BitHEP — The limits of low-precision ML in HEP

Claudius Krause[1], **Daohan Wang**[1] **and Ramon Winterhalder**[2]

1 HEPHY, Austrian Academy of Sciences (OeAW), Vienna, Austria
2 TIFLab, Università degli Studi di Milano & INFN Sezione di Milano, Italy

Abstract

The increasing complexity of modern neural network architectures demands fast and memory-efficient implementations to mitigate computational bottlenecks. In this work, we evaluate the recently proposed BitNet architecture in HEP applications, assessing its performance in classification, regression, and generative modeling tasks. Specifically, we investigate its suitability for quark-gluon discrimination, SMEFT parameter estimation, and detector simulation, comparing its efficiency and accuracy to state-of-the-art methods. Our results show that while BitNet consistently performs competitively in classification tasks, its performance in regression and generation varies with the size and type of the network, highlighting key limitations and potential areas for improvement.

Received 2025-05-26
Accepted 2026-01-07
Published 2026-02-10
doi:10.21468/SciPostPhys.20.2.038

Check for updates

Contents

1 Introduction

The upcoming high-luminosity phase of the LHC (HL-LHC) will push the boundaries of precision measurements and new physics searches, necessitating unprecedented advances in computational methods. The ability to efficiently generate and analyze vast amounts of collision data is essential for maximizing the scientific potential of the HL-LHC. To this end, modern machine learning (ML) techniques have become indispensable tools in high-energy physics, enabling more precise and efficient modeling of complex physical processes [1, 2].

Machine Learning has already demonstrated its utility in various HEP domains, accelerating various aspects of a sophisticated simulation and analysis chain. In particular, for event generation, deep learning models are widely used for tasks such as scattering amplitude evaluations [3–10], phase-space sampling [11–22], parton shower generation [23–30], and detector simulations [31–58]. Similarly, for inference and analysis, ML-based approaches have been extensively used for numerous applications, including for instance quark-gluon tagging [59–84], and parameter estimation [85–124].

However, many of these approaches face significant scalability challenges when deployed in realistic experimental environments. For example, real-time triggering, event reconstruction, and particle tracking applications require ultra-fast inference on resource-constrained hardware such as field-programmable gate arrays (FPGAs) [125–142], where the complexity of deep neural networks poses a significant bottleneck. Other tasks, like detector simulation or recently-proposed foundation models [143–151] require larger and larger networks, which in turn need more disk space to be stored and energy to run.

A promising avenue for addressing these scalability challenges is model quantization [152–161], where neural networks are compressed to lower-bit representations while retaining competitive accuracy. In natural language processing (NLP) and large language models (LLMs), recent research has explored weight matrices with only a few discrete states, significantly reducing memory and computational requirements [162–164]. Similar techniques were investigated in HEP a few years ago, mainly for classification tasks [165, 166]. However, the potential of such approaches for more general HEP tasks, such as generative modeling [167], remains largely unexplored. Given these astonishing performances at reduced resource consumption in these initial studies, we expect more development towards broadly-available hardware, dedicated for low-precision computations in the near future, further motivating our study [168].

In this work, we evaluate the recently proposed BitNet architecture [162, 163] for key HEP applications, focusing on three fundamental tasks: (i) quark-gluon tagging with a Particle Dual Attention Transformer (P-DAT) [76], (ii) estimating EFT parameters through regression with SMEFTNet [117], and (iii) generative modeling for detector simulation using CaloINN [55] and CaloDREAM [56]. Our goal is to assess whether quantization-aware training (QAT) of BitNet-based models can achieve accuracy comparable to conventional high-precision networks while reducing computational overhead.

This paper is structured as follows: In Sec. 2, we introduce the BitNet architecture and outline our experimental setup. Afterwards, in Secs. 3 – 5, we present our results across the three application domains. Finally, we discuss the implications of our results for future ML-based solutions in HEP and conclude in Sec. 6.

2 The BitNet architecture

We employ BitNet, originally designed for LLMs [162, 163], to various HEP applications. In particular, our approach leverages BitLinear layers to efficiently manage memory usage and computational demands. BitNet employs either binary or ternary weights with low precision and quantized inputs during forward pass while maintaining high precision for optimizer states and gradients throughout training. This balance ensures both scalability and stability, making it well-suited for the intensive computational requirements of HEP. After training, the network weights can be saved at low precision and only a single floating-point value per layer (β, to be introduced below) needs to be stored in addition. In contrast to post-training quantization, this quantization-aware training (QAT) approach has the model in low precision already during the training process, which typically leads to better accuracy.

2.1 BitLinear layer

As illustrated in Fig. 1, both layers begin by mapping the trainable weights θ onto a quantized representation θ_q. The choice of quantization depends on the acceptable trade-off between precision and efficiency. The standard approach [162] uses binary weights, $\theta_q \in \{+1, -1\}$, while an extended version [163] allows for ternary weights, $\theta_q \in \{+1, 0, -1\}$, improving feature filtering. Both weight quantizations are parametrized as

$$\begin{aligned} \text{binary (1-bit):} \quad & \theta_q = \text{sign}\left(\theta - \langle\theta\rangle\right) , \\ \text{ternary (1.58-bit):} \quad & \theta_q = \max\left(-1, \min\left(1, \text{round}\left(\frac{\theta}{\beta}\right)\right)\right) , \qquad \text{with} \quad \beta = \langle|\theta|\rangle \, . \end{aligned} \tag{1}$$

Here, $\langle\cdot\rangle$ indicates the mean. Beyond weight quantization, both layers apply absmax quantization to the input, ensuring b-bit precision. This method scales the input within $[-Q_b, Q_b]$ ($Q_b = 2^{b-1}$) by normalizing them against the absolute maximum input value:

$$x_q = \max\left(-Q_b, \min\left(Q_b, \text{round}\left(\frac{x Q_b}{\gamma}\right)\right)\right) , \qquad \text{with} \quad \gamma = \max(|x|) \, . \tag{2}$$

Afterwards, we perform the matrix multiplication between the quantized weights and the input. The output is then rescaled and dequantized using $\{\beta, \gamma\}$ to restore its original precision, allowing the BitLinear layer to be parametrized as

$$y = \theta_q x_q \times \frac{\beta\gamma}{Q_b} \, . \tag{3}$$

The key part of Eq. (3) is the matrix multiplication of $\theta_q x_q$. Instead of expensive floating point multiplications, followed by a sum, the quantization of the weights just results in a sign for x_q, so $\theta_q x_q$ collapses to a sum without floating point multiplication.

In all our experiments, we only employ ternary (1.58-bit) quantized weights and use an 8-bit input quantization, i.e. $b = 8$ and $Q_b = 128$. Hence, for the sake of readability, we will always refer to BitLinear or BitNet in the following without explicitly mentioning 1.58-bit anymore.

2.2 Implementation and code availability

The code and data for this paper can be found on GitHub,[1] providing readers with the resources needed to reproduce our results or adapt the BitLinear layer for their own implementations.

[1] See https://github.com/ramonpeter/hep-bitnet.

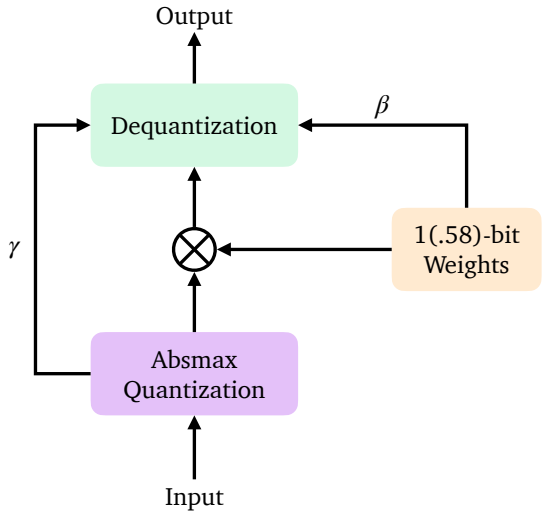

Figure 1: Illustration of the BitLinear layer.

Currently, our implementation of BitLinear follows a pseudo-quantized approach. While weights are constrained to binary or ternary values, matrix multiplications are performed in full precision (32 or 64 bits). This is because current GPU hardware does not yet support efficient computations at such low precisions, but only to 4-bit integers in some cases [169]. However, specific computation kernels for CPUs are currently in development [170, 171]. A proper study on timing and resource consumption of BitNet is therefore not part of the present work and remains an avenue for future research. Our pseudo-quantized approach is, however, sufficient for this work, as we focus on evaluating performance metrics rather than computational efficiency.

2.3 Computational resource requirements

The relative timing of different numerical operations, e.g. floating-point operations (FLOPs) or integer operations (IntOPs), is highly hardware dependent. Since we want to factorize out this component from our analysis, we report on the number of required operations instead. This will serve as a proxy for the resource complexity. The number of operations of a regular linear layer, including a bias term but excluding an activation function, is

$$N_{\text{OPs}} = N_{\text{FLOPs}} = b \times n_{\text{out}} \times 2n_{\text{in}}\,, \tag{4}$$

where b is the batch size and $n_{\text{in/out}}$ denotes the size of the input/output. In the BitLinear and BitLinear158 layer, the number of operations is

$$N_{\text{OPs}} \leq N_{\text{FLOPs}} + N_{\text{IntOPs}} + N_{\text{SignOPs}}\,, \qquad \text{with} \qquad \begin{aligned} N_{\text{FLOPs}} &= b \times (3n_{\text{out}} + n_{\text{in}} + 1)\,, \\ N_{\text{IntOPs}} &= b \times n_{\text{out}} \times (n_{\text{in}} - 1)\,, \\ N_{\text{SignOPs}} &= b \times n_{\text{out}} \times n_{\text{in}}\,. \end{aligned} \tag{5}$$

The “less than” factor arises because zero weights in the ternary quantization do not contribute to the sum, so the number of IntOPs is smaller than the sum over all elements. The three FLOPs come from the scaling with β, γ, and the bias term. The term independent of the output size comes from the initial quantization of the input vector, which is performed once for all outputs.

The timing comparisons among FLOP, SignOP, and IntOP are hardware dependent, with dedicated hardware obviously much faster than the general-purpose hardware usually available. In general, the SignOP is basically free, as it is a copy/bitflip/skip operation for weights

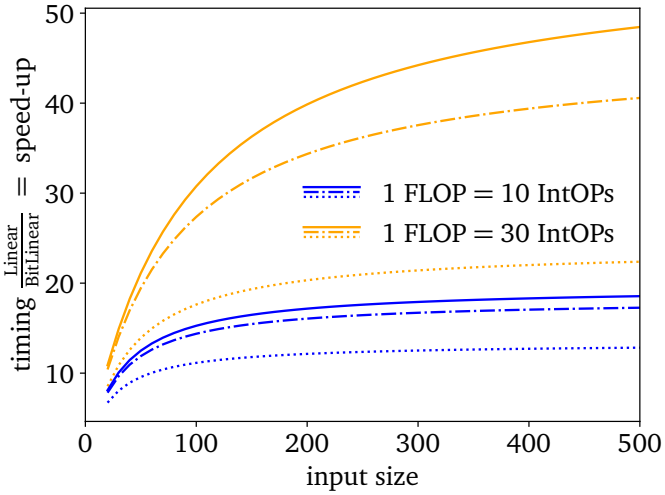

Figure 2: Relative runtime of a regular linear layer and the BitLinear layer as a function of the layer input size for two different assumptions on the relative speed of FLOPs and IntOPs and for three different output sizes: 20 (dotted), 100 (dash-dotted), and 500 (solid).

$\{1, -1, 0\}$ on any hardware. The IntOP depends on the kind of FLOP one compares against and the kind of integer used. Here, we have integer additions, which are even cheaper than integer multiplications. An estimated difference between this IntOP and a FLOP is of the order of a factor 10-30.

Figure 2 shows the ratio of the two scalings for the conservative assumption 1 FLOP = 10 IntOPs and 1 FLOP = 30 IntOPs, for different input and output sizes. In all cases, we assume vanishing costs for SignOPs and the maximal number of IntOPs (i.e. no zero weights that reduce the size of the sum).

Based on these considerations, it is clear that two networks with the same number of trainable parameters and the same fraction of quantization can still achieve very different speed-ups depending on the number of layers and their widths within the network.

3 Classification: Quark-gluon tagging

In this section, we employ BitNet for a quark-gluon discrimination task, utilizing the Particle Dual Attention Transformer (P-DAT) [76]. This architecture is specifically designed to capture both local particle-level information and global jet-level correlations.

To evaluate the performance of the default P-DAT and its quantized counterpart, P-DAT-Bit, we utilize the Quark-Gluon benchmark dataset [172], which consists of:

$$\begin{aligned} \text{Signal}: &\quad q\bar{q} \to \mathrm{Z}(\to \nu\bar{\nu}) + q_{uds}\,, \\ \text{Background}: &\quad q\bar{q} \to \mathrm{Z}(\to \nu\bar{\nu}) + \mathrm{g}\,. \end{aligned} \tag{6}$$

Jet clustering is performed using the anti-k_T algorithm with a radius parameter of $R = 0.4$ using FastJet [173]. We select only jets with transverse momentum $p_\mathrm{T} \in [500, 550]$ GeV and rapidity $|y| < 1.7$ for further analysis. The dataset contains not only the four-momenta of each particle but also particle identification labels, including electron, muon, photon, as well as individual categories for the different charged and neutral hadrons, i.e. the full PID parameterization is used.

