Published by IOP Publishing for Sissa Medialab

Received: *January 24, 2025*
Accepted: *June 8, 2025*
Published: *July 4, 2025*

OmniJet-α_C: learning point cloud calorimeter simulations using generative transformers

Joschka Birk[a], **Frank Gaede**[b], **Anna Hallin**[a,*], **Gregor Kasieczka**[a], **Martina Mozzanica**[a] and **Henning Rose**[a]

[a] *Institute for Experimental Physics, Universität Hamburg*
Luruper Chaussee 149, 22761 Hamburg, Germany
[b] *Deutsches Elektronen-Synchrotron DESY,*
Notkestr. 85, 22607 Hamburg, Germany

E-mail: anna.hallin@uni-hamburg.de

Abstract: We show the first use of generative transformers for generating calorimeter showers as point clouds in a high-granularity calorimeter. Using the tokenizer and generative part of the OmniJet-α model, we represent the hits in the detector as sequences of integers. This model allows variable-length sequences, which means that it supports realistic shower development and does not need to be conditioned on the number of hits. Since the tokenization represents the showers as point clouds, the model learns the geometry of the showers without being restricted to any particular voxel grid.

Keywords: Analysis and statistical methods; Calorimeter methods; Data processing methods; Simulation methods and programs

ArXiv ePrint: 2501.05534

*Corresponding author.

https://doi.org/10.1088/1748-0221/20/07/P07007

Contents

1 Introduction 1

2 Dataset 2

3 Methods 3

4 Results 3
- 4.1 Token quality 4
- 4.2 Shower generation 5

5 Conclusion 7

A Model details and hyperparameters 8

B Postprocessing 9

C Generation quality 9
- C.1 Generation 9
- C.2 Correlations 9

2025 JINST 20 P07007

1 Introduction

Machine learning (ML) methods have been a common ingredient in particle physics research for a long time, with neural networks being applied to object identification already in analyses at LEP [1]. Since then, the range of applications has grown drastically, with ML methods being developed and used for example in tagging [2–4], anomaly detection [5–8], individual reconstruction stages like particle tracking [9–11] or even full event interpretation and reconstruction [12]. Another important use case for ML in high energy physics (HEP) is detector simulation. With the increasing luminosity of the large-scale experiments in HEP, the computational cost of high-precision Monte-Carlo (MC) simulations is going to exceed the available computing resources [13]. Generative methods have the potential to significantly reduce this resource requirement, which is why a considerable amount of research has been spent on exploring machine learning architectures for detector simulation [14, 15]. Examples include GANs [16–27], variational autoencoders (VAEs) and their variants [27–32], normalizing flows and various types of diffusion models [32–53].

Most ML methods in HEP are designed, developed and trained for very specific tasks. The focus on specialized models means that the full potential of the vast datasets we have access to is not being utilized. Furthermore, while these models may be more resource efficient than the traditional methods they seek to enhance or replace, developing and training each model from scratch still requires significant amounts of both human and computational resources. For reasons like these, there has been a growing interest in developing foundation models for particle physics [54–62] in the past couple of years. A foundation model is a machine learning model that has been pre-trained on a large amount

of data, and can then be fine-tuned for different downstream tasks [63]. The idea behind utilizing pre-trained models is that their outputs can significantly enhance the performance of downstream tasks, yielding better results than if the model were to be trained from scratch. While the models mentioned above have focused on exploring different tasks in specific subdomains, like jet physics, a more ambitious goal eventually would be to develop a foundation model for all tasks in all subdomains, including for example tracking, shower generation and anomaly detection in general (not restricted to jets). The hope would be that it could then utilize the full amount of diverse data from our experiments, to boost the performance of all possible downstream tasks. The first step towards such a model must be to be able to handle tasks from different subdomains in the same computational framework.

In this work, we apply the generative part of OmniJet-α [57], originally developed for jet physics, to a completely different subdomain: electromagnetic shower generation in collider calorimeters. We show that the OmniJet-α architecture and workflow also works for generating showers, opening up the possibility of exploring transfer learning for showers in a setting that has already proved successful in the context of jet physics. This is the first example of an autoregressive generative model utilizing the GPT architecture for calorimeter point clouds (as opposed to the fixed calorimeter geometries of ref. [64]). We denote this extended model capable of handling showers as OmniJet-α_C (OmniJet-α Calorimeter). Showing that we can use the same framework for two very different subdomains is an important step towards developing a foundation model for all computing and data analysis tasks in particle physics.

This paper is organized as follows. Section 2 describes the dataset used, section 3 the experimental setup, and section 4 presents the results. Finally, we offer our conclusions in section 5.

2 Dataset

The International Large Detector (ILD) [65] is one of two detector concepts proposed for the International Linear Collider (ILC) [66], an electron-positron collider that is initially operated at 250 GeV center-of-mass energy and extendable to higher energies up to 1 TeV. ILD is optimized for the Particle Flow Algorithm [67] that aims at reconstructing every individual particle. The detector therefore combines precise tracking and vertexing capabilities with good hermiticity, and highly granular sandwich calorimeters. The electromagnetic calorimeter of ILD (the Si-W ECAL [68]) consists of 20 layers with 2.1 mm thick W-absorbers followed by 10 layers with 4.2 mm W-absorbers, all interleaved with 0.5 mm thick Si-sensors that are subdivided into 5 mm $\times$ 5 mm cells.

The dataset used in this work was originally created for ref. [28], where more details on the detector and simulation can be found. While the dataset itself is not publicly available, it can be fully recreated by following the simulation and processing instructions provided in ref. [28]. Showers of photons with initial energies uniformly distributed between 10–100 GeV are simulated with Geant4 [69] using a detailed and realistic detector model implemented in DD4hep [70]. The resulting showers are projected into a regular 3D grid with $30 \times 30 \times 30 = 27\,000$ voxels. The 3D-grid data is converted into a point cloud format, where each point has four features: the x- and y-position (transverse to the incident particle direction), the z-position (parallel to the incident particle direction), and the energy. On average, each shower contains approximately 930, but not more than 1700 points with non-zero energy depositions, representing only a small fraction of the total 27,000 voxels in the grid. The incoming photon enters the calorimeter at perpendicular incident angle from the bottom at $z = 0$ and traverses along the z-axis, hitting cells in the center of the x-y plane. A staggered cell geometry results in small shifts between the layers.

2025 JINST 20 P07007

We preprocess the four input features (x, y, z and energy) by standardization. The energy feature is log-transformed before being scaled and shifted, which has the additional advantage that generated energies are by design non-negative.

