Towards a deep learning model for hadronization

Aishik Ghosh,[1,2] Xiangyang Ju,[2] Benjamin Nachman,[2,3] and Andrzej Siodmok[4,*]

[1]*Department of Physics and Astronomy, University of California, Irvine, California 92697, USA*
[2]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*
[3]*Berkeley Institute for Data Science, University of California, Berkeley, California 94720, USA*
[4]*Jagiellonian University, ul. Lojasiewicza 11, 30-348 Kraków, Poland*

(Received 15 April 2022; accepted 4 November 2022; published 28 November 2022)

Hadronization is a complex quantum process whereby quarks and gluons become hadrons. The widely used models of hadronization in event generators are based on physically inspired phenomenological models with many free parameters. We propose an alternative approach whereby neural networks are used instead. Deep generative models are highly flexible, differentiable, and compatible with graphical processing units. We make the first step towards a data-driven machine learning-based hadronization model. In that step, we replace a component of the hadronization model within the Herwig event generator (cluster model) with HADML, a computer code implementing a generative adversarial network. We show that a HADML is capable of reproducing the kinematic properties of cluster decays. Furthermore, we integrate it into Herwig to generate entire events that can be compared with the output of the public Herwig simulator as well as with e^+e^- data.

DOI: 10.1103/PhysRevD.106.096020

I. INTRODUCTION

Simulations are essential tools for nearly all aspects of data analysis at particle colliders (see e.g., Ref. [1]). These simulations are rooted in particle and nuclear physics and must model a large range in energy scales. At the smallest distance scales, various forms of perturbation theory offer accurate, first-principles descriptions of hard-scatter particle reactions and collinear parton shower radiation. The conversion from quarks and gluons to hadrons is performed using hadronization models. Such approaches are physically inspired but are ultimately phenomenological models with many parameters that must be fit to data. There are currently two main hadronization models, each inspired by a different description of strong dynamics in the low-energy region. The linear confining potential motivated the string model [2,3] implemented in Pythia [4,5] and preconfinement [6,7] inspired the cluster model [8] in Herwig [9–12] and Sherpa [13,14]. In both models, there is an intermediate object between quarks/gluons and hadrons. This intermediate object (string or cluster) takes, as input, the kinematic and flavor information from quarks and gluons and then has an approximately universal fragmentation into different hadron species that carry some fraction of the object's momentum.

While existing hadronization models have been used successfully in a large number of phenomenological and experimental studies at the Large Hadron Collider and beyond, there is also significant room for innovation. Existing models are not flexible enough to describe all of the properties of hadronization needed for physics measurements and searches (see e.g., Ref. [15]) and each model has its own strengths and limitations. While the string model offers a predictive framework of how its space-time motion and breakup translate into an energy-momentum distribution of the primary hadrons, its weakness lies in the description of the flavor properties of hadrons. The cluster model on the other hand has a simpler energy-momentum description but better flavor composition [1]. In addition, the busy environment of high-energy hadronic collisions leads to nontrivial collective effects [16–19] that are currently not simulated in the basic versions of either model. Even so, these models still have a large number of parameters that need to be fit to data, which are adjusted ('tuned') using semiautomated programs like Professor [20]. Existing tuning methods are not able to process high-dimensional observables or simultaneously tune many parameters because they rely on relatively simple *surrogate models* to approximate the dependence of the data on the model. A number of recently proposed automated tuning approaches employ sophisticated surrogate models [21–23], but they all still require approximating complex relationships in high dimensions

*Corresponding author.
andrzej.siodmok@uj.edu.pl

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

and therefore are often limited to relatively low-dimensional parameter spaces.

One natural alternative to the existing hadronization simulations is deep generative modeling. Machine learning-based generators are highly flexible and differentiable by construction, which can aid parameter tuning. Three standard approaches to deep generative models include generative adversarial networks (GANs) [24,25], (variational) autoencoders (VAEs) [26,27], and normalizing flows (NFs) [28,29]. While first proposed in high-energy physics (HEP) to emulate an entire parton shower [30] or detector simulations [31,32], deep-generative models have now been proposed for many aspects of HEP simulations including matrix element generation [33–39], parton showers [30,40–47], detector simulation [31,32,48–81], and more (see Ref. [82–84] for reviews). Using neural networks for modeling nonpertrubative inputs has a long history in the context of parton distribution functions (PDFs) (Ref. [85] through Ref. [86]). Similarly to hadronization models, PDFs cannot be calculated using perturbation theory. In contrast to hadronization, PDFs are modeled as deterministic functions that are evolved in energy scale using perturbation theory [87–89].

Building a complete machine learning-based hadronization model is a long-term program that will require a number of intermediate milestones. The ultimate model will take as input partons and output hadrons, with input/output containing kinematic and flavor information. This model will be trained directly on data so that it can be more precise than existing models and is not bound by their assumptions. On the path towards a fully flexible, data-optimized, machine-learning based hadronization model, we demonstrate the first step by training a GAN to mimic a component of the cluster hadronization implementation in Herwig. In particular, we replace part of the cluster decayer inside Herwig with a GAN using the Open Neural Network Exchange (ONNX) [90] interface to call the neural network inside the C++ code. This GAN-based cluster decayer, HADML, is trained on Herwig. In particular, our contributions are twofold; within the context of cluster hadronization, we show that a neural network can mimic the cluster fragmentation and that this model can be integrated into a full parton shower Monte Carlo program to generate full events. Future work will add additional complexity (cluster to cluster decays, color reconnection of clusters [91–93], etc.) and will ultimately lead to a model that can be trained (tuned) on data. This ultimate model will benefit from new, high-dimensional future measurements [94] that will provide the necessary constraining power for the flexible neural network approaches.

This paper is organized as follows. Section II briefly introduces details of the Herwig Monte Carlo event generator and how we interface a GAN in the hadronization stage. Then, Sec. III presents the first numerical results with the HADML hadronization model. The paper ends with conclusions and an outlook in Sec. IV.

II. METHODS

A. Dataset

The training data was created using the hadronization cluster model [8]. The cluster model is based on t'Hooft's planar diagram theory [95]; the dominant color structure of quantum chromodynamics (QCD) diagrams in the perturbation expansion in $1/N_c$ can be represented in a planar form using color lines, which is commonly known as the limit $N_c \to \infty$. The resulting color topology in Monte Carlo events with partons in the final-state color features open color lines after the parton showers. Following a non-perturbative isotropic decay of any left gluons in the parton jets to quark-antiquark pairs, the event finally consists of color-connected partons in color triplet or antitriplet states. These parton pairs form color-singlet clusters. This is the so-called color preconfinement [6]; the tendency of the partons generated in the parton shower to be arranged in color singlet clusters (prehadrons) with limited extension in both coordinate and momentum space. The principle of color preconfinement states that the mass distribution of these clusters is independent of the hard-scattering process and its center-of-mass energy. The cluster mass spectrum is not only universal but also peaked at low masses; therefore, most of the clusters decay into two hadrons and some just into one hadron. However, there is a small fraction of clusters that are too heavy for this to be a reasonable approach. Therefore, these heavy clusters are first split into lighter clusters before they decay. Such decays of massive clusters are beyond the scope of this publication, and we will consider it in future work. Since the kinematics of a cluster decaying into a single hadron is trivial, our training data set only includes cases of decay into two hadrons. To further simplify the training data, we consider only decays into pairs of π^0. Each decay in our data set was described with the following information; the four-momentum of the cluster, the four-momenta of the two hadrons together with their flavor (encoded as a Particle Data Group (PDG) [96] code), and the Pert flag. Pert $= 1$ means that hadrons that contain a parton produced in the perturbative stage of the event remember the direction of the parton in the rest frame of the cluster. To create the training data, we used e^+e^- collisions at $\sqrt{s} = 91.2$ GeV generated by Herwig version 7.2.1. The only modification to the default generator settings was the change that the hadrons produced from cluster decays were on the mass shell.[1]

B. GAN model and training

There has been a growing interest in training generative networks to do particle physics simulations. However, significant research work goes into finding algorithms that work for such data. For example, prior work for simulating

[1]This setting can be achieved by adding the command: setClusterDecayer : OnShellYes in the input file.

calorimeters and particle four-momenta have found that new innovations are needed to force the generative networks to reproduce certain important physics distributions well, like energy and the mass of a particle [34,97,98]. Recent work has also studied the tradeoffs between accurate models and models that are fast to evaluate [80] and the sensitivity analysis of GAN hyperparameters to guide users on how to optimize a GAN [99]. Scaling to larger input sizes is another active area of research [71,98]. The model described below is optimized for the hadronization task, and the output structure, preprocessing, and postprocessing were optimized to make the learning easier and more stable and prevent the network from generating certain unphysical results. The architecture was chosen so as to allow fast evaluation after integrating it into Herwig.