Table 1: Performance comparison for P-DAT, P-DAT-Bit, and some existing classification algorithms on the quark-gluon discrimination dataset. The uncertainties on rejection rates are calculated by taking the standard deviation of 5 independent training runs. The error bars for the accuracy and the AUC are not displayed as they are negligible.

	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{30\%}$
ParticleNet [174]	0.840	0.9116	39.8(2)	98.6(13)
PCT [175]	0.841	0.9140	43.2(7)	118.0(22)
LorentzNet [176]	0.844	0.9156	42.4(4)	110.2(13)
ParT [177]	0.849	0.9203	47.9(5)	129.5(9)
P-DAT [76]	0.839	0.9092	39.2(6)	95.1(13)
P-DAT-Bit	0.834	0.9040	35.0(3)	83.3(12)

The dataset is divided into 1.6M training events, 200k validation events, and 200k test events. Our study focuses on the leading 100 constituents per jet, leveraging their four-momentum components and particle identification labels as input features for training. For jets containing fewer than 100 particles, we apply zero-padding to maintain uniform input dimensionality.

3.1 Particle dual attention transformer

P-DAT takes the particle information within the jet as input and consists of three main components: (i) the feature extractor, (ii) the particle attention module, and (iii) the channel attention module. In this work, we introduce a quantized variant, P-DAT-Bit, where we apply different quantization strategies to various parts of the model.

Specifically, we employ QAT for the particle and channel attention modules while keeping other components, including the feature extractor, 1D CNN, and final multi-layer perceptron (MLP) classifier, in full precision. This choice is motivated by computational efficiency and robustness considerations. The attention modules contain the majority ($\tilde{6}3\%$) of parameters and are the most computationally demanding, thus offering the greatest benefit from quantization. Furthermore, transformer-based attention mechanisms have been shown to exhibit notable robustness to quantization, making them an ideal starting point for a proof-of-concept study. Therefore, we implement QAT by replacing all linear layers with BitLinear layers in the two particle attention and two channel attention modules, affecting approximately 63% of the total weight parameters. The attention modules incorporate physics-motivated bias terms in the scaled dot-product attention, which remain in full precision to preserve critical information. The final output undergoes a 1D CNN transformation and global average pooling before being fed into an MLP classifier. Further architectural details can be found in Ref. [76]. The new model is trained from scratch, with all hyperparameters identical to the non-quantized version.

3.2 Performance comparison

As depicted in Tab. 1, while P-DAT itself offers robust performance with an accuracy of 0.839 and an AUC of 0.9092, the adaptation of this model into P-DAT-Bit reveals both the strengths and the trade-offs of utilizing BitLinear layers within the attention blocks. The integration of BitLinear layers in P-DAT-Bit results in a slight decrease in accuracy and AUC compared to the non-quantized variant. Despite this modest drop, the performance metrics remain highly com-

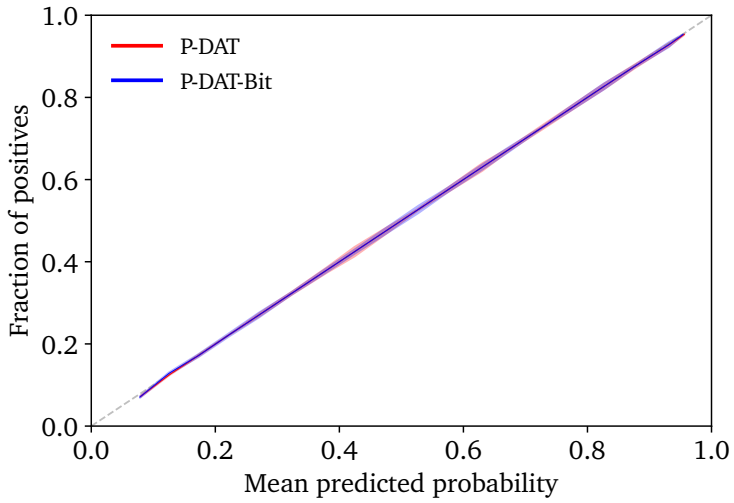

Figure 3: Calibration curves of P-DAT and P-DAT-Bit models for quark/gluon discrimination. The errorband has been obtained from evaluating 5 independent runs.

petitive, indicating that the reduced precision computation does not drastically compromise the model's discriminative capability. Specifically, the background rejection rates at 50% signal efficiency ($\text{Rej}_{50\%}$) and 30% signal efficiency ($\text{Rej}_{30\%}$) are 35.0 and 83.3, respectively. This variation between P-DAT and P-DAT-Bit highlights the balancing act between computational efficiency and performance accuracy. It emphasizes that while BitLinear layers streamline model operations,particularly advantageous in resource-constrained environments, there is a nuanced impact on the model's ability to manage complex discriminative tasks.

Finally, in Fig. 3, we present the calibration curves of P-DAT and P-DAT-Bit models for quark/gluon discrimination. The red line represents the calibration curve for the original P-DAT model, while the blue line represents the P-DAT-Bit model. Each model was evaluated over five runs to compute the average calibration curve and the associated standard deviation. Both models demonstrate good calibration performance, with their curves closely following the diagonal line, indicating that the predicted probabilities are well-calibrated with the actual positive class probabilities. Notably, the error bands for both models are very small, suggesting that the predictions of both models are consistent across multiple runs. Besides, the blue band of the P-DAT-Bit model is slightly broader compared to the red band of the P-DAT model, which can be attributed to its use of lower bit precision for weights and inputs in 60% of its parameters. Despite this, the P-DAT-Bit model is well calibrated and closely following the ideal diagonal line, indicating that its predicted probabilities are generally accurate.

To quantify the computational cost of the P-DAT-Bit, we compute the number of FLOPs, IntOPs, and SignOPs of its two building blocks, the Particle Attention Block and the Channel Attention Block in Tab. 2.

4 Regression: SMEFT parameter estimation

In this section, we evaluate the performance of BitNet for a regression task and focus on the SMEFTNet [117] architecture. We denote its quantized version as SMEFTNet-Bit, where either all or some linear layers are replaced with its BitLinear counterpart. To perform a meaningful comparison, we employ the same simulated WZ event samples used in Ref. [117], focusing on

Table 2: Computational cost for a single pass of the full-precision and BitNet implementations of the Particle Attention and Channel Attention Blocks in the P-DAT architecture. Percentages are in relation to the number of FLOPs of P-DAT. Assumptions: 1 FLOP = 10 IntOPs, SignOP = 0 FLOPs.

Block	Operations	P-DAT	P-DAT-Bit
Particle Attention Block	FLOPs	7 795 200	2 995 590
	SignOPs	-	2 457 600
	IntOPs	-	2 419 200
	Total FLOPs	7 795 200	3 237 510 (41%)
Channel Attention Block	FLOPs	5 896 192	1 896 517
	SignOPs	-	2 048 000
	IntOPs	-	1 209 600
	Total FLOPs	5 896 192	2 017 477 (34%)

a semi-leptonic decay chain

$$\mathrm{pp} \to \mathrm{W}(\to q\bar{q})\,\mathrm{Z}(\to \ell\bar{\ell})\,. \tag{7}$$

The goal is to predict the decay plane angle $\phi^{\text{jet}}_{\text{decay}}$ of the parton-level quarks from the particle-level jet information, as this observable is sensitive to linear SMEFT–SM interference effects. This decay plane angle depends on the exact momenta of the up-type and down-type quarks, and interchanging these momenta at the parton level results in a difference of π. However, the particle-level decay products do not contain this information and are thus invariant under this permutation. Hence, in order to capture this ambiguity in the training of the network $f_\theta(x)$ with trainable parameters θ, we use the modified loss function [117]

$$\mathcal{L} = \left\langle \sin^2\left(f_\theta(x_i) - \phi^{\text{jet}}_{i,\text{decay}}\right)\right\rangle_{(x_j,\varphi_j)\sim\mathcal{D}_{\text{train}}}, \tag{8}$$

where the sin encodes the symmetry of shifting with π. The input x consists of the lab-frame momenta of each particle in a given event j parametrized as

$$x = \{p_{\mathrm{T},1}, \varphi_1, \Delta R_1, \cdots, p_{\mathrm{T},N_j}, \varphi_{N_j}, \Delta R_{N_j}\}\,, \tag{9}$$

where N_j denotes the number of particles in this event.

4.1 SMEFTNet architecture

SMEFTNet is an IRC-safe and rotation-equivariant graph neural network. It is designed to provide an optimal observable for small deviations from the SM and enhance SMEFT sensitivity, specifically focusing on the linear SM-SMEFT interference within SMEFT. To preserve sensitivity to the linear term, it is crucial to consider the orientation of the decay planes of the W or Z boson, as this orientation helps resolve the helicity configuration of the amplitude that is altered in the SMEFT [178]. To study the hadronic final states of W or Z boson, SMEFTNet is constructed to be equivariant to azimuthal rotations of the constituents of the boosted jet around the jet axis, maintaining SO(2) symmetry regardless of the chosen reference frame. It processes inputs as variable-length lists of particle constituents of a fat jet, originating from the hadronic decay of a boosted massive particle. For details on the architecture, we refer to Ref. [117]. Although the decay-plane angle regression is a relatively lightweight task, investigating quantization in this controlled setting is still meaningful. In realistic SMEFT inference

tasks, surrogate models have to be evaluated repeatedly when profiling over many nuisance parameters to estimate systematic uncertainties, or when integrating over latent observables, where lighter and faster models become highly advantageous, and even moderate speedups can lead to substantial reductions in end-to-end inference time. Thus, assessing whether low-precision variants of lightweight models retain accuracy is a necessary first step before applying such techniques in more computationally intensive inference workflows.

In our study, we consider three quantized variants of the SMEFTNet model and apply the three variants to predict the decay plane angle from the jet's constituents and compare their performance with the results presented in Ref [117]. In the first variant, all linear layers are replaced with BitLinear layers, which we refer to as SMEFTNet-Bit100. In the second variant, only the linear layers within the MLP block are quantized, corresponding to approximately 70% of the total weight parameters; this configuration is labeled SMEFTNet-Bit70. In the third variant, only the linear layers in the Message Passing Neural Network block are quantized, accounting for about 30% of the weights; this is denoted SMEFTNet-Bit30.

4.2 Performance comparison

To visualize the comparative analysis of SMEFTNet and SMEFTNet-Bit100, Fig. 4 presents the scatter plots of the true and the predicted φ_{decay} in the remaining 20% WZ dataset, respectively. The results reveal that while the SMEFTNet-Bit100 model marginally underperforms relative to the original SMEFTNet, the differences are minimal, showcasing the effectiveness of the low-bit model in capturing the essential structure of the data despite its limited weight and input precision. Notably, the scatter plot of SMEFTNet exhibits two faint vertical red bands at

Table 3: Comparison of the Wasserstein distance and separation power between the residual distributions of each of the three quantized SMEFTNet variants, an independently re-trained full-precision SMEFTNet, and that of the original SMEFTNet.

Model Comparison	Wasserstein Distance	Separation Power
SMEFTNet	0.0021	0.0001
SMEFTNet-Bit100	0.4546	1.4860
SMEFTNet-Bit70	0.2528	0.6138
SMEFTNet-Bit30	0.1040	0.1503

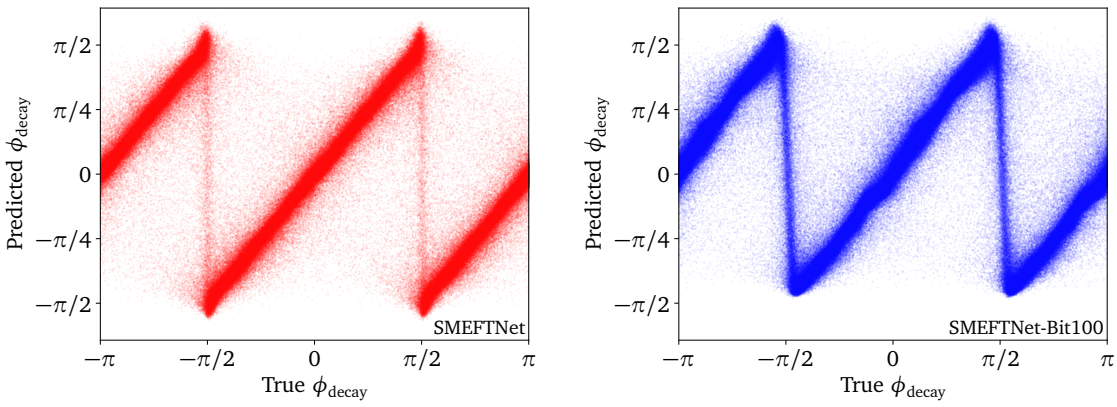

Figure 4: Two-dimensional scatter plots of the true and predicted angle ϕ_{decay} in the test dataset. Left: SMEFTNet with regular linear layers. Right: SMEFTNet-Bit100 with BitLinear layers.

