two critical choices (1) which family of functions $\phi_\theta \in \mathcal{F}$ do we chose to approximate the embedding, and (2) which geometrical space $\mathcal{Y}$ do we choose to embed into.

The family of functions $\phi_\theta \in \mathcal{F}$ we choose in this paper is a family of deep neural networks. An appropriate choice of the neural network is made depending on the input format of the data.

In this paper, we will look at two types of data. As a first demonstration of embedding, we embed the MNIST [53] handwritten digit images into a 2-dimensional Euclidean space. As result, we choose convolutional neural networks (CNN) to handle image data. For the rest of the datasets, we use simulated collider events defined by a p_{T}-sorted sequence

JHEP07(2023)108

of final state particles. For the collider data, we use transformer networks with positional encoding. Since our input is a p_{T}-sorted sequence of particles where the flow of information between any particle is allowed, we believe this is a good choice and reflects the state of the art in data assimilation.

The choice of metric space for $\mathcal{Y}$ is quite flexible. Previous studies, outside of physics, have considered Euclidean spaces, Hyperbolic spaces [39], and Wasserstein spaces [45, 54]. Here, we focus on a Euclidean space with a l_2-norm, $(\mathcal{Y}, d_{\mathcal{Y}}) = (\mathbb{R}^n, l_2)$, and the Hyberbolic space defined by the Poincaré ball $(\mathcal{Y}, d_{\mathcal{Y}}) = (\mathcal{B}^n, d_p)$. Lastly, to make the training tractable, we only train on the subset of available event pairs, by randomly sampling pairs from a total set of 10^6 available events, and subsequently not considering all $O(10^{12})$ total possible pairings.

3 Datasets and neural network

Before we embark on the construction of the full NE, we would like to elaborate on the dataset construction used for these studies. Our ultimate goal with these studies is to demonstrate the broad applicability of this framework through the use of a variety of datasets including MNIST [53].

Furthermore, we will show the flexibility of the NE construction on progressively more complicated datasets leading towards a realistic dataset. Our goal with adding hierarchies of complexity is to show how NE is capable of transcending the obfuscation present from a more complicated dataset to extract the core physics features embedded within.

To study NE, we utilize hadronically decaying particles at the LHC. This dataset consists of both new physics resonances with quarks in the final state or standard model production of quarks and gluons (QCD). Quarks, and gluons at the LHC will shower into many particles eventually leading to final state hadrons. These showers are then resolved at the LHC through jet clustering algorithms, yielding jets [55, 56]. In this paper, we will focus on applying NE to a single jet. The large number of particles and complex topologies within a jet make them a difficult tool to study, and in many studies, it has been shown that jets benefit enormously from machine learning approaches.

Since jets are complicated objects, we created a series of hierarchical datasets whereby we progressively made each dataset more and more complicated. As a consequence, we developed several jet simulations that allow for the isolation of a fixed number of hidden parameters, so that we can effectively study how NE can extract the critical patterns hidden within the data. In this section we present the two main simulations used to study NE on jets, the toy jet generator, and the realistic jet generator.

3.1 Toy jet generator

In order to be able to progressively add levels of complexity in the data generation, a toy jet generator was constructed. The toy jet emulates a typical parton shower, while also storing the individual latent variables, so that we can later extract them directly, and explicitly check what information is learned.

JHEP07(2023)108

3.1.1 Jet generation

The main goal of the toy jet generator is to isolate parameters of the parton shower so that we can see how the NE organizes the embedded space. In light of this, we constructed the toy jet generator such that the masses and momentum of each splitting can be fixed, and the angles of the subsequent splittings can be sampled from a fixed prior. For these studies, we fixed the momentum to be 400 GeV, while we allowed the masses to be sampled from a fixed distribution.

We implemented two different versions of the toy jet generator, a simple version where the hard and soft splittings are distinctly different and a realistic version where the soft splittings approximate the matrix element of normal quark and gluon fragmentation.

In both types of jet generators, jets are generated with specified fixed "prongs", a fixed total number of particles, a fixed momentum, and fixed mass distributions. The hard splittings are designed to mimic the decays of a resonance. Namely, they are sampled from an angular distribution in the rest frame of the mother particle of the shower. The number of prongs within a jet defines the number of hard splittings used in the shower. To reach the total number of particles required by the generation, we continue to shower the jet with soft splitting until we reach the final multiplicity. In all cases, we force the jet construction to be in a fixed coordinate system whereby the original parton direction is at the origin of the three-dimensional vector space, and the first splitting occurs along the x-axis. Subsequent splittings are then randomized in ϕ about the particle direction, with the angle of the two particle split, θ, being defined by a sampling prior that varies depending on the jet generator type. The sampling prior for the soft and hard splittings is what defines the difference between the simplified and realistic toy jet generator. All other components of the generation remain the same.

For the hard splitting "signal" models, we force a decay chain of resonances characteristic of the top quark. In particular, we set the mass of the first "signal" parton always to be 172 GeV, and with potential decay components having a resonance of 80 GeV and 4 GeV. As a result, when we simulate two prong signal jets, we take the jet mass to be 172 GeV, with its decay components being massless quarks. For 3-prong jets, we have the 172 GeV resonance decays to a secondary resonance of 80 GeV that decays to quarks and a massless quark. For 4 prong jets, we have a final splitting with a mass of 4 GeV that then decays to two massless quarks. We do not explore jets beyond four prongs. However, we continue to decay the particles until we reach the particle multiplicity of the desired generator.

In the following subsections, we present the difference between the two toy jet generators. The only difference is the splitting angle in the rest frame. However, this has a large impact on the resulting kinematics.

3.2 Simple jet generator

For the simple jet generator scenario, we want to enhance the ability to distinguish hard scatters from soft scatters. To that extent, we define an unphysical sampling prior that is distinctly different between the hard and the soft scatters. This is achieved by drawing θ_{branch}, the splitting in the rest frame of the parton, from two distinct distributions. The

JHEP07(2023)108

hard splitting angle θ_{branch} is drawn from a normal distribution $\mathcal{N}(1/2, 0.1)$, about $\pi/2$ distribution with narrow variance. The soft splitting angle is drawn from the half-normal distribution with a wide variance of 0.1 radians.

In addition to the angle in the rest frame, we also plot the splitting angle of the first splitting, we define as

$$z_g = \frac{\max p_{\mathrm{T},1}, p_{\mathrm{T},2}}{p_{\mathrm{T},1} + p_{\mathrm{T},2}} \tag{3.1}$$

where $p_{\mathrm{T},1}, p_{\mathrm{T},2}$ denote transverse momentum of two split partons, where the transverse momentum p_{T} is the component of the momentum transverse to the beam line, $p_{\mathrm{T}} = \sqrt{p_x^2 + p_y^2}$ when the beam line direction is the z-axis.

For the simple jet generator, the splitting angle of the first splitting is shown in figure 3. In addition to the splitting angle we observe a distinct difference in the splitting fraction z_g of the first splitting (figure 2). For this sample, we generated 2M simple toy jets with prong numbers varying from 1 to 4 prongs (QCD(1p), 2p, 3p, 4p). We use 200k jets of each type for validation and testing. Plots of sample jets are shown in B.1.

3.3 Realistic jet generator

For the realistic jet generator, we follow a sampling prior characteristic of real physical decays. For the hard splitting, we sample θ_{branch} from a flat prior. The soft splittings are computed by sampling a probability distribution given by $p = \frac{1}{\theta z}$ where θ is the angle of the splitting and z is the momentum fraction of the jet. This closely approximates a typical true parton shower. Figure 3 shows the splitting energy fraction z_g for a hard and soft splitting. We observe a behavior similar to what is observed in previous studies in data [57–59]. Similarly, figure 3 shows the splitting for both hard and soft splittings, the bias towards small θ is very clear in this scenario.

For this study, we generate 1M realistic toy jets of each category (QCD(1p), 2p, 3p, 4p) and 200k for validation and testing. Plots of sample jets are shown in B.2.

3.4 Network architectures

The choice of a parametric family of functions $\phi_\theta \in \mathcal{F}$ used to approximate the embedding is important since we have to choose a family of deep neural networks with strong expressive power. We furthermore want the chosen neural network family to have properties (such as invariance and equivariance) that is appropriate for the data we have.

As a first demonstration of how to apply NE to a generic dataset, we perform training on the MNIST character recognition dataset [53]. For these collections of digit images, we use CNNs to approximate the embedding.

Next, we aim to demonstrate NE in a simplified physics-like environment by creating a "toy jet" generator that constructs jets with a fixed momentum, mass, and a fixed number of particles but a varying number of hard and soft splittings. For this dataset, we rely on transformer networks with multi-headed attention [60] applied to the p_{T}-sorted particle 4-vector dataset.

Finally, we demonstrate the NE on a set of fully simulated jets with characteristic detector resolutions using Delphes [61]. Since this dataset yields particles in a similar

JHEP07(2023)108

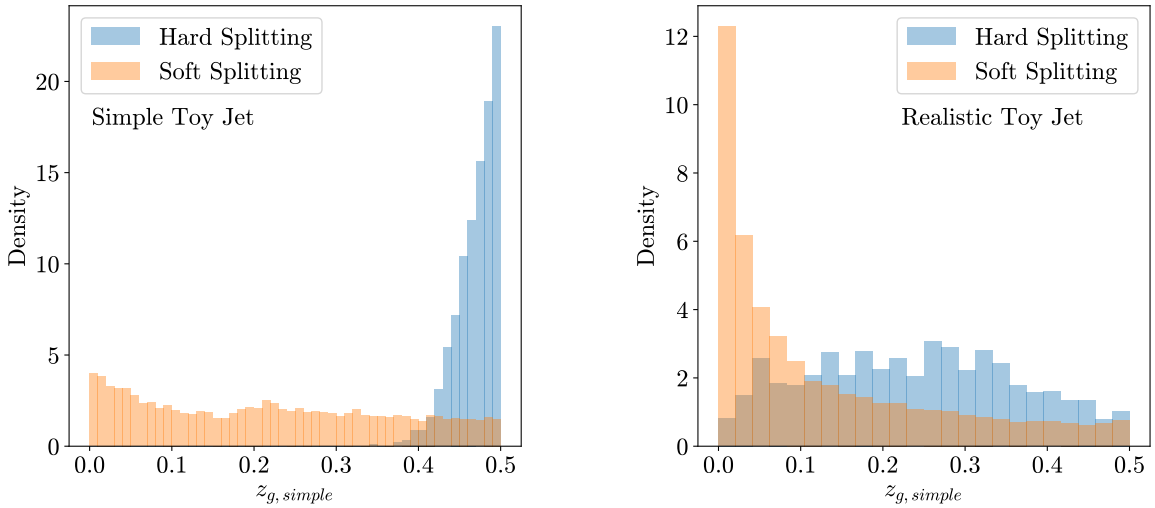

Figure 2. (Left) The distribution of parton momentum sharing variable z_g for each parton splitting for hard and soft splitting, for simple toy jet generator. (Right) The distribution of parton momentum sharing variable z_g for each parton splitting for hard and soft splitting, for realistic toy jet generator.

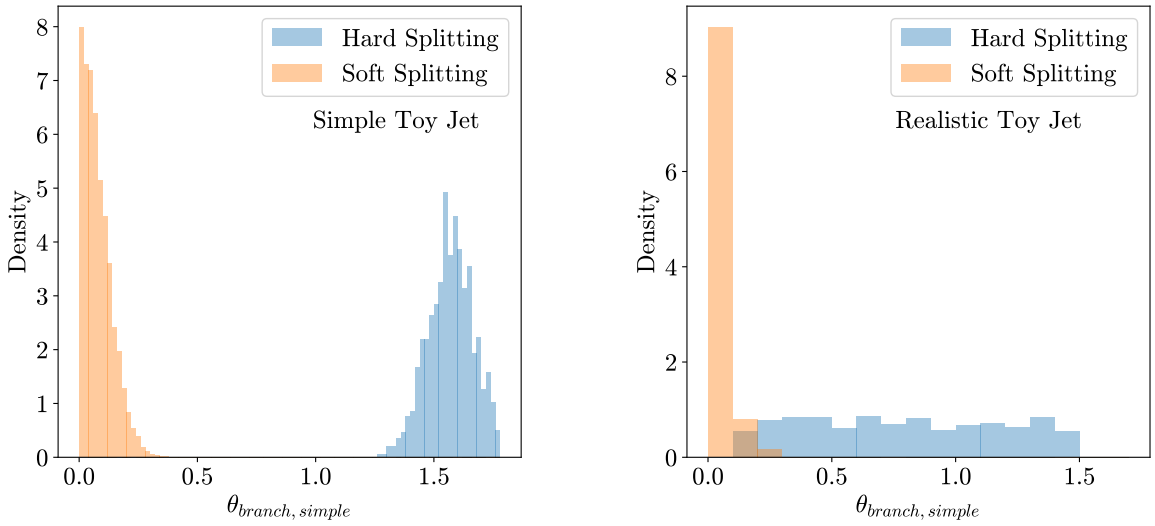

Figure 3. (Left) The distribution of splitting angle θ_{branch} for each parton splitting for hard and soft splitting, for simple toy jet generator. (Right) The distribution of parton momentum sharing variable θ_{branch} for each parton splitting for hard and soft splitting, for realistic toy jet generator.

format to the "toy jet" dataset, we employ an identical network architecture to that of the "toy jet" generator.

The details of the neural network architecture are explained in detail in appendix D.

3.5 Summary

The details of the studies are summarized in table 1.

JHEP07(2023)108

Section	Dataset	Architecture	Geometry
Appendix A	MNIST [53]	CNN	Euclidean
Section 4.1.1	Simple Toy Jets 3.2	Transformer	Euclidean
Section 4.1.2	Realistic Toy Jets 3.3	Transformer	Euclidean
Section 4.2	Simulated Jets 4.2	Transformer	Euclidean
Section 4.3	Simulated Jets 4.2	Transformer	Hyperbolic

Table 1. Summary of datasets, network architecture and geometry of the embedded space presented in this paper.

4 Experiment

4.1 Toy jets

To test the NE on a more complicated dataset, we consider varying sets of progressively more complicated datasets using the toy jet generator. With each dataset, we take the first K highest transverse momentum constituents, each with $(p_{\mathrm{T}}, \eta, \phi)$ information yielding a mapping from $\mathbb{R}^{3K}$ to the lower dimensional space. For the toy jets generator and future particle based studies, we take $K = 16$, and our embedding function thus becomes:

$$\phi_{\theta,\mathrm{Transformer}} : (\mathcal{X}_{\mathrm{jets}} \subset \mathbb{R}^{48}, d_{\mathrm{EMD}}) \to (\mathbb{R}^2, l_2) \tag{4.1}$$

4.1.1 Simple toy jets

For toy jets, we train on 1-prong(QCD) jets and resonant 2-prong and 3-prong jets with a fixed mass at 172 GeV and transverse momentum 400 GeV, and we test on different 1-prong, 2-prong, 3-prong jets, with 4-prong generated jets added as well. We train the transformer model on 2M jets for each type of jet and validate on 2k jets each. The 4-prong jets are reserved just for prediction to see if the embedding can be extrapolated to jets drawn from the toy jet model with different parameters compared to what was shown in the training.