We trained a conditional GAN to simulate the cluster decays. In a GAN, there is a Generator neural network (Generator for short) and a Discriminant neural network (Discriminator for short). Inputs to the Generator are the cluster's four vectors (E, p_x, p_y, p_z), and N features sampled from a Gaussian distribution. The N numbers are called *noise*. N is a hyperparameter and tuned to be 10. Outputs[2] of the Generator are the polar angle, θ, and azimuthal angle, ϕ, of the leading hadron's momentum in the spherical coordinate system in the cluster frame, in which the two hadrons are created back to back. With the two angular variables, θ and ϕ, and the cluster's four vector, we reconstruct the four vectors of the two outgoing hadrons as a postprocessing step. Inputs to the Discriminator are just the two angular variables coming from either the Generator, labeled as background, or those from the Herwig, labeled as signal. The output of the Discriminator is a score that is higher for events from the Herwig and lower for events from the Generator. The Discriminator is trained to separate signal from background. However, the Generator is trained to yield a signal-like Discriminator score.

The GAN is based on multilayer perceptrons (MLPs). Both the Generator and the Discriminator are composed of a two-layer perceptron. Each perceptron consists of a sequence of KERAS [100] modules; a fully connected (dense) network of a hidden size of 256, a batch normalization layer, and a LEAKYRELU activation function [101]. These parameters were not extensively optimized.

To help train a GAN, we preprocess the training data. The incoming cluster's four vector is scaled so that their values are between -1 and 1; so are the two angular variables (ϕ and θ). In this way, all inputs and outputs are within the same scale. Finally, we use the tanh activation function as the last layer of the Generator. The Discriminator and the Generator are trained separately and alternately by two independent ADAM optimizers [102], both with a learning rate of 10^{-4}, for about 1000 epochs.

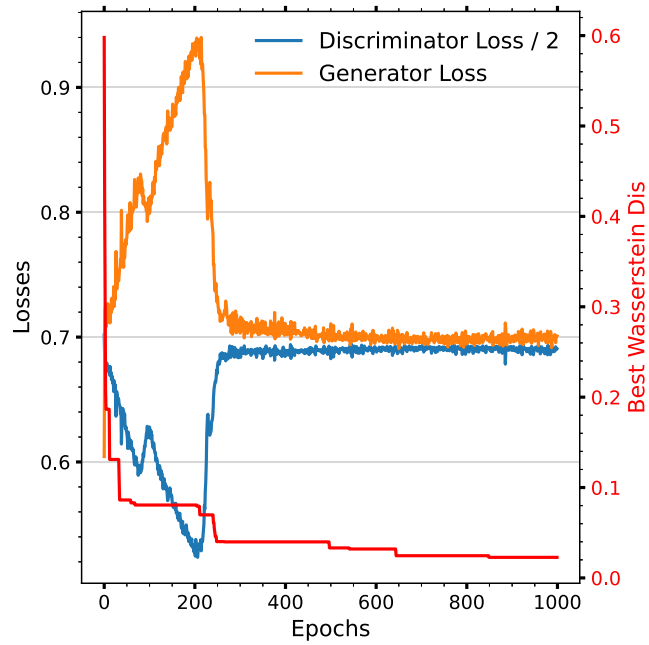

FIG. 1. Generator loss and discriminator loss and progressive best Wasserstein distance as a function of the training epochs for training a GAN with events where two partons are with Pert = 0. Both Generator and Discriminator loss are the binary-crossentropy loss, and the Discriminator loss is divided by two for visualization purposes. The progressive Wasserstein distance is gauged by the right side of the y axis.

Figure 1 shows the evolution of the Discriminator loss, which is divided by two for visualization purposes, the Generator loss, and the progressive best total Wasserstein distances[3] [103,104] for training a GAN with events where two partons are with Pert = 0. The total Wasserstein distance summing over the distances of all variables, is calculated after training for one epoch and only the smallest value is plotted. At the beginning of the training (epoch < 70), even though the Generator loss is going up, we see a rapid drop in the Wasserstein distance until the Generator loss is beyond 0.8. For more than 100 epochs, the Discriminator keeps outperforming the Generator as seen by the increasing Generator loss and the decreasing Discriminator loss. This situation is changed around epoch 200 and finally, the two networks reach an equilibrium around epoch 250. Beyond epoch 600, we only see about 0.002 improvements in the Wasserstein distance. The best model for events with partons of Pert = 0, is found at the epoch 849 with a total Wasserstein distance of 0.0228.

[2]We also tried other outputs such as the four vector of the two hadrons and found the Generator could not preserve the momentum conservation.

[3]This is a common metric in machine learning that quantifies the minimal "work" required to transform one density into another, where work, in this case, is defined as the integral of the density multiplied by the distance moved.

A similar analysis was performed when training events with at least one parton with Pert = 1.

C. Integration into Herwig

Each part of Herwig is implemented as a C++ class that contains the implementation of the Herwig physics models, inheriting from an abstract base class in ThePEG [105]. The ClusterHadronizationHandler is the class that controls the cluster hadronization model. Our ultimate goal will be to replace the entire ClusterHadronizationHandler with its ML counterpart. However, since in these studies, we concentrate on the decay of clusters into two hadrons, it was sufficient to modify ClusterDecayer—a helper class of the ClusterHadronizationHandler that controls this process. The generative model trained in Python using TensorFlow is converted into the ONNX format [90] and integrated into the Herwig chain using the C++ API of ONNX Runtime [106]. The advent of the ONNX format makes it possible to train a model in one software and hardware environment and then apply it in a completely different environment. ONNX Runtime is well suited for running fast neural network inference as part of a large C++ workflow, and by using it, we avoid having to integrate and maintain TENSORFLOW [107] within the Herwig framework.

All preprocessing and postprocessing steps performed for training are repeated within Herwig for inference. The entire simulation chain, including the GAN, is then run in Herwig in order to produce the final comparisons and results.

III. RESULTS

Section III A provides low-level results of individual cluster decays while Sec. III B includes full event simulations and comparisons to e^+e^- data.