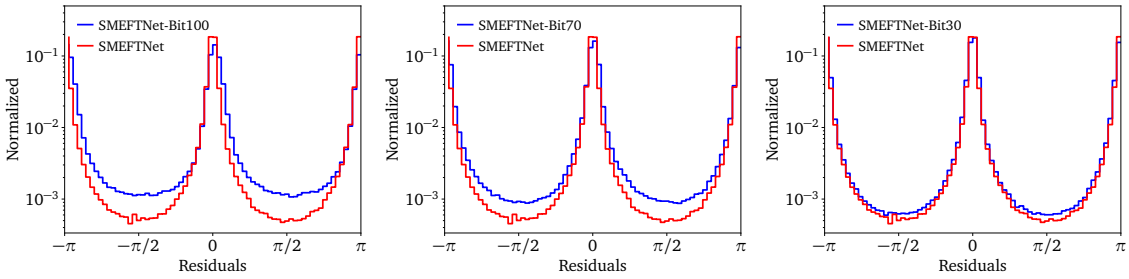

Figure 5: Histograms of residuals, defined as the differences between truths and predictions for ϕ_{decay} regression task. The histogram in blue represents SMEFTNet without BitLinear layers, whereas the histogram in red corresponds to SMEFTNet with BitLinear layers. From left to right: 100%, 70%, and 30% of the weights are quantized.

true ϕ_{decay} values of $\pm\pi/2$, which become more pronounced and broader in the SMEFTNet-Bit100 scatter plot. This phenomenon is attributed to both models' tendency to stabilize at local minima due to the inherent challenges posed by multi-objective optimization. In the critical region of true $\phi_{\text{decay}} \approx \pm\pi/2$, two physically indistinguishable parton-level configurations — arising from the permutation of up-type and down-type quarks — correspond to decay plane angles that differ by π. Since the particle-level information does not resolve this ambiguity, the regression model is trained with both values as plausible targets. As a result, it is forced to compromise between conflicting objectives, potentially leading to suboptimal predictions for either configuration individually. The broader bands in the SMEFTNet-Bit100 results are explained by its limited precision due to weight and input quantization. This restricted precision leads to significant fluctuations in the predicted ϕ_{decay} values, resulting in the observed wider bands in the visual representations.

Finally, Fig. 5 presents three histograms of residuals, which represent the differences between truths and predictions for $\phi_{\text{j, decay}}$. In each histogram, the red bars correspond to the original SMEFTNet, while the blue bars represent different configurations of the SMEFTNet-Bit model. In the left histogram, the blue bars correspond to SMEFTNet-Bit100. In the middle histogram, the blue bars represent the SMEFTNet-Bit70. In the right histogram, the blue bars represent the SMEFTNet-Bit30. In all cases, the SMEFTNet model exhibits a sharp peak at zero, indicating a precise alignment of predictions with true values. When fully quantized (100%), the SMEFTNet-Bit100 distribution broadens prominently, reflecting larger variations in prediction accuracy. Reducing the fraction of quantized layers to 70% yields intermediate performance: the residuals of SMEFTNet-Bit70 remain somewhat more dispersed than SMEFTNet but are considerably narrower than the fully quantized version. At 30% quantization, the residuals of SMEFTNet-Bit30 closely resemble those of the original SMEFTNet, showing only a modest increase in width. Consequently, it is evident that as more linear layers are replaced by BitNet layers, the performance of the model deteriorates. Moreover, a notable feature in all plots is the periodic structure of the residuals, which arises because the model faces a periodic ambiguity in which angles separated by π map to the same predicted value, creating these peaks around $\pm\pi$ and 0 in the residual distribution. Furthermore, the fraction at $\pm\frac{\pi}{2}$ is higher for SMEFTNet-Bit100, reflecting its broader vertical band in its 2D scatter plots. This broadening stems from the limited precision of BitLinear layers, which amplifies prediction fluctuations near critical values of $\pm\pi$. Notably, as the proportion of quantized layers increases, the fraction at $\pm\frac{\pi}{2}$ also becomes higher, indicating that stronger quantization further destabilizes the predictions near critical values of $\pm\pi$. By contrast, SMEFTNet shows narrower bands, indicating more accurate predictions and a lower fraction at $\pm\frac{\pi}{2}$. Lastly, across all three configurations,

Table 4: Computational cost for a single pass of the Full-Precision and BitNet implementations of the Message Passing and MLP blocks in the SMEFTNet architecture. Percentages are in relation to the number of FLOPs of the SMEFTNet. Assumptions: 1 FLOP = 10 IntOPs, SignOP = 0 FLOPs.

Block	Operations	SMEFTNet	SMEFTNet-Bit100
Message Passing Block	FLOPs	1080	188
	SignOPs	-	520
	IntOPs	-	479
	Total FLOPs	1080	288 (27%)
MLP Block	FLOPs	3776	411
	SignOPs	-	1824
	IntOPs	-	1759
	Total FLOPs	3776	769 (20%)

both models overlap around zero residuals, highlighting their overall reliability. However, the progressive widening of SMEFTNet-Bit's distributions with increasing quantization underscores the trade-off between model compression and predictive precision. This demonstrates that partial, rather than complete, quantization can give a better balance for resource-limited applications.

Table 3 complements Fig. 5 with two quantitative metrics. The Wasserstein distance captures the average angular shift between residual distributions, whereas the separation power reflects their overall shape divergence. The additional "SMEFTNet" entry corresponds to a second model trained with the same architecture and hyper-parameters as the reference model but initialized with a different random seed. Its near-zero Wasserstein distance (0.002) and separation power (1×10^{-4}) serve as a self-consistency check, confirming that both metrics are sensitive only to distributional shifts and robust to statistical fluctuations. By contrast, the three quantized variants show progressively larger values as the proportion of BitLinear layers increases, quantitatively substantiating the broadening already visible in Fig. 5.

To estimate the resource consumption of the SMEFTNet-Bit100, we compute the number of FLOPs, IntOPs, and SignOPs of its two building blocks, the Message Passing Block and the MLP Block in Tab. 4. Since the Message Passing Block is applied once per edge while the MLP Block is applied once per event, we report their per-pass costs separately.

5 Generative: Detector simulation

Next, we consider a generative task. These are very important in HEP, as the process of sampling random events from a given (complicated) conditional probability density is precisely what happens in numerical simulations of particle collisions (See [1] for an overview on ML applications to the entire simulation chain.). One notable challenge is detector simulations, which involve modeling showers of energetic particles in the calorimeters, a task characterized by its high numerical complexity and dimensionality. In the past years, this field has seen many new ideas and approaches [179], which led to the conception of the CaloChallenge [180] to compare existing models on equal footing and spur even more development of state-of-the-art methods. The CaloChallenge was a data challenge in the HEP community, with four different datasets, increasing in their dimensionality. The goal of the challenge was to train generative networks on the datasets and to generate artificial samples as fast and precise as possible [181]. The dataset dimensionalities range from a few hundred in the easiest

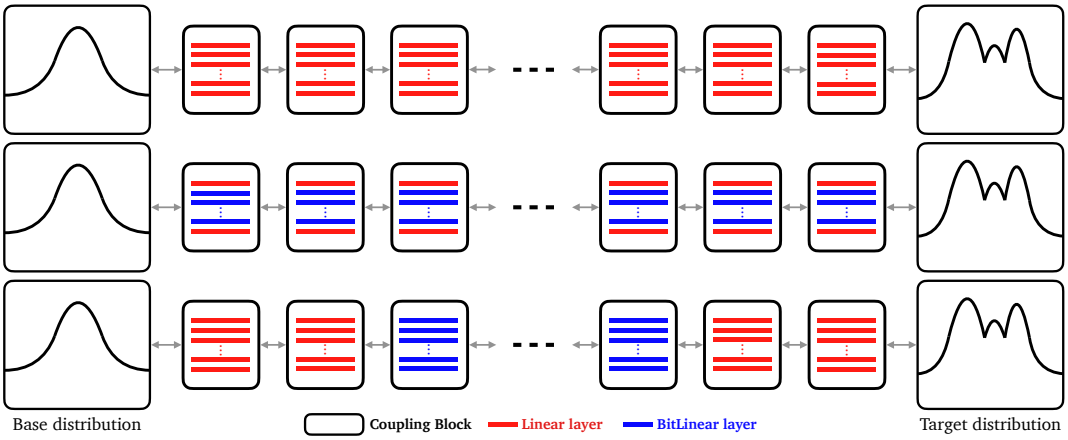

Figure 6: Illustration of different levels of quantization for the flow. Red lines represent regular linear layers, while blue lines indicate BitLinear layers. The three setups, from top to bottom, are: *regular*, *NNCentral*, and *BlockCentral*.

case to a few tens of thousands in the most complicated case. These are single particle showers, simulated with GEANT4 [182–184] and available at [185–188]. For more details on the datasets we refer to [180].

When evaluating generative models, one can pick from various metrics [180, 189, 190]. Here, we use classifiers to evaluate the generative performance. The classifiers are trained on the task of distinguishing generated samples from the GEANT4 reference. If a powerful, well-trained classifier cannot distinguish between the two samples, we conclude that the samples were drawn from the same underlying distribution, i.e. the generative model learned the underlying distribution well [39]. Summarized in a single quantity, we use the area under the receiver operating characteristic (AUC). A lower AUC indicates a better generative model.

As an example for generative networks, we consider two well-performing submissions of the CaloChallenge [180]: CALOINN [55] based on a normalizing flow and CALODREAM [56] based on conditional flow matching. Normalizing flows were the first generative model that passed the "classifier test" [39, 190] in calorimeter shower simulations and have seen various applications to this task [39, 41, 46, 48, 55, 57, 180, 191–193]. Conditional flow matching is the most recent generative model that was explored in this context and shows impressive performance in many applications [56, 194, 195].

5.1 Normalizing flow – CALOINN

CALOINN [55] is a normalizing-flow-based generative model, learning a bijective transformation between a simple base distribution, usually a Gaussian, and a more complicated distribution, the target distribution. In detail, CALOINN is based on coupling layer-based normalizing flow [196, 197] (which are sometimes also called INN, invertible neural networks [198], even though they all are invertible). These types of flows are equally fast to evaluate in both directions (as density estimator and generative model), making training and generation more efficient than in an autoregressive setup [39, 41, 46]. The bijective transformation that is used in the coupling blocks is based on splines [199–201].

The calorimeter showers are normalized to unit energy in each calorimeter layer, and the corresponding layer energies are then appended to this array and encoded via ratios to the incident energy. This setup allows CALOINN to learn the distribution of calorimeter showers in a single step instead of a more time-consuming two-step procedure [39, 41, 46]. The downside of this approach is the scaling with the dimensionality of the dataset. The spline-based trans-

formation requires a large number of output parameters for the individual NNs in the coupling layers, which in turn increases the number of trainable parameters of the NN. For dataset 2, CALOINN already has 270M parameters, making an application to the six times bigger dataset 3 of the CaloChallenge impossible.

All hyperparameters are as in the original publication [55], with the flow consisting of 12 (14) coupling blocks and the individual NNs having 256 hidden nodes, and 4 (3) hidden layers for dataset 1 (2). For dataset 1, CALOINN uses rational quadratic splines [200], for dataset 2 it uses cubic splines [201].[2] For quantization, we consider 5 different setups, or quantization strategies, which we can summarize as follows:

Default: The setup used in the original publication [55], without quantization. It serves as a baseline as well as a confirmation that we use the same hyperparameters as Ref. [55].

Exchange Permutation: A setup where we change the permutations between the bijector blocks. Instead of using random permutations throughout, we exchange the sets that are transformed and that go into the NN after the first and before the last bijector. This strategy ensures that every dimension is transformed exactly once by both the first and last two bijectors. There is no quantization in this setup. It serves as a baseline for the *BlockCentral* setup, as it shares the same permutation scheme.

NNCentral: Only the central layers in each NN are quantized. Since in spline-based flows the final layer typically has the most parameters, this configuration has minimal effects.

BlockCentral: Quantization is applied to all layers of NNs in the *central* bijectors, i.e. neither the first two nor the last two in the chain. Paired with the same permutation strategy as in *Exchange Permutation*, this ensures each dimension is first transformed by a regular bijector, then by quantized ones, and finally again by a regular bijector.

All: All linear layers in all bijectors are quantized.

Fig. 6 shows an illustration of the *regular*, *NNCentral* and *BlockCentral*. Their fractions of quantized weights are given in Tab. 5. To evaluate the performance of the quantized CALOINN, we use the exactly same classifier architecture and training strategy as in [180], making the results directly comparable. The main results are shown in Tab. 5.

The AUCs we see in the regular setup are consistent with the ones reported in [180], where ds1 photon was scored 0.626(4) / 0.638(3), ds1 pion was scored 0.784(2) / 0.732(2), and ds2 was scored 0.743(2) / 0.865(3) for low / high-level observables, respectively. These AUCs differ from the ones presented in [55] as the classifier architecture in the original publication was different than the one used here and in [180]. The modified permutations usually improve the AUCs slightly, but mostly they agree with the regular setup within one standard deviation.

We observe that quantizing the NN weights with the BITNET degrades performance, with a clear correlation between the fraction of quantization and the AUC. Across all datasets, the *NNCentral* setup gives AUCs almost as good as the *Regular* setup, but in this case only a very low fraction of weights is actually quantized. The *All* setup degrades the generative performance a lot, with the gap being larger in the high-dimensional dataset 2. The *BlockCentral* setup results in an overall good performance. This is an interplay of not having quantized all of the parameters and the choice to only quantize the central coupling layers instead of the outer ones. This latter choice allows for the bulk of the bijection being carried out by a quantized flow, while the final details will be adjusted in the outer coupling blocks with a regular flow setup.

[2] This choice was motivated in [55] through an improved stability. Indeed, we also observe a few cases of NaNs being sampled: about 40 cases in the *NNCentral* setup, 1 case in the *BlockCentral*, and 4000 cases in the *All* scenario.

To estimate the resource consumption of the quantized CaloINN, we compute the number of FLOPs, IntOPs, and SignOPs of the three different types of coupling layers that make up our quantization strategies in Tab 6. While these number take into account the batchnormalizations in the NN, they do not include the FLOPs that are used in the spline evaluation.

