Point cloud diffusion models for the Electron-Ion Collider

Jack Y. Araz
Center for Nuclear Theory, Department of Physics and Astronomy
Stony Brook University
New York 11794, USA
jack.araz@stonybrook.edu

Vinicius Mikuni
National Energy Research Scientific Computing Center
Berkeley Lab, Berkeley, CA 94720
vmikuni@lbl.gov
Felix Ringer
Department of Physics
Old Dominion University, Norfolk, VA 23529
felix.ringer@stonybrook.edu

Nobuo Sato
Thomas Jefferson National Accelerator Facility
Newport News, VA 23606
nsato@jlab.org

Fernando Torales Acosta Physics Division
Lawrence Berkeley National Laboratory Berkeley, CA 94720
ftoralesacosta@lbl.gov

Richard Whitehill
Department of Physics
Old Dominion University, Norfolk, VA 23529
rwhit058@odu.edu

Abstract

At high-energy collider experiments, generative models can be used for a wide range of tasks, including fast detector simulations, unfolding, searches of physics beyond the Standard Model, and inference tasks. In particular, it has been demonstrated that score-based diffusion models can generate high-fidelity and accurate samples of jets or collider events. This work expands on previous generative models in three distinct ways. First, our model is trained to generate entire collider events, including all particle species with complete kinematic information. We quantify how well the model learns event-wide constraints such as the conservation of momentum and discrete quantum numbers. We focus on the events at the future Electron-Ion Collider, but we expect that our results can be extended to proton-proton and heavy-ion collisions. Second, previous generative models often relied on image-based techniques. The sparsity of the data can negatively affect the fidelity and sampling time of the model. We address these issues using point clouds and a novel architecture combining edge creation with transformer modules called Point Edge Transformers. Third, we adapt the foundation model OmniLearn, to generate full collider events. This approach may indicate a transition

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

toward adapting and fine-tuning foundation models for downstream tasks instead of training new models from scratch.

1 Introduction

High-energy collider experiments enable probes of the internal dynamics of protons and nuclei, study emergent phenomena such as hadronization, and search for physics beyond the Standard Model (BSM) of particle physics. By analyzing the particles observed in detectors centered around the scattering vertex, it is possible to infer the dynamics of particles at subatomic scales. The next-generation experiment will be the future Electron-Ion Collider (EIC) [1], where high-luminosity electron-proton/nucleus scattering will be studied at center-of-mass (CM) energies up to $\sqrt{s} = 140$ GeV.

A key tool to advance different areas of collider phenomenology are generative models, and can aid in data deconvolution, searches for new physics, fast surrogate modeling, and much more [2]. Various architectures have been trained to generate collider events or jets including GANs [3, 4], Variational Autoencoders [5, 6], normalizing flows [7, 8, 9] and score-based diffusion models [10, 11, 12]. In particular, diffusion models have been demonstrated to produce high-fidelity samples and their generation speed has been improved significantly using techniques such as progressive distillation [13]. Score-based diffusion models are based on slowly perturbing data over time using a time parameter $t \in \mathbb{R}$ that determines the perturbation level. The task of the neural network is to approximate the gradients of the log probability of the data, $\log p_{\text{data}}(\mathbf{x})$, also called the score function $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \in \mathbb{R}^D$, based on data observations $\mathbf{x} \in \mathbb{R}^D$ in D-dimensional space. This can be approximated by a denoising score-matching strategy [14]. While denoising diffusion models were developed to work with images [15, 16], the work in [17] demonstrated the advantages of using point-clouds over images for collider physics applications. Point cloud based generative models for particle/nuclear physics applications have seen rapid development in recent years [18, 19, 20, 21, 22].