A. Low-level validation

Since the training data contained only clusters produced in e^+e^- collisions at $\sqrt{s} = 91.2$ GeV that decayed into π^0 pairs, we begin by comparing the π^0 kinematic variables generated by HADML and Herwig precisely in such decays. The data generated by Herwig, with which we compared the results of HADML in this section, were not used for training. In Fig. 2 we show the distribution of the pseudorapidity (left panels) and transverse momentum distribution (right panels) of π^0 from the decays of the Pert = 0 (upper panels) and Pert = 1 (lower panels) clusters. As expected, we see that the transverse momentum spectra of pions coming from clusters containing

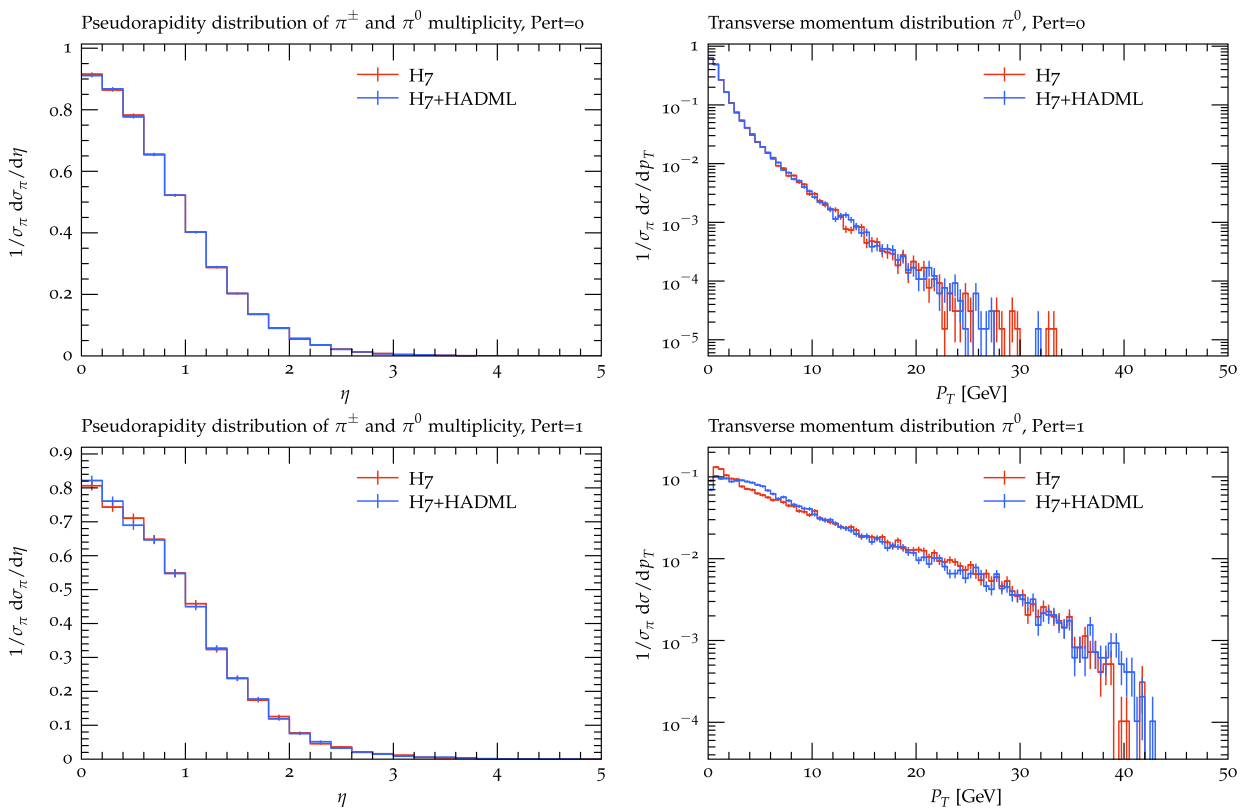

FIG. 2. Pseudorapidity (left panels) transverse momentum (right panels) distribution of π^0 from decays of Pert = 0 (upper panels) and Pert = 1 (lower panels) clusters produced in e^+e^- collisions at $\sqrt{s} = 91.2$ GeV.

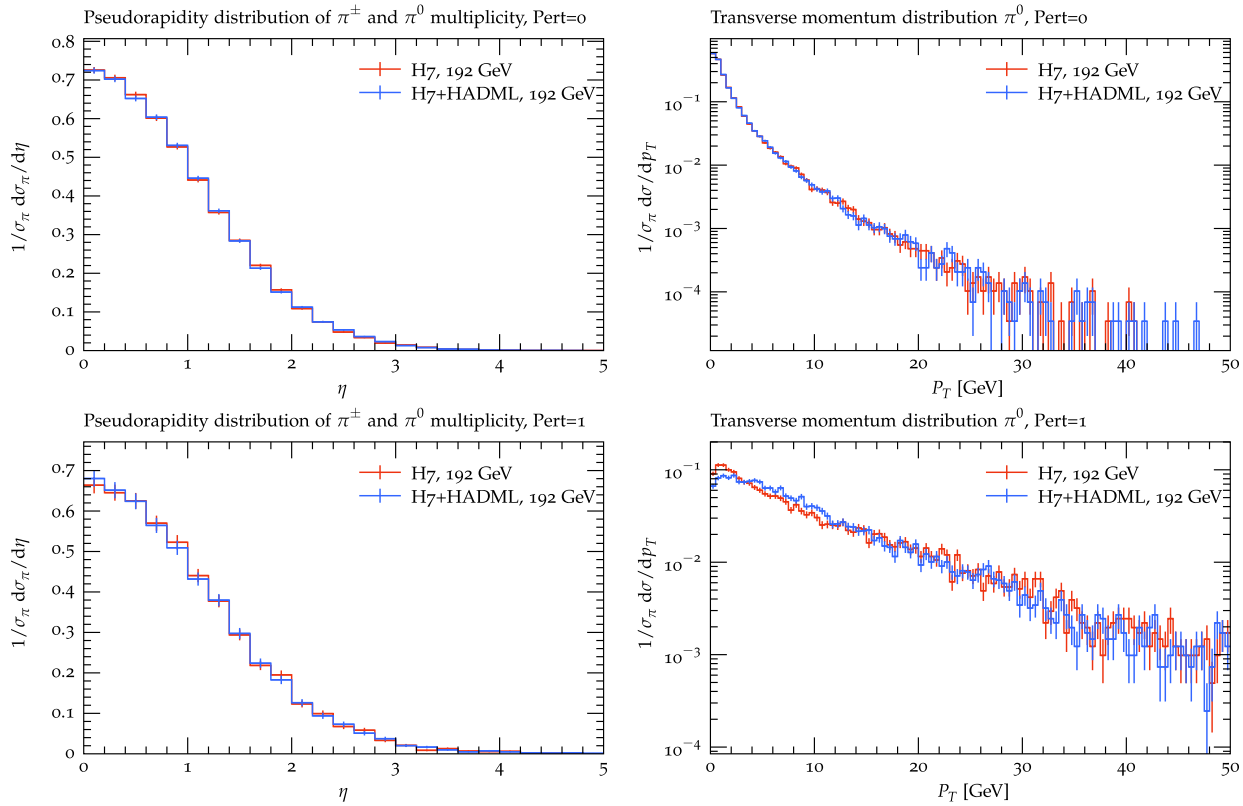

FIG. 3. Pseudorapidity (left panels) transverse momentum (right panels) distribution of π^0 from decays of Pert $= 0$ (upper panels) and Pert $= 1$ (lower panels) clusters produced in e^+e^- collisions at $\sqrt{s} = 192$ GeV.

"perturbative" quarks (Pert $= 1$) are harder compared to those containing only nonperturbative partons (Pert $= 0$). However, the most important observation from Fig. 2 is that Herwig7 + HADML (labeled on figures as H7 + HADML) matches the pseudorapidity of the pions generated by Herwig7 with the cluster model (labeled as H7 on figures). Transverse momentum spectra that extend over several orders of magnitude are also well approximated by H7 + HADML. Taking a closer look at these distributions, we see minor differences for low transverse momenta in the case of clusters that have a memory of perturbative quarks (bottom-left panel in Fig. 2). Such small differences are, of course, acceptable, especially since the information about the four-momentum of partons that make up the clusters were not used for training. Taking this additional information into account in the training process will likely eliminate these minor differences. However, this is beyond the scope of this publication, and we will leave this problem for future work.