Table 5: Performance of default and BitNet CaloINN using the classifier metric of the CaloChallenge [180]. Uncertainties show the standard deviation over 10 random initializations and trainings of the classifier on the same CaloINN sample.

Dataset		Setup	Quantization	Low-Level AUC	High-Level AUC
ds1–γ	reg.	Default	-	0.633(3)	0.656(3)
		Exchange Perm.	-	0.640(4)	0.651(3)
	quant.	NNCentral	8.4%	0.640(3)	0.650(2)
		BlockCentral	66.6%	0.680(3)	0.669(3)
		All	99.9%	0.759(2)	0.828(2)
ds1–π^+	reg.	Default	-	0.793(3)	0.742(3)
		Exchange Perm.	-	0.784(2)	0.736(3)
	quant.	NNCentral	5.9%	0.801(2)	0.751(3)
		BlockCentral	66.6%	0.852(1)	0.807(2)
		All	99.9%	0.882(2)	0.907(2)
ds2	reg.	Default	-	0.738(4)	0.859(2)
		Exchange Perm.	-	0.728(6)	0.857(3)
	quant.	NNCentral	0.3%	0.780(3)	0.876(4)
		BlockCentral	71.4%	0.950(2)	0.979(1)
		All	99.9%	0.993(1)	0.998(0)

Table 6: Computational cost for a single pass through one of the Coupling Blocks (CB) of the CaloINN. Percentages are in relation to the number of FLOPs of the regular CB. Assumptions: 1 FLOP = 10 IntOPs, SignOP = 0 FLOPs.

Dataset	Operations	Default	NNCentral	All
ds1–γ	FLOPs	3 121 664	2 861 570	20 982
	SignOPs	-	131 072	1 560 064
	IntOPs	-	130 560	1 553 902
	Total FLOPs	3 121 664	2 874 626 (92%)	176 372 (6%)
ds1–π^+	FLOPs	4 411 392	4 151 298	28 373
	SignOPs	-	131 072	2 204 928
	IntOPs	-	130 560	2 196 330
	Total FLOPs	4 411 392	4 164 354 (94%)	248 006 (6%)
ds2	FLOPs	38 545 408	38 415 361	220 607
	SignOPs	-	65 536	19 272 704
	IntOPs	-	65 280	19 200 428
	Total FLOPs	38 545 408	38 421 889 (99.7%)	2 140 650 (5.6%)

5.2 Conditional flow matching – CALODREAM

CALODREAM [56] is a generative architecture combining Conditional Flow Matching (CFM) with transformer elements.

In detail, CALODREAM consists of two networks, both trained with conditional flow matching: (i) an autoregressive transformer, called the *energy network*, to learn the 45 layer energies, and (ii) a vision transformer, called the *shape network*, learning the normalized showers. The self-attention mechanism in the transformer layers is highly beneficial for modeling the sparsity of calorimeter showers and the correlations across detector locations. Crucially, the introduction of *patching*, which groups nearby voxels into coarser units, reduces the impact of the quadratic scaling and enables the model to scale efficiently to larger calorimeter geometries such as DS3. At the same time, conditional flow matching ensures efficient training of the underlying continuous normalizing flow model.

The main difference of CALODREAM to CALOINN when considering the quantization lies in the distribution of NNs in the generative process: instead of many "smaller" NNs in CALOINN, CALODREAM has one large NN for each of the two steps, so we expect the impact of quantization on performance to be smaller.

The preprocessing of the calorimeter shower data is done as in the CALOINN setup. Showers are normalized per calorimeter layer, and the corresponding layer energies are encoded as ratios to constrain them to the range $[0, 1]$. The energy network now learns the latter independently from the normalized showers, which are learned by the shape network.

Since the two networks factorize, we can discuss quantization strategies separately and combine them ad libitum in generation.

Quantization of the energy network

In the energy network, an autoregressive transformer defines the embeddings of the energy values that are then passed to a MLP performing the CFM. We keep the original hyperparameter choices for these networks, so the former is based on 4 attention heads with an embedding dimension of 64, see [56]. The feed-forward CFM network consists of 8 hidden layers with 256 nodes each. It takes the concatenated 64-dimensional time embedding, the 64-dimensional autoregressive embedding of previous calorimeter layer energies as well as the 1-dimensional current calorimeter layer energy as input and predicts the velocity field of the current calorimeter layer as output. We consider two quantization setups for the energy network:

Regular: The non-quantized version, as used in the original publication [56].

Quantized: Quantizes the central 6 of the 8 hidden layers of the CFM network, leaving the transformer-based embedding networks untouched. This results in 66.09% of the trainable parameters of the energy network being quantized. However, since the energy network is much smaller than the shape network, this translates to only 5.54% of the total CALODREAM model being quantized.

Quantization of the shape network

In the shape network, all 6480 voxels of the calorimeter are split into 135 patches of 48 voxels each, and all operations are performed on these patches in parallel. Diffusion time t, ratios of layer energies, incident energy, and voxel patches are transformed using three different embedding networks: (i) for time, (ii) for conditionals, and (iii) for position. These embeddings are passed to a chain of 6 Vision Transformer (ViT) blocks, each consisting of a self-attention layer, a projection, and a subsequent MLP step. The output is then passed through a final MLP layer and reassembled from patches into complete calorimeter showers.

Table 7: Performance of regular and BITNET CALODREAM using the classifier metric of the CaloChallenge [180]. Uncertainties show the standard deviation over 10 random initializations and trainings of the classifier on the same CALODREAM sample.

Energy net	Shape net	Quantization	low-level AUC	high-level AUC
regular	regular	-	0.531(3)	0.523(3)
quantized		5.5%	0.532(3)	0.525(3)
regular	no embedding	58.5%	0.611(2)	0.543(3)
quantized		63.9%	0.610(5)	0.545(2)
regular	full	60.7%	0.735(4)	0.942(2)
quantized		66.2%	0.738(4)	0.944(3)

Given the modular nature of the shape network, several quantization strategies are possible. Here, we report three:

Regular: The non-quantized version, identical to the setup in the original publication [56].

No embedding: Quantizes the core elements of the ViT blocks, i.e. QKV matrices, projections, and MLPs, while keeping the embedding networks and the final MLP layer unquantized. This configuration results in 63.8% of the shape network being quantized.

Full: Extends the *no embedding* setup by also quantizing the linear layers in the position, time, and conditional embedding networks. This increases the fraction of quantized parameters in the shape network to 66.22%.

Performance comparison

Also for the evaluation of CALODREAM, we use the same classifier architecture and training strategy as in the CaloChallenge [180], making the results directly comparable. The main results are shown in Tab. 7.

Our retraining of CALODREAM in the regular setup reproduces the scores of the CaloChallenge [180] with 0.531(3) / 0.521(2) for low/high-level AUC.

The first thing we observe in Tab. 7 is that swapping the regular for the quantized energy network has no effect on the resulting AUC scores, so 66% of trainable parameters in six out of eight hidden layers inside the CFM can safely be quantized without loss of sample quality.

In the shape network, however, the performance strongly depends on which parts are quantized. A substantial part of the ViT blocks can be quantized with almost no loss in shower quality, but as soon as the embedding layers are quantized, the performance drops a lot. Note that the *no embedding* CALODREAM, which is quantized to about 60% still has the best high-level, and second best low-level AUCs of the CaloChallenge submissions [180].

We also observe the spread between high- and low-level AUC to be larger. Especially in the *fully* quantized case, the high-level features were already very sensitive. The low-level AUC score was not as bad, but that is likely a remnant of the classifier architecture choice, see the discussion in [180].

In tables 8 and 9 we give an estimate of the computational resources required for different components of CALODREAM. The numbers are based on a single run through the networks, so they only show a fraction of the FLOPs required to generate a full shower. In addition, we also do not include the transformer decoder and encoder networks of the energy network, as they are not quantized in our studies.

Table 8: Computational cost for a single pass through the energy network. Percentages are in relation to the number of FLOPs of the regular network. Assumptions: 1 FLOP = 10 IntOPs, SignOP = 0 FLOPs.

Block	Operations	Regular	Quantized
Energy Network	FLOPs	3 297 292	163 858
	SignOPs	-	1 572 864
	IntOPs	-	1 569 792
	Total FLOPs	3 297 292	320 837 (10%)

Table 9: Computational cost for a single pass through the shape network. Percentages are in relation to the number of FLOPs of the regular network. Assumptions: 1 FLOP = 10 IntOPs, SignOP = 0 FLOPs.

Block	Operations	Regular	No embedding	Full
Embedding Layers	FLOPs	1 257 600	1 257 600	8515
	SignOPs	-	-	628 800
	IntOPs	-	-	626 400
	Total FLOPs	1 257 600	1 257 600 (100%)	71 155 (6%)
Diffusion Transformer Blocks	FLOPs	50 771 546	17 691 890	17 691 890
	SignOPs	-	16 588 800	16 588 800
	IntOPs	-	16 562 880	16 562 880
	Total FLOPs	50 771 546	19 348 178 (38%)	19 348 178 (38%)

6 Conclusions and outlook

In this paper, we investigated the applicability of the BitNet architecture to various tasks in high-energy physics. We demonstrated that QAT can offer a promising path forward for large-scale neural network applications in HEP. By applying these techniques to classification, regression, and generative tasks, we observed that performance remains competitive, especially for classification tasks such as quark–gluon tagging. However, the impact of quantization on regression and generative modeling is more nuanced and requires careful consideration of network size and architecture. The results highlight that:

- **Larger networks can be quantized more easily.** Our experiments with different generative architectures used in detector simulation, specifically the normalizing flow-based CaloINN and the flow matching-based CaloDREAM, indicate that larger models often exhibit smoother quantization behavior. For instance, with around 66% of its parameters quantized, CaloINN experiences a noticeable drop in sample quality, whereas CaloDREAM, which also quantizes roughly 63.8% of its shape network, exhibits only minor performance degradation. These observations underscore how larger networks provide greater representational capacity, making them more resilient to the information loss introduced by low-bit representations. Consequently, these results highlight the potential of quantized, large-scale generative models in tackling the complex, high-dimensional tasks typical of modern particle physics experiments.

- **Performance depends on the layer chosen for quantization.** In our generative studies for detector simulation, CaloINN benefits from selectively quantizing all layers within the central bijectors (BlockCentral), thereby maintaining decent sample quality even at

about 66% quantized weights, while fully quantizing all layers (99.9%) severely degrades performance. Likewise, in CALODREAM, quantizing the elements within the shape network's ViT blocks (QKV matrices, projections, and MLPs) results in minimal performance loss; however, once the embedding layers are quantized, performance drops considerably. These findings underscore how carefully selecting which sections of the network to quantize – central layers versus outer layers, embedding networks versus projection layers – can preserve model fidelity and better adapt to each architecture's demands, especially for high-dimensional tasks in particle physics. Further studies in automatic, heterogeneous quantization [202,203] are needed to fully extract the best performance of the networks.

- **Fully quantized models show the largest performance degradation.** Our SMEFTNet experiments demonstrate that the fully quantized variant performs worst among all tested configurations. Specifically, SMEFTNet-Bit100 exhibits a larger degradation in accuracy than the configurations where only a subset of layers is quantized (SMEFTNet-Bit30 or SMEFTNet-Bit70). These results highlight the trade-off between model compression and accuracy, emphasizing the need for selective quantization to balance efficiency and precision.

- **Certain layers are robust to quantization.** Self-attention layers in transformer-based models, such as P-DAT or the shape network ViT in CALODREAM, exhibit surprisingly minimal performance degradation thanks to their ability to encode attention patterns effectively with discrete weights. This finding points to the potential of applying low-bit quantization in attention-driven architectures, enabling significant memory and computational savings.

- **Low-bit quantization aligns with future hardware and energy constraints.** As the HL-LHC generates increasingly large datasets which require more sophisticated analyses and simulations, achieving high performance while keeping the energy demands reasonable will be crucial. The low-bit QAT explored here aligns well with emerging hardware trends, where dedicated architectures for quantized operations are expected to play a central role.

The relevance of such quantization techniques is expected to grow as networks become larger and more prevalent throughout HEP workflows. Future research directions include exploring alternative weight quantization configurations, integrating these methods seamlessly into existing large-scale ML models, and adapting BITNET for high-energy physics applications through fully quantized implementations with low-precision operations, enabling a quantitative evaluation of energy, memory, and latency reductions. Moreover, extending the scope to tasks requiring real-time or near-real-time performance will further validate the robustness of quantized models under realistic experimental conditions. A typical example is the LHC trigger system, which demands extremely fast classification on resource-limited hardware such as FPGAs. While quantization techniques have been successfully applied in this context for many years, our study highlights the potential of QAT to push these approaches further, beyond the trigger. By incorporating QAT, it may become feasible to implement larger and more expressive models on FPGAs or other hardware, potentially leading to more energy-efficient, scalable, yet accurate ML applications in HEP.

Acknowledgments

We would like to thank Luigi Favaro, Ayodele Ore, and Sofia Palacios Schweitzer for assistance in setting up CALOINN and CALODREAM as well as Suman Chatterjee and Robert Schöfbeck for help with SMEFTNet. We would also like to thank Thea Årrestad for fruitful discussions on weight quantization.

Funding information RW is supported through funding from the European Union NextGeneration EU program – NRP Mission 4 Component 2 Investment 1.1 – MUR PRIN 2022 – CUP G53D23001100006. The computational results presented were obtained using the CLIP cluster (https://clip.science).

