Particle Cloud Generation with Message Passing Generative Adversarial Networks

Raghav Kansal, Javier Duarte, Hao Su
University of California San Diego
La Jolla, CA 92093, USA

Breno Orzari, Thiago Tomei
Universidade Estadual Paulista
São Paulo/SP - CEP 01049-010, Brazil

Maurizio Pierini, Mary Touranakou*
European Organization for Nuclear Research (CERN)
CH-1211 Geneva 23, Switzerland

Jean-Roch Vlimant
California Institute of Technology
Pasadena, CA 91125, USA

Dimitrios Gunopulos
National and Kapodistrian University of Athens
Athens 15772, Greece

Abstract

In high energy physics (HEP), jets are collections of correlated particles produced ubiquitously in particle collisions such as those at the CERN Large Hadron Collider (LHC). Machine-learning-based generative models, such as generative adversarial networks (GANs), have the potential to significantly accelerate LHC jet simulations. However, despite jets having a natural representation as a set of particles in momentum-space, a.k.a. a particle cloud, to our knowledge there exist no generative models applied to such a dataset. We introduce a new particle cloud dataset (JetNet), and, due to similarities between particle and point clouds, apply to it existing point cloud GANs. Results are evaluated using (1) the 1-Wasserstein distance between high- and low-level feature distributions, (2) a newly developed Fréchet ParticleNet Distance, and (3) the coverage and (4) minimum matching distance metrics. Existing GANs are found to be inadequate for physics applications, hence we develop a new message passing GAN (MPGAN), which outperforms existing point cloud GANs on virtually every metric and shows promise for use in HEP. We propose JetNet as a novel point-cloud-style dataset for the machine learning community to experiment with, and set MPGAN as a benchmark to improve upon for future generative models.

1 Introduction

Over the past decade, machine learning (ML) has become the de facto way to analyze jets, collimated high-energy sprays of particles [1] produced at the CERN Large Hadron Collider (LHC). To apply ML to jets, the most natural representation is a *particle cloud*, a variable-sized set of points in momentum space, whose radiation pattern contains rich information about the underlying physics known as quantum chromodynamics (QCD). A fundamental question is whether ML algorithms can model this underlying physics and successfully reproduce the rich high- and low-level structure in jets.

*Also at National and Kapodistrian University of Athens, Athens, Greece.

Preprint. Under review.

Answering this question affirmatively has important practical applications. At the LHC, large simulated data samples of collision events[2] are generated using Monte Carlo (MC) methods in order to translate theoretical predictions into observable distributions. These samples, numbering in the billions of events, require computationally expensive modeling of the interaction of particles traversing the detector material. Recently developed generative frameworks in ML, such as generative adversarial networks (GANs), if accurate enough can be used to accelerate this simulation by potentially five orders of magnitude [2].

In this work, we advocate for a benchmark jet dataset (JetNet) and propose several physics- and computer-vision-inspired metrics with which the ML community can improve and evaluate generative models in high energy physics (HEP). We then apply existing point cloud GANs on JetNet and find the results to be inadequate for physics applications. Finally we develop our own message passing GAN (MPGAN), which dramatically improves results on virtually every metric, and propose it as a benchmark on JetNet.

2 Jets

High-energy proton-proton collisions at the LHC produce elementary particles like quarks and gluons, which cannot be isolated due to the QCD property of color confinement [3]. These particles continuously radiate or "split" into a set of particles, known as a parton shower. Eventually they cool to an energy at which they undergo the process of hadronization, where the fundamental particles combine to form more stable hadrons, such as pions and protons. The final set of collimated hadrons produced after such a process is referred to as a jet.

The task of simulating a single jet can be algorithmically defined as inputting an initial particle, which produces the jet, and outputting the final set of particles a.k.a. the jet constituents. Typically in HEP the parton shower and hadronization are steps that are simulated sequentially using MC event generators such as PYTHIA [4] or HERWIG [5]. Simulating either process exactly is not possible because of the complex underlying physics (QCD), and instead these event generators fit simplified physics-inspired stochastic models, such as the Lund string model for hadronization [6], to existing data using MC methods. These models are then used to generate more data for use in analyses. The present work can be seen as an extension of this idea, using a simpler, ML-based model, also fitted to data, for generating the jet in one shot, where we are effectively trading the interpretability of the MC methods for the speed of GPU-accelerated ML generators.

Representations. As common for collider physics, we use a Cartesian coordinate system with the z axis oriented along the beam axis, the x axis on the horizontal plane, and the y axis oriented upward. The x and y axes define the transverse plane, while the z axis identifies the longitudinal direction. The azimuthal angle ϕ is computed with respect to the x axis. The polar angle θ is used to compute the pseudorapidity $\eta = -\log(\tan(\theta/2))$. The transverse momentum (p_T) is the projection of the particle momentum on the (x, y) plane. As is customary, we transform the particle momenta from Cartesian coordinates (p_x, p_y, p_z) to longitudinal-boost-invariant pseudo-angular coordinates $(p_\mathrm{T}, \eta, \phi)$, as shown in Fig. 1.

In the context of ML, jets can be represented in multiple ways. One popular representation is as images [7, 8], created by projecting each jet's particle constituents onto a discretized angular η-ϕ plane, and taking the intensity of each "pixel" in this grid to be a monotonically increasing function of the corresponding particle p_T. These tend to be extremely sparse, with typically less than 10% of pixels nonempty [9], and the discretization process can lower the resolution.

Two more spatially efficient representations are as ordered lists or un-ordered sets [10] of the jet constituents and their features. The difficulty with the former is that there is no particular preferred ordering of the particles — one would have to impose an arbitrary ordering such as by transverse momentum [11]. The more natural representation is the un-ordered set of particles in momentum space, which we refer to as a "particle cloud." This is in analogy to point cloud representations of 3D objects in position-space prevalent in computer vision, created e.g. by sampling from 3D ShapeNet models [12].

[2] An event is a set of observed particles with well-defined momenta, but not generally well-defined positions.

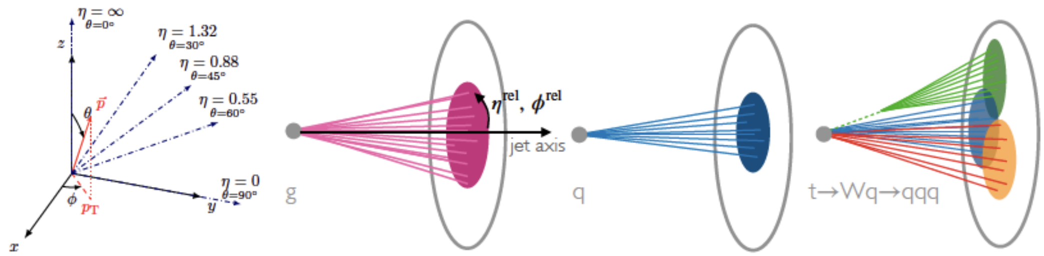

Figure 1: The collider physics coordinate system defining $(p_{\mathrm{T}}, \eta, \phi)$ (left). The three jet classes in our dataset (right). Gluon (g) and light quark (q) jets have simple topologies, with q jets generally containing fewer particles. Top quark (t) jets have a complex three-pronged structure. Shown also are the relative angular coordinates η^{rel} and ϕ^{rel}, measured from the jet axis.