It is crucial that the hadronization model is universal, i.e., that it works independently of the hard process or collision energy. As we described in the Sec. II A the cluster model has this property. To test whether HADML also is universal, we decided to repeat the comparison made at the beginning of this section, but this time generating events with collision energies twice as high as those used in the training data. In Fig. 3 we show π^0 kinematic variables generated by H7 + HADML and Herwig7 in e^+e^- collisions at $\sqrt{s} = 192$ GeV. We can see that all distributions are described very similarly by both models, which reassured us that the HADML model is also universal.

The last thing we need to check before using HADML to simulate the decay of all clusters into hadron pairs in Herwig is whether the model is able to describe the kinematics of other hadrons than π^0. In Fig. 4 we present the pseudorapidity (left panels) and transverse momentum (right panels) distribution of $\pi^{\pm}$ and π^0 (first row), kaons (second row), and lambdas (third row). We see that the distributions differ for the various hadrons, but they are all described almost identically by both models. This encouraged us to perform a comparison with experimental data in which the kinematics of all hadrons[4] in Herwig are generated by HADML model.

B. Full-event validation

In this section we generate full events using HADML integrated into Herwig and compare the results also to data

[4]Except for a small number of hadrons that come from the decay of a cluster into a single hadron for which the kinematics is trivial.

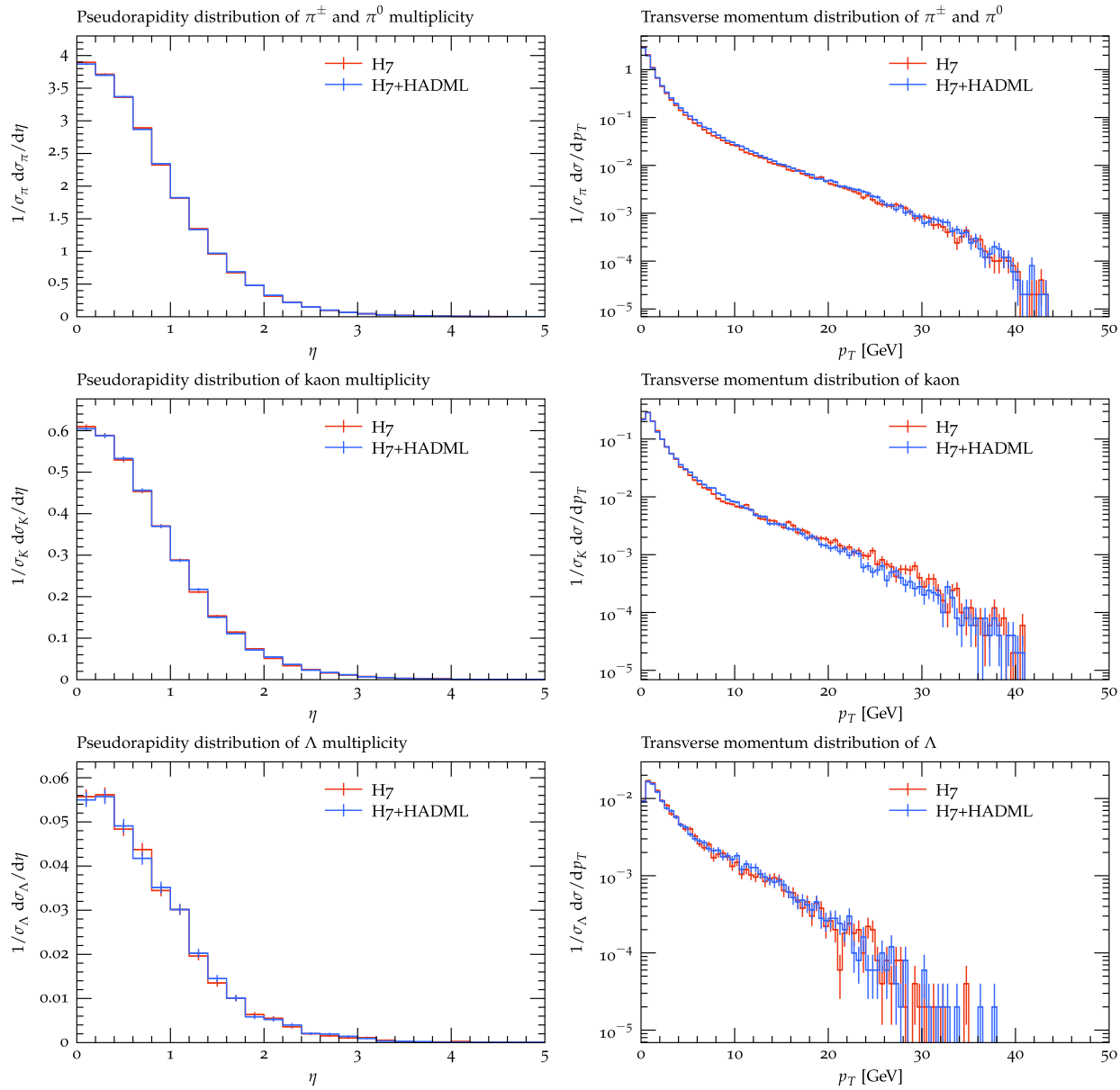

FIG. 4. Pseudorapidity (left panels) and transverse momentum (right panels) distribution of $\pi^{\pm}$ and π^0 (first row), kaons (second row), and lambdas (third row).

from LEP.[5] In particular, we consider an analysis from DELPHI with data collected at $\sqrt{s} = 91.2$ GeV [108] using RIVET[6] [109]. These events correspond to hadronic Z boson decays with a number of event shape and identified hadron spectra. These data have been used for hadronization parameter tuning [108,110].

Figure 5 shows histograms of various event shapes. Thrust [111,112] is the quintessential e^+e^- event shape,

$$T = \max_{\vec{n}} \left(\frac{\sum |\vec{p}_i \cdot \vec{n}|}{\sum |\vec{p}_i|} \right), \tag{3.1}$$

where the sum runs over all final-state particle three-momenta. The direction $\vec{n}$ that maximizes the argument of Eq. (3.1) is called the Thrust axis. Thrust major is defined similarly to Eq. (3.1) but with $\vec{n}$ replaced with vectors transverse to the Thrust axis and Thrust minor is the same, but with an optimization only over directions perpendicular to both the Thurst and Thurst major axes. The Sphericity is computed from the eigenvalues of the quadratic momentum tensor

[5]Note that the data are for illustration only; given that the GAN is trained on Herwig, we cannot expect it to outperform Herwig. Tuning to data is a longer-term goal of this research (see Sec. IV).

[6]https://rivet.hepforge.org/analyses/DELPHI_1996_S3430090.