References

[1] A. Butter et al., *Machine learning and LHC event generation*, SciPost Phys. **14**, 079 (2023), doi:10.21468/SciPostPhys.14.4.079 [preprint doi:10.48550/arXiv.2203.07460].

[2] T. Plehn, A. Butter, B. Dillon, T. Heimel, C. Krause and R. Winterhalder, *Modern machine learning for LHC physicists*, (arXiv preprint) doi:10.48550/arXiv.2211.01421.

[3] F. Bishara and M. Montull, *Machine learning amplitudes for faster event generation*, Phys. Rev. D **107**, L071901 (2023), doi:10.1103/PhysRevD.107.L071901 [preprint doi:10.48550/arXiv.1912.11055].

[4] S. Badger and J. Bullock, *Using neural networks for efficient evaluation of high multiplicity scattering amplitudes*, J. High Energy Phys. **06**, 114 (2020), doi:10.1007/JHEP06(2020)114 [preprint doi:10.48550/arXiv.2002.07516].

[5] J. Aylett-Bullock, S. Badger and R. Moodie, *Optimising simulations for diphoton production at hadron colliders using amplitude neural networks*, J. High Energy Phys. **08**, 066 (2021), doi:10.1007/JHEP08(2021)066 [preprint doi:10.48550/arXiv.2106.09474].

[6] D. Maître and H. Truong, *A factorisation-aware matrix element emulator*, J. High Energy Phys. **11**, 066 (2021), doi:10.1007/JHEP11(2021)066 [preprint doi:10.48550/arXiv.2107.06625].

[7] R. Winterhalder, V. Magerya, E. Villa, S. Jones, M. Kerner, A. Butter, G. Heinrich and T. Plehn, *Targeting multi-loop integrals with neural networks*, SciPost Phys. **12**, 129 (2022), doi:10.21468/SciPostPhys.12.4.129 [preprint doi:10.48550/arXiv.2112.09145].

[8] S. Badger, A. Butter, M. Luchmann, S. Pitz and T. Plehn, *Loop amplitudes from precision networks*, SciPost Phys. Core **6**, 034 (2023), doi:10.21468/SciPostPhysCore.6.2.034 [preprint doi:10.48550/arXiv.2206.14831].

[9] D. Maître and H. Truong, *One-loop matrix element emulation with factorisation awareness*, J. High Energy Phys. **05**, 159 (2023), doi:10.1007/JHEP05(2023)159 [preprint doi:10.48550/arXiv.2302.04005].

[10] H. Bahl, N. Elmer, L. Favaro, M. Haussmann, T. Plehn and R. Winterhalder, *Accurate surrogate amplitudes with calibrated uncertainties*, SciPost Phys. Core **8**, 073 (2025), doi:10.21468/SciPostPhysCore.8.4.073 [preprint doi:10.48550/arXiv.2412.12069].

[11] J. Bendavid, *Efficient Monte Carlo integration using boosted decision trees and generative deep neural networks*, (arXiv preprint) doi:10.48550/arXiv.1707.00028.

[12] M. Klimek and M. Perelstein, *Neural network-based approach to phase space integration*, SciPost Phys. **9**, 053 (2020), doi:10.21468/SciPostPhys.9.4.053 [preprint doi:10.48550/arXiv.1810.11509].

[13] I.-K. Chen, M. Klimek and M. Perelstein, *Improved neural network Monte Carlo simulation*, SciPost Phys. **10**, 023 (2021), doi:10.21468/SciPostPhys.10.1.023 [preprint doi:10.48550/arXiv.2009.07819].

[14] C. Gao, J. Isaacson and C. Krause, *i-flow: High-dimensional integration and sampling with normalizing flows*, Mach. Learn.: Sci. Technol. **1**, 045023 (2020), doi:10.1088/2632-2153/abab62 [preprint doi:10.48550/arXiv.2001.05486].

[15] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale and S. Schumann, *Exploring phase space with neural importance sampling*, SciPost Phys. **8**, 069 (2020), doi:10.21468/SciPostPhys.8.4.069 [preprint doi:10.48550/arXiv.2001.05478].

[16] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Event generation with normalizing flows*, Phys. Rev. D **101**, 076002 (2020), doi:10.1103/PhysRevD.101.076002 [preprint doi:10.48550/arXiv.2001.10028].

[17] K. Danziger, T. Janßen, S. Schumann and F. Siegert, *Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates*, SciPost Phys. **12**, 164 (2022), doi:10.21468/SciPostPhys.12.5.164 [preprint doi:10.48550/arXiv.2109.11964].

[18] T. Heimel, R. Winterhalder, A. Butter, J. Isaacson, C. Krause, F. Maltoni, O. Mattelaer and T. Plehn, *MadNIS - Neural multi-channel importance sampling*, SciPost Phys. **15**, 141 (2023), doi:10.21468/SciPostPhys.15.4.141 [preprint doi:10.48550/arXiv.2212.06172].

[19] T. Janßen, D. Maître, S. Schumann, F. Siegert and H. Truong, *Unweighting multijet event generation using factorisation-aware neural networks*, SciPost Phys. **15**, 107 (2023), doi:10.21468/SciPostPhys.15.3.107 [preprint doi:10.48550/arXiv.2301.13562].

[20] E. Bothmann, T. Childers, W. Giele, F. Herren, S. Höche, J. Isaacson, M. Knobbe and R. Wang, *Efficient phase-space generation for hadron collider event simulation*, SciPost Phys. **15**, 169 (2023), doi:10.21468/SciPostPhys.15.4.169 [preprint doi:10.48550/arXiv.2302.10449].

[21] N. Deutschmann and N. Götz, *Accelerating HEP simulations with neural importance sampling*, J. High Energy Phys. **03**, 083 (2024), doi:10.1007/JHEP03(2024)083 [preprint doi:10.48550/arXiv.2401.09069].

[22] T. Heimel, O. Mattelaer, T. Plehn and R. Winterhalder, *Differentiable MadNIS-Lite*, SciPost Phys. **18**, 017 (2025), doi:10.21468/SciPostPhys.18.1.017 [preprint doi:10.48550/arXiv.2408.01486].

[23] 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), doi:10.1007/s41781-017-0004-6 [preprint doi:10.48550/arXiv.1701.05927].

[24] A. Andreassen, I. Feige, C. Frye and M. D. Schwartz, *JUNIPR: A framework for unsupervised machine learning in particle physics*, Eur. Phys. J. C **79**, 102 (2019), doi:10.1140/epjc/s10052-019-6607-9 [preprint doi:10.48550/arXiv.1804.09720].

[25] E. Bothmann and L. Del Debbio, *Reweighting a parton shower using a neural network: The final-state case*, J. High Energy Phys. **01**, 033 (2019), doi:10.1007/JHEP01(2019)033 [preprint doi:10.48550/arXiv.1808.07802].

[26] K. Dohi, *Variational autoencoders for jet simulation*, (arXiv preprint) doi:10.48550/arXiv.2009.04842.

[27] E. Buhmann, G. Kasieczka and J. Thaler, *EPiC-GAN: Equivariant point cloud generation for particle jets*, SciPost Phys. **15**, 130 (2023), doi:10.21468/SciPostPhys.15.4.130 [preprint doi:10.48550/arXiv.2301.08128].

[28] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch and T. Golling, *PC-JeDi: Diffusion for particle cloud generation in high energy physics*, SciPost Phys. **16**, 018 (2024), doi:10.21468/SciPostPhys.16.1.018 [preprint doi:10.48550/arXiv.2303.05376].

[29] V. Mikuni, B. Nachman and M. Pettee, *Fast point cloud generation with diffusion models in high energy physics*, Phys. Rev. D **108**, 036025 (2023), doi:10.1103/PhysRevD.108.036025 [preprint doi:10.48550/arXiv.2304.01266].

[30] E. Buhmann et al., *EPiC-ly fast particle cloud generation with flow-matching and diffusion*, (arXiv preprint) doi:10.48550/arXiv.2310.00049.

[31] 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), doi:10.1103/PhysRevLett.120.042003 [preprint doi:10.48550/arXiv.1705.02355].

[32] 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), doi:10.1088/1742-6596/1085/4/042017 [preprint doi:10.48550/arXiv.1711.08813].

[33] 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), doi:10.1103/PhysRevD.97.014021 [preprint doi:10.48550/arXiv.1712.10321].

[34] 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), doi:10.1007/s41781-018-0008-x [preprint doi:10.48550/arXiv.1802.03325].

[35] 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), doi:10.1007/s41781-018-0019-7 [preprint doi:10.48550/arXiv.1807.01954].

[36] D. Belayneh et al., *Calorimetry with deep learning: Particle simulation and reconstruction for collider physics*, Eur. Phys. J. C **80**, 688 (2020), doi:10.1140/epjc/s10052-020-8251-9 [preprint doi:10.48550/arXiv.1912.06794].

[37] 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), doi:10.1007/s41781-021-00056-0 [preprint doi:10.48550/arXiv.2005.05334].

[38] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol and K. Krüger, *Decoding photons: Physics in the latent space of a BIB-AE generative network*, Eur. Phys. J. Web Conf. **251**, 03003 (2021), doi:10.1051/epjconf/202125103003 [preprint doi:10.48550/arXiv.2102.12491].

[39] C. Krause and D. Shih, *Fast and accurate simulations of calorimeter showers with normalizing flows*, Phys. Rev. D **107**, 113003 (2023), doi:10.1103/PhysRevD.107.113003 [preprint doi:10.48550/arXiv.2106.05285].

[40] ATLAS collaboration, *AtlFast3: The next generation of fast simulation in ATLAS*, Comput. Softw. Big Sci. **6**, 7 (2022), doi:10.1007/s41781-021-00079-7 [preprint doi:10.48550/arXiv.2109.02551].

[41] C. Krause and D. Shih, *Accelerating accurate simulations of calorimeter showers with normalizing flows and probability density distillation*, Phys. Rev. D **107**, 113004 (2023), doi:10.1103/PhysRevD.107.113004 [preprint doi:10.48550/arXiv.2110.11377].

[42] E. Buhmann et al., *Hadrons, better, faster, stronger*, Mach. Learn.: Sci. Technol. **3**, 025014 (2022), doi:10.1088/2632-2153/ac7848 [preprint doi:10.48550/arXiv.2112.09709].

[43] C. Chen, O. Cerri, T. Q. Nguyen, J. R. Vlimant and M. Pierini, *Analysis-specific fast simulation at the LHC with deep learning*, Comput. Softw. Big Sci. **5**, 15 (2021), doi:10.1007/s41781-021-00060-4.

[44] V. Mikuni and B. Nachman, *Score-based generative models for calorimeter shower simulation*, Phys. Rev. D **106**, 092009 (2022), doi:10.1103/PhysRevD.106.092009 [preprint doi:10.48550/arXiv.2206.11898].

[45] ATLAS collaboration, *Deep generative models for fast photon shower simulation in ATLAS*, Comput. Softw. Big Sci. **8**, 7 (2024), doi:10.1007/s41781-023-00106-9 [preprint doi:10.48550/arXiv.2210.06204].

[46] C. Krause, I. Pang and D. Shih, *CaloFlow for CaloChallenge dataset 1*, SciPost Phys. **16**, 126 (2024), doi:10.21468/SciPostPhys.16.5.126 [preprint doi:10.48550/arXiv.2210.14245].

[47] J. C. Cresswell, B. L. Ross, G. Loaiza-Ganem, H. Reyes-Gonzalez, M. Letizia and A. L. Caterini, *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*, (arXiv preprint) doi:10.48550/arXiv.2211.15380.

[48] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, C. Krause, I. Shekhzadeh and D. Shih, *L2LFlows: Generating high-fidelity 3D calorimeter images*, J. Instrum. **18**, P10017 (2023), doi:10.1088/1748-0221/18/10/P10017 [preprint doi:10.48550/arXiv.2302.11594].

[49] B. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn and T. Kuhr, *Ultra-high-granularity detector simulation with intra-event aware generative adversarial network and self-supervised relational reasoning*, Nat. Commun. **15**, 4916 (2024), doi:10.1038/s41467-024-49104-4 [preprint doi:10.48550/arXiv.2303.08046].

[50] A. Xu, S. Han, X. Ju and H. Wang, *Generative machine learning for detector response modeling with a conditional normalizing flow*, J. Instrum. **19**, P02003 (2024), doi:10.1088/1748-0221/19/02/P02003 [preprint doi:10.48550/arXiv.2303.10148].

[51] S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, K. Krüger, P. McKeown and L. Rustige, *New angles on fast calorimeter shower simulation*, Mach. Learn.: Sci. Technol. **4**, 035044 (2023), doi:10.1088/2632-2153/acefa9 [preprint doi:10.48550/arXiv.2303.18150].

[52] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasicezka, A. Korol, W. Korcari, K. Krüger and P. McKeown, *CaloClouds: Fast geometry-independent highly-granular calorimeter simulation*, J. Instrum. **18**, P11025 (2023), doi:10.1088/1748-0221/18/11/P11025 [preprint doi:10.48550/arXiv.2305.04847].

[53] M. R. Buckley, I. Pang, D. Shih and C. Krause, *Inductive simulation of calorimeter showers with normalizing flows*, Phys. Rev. D **109**, 033006 (2024), doi:10.1103/PhysRevD.109.033006 [preprint doi:10.48550/arXiv.2305.11934].

[54] S. Diefenbacher, V. Mikuni and B. Nachman, *Refining fast calorimeter simulations with a Schrödinger bridge*, J. Instrum. **20**, P08007 (2025), doi:10.1088/1748-0221/20/08/P08007 [preprint doi:10.48550/arXiv.2308.12339].