Apart from how the samples are produced, two significant differences between jets and ShapeNet-based point clouds are that (1) jets have physically meaningful low- and high-level features such as particle momentum, total mass of the jet, the number of sub-jets, and n-particle energy correlations, which are important to reproduce correctly for physics analysis applications as these are the physical observables by which we characterize jets and (2) current work on point clouds GANs to our knowledge all use fixed-size clouds, whereas jets, due to factors such as the jet momentum and the stochastic nature of particle production, inherently contain varying numbers of particles.

JetNet. We publish JetNet[3] [13] under the CC-BY 4.0 license, to facilitate and advance ML research in HEP, and provide a new point-cloud-style dataset to experiment with. Derived from Ref. [14][4], it consists of simulated particle jets with transverse momenta $p_{\mathrm{T}}^{\mathrm{jet}} \approx 1\,\mathrm{TeV}$, originating from gluons, top quarks, and light quarks produced in 13 TeV proton-proton collisions in a simplified detector. Technical details of the generation process are given in App. A. We limit the number of constituents to the 30 highest p_{T} particles per jet, allowing for jets with potentially fewer than 30 by zero-padding. For each particle we provide the following four features: the relative angular coordinates $\eta^{\mathrm{rel}} = \eta^{\mathrm{particle}} - \eta^{\mathrm{jet}}$ and $\phi^{\mathrm{rel}} = \phi^{\mathrm{particle}} - \phi^{\mathrm{jet}} \pmod{2\pi}$, relative transverse momentum $p_{\mathrm{T}}^{\mathrm{rel}} = p_{\mathrm{T}}^{\mathrm{particle}} / p_{\mathrm{T}}^{\mathrm{jet}}$ and a binary mask feature classifying the particle as genuine or zero-padded.

We choose three jet classes, depicted in Fig. 1, to individually target the unique and challenging properties of jets. Gluons provide a useful baseline test, as they typically radiate into a large number of particles before hadronization, largely avoiding the variable-sized cloud issue (at least with a 30 particle maximum), and have a relatively simple topology. Light quarks share the simple topology, but produce fewer final-state particles, resulting in a larger fraction of zero-padded particles in the dataset. They allow evaluation of a models' ability to handle variable-sized clouds. Finally, top quarks decay into three lighter quarks through an intermediate particle called a W boson, which each may produce their own sub-jets, meaning they also produce a large number of particles. In addition, they have a more complex two- or three-pronged topology depending on whether the jet clustering algorithm captures all three or just two of these sub-jets. This results in bimodal jet feature distributions (one peak corresponding to fully merged top quark jets and the other to semi-merged — see Fig. 4). Thus, top quark jets test the models' ability to learn the rich global structure and clustering history of a particle cloud.

3 Related Work

Generative models in HEP. Past work in this area has exclusively used image-based representations for HEP data. One benefit of this is the ability to employ convolutional neural network (CNN) based generative models, which have been highly successful on computer vision tasks. Refs. [2, 15–18], for example, build upon CNN-based GANs, and Ref. [19] uses an auto-regressive model, to output jet- and detector-data-images.

[3]Blinded for NeurIPS submission.

[4]This dataset was also released under the CC-BY 4.0 license.

In addition to the issues with such representations outlined in Sec. 2, the high sparsity of the images can lead to training difficulties in GANs, and the irregular geometry of the data — a single LHC detector can typically have multiple sections with differing pixel sizes and shapes — poses a challenge for CNN GANs which output uniform matrices. While these can be mitigated to an extent with techniques such as batch normalization [20] and using larger/more regular pixels [16], our approach avoids both issues by generating particle-cloud-representations of the data, as these are inherently sparse data structures and are completely flexible to the underlying geometry.

GANs for point clouds. Particle clouds and point clouds have similarities insomuch as they represent sets of elements in some physical space, hence we first test existing point cloud GANs on JetNet. The three published point cloud generators are r-GAN [21], GraphCNN-GAN [22], and TreeGAN [23]. r-GAN uses a fully-connected (FC) network, GraphCNN-GAN uses graph convolutions based on dynamic k-nn graphs in the intermediate feature spaces, and TreeGAN iteratively up-samples the graphs with information passing from ancestor to descendant nodes.

Past work has used either an FC or a PointNet [24]-style discriminator. Ref. [25] is the first work to study point cloud discriminator design in detail and finds amongst a number of PointNet and graph convolutional models that PointNet-Mix, which uses both max- and average-pooled features, is the most performant. We apply the FC and GraphCNN generators — TreeGAN's up-sampling method is not well-suited to our small 30-particle clouds, but should be explored in future work on larger clouds — and FC and PointNet-Mix discriminators as baselines to our datasets, but find jet structure is not adequately reproduced. GraphCNN's local convolutions make learning global structure difficult, and while the FC generator and PointNet discriminator combination is an improvement, is not able to learn multi-particle correlations, particularly for the complex top quark jets, nor deal with the variable-sized light quark jets to the extent necessary for physics applications.

Message Passing Neural Networks. We attempt to overcome limitations of existing GANs by designing a novel generator and discriminator which can learn such correlations and handle variable-sized particle clouds. For both networks we build upon the generic message-passing neural network (MPNN) [26] framework with physics-conscious design choices, which we call message-passing GAN (MPGAN). We find MPGAN outperforms existing models on virtually all evaluation metrics.

3.1 Evaluating generative models.

Evaluating generative models is a difficult task, however there has been extensive work in this area in both the physics and computer-vision communities.

Physics-inspired metrics. An accurate jet simulation algorithm should reproduce both low-level and high-level features (such as those described in Sec. 2), hence a standard method of validating generative models, which we employ, is to compare the distributions of such features between the real and generated samples [2, 15–18, 27].

For application in HEP, a generative model needs to produce jets with physical features indistinguishable from real. Therefore, we propose the validation criteria that differences between real and generated sample features may not exceed those between sets of randomly chosen real samples. To verify this, we use bootstrapping to compare between random samples of only real jets as a baseline.

A practically useful set of features to validate against are the so-called "energy-flow polynomials" (EFPs) [28], which are a set of multi-particle correlation functions. Importantly, the set of all EFPs forms a linear basis for all useful jet-level features[5]. Therefore, we claim that if we observe all EFP distributions to be reproduced with high fidelity and to match the above criteria, we can conclude with strong confidence that our model is outputting accurate particle clouds.

Computer-vision-inspired metrics A popular metric for evaluating images which has shown to be sensitive to output quality and mode-collapse, although it has its limitations [29], is the Fréchet Inception Distance [30] (FID). FID is defined as the Fréchet distance between Gaussian distributions fitted to the activations of a fully-connected layer of the Inception-v3 image classifier in response to real and generated samples. We develop a particle-cloud-analogue of this metric, which we call

[5] In the context of HEP, this means all infrared- and colinear-safe observables.