The distribution of pairwise energy mover's distance(EMD) for simple toy jets is shown in figure 4.

From the observed EMD, we can start to infer the expected shape of the embedded space. Since the distances between 1-prong jets are wide, we expect that they would not form a closely grouped cluster in the embedded space. Also, since EMD distribution between 1-prong and 3-prong is larger than between 1-prong and 2-prong, we can expect that the distance between clusters of 1-prong and 3-prong jets would be larger than 1-prong and 2-prong. Furthermore, we expect that 3-prong jets will form a small cluster since the EMD between 3-prong jets are small. We see that 3-prong and 4-prong jets have a similar distribution, and we can guess that 3- and 4-prong will form close clusters while 2-prong and 1-prong(QCD) jets will form separate distinct clusters.

The result of the NE is shown in figure 5. We observe that all the points form a small cluster according to their pronginess, with a small diffuse shape for 2-prong and 1-prong(QCD) jets, and 3-prong and 4-prong jets get mapped to an almost identical region. The embedding shows a simple structure and we see that it reflects the raw EMD distribution in figure 4 well.

JHEP07(2023)108

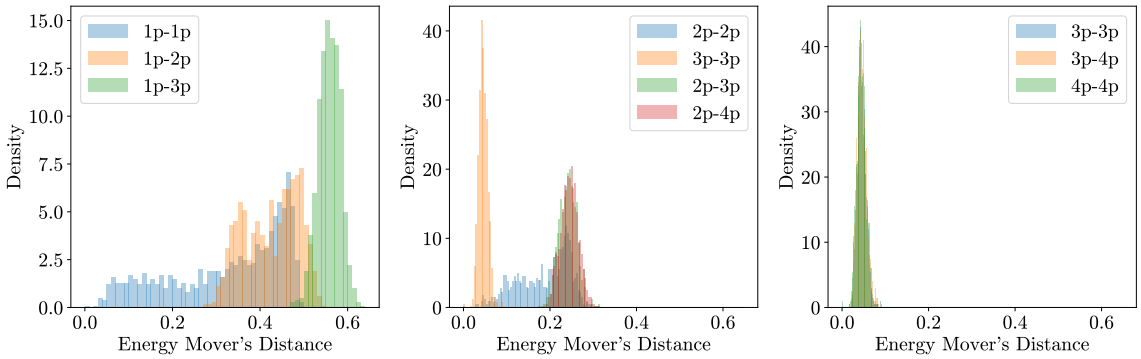

Figure 4. (Left) The distribution of energy mover's distance (EMD) between QCD jets and two-prong, three-prong, and four-prong jets. (Middle) The distribution of energy mover's distance between QCD jets and two-prong, three-prong, and four-prong jets. (Right) The distribution of the parton momentum sharing variable z_g for each parton splitting for hard and soft splitting for the realistic toy jet generator.

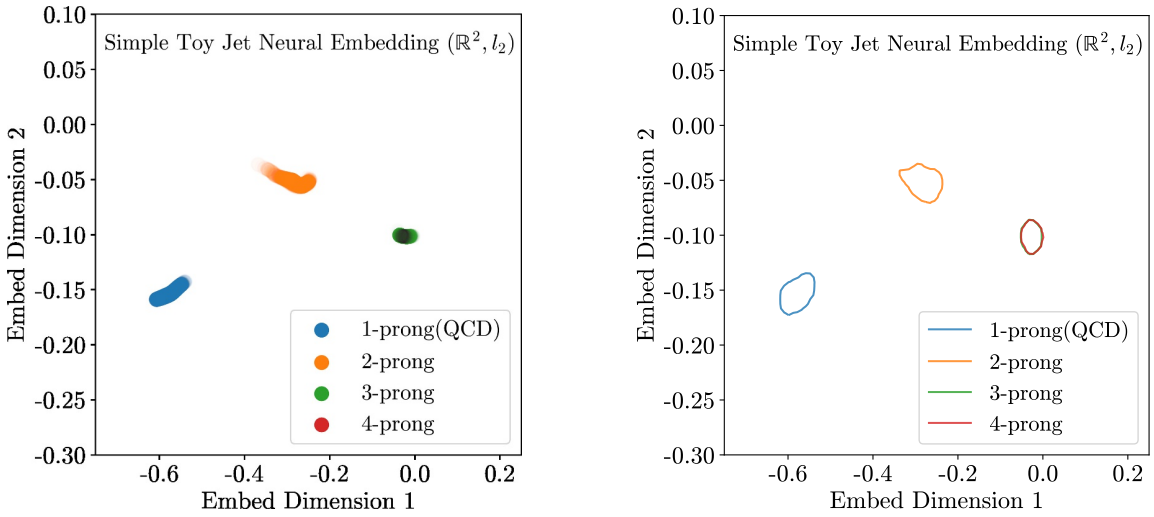

Figure 5. (Left) The embedding of realistic toy jets for 1-prong(QCD), 2-prong, 3-prong, and 4-prong jets. The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5 and 0.8.

4.1.2 Realistic toy jets

The distribution of pairwise energy mover's distance for realistic toy jets is shown in figure 6. The NE is shown in figure 7. We observe a similar trend to that of the simple toy jets. We see that 1-prong(QCD) jets form a cluster on their own, and 2-prong, 3-prong, and 4-prong forming clusters around 1-prong jets.

We can further investigate what is learned in these embeddings by choosing different regions of the embedded space and looking at the first splitting angle θ_{branch} in the rest frame of the jets. We see in figure 8 that in the case of realistic toy jets, the first splitting angle is learned very well by the embedding, and it uses this feature to start organizing the dataset.

JHEP07(2023)108

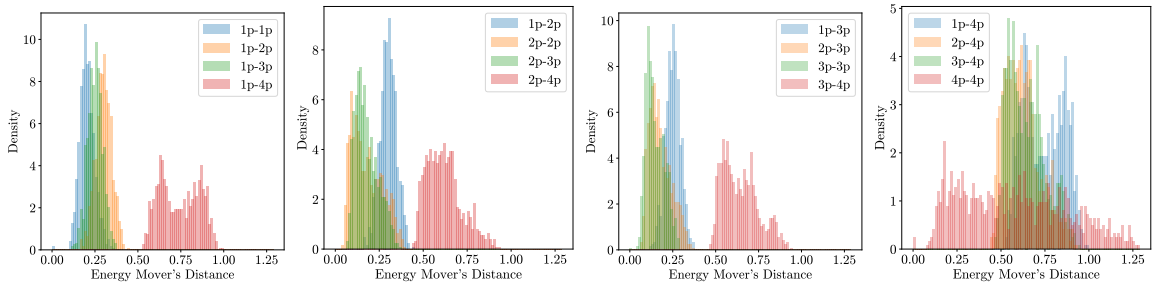

Figure 6. (Left) The distribution of energy mover's distance between QCD jets and other jets. (Middle Left) The distribution of energy mover's distance between 2-prong jets and other jets. (Middle Right) The distribution of energy mover's distance between 3-prong jets and other jets. (Right) The distribution of energy mover's distance between four prong jets and other jets.

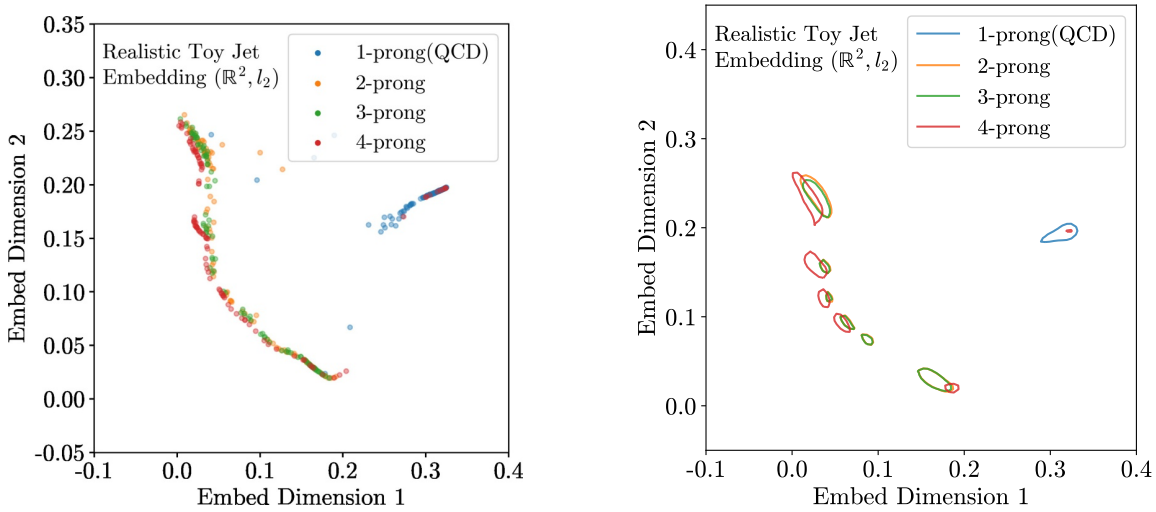

Figure 7. (Left) The embedding of realistic toy jets for 1-prong(QCD), 2-prong, 3-prong, and 4-prong jets. (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5.

4.2 Simulated jets

Finally, we consider a set of simulated true jets. For these events, we rely on events generated with MADGRAPH 5 [62], showered with Pythia 8.1 [63, 64] and then smeared using Delphes 3 [61]. Jets were clustered with FastJet 3 [65, 66]. In this scenario, we generate events from a wide variety of different topologies consisting of QCD, 2-prong, and 3-prong jets of varying masses. From this dataset, we perform a single training on all of these topologies to construct the embedded space. In all cases, we train on a single jet.

Table 2 summarizes the different samples utilized for the training and testing of the embedded space. To demonstrate the robustness of the construction, we eliminated a variety of mass points in the training, along with all 4-prong samples. However, we still use these samples in the testing of the space.

The testing is done in two different datasets, the interpolation, and the extrapolation dataset. The extrapolation set is a collection of jets of types not shown in training. This

JHEP07(2023)108

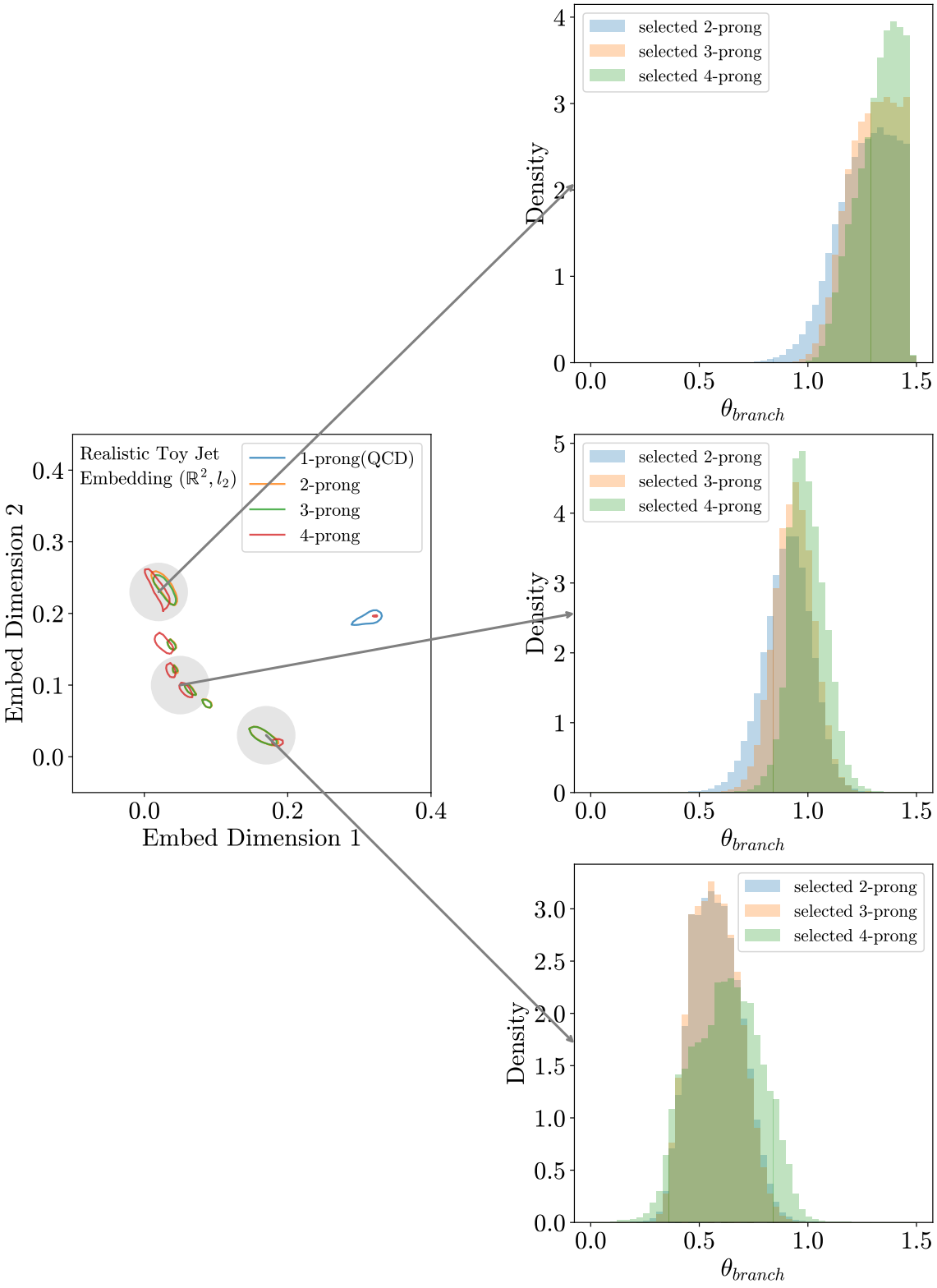

Figure 8. Selecting three different regions of the embedded space for realistic toy jets and plotting the first splitting angle θ_{branch} of the jets that fall into each of those regions.

JHEP07(2023)108

Events	Jet Pronginess	Jet Mass	Used in Training	Test Dataset
$X \to YY'$	2	25 GeV	✗	Extrapolation
$X \to YY'$	2	80 GeV	✓	Interpolation
$X \to YY'$	2	170 GeV	✗	Extrapolation
$X \to YY'$	2	400 GeV	✓	Interpolation
$W' \to B'T$	3	25 GeV	✗	Extrapolation
$W' \to B'T$	3	80 GeV	✓	Interpolation
$W' \to B'T$	3	170 GeV	✗	Extrapolation
$W' \to B'T$	3	400 GeV	✓	Interpolation
$V_{kk} \to (VV)V$	4	170 GeV	✗	Extrapolation
$V_{kk} \to (VV)V$	4	400 GeV	✗	Extrapolation

Table 2. Summary of simulated jet samples.

dataset includes 4-prong jets and 2-prong and 3-prong jets with masses eliminated from the training. The interpolation set is a collection of jets of types shown in training that shows the interpolation capability of these methods, such as 2-prong and 3-prong jets with masses shown in training that were held out for testing and have no overlap with the training dataset. For all jet types, one million jets were used in training, 200k jets each for validation and testing, and 10k jets were used for presenting the results.