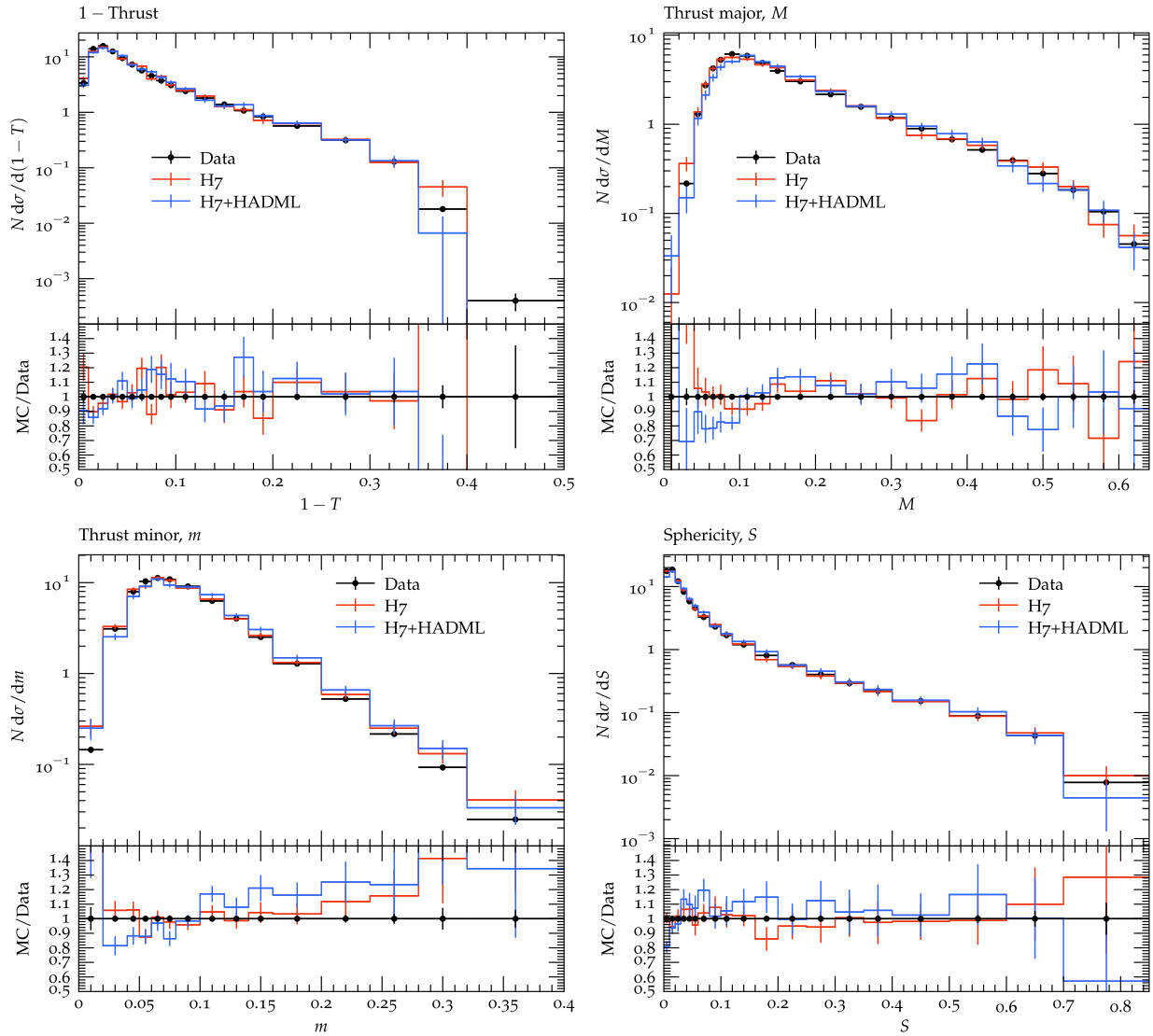

FIG. 5. Normalized, differential cross sections of Thrust (top left), Thrust major (top right), Thrust minor (lower left), and Sphericity (lower right) for Herwig, Herwig with HADML, and for data from DELPHI at LEP. Error bars on the predictions represent statistical uncertainties.

$$M^{\alpha\beta} = \sum p_i^\alpha p_i^\beta, \tag{3.2}$$

where α, β are the spatial momentum indices, and the sum runs over the same particles as in Eq. (3.1). Sphericity is defined as $\frac{3}{2}(\lambda_2 + \lambda_3)$ for eigenvalues λ_i of the 3×3 matrix defined in Eq. (3.2) and $\lambda_3 \leq \lambda_2 \leq \lambda_1$. Hadronization shifts event shapes (see e.g., Ref. [113]) and so these observables are sensitive to hadronization modeling. Figure 5 shows that HADML agrees with Herwig within 10% across most of the spectra, which itself agrees with data at a similar level. Individual particle spectra are shown in Fig. 6 for the transverse momenta along the Thurst major and minor directions. The level of agreement is similar to the event shapes where there is sufficient statistical power.

IV. SUMMARY AND OUTLOOK

In this paper, we have established a first step on the path towards a neural network-based hadronization model. The cluster hadronization model from Herwig has been emulated with HADML, a computer code implementing a generative adversarial network. This model is designed to reproduce the two-body decay of clusters into pions. The HADML is integrated into the full Herwig program by using all other hadronization components from the Herwig default model. The kinematic properties of other hadrons are emulated using the pion model and conservation of energy. We have shown that the HADML is able to reproduce Herwig's light cluster decays and when integrated with the full Herwig simulation, is able to reproduce results from e^+e^- data as well.

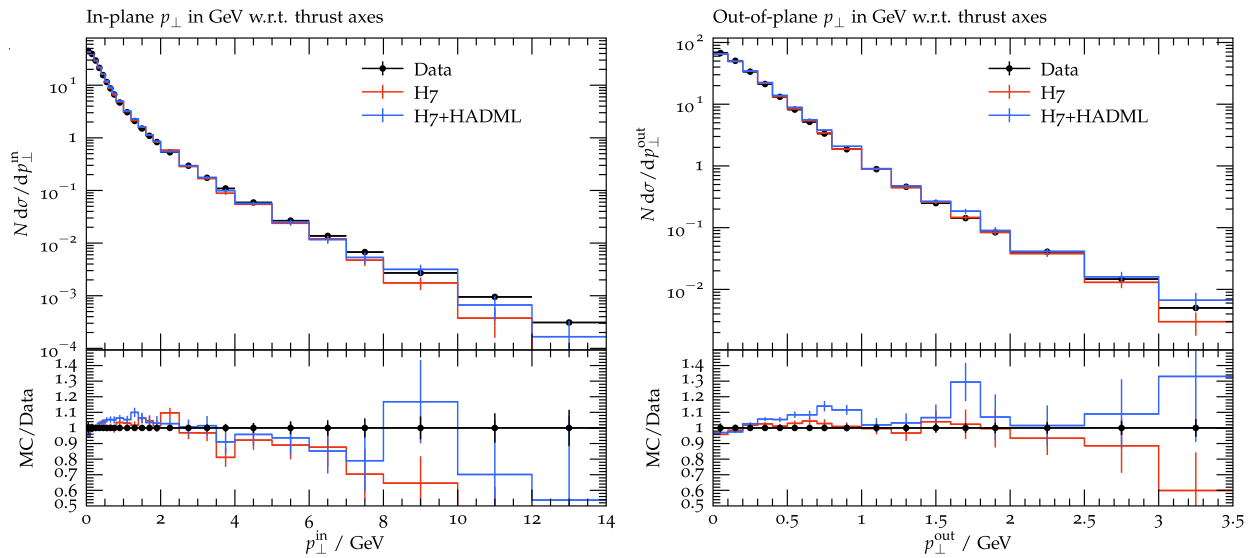

FIG. 6. Normalized, differential cross sections of particle transverse momenta along the Thrust major (left) and Thurst minor (right) axes for Herwig, Herwig with our HADML, and for data from DELPHI at LEP. Error bars on the predictions represent statistical uncertainties.

The ultimate goal of this research direction is to train the ML model directly on data to improve upon the existing hadronization models. A number of technical and methodological steps are required to achieve this vision. First, the deep generative model needs to be extended to directly accommodate multiple hadron species and to model the relative probabilities of the various final states. In this work, we have modeled different hadron species using conservation of energy, but this means that the fragmentation is assumed universal. Architectural modifications could allow for perturbations on universality. Hyperparameter optimization, including the investigation of alternative generative models, is an important component of future work. Once the deep generative model has the capacity to reproduce all of the physics of the Herwig cluster model, methodological innovation is required to explore how to tune the model to data. Traditionally, e^+e^- data are used for tuning. Optimization with a large set of one-dimensional, binned measurements will need to be explored. A nontrivial aspect of this optimization is that while the hadronization model would be differentiable, the parton shower input would not be. Building in a model of uncertainty would also be a central aspect of model tuning. It may also be possible to tune with unbinned, and higher-dimensional results from ep and pp data [94,114–118].