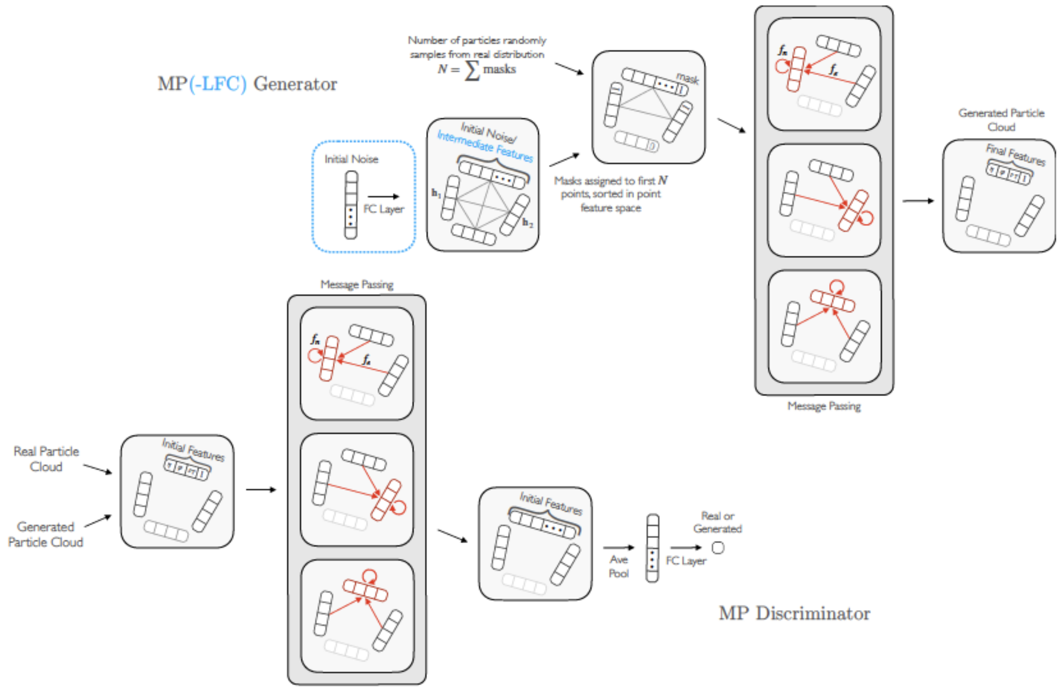

Figure 2: Top: The MP generator uses message passing to generate a particle cloud. In blue is the initial latent vector and FC layer part of the MP-LFC variant. Bottom: The MP discriminator uses message passing to classify an input particle cloud as real or generated.

Fréchet ParticleNet Distance (FPND), using the state-of-the-art ParticleNet graph convolutional jet classifier [9] in lieu of the Inception network. We note that the FPND and comparing distributions as above is conceptually equivalent, except here instead of physically meaningful and easily interpretable features, we are comparing those found to be statistically optimum for distinguishing jets.

Two common metrics for evaluating point cloud generators are coverage (COV) and minimum matching distance (MMD) [21]. Both involve finding the closest point cloud in a sample X to each cloud in another sample Y, based on a metric such as the Chamfer distance or the earth mover's distance. Coverage is defined as the fraction of samples in X which were matched to one in Y, measuring thus the diversity of the samples in Y relative to X, and MMD is the average distance between matched samples, measuring the quality of samples. We use both, and due to drawbacks of the Chamfer distance pointed out in Ref. [21], for our distance metric choose only the analogue of the earth mover's distance for particle clouds a.k.a. the energy mover's distance (EMD) [31]. We discuss the effectiveness and complementarity of all four metrics in evaluating clouds in Sec. 5.

4 MPGAN Architecture

In this section, we describe the architecture of our MPGAN model (Fig. 2), noting particle cloud-motivated aspects compared to its r-GAN and GraphCNN-GAN predecessors.

Message passing. Jets originate from a single source particle decaying and hadronizing, hence they end up with important high-level jet features and a rich global structure, known as the jet substructure [1], stemming from the input particle. Indeed any high-level feature useful for analyzing jets, such as jet mass or multi-particle correlations, is necessarily global [28]. Because of this, while past work in learning on point clouds [9, 32, 33], including GraphCNN-GAN, has used a locally connected graph structure and convolutions for message passing, we choose a fully connected graph, equally weighting messages from all particles in the clouds. Rather than subtracting particle features for messages between particles, useful in graph convolutions to capture local differences within a neighborhood, the respective features are concatenated to preserve the global structure. In addition, the difference between particle features is only physically meaningful if they are in the 4-vector

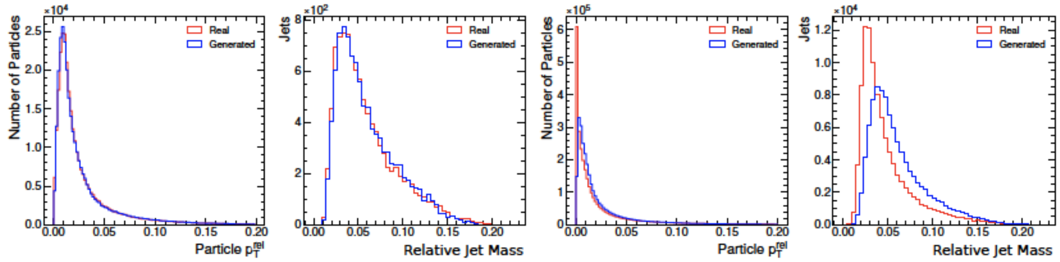

Figure 3: Particle $p_{\mathrm{T}}^{\mathrm{rel}}$ and relative jet mass distributions of real jets and those generated by MPGAN without our masking strategy. Left: gluon, right: light quark jets. We see that while for gluon jets the generator learns distributions correctly, it struggles to learn the discontinuous spike, due to the zero-padded particles, in the light quark $p_{\mathrm{T}}^{\mathrm{rel}}$ distribution. This also leads to a distorted mass distribution.

representation of the Lorentz group. During the update step in the message passing we find it empirically beneficial to incorporate a residual connection to previous particle features.

The operation can be described as follows. For an N-particle cloud $J^t = \{p_1^t, \cdots, p_N^t\}$ after t iterations of message passing ($t = 0$ corresponds to the original input cloud), each particle p_i^t is represented by features h_i^t. One iteration of message passing is then defined as

$$\mathbf{m}_{ij}^{t+1} = f_e^{t+1}(\mathrm{h}_i^t \oplus \mathrm{h}_j^t) \tag{1}$$

$$\mathrm{h}_i^{t+1} = f_n^{t+1}(\mathrm{h}_i^t \oplus \sum_{j \in J} \mathbf{m}_{ij}^{t+1}), \tag{2}$$

where $\mathbf{m}_{ij}^{t+1}$ is the message vector sent from particle j to particle i, h_i^{t+1} are the updated features of particle i, and f_e^{t+1} and f_n^{t+1} are arbitrary functions which, in our case, are implemented as multilayer perceptrons (MLPs) with 3 FC layers.