The dataset has 950 000 samples, of which 760 000 are used for training, 95 000 for validation, and 95 000 as test samples.

3 Methods

This work uses the workflow of OmniJet-α [57], which is a foundation model originally developed for jet physics. We do not use a pretrained version of OmniJet-α, but rather implement the same autoregressive architecture and train it from scratch for generating calorimeter showers. OmniJet-α uses a VQ-VAE [56, 71–73] to tokenize the input features. The VQ-VAE transforms high-dimensional features into discrete latent representations by encoding the data and quantizing it to the nearest vectors in a learned codebook. The constituents of the jets, or in this case the voxel hits of the showers, are represented as a sequence of integers, which correspond to codebook vectors. A start token and a stop token are added to the beginning and the end of each sequence. These are special tokens that are needed for the autoregressive generation, as described in section 4.2. The sequences are used as input for the generative model, which is a GPT-style [74] model. Since the model only expects integers, it is not dependent on a specific type of data as input as long as it can be represented in this format. Moreover, the model accepts variable-length sequences, which means that it can be used equally well for jets with a variable number of constituents as for showers with a variable number of hits. The training target of the model is next token prediction, that is, it learns the probability of each token given a sequence of previous tokens, $p(x_i|x_{i-1}, \ldots, x_0)$. This means that it is straightforward to use the trained model for autoregressive generation, where each new token is generated conditioned on the previous ones in the sequence. While OmniJet-α also has classification capabilities, this work only focuses on the generative part. One key feature of OmniJet-α is that it learns the sequence length from context. This removes the need for specifying the number of elements in the sequence beforehand.

The VQ-VAE and generative model were trained using the hyperparameters described in appendix A. For the VQ-VAE, the best epoch was selected via lowest validation loss. After training, the VQ-VAE was frozen. The input data was tokenized using this model, and then fed into the generative model for training. Here again the epoch with the lowest validation loss was chosen as the best epoch. New showers in the form of integer sequences were then generated using this final generative model, and the frozen VQ-VAE was used to decode these integer sequences back into physical space.

4 Results

In the following we will present the results of the training of the VQ-VAE and the generative model. For comparison we use the test dataset, which the models never saw during training. As a benchmark for shower generation the performance of OmniJet-α_C is compared to two state-of-the-art generative networks: one point cloud model, CaloClouds II [75], and one fixed-grid model, L2LFlows [76]. CaloClouds II is a continuous time score-based diffusion model that has been further distilled into a consistency model (CM), whereas L2LFlows is a flow-based model using coupling flows with convolutional layers. L2LFlows has already been trained on this dataset in [76], and showers

2025 JINST 20 P07007

generated by this model were provided to us directly by the authors. For CaloClouds II however, no such training was available. Instead we ran this training ourselves, using the same hyperparameters as in [75] with the exception of training the diffusion model for 3.5 M iterations instead of 2 M, and the consistency model for 2.5 M iterations instead of 1 M. This is the first time CaloClouds II has been trained on a dataset in which the granularity matches the one of the calorimeter.

4.1 Token quality

We first investigate the encoding and decoding capabilities of the VQ-VAE. To judge the effect of the tokenization and potential loss of information, we compare the individual hits in the original showers with the corresponding hits in the reconstructed showers. A perfect reconstruction would yield a Dirac delta function for the difference between reconstructed and original values for each feature. However, as shown in figure 1, while the distributions surrounding the center are indeed narrow, they do have some spread. A codebook size of 65 536 shows a narrower resolution distribution than a codebook size of 8 192. In particular, the reconstruction of z for the latter has a larger spread of $\sigma^z_{8\,192} = 0.66$ layers compared to $\sigma^z_{65\,536} = 0.4$ layers with the larger codebook size. For the energy, the respective spread values are $\sigma^{\text{energy}}_{8\,192} = 0.11$ MeV and $\sigma^{\text{energy}}_{65\,536} = 0.07$ MeV. Furthermore, the reconstructed z distribution demonstrates a broader spread and a more complex reconstruction relative to the transverse coordinates x and y, which exhibit similar and narrower distributions. This difference in reconstruction accuracy can be attributed to a broader spatial extent of the showers along the longitudinal axis z. However, because voxels are discrete, the three spatial features need to be rounded to integers. Perfect resolution is achieved if these values remain within ±0.5 before rounding, the region indicated by the light gray lines in figure 1.

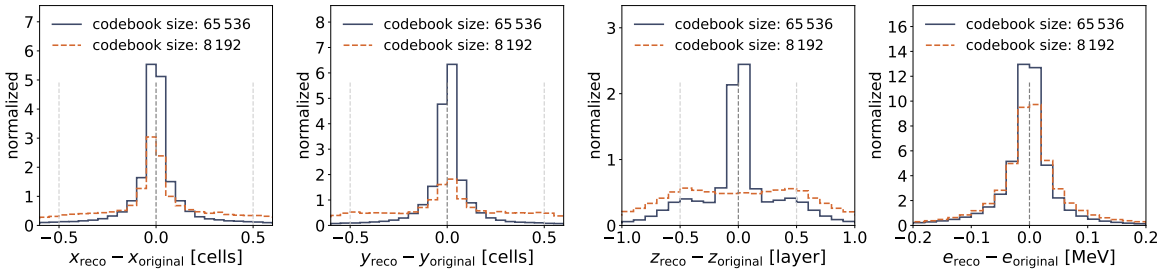

Figure 1. Reconstruction resolution for the input features (x, y, z, energy) for different codebook sizes.

To accurately compare the reconstructed showers with the original showers, we need to apply postprocessing. This step is explained in appendix B and essentially projects hits back into the voxel grid and processes duplicate hits (hits that are identical in all of the three spatial features). For the following analysis, showers are converted to tokens and then back to physical space. Figure 2 shows different feature distributions of the original and reconstructed showers, showcasing an overall good agreement between the two. Rare tokens, such as those located at the edges of the shower or tokens associated with high-energy hits, exhibit the lowest reconstruction quality. Again the VQ-VAE with the codebook size of 65 536 performs better and has the smallest loss of information and is selected for tokenizing the showers for the generative training.