The QCD jets are constructed from events generated with MADGRAPH 5 [62] and showered with PYTHIA 8 [63], with an HT range of 1500 to 2000 GeV. A pre-selection on the jets is applied so that the p_{T} of the jets are greater than 300 GeV. The two prong jets are generated from $X \to YY'$ process, with Y and Y' masses 25,80, 170, 400 GeV. We use masses 80, 400 GeV for training and 25, 170 GeV for testing in the interpolated dataset. The 3-prong jets are generated from $W' \to B'T$ events, with both W' and Top quark mass varied, both decaying to a 3-prong. As with the 2-prong sample masses, 80, 400 GeV are used for the training and 25 and 170 GeV for testing. Lastly, one million 3-prong jets are used for training, and 200k jets for validation and testing, just as in the 2-prong case. For the 4-prong jets, two mass points 170 and 400 GeV are generated from triboson events $V_{kk} \to (VV)V$, where two bosons, both decaying 2-prong, get clustered in the same jet. These jets weren't shown in the training at all and constitute the extrapolation test dataset. For the respective labels, we only take jets with explicitly 2,3,or 4 prongs for the NE.

The OT-based distances between the jets are very sensitive to preprocessing. As a result, we apply a more complex pre-processing scheme. First, the jets are centered so that the jet η and ϕ are centered to the origin. Then the jet constituents are rotated with respect to the origin so that two most energetic components are aligned along the y axis of the $(\Delta\eta = \eta_i - \eta_{\mathrm{jet}}, \Delta\phi = \phi_{\mathrm{jet}})$ coordinate system. Finally, the jets are flipped so that the maximum sum of energy of constituents is placed in the first quadrant in the $(\Delta\eta, \Delta\phi)$ plane. Examples of such jets are shown in appendix B.

Figure 9 shows the EMD for the jets, we observe that the energy mover's distance is more sensitive to varying the mass compared to varying the pronginess of the jets.

JHEP07(2023)108

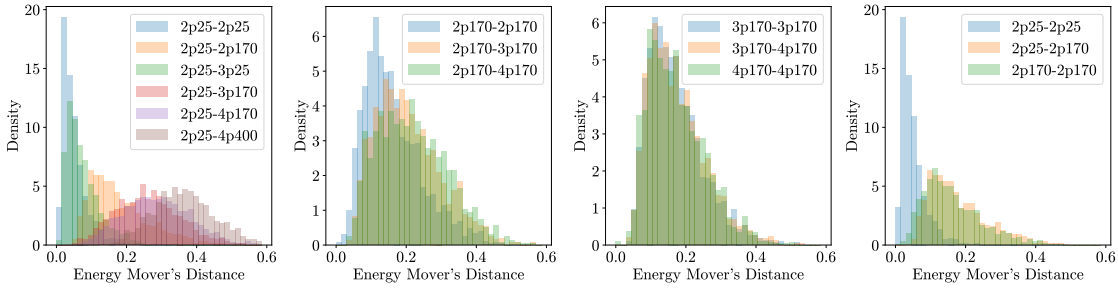

Figure 9. (Left) The distribution of energy mover's distance(EMD) between 2-prong jets with mass 25 GeV jets and other jets. (Middle Left) The distribution of energy mover's distance between different jets with fixed mass of 170 GeV. (Middle Right) The distribution of energy mover's distance between different jets with fixed mass of 170 GeV. (Right) The distribution of energy mover's distance between 2-prong jets with different masses.

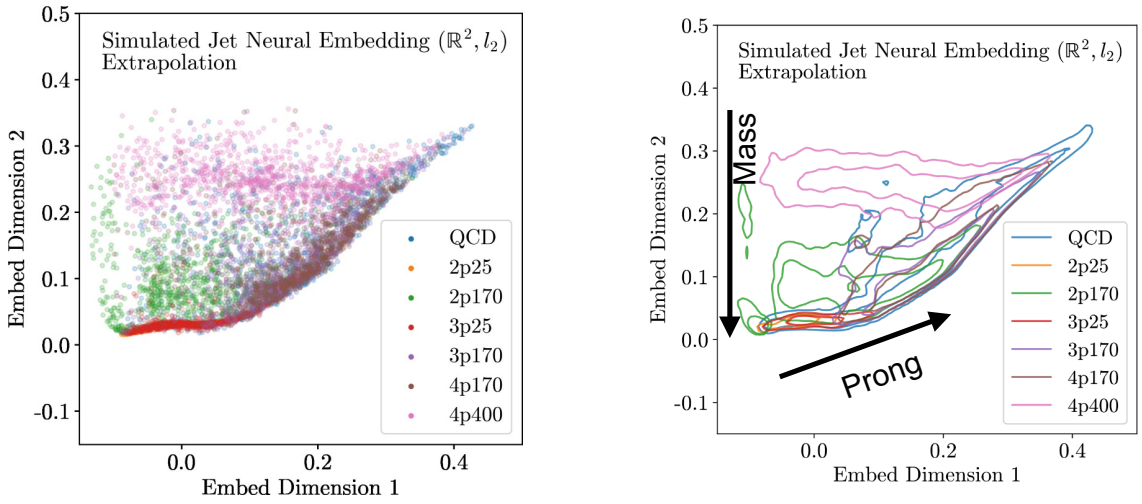

Figure 10. (Left) The embedding (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to CDF value 0.5 and 0.8. Labels indicate the pronginess and the mass of the jet. For instance, 2p25 indicates 2-prong jets with mass 25 GeV, generated from $X \to YY'$ model.

Figure 10 shows the result of the NE applied to the extrapolation dataset. We observe a strong grouping according to the jet masses with a general progression towards smaller masses as one goes to smaller values on the y-axis. We also observe a trend towards lower prongs as one moves closer to the origin in the embedded space. As a consequence of these trends, we find 2-prong and 3-prong jets with 25 GeV mass get grouped in the bottom left corner, and 2-prong, 3-prong, and 4-prong jets with masses 170 GeV get grouped above the 25 GeV mass group. Above the 170 GeV group, 4-prong jets with 400 GeV mass are placed.

Figure 11 shows the result of the NE on the interpolated datasets. With these datapoints, we again observe that there is an even stronger grouping based on the mass, QCD with all the 80 GeV jets getting mapped to the bottom half of the space, and all the 400 GeV jets getting mapped to the top half of the space.

JHEP07(2023)108

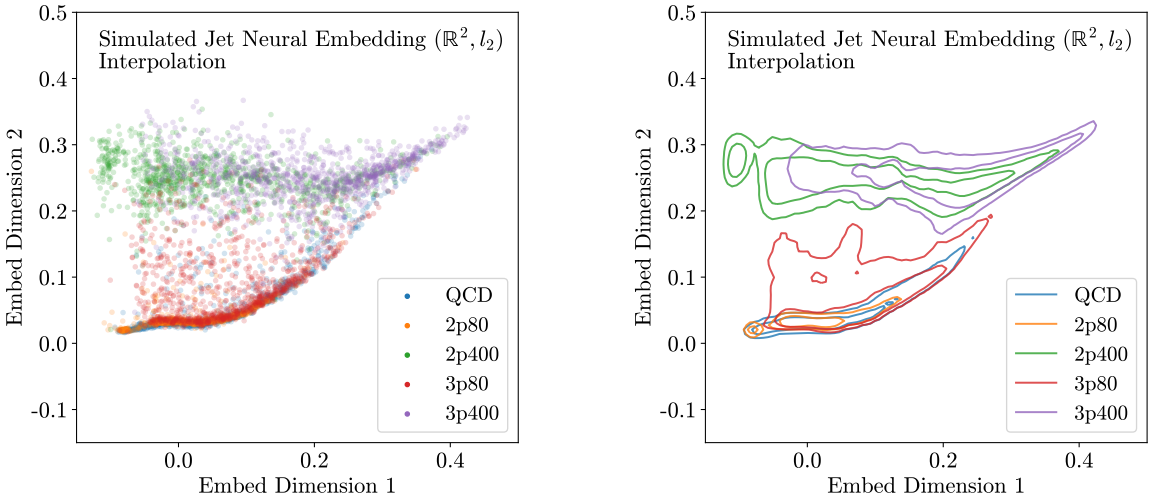

Figure 11. (Left) The embedding (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5 and 0.8.

In order to further understand what is learned by the embedding functions, in figure 11, we look at different regions of the NE and plot physical observables. We choose different regions of the Euclidean space and plot the histograms of subjettiness variables $\tau_{21}, \tau_{32}, \tau_{43}$. We find a strong correlation among selected subjettiness variables even for QCD jets. In particular, the pronginess consistently goes down to lower values as one progresses towards the origin of the embedded space. Already, we can see that with this embedded space, we can start to classify jets into distinct regions based on their features and their generated properties.

4.3 Hyperbolic embedding of jets

In the above section, we performed a NE into a Euclidean space. In this section, we present results on embedding into non-Euclidean, Hyperbolic spaces. For this paper, we primarily study embedding into the two-dimensional Poincaré disks. It is well known that tree-like structure embeds well into the Poincaré disk since the distance gets stretched close to the boundary of Poincaré disks [39, 67–69]. Thus it is interesting to view jets as tree structures and embed them into Poincaré disks.

Hyperbolic space has a physical analog and is used to define the motion of high momentum objects within Minkowski space. The jet is a high momentum object that decays into a spray of high momentum particles. As a result, its decay products and the forces causing the decay undergo relativistic motion giving rise to a curved hyperbolic geometry. Lastly, since it is impossible to embed non-Euclidean manifolds into Euclidean space without a big distortion, alternative geometries are well motivated extensions of NE and should, in general, be pursued.

With the same metric space and EMD distribution as in figure 9 and the same training set, we learn the function in eq. 4.2, the embedding into the Poincaré disk($\mathcal{B}^2$) denoted as

$$\phi_{\theta,\text{Transformer}} : (\mathcal{X}_{\text{jets}} \subset \mathbb{R}^{48}, d_{\text{EMD}}) \to (\mathcal{B}^2, d_p)\,, \tag{4.2}$$

JHEP07(2023)108

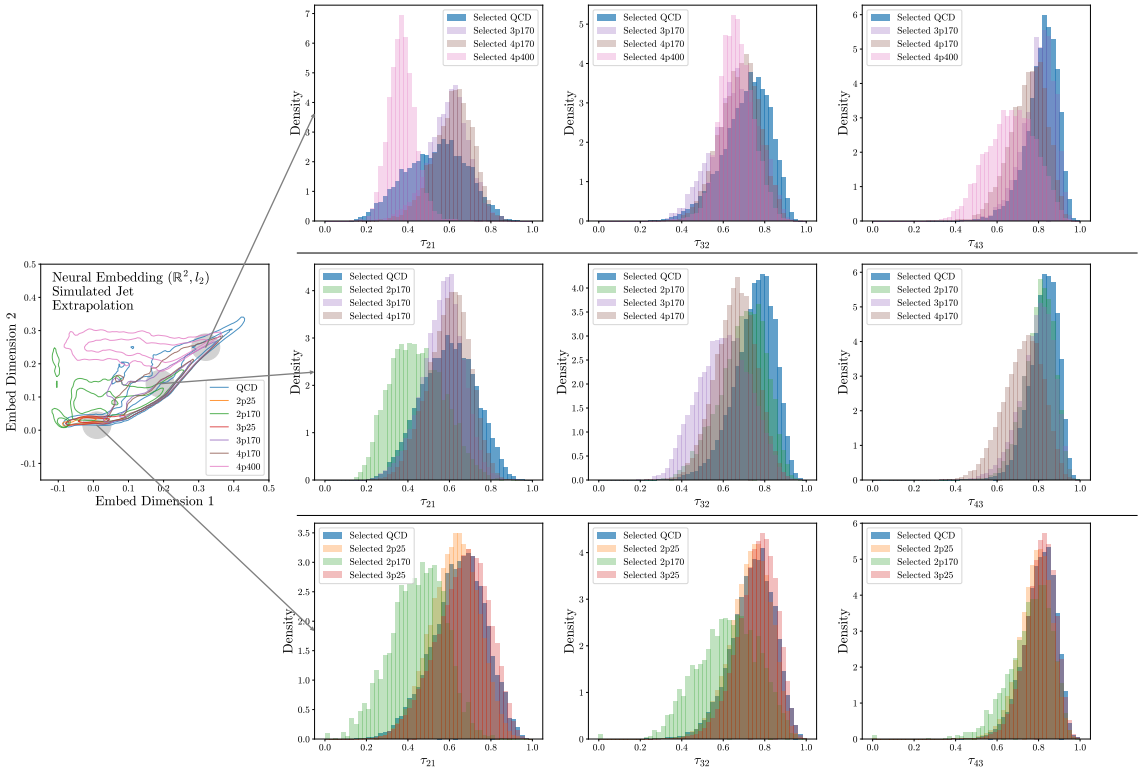

Figure 12. (Left) The embedding (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5.

where the metric distance on the disk d_p is given by Eq 4.3.

$$d_p(x, y) = \operatorname{arcosh}\left(1 + 2 \cdot \frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right) \tag{4.3}$$

As in the Euclidean embedding cases in section 4.2, we present the results in two different cases on the same extrapolation and interpolation prediction datasets. The result of the embedding into Poincaré disks for the extrapolation dataset is shown in figure 13. There is a clear trend towards heavier objects as one goes downwards along the y-axis. Additionally, we observe a trend toward more prongs as one moves downwards and to the left. As a consequence, we observe a strong grouping based on the mass of the jets. The 25 GeV mass jets are grouped together and also 170 GeV jets get mapped to the same regions. Finally, The 400 GeV jets form a cluster of their own. This grouping based on mass seems to be stronger compared to the Euclidean embedding case.

To further see whether the latent structure is learned by the embedding, figure 15 shows $n-$subjettiness variables of jets that get mapped to different regions of the embedded space. We can see that the distributions of n-subjettiness are highly correlated within the local regions of the space, even stronger than Euclidean embedding, and we conclude that interpretability is better for embedding into hyperbolic spaces compared to Euclidean spaces.