Generator. We test two initializations of a particle cloud for the MPGAN generator: (1) directly initializing the cloud with N particles with L randomly sampled features, which we refer to as the MP generator, and (2) inputting a single Z-dimensional latent noise vector and transforming it via an FC layer into an $N \times L$-dimensional matrix, which we refer to as the MP-Latent-FC (MP-LFC) generator. The MP-LFC uses a latent space which can intuitively be understood as representing the initial source particle's features along with parameters to capture the stochasticity of the jet production process. Due to the complex nature of this process, however, we posit that this global, flattened latent space cannot capture the full phase space of individual particle features. Hence, we introduce the MP generator, which samples noise directly per particle, and find that it outperforms MP-LFC (Table 2).

Discriminator. We find the MP generator, in conjunction with a PointNet discriminator, to be a significant improvement on every metric compared to FC and GraphCNN generators. However, the jet-level features are not yet reproduced to a high enough accuracy (Sec. 5). While PointNet is able to capture global structural information, it can miss the complex interparticle correlations in real particle clouds. We find we can overcome this limitation by incorporating message passing in the discriminator as well as in the generator. Concretely, our MP discriminator receives the real or generated cloud and applies MP layers to produce intermediate features for each particle, which are then aggregated via a feature-wise average-pooling operation and passed through an FC layer to output the final scalar feature. We choose 2 MP layers for both networks.

Variable-sized point clouds. In JetNet, jets with fewer than 30 particles are zero-padded to fill the 30-particle point cloud. Such zero-padded particles pose a problem for the generator, which is not able to learn this sharp discontinuity in the jet constituents, (Fig. 3).

To address this, we introduce an additional binary "masking" particle feature classifying the particle as genuine or zero-padded. Particles in the zero-padded class are ignored entirely in the message passing and pooling operations. The MP generator adds mask features to the initial particle cloud, using an additional input of the size of the jet N, sampled from the real distribution, before the

Table 1: W_1 distances between real jet mass (W_1^{M}), averaged particle features (W_1^{P}), and averaged jet EFPs (W_1^{EFP}) distributions calculated as a baseline, for three classes of jets.

Jet class	W_1^{M} ($\times 10^{-3}$)	W_1^{P} ($\times 10^{-3}$)	W_1^{EFP} ($\times 10^{-5}$)
Gluon	0.7 ± 0.2	0.44 ± 0.09	0.62 ± 0.07
Top quark	0.51 ± 0.07	0.55 ± 0.07	1.1 ± 0.1
Light quark	0.5 ± 0.1	0.5 ± 0.1	0.46 ± 0.04

message passing layers based on sorting in particle feature space. A variety of alternative masking strategies experimented with, out of which this was most successful, are discussed in App. C.

5 Experiments

Evaluation. We use four techniques discussed in Sec. 3.1 for evaluating and comparing models. The distributions of physical particle and jet features are compared visually and quantitatively using the Wasserstein-1 (W_1) distance between them. For ease of evaluation, we report (1) the average scores of the three particle features (W_1^{P}) η^{rel}, ϕ^{rel}, and $p_{\mathrm{T}}^{\mathrm{rel}}$, (2) the jet mass ($W_1^{\mathrm{M}}$), and (3) the average of a subset of the EFPs[6] (W_1^{EFP}), which together provide a holistic picture of the low- and high-level aspects of a jet. The W_1 distances are calculated for each feature between random samples of 10,000 real and generated jets, and averaged over 5 batches. Baseline W_1 distances are calculated between two sets of randomly sampled real jets with 10,000 samples each, and are listed for each feature in Table 1. The real samples split 70/30 for training/evaluation.

We train[7] ParticleNet for classification on our dataset to develop the FPND metric. FPND is calculated between 50,000 random real and generated samples, based on the activations of the first FC layer in our trained model[8]. Coverage and MMD are calculated between 100 real and generated samples.

Results. On each of JetNet's three classes, we test r-GAN's FC and the GraphCNN generators with r-GAN's FC and the PointNet-Mix discriminators, and compare them to MPGAN's MP generator and discriminator models, including the MP and MP-LFC generator variations. Training and implementation details for each can be found in App. B.

We choose model parameters which, during training, yield the lowest W_1^{M} score. This is because (1) W_1 scores between physical features are more relevant for physics applications than the other three metrics, and (2) qualitatively we find it be a better discriminator of model quality than particle feature or EFP scores. Table 2 lists the scores for each model and class, and Fig. 4 shows plots of selected feature distributions of real and generated jets, for the best performing FC, GraphCNN, and MP generators. Overall we find that MPGAN is a significant improvement over the best FC and GraphCNN models, particularly for top and light quark jets. This is evident both visually and quantitatively in every metric, especially jet W_1s and FPND, apart from W_1^{P} where only the FC generator and PointNet discriminator (FC + PointNet) combination is more performant.

Real baseline comparison. We find that MPGAN's jet-level W_1 scores all fall within error of the baselines in Table 1, while those of FC and GraphCNN generators are several standard deviations away. This is particularly an issue with complex top quark particle clouds, where we can see in Fig. 4 neither generator is able to learn the bimodal jet feature distributions, and smaller light quark clouds where we see distortion of jet features due to difficulty reproducing the zero-padded particle features. No model is able to achieve particle-level scores close to the baseline, and only those of the FC + PointNet combination and MPGAN are of the same order of magnitude. We conclude that MPGAN reproduces the physical observable distributions to the highest degree of accuracy, but note, however, that it requires further improvement in particle feature reconstruction before it is ready for practical application in HEP.

[6]We choose 5 EFPs corresponding to the set of loopless multigraphs with 4 vertices and 4 edges.

[7]Details of the training and architecture are given in App. B.3.

[8]The trained ParticleNet model can be found in the supplementary materials.