2025 JINST 20 P07007

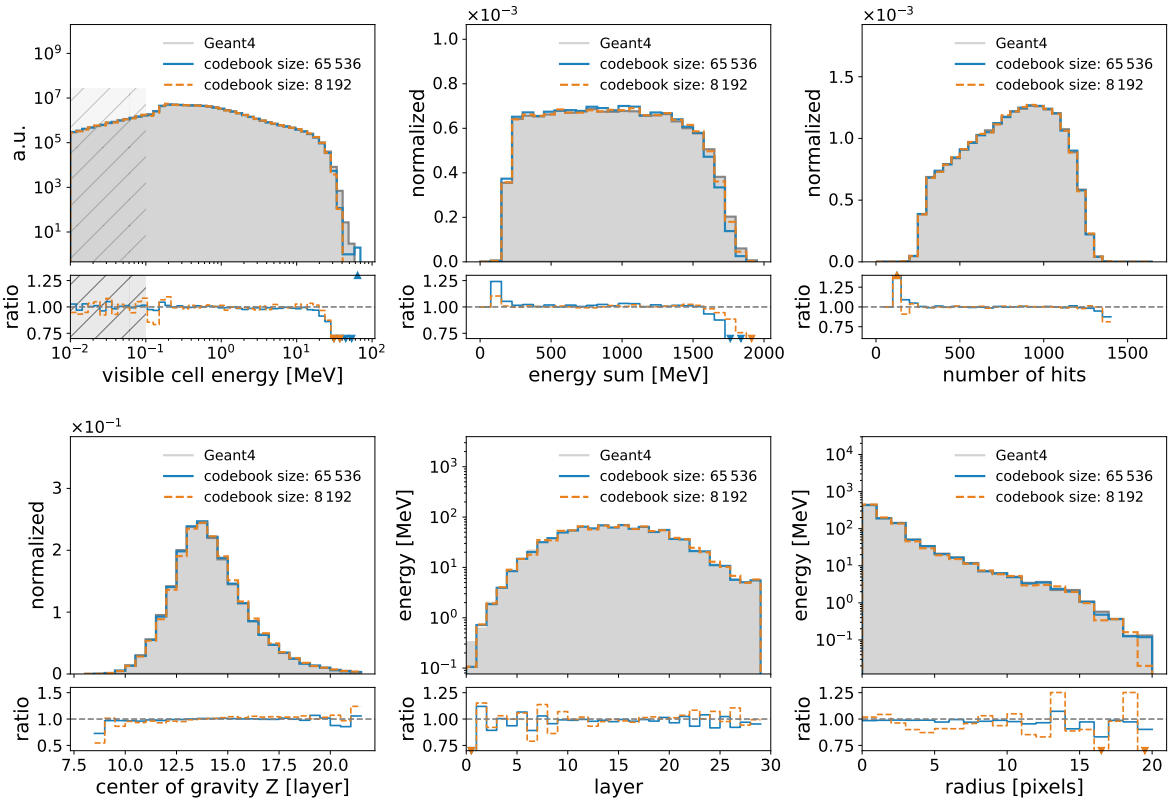

Figure 2. Distributions of physical observables between Geant4 (grey, filled) with the codebook size of 65 536 (blue) and codebook size of 8 192 (orange). Hits that were below the MIP threshold (0.1 MeV), i.e. those in the shaded region of the visible cell energy plot, were not considered for the comparison in the remaining distributions. This cutoff can affect the number of hits for reconstructed showers.

4.2 Shower generation

Following training, OmniJet-α_C generates point clouds autoregressively. Initialized with a start token (a special token that initiates the autoregressive generation process), the model predicts the probability distribution for the next token based on the preceding sequence. OmniJet-α_C then samples from this distribution, appending the chosen token to the growing sequence. This process continues until a stop token (a special token that represents the end of the generated sequence) is generated or the maximum sequence length of 1700 tokens is reached. Unlike most ML-based shower generators, OmniJet-α_C is not trained to generate showers for specific incident photon energies. Instead, the model learns to generate showers with a variety of energies. We reserve a study of how to condition the model on the incident energy for future work. This would allow the user to request showers of a specific energy. In this first version however, we will only compare the full spectrum of showers.

We see in figure 3 that OmniJet-α_C, CaloClouds II (CM) and L2LFlows generate showers that appear to be visually acceptable compared to Geant4. Next, we compare the performance of OmniJet-α_C to CaloClouds II (CM) and L2LFlows for three different quantities.[1]

[1]Note that compared to the original training of CaloClouds II in ref. [75], this training is done at physical, ie. lower, resolution.

2025 JINST 20 P07007

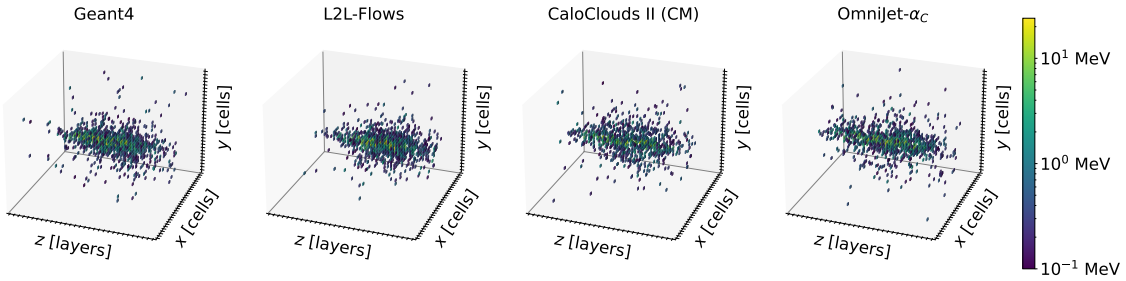

Figure 3. Examples of individual photon showers with a total energy sum of 1000 MeV generated by Geant4 (left), L2LFlows (center left), the CaloClouds II (CM) (center right) and OmniJet-α_C (right).

Figure 4 (left) compares the cell energies. We observe an accurate performance of OmniJet-α_C across almost the entire energy range, on par with L2LFlows. For the higher energies we see some deviations for both OmniJet-α_C and CaloClouds II (CM). As seen in figure 2, the mismodeling for OmniJet-α_C is introduced by the VQ-VAE. The behavior of CaloClouds II (CM) is consistent with what was seen in the original paper. The shaded area in the histogram corresponds to the region below half the energy of a minimal ionizing particle (MIP). In real detectors, read-outs at such small energies are dominated by noise. Therefore, cell energies below 0.1 MeV will not be considered in the following discussion, and the remaining plots and distributions only include cells above this cut-off.

Figure 4 (center) shows the distribution of the total energy sum of showers. For this calculation, the energy of all hits surpassing half the MIP energy are added up for each shower. This distribution is strongly correlated to the incident photon energy on which L2LFlows and CaloClouds II (CM) are conditioned. OmniJet-α_C has to learn this distribution on its own.