[55] F. Ernst, L. Favaro, C. Krause, T. Plehn and D. Shih, *Normalizing flows for high-dimensional detector simulations*, SciPost Phys. **18**, 081 (2025), doi:10.21468/SciPostPhys.18.3.081 [preprint doi:10.48550/arXiv.2312.09290].

[56] L. Favaro, A. Ore, S. Palacios Schweitzer and T. Plehn, *CaloDREAM – Detector response emulation via attentive flow matching*, SciPost Phys. **18**, 088 (2025), doi:10.21468/SciPostPhys.18.3.088 [preprint doi:10.48550/arXiv.2405.09629].

[57] T. Buss, F. Gaede, G. Kasieczka, C. Krause and D. Shih, *Convolutional L2LFlows: Generating accurate showers in highly granular calorimeters using convolutional normalizing flows*, J. Instrum. **19**, P09003 (2024), doi:10.1088/1748-0221/19/09/P09003 [preprint doi:10.48550/arXiv.2405.20407].

[58] G. Quétant, J. A. Raine, M. Leigh, D. Sengupta and T. Golling, *Generating variable length full events from partons*, Phys. Rev. D **110**, 076023 (2024), doi:10.1103/PhysRevD.110.076023 [preprint doi:10.48550/arXiv.2406.13074].

[59] ATLAS collaboration, *Quark versus gluon jet tagging using jet images with the ATLAS detector*, Tech. Rep. ATL-PHYS-PUB-2017-017, CERN, Geneva, Switzerland (2017), https://cds.cern.ch/record/2275641.

[60] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, *Deep learning in color: Towards automated quark/gluon jet discrimination*, J. High Energy Phys. **01**, 110 (2017), doi:10.1007/JHEP01(2017)110 [preprint doi:10.48550/arXiv.1612.01551].

[61] T. Cheng, *Recursive neural networks in quark/gluon tagging*, Comput. Softw. Big Sci. **2**, 3 (2018), doi:10.1007/s41781-018-0007-y [preprint doi:10.48550/arXiv.1711.02633].

[62] M. Stoye, J. Kieseler, H. Qu, L. Gouskos and M. Verzetti, *DeepJet: Generic physics object based jet multiclass classification for LHC experiments*, Workshop on Deep Learning for Physical Sciences, NIPS, Long Beach, USA (2017).

[63] Y.-T. Chien and R. Kunnawalkam Elayavalli, *Probing heavy ion collisions using quark and gluon jet substructure*, (arXiv preprint) doi:10.48550/arXiv.1803.03589.

[64] E. A. Moreno et al., *JEDI-net: A jet identification algorithm based on interaction networks*, Eur. Phys. J. C **80**, 58 (2020), doi:10.1140/epjc/s10052-020-7608-4 [preprint doi:10.48550/arXiv.1908.05318].

[65] G. Kasieczka, N. Kiefer, T. Plehn and J. Thompson, *Quark-gluon tagging: Machine learning vs detector*, SciPost Phys. **6**, 069 (2019), doi:10.21468/SciPostPhys.6.6.069 [preprint doi:10.48550/arXiv.1812.09223].

[66] G. Kasieczka, S. Marzani, G. Soyez and G. Stagnitto, *Towards machine learning analytics for jet substructure*, J. High Energy Phys. **09**, 195 (2020), doi:10.1007/JHEP09(2020)195 [preprint doi:10.48550/arXiv.2007.04319].

[67] J. S. H. Lee, S. M. Lee, Y. Lee, I. Park, I. J. Watson and S. Yang, *Quark gluon jet discrimination with weakly supervised learning*, J. Korean Phys. Soc. **75**, 652 (2019), doi:10.3938/jkps.75.652 [preprint doi:10.48550/arXiv.2012.02540].

[68] J. S. H. Lee, I. Park, I. J. Watson and S. Yang, *Quark-gluon jet discrimination using convolutional neural networks*, J. Korean Phys. Soc. **74**, 219 (2019), doi:10.3938/jkps.74.219 [preprint doi:10.48550/arXiv.2012.02531].

[69] F. A. Dreyer and H. Qu, *Jet tagging in the Lund plane with graph networks*, J. High Energy Phys. **03**, 052 (2021), doi:10.1007/JHEP03(2021)052 [preprint doi:10.48550/arXiv.2012.08526].

[70] A. Romero, D. Whiteson, M. Fenton, J. Collado and P. Baldi, *Safety of quark/gluon jet classification*, (arXiv preprint) doi:10.48550/arXiv.2103.09103.

[71] J. Filipek, S.-C. Hsu, J. Kruper, K. Mohan and B. Nachman, *Identifying the quantum properties of hadronic resonances using machine learning*, SciPost Phys. Core **8**, 039 (2025), doi:10.21468/SciPostPhysCore.8.2.039 [preprint doi:10.48550/arXiv.2105.04582].

[72] F. A. Dreyer, G. Soyez and A. Takacs, *Quarks and gluons in the Lund plane*, J. High Energy Phys. **08**, 177 (2022), doi:10.1007/JHEP08(2022)177 [preprint doi:10.48550/arXiv.2112.09140].

[73] S. Bright-Thonney, I. Moult, B. Nachman and S. Prestel, *Systematic quark/gluon identification with ratios of likelihoods*, J. High Energy Phys. **12**, 021 (2022), doi:10.1007/JHEP12(2022)021 [preprint doi:10.48550/arXiv.2207.12411].

[74] M. Crispim Romão, J. G. Milhano and M. van Leeuwen, *Jet substructure observables for jet quenching in quark gluon plasma: A machine learning driven analysis*, SciPost Phys. **16**, 015 (2024), doi:10.21468/SciPostPhys.16.1.015 [preprint doi:10.48550/arXiv.2304.07196].

[75] D. Athanasakos, A. J. Larkoski, J. Mulligan, M. Płoskoń and F. Ringer, *Is infrared-collinear safe information all you need for jet classification?*, J. High Energy Phys. **07**, 257 (2024), doi:10.1007/JHEP07(2024)257 [preprint doi:10.48550/arXiv.2305.08979].

[76] M. He and D. Wang, *Quark/gluon discrimination and top tagging with dual attention transformer*, Eur. Phys. J. C **83**, 1116 (2023), doi:10.1140/epjc/s10052-023-12293-1 [preprint doi:10.48550/arXiv.2307.04723].

[77] W. Shen, D. Wang and J. M. Yang, *Hierarchical high-point energy flow network for jet tagging*, J. High Energy Phys. **09**, 135 (2023), doi:10.1007/JHEP09(2023)135 [preprint doi:10.48550/arXiv.2308.08300].

[78] M. J. Dolan, J. Gargalionis and A. Ore, *Quark-versus-gluon tagging in CMS open data with CWoLa and TopicFlow*, J. High Energy Phys. **08**, 024 (2025), doi:10.1007/JHEP08(2025)024 [preprint doi:10.48550/arXiv.2312.03434].

[79] F. Blekman, F. Canelli, A. De Moor, K. Gautam, A. Ilg, A. Macchiolo and E. Ploerer, *Tagging more quark jet flavours at FCC-ee at 91 GeV with a transformer-based neural network*, Eur. Phys. J. C **85**, 165 (2025), doi:10.1140/epjc/s10052-025-13785-y [preprint doi:10.48550/arXiv.2406.08590].

[80] J. Orozco Sandoval, V. Manian and S. Malik, *A multicategory jet image classification framework using deep neural network*, (arXiv preprint) doi:10.48550/arXiv.2407.03524.

[81] Y. Wu, K. Wang, C. Li, H. Qu and J. Zhu, *Jet tagging with more-interaction particle transformer*, Chin. Phys. C **49**, 013110 (2025), doi:10.1088/1674-1137/ad7f3d [preprint doi:10.48550/arXiv.2407.08682].

[82] R. Tagami, T. Suehara and M. Ishino, *Application of particle transformer to quark flavor tagging in the ILC project*, Eur. Phys. J. Web Conf. **315**, 03011 (2024), doi:10.1051/epjconf/202431503011 [preprint doi:10.48550/arXiv.2410.11322].

[83] J. Brehmer, V. Bresó, P. de Haan, T. Plehn, H. Qu, J. Spinner and J. Thaler, *A Lorentz-equivariant transformer for all of the LHC*, SciPost Phys. **19**, 108 (2025), doi:10.21468/SciPostPhys.19.4.108 [preprint doi:10.48550/arXiv.2411.00446].

[84] J. Geuskens, N. Gite, M. Krämer, V. Mikuni, A. Mück, B. Nachman and H. Reyes-González, *Fundamental limit of jet tagging*, Phys. Rev. D **112**, L091901 (2025), doi:10.1103/mjj1-w2b1 [preprint doi:10.48550/arXiv.2411.02628].

[85] A. Andreassen and B. Nachman, *Neural networks for full phase-space reweighting and parameter tuning*, Phys. Rev. D **101**, 091901 (2020), doi:10.1103/PhysRevD.101.091901 [preprint doi:10.48550/arXiv.1907.08209].

[86] M. Stoye, J. Brehmer, G. Louppe, J. Pavez and K. Cranmer, *Likelihood-free inference with an improved cross-entropy estimator*, (arXiv preprint) doi:10.48550/arXiv.1808.00973.

[87] J. Hollingsworth and D. Whiteson, *Resonance searches with machine learned likelihood ratios*, (arXiv preprint) doi:10.48550/arXiv.2002.04699.

[88] J. Brehmer, K. Cranmer, G. Louppe and J. Pavez, *Constraining effective field theories with machine learning*, Phys. Rev. Lett. **121**, 111801 (2018), doi:10.1103/PhysRevLett.121.111801 [preprint doi:10.48550/arXiv.1805.00013].

[89] J. Brehmer, K. Cranmer, G. Louppe and J. Pavez, *A guide to constraining effective field theories with machine learning*, Phys. Rev. D **98**, 052004 (2018), doi:10.1103/PhysRevD.98.052004 [preprint doi:10.48550/arXiv.1805.00020].

[90] J. Brehmer, F. Kling, I. Espejo and K. Cranmer, *MadMiner: Machine learning-based inference for particle physics*, Comput. Softw. Big Sci. **4**, 3 (2020), doi:10.1007/s41781-020-0035-2 [preprint doi:10.48550/arXiv.1907.10621].

[91] J. Brehmer, G. Louppe, J. Pavez and K. Cranmer, *Mining gold from implicit models to improve likelihood-free inference*, Proc. Natl. Acad. Sci. **117**, 5242 (2020), doi:10.1073/pnas.1915980117 [preprint doi:10.48550/arXiv.1805.12244].

[92] K. Cranmer, J. Pavez and G. Louppe, *Approximating likelihood ratios with calibrated discriminative classifiers*, (arXiv preprint) doi:10.48550/arXiv.1506.02169.

[93] A. Andreassen, S.-C. Hsu, B. Nachman, N. Suaysom and A. Suresh, *Parameter estimation using neural networks in the presence of detector effects*, Phys. Rev. D **103**, 036001 (2021), doi:10.1103/PhysRevD.103.036001 [preprint doi:10.48550/arXiv.2010.03569].

[94] A. Coogan, K. Karchev and C. Weniger, *Targeted likelihood-free inference of dark matter substructure in strongly-lensed galaxies*, (arXiv preprint) doi:10.48550/arXiv.2010.07032.

[95] F. Flesher, K. Fraser, C. Hutchison, B. Ostdiek and M. D. Schwartz, *Parameter inference from event ensembles and the top-quark mass*, J. High Energy Phys. **09**, 058 (2021), doi:10.1007/JHEP09(2021)058 [preprint doi:10.48550/arXiv.2011.04666].

[96] S. Bieringer, A. Butter, T. Heimel, S. Höche, U. Köthe, T. Plehn and S. T. Radev, *Measuring QCD splittings with invertible networks*, SciPost Phys. **10**, 126 (2021), doi:10.21468/SciPostPhys.10.6.126 [preprint doi:10.48550/arXiv.2012.09873].

[97] B. Nachman and J. Thaler, *Learning from many collider events at once*, Phys. Rev. D **103**, 116013 (2021), doi:10.1103/PhysRevD.103.116013 [preprint doi:10.48550/arXiv.2101.07263].

[98] S. Chatterjee, N. Frohner, L. Lechner, R. Schöfbeck and D. Schwarz, *Tree boosting for learning EFT parameters*, Comput. Phys. Commun. **277**, 108385 (2022), doi:10.1016/j.cpc.2022.108385 [preprint doi:10.48550/arXiv.2107.10859].

[99] S. Shirobokov, V. Belavin, M. Kagan, A. Ustyuzhanin and A. G. Baydin, *Black-box optimization with local generative surrogates*, (arXiv preprint) doi:10.48550/arXiv.2002.04632.

[100] S. Mishra-Sharma and K. Cranmer, *Neural simulation-based inference approach for characterizing the Galactic Center γ-ray excess*, Phys. Rev. D **105**, 063017 (2022), doi:10.1103/PhysRevD.105.063017 [preprint doi:10.48550/arXiv.2110.06931].

[101] R. K. Barman, D. Gonçalves and F. Kling, *Machine learning the Higgs boson-top quark CP phase*, Phys. Rev. D **105**, 035023 (2022), doi:10.1103/PhysRevD.105.035023 [preprint doi:10.48550/arXiv.2110.07635].