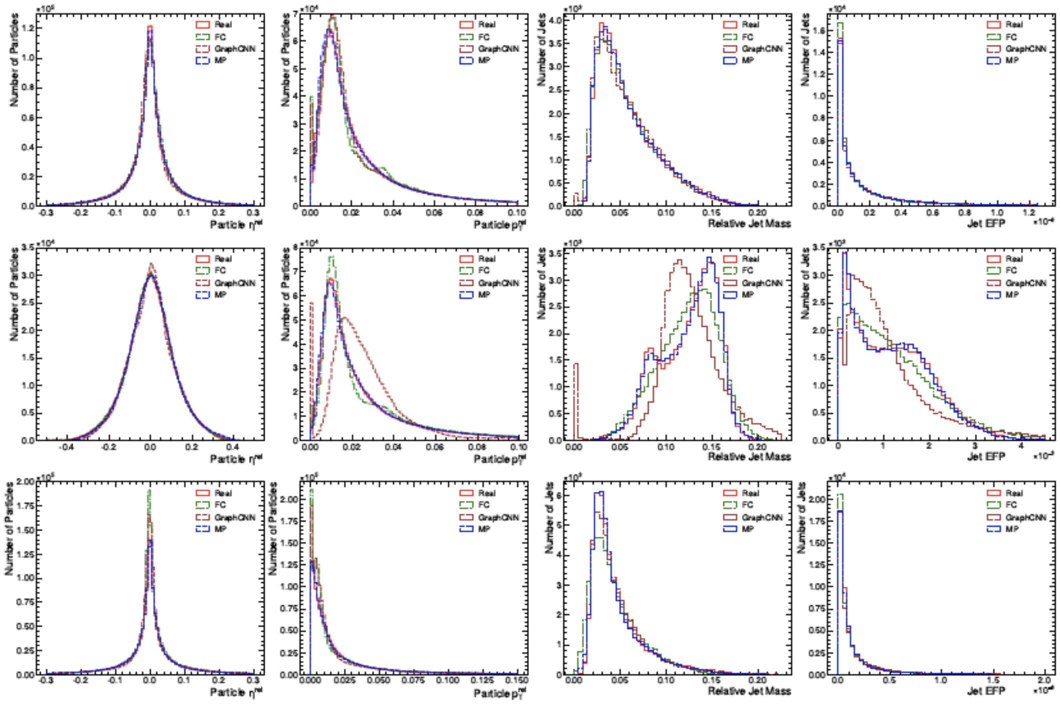

Figure 4: Comparison of real and generated distributions for a subset of jet and particle features. We use the best performing model for each of the FC, GraphCNN, and MP generators, as per Table 2. Top: gluon jet features, Middle: top quark jet, Bottom: lighter quark jets.

Architecture discussion. To disentangle the effectiveness of the MP generator and discriminator, we train each individually with alternative counterparts (Table 2). With the same PointNet discriminator, the GraphCNN generator performs worse than the simple FC generator for every metric on all three datasets. The physics-motivated MP generator on the other hand outperforms both on the gluon and top quark datasets, and significantly so on the jet-level W_1 scores and the FPND. We note, however, that the MP generator is not a significant improvement over FC or GraphCNN with an FC discriminator. Holding the generator fixed, the PointNet discriminator performs significantly better over the FC for all metrics. With the FC and GraphCNN generators, PointNet is also an improvement over the MP discriminator. With an MP generator, the MP discrimimator is more performant on jet-level W_1 and FPND scores but, on the gluon and top quark datasets, degrades W_1^{P} relative to PointNet.

We learn from these three things: (1) a generator or discriminator architecture is only as effective as its counterpart — even though the MPGAN combination is the best overall, when paired with a network which is not able to learn complex substructure, or which breaks the permutation symmetry, neither the generator or discriminator is performant, (2) for high-fidelity jet feature reconstruction, both networks must be able to learn complex multi-particle correlations — however, this can come at the cost of low-level feature accuracy, and (3) MPGAN's masking strategy is highly effective as both MP networks are improvements all around on light quark jets.

Particle cloud evaluation metrics. Each metric proposed here has unique merit. We see that models with low W_1 scores relative to the baseline have the best coverage and MMD scores as well. This indicates that the W_1 metrics are sensitive to both mode collapse (measured by coverage) — this is expected as, in terms of feature distributions, mode collapse manifests as differing supports, to which the W_1 distance is reasonably sensitive, as well as to individual sample quality (measured by MMD) — this supports our claim that recovering jet feature distributions implies accurate learning of individual cloud structure. Together this suggests that low W_1 scores are alone sufficient to validate sample quality and against mode collapse, and justifies our criteria that a practical ML simulation alternative have W_1 scores close to the baselines in Table 2. However, MMD and coverage, being focused tests of these aspects of generation, are useful for understanding failure modes.

Table 2: Six evaluation scores on different generator and discriminator combinations. Lower is better for all metrics except COV.

Jet class	Generator	Discriminator	W_1^{M} ($\times 10^{-3}$)	W_1^{P} ($\times 10^{-3}$)	W_1^{EFP} ($\times 10^{-5}$)	FPND	COV ↑	MMD
Gluon	FC	FC	1.6 ± 0.2	17.7 ± 0.2	8 ± 5	502	0.20	0.053
	GraphCNN	FC	6.0 ± 0.2	48.1 ± 1.0	50 ± 30	540	0.08	0.122
	FC	PointNet	1.2 ± 0.2	1.3 ± 0.1	1.4 ± 0.7	633	0.47	0.038
	GraphCNN	PointNet	1.9 ± 0.2	15.8 ± 1.9	5k ± 2k	280k	0.48	0.040
	MP	MP	0.6 ± 0.2	1.8 ± 0.2	$\mathbf{0.9 \pm 0.4}$	**3.2**	**0.54**	**0.036**
	MP-LFC	MP	$\mathbf{0.5 \pm 0.1}$	2.0 ± 0.1	1.0 ± 0.5	3.3	0.51	**0.036**
	FC	MP	4.3 ± 0.3	21.2 ± 0.1	8 ± 4	1.2k	0.10	0.084
	GraphCNN	MP	2.6 ± 0.2	9.6 ± 0.1	11 ± 5	543	0.39	0.048
	MP	FC	1.3 ± 0.1	3.5 ± 0.1	1.5 ± 0.7	506	0.42	0.040
	MP	PointNet	0.7 ± 0.1	$\mathbf{1.2 \pm 0.2}$	1.8 ± 0.8	21	**0.54**	**0.036**
Top quark	FC	FC	4.8 ± 0.1	13.8 ± 0.2	30 ± 20	168	0.28	0.103
	GraphCNN	FC	12.3 ± 0.4	73 ± 2	3k ± 1k	147	0.15	0.171
	FC	PointNet	2.6 ± 0.1	$\mathbf{1.5 \pm 0.2}$	8 ± 3	225	0.56	0.076
	GraphCNN	PointNet	10.8 ± 0.5	37 ± 3	39 ± 18	2M	0.39	0.084
	MP	MP	$\mathbf{0.7 \pm 0.1}$	2.1 ± 0.2	$\mathbf{1.8 \pm 0.8}$	6.4	0.56	**0.071**
	MP-LFC	MP	0.9 ± 0.3	2.2 ± 0.1	2 ± 1	**2.9**	0.55	**0.071**
	FC	MP	6.8 ± 0.2	39.2 ± 0.1	15 ± 8	272	0.25	0.120
	GraphCNN	MP	6.8 ± 0.1	7.9 ± 0.2	70 ± 30	149	0.51	0.081
	MP	FC	6.4 ± 0.1	45.2 ± 0.1	19 ± 9	253	0.08	0.120
	MP	PointNet	1.0 ± 0.3	$\mathbf{1.5 \pm 0.1}$	5 ± 2	12	**0.58**	**0.071**
Light quark	FC	FC	5.9 ± 0.3	15.6 ± 0.2	4 ± 2	16k	0.19	0.055
	GraphCNN	FC	3.1 ± 0.3	38 ± 1	1.0k ± 0.4k	15k	0.14	0.057
	FC	PointNet	2.9 ± 0.2	$\mathbf{1.5 \pm 0.1}$	2 ± 1	22k	0.36	**0.026**
	GraphCNN	PointNet	4.2 ± 1.6	3.9 ± 0.2	20M ± 10M	19k	0.37	0.031
	MP	MP	$\mathbf{0.6 \pm 0.1}$	2.1 ± 0.1	$\mathbf{0.9 \pm 0.4}$	**2.4**	**0.54**	**0.026**
	MP-LFC	MP	1.0 ± 0.2	2.5 ± 0.1	1.0 ± 0.5	3.8	0.48	**0.026**
	FC	MP	6.1 ± 0.1	16.6 ± 0.1	4 ± 2	20k	0.11	0.070
	GraphCNN	MP	3.5 ± 0.2	14.0 ± 0.1	9 ± 5	15k	0.26	0.038
	MP	FC	1.3 ± 0.2	5.0 ± 0.1	2 ± 1	22k	0.36	0.031
	MP	PointNet	3.2 ± 0.2	22.0 ± 0.1	5 ± 2	3.6k	0.22	0.035