Finally, figure 4 (right) shows the number of hits. While the L2LFlows and CaloClouds II (CM) are conditioned on this distribution, OmniJet-α_C is able to achieve good agreement with the Geant4 distribution without this conditioning. The discrepancies we see are a small peak at a shower length of around 400 to 500, and also some showers that are too long.

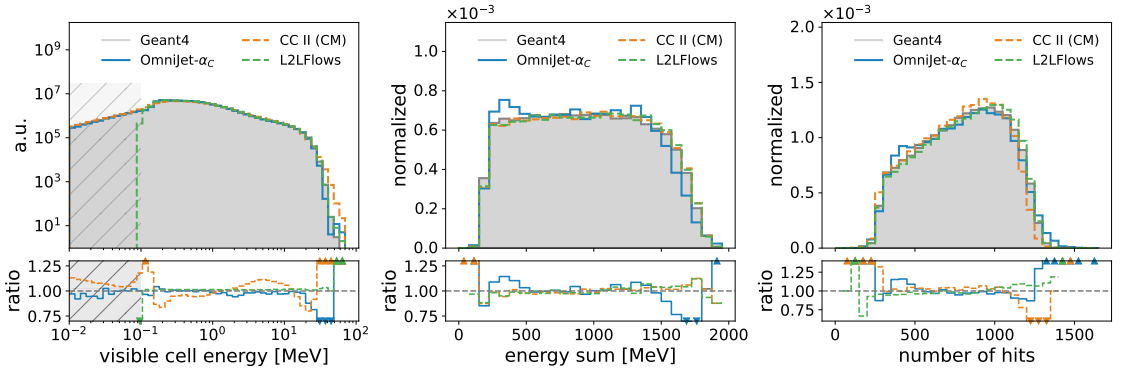

Figure 4. Distributions of per-cell energy (left), total energy sum (middle) and the number of hits above 0.1 MeV (right) between Geant4 (grey, filled) and the generative models: OmniJet-α_C (blue), the CaloClouds II (CM) (orange, dashed) and L2LFlows (green, dashed).

In figure 5 we compare the spatial properties of the shower. The left plot shows that the Geant4 distribution of the center of gravity along the z-axis is well modeled by all three architectures. OmniJet-α_C performs better in the center of the peak than at the edges.

2025 JINST 20 P07007

The longitudinal energy distribution, depicted in the middle plot of figure 5, reveals a comparatively weaker performance of the OmniJet-α_C model and CaloClouds II (CM) compared to L2LFlows in the initial 10 layers. However, OmniJet-α_C outperforms CaloClouds II (CM) in the first 4 layers. The mismodeling of OmniJet-α_C in the initial layers is likely attributable to the tokenization process (see figure 2), where these layers, being less common, are represented by a limited number of tokens. A similar degradation is observed in the outer regions of the radial energy distribution (right plot of figure 5), although OmniJet-α_C still outperforms CaloClouds II (CM).

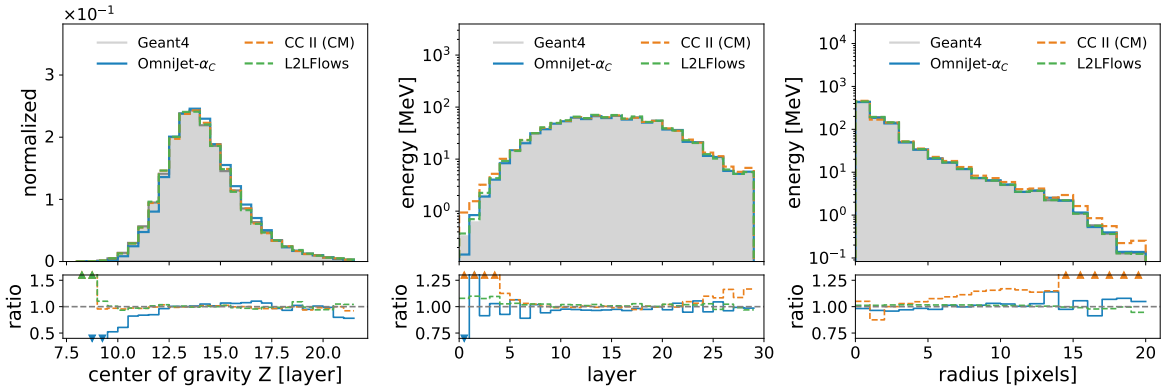

Figure 5. Distributions of center of gravity (left), mean energy per layer (middle) and the radius (right) between Geant4 (grey, filled) and the generative models: OmniJet-α_C (blue), the CaloClouds II (CM) (orange, dashed) and L2LFlows (green, dashed).

Another important aspect for comparing generative models is the single-shower generation time. Generating 1000 showers with OmniJet-α_C, randomly sampled across all incident energies, resulted in a mean and standard deviation of (3 ± 1) s per shower. The generation was performed with a batch size of 2 on an NVIDIA® A100 GPU. In contrast, Geant4 on a CPU required (4.1 ± 0.2) s per shower [28]. Therefore, our model demonstrates a speedup factor of 1.39 in this case. On identical hardware and with a batch size of 1000, L2LFlows achieves per-shower generation times of (3.24 ± 0.05) ms and a speedup factor of 1260. CaloClouds II on identical hardware but with a batch size of 100 generates one shower in (16 ± 6) ms and achieves a speedup factor of 255. The comparatively slow performance of OmniJet-α_C is attributable to the characteristic quadratic scaling $O(N^2)$ of the autoregressive transformer architecture with respect to the sequence length. Since this study did not prioritize generation speed, optimizations such as multi-token generation are left for future work.

5 Conclusion

In this work, we take a first important step towards building a foundation model for several subdomains of particle physics. We show that we are able to use the architecture and workflow of a foundation model originally developed for jet physics to generate electromagnetic showers in a calorimeter, a fundamentally different problem. This is a notable difference to previous efforts for foundation models in HEP, which so far focused on tasks within one subdomain, mostly different tasks within jet physics. Our work demonstrates that the same architecture can be successfully reused across distinct physical domains, without significant modifications. It is also the first implementation of a GPT-style autoregressive generative model for calorimeter shower point cloud generation.

2025 JINST 20 P07007

The next immediate step will be to investigate whether this model can be used for transfer learning between different types of showers. In the long term, we aim to develop a joint model that can work with both jets and showers. Combining tasks from different subdomains in one single framework is a necessary step towards a foundation model for particle physics that can handle a variety of data types and tasks.