[102] H. Bahl and S. Brass, *Constraining $\mathcal{CP}$-violation in the Higgs-top-quark interaction using machine-learning-based inference*, J. High Energy Phys. **03**, 017 (2022), doi:10.1007/JHEP03(2022)017 [preprint doi:10.48550/arXiv.2110.10177].

[103] E. Arganda, X. Marcano, V. Martín Lozano, A. D. Medina, A. D. Perez, M. Szewc and A. Szynkman, *A method for approximating optimal statistical significances with machine-learned likelihoods*, Eur. Phys. J. C **82**, 993 (2022), doi:10.1140/epjc/s10052-022-10944-3 [preprint doi:10.48550/arXiv.2205.05952].

[104] K. Kong, K. T. Matchev, S. Mrenna and P. Shyamsundar, *New machine learning techniques for simulation-based inference: InferoStatic nets, kernel score estimation, and kernel likelihood ratio estimation*, SciPost Phys. Codebases 14 (2023), doi:10.21468/SciPostPhysCodeb.14 [preprint doi:10.48550/arXiv.2210.01680].

[105] E. Arganda, A. D. Perez, M. de los Rios and R. M. Sandá Seoane, *Machine-learned exclusion limits without binning*, Eur. Phys. J. C **83**, 1158 (2023), doi:10.1140/epjc/s10052-023-12314-z [preprint doi:10.48550/arXiv.2211.04806].

[106] A. Butter, T. Heimel, T. Martini, S. Peitzsch and T. Plehn, *Two invertible networks for the matrix element method*, SciPost Phys. **15**, 094 (2023), doi:10.21468/SciPostPhys.15.3.094 [preprint doi:10.48550/arXiv.2210.00019].

[107] M. Neubauer, M. Feickert, M. Katare and A. Roy, *Deep learning for the matrix element method*, Proc. Sci. **414**, 246 (2022), doi:10.22323/1.414.0246 [preprint doi:10.48550/arXiv.2211.11910].

[108] S. Rizvi, M. Pettee and B. Nachman, *Learning likelihood ratios with neural network classifiers*, J. High Energy Phys. **02**, 136 (2024), doi:10.1007/JHEP02(2024)136 [preprint doi:10.48550/arXiv.2305.10500].

[109] L. Heinrich, S. Mishra-Sharma, C. Pollard and P. Windischhofer, *Hierarchical neural simulation-based inference over event ensembles*, (arXiv preprint) doi:10.48550/arXiv.2306.12584.

[110] D. Breitenmoser, F. Cerutti, G. Butterweck, M. M. Kasprzak and S. Mayer, *Emulator-based Bayesian inference on non-proportional scintillation models by compton-edge probing*, Nat. Commun. **14**, 7790 (2023), doi:10.1038/s41467-023-42574-y [preprint doi:10.48550/arXiv.2302.05641].

[111] M. Erdogan, N. B. Baytekin, S. E. Coban and A. Demir, *Machine learning and Kalman filtering for nanomechanical mass spectrometry*, IEEE Sens. J. **24**, 6303 (2024), doi:10.1109/JSEN.2024.3350730 [preprint doi:10.48550/arXiv.2306.00563].

[112] A. Morandini, T. Ferber and F. Kahlhoefer, *Reconstructing axion-like particles from beam dumps with simulation-based inference*, Eur. Phys. J. C **84**, 200 (2024), doi:10.1140/epjc/s10052-024-12557-4 [preprint doi:10.48550/arXiv.2308.01353].

[113] R. Barrué, P. Conde Muíño, V. Dao and R. Santos, *Simulation-based inference in the search for CP violation in leptonic WH production*, J. High Energy Phys. **04**, 014 (2024), doi:10.1007/JHEP04(2024)014 [preprint doi:10.48550/arXiv.2308.02882].

[114] I. Espejo, S. Pérez, K. Hurtado, L. Heinrich and K. Cranmer, *Scaling MadMiner with a deployment on REANA*, (arXiv preprint) doi:10.48550/arXiv.2304.05814.

[115] T. Heimel, N. Huetsch, R. Winterhalder, T. Plehn and A. Butter, *Precision-machine learning for the matrix element method*, SciPost Phys. **17**, 129 (2024), doi:10.21468/SciPostPhys.17.5.129 [preprint doi:10.48550/arXiv.2310.07752].

[116] S. Chai, J. Gu and L. Li, *From optimal observables to machine learning: An effective-field-theory analysis of $e^+e^- \to W^+W^-$ at future lepton colliders*, J. High Energy Phys. **05**, 292 (2024), doi:10.1007/JHEP05(2024)292 [preprint doi:10.48550/arXiv.2401.02474].

[117] S. Chatterjee, S. Sánchez Cruz, R. Schöfbeck and D. Schwarz, *Rotation-equivariant graph neural network for learning hadronic SMEFT effects*, Phys. Rev. D **109**, 076012 (2024), doi:10.1103/PhysRevD.109.076012 [preprint doi:10.48550/arXiv.2401.10323].

[118] E. Alvarez, L. Da Rold, M. Szewc, A. Szynkman, S. A. Tanco and T. Tarutina, *Improvement and generalization of ABCD method with Bayesian inference*, SciPost Phys. Core **7**, 043 (2024), doi:10.21468/SciPostPhysCore.7.3.043 [preprint doi:10.48550/arXiv.2402.08001].

[119] M. A. Diaz, G. Cerro, S. Dasmahapatra and S. Moretti, *Bayesian active search on parameter space: A 95 GeV spin-0 resonance in the (B−L)SSM*, SciPost Phys. Core **8**, 068 (2025), doi:10.21468/SciPostPhysCore.8.4.068 [preprint doi:10.48550/arXiv.2404.18653].

[120] R. Mastandrea, B. Nachman and T. Plehn, *Constraining the Higgs potential with neural simulation-based inference for di-Higgs production*, Phys. Rev. D **110**, 056004 (2024), doi:10.1103/PhysRevD.110.056004 [preprint doi:10.48550/arXiv.2405.15847].

[121] JETSCAPE collaboration: R. Ehlers et al., *Bayesian inference analysis of jet quenching using inclusive jet and hadron suppression measurements*, Phys. Rev. C **111**, 054913 (2025), doi:10.1103/PhysRevC.111.054913 [preprint doi:10.48550/arXiv.2408.08247].

[122] H. Bahl, V. Bresó-Pla, G. De Crescenzo and T. Plehn, *Advancing tools for simulation-based inference*, SciPost Phys. Core **8**, 060 (2025), doi:10.21468/SciPostPhysCore.8.3.060 [preprint doi:10.48550/arXiv.2410.07315].

[123] D. Maître, V. S. Ngairangbam and M. Spannowsky, *Optimal equivariant architectures from the symmetries of matrix-element likelihoods*, Mach. Learn.: Sci. Technol. **6**, 015059 (2025), doi:10.1088/2632-2153/adbab1 [preprint doi:10.48550/arXiv.2410.18553].

[124] T. Heimel, T. Plehn and N. Schmal, *Profile likelihoods on ML-steroids*, (arXiv preprint) doi:10.48550/arXiv.2411.00942.

[125] J. Duarte et al., *Fast inference of deep neural networks in FPGAs for particle physics*, J. Instrum. **13**, P07027 (2018), doi:10.1088/1748-0221/13/07/P07027 [preprint doi:10.48550/arXiv.1804.06913].

[126] S. Summers et al., *Fast inference of boosted decision trees in FPGAs for particle physics*, J. Instrum. **15**, P05026 (2020), doi:10.1088/1748-0221/15/05/P05026 [preprint doi:10.48550/arXiv.2002.02534].

[127] Y. Iiyama et al., *Distance-weighted graph neural networks on FPGAs for real-time particle reconstruction in high energy physics*, Front. Big Data **3**, 598927 (2021), doi:10.3389/fdata.2020.598927 [preprint doi:10.48550/arXiv.2008.03601].

[128] A. Heintz et al., *Accelerated charged particle tracking with graph neural networks on FPGAs*, (arXiv preprint) doi:10.48550/arXiv.2012.01563.

[129] T. Aarrestad et al., *Fast convolutional neural networks on FPGAs with hls4ml*, Mach. Learn.: Sci. Technol. **2**, 045015 (2021), doi:10.1088/2632-2153/ac0ea1 [preprint doi:10.48550/arXiv.2101.05108].

[130] T. M. Hong, B. T. Carlson, B. R. Eubanks, S. T. Racz, S. T. Roche, J. Stelzer and D. C. Stumpp, *Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics*, J. Instrum. **16**, P08016 (2021), doi:10.1088/1748-0221/16/08/P08016 [preprint doi:10.48550/arXiv.2104.03408].

[131] M. Migliorini, J. Pazzini, A. Triossi, M. Zanetti and A. Zucchetta, *Muon trigger with fast neural networks on FPGA, a demonstrator*, J. Phys.: Conf. Ser. **2374**, 012099 (2022), doi:10.1088/1742-6596/2374/1/012099 [preprint doi:10.48550/arXiv.2105.04428].

[132] E. Govorkova et al., *Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider*, Nat. Mach. Intell. **4**, 154 (2022), doi:10.1038/s42256-022-00441-3 [preprint doi:10.48550/arXiv.2108.03986].

[133] A. Elabd et al., *Graph neural networks for charged particle tracking on FPGAs*, Front. Big Data **5**, 828666 (2022), doi:10.3389/fdata.2022.828666 [preprint doi:10.48550/arXiv.2112.02048].

[134] C. Sun, T. Nakajima, Y. Mitsumori, Y. Horii and M. Tomoto, *Fast muon tracking with machine learning implemented in FPGA*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **1045**, 167546 (2023), doi:10.1016/j.nima.2022.167546 [preprint doi:10.48550/arXiv.2202.04976].

[135] E. E. Khoda et al., *Ultra-low latency recurrent neural network inference on FPGAs for physics applications with hls4ml*, Mach. Learn.: Sci. Technol. **4**, 025004 (2023), doi:10.1088/2632-2153/acc0d7 [preprint doi:10.48550/arXiv.2207.00559].

[136] B. T. Carlson, Q. Bayer, T. M. Hong and S. T. Roche, *Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics*, J. Instrum. **17**, P09039 (2022), doi:10.1088/1748-0221/17/09/P09039 [preprint doi:10.48550/arXiv.2207.05602].

[137] H. Abidi, A. Boveia, V. Cavaliere, D. Furletov, A. Gekow, C. W. Kalderon and S. Yoo, *Charged particle tracking with machine learning on FPGAs*, (arXiv preprint) doi:10.48550/arXiv.2212.02348.

[138] R. Herbst, R. Coffee, N. Fronk, K. Kim, K. Kim, L. Ruckman and J. J. Russell, *Implementation of a framework for deploying AI inference engines in FPGAs*, in *Accelerating science and engineering discoveries through integrated research infrastructure for experiment, big data, modeling and simulation*, Springer, Cham, Switzerland, ISBN 9783031236051 (2022), doi:10.1007/978-3-031-23606-8_8 [preprint doi:10.48550/arXiv.2305.19455].

[139] A. Coccaro, F. A. Di Bello, S. Giagu, L. Rambelli and N. Stocchetti, *Fast neural network inference on FPGAs for triggering on long-lived particles at colliders*, Mach. Learn.: Sci. Technol. **4**, 045040 (2023), doi:10.1088/2632-2153/ad087a [preprint doi:10.48550/arXiv.2307.05152].

[140] M. Neu, J. Becker, P. Dorwarth, T. Ferber, L. Reuter, S. Stefkova and K. Unger, *Real-time graph building on FPGAs for machine learning trigger applications in particle physics*, Comput. Softw. Big Sci. **8**, 8 (2024), doi:10.1007/s41781-024-00117-0 [preprint doi:10.48550/arXiv.2307.07289].

[141] L. Borella, A. Coppi, A. Triossi, J. Pazzini, M. Zanetti, A. Stanco and M. Trenti, *Ultra-low latency quantum-inspired machine learning predictors implemented on FPGA*, Mach. Learn.: Sci. Technol. **6**, 045062 (2025), doi:10.1088/2632-2153/ae25b5 [preprint doi:10.48550/arXiv.2409.16075].

[142] P. Serhiayenka, S. T. Roche, B. T. Carlson and T. M. Hong, *Nanosecond hardware regression trees in FPGA at the LHC*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **1072**, 170209 (2025), doi:10.1016/j.nima.2025.170209 [preprint doi:10.48550/arXiv.2409.20506].

[143] L. Tani, J. Pata and J. Birk, *Reconstructing hadronically decaying tau leptons with a jet foundation model*, SciPost Phys. Core **8**, 046 (2025), doi:10.21468/SciPostPhysCore.8.3.046 [preprint doi:10.48550/arXiv.2503.19165].

[144] V. Mikuni and B. Nachman, *Method to simultaneously facilitate all jet physics tasks*, Phys. Rev. D **111**, 054015 (2025), doi:10.1103/PhysRevD.111.054015 [preprint doi:10.48550/arXiv.2502.14652].

[145] O. Amram et al., *Aspen open jets: Unlocking LHC data for foundation models in particle physics*, Mach. Learn.: Sci. Technol. **6**, 030601 (2025), doi:10.1088/2632-2153/ade58f [preprint doi:10.48550/arXiv.2412.10504].

[146] J. Ho, B. R. Roberts, S. Han and H. Wang, *Pretrained event classification model for high energy physics analysis*, (arXiv preprint) doi:10.48550/arXiv.2412.10665.

[147] A. J. Wildridge, J. P. Rodgers, E. M. Colbert, Y. Yao, A. W. Jung and M. Liu, *Bumblebee: Foundation model for particle physics discovery*, (arXiv preprint) doi:10.48550/arXiv.2412.07867.