Additionally, we observe that models with the lowest FPND scores on each dataset also possess the lowest W_1 scores, as ParticleNet's deep activations are likely to be correlated with jet features. Overall, we expect FPND to be a better measure of quality since the features being compared are statistically optimum for classifying jets. However, as these features are not necessarily meaningful and less interpretable from a physics-perspective, W_1 scores are also valuable.

6 Summary

In this work, we publish JetNet: a novel particle cloud dataset to (1) facilitate machine learning (ML) research in high energy physics (HEP), and (2) provide a novel point-cloud-style dataset, containing rich underlying physics, for the ML community to experiment with, particularly in the context of generative models. We apply existing state-of-the-art point cloud generative models to JetNet, and propose types of physics- and computer-vision-inspired metrics to evaluate the generated clouds. We find that existing models are not performant on a number of metrics, and fail to reproduce high-level jet features, which is the most important aspect for HEP. Our new message-passing generative adversarial network (MPGAN) model, designed to capture complex global structure and handle variable-sized clouds significantly improves performance in this area, as well as other metrics. We propose MPGAN as a new baseline model on JetNet and invite others to improve upon it.

Impact An accurate and fast ML particle cloud generator will have significant impact in (1) lowering the computational and energy cost of HEP research, as well as (2) increasing precision and sensitivity to new physics at the Large Hadron Collider and future colliders by providing more high-quality simulated data samples. One negative consequence may be a loss of interpretability, and hence trustability, of the particle production generative model, which may ultimately increase uncertainties — though the metrics we propose should mitigate against this. More broadly, further advancements in the field of ML point cloud generation may result in fake visual data generation for proliferation of misinformation and impersonation/identity theft.

Acknowledgments and Disclosure of Funding

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369). R. K. was partially supported by an IRIS-HEP fellowship through the U.S. National Science Foundation (NSF) under Cooperative Agreement OAC-1836650. J. D. is supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187 and by the DOE, Office of Advanced Scientific Computing Research under Award No. DE-SC0021396 (FAIR4HEP). B. O and T. T are supported by grant 2018/25225-9, São Paulo Research Foundation (FAPESP). B. O was also partially supported by grants #2018/01398-1 and #2019/16401-0, São Paulo Research Foundation (FAPESP). J-R. V. is partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369) and by the DOE, Office of Science, Office of High Energy Physics under Award No. DE-SC0011925, DE-SC0019227, and DE-AC02-07CH11359. 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. Funding for cloud credits was supported by NSF Award #1904444 Internet2 supported E-CAS Exploring Clouds to Accelerate Science.

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] L. de Oliveira, M. Paganini, and B. Nachman, "Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis", *Comput. Softw. Big Sci.* **1** (2017) 4, doi:10.1007/s41781-017-0004-6, arXiv:1701.05927.

[3] R. K. Ellis, W. J. Stirling, and B. R. Webber, "QCD and collider physics". Cambridge monographs on particle physics, nuclear physics, and cosmology. Cambridge University Press, Cambridge, 2003. Photography by S. Vascotto. doi:10.1017/CBO9780511628788.

[4] T. Sjöstrand et al., "An introduction to PYTHIA 8.2", *Comput. Phys. Commun.* **191** (2015) 159, doi:10.1016/j.cpc.2015.01.024, arXiv:1410.3012.

[5] J. Bellm et al., "HERWIG 7.0/HERWIG ++3.0 release note", *Eur. Phys. J. C* **76** (2016), no. 4, 196, doi:10.1140/epjc/s10052-016-4018-8, arXiv:1512.01178.

[6] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjostrand, "Parton Fragmentation and String Dynamics", *Phys. Rept.* **97** (1983) 31, doi:10.1016/0370-1573(83)90080-7.

[7] J. Cogan, M. Kagan, E. Strauss, and A. Schwarztman, "Jet-Images: Computer Vision Inspired Techniques for Jet Tagging", *JHEP* **02** (2015) 118, doi:10.1007/JHEP02(2015)118, arXiv:1407.5675.

[8] L. de Oliveira et al., "Jet-images — deep learning edition", *JHEP* **07** (2016) 069, doi:10.1007/JHEP07(2016)069, arXiv:1511.05190.

[9] H. Qu and L. Gouskos, "ParticleNet: Jet Tagging via Particle Clouds", *Phys. Rev. D* **101** (2020), no. 5, 056019, doi:10.1103/PhysRevD.101.056019, arXiv:1902.08570.

[10] J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph neural networks in particle physics", *Mach. Learn.: Sci. Technol.* **2** (2021), no. 2, 021001, doi:10.1088/2632-2153/abbf9a, arXiv:2007.13681.

[11] CMS Collaboration, "Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques", *JINST* **15** (2020), no. 06, P06005, doi:10.1088/1748-0221/15/06/P06005, arXiv:2004.08262.

[12] A. X. Chang et al., "ShapeNet: An Information-Rich 3D Model Repository", Technical Report, 2015. arXiv:1512.03012.

[13] Anonymous, "JetNet", 05, 2021. doi:10.5281/zenodo.4834876, https://zenodo.org/record/4834876.

[14] M. Pierini, J. M. Duarte, N. Tran, and M. Freytsis, "hls4ml LHC jet dataset (30 particles)", 01, 2020. doi:10.5281/zenodo.3601436.

[15] 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), no. 1, 014021, doi:10.1103/PhysRevD.97.014021, arXiv:1712.10321.

[16] ATLAS Collaboration, "Fast simulation of the ATLAS calorimeter system with Generative Adversarial Networks", ATLAS Software Public Note, 11, 2020.

[17] M. Erdmann, J. Glombitza, and T. Quast, "Precise simulation of electromagnetic calorimeter showers using a Wasserstein generative adversarial network", *Computing and Software for Big Science* **3** (01, 2019) doi:10.1007/s41781-018-0019-7.

[18] F. Carminati et al., "Generative Adversarial Networks for fast simulation", *J. Phys. Conf. Ser.* **1525** (2020), no. 1, 012064, doi:10.1088/1742-6596/1525/1/012064.