Acknowledgments

The authors would like to thank William Korcari for support with the dataset, as well as Thorsten Buss for providing the L2LFlows samples. JB, AH, GK, MM and HR are supported by the DFG under the German Excellence Initiative — EXC 2121 Quantum Universe – 390833306, and by PUNCH4NFDI – project number 460248186. This work has used the Maxwell computational resources at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

A Model details and hyperparameters

Different hyperparameter configurations were tested for the individual model components of OmniJet-α_C. The configurations presented in the following were found to lead to stable trainings. However, no extensive hyperparameter optimization was performed.

Table 1. Hyperparameters used in the VQ-VAE training.

Hyperparameter	**Value**
Learning rate	0.001
Optimizer	Ranger
Batch size	152
Batches per epoch	1000
Number of epochs	588
Hidden dimension	128
Codebook size	65 536
β	0.8
α	10
Replacement frequency	100

The hyperparameters used for the VQ-VAE training are shown in table 1. Only the codebook size, replacement frequency and the hyperparameter β were adjusted. The remaining hyperparameters are the same as in OmniJet-α. An increase of the codebook size from 8 192 to 65 536 was found to improve the reconstruction capabilities (i.e. the resolution of the tokenized showers). The codebook utilization, i.e. the fraction of used tokens, is also monitored during the training to ensure that the resulting codebook is used completely. Unused tokens would drastically increase the number of parameters of the generative model while not adding any potential improvements in the performance of the generative model. The current setup results in a codebook utilization of the final VQ-VAE model of 99.65%. The hyperparameter β which defines the relative importance of how much weight should be given to updating the encoder embeddings z_e towards the codebook vectors z_q and vice versa, is decreased from 0.9 to 0.8. This leads to a higher emphasis on adapting the encoder to bring the embeddings z_e closer to

2025 JINST 20 P07007

the codebook vectors z_q. Furthermore, the optimization process employs a token replacement strategy based on usage frequency. The chosen replacement frequency of 100 batches (instead of 10) indicates that a token must be used at least once within the preceding 100 batches to avoid being replaced by a new token. We used the Lookahead optimizer [77] with RAdam as the inner optimizer [78].

For the hyperparameters of the backbone, no changes compared to OmniJet-α were made except for the batch size. The hyperparameters used are listed in table 2.

Table 2. Hyperparameters used in the generative model training.

Hyperparameter	**Value**
Learning rate	0.001
Optimizer	Ranger
Batch size	72
Batches per epoch	6000
Number of epochs	106
Embedding dimension	256
Number of heads	8
Number of GPT blocks	3

B Postprocessing

Projecting the hits of a point cloud model back onto the voxel grid can result in duplicate hits in some voxels. To resolve these duplicates, the voxels with lower energy are translated along the z-axis to the nearest unoccupied voxel position. This heuristic preserves both the total energy and the hit count while minimally impacting the z-distribution. We could also translate the voxels along the x- or y-axis, but as shown in figure 6 the hit energies are not invariant in these directions.

C Generation quality

C.1 Generation

To isolate the impact of the tokenization and subsequent reconstruction from that of the generative model itself, figure 7 compares showers generated by OmniJet–α_C to Geant4 showers that were first tokenized and then reconstructed. This figure should be viewed in context of figure 2, which shows the performance of the tokenization and reconstruction, and figures 4 and 5 which show the generated showers compared to the original Geant4 showers. The same discrepancy in the energy sum and number of hits that was seen in figure 4 can also be seen here. This means that this is an effect of the generative model itself, not the VQ-VAE.

C.2 Correlations

Figure 8 summarizes how well each model preserves the linear relationships among shower observables. The relation between number of hits and energy sum, which is an important relation in the calorimeter context, is preserved in all models. The correlations between the other shower features are weak in the Geant4 baseline, a characteristic that is also reflected by all models.

2025 JINST 20 P07007

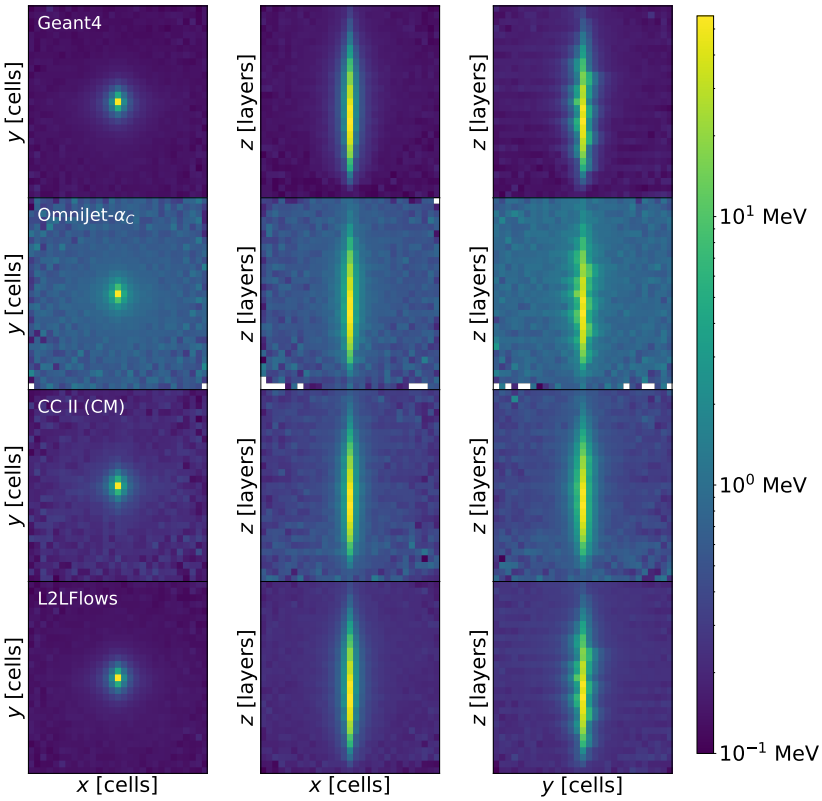

Figure 6. Overlay of 10k showers for all simulators for the full spectrum, where the voxel energies are summed along the z- (left), y- (middle) and x-axis (right). In all plots, the mean over the number of showers is taken.