In this work, we build upon these earlier results in the literature to develop a point cloud-based diffusion model for full EIC event generation. The use of point clouds as well as a novel architecture combining edge creation with transformer modules, termed Point Edge Transformers, allows us to address several of the issues encountered in earlier work. We adapt the pre-existing foundation model OMNILEARN [23] to generate full collider events. The model was initially developed for both classification and generation tasks in the context of jet physics at the LHC, however we find that the model is very well suited to generate EIC events. While the work in [12] used images and generated only a small subset of the particle species in an event, this works generates point cloudes for all particles in the event. While we train the model developed here from scratch instead of fine tuning the original foundation model, the model architecture of OMNILEARN is not changed at all. Our results therefore point toward a transition toward adapting foundation models for downstream tasks at collider experiments.

2 Electron-Proton Data

Following Ref. [12], we generate events for electron-proton collisions using PYTHIA8 [24] at a center-of-mass energy of $\sqrt{s} = 105$ GeV, with an electron and proton beam energy of 10 GeV and 275 GeV, respectively. We avoid the low-Q^2 photoproduction region by imposing a cut of $Q^2 > 25$ GeV^2. We include all particles in the rapidity range $|\eta| < 5$ and we do not impose a lower cut on the transverse momentum. We include the following list of stable particles, defined as particles in the event with a lifetime $c\tau \geq 10$mm, in the data set.

$$e^{\pm}, \mu^{\pm}, \nu, \bar{\nu}, \pi^{\pm}, \pi^0, K^{\pm}, p, n, \gamma. \tag{1}$$

The data is categorized into event-level and particle level features, to support the two-model strategy described in Sec. 3. Due to its relevance in Deep Inelastic Scattering (DIS), the modeling of the scattered electron kinematics plays a critical role in electron-proton events that requires special attention, and are therefore categorized as event-level features. Additionally, the total number of particles in the event, N, of the event is used. The resulting set of event features is:

$$N,\ p_{\text{T}}^e,\ \eta^e,\ \phi^e. \tag{2}$$

For each particle i in the event, we record its transverse momentum p_{Ti}, pseudo-rapidity η_i, azimuthal angle ϕ_i, and Particle Identification (PID), and charge C. In addition, we consider the dimensionless

quantity

$$z_i = \frac{2p_{Ti}}{\sqrt{s}} \cosh \eta_i \,, \tag{3}$$

which is of particular interest for it's relevance in testing momentum conservation. The particle level features are normalized (or shifted) according to the scattered electron kinematics. This simplifies the learning task of the second diffusion model to just learning the relative p_{T} and η, for example, per particle, rather than the absolute scale of each feature.

The set of generated particle features is:

$$\log_{10}(p_{\mathrm{T}}/p_{\mathrm{T}}^{e}),\ \eta + \eta^{e},\ \phi,\ \log_{10}(z),\ C,\ \mathrm{PID}. \tag{4}$$

3 Model Architecture and Training

This work extends the generalized machine learning model, OMNILEARN [23], designed for analyzing data from particle physics experiments. Fine-tuning on DIS events was attempted, but did not perform well. This is likely because while the structure of OMNILEARN is well suited for whole event generation, it was originally trained for a significantly different set of physics tasks - a variety of Jet classification and recorstruction tasks in high-energy collisions. As a result, the model was re-trained on electron-proton DIS events.

A high level schematic of the model architecture is shown in Fig1. In all metrics investigated, the model shows similar or improved performance compared to previous models. The model processes inputs consisting of particles and event-level information and incorporates a parameter related to the diffusion time of these particles. The time information for the diffusion process, as done in previous diffusion models for collider physics [10, 25, 26, 21, 27], is encoded to a higher dimensional space using a time embedding layer. This embedding layer utilizes Fourier features [28] and is further processed by two multi-layer perceptrons (MLPs) employing a GELU activation function [29]. In this model, each MLP layer is followed by a non-linear GELU activation unless otherwise specified. The model then integrates the time-related data with particle-specific information, which includes both the kinematics of each particle and their particle identification (PID) code. These inputs are transformed into a higher dimensional space using a feature embedding composed of two MLP layers. The output from this embedding process is modified through a shift and scaling operation to merge it with the time-related data. Prior to the transformer block – which is responsible for processing data in a manner that considers the relationships between particles – we insert a positional token. This token encodes the geometric context surrounding each particle in the event, aiding the transformer in understanding local particle arrangements. Although transformers are capable of capturing broad correlations among particles, adding local geometric data typically enhances the model's performance by creating a latent representation aware of particle distances [30]. The local encoding is constructed using dynamic graph convolutional network (DGCNN) layers, which define each particle's neighborhood through a k-nearest neighbor algorithm, set to include precisely ten neighbors. The distances between these neighbors are measured in the specific pseudorapidity-azimuthal angle space. For each of the k-neighbors, edge features are defined by concatenating the particle features with the subtraction between those features and the features of each respective neighbor. These edge features are then processed by a multi-layer perception (MLP), followed by an average pooling operation performed across the dimensions of the neighbors.