While we have focused on hadronization in the context of collider physics, the ideas and concepts described in this paper have broader implications. First of all, hadronization is used across high-energy particles and nuclear physics (see e.g., Ref. [119]) and perturbations on the collider model may be required to accurately describe other systems. Second, there are other physical systems where first-principles input is combined with phenomenological models. For example, a complete description of observational cosmology requires an N-body simulation of the dark matter to be combined with a description of visible matter around dark matter halos (see e.g., Ref. [120–124]). While different applications call for domain-specific adaptations, some components and core methodology is common. Further development in this research area will enable important advances in simulation to improve inference in high-energy physics and beyond.

ACKNOWLEDGMENTS

The work of A. S. was funded by Grant No. 2019/34/E/ST2/00457 of the National Science Centre, Poland and the Priority Research Area Digiworld under the program Excellence Initiative—Research University at the Jagiellonian University in Krakow. B. N. and X. J. were supported by the Department of Energy, Office of Science under Contract No. DE-AC02-05CH11231. A. G. is supported by the U.S. Department of Energy (DOE), Office of Science under Grant No. DE-SC0009920.

Note added.—As this paper was being finalized, we became aware of the recent work in Ref. [125], which has a similar goal. That study uses a different Monte Carlo program (Pythia instead of Herwig) and uses a different generative model (variational autoencoder instead of a GAN). Reference [125] also focuses on the pion-only case.

[1] A. Buckley *et al.*, General-purpose event generators for LHC physics, Phys. Rep. **504**, 145 (2011).

[2] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjostrand, Parton fragmentation and string dynamics, Phys. Rep. **97**, 31 (1983).

[3] T. Sjostrand, Jet fragmentation of nearby partons, Nucl. Phys. **B248**, 469 (1984).

[4] T. Sjöstrand, S. Mrenna, and P. Z. Skands, A brief introduction to PYTHIA 8.1, Comput. Phys. Commun. **178**, 852 (2008).

[5] T. Sjöstrand, S. Mrenna, and P. Z. Skands, PYTHIA 6.4 physics and manual, J. High Energy Phys. 05 (2006) 026.

[6] D. Amati and G. Veneziano, Preconfinement as a property of perturbative QCD, Phys. Lett. **83B**, 87 (1979).

[7] A. Bassetto, M. Ciafaloni, and G. Marchesini, Color singlet distributions and mass damping in perturbative QCD, Phys. Lett. **83B**, 207 (1979).

[8] B. R. Webber, A QCD model for jet fragmentation including soft gluon interference, Nucl. Phys. **B238**, 492 (1984).

[9] G. Corcella, I. G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, , M. H. Seymour, and B. R. Webber, HERWIG 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes), J. High Energy Phys. 01 (2001) 010.

[10] M. Bahr *et al.*, Herwig++ physics and manual, Eur. Phys. J. C **58**, 639 (2008).

[11] J. Bellm *et al.*, Herwig 7.0/Herwig++ 3.0 release note, Eur. Phys. J. C **76**, 196 (2016).

[12] J. Bellm *et al.*, Herwig 7.2 release note, Eur. Phys. J. C **80**, 452 (2020).

[13] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter, Event generation with SHERPA 1.1, J. High Energy Phys. 02 (2009) 007.

[14] E. Bothmann *et al.* (Sherpa Collaboration), Event generation with sherpa 2.2, SciPost Phys. **7**, 034 (2019).

[15] ATLAS Collaboration, Measurement of the Lund Jet Plane Using Charged Particles in 13 TeV Proton-Proton Collisions with the ATLAS Detector, Phys. Rev. Lett. **124**, 222002 (2020).

[16] V. Khachatryan *et al.* (CMS Collaboration), Observation of long-range near-side angular correlations in proton-proton collisions at the LHC, J. High Energy Phys. 09 (2010) 091.

[17] B. B. Abelev *et al.* (ALICE Collaboration), Production of $\Sigma(1385)^{\pm}$ and $\Xi(1530)^{0}$ in proton-proton collisions at $\sqrt{s} = 7$ TeV, Eur. Phys. J. C **75**, 1 (2015).

[18] V. Khachatryan *et al.* (CMS Collaboration), Evidence for collectivity in pp collisions at the LHC, Phys. Lett. B **765**, 193 (2017).

[19] J. Adam *et al.* (ALICE Collaboration), Enhanced production of multi-strange hadrons in high-multiplicity proton-proton collisions, Nat. Phys. **13**, 535 (2017).

[20] A. Buckley, H. Hoeth, H. Lacker, H. Schulz, and J. E. von Seggern, Systematic event generator tuning for the LHC, Eur. Phys. J. C **65**, 331 (2010).

[21] P. Ilten, M. Williams, and Y. Yang, Event generator tuning using Bayesian optimization, J. Instrum. **12**, P04028 (2017).

[22] A. Andreassen and B. Nachman, Neural networks for full phase-space reweighting and parameter tuning, Phys. Rev. D **101**, 091901 (2020).

[23] W. Wang, M. Krishnamoorthy, J. Muller, S. Mrenna, H. Schulz, X. Ju, S. Leyffer, and Z. Marshall, BROOD: Bilevel and robust optimization and outlier detection for efficient tuning of high-energy physics event generators, SciPost Phys. Core **5**, 1 (2021).

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair *et al.*, Generative adversarial nets, in *Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, NIPS'14* (MIT Press, Cambridge, MA, 2014), pp. 2672–2680, http://dl.acm.org/citation.cfm?id=2969033.2969125.

[25] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. **35**, 53 (2018).

[26] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv:1312.6114.

[27] D. P. Kingma and M. Welling, An introduction to variational autoencoders, Found. Trends Mach. Learn. **12**, 307 (2019).

[28] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, Int. Conf. Mach. Learn. **37**, 1530 (2015).

[29] I. Kobyzev, S. Prince, and M. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell., 43 1 (2020).

[30] 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**, 4 (2017).

[31] 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**, 014021 (2018).

[32] M. Paganini, L. de Oliveira, and B. Nachman, Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters, Phys. Rev. Lett. **120**, 042003 (2018).

[33] J. Bendavid, Efficient Monte Carlo integration using boosted decision trees and generative deep neural networks, arXiv:1707.00028.

[34] A. Butter, T. Plehn, and R. Winterhalder, How to GAN LHC events, SciPost Phys. **7**, 075 (2019).

[35] Y. Alanazi *et al.*, Machine learning-based event generator for electron-proton scattering, Phys. Rev. D **106**, 096002 (2022).

[36] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale, and S. Schumann, Exploring phase space with neural importance sampling, SciPost Phys. **8**, 069 (2020).

[37] C. Gao, S. Höche, J. Isaacson, C. Krause, and H. Schulz, Event generation with normalizing flows, Phys. Rev. D **101**, 076002 (2020).

[38] C. Gao, J. Isaacson, and C. Krause, i-flow: High-dimensional integration and sampling with normalizing flows, Mach. Learn. Sci. Tech. **1**, 045023 (2020).

[39] A. Butter, T. Heimel, S. Hummerich, T. Krebs, T. Plehn, A. Rousselot *et al.*, Generative networks for precision enthusiasts, arXiv:2110.13632.

[40] J. W. Monk, Deep learning as a parton shower, J. High Energy Phys. 12 (2018) 021.