The result of the embedding into Poincaré disks for the interpolation dataset is shown in figure 14; the behavior is similar to that of the extrapolation dataset. Overall, we

JHEP07(2023)108

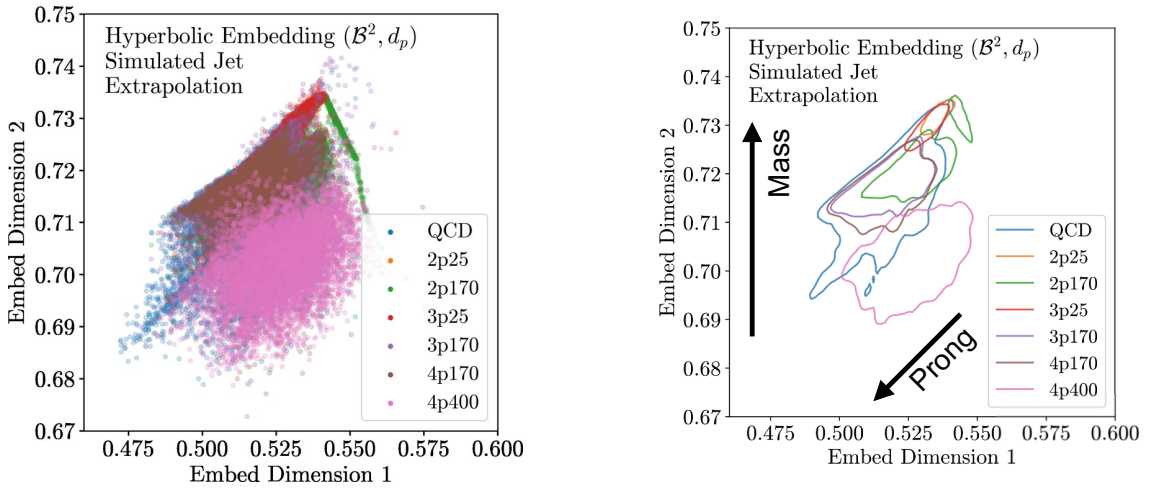

Figure 13. (Left) Scatterplot of Hyperbolic embedding of simulated jets into Poincaré disks $(\mathcal{B}^2, d_p)$ (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5 and 0.8. Labels indicate the pronginess and the mass of the jet. For instance, 2p25 indicates 2-prong jets with mass 25 GeV, generated from $X \to YY'$ model.

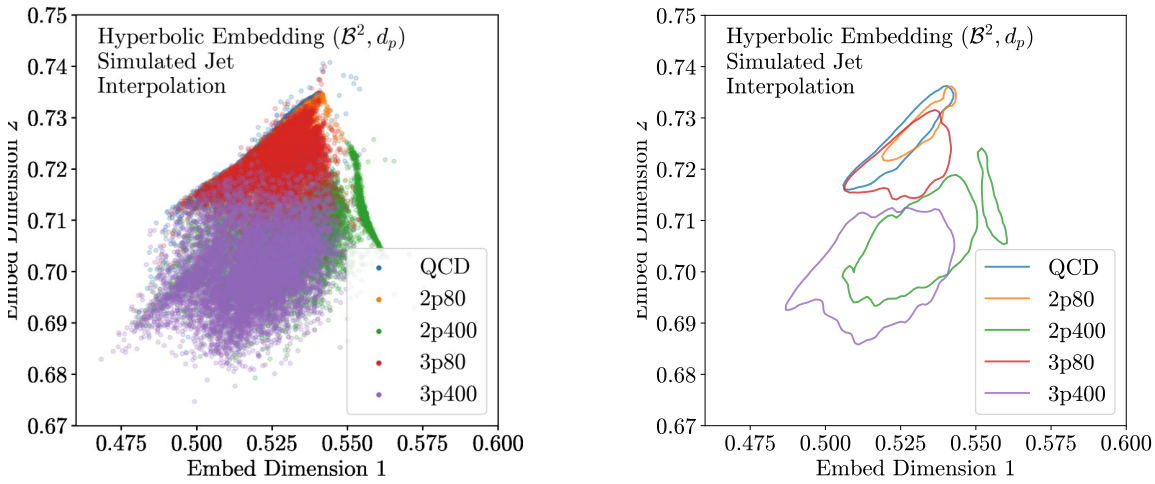

Figure 14. (Left) Scatterplot of Hyperbolic embedding of simulated jets into Poincaré disks $(\mathcal{B}^2, d_p)$ (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5 and 0.8. Labels indicate the pronginess and the mass of the jet. For instance, 2p25 indicates 2-prong jets with mass 25 GeV, generated from $X \to YY'$ model.

observe there is a strong grouping of objects within this space. This implies the NE has "self-organized" the dataset along the EMD criterion, yielding a physically interpretable space consistent with that of EMD.

As we can see from the good separation between different types of events for both Euclidean and Hyperbolic embeddings, we see that these NE can be used to flag interesting anomalies by looking for events within this space. However, a related application that we can also perform is to quantify the effectiveness of anomaly detection algorithms in detecting interesting regions of phase space. This exciting application is covered in 5.

JHEP07(2023)108

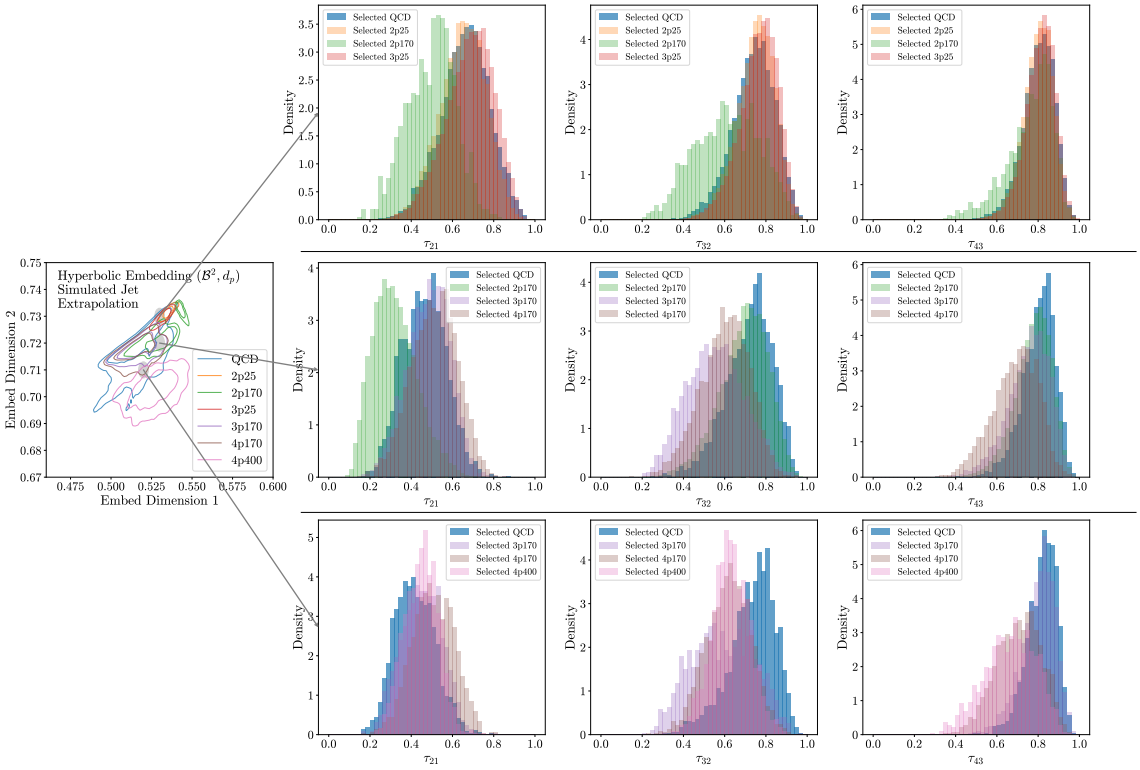

Figure 15. (Left) The embedding (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.5.

4.4 Empirical estimation of distortion

In addition to a physical clustering of events, we can also look to see how well the embedded space preserves the embedded metric within the space [70]. With the given embedding $\phi : (\mathcal{X}, d_{\mathcal{X}}) \to (\mathcal{Y}, d_{\mathcal{Y}})$, we define the distortion as the ratio of measured EMD after NE compared to the true EMD, given by

$$\rho_{\phi}(u, v) = \frac{d_{\mathcal{Y}}(\phi(u), \phi(v))}{d_{\mathcal{X}}(u, v)} \tag{4.4}$$

The distortion measures how far the new distances $d_{\mathcal{Y}}(\phi(u), \phi(v))$ between the embedded points deviate from the original distances $d_{\mathcal{X}}(u, v)$ for an arbitrary pair of points $(u, v) \in \mathcal{X}$.

To quantify the level of distortion, we condense the distortion response and variation to two numbers the mean, μ, and the standard deviation, σ, of the distortion. Where the variation for any embedding ϕ can be written in terms of the normalized ratio of the distances, $\tilde{\rho_{\phi}}(u, v)$, given by

$$\tilde{\rho_{\phi}}(u, v) = \frac{M \rho_{\phi}(u, v)}{\sum_{i=1}^{M} \rho_{\phi}(u_i, v_i)}, \tag{4.5}$$

where the summation is done for all pairs in the test dataset and M denoting the total number of pairs. The distortion variation σ-distortion is defined as, letting $\Pi = P \times P$,

JHEP07(2023)108

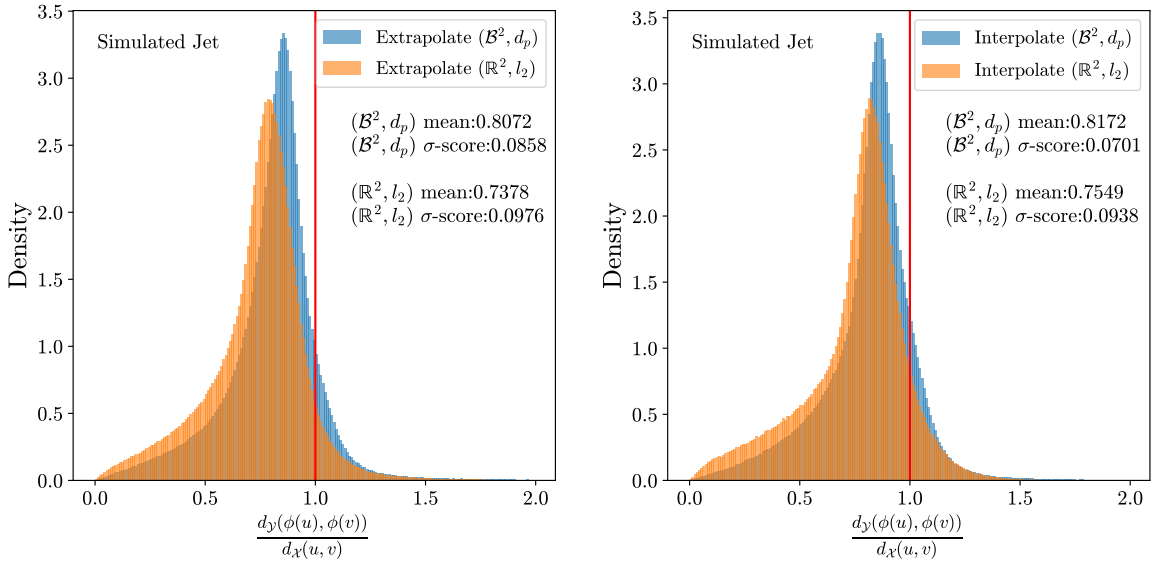

Figure 16. (Left) Histogram of pairwise ratios on the extrapolation set. (Right) Histogram of pairwise ratios on the interpolation set.

where P is a distribution over $\mathcal{X}$,

$$\sigma\text{−distortion} = \mathbb{E}_{\Pi}(\tilde{\rho_{\phi}}(u, v) - 1)^2 \,. \tag{4.6}$$

When P is a uniform probability distribution over $\mathcal{X}$, then σ-distortion measures the variance of the distribution of the normalized ratio of distances, $\tilde{\rho_{\phi}}(u, v)$. For the NE, we aim for an embedding with low distortion and a small σ-distortion.

To test the embedding, we compute the distortion on both the extrapolation and interpolation test datasets. The result is shown in figure 16. Firstly, we verify that low distortion and low σ-score embedding is achievable in very low dimensions, in two-dimensional Euclidean and Hyperbolic spaces. Indeed, we see a sharp distribution that peaks near 1, the ideal value. Comparing the performance of embeddings on the interpolation and extrapolation datasets, we see that the performance of the two datasets is very similar. We can conclude that the embedding is surprisingly good and has good extrapolation capabilities.

Figure 17 shows pairwise ratios and σ-distortion as a function of original metric distance. The performance is the worst (pairwise ratios deviate furthest from the ideal value 1 and σ-distortion is the largest) near extreme original metric distances 0 and 1.2, where there is less training data available. However, even with significantly less training data available, the learned neural embedding functions perform reasonably well with pairwise ratios on the same scale as the best-achieved values.

We also see that Hyperbolic embedding performs better than Euclidean on both the interpolation and extrapolation datasets. For both datasets, the Hyperbolic embedding gets the mean of pairwise ratios closer to one and achieves a lower σ-distortion. The improvement is even more striking for extreme values of the original metric distances. Looking at figure 17, we see that for both interpolation and extrapolation cases, near the largest values

JHEP07(2023)108

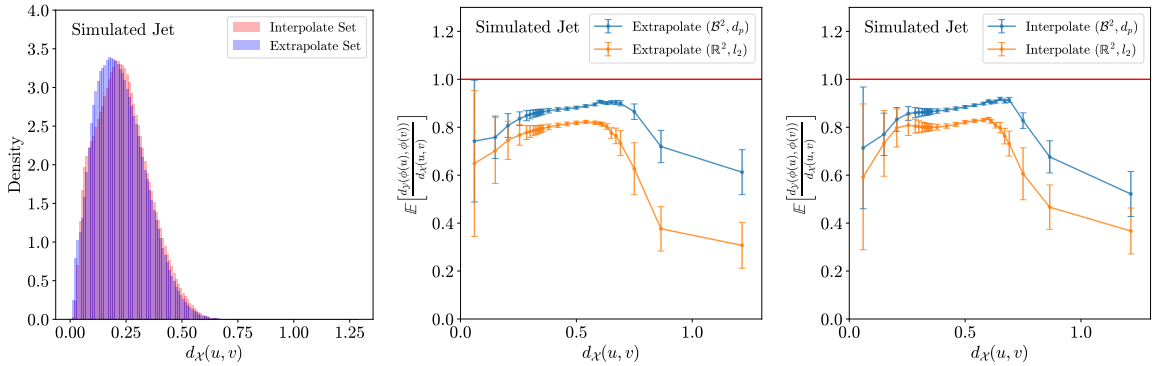

Figure 17. (Left) Histogram of original metric (EMD) distribution for interpolation and extrapolation set. (Middle) The pairwise ratios plotted as a function of the original metric distance, for the extrapolation set. The points are the mean of pairwise ratios in each bin of original metric distance, and the error bars are the σ-distortion in each bin. (Right) The pairwise ratios plotted as a function of the original metric distance, for the interpolation set.