We adopt the two-model strategy implemented in [10]. A model is trained to exclusively learn the kinematic information of the event, which is then utilized as conditional information for a diffusion model that processes particles as inputs. Most important for this process is the total number of particles in the event, N, which is learned by the first diffusion model. The multiplicity is then shared with the second diffusion model that generates the corresponding number of particles for that event.

Up to 50 particles are saved per event to be used during training, the maximum of all pythia events in the samples used. The training is carried out on the Perlmutter Supercomputer [31] using 128 GPUs simultaneously with Horovod [32] package for data distributed training. A local batch size of size 256 is used with model training up to 200 epochs. OMNILEARN is implemented in TENSORFLOW [33] with KERAS [34] backend. The cosine learning rate schedule [35] is used with an initial learning rate of 3×10^{-5}, increasing to $3\sqrt{128} \times 10^{-5}$ after three epochs and decreasing to 10^{-6} until the end of

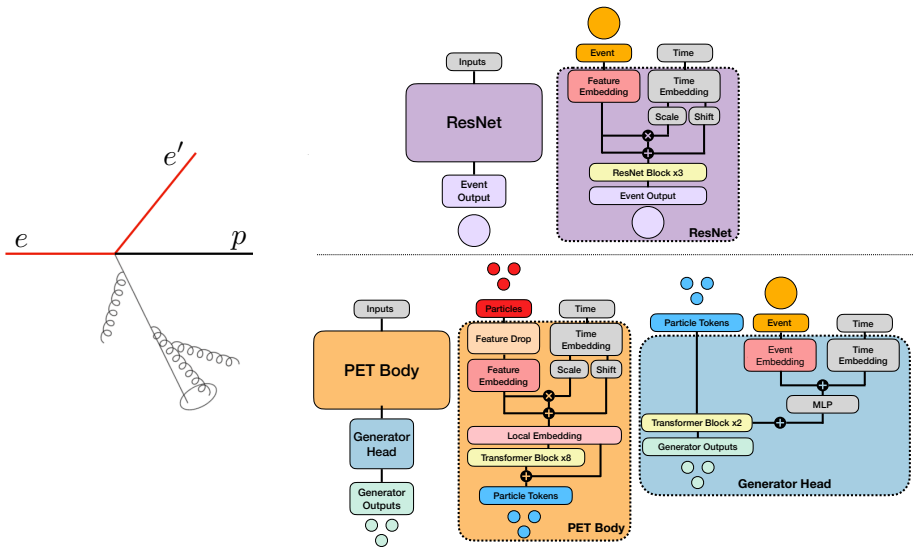

Figure 1: Left: Electron-proton scattering event $e + p \to e' + X$. Right: Model architecture adapted from the foundation model OmniLearn [23]. The final model is composed of two diffusion models: One that generates the scattered electron and the event properties, such as the multiplicity (top), and a second model that generates all other particles in the event with their kinematics (bottom)

the training. The LION optimizer [36] is used with parameters $\beta_1 = 0.95$ and $\beta_2 = 0.99$. The PET body model has 1.3M trainable weights, while the generator head has 416k trainable parameters.