[41] S. Carrazza and F. A. Dreyer, Lund jet images from generative and cycle-consistent adversarial networks, Eur. Phys. J. C **79**, 979 (2019).
[42] R. Kansal, J. Duarte, B. Orzari, T. Tomei, M. Pierini, M. Touranakou *et al.*, Graph generative adversarial networks for sparse data generation in high energy physics, in *Proceedings of the 34th Conference on Neural Information Processing Systems* (2020), arXiv:2012.00173.
[43] Y. S. Lai, D. Neill, M. Płoskoń, and F. Ringer, Explainable machine learning of the underlying physics of high-energy particle collisions, Phys. Lett. B **829**, 137055 (2022).
[44] R. Kansal, J. Duarte, H. Su, B. Orzari, T. Tomei, M. Pierini *et al.*, Particle cloud generation with message passing generative adversarial networks, arXiv:2106.11535.
[45] B. Orzari, T. Tomei, M. Pierini, M. Touranakou, J. Duarte, R. Kansal *et al.*, Sparse data generation for particle-based simulation of hadronic jets in the LHC, in *Proceedings of the 38th International Conference on Machine Learning Conference* (2021), arXiv:2109.15197.
[46] S. Tsan, R. Kansal, A. Aportela, D. Diaz, J. Duarte, S. Krishna *et al.*, Particle graph autoencoders and differentiable, learned energy Mover's distance, in *Proceedings of the 35th Conference on Neural Information Processing Systems* (2021), arXiv:2111.12849.
[47] M. Touranakou, N. Chernyavskaya, J. Duarte, D. Gunopulos, R. Kansal, B. Orzari, M. Pierini, T. Tomei, and J.-R. Vlimant, Particle-based fast jet simulation at the LHC with variational autoencoders, Mach. Learn.: Sci. Technol. **3**, 035003 (2022).
[48] S. Vallecorsa, F. Carminati, and G. Khattak, 3D convolutional GAN for fast simulation, EPJ Web Conf. **214**, 02010 (2019).
[49] C. Ahdida *et al.* (SHiP Collaboration), Fast simulation of muons produced at the SHiP experiment using generative adversarial networks, J. Instrum. **14**, P11028 (2019).
[50] V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin, and E. Zakharov, Generative models for fast calorimeter simulation. LHCb case, EPJ Web Conf. **214**, 02034 (2019).
[51] ATLAS Collaboration, Deep generative models for fast shower simulation in ATLAS, Report No. ATL-SOFT-PUB-2018-001, 2018.
[52] F. Carminati, A. Gheata, G. Khattak, P. Mendez Lorenzo, S. Sharan, and S. Vallecorsa, Three dimensional generative adversarial networks for fast simulation, J. Phys. Conf. Ser. **1085**, 032016 (2018).
[53] S. Vallecorsa, Generative models for fast simulation, J. Phys. Conf. Ser. **1085**, 022005 (2018).
[54] P. Musella and F. Pandolfi, Fast and accurate simulation of particle detectors using generative adversarial networks, Comput. Softw. Big Sci. **2**, 8 (2018).
[55] M. Erdmann, L. Geiger, J. Glombitza, and D. Schmidt, Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks, Comput. Softw. Big Sci. **2**, 4 (2018).
[56] K. Deja, T. Trzcinski, and u. Graczykowski, Generative models for fast cluster simulations in the TPC for the ALICE experiment, EPJ Web Conf. **214**, 06003 (2019).
[57] D. Derkach, N. Kazeev, F. Ratnikov, A. Ustyuzhanin, and A. Volokhova, RICH 2018, Nucl. Instrum. Methods Phys. Res., Sect. A **952**, 161804 (2020).
[58] M. Erdmann, J. Glombitza, and T. Quast, Precise simulation of electromagnetic calorimeter showers using a Wasserstein generative adversarial network, Comput. Softw. Big Sci. **3**, 4 (2019).
[59] L. de Oliveira, M. Paganini, and B. Nachman, Tips and tricks for training GANs with physics constraints, 2017, https://dl4physicalsciences.github.io/files/nips_dlps_2017_26.pdf.
[60] L. de Oliveira, M. Paganini, and B. Nachman, Controlling physical attributes in GAN-accelerated simulation of electromagnetic calorimeters, J. Phys. Conf. Ser. **1085**, 042017 (2018).
[61] B. Hooberman, A. Farbin, G. Khattak, V. Pacela, M. Pierini, J.-R. Vlimant *et al.*, Calorimetry with deep learning: Particle classification, energy regression, and simulation for high-energy physics, 2017, https://dl4physicalsciences.github.io/files/nips_dlps_2017_15.pdf.
[62] D. Belayneh *et al.*, Calorimetry with deep learning: Particle simulation and reconstruction for collider physics, Eur. Phys. J. C **80**, 688 (2020).
[63] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol *et al.*, Getting high: High fidelity simulation of high granularity calorimeters with high speed, arXiv:2005.05334.
[64] S. Diefenbacher, E. Eren, G. Kasieczka, A. Korol, B. Nachman, and D. Shih, DCTRGAN: Improving the precision of generative models with reweighting, J. Instrum. **15**, P11004 (2020).
[65] A. Maevskiy, F. Ratnikov, A. Zinchenko, and V. Riabov, Simulating the time projection chamber responses at the MPD detector using generative adversarial networks, Eur. Phys. J. C **81**, 599 (2021).
[66] K. Deja, J. Dubiński, P. Nowak, S. Wenzel, and T. Trzciński, End-to-end sinkhorn autoencoder with noise generator, IEEE Access **9**, 7211 (2021).
[67] K. Dohi, Variational autoencoders for jet simulation, arXiv:2009.04842.
[68] F. Rehm, S. Vallecorsa, V. Saletore, H. Pabst, A. Chaibi, V. Codreanu *et al.*, Reduced precision strategies for deep learning: A high energy physics generative adversarial network use case, arXiv:2103.10142.
[69] F. Rehm, S. Vallecorsa, K. Borras, and D. Krücker, Validation of deep convolutional generative adversarial networks for high energy physics calorimeter simulations, arXiv:2103.13698.
[70] F. Rehm, S. Vallecorsa, K. Borras, and D. Krücker, Physics validation of novel convolutional 2D architectures for speeding up high energy physics simulations, EPJ Web Conf. **251**, 03042 (2021).
[71] G. R. Khattak, S. Vallecorsa, F. Carminati, and G. M. Khan, Fast simulation of a high granularity calorimeter by generative adversarial networks, Eur. Phys. J. C **82**, 386 (2022).
[72] L. Anderlini, Machine learning for the LHCb simulation, arXiv:2110.07925.
[73] C. Fanelli and J. Pomponi, DeepRICH: Learning deeply cherenkov detectors, Mach. Learn. Sci. Tech. **1**, 015010 (2019).

[74] Y. Lu, J. Collado, D. Whiteson, and P. Baldi, SARM: Sparse autoregressive model for scalable generation of sparse images in particle physics, Phys. Rev. D **103**, 036012 (2021).

[75] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, D. Hundhausen, G. Kasieczka *et al.*, Hadrons, better, faster, stronger, Mach. Learn. Sci. Tech. **3**, 025014 (2022).

[76] C. Krause and D. Shih, CaloFlow: Fast and accurate generation of calorimeter showers with normalizing flows, arXiv:2106.05285.

[77] A. Hariri, D. Dyachkova, and S. Gleyzer, Graph generative models for fast detector simulations in high energy physics, arXiv:2104.01725.

[78] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol *et al.*, Decoding photons: Physics in the latent space of a BIB-AE generative network, EPJ Web Conf. **251**, 03003 (2021).

[79] G. Aad *et al.* (ATLAS Collaboration), AtlFast3: The next generation of fast simulation in ATLAS, Comput. Softw. Big Sci. **6**, 7 (2022).