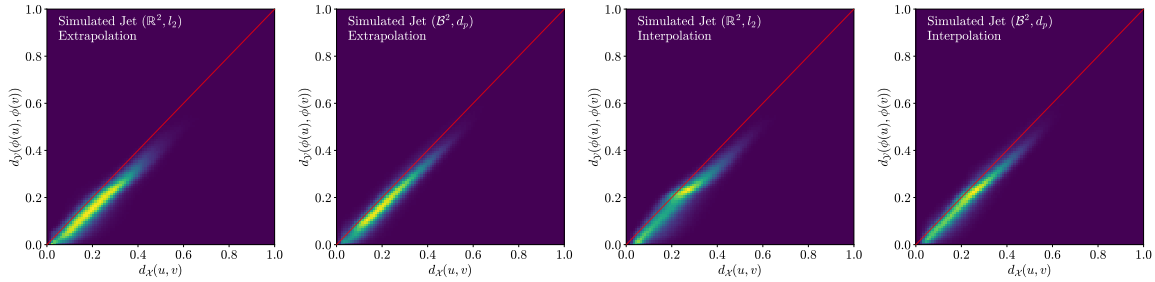

Figure 18. (Left) Correlation between the distances in the original metric space and the distances in the embedded space, for extrapolation dataset into Euclidean embedded space. (Middle Left) The same plot for extrapolation dataset into Hyperbolic embedded space. (Middle Right) The same plot for interpolation dataset into Euclidean embedded space. (Right) The same plot for interpolation dataset into Hyperbolic embedded space.

of the original metric distances Hyperbolic embedding significantly outperforms Euclidean embedding.

Lastly, in figure 18 we visualize the relation between the distances in the original metric space and the distances in the embedded space. We see that they are very highly correlated and almost fall in $y = x$ line, as we expect, for all cases.

Overall, considering that we are working in two dimensions, a higher dimensional embedding would likely perform even better. In higher dimensional embedded spaces, Hyperbolic has the potential to further outperform Euclidean embedding due to inherent geometry.

4.5 Comparison with other manifold learning methods

There are many well-studied methods to embed data into Euclidean spaces. In addition to NE, t-SNE and UMAP are alternative approaches capable of embedding the original data manifold into the lower-dimensional Euclidean space. Some exploration has been done to

JHEP07(2023)108

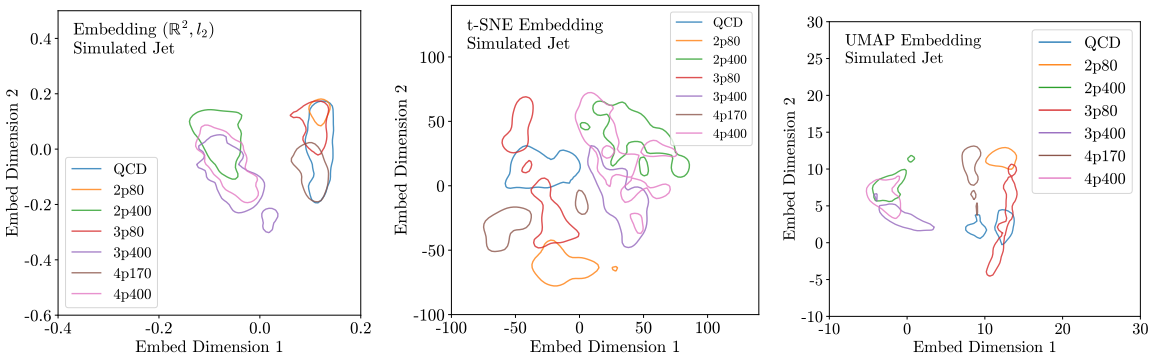

Figure 19. (Left) The neural embedding we propose smoothed with kernel density estimator, with contour lines corresponding to the CDF value 0.4. (Middle) t-SNE result on the same dataset (Right) UMAP result on the same dataset.

use these embeddings on jets. In particular, t-SNE has been used for embedding jets in previous works [71].

Both t-SNE and UMAP are fundamentally different methods from NE and suffer from limitations in their applicability. First, most out-of-the-box embedding methods such as t-SNE [37] and UMAP [38] deal with embedding a fixed number of points into the space by training and embedding on the same datasets. This means that we don't have the ability to perform parallel evaluation, which limits the scalability when compared with NE. Secondly, since t-SNE and UMAP learn the specific relationship in a given dataset yielding an unphysical metric. The mapping cannot be applied to alternative datasets. NE avoids this problem and we have already demonstrated robust performance on a 4-pronged dataset not used within the training.

Thirdly, t-SNE focuses on preserving the local structure of a dataset at the cost of inducing severe global distortions. As a result, the Euclidean distance in t-SNE space is hard to interpret since t-SNE does not preserve distance and global structure.

Both methods are tested with our setup in figure 19 along with the neural embedding method we propose on a small subset of the jet interpolation dataset. Although there is some benefit, low distortion for EMD is not guaranteed at all for UMAP and t-SNE limiting our ability physically interpret the space. We also argue that by looking at figure 19, the neural embedding method offers the best interpretability since it shows a characteristic ordering of mass and pronginess.

5 Anomaly quantification

Following the construction of the embedded space, we can perform a variety of explorations to understand what has been learned in the space. In this section, we look at how we can use the NE space to define a diversity metric for the scope of signatures that anomaly detection has identified.

When trying to quantify the effectiveness of an anomaly detection algorithm, the conventional metrics fall short when the target is unknown. More specifically, in model agnostic

JHEP07(2023)108

searches at colliders, we can't a priori know the beyond the standard model (BSM) physics signal. The conventional metrics of evaluation, such as comparing ROC curves and the significance of the extracted signal, often do not tell the full story since the performance of the algorithm depends critically on the chosen evaluation dataset. Moreover, the significance of a single dataset does not characterize the ability of the anomaly detection algorithm to find unexpected signals.Good performance of the algorithm on one test dataset does not guarantee performance on some very different collections of events. Currently, there is no clear way to handle this notion of "wideness" of the search capability.

With the embedded space, we propose a new metric to indicate the coverage of phase space of a single algorithm. Since the embedded space compresses high dimensional objects into a low dimensional space of physical features, we can utilize the notion of the volume in the embedded space as a way to define algorithm coverage. When the embedded space is a Euclidean space, the volume is straightforward to calculate, and in two-dimensional Euclidean space, this equates to the area within the embedded space.

To understand how area coverage characterizes the wideness of an anomaly search, we introduce the idea of area adjusted ROC curve. To compute the area adjusted ROC curve, we first prepare a signal evaluation ensemble that consists of a wide variety of event topologies so that the phase space over which we wish to compare the two algorithms is broadly covered. With the evaluation ensemble and the background dataset, we then evaluate how the chosen algorithm covers the embedded space of the signal ensemble when compared with the background. For each point on the regular ROC curve, we map the selected signal points that pass the threshold for the true positive rate (TPR) of our chosen algorithm to the embedded space and calculate the ratio of the total embedded space area covered by the selected points; this yields the Area TPR defined in eq. 5.1.

$$\text{Area TPR} = \epsilon_{\text{sig, area}} = \frac{\text{Selected Signal Area}}{\text{Total Signal Ensemble Area}} \tag{5.1}$$

Similarly, we map the selected QCD background points to the embedded space and calculate the ratio of selected background points to the total QCD area. This procedure defines the Area FPR defined in eq. 5.2, and it tells us the efficiency of the area of QCD rejection, defined as

$$\text{Area FPR} = 1 - \epsilon_{\text{bkg, area}} = \frac{\text{Selected Background Area}}{\text{Total QCD Area}} \tag{5.2}$$

With Area TPR and Area FPR, we can construct area adjusted ROC curve. The roc curve is adjusted based on whether the algorithm casts a wide net or only looks at the narrow region of the phase space.

We diagrammatically show how this adjusted ROC curve is made in practice in figure 21, for a supervised learning algorithm constructed from an MLP network training QCD vs 2-prong jet with a secondary mass of 170 GeV.

In figure 21, we see how a point(a red star point) in the normal ROC curve gets translated to a point on area adjusted ROC curve. For a given TPR, we compute the area the selected points cover compared to the full area coverage of our ensemble set. Similarly,

JHEP07(2023)108

for the background, we compute the fraction of the area of the embedded space volume that gets rejected by the chosen algorithm.

The area ratio is calculated for the Area TPR and Area FPR by dividing the embedded space into grids and counting the number of bins that has data points above a certain threshold, which we call a threshold parameter. By default, the regions with counts more than 3 are considered, but the ROC curve is stable regarding the choice of the threshold parameter. The stability of the area-adjusted ROC curve regarding the choice of this threshold parameter is discussed in section C, figure 32.

This can be understood as a measure of the total phase space of events, represented in the form of the NE. The area coverage is the portion of the total phase space volume that the algorithm covers.

Careful consideration of these ROC curves should be taken into account since there is a dependence of area adjusted ROC curve on the selection of this evaluation dataset. To make this approach as general as possible for anomaly detection, we choose a dataset where the area it spans in the embedded space is wide. Later we show in figure 23 that as long as the evaluation ensemble covers the search space well, the dependence on this set of an ensemble is very small.

To see the usefulness of this anomaly quantification method, we compare two different anomaly detection algorithms, a fully supervised algorithm (MLP) trained to do QCD vs 2-prong jet with mass 170 GeV, and an unsupervised algorithm comprising of an autoencoder trained on just QCD background. The evaluation dataset was the ensemble dataset constructed by mixing QCD, two-prong jets with masses 25,80 170,400 GeV, three-prong jets with masses 80, 170, 400 GeV, and four-prong jets with masses 170, 400 GeV with equal proportion. The evauation dataset was chosen to measure sensitivity to a broad spectrum of signal.

Figure 20 and figure 21 shows the construction of area adjusted ROC curve for these two different algorithms.

Figure 22 compares the normal ROC and area-adjusted ROC directly for these two algorithms. It aligns with our intuition that since MLP is hyper-optimized to do well on one specific dataset, it has lower search capability and focuses on the narrow region of the phase space. Therefore, we see that even though the MLP algorithm seems to be doing fine on the regular ROC curve, the area adjusted ROC curve reveals that the AE algorithm is more efficient at searching the wider area in the embedded space, which, here, acts as a proxy for the phase space.

Another benefit of this procedure is that we can visualize which region of embedded space each algorithm searches and compare this between different methods. By comparing the colored area of the selected points in the embedded space in figure 20 and figure 21, we can observe that for the same TPR working point, two algorithms presented in figure 20 and figure 21 choose complementary regions of the embedded space.

Finally, we show in figure 23 that area adjusted ROC curves reduce the dependence on the test dataset we choose. Ensemble 1 dataset was constructed by mixing QCD, two-prong jets with masses 25, 170 GeV, three-prong jets with masses 25, 170 GeV, and four-prong jets with masses 170 and 400 GeV , and Ensemble 2 was constructed by mixing

JHEP07(2023)108

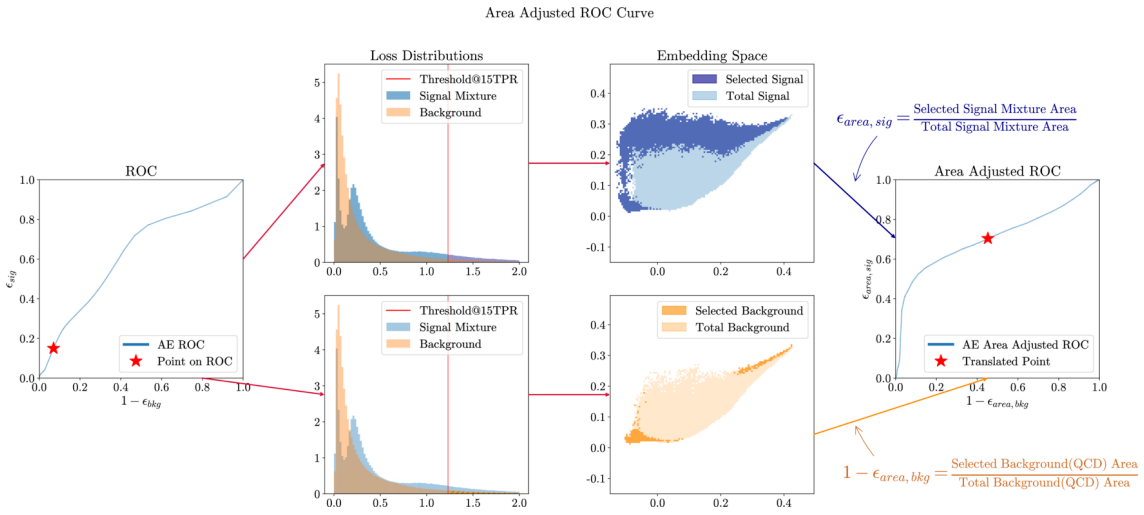

Figure 20. The process of calculating area adjusted ROC curve for anomaly detection algorithm trained with autoencoder on QCD.

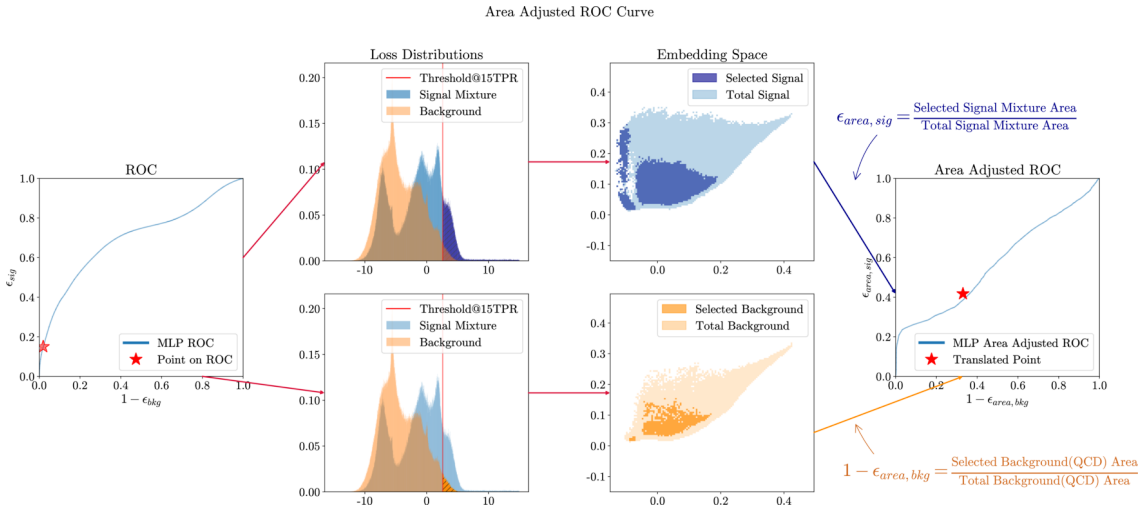

Figure 21. The process of calculating area adjusted ROC curve for anomaly detection algorithm trained with MLP architecture on QCD vs. 2-prong 170 GeV jet task.