4 Results

Figure 2 shows the distributions for all particles generated by the OMNILEARN model, as well as the original PYTHIA distribution. The diffusion model is extremely accurate, with only rare exceptions of a handfull of bins deviating. Similar distributions for each particle flavor (Eq. 2 are omitted for brevity, but are generated with very similar accuracy, with the exception of neutrinos that have deviations above the 20% threshold shown in this work. This is a marked improvement over previous image-based techniques for generating EIC events, where the center of the distributions had similar accuracy as this work, but the tails of the distribution exhibited deviations on the order of 200% [12]

Figure 3 shows the DIS kinematics of the generated and pythia events, as well as the ratio of the two. The plot shows $\log_{10} Q^2$ vs. $\log_{10} x$, both calculated using the beam energy and scattered electron kinematics. The diffusion model generates these important DIS quantities extremely accurately as well.

	$W_1^P(\eta)$	$W_1^P(\phi)$	$W_1^P(p_T)$	Cov	MMD	KPD
e^-	0.266 ± 0.009	0.015 ± 0.004	0.251 ± 0.004	0.546	0.166	0.0023 ± 0.0003
K	0.041 ± 0.003	0.025 ± 0.009	0.129 ± 0.005	0.518	0.382	0
π	0.310 ± 0.016	0.158 ± 0.003	0.464 ± 0.007	0.473	0.595	0.0062 ± 0.0009

Table 1: Comparison of the results obtained between generative model and pythia for the new diffusion model. Small values are preferred for each of the metrics except coverage.

To better evaluate the improvements achieved in this work compared to the image-based diffusion model from Ref. [12] for electron-proton scattering events, we present the values for several quantitative metrics in Table 2. We only focus on the comparison for electrons, kaons, and pions, as the image-based diffusion model in Ref. [12] was limited to these three particle species. Lower values of the different metrics indicate better results except for the coverage, where higher values are preferred.

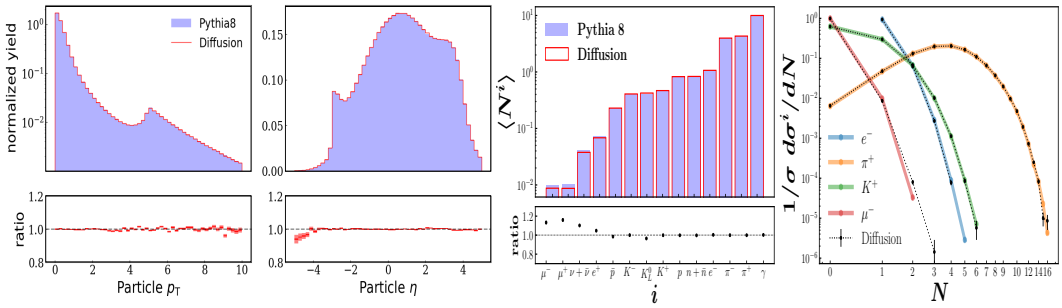

Figure 2: The p_{T} (left panel), η, (center left) and multiplicity (center right) of all particles is shown. The red lines in each plot represent the generated distribution, while the shaded light blue regions are the original PYTHIA8 distributions. The bottom panels of each distributions show the ratio of the generated model to PYTHIA8. The width of the red bands represent the statistical uncertainty in each bin. Lastly, comparison of the event-wide particle multiplicity distributions specific particle species (right panel), electrons e^-, pions π^+, kaons K^+, and muons μ^- in shown.