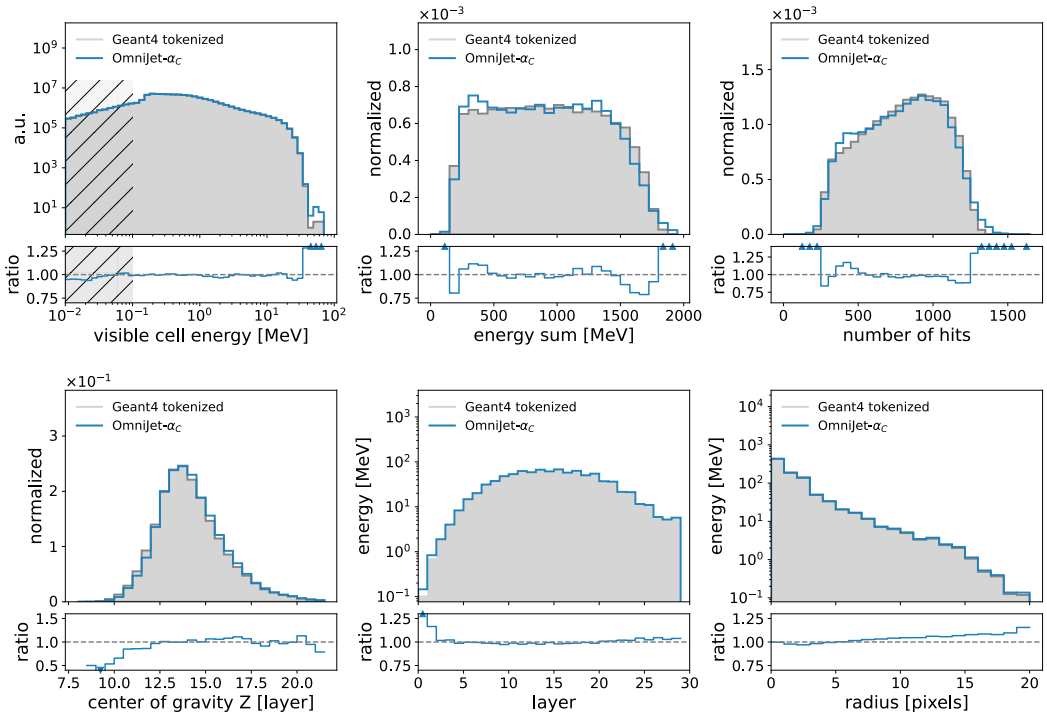

Figure 7. Distributions for Geant4 after tokenization and reconstruction (gray) and generated showers by OmniJet-α_C (blue).

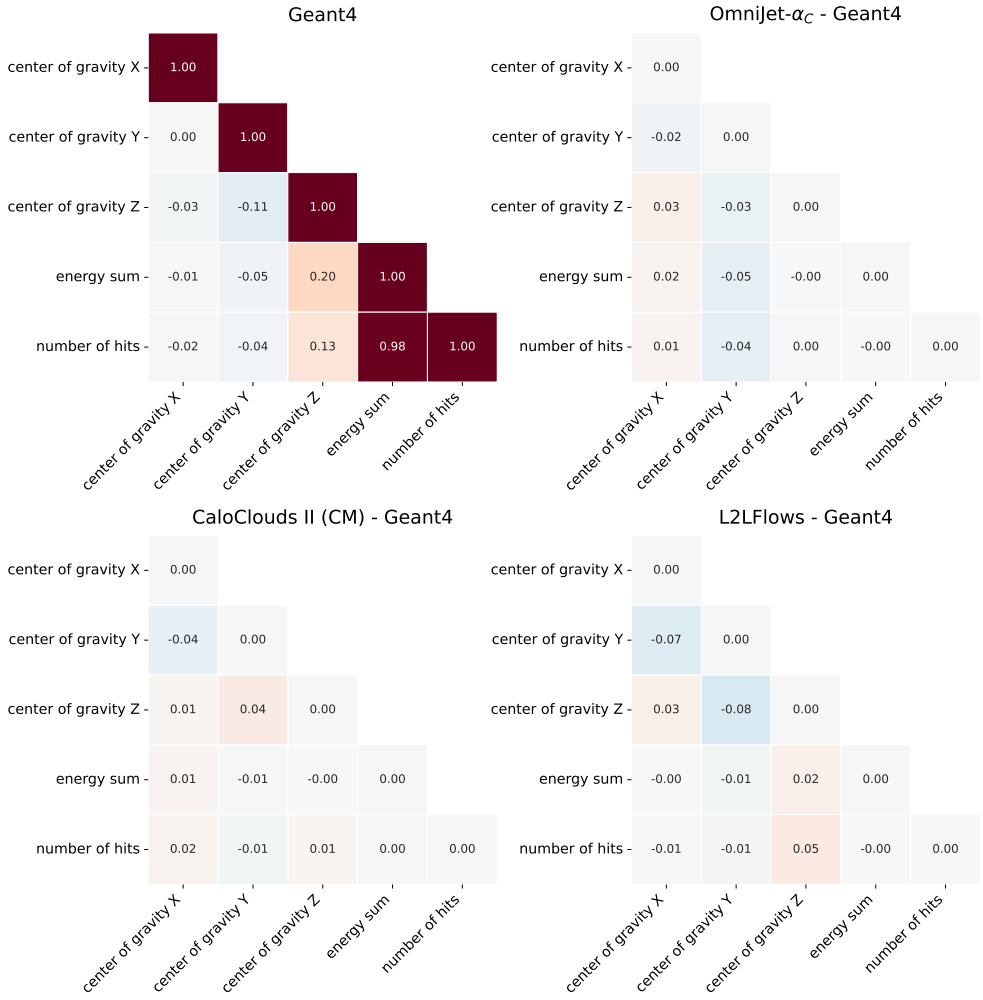

Figure 8. Linear correlation of shower features for Geant4 (top left) and the residuals for OmniJet-α_C (top right), CaloClouds II (CM) (bottom left) and L2LFlows (bottom right).

Code Availability Statement. This article has associated code in a code repository. The code for this paper can be found at https://github.com/uhh-pd-ml/omnijet_alpha_c. The code for recreating the dataset can be found at https://github.com/FLC-QU-hep/getting_high.

References

[1] T. Behnke and D.G. Charlton, *Electroweak measurements using heavy quarks at LEP*, *Phys. Scripta* **52** (1995) 133.

[2] ATLAS collaboration, *Flavour Tagging with Graph Neural Network at ATLAS*, ATL-PHYS-PROC-2024-081, CERN, Geneva (2024).

[3] ALICE collaboration, *Particle identification with machine learning from incomplete data in the ALICE experiment*, 2024 *JINST* **19** C07013 [arXiv:2403.17436].

[4] S. Mondal and L. Mastrolorenzo, *Machine learning in high energy physics: a review of heavy-flavor jet tagging at the LHC*, *Eur. Phys. J. ST* **233** (2024) 2657 [arXiv:2404.01071].