QCD, two-prong jets with masses 80, 400 GeV, three-prong jets with masses 80, 400 GeV, and four-prong jets with masses 170 and 400 GeV. Regular ROC curves can vary wildly depending on what test dataset we choose, as can be seen in the upper left and lower left plots comparing ROC curves on two different signal ensemble datasets. However, for area adjusted ROC curves in the upper right and lower right panels, we observe a significantly smaller variation across test datasets.

By adjusting the area with an ensemble dataset, we effectively reduce the sample dependence of the evaluation at the same time.

JHEP07(2023)108

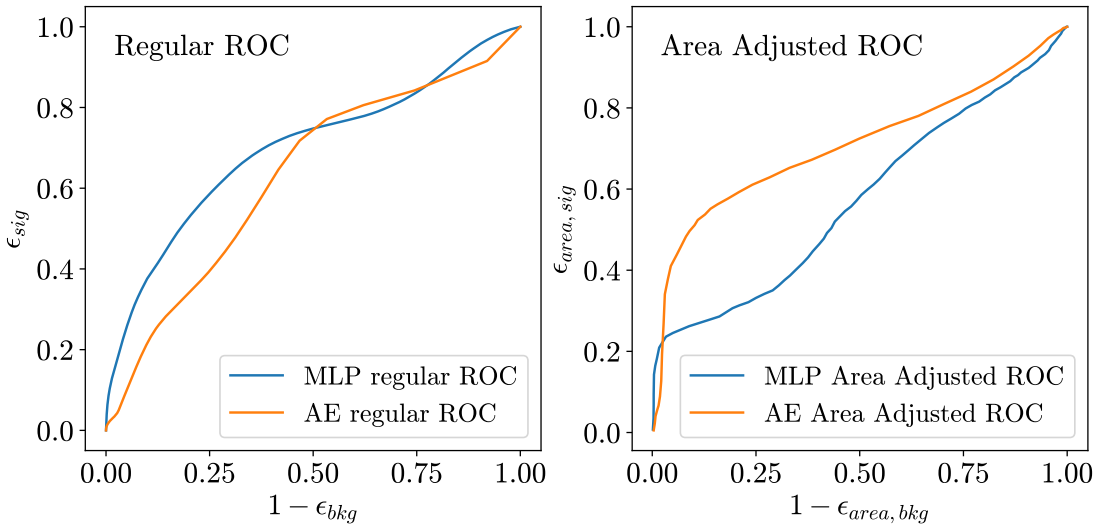

Figure 22. (Left) Comparison of regular ROC curves for two different algorithms, MLP and AE. (Right) Comparison of area adjusted roc curves for the two algorithms, MLP and AE.

6 Conclusion

This paper introduces a method of embedding the physics data manifold with a metric structure into different lower dimensional spaces with simpler metrics. We show neural embedding is capable of condensing complex high-dimensional data into physically meaningful spaces. Furthermore, we explore various types of embedding spaces covering both Euclidean and Hyperbolic embedding.

Using collider physics simulated events of hadronically decaying objects, we demonstrate a neural embedding algorithm that embeds jets into a space where the energy mover's distance is preserved. Using a hierarchical set of progressively more realistic simulations, we find that our neural embedding can preserve the core physical features and self-organize jet datasets into their respective decay types. Furthermore, we find that a Hyperbolic embedding space improves the overall physics interpretation compared to a Euclidean embedding.

We further demonstrate that neural embedding can be used to provide a solution to the complex problem of quantifying the performance of different model agnostic search algorithms, which is an obstacle that needs to be solved if we plan to move towards model-agnostic searches. With the notion of volume in lower dimensional Euclidean spaces, we introduced volume-adjusted roc curves, which aim to quantify the true search breadth of a given algorithm. We find that once we apply the volume-adjusted ROC curve, an autoencoder outperforms supervised learning in its ability to search across the whole manifold of physics events.

Additionally, we note that the optimal transport computation between sets of jets can be very time-consuming. By constructing a neural embedding with the energy mover's distance metric, we avoid the need to recompute optimal transport, allowing for a significantly faster calculation that is embarrassingly parallel.

JHEP07(2023)108

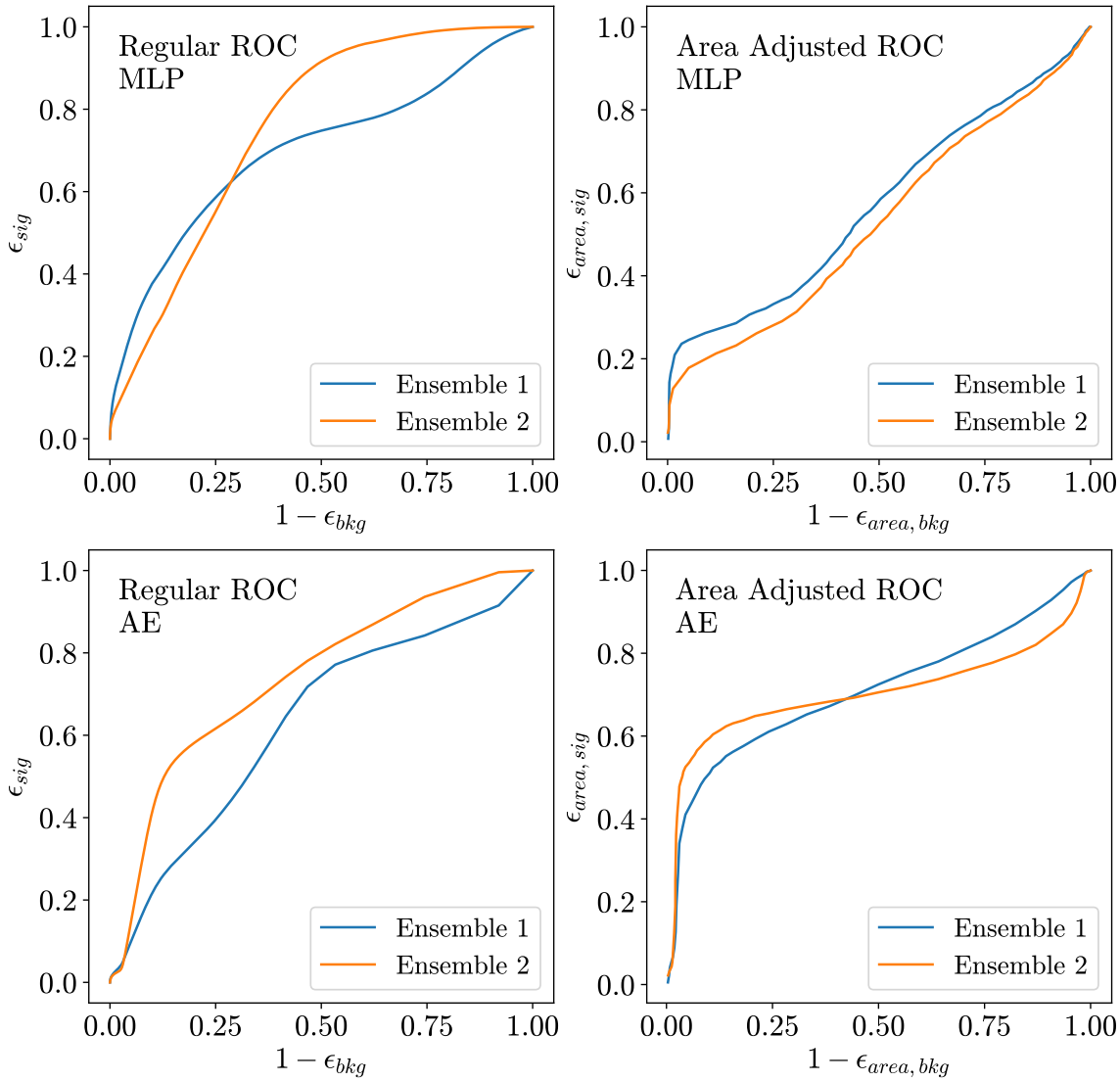

Figure 23. Comparison of stability of regular ROC curves and area adjusted ROC curves for two different anomaly detection algorithms, MLP and AE, on two different ensemble datasets. We see that area adjusted ROC curves are more stable against changing test data ensemble. (Upper Left) Regular ROC curves for MLP (Upper Right) Area adjusted ROC curves for MLP (Lower Left) Regular ROC curves for AE (Lower Right) Area adjusted ROC curves for AE.

We conclude that given a complex physics manifold with a metric structure, it can be beneficial to embed it into different spaces to extract meaningful information. With cheap computational cost and good capability to learn the latent structure, embedding has the potential to find use cases in different collider physics scenarios.

Embedding also provides an alternative way to build a more straightforward space for various tasks without relying on latent variable or probabilistic modelings such as VAEs and flow models. Embedding the QCD physics manifold into different manifolds with desirable geometric properties such as the Poincaré ball has been studied for the first time.

JHEP07(2023)108

This paper realizes a neural embedding using hadron collider events. Despite only exploring a few avenues within our embedded space, we are able to perform quantifications that were previously difficult. As a result, we believe embedding will be an invaluable tool in the physics data analysis pipeline. We believe neural embedding will find use in solving a wide variety of practical problems, such as data compression, anomaly detection, quantification of anomaly detection, organizing physics datasets, and many more.

Acknowledgments

We thank Jesse Thaler, Matthew Schwartz, and Javier Duarte for useful discussions and comments. Additionally, we thank the discussion group with Katherine Fraser, Samuel Homiller, Rashmish K. Mishra, and Patrick McCormack where the idea for this paper originated. P.H. acknowledges support by DOE grant de-sc0021943 and NSF CSSI award #1934700. SEP acknowledges support by DOE grant DE-SC0021225, and the Institute for Fundamental Interactions and Artificial Intelligence (NSF Award #2019786). We thank B. Wyslouch, J. Formaggio, and P. Fisher for providing office space on the 5th floor of MIT building 24.

A MNIST

The MNIST dataset [53] consists of images of characters, each presented in a square array of 28x28 pixels, or 784 total pixels. To perform the NE, we consider one million pairs of MNIST images, including all ten digits. To define the distance between any image, we utilize the optimal transport calculated by POT package [72]. The embedding function is approximated by convolutional neural networks (CNNs) with 4 hidden layers with MLP layers attached at the end that output two numbers, yielding a two-dimensional embedding space.

The embedding into a two-dimensional Euclidean space with a l_2-norm is achieved by learning the function:

$$\phi_{\theta,\mathrm{CNN}} : (\mathcal{X}_{\mathrm{digits}} \subset \mathbb{R}^{784}, \mathcal{W}_2) \to (\mathbb{R}^2, l_2)\,. \tag{A.1}$$

The distributions of optimal transport distances between pairs of images for selected digits are shown in figure 24. If we look at the histograms of the 2-Wasserstein distance distributions, we can see that the distance between 0 and 1 is very far, and digit 9 is about equidistant from both 0 and 1.

With the pairwise optimal transport distances, figure 25 shows the embedding into the Euclidean space with l_2-norm for five selected digits. We show both scatter plots of embedded digits and the contours of the cumulative distribution function (CDF). The contours are obtained by first applying kernel density estimation, then integrating from the maximum probability density function (PDF) value of the two-dimensional distribution by lowering the threshold of the PDF value until the desired enclosed probability mass is achieved (usually chosen to be 0.5 and 0.8). The result of the corresponding NE yields a space with the similarity between digits that we would naively expect from our knowledge of digits and from the optimal transport distance between the images we observed from

JHEP07(2023)108

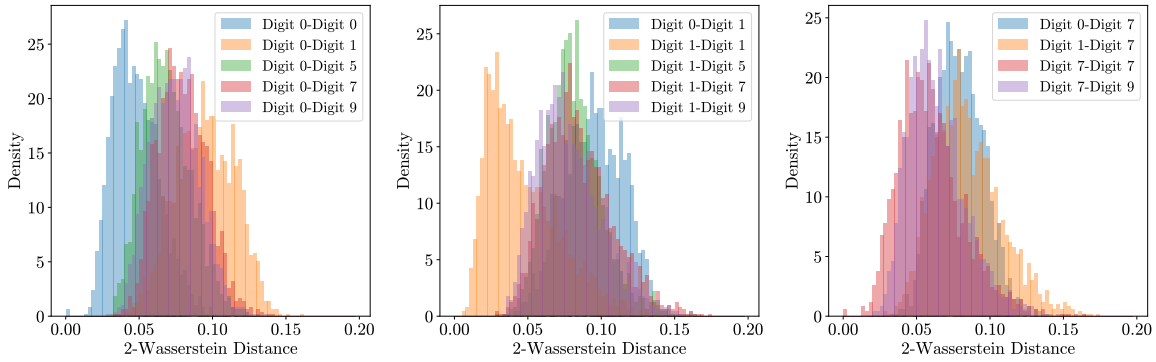

Figure 24. 2-Wasserstein distance $\mathcal{W}_2$ distribution for select digits. (Left) The optimal transport distance with respect to digit 0. (Middle) The optimal transport distance with respect to digit 1. (Right) The optimal transport distance with respect to digit 7.

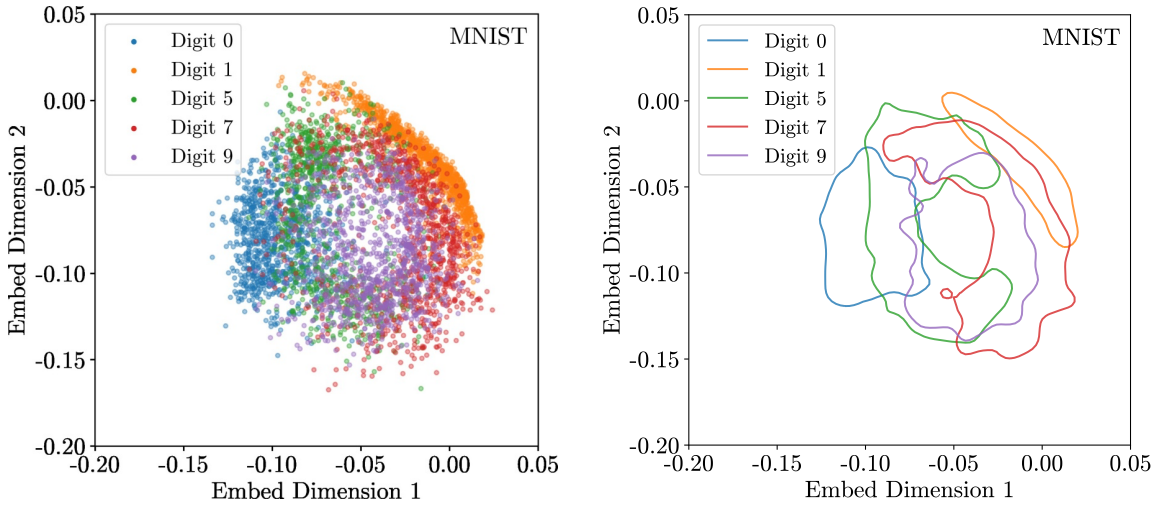

Figure 25. (Left) The scatterplot of embedding of select MNIST digits, 0,1,5,7,9. (Right) The same embedding smoothed with kernel density estimator, with contour lines corresponding to cdf value 0.8.

figure 24. In particular, we observe in the embedded space that the digit 0 and 1 form two distant clusters, while the cluster of digits 5,7, and 9 are located between those two clusters.