[80] C. Krause and D. Shih, CaloFlow II: Even faster and still accurate generation of calorimeter showers with normalizing flows, arXiv:2110.11377.

[81] S. Bieringer, A. Butter, S. Diefenbacher, E. Eren, F. Gaede, D. Hundhausen, G. Kasieczka, B. Nachman, T. Plehn, and M. Trabs, Calomplification—The power of generative calorimeter models, J. Instrum. **17**, P09028 (2022).

[82] A. Butter *et al.*, Machine learning and LHC event generation, arXiv:2203.07460.

[83] A. Butter and T. Plehn, Generative networks for LHC events, arXiv:2008.08558.

[84] M. Feickert and B. Nachman, A living review of machine learning for particle physics, arXiv:2102.02770.

[85] S. Forte, L. Garrido, J. I. Latorre, and A. Piccione, Neural network parametrization of deep inelastic structure functions, J. High Energy Phys. 05 (2002) 062.

[86] R. D. Ball *et al.*, The path to proton structure at one-percent accuracy, Eur. Phys. J. C **82**, 428 (2022).

[87] V. N. Gribov and L. N. Lipatov, Deep inelastic e p scattering in perturbation theory, Sov. J. Nucl. Phys. **15**, 438 (1972).

[88] Y. L. Dokshitzer, Calculation of the structure functions for deep inelastic scattering and e+ e- annihilation by perturbation theory in quantum chromodynamics, Sov. Phys. JETP **46**, 641 (1977).

[89] G. Altarelli and G. Parisi, Asymptotic freedom in parton language, Nucl. Phys. **B126**, 298 (1977).

[90] J. Bai, F. Lu, K. Zhang *et al.*, Onnx: Open neural network exchange, https://onnx.ai/.

[91] S. Gieseke, C. Rohr, and A. Siodmok, Colour reconnections in Herwig++, Eur. Phys. J. C **72**, 2225 (2012).

[92] S. Gieseke, P. Kirchgaeßer, S. Plätzer, and A. Siódmok, Soft gluon evolution as guiding principle for colour reconnection, Acta Phys. Pol. B **50**, 1871 (2019).

[93] J. Bellm, C. B. Duncan, S. Gieseke, M. Myska, and A. Siódmok, Spacetime colour reconnection in Herwig 7, Eur. Phys. J. C **79**, 1003 (2019).

[94] M. Arratia *et al.*, Presenting unbinned differential cross section results, J. Instrum. **17**, P01024 (2022).

[95] G. 't Hooft, A planar diagram theory for strong interactions, Nucl. Phys. **B72**, 461 (1974).

[96] M. Tanabashi *et al.* (Particle Data Group), Review of particle physics, Phys. Rev. D **98**, 030001 (2018).

[97] A. Ghosh, Deep generative models for fast shower simulation in ATLAS, J. Phys. Conf. Ser. **1525**, 012077 (2020).

[98] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, and K. Krüger, Getting high: High fidelity simulation of high granularity calorimeters with high speed, Comput. Softw. Big Sci. **5**, 13 (2021).

[99] V. Dumont, X. Ju, and J. Mueller, Hyperparameter optimization of generative adversarial network models for high-energy physics simulations, arXiv:2208.07715.

[100] F. Chollet, Keras, https://github.com/fchollet/keras (2017).

[101] A. Y. Hannun, A. L. Maas, and A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (International Conference on Machine Learning, Atlanta, Georgia, 2013).

[102] D. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[103] L. N. Vaserstein, Markov processes over denumerable products of spaces, describing large systems of automata, Prob. Peredachi Inf. **5**, 64 (1969).

[104] L. V. Kantorovich, Mathematical methods of organizing and planning production, Manage. Sci. **6**, 366 (1939).

[105] L. Lonnblad, ThePEG, Pythia7, herwig++ and Ariadne, Nucl. Instrum. Methods Phys. Res., Sect. A **559**, 246 (2006).

[106] O. R. developers, Onnx runtime, https://onnxruntime.ai/ (2021).

[107] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean *et al.*, Tensorflow: A system for large-scale machine learning, in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)* (USENIX Symposium on Operating Systems Design and Implementation, Savannah, Georgia, 2016), Vol. 16, pp. 265–283.

[108] P. Abreu *et al.* (DELPHI Collaboration), Tuning and test of fragmentation models based on identified particles and precision event shape data, Z. Phys. C **73**, 11 (1996).

[109] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lonnblad, J. Monk *et al.*, Rivet user manual, Comput. Phys. Commun. **184**, 2803 (2013).

[110] D. Reichelt, P. Richardson, and A. Siodmok, Improving the simulation of quark and gluon jets with Herwig 7, Eur. Phys. J. C **77**, 876 (2017).

[111] E. Farhi, A QCD Test for Jets, Phys. Rev. Lett. **39**, 1587 (1977).

[112] S. Brandt, C. Peyrou, R. Sosnowski, and A. Wroblewski, The principal axis of jets. An Attempt to analyze high-energy collisions as two-body processes, Phys. Lett. **12**, 57 (1964).

[113] R. Abbate, M. Fickinger, A. H. Hoang, V. Mateu, and I. W. Stewart, Thrust at N^3LL with power corrections and a precision global fit for $\alpha_s(mZ)$, Phys. Rev. D **83**, 074021 (2011).

[114] V. Andreev *et al.* (H1 Collaboration), Measurement of Lepton-Jet Correlation in Deep-Inelastic Scattering with the H1 Detector Using Machine Learning for Unfolding, Phys. Rev. Lett. **128**, 132002 (2022).

[115] M. Vandegar, M. Kagan, A. Wehenkel, and G. Louppe, Neural empirical bayes: Source distribution estimation and its applications to simulation-based inference, in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, edited by A. Banerjee and K. Fukumizu, Vol. 130 of Proceedings of Machine Learning Research (PMLR, 2021), pp. 2107–2115, https://proceedings.mlr.press/v130/vandegar21a.html.

[116] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, and R. Winterhalder, How to GAN away Detector Effects, SciPost Phys. **8**, 070 (2020).

[117] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, A. Rousselot, and R. Winterhalder, Invertible networks or partons to detector and back again, SciPost Phys. **9**, 074 (2020).

[118] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, OmniFold: A Method to Simultaneously Unfold All Observables, Phys. Rev. Lett. **124**, 182001 (2020).

[119] T. Sjöstrand and M. Utheim, Hadron interactions for arbitrary energies and species, with applications to cosmic rays, Eur. Phys. J. C **82**, 21 (2022).

[120] Y. Feng, M.-Y. Chu, U. Seljak, and P. McDonald, FASTPM: A new scheme for fast simulations of dark matter and haloes, Mon. Not. R. Astron. Soc. **463**, 2273 (2016).

[121] C. Modi, F. Lanusse, and U. Seljak, FlowPM: Distributed TensorFlow implementation of the FastPM cosmological N-body solver, Astron. Comput. **37**, 100505 (2021).

[122] B. Dai and U. Seljak, Learning effective physical laws for generating cosmological hydrodynamics with Lagrangian Deep Learning, Proc. Natl. Acad. Sci. U.S.A. **118**, e2020324118 (2021).

[123] V. Böhm, Y. Feng, M. E. Lee, and B. Dai, MADLens, a python package for fast and differentiable non-Gaussian lensing simulations, Astron. Comput. **36**, 100490 (2021).

[124] C. Modi, F. Lanusse, U. Seljak, D. N. Spergel, and L. Perreault-Levasseur, CosmicRIM: Reconstructing early universe by combining differentiable simulations with recurrent inference machines, arXiv:2104.12864.

[125] P. Ilten, T. Menzo, A. Youssef, and J. Zupan, Modeling hadronization using machine learning, arXiv:2203.04983.