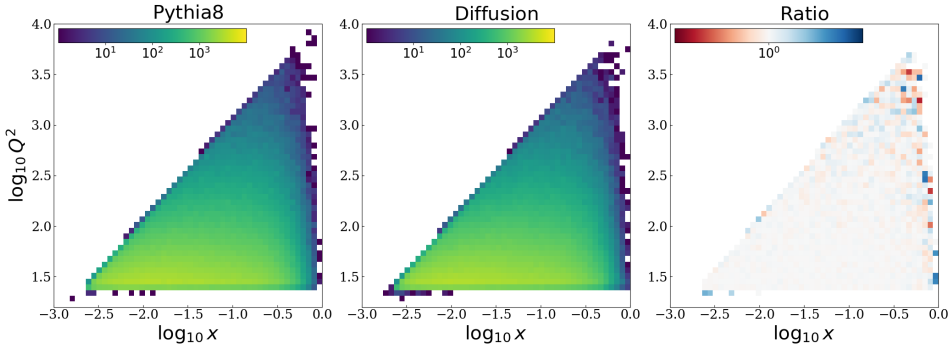

Figure 3: DIS kinematics of the PYTHIA8 data (left panel), the generated data (mid panel) and the ratio of the two (right panel). The DIS kinematics are calculated using the beam energy and scattered electron information, which is used in training the diffusion models.

	Image-based diffusion model, Ref. [12]			Point cloud-based diffusion model (this work)		
	e^-	K^+	π^+	e^-	K^+	π^+
$W_1^P(\eta)$	63.167 ± 0.035	36.669 ± 0.029	57.887 ± 0.062	0.266 ± 0.009	0.041 ± 0.003	0.310 ± 0.016
$W_1^P(\phi)$	18.910 ± 0.054	18.736 ± 0.048	18.789 ± 0.030	0.015 ± 0.004	0.025 ± 0.009	0.158 ± 0.003
$W_1^P(p_T)$	5.917 ± 0.005	0.323 ± 0.002	0.820 ± 0.007	0.251 ± 0.004	0.129 ± 0.005	0.464 ± 0.007
Cov	0.011	0.017	0.010	0.546	0.518	0.473
MMD	1.266	2.160	1.945	0.166	0.382	0.595
KPD	$7 \times 10^7 \pm 1 \times 10^7$	20.576 ± 26.608	$4.6 \times 10^3 \pm 1.5 \times 10^3$	0.0023 ± 0.0003	0	0.0062 ± 0.0009

Table 2: Metrics quantifying the performance of the image- and point cloud-based diffusion models compared to PYTHIA8. Small values are preferred for each metric except for the coverage.

While some metrics show a more significant improvement than others, the point cloud-based model presented here consistently outperforms the image-based diffusion model of Ref. [12]. This can be attributed to both its more advanced architecture and the loss of granularity in the image-based model due to pixelation.

5 Conclusion and Outlook

The diffusion models in OMNILEARN generate electron-proton collision as point clouds extremely accurately. It correctly generates the correct number of each individual particle and it's full kinematics, outperforming previous image-based approaches. At the same time, event-wide characteristics, such as the DIS kinematics, multiplicity, and momentum conservation are correctly learned.

This work presents the first standalone results obtained from a foundation model designed for high energy physics. No modifications to the model architecture were implemented, and simply a retraining of the model on the data specific to this work was required. This points to the transition of machine learning workflows in high energy physics from custom models designed for individual works, to shared, collaborative foundation models such as OMNILEARN.

References

[1] R. Abdul Khalek et al. Science Requirements and Detector Concepts for the Electron-Ion Collider: EIC Yellow Report. *Nucl. Phys. A*, 1026:122447, 2022.

[2] Matthew Feickert and Benjamin Nachman. A Living Review of Machine Learning for Particle Physics. 2 2021.

[3] Anni Li, Venkat Krishnamohan, Raghav Kansal, Rounak Sen, Steven Tsan, Zhaoyu Zhang, and Javier Duarte. Induced Generative Adversarial Particle Transformers. In *37th Conference on Neural Information Processing Systems*, 12 2023.

[4] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev.*, D97(1):014021, 2018.

[5] Abhishek Abhishek, Eric Drechsler, Wojciech Fedorko, and Bernd Stelzer. CaloDVAE : Discrete Variational Autoencoders for Fast Calorimeter Shower Simulation. 10 2022.

[6] Mary Touranakou, Nadezda Chernyavskaya, Javier Duarte, Dimitrios Gunopulos, Raghav Kansal, Breno Orzari, Maurizio Pierini, Thiago Tomei, and Jean-Roch Vlimant. Particle-based Fast Jet Simulation at the LHC with Variational Autoencoders. *Mach.Learn.Sci.Tech.*, 3:035003, 3 2022.