B Example of jets

Some of the examples of jets visualized by plotting each constituent in the $\eta - \phi$ plane with circles of sizes proportional to its p_{T} is shown.

B.1 Simple toy jet

In figure 26 and figure 27 we show some examples of simple toy jets.

B.2 Realistic toy jet

In figure 28 and figure 29 we show some examples of realistic toy jets.

JHEP07(2023)108

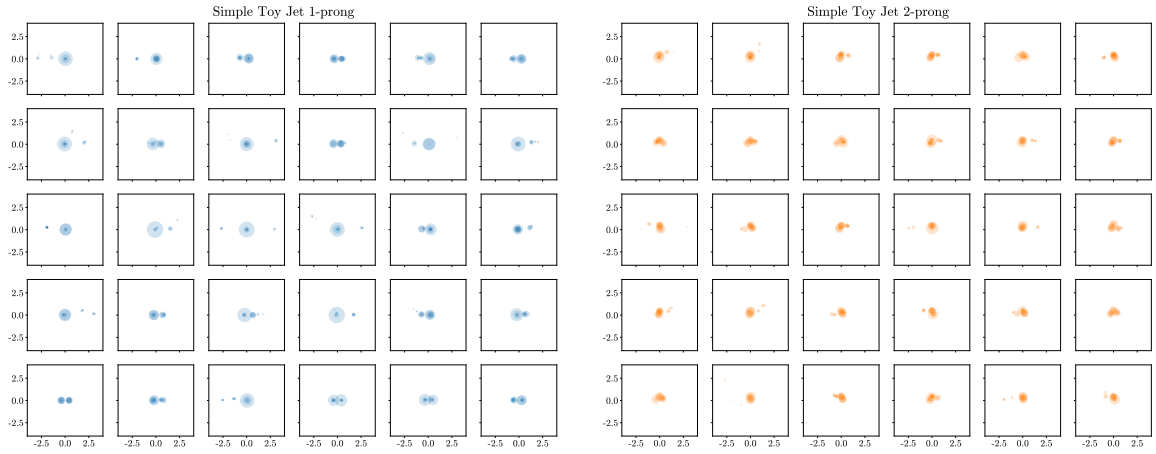

Figure 26. Samples of simple toy jet, (Left) 1-prong(QCD) (Right) 2-prong.

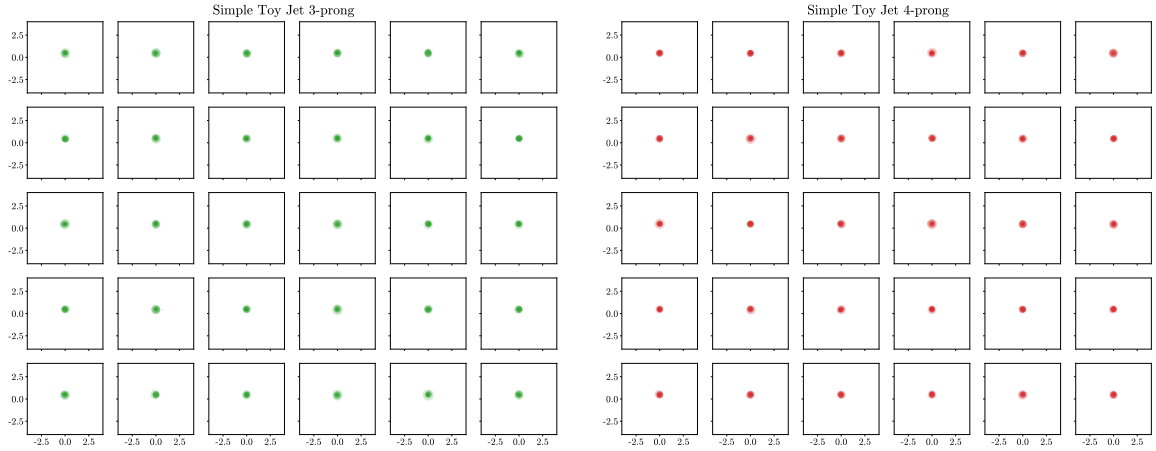

Figure 27. Samples of simple toy jet, (Left) 3-prong (Right) 4-prong.

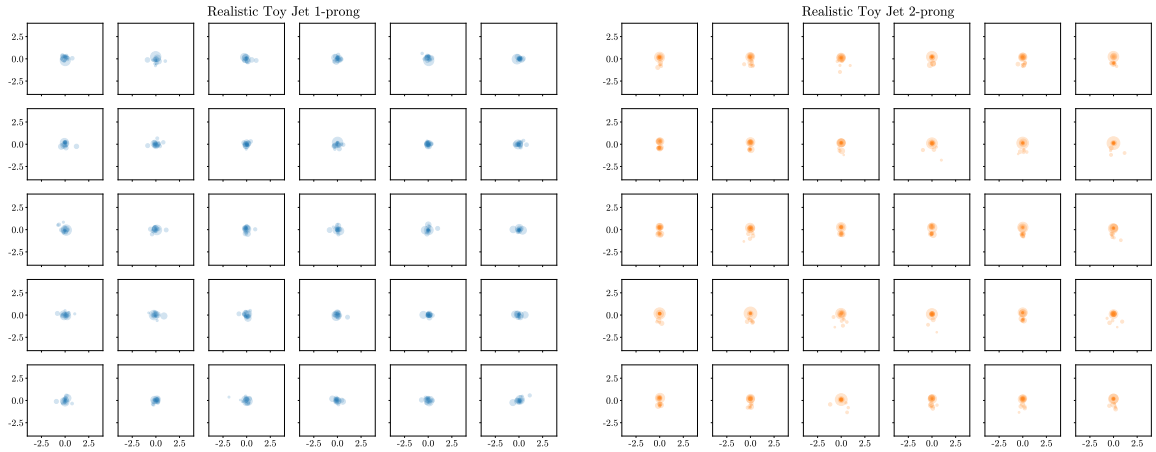

Figure 28. Samples of realistic toy jet, (Left) 1-prong(QCD) (Right) 2-prong.

JHEP07(2023)108

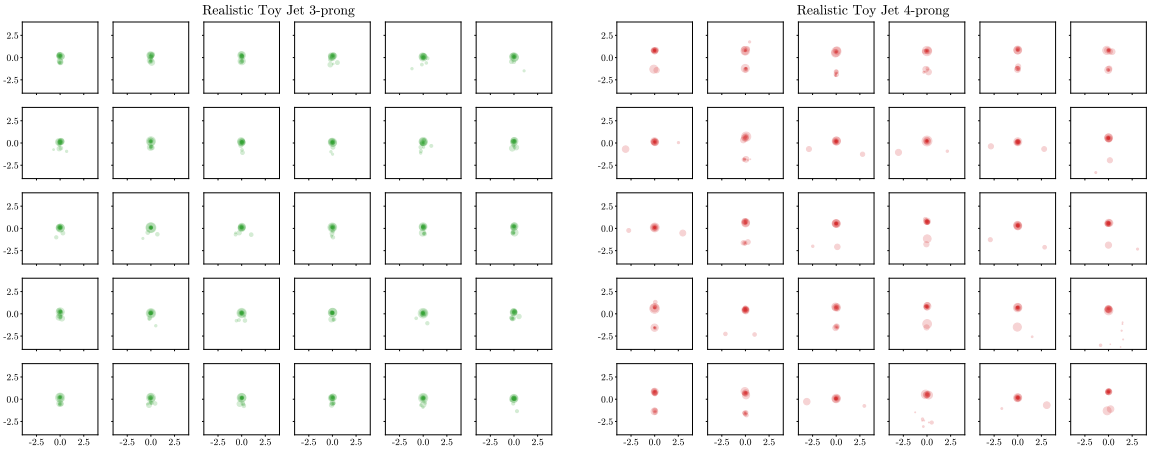

Figure 29. Samples of realistic toy jet, (Left) 3-prong (Right) 4-prong.

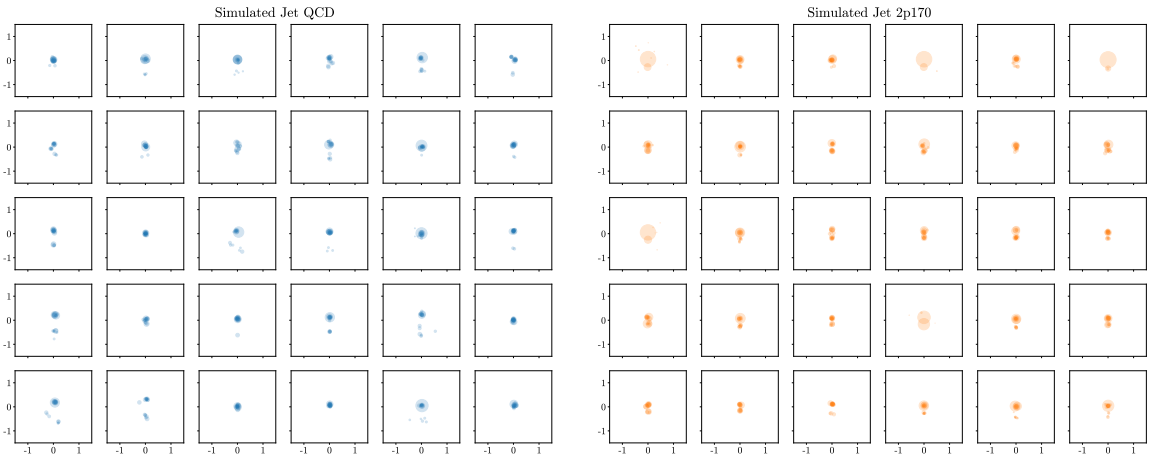

Figure 30. Samples of simulated jet, (Left) QCD (Right) 2-prong 170 GeV.

B.3 Simulated jet

In figure 30 and figure 31 we show some examples of simulated jets.

C Stability of area adjusted ROC curve

We study the effect of binning by varying the threshold parameter when calculating the area ROC curve. Figure 32 shows the area adjusted ROC curve with varying threshold parameter for minimum required number of events in each 2D bin. We see that the new ROC curve is robust against choosing binning and thresholding of this minimum number of events. The cutoff was varied up to 1 percent of the evaluation set, and within that range of the cutoff the area adjusted ROC curve is stable.

JHEP07(2023)108

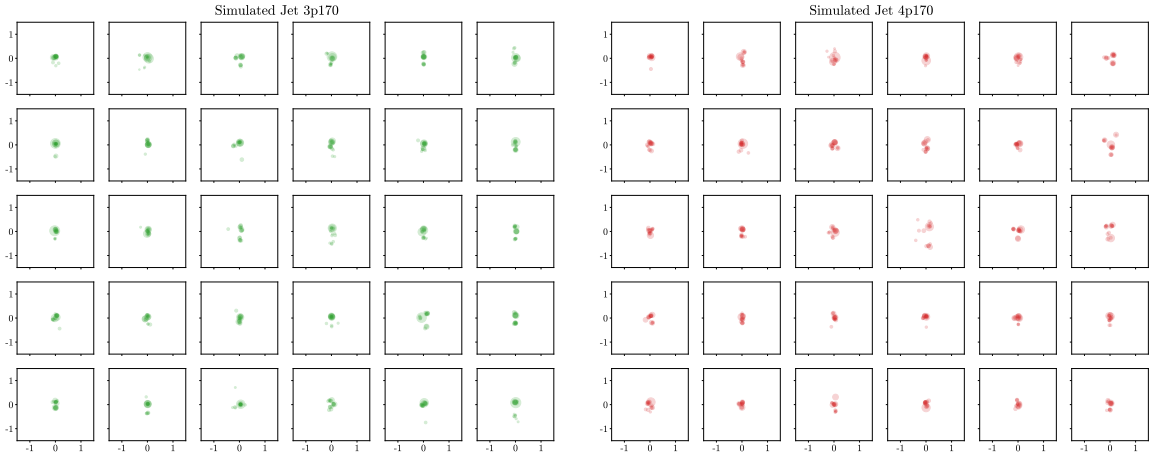

Figure 31. Samples of simulated jet, (Left) 3-prong 170 GeV (Right) 4-prong 170 GeV.

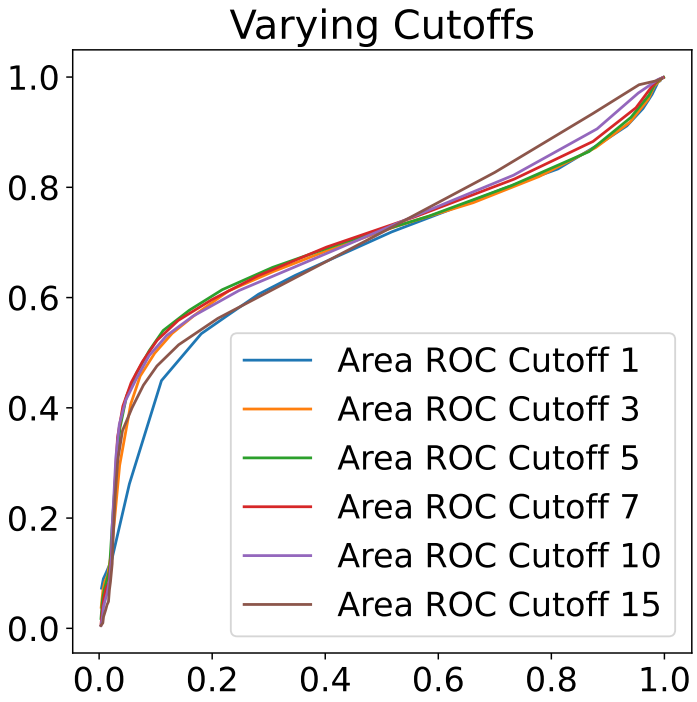

Figure 32. The area adjusted roc curve calculated for different thresholds.

D Neural network architecture details

D.1 CNN

The CNN architecture used in A is made of 3 2-D convolution layers with kernel size 5, with max pooling and ReLU activation. The linear layers have 1000, 400, 200 neurons with leaky ReLU activation, with batch normalization [73], and dropout [74]. There are 1M total number of parameters for this model.

JHEP07(2023)108

Section	Dataset	Attention Heads	Linear Layers	Dropout Prob.	Params
Section 4.1.1	Simple Toy Jets 3.2	4	1200,450,30	0.25	1.6M
Section 4.1.2	Realistic Toy Jets 3.3	4	1000,400,20	0.2	1.3M
Section 4.2	Simulated Jets 4.2	4	1000,400,20	0.2	1.3M
Section 4.3	Simulated Jets 4.2	4	1000,500,20	0.25	1.4M

Table 3. Summary of network architectures for jet data.