[19] Y. Lu, J. Collado, D. Whiteson, and P. Baldi, "Sparse autoregressive models for scalable generation of sparse images in particle physics", *Phys. Rev. D* **103** (2021), no. 3, 036012, doi:10.1103/PhysRevD.103.036012, arXiv:2009.14017.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, eds., volume 37, p. 448. PMLR, 2015. arXiv:1502.03167.

[21] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning representations and generative models for 3D point clouds”, in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, eds., volume 80, p. 40. PMLR, 2018.

[22] D. Valsesia, G. Fracastoro, and E. Magli, “Learning localized generative models for 3D point clouds via graph convolution”, in *International Conference on Learning Representations (ICLR) 2019*. 2019.

[23] D. W. Shu, S. W. Park, and J. Kwon, “3D point cloud generative adversarial network based on tree structured graph convolutions”, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, p. 3858. 2019. arXiv:1905.06292. doi:10.1109/ICCV.2019.00396.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July, 2017.

[25] H. Wang et al., “Rethinking sampling in 3D point cloud generative adversarial networks”, (2020). arXiv:2006.07029.

[26] 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, p. 1263. PMLR, 2017. arXiv:1704.01212.

[27] CMS Collaboration, “Recent Developments in CMS Fast Simulation”, *PoS* **ICHEP2016** (2016) 181, doi:10.22323/1.282.0181, arXiv:1701.03850.

[28] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Energy flow polynomials: A complete linear basis for jet substructure”, *JHEP* **04** (2018) 013, doi:10.1007/JHEP04(2018)013, arXiv:1712.07124.

[29] A. Borji, “Pros and cons of GAN evaluation measures: New developments”, (2021). arXiv:2103.09396.

[30] M. Heuselet al., “GANs trained by a two time-scale update rule converge to a local nash equilibrium”, in *Advances in Neural Information Processing Systems*, I. Guyon et al., eds., volume 30, p. 6626. Curran Associates, Inc., 2017. arXiv:1706.08500.

[31] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Metric Space of Collider Events”, *Phys. Rev. Lett.* **123** (2019), no. 4, 041801, doi:10.1103/PhysRevLett.123.041801, arXiv:1902.02346.

[32] Y. Wang et al., “Dynamic graph CNN for learning on point clouds”, *ACM Trans. Graph.* **38** (2019) doi:10.1145/3326362, arXiv:1801.07829.

[33] F. Monti et al., “Geometric deep learning on graphs and manifolds using mixture model CNNs”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 5425. 2017. arXiv:1611.08402. doi:10.1109/CVPR.2017.576.

[34] 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, doi:10.1007/JHEP07(2014)079, arXiv:1405.0301.

[35] R. D. Ball et al., “Parton distributions with LHC data”, *Nucl. Phys. B* **867** (2013) 244–289, doi:10.1016/j.nuclphysb.2012.10.003, arXiv:1207.1303.

[36] P. Skands, S. Carrazza, and J. Rojo, “Tuning PYTHIA 8.1: The Monash 2013 Tune”, *Eur. Phys. J. C* **74** (2014), no. 8, 3024, doi:10.1140/epjc/s10052-014-3024-y, arXiv:1404.5630.

[37] M. Cacciari, G. P. Salam, and G. Soyez, “The anti-k_{T} jet clustering algorithm”, *JHEP* **04** (2008) 063, doi:10.1088/1126-6708/2008/04/063, arXiv:0802.1189.

[38] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet user manual”, *Eur. Phys. J. C* **72** (2012) 1896, doi:10.1140/epjc/s10052-012-1896-2, arXiv:1111.6097.

[39] M. Cacciari and G. P. Salam, “Dispelling the n^3 myth for the k_{T} jet-finder”, *Phys. Lett.* **B641** (2006) 57, doi:10.1016/j.physletb.2006.08.037, arXiv:hep-ph/0512210.

[40] X. Mao et al., “Multi-class generative adversarial networks with the L2 loss function”, (2016). arXiv:1611.04076.

[41] N. Srivastavaet al., “Dropout: A simple way to prevent neural networks from overfitting”, *J. Mach. Learn. Res.* **15** (2014) 1929.

[42] I. Gulrajaniet al., "Improved training of Wasserstein GANs", in *Advances in Neural Information Processing Systems*, I. Guyon et al., eds., volume 30, p. 5767. Curran Associates, Inc., 2017. arXiv:1704.00028.

[43] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks", in *6th International Conference on Learning Representations*. 2018. arXiv:1802.05957.

[44] F. Schäfer, H. Zheng, and A. Anandkumar, "Implicit competitive regularization in GANs", (2019). arXiv:1910.05852.

[45] T. Karraset al., "Training generative adversarial networks with limited data", in *Advances in Neural Information Processing Systems*, H. Larochelle et al., eds., volume 33, p. 12104. Curran Associates, Inc., 2020. arXiv:2006.06676.

[46] N.-T. Tran et al., "On data augmentation for GAN training", *IEEE Trans. Image Process.* **30** (2021) 1882, doi:10.1109/tip.2021.3049346, arXiv:2006.05338.

[47] Z. Zhao et al., "Image augmentations for GAN training", (2020). arXiv:2006.02595.

A JetNet Generation

The so-called parton-level events are first produced at leading-order using MADGRAPH5_aMCATNLO 2.3.1 [34] with the NNPDF 2.3LO1 parton distribution functions [35]. To focus on a relatively narrow kinematic range, the transverse momenta of the partons and undecayed gauge bosons are generated in a window with energy spread given by $\Delta p_{\mathrm{T}}/p_{\mathrm{T}} = 0.01$, centered at 1 TeV. These parton-level events are then decayed and showered in PYTHIA 8.212 [4] with the Monash 2013 tune [36], including the contribution from the underlying event. For each original particle type, 200,000 events are generated. Jets are clustered using the anti-k_{T} algorithm [37], with a distance parameter of $R = 0.8$ using the FASTJET 3.1.3 and FASTJET CONTRIB 1.027 packages [38, 39]. Even though the parton-level p_{T} distribution is narrow, the jet p_{T} spectrum is significantly broadened by kinematic recoil from the parton shower and energy migration in and out of the jet cone. We apply a restriction on the measured jet p_{T} to remove extreme events outside of a window of $0.8\,\mathrm{TeV} < p_{\mathrm{T}} < 1.6\,\mathrm{TeV}$ for the $p_{\mathrm{T}} = 1\,\mathrm{TeV}$ bin.

B Training and Implementation Details

PyTorch code and trained parameters for each model in Table 2 is provided in the supplementary materials. Models were trained and hyperparameters optimized on clusters of GeForce RTX 2080 Ti and Tesla V100 GPUs.

B.1 MPGAN

We use the least squares loss function [40] and the RMSProp optimizer with a two time-scale update rule [30] with a learning rate (LR) for the discriminator three times greater than that of the generator. The absolute rate differed per jet type. We use LeakyReLU activations (with negative slope coefficient 0.2) after all MLP layers except for the final generator and discriminator outputs where tanh and sigmoid activations respectively are applied. We attempted discriminator regularization to alleviate mode collapse via dropout [41], batch normalization [20], a gradient penalty [42], spectral normalization [43], adaptive competitive gradient descent [44] and data augmentation of real and generated graphs before the discriminator [45–47]. Apart from dropout (with fraction 0.5), none of these demonstrated significant improvement with respect to mode dropping or cloud quality.