2025 JINST 20 P07007

[5] ATLAS collaboration, *Dijet resonance search with weak supervision using $\sqrt{s}$ = 13 TeV pp collisions in the ATLAS detector*, *Phys. Rev. Lett.* **125** (2020) 131801 [arXiv:2005.02983].

[6] ATLAS collaboration, *Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic final states using $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector*, *Phys. Rev. D* **108** (2023) 052009 [arXiv:2306.03637].

[7] ATLAS collaboration, *Search for New Phenomena in Two-Body Invariant Mass Distributions Using Unsupervised Machine Learning for Anomaly Detection at s=13 TeV with the ATLAS Detector*, *Phys. Rev. Lett.* **132** (2024) 081801 [arXiv:2307.01612].

[8] CMS collaboration, *Model-agnostic search for dijet resonances with anomalous jet substructure in proton–proton collisions at $\sqrt{s}$ = 13 TeV*, *Rept. Prog. Phys.* **88** (2025) 067802 [arXiv:2412.03747].

[9] ATLAS collaboration, *Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain*, ATL-SOFT-PROC-2023-047, CERN, Geneva (2023).

[10] ATLAS collaboration, *Computational Performance of the ATLAS ITk GNN Track Reconstruction Pipeline*, ATL-PHYS-PUB-2024-018, CERN, Geneva (2024).

[11] A. Correia et al., *Graph Neural Network-Based Pipeline for Track Finding in the Velo at LHCb*, in the proceedings of the *Connecting The Dots 2023*, Toulouse, France, October 10–13 (2023) [arXiv:2406.12869].

[12] J. García Pardinas et al., *GNN for Deep Full Event Interpretation and Hierarchical Reconstruction of Heavy-Hadron Decays in Proton–Proton Collisions*, *Comput. Softw. Big Sci.* **7** (2023) 12 [arXiv:2304.08610].

[13] A. Adelmann et al., *New directions for surrogate models and differentiable programming for High Energy Physics detector simulation*, in the proceedings of the *Snowmass 2021*, Seattle, U.S.A., July 17–26 (2022) [arXiv:2203.08806].

[14] S. Badger et al., *Machine learning and LHC event generation*, *SciPost Phys.* **14** (2023) 079 [arXiv:2203.07460].

[15] O. Amram et al., *CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation*, arXiv:2410.21611.

[16] M. Paganini, L. de Oliveira and B. Nachman, *Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters*, *Phys. Rev. Lett.* **120** (2018) 042003 [arXiv:1705.02355].

[17] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev. D* **97** (2018) 014021 [arXiv:1712.10321].

[18] L. de Oliveira, M. Paganini and B. Nachman, *Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters*, *J. Phys. Conf. Ser.* **1085** (2018) 042017 [arXiv:1711.08813].

[19] 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** (2018) 4 [arXiv:1802.03325].

[20] P. Musella and F. Pandolfi, *Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks*, *Comput. Softw. Big Sci.* **2** (2018) 8 [arXiv:1805.00850].

[21] M. Erdmann, J. Glombitza and T. Quast, *Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network*, *Comput. Softw. Big Sci.* **3** (2019) 4 [arXiv:1807.01954].

2025 JINST 20 P07007

[22] D. Belayneh et al., *Calorimetry with deep learning: particle simulation and reconstruction for collider physics*, *Eur. Phys. J. C* **80** (2020) 688 [arXiv:1912.06794].

[23] A. Butter et al., *GANplifying event samples*, *SciPost Phys.* **10** (2021) 139 [arXiv:2008.06545].

[24] ATLAS collaboration, *The Fast Simulation Chain in the ATLAS experiment*, *EPJ Web Conf.* **251** (2021) 03012.

[25] S. Bieringer et al., *Calomplification — the power of generative calorimeter models*, 2022 *JINST* **17** P09028 [arXiv:2202.07352].

[26] B. Hashemi et al., *Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning*, *Nature Commun.* **15** (2024) 4916 [*Erratum ibid.* **115** (2024) 5825] [arXiv:2303.08046].

[27] ATLAS collaboration, *Deep Generative Models for Fast Photon Shower Simulation in ATLAS*, *Comput. Softw. Big Sci.* **8** (2024) 7 [arXiv:2210.06204].

[28] E. Buhmann et al., *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*, *Comput. Softw. Big Sci.* **5** (2021) 13 [arXiv:2005.05334].

[29] E. Buhmann et al., *Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network*, *EPJ Web Conf.* **251** (2021) 03003 [arXiv:2102.12491].

[30] E. Buhmann et al., *Hadrons, better, faster, stronger*, *Mach. Learn. Sci. Tech.* **3** (2022) 025014 [arXiv:2112.09709].

[31] J.C. Cresswell et al., *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*, in the proceedings of the *36th Conference on Neural Information Processing Systems: Workshop on Machine Learning and the Physical Sciences*, New Orleans, LA, U.S.A., December 3 (2022) [arXiv:2211.15380].

[32] S. Diefenbacher et al., *New angles on fast calorimeter shower simulation*, *Mach. Learn. Sci. Tech.* **4** (2023) 035044 [arXiv:2303.18150].

[33] J. Sohl-Dickstein, E.A. Weiss, N. Maheswaranathan and S. Ganguli, *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*, arXiv:1503.03585.

[34] Y. Song and S. Ermon, *Generative Modeling by Estimating Gradients of the Data Distribution*, arXiv:1907.05600.

[35] Y. Song and S. Ermon, *Improved Techniques for Training Score-Based Generative Models*, arXiv:2006.09011.

[36] J. Ho, A. Jain and P. Abbeel, *Denoising Diffusion Probabilistic Models*, arXiv:2006.11239.

[37] Y. Song et al., *Score-Based Generative Modeling through Stochastic Differential Equations*, arXiv:2011.13456.

[38] V. Mikuni and B. Nachman, *Score-based generative models for calorimeter shower simulation*, *Phys. Rev. D* **106** (2022) 092009 [arXiv:2206.11898].

[39] E. Buhmann et al., *CaloClouds: fast geometry-independent highly-granular calorimeter simulation*, 2023 *JINST* **18** P11025 [arXiv:2305.04847].

[40] F.T. Acosta et al., *Comparison of point cloud and image-based models for calorimeter fast simulation*, 2024 *JINST* **19** P05003 [arXiv:2307.04780].

[41] V. Mikuni and B. Nachman, *CaloScore v2: single-shot calorimeter shower simulation with diffusion models*, 2024 *JINST* **19** P02001 [arXiv:2308.03847].

2025 JINST 20 P07007

[42] O. Amram and K. Pedro, *Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation*, *Phys. Rev. D* **108** (2023) 072014 [arXiv:2308.03876].