D.2 Transformers

For the transformer architecture used in 4.1.1, 4.1.2, 4.2 4.3, architecture search was performed for each setting to achieve the minimum distortion. Since the feature embedding dimension was always fixed to 32, the positional encoding in eq. D.1 was used.

$$f(t)^{(i)} := \begin{cases} \sin(\omega_t \cdot t) & \text{if } i = 2k \\ \cos(\omega_t \cdot t) & \text{if } i = 2k+1 \end{cases} \quad \text{where } \omega = \frac{1}{10000^{2k/32}} \tag{D.1}$$

For transformers, dropout [74] was used for regularization. The details of architectures for each cases is summarized in the table 3.

References

[1] P.T. Komiske, E.M. Metodiev and J. Thaler, *Metric space of collider events*, *Phys. Rev. Lett.* **123** (2019) 041801 [arXiv:1902.02346] [INSPIRE].

[2] T. Cai, J. Cheng, K. Craig and N. Craig, *Which metric on the space of collider events?*, *Phys. Rev. D* **105** (2022) 076003 [arXiv:2111.03670] [INSPIRE].

[3] S. Kolouri et al., *Generalized sliced Wasserstein distances*, arXiv:1902.00434.

[4] M. Crispim Romão et al., *Use of a generalized energy Mover's distance in the search for rare phenomena at colliders*, *Eur. Phys. J. C* **81** (2021) 192 [arXiv:2004.09360] [INSPIRE].

[5] S. Tsan et al., *Particle graph autoencoders and differentiable, learned energy Mover's distance*, in the proceedings of the 35th *conference on neural information processing systems*, (2021) [arXiv:2111.12849] [INSPIRE].

[6] K. Fraser et al., *Challenges for unsupervised anomaly detection in particle physics*, *JHEP* **03** (2022) 066 [arXiv:2110.06948] [INSPIRE].

[7] J.H. Collins, *An exploration of learnt representations of W jets*, arXiv:2109.10919 [INSPIRE].

[8] ATLAS collaboration, *Measurements of multijet event isotropies using optimal transport with the ATLAS detector*, ATLAS-CONF-2022-056, CERN, Geneva, Switzerland (2022).

[9] E.M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: learning from mixed samples in high energy physics*, *JHEP* **10** (2017) 174 [arXiv:1708.02949] [INSPIRE].

JHEP07(2023)108

[10] J.H. Collins, K. Howe and B. Nachman, *Anomaly detection for resonant new physics with machine learning*, *Phys. Rev. Lett.* **121** (2018) 241803 [arXiv:1805.02664] [INSPIRE].

[11] J.H. Collins, K. Howe and B. Nachman, *Extending the search for new resonances with machine learning*, *Phys. Rev. D* **99** (2019) 014038 [arXiv:1902.02634] [INSPIRE].

[12] B. Nachman and D. Shih, *Anomaly detection with density estimation*, *Phys. Rev. D* **101** (2020) 075042 [arXiv:2001.04990] [INSPIRE].

[13] T. Heimel, G. Kasieczka, T. Plehn and J.M. Thompson, *QCD or what?*, *SciPost Phys.* **6** (2019) 030 [arXiv:1808.08979] [INSPIRE].

[14] M. Farina, Y. Nakai and D. Shih, *Searching for new physics with deep autoencoders*, *Phys. Rev. D* **101** (2020) 075021 [arXiv:1808.08992] [INSPIRE].

[15] O. Cerri et al., *Variational autoencoders for new physics mining at the Large Hadron Collider*, *JHEP* **05** (2019) 036 [arXiv:1811.10276] [INSPIRE].

[16] M. Kuusela et al., *Semi-supervised anomaly detection — towards model-independent searches of new physics*, *J. Phys. Conf. Ser.* **368** (2012) 012032 [arXiv:1112.3329] [INSPIRE].

[17] T.S. Roy and A.H. Vijay, *A robust anomaly finder based on autoencoders*, arXiv:1903.02032.

[18] A. Blance, M. Spannowsky and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, *JHEP* **10** (2019) 047 [arXiv:1905.10384] [INSPIRE].

[19] J. Hajer, Y.-Y. Li, T. Liu and H. Wang, *Novelty detection meets collider physics*, *Phys. Rev. D* **101** (2020) 076015 [arXiv:1807.10261] [INSPIRE].

[20] R.T. D'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, *Learning multivariate new physics*, arXiv:1912.12155.

[21] R.T. D'Agnolo and A. Wulzer, *Learning new physics from a machine*, *Phys. Rev. D* **99** (2019) 015014 [arXiv:1806.02350] [INSPIRE].

[22] M. Crispim Romão, N.F. Castro and R. Pedro, *Finding new physics without learning about it: anomaly detection as a tool for searches at colliders*, *Eur. Phys. J. C* **81** (2021) 27 [*Erratum ibid.* **81** (2021) 1020] [arXiv:2006.05432] [INSPIRE].

[23] C. Fanelli, J. Giroux and Z. Papandreou, *"Flux+Mutability": a conditional generative approach to one-class classification and anomaly detection*, *Mach. Learn. Sci. Tech.* **3** (2022) 045012 [arXiv:2204.08609] [INSPIRE].

[24] B.M. Dillon, R. Mastandrea and B. Nachman, *Self-supervised anomaly detection for new physics*, *Phys. Rev. D* **106** (2022) 056005 [arXiv:2205.10380] [INSPIRE].

[25] S. Alvi, C.W. Bauer and B. Nachman, *Quantum anomaly detection for collider physics*, *JHEP* **02** (2023) 220 [arXiv:2206.08391] [INSPIRE].

[26] L. Bradshaw, S. Chang and B. Ostdiek, *Creating simple, interpretable anomaly detectors for new physics in jet substructure*, *Phys. Rev. D* **106** (2022) 035014 [arXiv:2203.01343] [INSPIRE].

[27] V.S. Ngairangbam, M. Spannowsky and M. Takeuchi, *Anomaly detection in high-energy physics using a quantum autoencoder*, *Phys. Rev. D* **105** (2022) 095004 [arXiv:2112.04958] [INSPIRE].

[28] S. Chekanov and W. Hopkins, *Event-based anomaly detection for searches for new physics*, *Universe* **8** (2022) 494 [arXiv:2111.12119] [INSPIRE].

JHEP07(2023)108

[29] V. Mikuni, B. Nachman and D. Shih, *Online-compatible unsupervised nonresonant anomaly detection*, *Phys. Rev. D* **105** (2022) 055006 [arXiv:2111.06417] [INSPIRE].

[30] J.A. Aguilar-Saavedra, *Anomaly detection from mass unspecific jet tagging*, *Eur. Phys. J. C* **82** (2022) 130 [arXiv:2111.02647] [INSPIRE].

[31] B. Ostdiek, *Deep set auto encoders for anomaly detection in particle physics*, *SciPost Phys.* **12** (2022) 045 [arXiv:2109.01695] [INSPIRE].

[32] G. Kasieczka, B. Nachman and D. Shih, *New methods and datasets for group anomaly detection from fundamental physics*, in the proceedings of the *Conference on knowledge discovery and data mining*, (2021) [arXiv:2107.02821] [INSPIRE].

[33] S. Caron, L. Hendriks and R. Verheyen, *Rare and different: anomaly scores from a combination of likelihood and out-of-distribution models to detect new physics at the LHC*, *SciPost Phys.* **12** (2022) 077 [arXiv:2106.10164] [INSPIRE].

[34] T. Dorigo et al., *RanBox: anomaly detection in the copula space*, *JHEP* **01** (2023) 008 [arXiv:2106.05747] [INSPIRE].

[35] O. Atkinson et al., *Anomaly detection with convolutional graph neural networks*, *JHEP* **08** (2021) 080 [arXiv:2105.07988] [INSPIRE].

[36] T. Finke et al., *Autoencoders for unsupervised anomaly detection in high energy physics*, *JHEP* **06** (2021) 161 [arXiv:2104.09051] [INSPIRE].

[37] L. van der Maaten and G. Hinton, *Visualizing data using t-SNE*, *J. Mach. Learn. Res.* **9** (2008) 2579.

[38] L. McInnes, J. Healy and J. Melville, *UMAP: Uniform Manifold Approximation and Projection for dimension reduction*, arXiv:1802.03426.

[39] G. Corso et al., *Neural distance embeddings for biological sequences*, *Adv. Neural Inf. Process. Syst.* **34** (2021) 18539 [arXiv:2109.09740].

[40] A. Narayanan et al., *graph2vec: learning distributed representations of graphs*, arXiv:1707.05005.

[41] B. Rozemberczki and R. Sarkar, *Fast sequence-based embedding with diffusion graphs*, in the proceedings of the *International workshop on complex networks*, (2018), p. 99.

[42] F. Gong et al., *SMR: medical knowledge graph embedding for safe medicine recommendation*, *Big Data Research* **23** (2021) 100174.

[43] N.K. Ahmed et al., *Learning role-based graph embeddings*, arXiv:1802.02896.

[44] J. Pennington, R. Socher and C. Manning, *Glove: global vectors for word representation*, in the proceedings of the of the 2014 *conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2014) [DOI:10.3115/v1/d14-1162].

[45] C. Frogner, F. Mirzazadeh and J. Solomon, *Learning embeddings into entropic Wasserstein spaces*, in the proceedings of the *International conference on learning representations*, (2019).

[46] A. Akbik, D. Blythe and R. Vollgraf, *Contextual string embeddings for sequence labeling*, in the proceedings of the of the 27$^{\text{th}}$ *International conference on computational linguistics*, Santa Fe, NM, U.S.A. (2018), p. 1638.

[47] R. Bartusiak et al., *WordNet2Vec: corpora agnostic word vectorization method*, *Neurocomputing* **326-327** (2019) 141.

JHEP07(2023)108

[48] A. Sanakoyeu, V. Tschernezki, U. Buchler and B. Ommer, *Divide and conquer the embedding space for metric learning*, in the proceedings of the of the *IEEE/CVF conference on Computer Vision and Pattern Recognition*, (2019), p. 471.

[49] D. Garcia-Gasulla et al., *A visual embedding for the unsupervised extraction of abstract semantics*, *Cognitive Systems Research* **42** (2017) 73.

[50] B.M. Dillon et al., *Symmetries, safety, and self-supervision*, arXiv:2108.04253.

[51] B.M. Dillon, R. Mastandrea and B. Nachman, *Self-supervised anomaly detection for new physics*, *Phys. Rev. D* **106** (2022) 056005 [arXiv:2205.10380] [INSPIRE].

[52] A. Paszke et al., *PyTorch: an imperative style, high-performance deep learning library*, in *Advances in neural information processing systems* 32, Curran Associates Inc., U.S.A. (2019), p. 8024.

[53] Y. LeCun and C. Cortes, *MNIST handwritten digit database*, http://yann.lecun.com/exdb/mnist/.

[54] N. Courty, R. Flamary and M. Ducoffe, *Learning Wasserstein embeddings*, arXiv:1710.07457.

[55] M. Cacciari, G.P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, *JHEP* **04** (2008) 063 [arXiv:0802.1189] [INSPIRE].

[56] M. Cacciari, G.P. Salam and G. Soyez, *FastJet user manual*, *Eur. Phys. J. C* **72** (2012) 1896 [arXiv:1111.6097] [INSPIRE].

[57] CMS collaboration, *Measurement of the splitting function in pp and Pb-Pb collisions at* $\sqrt{s_{\rm NN}} = 5.02$ *TeV*, *Phys. Rev. Lett.* **120** (2018) 142302 [arXiv:1708.09429] [INSPIRE].

[58] A Large Ion Collider Experiment and ALICE collaborations, *Measurement of the groomed jet radius and momentum splitting fraction in pp and Pb-Pb collisions at* $\sqrt{s_{\rm NN}} = 5.02$ *TeV*, *Phys. Rev. Lett.* **128** (2022) 102001 [arXiv:2107.12984] [INSPIRE].

[59] ATLAS collaboration, *Properties of* $g \to b\bar{b}$ *at small opening angles in pp collisions with the ATLAS detector at* $\sqrt{s} = 13$ *TeV*, *Phys. Rev. D* **99** (2019) 052004 [arXiv:1812.09283] [INSPIRE].

[60] A. Vaswani et al., *Attention is all you need*, *Adv. Neural Inf. Process. Syst.* **30** (2017) [arXiv:1706.03762].

[61] DELPHES collaboration, *DELPHES 3, a modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [arXiv:1307.6346] [INSPIRE].

[62] J. Alwall et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079 [arXiv:1405.0301] [INSPIRE].

[63] P. Skands, S. Carrazza and J. Rojo, *Tuning PYTHIA* 8.1*: the Monash* 2013 *tune*, *Eur. Phys. J. C* **74** (2014) 3024 [arXiv:1404.5630] [INSPIRE].

[64] T. Sjöstrand et al., *An introduction to PYTHIA* 8.2, *Comput. Phys. Commun.* **191** (2015) 159 [arXiv:1410.3012] [INSPIRE].

[65] M. Cacciari and G.P. Salam, *Dispelling the* N^3 *myth for the* k_t *jet-finder*, *Phys. Lett. B* **641** (2006) 57 [hep-ph/0512210] [INSPIRE].

[66] M. Cacciari, G.P. Salam and G. Soyez, *FastJet user manual*, *Eur. Phys. J. C* **72** (2012) 1896 [arXiv:1111.6097] [INSPIRE].

JHEP07(2023)108

[67] M. Nickel and D. Kiela, *Poincaré embeddings for learning hierarchical representations*, *Adv. Neural Inf. Process. Syst.* **30** (2017) [arXiv:1705.08039].

[68] W. Peng et al., *Hyperbolic deep neural networks: a survey*, arXiv:2101.04562.

[69] A. Klimovskaia, D. Lopez-Paz, L. Bottou and M. Nickel, *Poincaré maps for analyzing complex hierarchies in single-cell data*, *Nature Commun.* **11** (2020) 1.

[70] L. Chennuru Vankadara and U. von Luxburg, *Measures of distortion for machine learning*, in the proceedings of the *Advances in neural information processing systems* 31, Curran Associates, Inc., U.S.A. (2018).

[71] P.T. Komiske et al., *Exploring the space of jets with CMS open data*, *Phys. Rev. D* **101** (2020) 034009 [arXiv:1908.08542] [INSPIRE].

[72] R. Flamary et al., *POT: Python Optimal Transport*, *J. Mach. Learn. Res.* **22** (2021) 1.

[73] S. Ioffe and C. Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, arXiv:1502.03167 [INSPIRE].

[74] N. Srivastava et al., *Dropout: a simple way to prevent neural networks from overfitting*, *J. Mach. Learn. Res.* **15** (2014) 1929.

JHEP07(2023)108