[7] Christina Gao, Stefan Höche, Joshua Isaacson, Claudius Krause, and Holger Schulz. Event Generation with Normalizing Flows. *Phys. Rev. D*, 101(7):076002, 2020.

[8] Claudius Krause and David Shih. CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows. *Phys.Rev.D*, 107:113004, 10 2021.

[9] Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. 3 2020.

[10] Vinicius Mikuni, Benjamin Nachman, and Mariel Pettee. Fast Point Cloud Generation with Diffusion Models in High Energy Physics. *Phys.Rev.D*, 108:036025, 4 2023.

[11] Lingxiao Wang, Gert Aarts, and Kai Zhou. Generative Diffusion Models for Lattice Field Theory. In *37th Conference on Neural Information Processing Systems*, 11 2023.

[12] Peter Devlin, Jian-Wei Qiu, Felix Ringer, and Nobuo Sato. Diffusion model approach to simulating electron-proton scattering events. *Phys. Rev. D*, 110(1):016030, 2024.

[13] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[14] Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 07 2011.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[16] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021.

[17] Fernando Torales Acosta, Vinicius Mikuni, Benjamin Nachman, Miguel Arratia, Bishnu Karki, Ryan Milton, Piyush Karande, and Aaron Angerami. Comparison of point cloud and image-based models for calorimeter fast simulation. *Journal of Instrumentation*, 19(05):P05003, may 2024.

[18] Raghav Kansal, Javier Duarte, Hao Su, Breno Orzari, Thiago Tomei, Maurizio Pierini, Mary Touranakou, Jean-Roch Vlimant, and Dimitrios Gunopulos. Particle Cloud Generation with Message Passing Generative Adversarial Networks. 6 2021.

[19] Erik Buhmann, Gregor Kasieczka, and Jesse Thaler. EPiC-GAN: Equivariant Point Cloud Generation for Particle Jets. *SciPost Phys.*, 15:130, 1 2023.

[20] Rob Verheyen. Event Generation and Density Estimation with Surjective Normalizing Flows. *SciPost Phys.*, 13:047, 5 2022.

[21] Matthew Leigh, Debajyoti Sengupta, Guillaume Quétant, John Andrew Raine, Knut Zoch, and Tobias Golling. PC-JeDi: Diffusion for Particle Cloud Generation in High Energy Physics. 3 2023.

[22] Erik Buhmann, Frank Gaede, Gregor Kasieczka, Anatolii Korol, William Korcari, Katja Krüger, and Peter McKeown. CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation. *JINST*, 19(04):P04020, 2024.

[23] Vinicius Mikuni and Benjamin Nachman. OmniLearn: A Method to Simultaneously Facilitate All Jet Physics Tasks. 4 2024.

[24] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An Introduction to PYTHIA 8.2. *Comput. Phys. Commun.*, 191:159–177, 2015.

[25] Erik Buhmann, Cedric Ewen, Gregor Kasieczka, Vinicius Mikuni, Benjamin Nachman, and David Shih. Full Phase Space Resonant Anomaly Detection. 10 2023.

[26] Vinicius Mikuni and Benjamin Nachman. High-dimensional and Permutation Invariant Anomaly Detection. 6 2023.

[27] Alexander Shmakov, Kevin Greif, Michael Fenton, Aishik Ghosh, Pierre Baldi, and Daniel Whiteson. End-To-End Latent Variational Diffusion Models for Inverse Problems in High Energy Physics. 5 2023.

[28] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[29] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[30] Vinicius Mikuni and Florencia Canelli. Point Cloud Transformers applied to Collider Physics. *Mach.Learn.Sci.Tech.*, 2:035027, 2 2021.

[31] Perlmutter system. `https://docs.nersc.gov/systems/perlmutter/system_details/`. Accessed: 2022-05-04.

[32] Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799*, 2018.

[33] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[34] Fraciois Chollet. Keras. `https://github.com/fchollet/keras`, 2017.

[35] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016.

[36] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.