We use a generator LR of 10^{-3} and train for 2000 epochs for gluon jets, 2×10^{-3} and 2000 epochs for top quark jets, and 0.5×10^{-3} and 2500 epochs for light quark jets. We use a batch size of 256 for all jets.

B.2 rGAN, GraphCNNGAN, and PointNet-Mix

For rGAN and GraphCNNGAN we train two variants: (1) using the original architecture hyperparameters as in Refs. [21, 22] for the 2048-node point clouds, and (2) using hyperparameters scaled down to 30-node clouds. The latter variant performed better for both models, and its scores are which are reported. LRs, batch sizes, loss functions, gradient penalties, optimizers, ratios of critic to generator updates, activations, number of epochs, are the same as in the original paper and code. We use the architecture defined in [25] for the PointNet-Mix discriminator.

B.3 FPND

Apart from the number of input particle features (three in our case, excluding the mask feature), we use the same ParticleNet architecture as in the original paper. We find training with the Adam optimizer, LR 10^{-4}, for 30 epochs outperformed the original recommendations on our dataset. Activations after the first fully connected layer, pre-ReLU, are used for FPND.

C Masking

We experiment with 5 masking strategies, out of which the one described in Sec. 4 was most successful. The four alternatives, which all involve the generator learning the mask without any external input, are shown in Fig. 5.

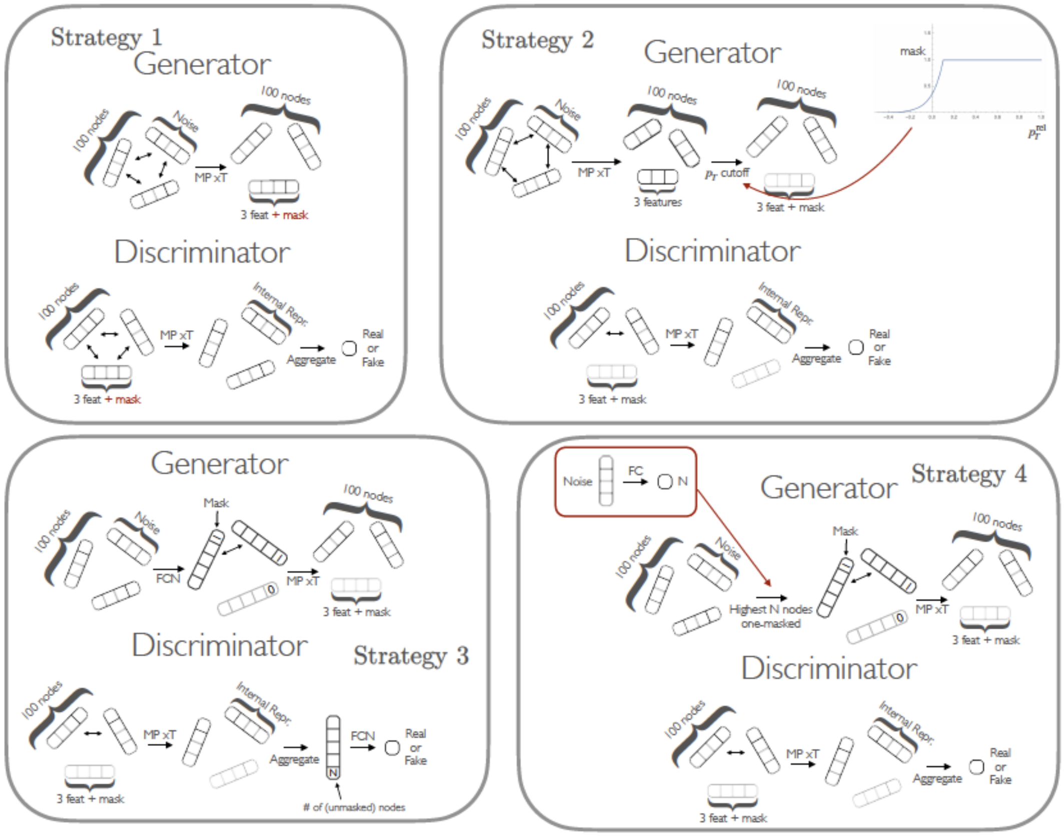

Figure 5: The four alternative masking strategies which we test.

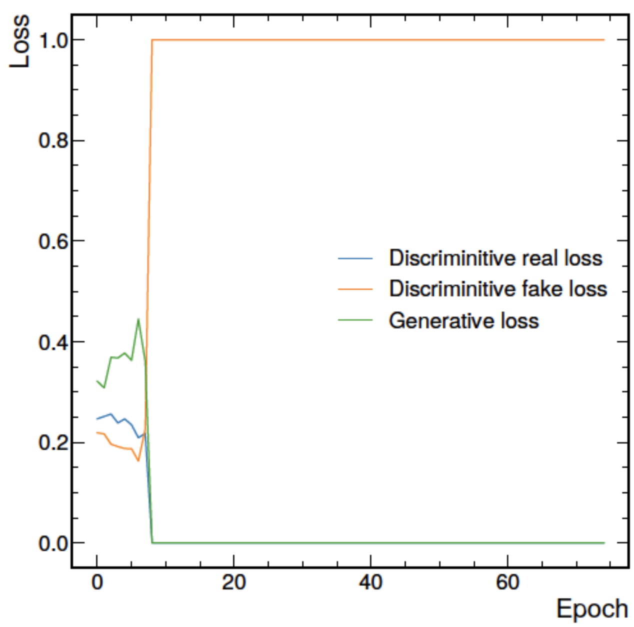

Figure 6: Typical loss curves on light quark jets with the four strategies, with the one depicted specifically from strategy 3.

Strategy 1 treats the mask homogenously as simply an extra feature to learn. A variation of this weights the nodes in the discriminator the mask.

In strategy 2, a mask is calculated for each generated particle using a function of $p_{\mathrm{T}}^{\mathrm{rel}}$, based on an empirical minimum cutoff in the dataset. We tested a step function and a continuous mask function as in the figure. The discriminator functions as the standard MP one described in Sec. 4.

Strategy 3 sees the generator applying an FC layer per particle in the initial cloud to learn the mask. Standard MP disciminator is used, as well as a variant with the number of unmasked nodes in the clouds added as an extra feature to the FC layer. In 1 and 3 we test learning both binary and continuous masks.

Finally, in strategy 4, we train an auxiliary network to choose a number of particles to mask (as opposed to sampling from the real distribution), which is then passed into the standard MP generator.

We find that while all of these strategies, or variations thereof, are able to successfully produce gluon jets, the training for each diverges in the fashion depicted in Fig. 6 (despite using all the discriminator regularization methods mentioned in App. B. We conclude that learning the number of particles is a significant challenge for the generator, while being a easy feature to discriminate with, and so to equalize this we use the strategy in Sec. 4 where the number of particles to produce is sampled from the real distribution instead.