[148] M. Leigh, S. Klein, F. Charton, T. Golling, L. Heinrich, M. Kagan, I. Ochoa and M. Osadchy, *Is tokenization needed for masked particle modeling?*, Mach. Learn.: Sci. Technol. **6**, 025075 (2025), doi:10.1088/2632-2153/addb98 [preprint doi:10.48550/arXiv.2409.12589].

[149] V. Mikuni and B. Nachman, *Solving key challenges in collider physics with foundation models*, Phys. Rev. D **111**, L051504 (2025), doi:10.1103/PhysRevD.111.L051504 [preprint doi:10.48550/arXiv.2404.16091].

[150] P. Harris, J. Krupa, M. Kagan, B. Maier and N. Woodward, *Resimulation-based self-supervised learning for pretraining physics foundation models*, Phys. Rev. D **111**, 032010 (2025), doi:10.1103/PhysRevD.111.032010 [preprint doi:10.48550/arXiv.2403.07066].

[151] J. Birk, A. Hallin and G. Kasieczka, *OmniJet-α: The first cross-task foundation model for particle physics*, Mach. Learn.: Sci. Technol. **5**, 035031 (2024), doi:10.1088/2632-2153/ad66ad [preprint doi:10.48550/arXiv.2403.05618].

[152] S. Han, J. Pool, J. Tran and W. J. Dally, *Learning both weights and connections for efficient neural networks*, (arXiv preprint) doi:10.48550/arXiv.1506.02626.

[153] S. Han, H. Mao and W. J. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding*, (arXiv preprint) doi:10.48550/arXiv.1510.00149.

[154] D. D. Lin, S. S. Talathi and V. S. Annapureddy, *Fixed point quantization of deep convolutional networks*, (arXiv preprint) doi:10.48550/arXiv.1511.06393.

[155] M. Courbariaux, Y. Bengio and J.-P. David, *BinaryConnect: Training deep neural networks with binary weights during propagations*, (arXiv preprint) doi:10.48550/arXiv.1511.00363.

[156] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv and Y. Bengio, *Binarized neural networks: Training deep neural networks with weights and activations constrained to* $+1$ *or* -1, (arXiv preprint) doi:10.48550/arXiv.1602.02830.

[157] B. Liu, F. Li, X. Wang, B. Zhang and J. Yan, *Ternary weight networks*, in *IEEE international conference on acoustics, speech and signal processing*, IEEE, New York, USA, ISBN 9781728163277 (2023), doi:10.1109/ICASSP49357.2023.10094626 [preprint doi:10.48550/arXiv.1605.04711].

[158] Y. Cheng, D. Wang, P. Zhou and T. Zhang, *A survey of model compression and acceleration for deep neural networks*, (arXiv preprint) doi:10.48550/arXiv.1710.09282.

[159] B. Moons, K. Goetschalckx, N. Van Berckelaer and M. Verhelst, *Minimum energy quantized neural networks*, in *51st Asilomar conference on signals, systems, and computers*, IEEE, New York, USA, ISBN 9781538618233 (2017), doi:10.1109/ACSSC.2017.8335699 [preprint doi:10.48550/arXiv.1711.00215].

[160] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi and M. Rastegari, *What's hidden in a randomly weighted neural network?*, in *IEEE/CVF conference on computer vision and pattern recognition*, IEEE, New York, USA, ISBN 9781728171685 (2020), doi:10.1109/CVPR42600.2020.01191 [preprint doi:10.48550/arXiv.1911.13299].

[161] M. Jin, Y. Hu and C. A. Argüelles, *Two Watts is all you need: Enabling in-detector real-time machine learning for neutrino telescopes via edge computing*, J. Cosmol. Astropart. Phys. 026 (2024), doi:10.1088/1475-7516/2024/06/026 [preprint doi:10.48550/arXiv.2311.04983].

[162] H. Wang et al., *BitNet: Scaling 1-bit transformers for large language models*, (arXiv preprint) doi:10.48550/arXiv.2310.11453.

[163] S. Ma et al., *The era of 1-bit LLMs: All large language models are in 1.58 bits*, (arXiv preprint) doi:10.48550/arXiv.2402.17764.

[164] M. Dörrich, M. Fan and A. M. Kist, *Impact of mixed precision techniques on training and inference efficiency of deep neural networks*, IEEE Access **11**, 57627 (2023), doi:10.1109/ACCESS.2023.3284388.

[165] J. Ngadiuba et al., *Compressing deep neural networks on FPGAs to binary and ternary precision with hls4ml*, Mach. Learn.: Sci. Technol. **2**, 015001 (2020), doi:10.1088/2632-2153/aba042 [preprint doi:10.48550/arXiv.2003.06308].

[166] B. Hawks, J. Duarte, N. J. Fraser, A. Pappalardo, N. Tran and Y. Umuroglu, *Ps and Qs: Quantization-aware pruning for efficient low latency neural network inference*, Front. Artif. Intell. **4**, 676564 (2021), doi:10.3389/frai.2021.676564 [preprint doi:10.48550/arXiv.2102.11289].

[167] F. Rehm, S. Vallecorsa, V. Saletore, H. Pabst, A. Chaibi, V. Codreanu, K. Borras and D. Krücker, *Reduced precision strategies for deep learning: A high energy physics generative adversarial network use case*, in *Proceedings of the 10th international conference on pattern recognition applications and methods*, SCITEPRESS, Setúbal, Portugal, ISBN 9789897584862 (2021), doi:10.5220/0010245002510258 [preprint doi:10.48550/arXiv.2103.10142].

[168] E. Suarez, J. Amaya, M. Frank, O. Freyermuth, M. Girone, B. Kostrzewa and S. Pfalzner, *Energy efficiency trends in HPC: What high-energy and astrophysicists need to know*, Front. Phys. **13**, 1542474 (2025), doi:10.3389/fphy.2025.1542474 [preprint doi:10.48550/arXiv.2503.17283].

[169] NVIDIA corporation, *Support matrix* (2026), https://docs.nvidia.com/deeplearning/tensorrt/latest/getting-started/support-matrix.html#hardware-precision-matrix.

[170] Microsoft, *BitNet* (2024), https://github.com/microsoft/BitNet.

[171] J. Wang et al., *Bitnet.cpp: Efficient edge inference for ternary LLMs*, (arXiv preprint) doi:10.48550/arXiv.2502.11880.

[172] P. T. Komiske, E. M. Metodiev and J. Thaler, *Energy flow networks: Deep sets for particle jets*, J. High Energy Phys. **01**, 121 (2019), doi:10.1007/JHEP01(2019)121 [preprint doi:10.48550/arXiv.1810.05165].

[173] M. Cacciari, G. P. Salam and G. Soyez, *FastJet user manual*, Eur. Phys. J. C **72**, 1896 (2012), doi:10.1140/epjc/s10052-012-1896-2 [preprint doi:10.48550/arXiv.1111.6097].

[174] H. Qu and L. Gouskos, *Jet tagging via particle clouds*, Phys. Rev. D **101**, 056019 (2020), doi:10.1103/PhysRevD.101.056019 [preprint doi:10.48550/arXiv.1902.08570].

[175] V. Mikuni and F. Canelli, *Point cloud transformers applied to collider physics*, Mach. Learn.: Sci. Technol. **2**, 035027 (2021), doi:10.1088/2632-2153/ac07f6 [preprint doi:10.48550/arXiv.2102.05073].

[176] S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian, W. Du, Z.-M. Ma and T.-Y. Liu, *An efficient Lorentz equivariant graph neural network for jet tagging*, J. High Energy Phys. **07**, 030 (2022), doi:10.1007/JHEP07(2022)030 [preprint doi:10.48550/arXiv.2201.08187].

[177] H. Qu, C. Li and S. Qian, *Particle transformer for jet tagging*, Proc. Mach. Learn. Res. **162**, 18281 (2022) [preprint doi:10.48550/arXiv.2202.03772].

[178] G. Panico, F. Riva and A. Wulzer, *Diboson interference resurrection*, Phys. Lett. B **776**, 473 (2018), doi:10.1016/j.physletb.2017.11.068 [preprint doi:10.48550/arXiv.1708.07823].

[179] B. Hashemi and C. Krause, *Deep generative models for detector signature simulation: A taxonomic review*, Rev. Phys. **12**, 100092 (2024), doi:10.1016/j.revip.2024.100092 [preprint doi:10.48550/arXiv.2312.09597].

[180] C. Krause et al., *CaloChallenge 2022: A community challenge for fast calorimeter simulation*, Rep. Prog. Phys. **88**, 116201 (2025), doi:10.1088/1361-6633/ae1304 [preprint doi:10.48550/arXiv.2410.21611].

[181] M. Faucci Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge* (2022), https://github.com/CaloChallenge/homepage.

[182] S. Agostinelli et al., *Geant4 — A simulation toolkit*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **506**, 250 (2003), doi:10.1016/S0168-9002(03)01368-8.

[183] J. Allison et al., *Geant4 developments and applications*, IEEE Trans. Nucl. Sci. **53**, 270 (2006), doi:10.1109/TNS.2006.869826.

[184] J. Allison et al., *Recent developments in Geant4*, Nucl. Instrum. Methods Phys. Res. A: Accel. Spectrom. Detect. Assoc. Equip. **835**, 186 (2016), doi:10.1016/j.nima.2016.06.125.

[185] M. Faucci Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 - dataset 1 version 2*, Zenodo (2022), doi:10.5281/zenodo.6234054.

[186] M. Faucci Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 - dataset 1 version 3*, Zenodo (2023), doi:10.5281/zenodo.8099322.

[187] M. Faucci Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 - dataset 2*, Zenodo (2022), doi:10.5281/zenodo.6366271.

[188] M. Faucci Giannelli, G. Kasieczka, C. Krause, B. Nachman, D. Salamani, D. Shih and A. Zaborowska, *Fast calorimeter simulation challenge 2022 - dataset 3*, Zenodo (2022), doi:10.5281/zenodo.6366324.

[189] R. Kansal, A. Li, J. Duarte, N. Chernyavskaya, M. Pierini, B. Orzari and T. Tomei, *Evaluating generative models in high energy physics*, Phys. Rev. D **107**, 076017 (2023), doi:10.1103/PhysRevD.107.076017 [preprint doi:10.48550/arXiv.2211.10295].

[190] R. Das, L. Favaro, T. Heimel, C. Krause, T. Plehn and D. Shih, *How to understand limitations of generative networks*, SciPost Phys. **16**, 031 (2024), doi:10.21468/SciPostPhys.16.1.031 [preprint doi:10.48550/arXiv.2305.16774].

[191] M. R. Buckley, I. Pang, D. Shih and C. Krause, *Inductive simulation of calorimeter showers with normalizing flows*, Phys. Rev. D **109**, 033006 (2024), doi:10.1103/PhysRevD.109.033006 [preprint doi:10.48550/arXiv.2305.11934].

[192] I. Pang, D. Shih and J. A. Raine, *Calorimeter shower superresolution*, Phys. Rev. D **109**, 092009 (2024), doi:10.1103/PhysRevD.109.092009 [preprint doi:10.48550/arXiv.2308.11700].

[193] S. Schnake, D. Krücker and K. Borras, *CaloPointFlow II generating calorimeter showers as point clouds*, (arXiv preprint) doi:10.48550/arXiv.2403.15782.

[194] J. C. Cresswell and T. Kim, *Scaling up diffusion and flow-based XGBoost models*, (arXiv preprint) doi:10.48550/arXiv.2408.16046.

[195] E. Dreyer, E. Gross, D. Kobylianskii, V. Mikuni and B. Nachman, *Conditional deep generative models for simultaneous simulation and reconstruction of entire events*, Phys. Rev. D **113**, 032005 (2026), doi:10.1103/14ph-482n [preprint doi:10.48550/arXiv.2503.19981].

[196] L. Dinh, D. Krueger and Y. Bengio, *NICE: Non-linear independent components estimation*, (arXiv preprint) doi:10.48550/arXiv.1410.8516.

[197] L. Dinh, J. Sohl-Dickstein and S. Bengio, *Density estimation using real NVP*, (arXiv preprint) doi:10.48550/arXiv.1605.08803.

[198] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother and U. Köthe, *Analyzing inverse problems with invertible neural networks*, (arXiv preprint) doi:10.48550/arXiv.1808.04730.

[199] T. Müller, B. Mcwilliams, F. Rousselle, M. Gross and J. Novák, *Neural importance sampling*, ACM Trans. Graph. **38**, 1 (2019), doi:10.1145/3341156.

[200] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios, *Neural spline flows*, (arXiv preprint) doi:10.48550/arXiv.1906.04032.

[201] C. Durkan, A. Bekasov, I. Murray and G. Papamakarios, *Cubic-spline flows*, (arXiv preprint) doi:10.48550/arXiv.1906.02145.

[202] C. N. Coelho et al., *Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors*, Nat. Mach. Intell. **3**, 675 (2021), doi:10.1038/s42256-021-00356-5 [preprint doi:10.48550/arXiv.2006.10159].

[203] C. Sun, Z. Que, T. K. Årrestad, V. Loncar, J. Ngadiuba, W. Luk and M. Spiropulu, *HGQ: High granularity quantization for real-time neural networks on FPGAs*, in *ACM/SIGDA international symposium on field programmable gate arrays*, Association for Computing Machinery, New York, USA, ISBN 9798400720796, doi:10.1145/3748173.3779200 [preprint doi:10.48550/arXiv.2405.00645].