[43] C. Chen et al., *Analysis-Specific Fast Simulation at the LHC with Deep Learning*, *Comput. Softw. Big Sci.* **5** (2021) 15.

[44] C. Krause and D. Shih, *Fast and accurate simulations of calorimeter showers with normalizing flows*, *Phys. Rev. D* **107** (2023) 113003 [arXiv:2106.05285].

[45] C. Krause and D. Shih, *Accelerating accurate simulations of calorimeter showers with normalizing flows and probability density distillation*, *Phys. Rev. D* **107** (2023) 113004 [arXiv:2110.11377].

[46] S. Schnake, D. Krücker and K. Borras, *CaloPointFlow II Generating Calorimeter Showers as Point Clouds*, arXiv:2403.15782.

[47] C. Krause, I. Pang and D. Shih, *CaloFlow for CaloChallenge dataset 1*, *SciPost Phys.* **16** (2024) 126 [arXiv:2210.14245].

[48] A. Xu, S. Han, X. Ju and H. Wang, *Generative machine learning for detector response modeling with a conditional normalizing flow*, 2024 *JINST* **19** P02003 [arXiv:2303.10148].

[49] M.R. Buckley, C. Krause, I. Pang and D. Shih, *Inductive simulation of calorimeter showers with normalizing flows*, *Phys. Rev. D* **109** (2024) 033006 [arXiv:2305.11934].

[50] F. Ernst et al., *Normalizing Flows for High-Dimensional Detector Simulations*, *SciPost Phys.* **18** (2025) 081 [arXiv:2312.09290].

[51] M. Omana Kuttan, K. Zhou, J. Steinheimer and H. Stöcker, *Towards a foundation model for heavy-ion collision experiments through point cloud diffusion*, arXiv:2412.10352.

[52] L. Favaro, A. Ore, S.P. Schweitzer and T. Plehn, *CaloDREAM — Detector Response Emulation via Attentive flow Matching*, *SciPost Phys.* **18** (2025) 088 [arXiv:2405.09629].

[53] J. Brehmer et al., *A Lorentz-Equivariant Transformer for All of the LHC*, arXiv:2411.00446.

[54] T. Kishimoto, M. Morinaga, M. Saito and J. Tanaka, *Pre-training strategy using real particle collision data for event classification in collider physics*, in the proceedings of the *37th Conference on Neural Information Processing Systems*, New Orleans, U.S.A., December 10–16 (2023) [arXiv:2312.06909].

[55] H. Qu, C. Li and S. Qian, *Particle Transformer for Jet Tagging*, arXiv:2202.03772.

[56] T. Golling et al., *Masked particle modeling on sets: towards self-supervised high energy physics foundation models*, *Mach. Learn. Sci. Tech.* **5** (2024) 035074 [arXiv:2401.13537].

[57] J. Birk, A. Hallin and G. Kasieczka, *OmniJet-α: the first cross-task foundation model for particle physics*, *Mach. Learn. Sci. Tech.* **5** (2024) 035031 [arXiv:2403.05618].

[58] P. Harris et al., *Resimulation-based self-supervised learning for pretraining physics foundation models*, *Phys. Rev. D* **111** (2025) 032010 [arXiv:2403.07066].

[59] V. Mikuni and B. Nachman, *Solving key challenges in collider physics with foundation models*, *Phys. Rev. D* **111** (2025) L051504 [arXiv:2404.16091].

[60] A.J. Wildridge et al., *Bumblebee: Foundation Model for Particle Physics Discovery*, in the proceedings of the *38th conference on Neural Information Processing Systems*, Vancouver, Canada, December 09–15 (2024) [arXiv:2412.07867].

[61] O. Amram et al., *Aspen Open Jets: Unlocking LHC Data for Foundation Models in Particle Physics*, arXiv:2412.10504.

[62] J. Ho, B.R. Roberts, S. Han and H. Wang, *Pretrained Event Classification Model for High Energy Physics Analysis*, arXiv:2412.10665.

2025 JINST 20 P07007

[63] R. Bommasani et al., *On the Opportunities and Risks of Foundation Models*, arXiv:2108.07258.

[64] Q. Liu et al., *Calo-VQ: Vector-Quantized Two-Stage Generative Model in Calorimeter Simulation*, arXiv:2405.06605.

[65] ILD Concept Group collaboration, *International Large Detector: Interim Design Report*, arXiv:2003.01116.

[66] T. Behnke et al., *The International Linear Collider Technical Design Report — Volume 1: Executive Summary*, arXiv:1306.6327.

[67] M.A. Thomson, *Particle flow calorimetry*, *J. Phys. Conf. Ser.* **293** (2011) 012021.

[68] T. Suehara et al., *Performance study of SKIROC2/A ASIC for ILD Si-W ECAL*, 2018 *JINST* **13** C03015 [arXiv:1801.02024].

[69] GEANT4 collaboration, *GEANT4 - A Simulation Toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250.

[70] M. Frank, F. Gaede, C. Grefe and P. Mato, *DD4hep: A Detector Description Toolkit for High Energy Physics Experiments*, *J. Phys. Conf. Ser.* **513** (2014) 022010.

[71] A. van den Oord, O. Vinyals and K. Kavukcuoglu, *Neural Discrete Representation Learning*, arXiv:1711.00937.

[72] H. Bao, L. Dong, S. Piao and F. Wei, *BEiT: BERT Pre-Training of Image Transformers*, arXiv:2106.08254.

[73] M. Huh, B. Cheung, P. Agrawal and P. Isola, *Straightening Out the Straight-Through Estimator: Overcoming Optimization Challenges in Vector Quantized Networks*, arXiv:2305.08842.

[74] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, *Improving language understanding by generative pre-training*, 2018.

[75] E. Buhmann et al., *CaloClouds II: ultra-fast geometry-independent highly-granular calorimeter simulation*, 2024 *JINST* **19** P04020 [arXiv:2309.05704].

[76] T. Buss et al., *Convolutional L2LFlows: generating accurate showers in highly granular calorimeters using convolutional normalizing flows*, 2024 *JINST* **19** P09003 [arXiv:2405.20407].

[77] M.R. Zhang, J. Lucas, G. Hinton and J. Ba, *Lookahead Optimizer: k steps forward, 1 step back*, arXiv:1907.08610.

[78] H. Yong, J. Huang, X. Hua and L. Zhang, *Gradient Centralization: A New Optimization Technique for Deep Neural Networks*, arXiv:2004.01461.

2025 JINST 20 P07007