SCALING COLLIDER EVENT GENERATION WITH RESIDUAL-QUANTIZED TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full detector simulation and reconstruction of collider events are projected to become major bottlenecks at the High-Luminosity Large Hadron Collider, motivating the development of fast, ML-based surrogates. At the same time, LLMs have driven fast progress in generative discrete modeling: autoregressive transformers trained on tokenized data now represent the state of the art across a range of generative tasks. We extend the discrete modeling paradigm by introducing a particle-level generative model trained on residual-quantized full-event data. We demonstrate the ability of this model family to perform conditional generation from detector-stable particles; we study its scaling behavior across a range of dataset and model sizes, characterize the effects of repeated data exposure and demonstrate that token-level loss systematically predicts downstream physical fidelity. These results provide an empirical framework for scalable collider full-event generation based on residual-quantized representations.

1 INTRODUCTION

At collider experiments, the growing sample size requirements for synthetic datasets are set to become a significant computational bottleneck. Projections for the High-Luminosity upgrade of the Large Hadron Collider (HL-LHC) estimate that, under current paradigms, full simulation tools will outstrip available compute resources, such that simulation and reconstruction of synthetic datasets will exceed the processing requirements of real data (ATLAS Collaboration, 2022; CMS Collaboration, 2026). Fast Simulation (*FastSim*) methods have emerged as solutions to this issue; these techniques use approximations to avoid modeling the full physical aspects of the simulation, hence consuming significantly fewer resources.

Over the past several years, Machine Learning (ML) has become a central tool in High Energy Physics (HEP), with deep learning algorithms now seeing wide adoption across the analysis chain (Butter et al., 2023; Krause et al., 2025; Aad et al., 2026). Concurrently, advances in discrete sequence generation have established robust architectures and well-characterized scaling laws. These developments have been adopted in scientific applications, motivating recent formulations of HEP simulation as a discrete modeling problem.

Related work Previous work in the field has focused on modeling at the jet (Birk et al., 2024; Golling et al., 2024; Amram et al., 2024; Bhimji et al., 2026) and calorimeter (Birk et al., 2025; Cardona-Giraldo et al., 2026) levels. More recently, generative approaches have been extended to conditional full-event simulation (Dreyer et al., 2026), but focusing on flow-matching techniques. In parallel, neural scaling laws have been investigated in the context of jet tagging and generation (Vigl et al., 2026; Amram et al., 2026), building on broader developments in model and data scaling.

In this work, we extend the discrete modeling paradigm to the full-event level. Full-event simulation is typically a two-stage process. Simulating the initial collision, subsequent interactions, and particle decays is relatively inexpensive, whereas simulating the detector response to stable particles and reconstructing the resulting signals is substantially more computationally intensive. The final output particles that are obtained after reconstruction are referred to as Particle Flow (*PF*) objects; these objects approximate the original set of particles entering the detector (Truth particles, *TR*), and are expected to maintain the physical properties and metrics used for downstream analysis. In principle, ML-based fast-simulation algorithms should be able to generate *PF* events without the need for *TR*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

conditioning, but conditional generation (and thus modeling of the detector response instead of the final output distribution) is strongly preferred for the ability to generalize to physical processes (rare or novel/theoretical) not seen during training. Since the model is conditioned on *TR*-level input, it learns the detector response itself as a physical map that is in principle independent of the process that produced the incoming particles, so a single surrogate can serve many signals and backgrounds rather than being re-tuned per analysis. While one could directly generate high-level observables, such as jets (sprays of particles clustered together), producing *PF* objects directly preserves the full event structure. This ensures compatibility with the existing downstream pipelines while providing a flexible foundation for developing new algorithms.

Contribution Building on this full-event formulation, we show how the benefits of discrete residual-quantized representations transfer to point-cloud generation. We then study the scaling behavior of models spanning 10M to 1B parameters, characterize their dependence on model and dataset size, and examine the effects of repeated data exposure. We further demonstrate that the achieved loss is a powerful predictor of modeling performance at the physical scale: improvements in cross-entropy translate predictably into higher fidelity across particle-, jet-, and global event observables. These findings demonstrate that token-level scaling laws can reliably guide model capacity, dataset size, and compute allocation when designing generative surrogates for collider physics.

2 METHODS

Our approach factorizes event generation into two stages. First, a quantizer network is trained to map each truth or reconstructed particle to a short sequence of discrete tokens drawn from a learned codebook, resulting in a fixed vocabulary in which an event is represented as a sequence of integers. Second, an autoregressive transformer models the distribution over these token sequences, so that generation reduces to sampling the next token from a categorical distribution. Once a token sequence is sampled, it is passed back through the autoencoder's decoder to recover continuous particle-level quantities. Decoupling the generative model from the data representation allows the transformer to operate on a compact discrete space, while the mapping to and from physical observables is handled by the separately trained fixed tokenizer.

2.1 DATASET

The model is trained and tested on several physical processes, from the CMS Open Data 2011 collection. Events are generated with Pythia 6.4.25 (Sjöstrand et al., 2008), passed through Geant4 (Agostinelli et al., 2003) simulation of the CMS detector, and subsequently reconstructed using the CMS Particle Flow algorithm (CMS Collaboration, 2009).

In each event we select from the dataset the set of generator-level final state *TR* particles and the corresponding set of *PF* particles. Each particle from these collections is described by its transverse momentum p_T, its pseudorapidity η and azimuthal angle ϕ, its class and the coordinates of its production vertex. Particles are categorized into one of five classes: charged hadron, electron, muon, neutral hadron, photon, and neutral *PF* particles have zeroed vertices (in real datasets the vertex of neutral particles can never be reconstructed since the tracking system only reads charged-particle deposits). Finally, the particles in each event are p_T-ordered, from hardest to softest.
To improve momentum reconstruction and lessen tracker inefficiencies, we follow Dreyer et al. (2026) and require $p_T > 1$ GeV for *PF* particles, and apply $p_T > 0.25$ GeV cut on the *TR* set to reduce its cardinality.

The training dataset is made of a mixture of $t\bar{t}$ and QCD samples, available with high statistics. The model is subsequently tested on a mixture of statistically identical held-out samples, and a series of out-of-distribution samples of physical processes not seen during training (CMS Collaboration, 2016a;b;c;d;e). An overview is available in Table 1.

2.2 RQ-VAE

Tokenizer networks such as vector-quantized autoencoders (Oord et al., 2018) (VQ-VAEs) compress an input into a discrete latent vector (code). An encoder first maps an input x to a continuous latent

Table 1: Datasets used for training and testing, in-distribution testing slices sum up to $50,000$ events. Parton-level p_T ranges are shown for QCD samples.

Dataset	$p_{\mathrm{T}}^{\min}$–$p_{\mathrm{T}}^{\max}$ **[GeV]**	**Type**	**Training [evts]**	**Testing [evts]**
$H \to 4\ell$	–	Out-of-distribution		50,000
$t\bar{t}$	–	In distribution	9,361,338	28,263
QCD	470–600	In distribution	3,737,811	11,285
QCD	600–800	In distribution	3,462,168	10,452
QCD	1000–1400	Out-of-distribution		50,000

z, which is then quantized against a learned codebook $\mathcal{C} = \{e_k\}_{k=1}^{K}$ by nearest-neighbor lookup, so that x is represented by a single index $k = \mathcal{Q}(z; \mathcal{C})$ and decoded from e_k:

$$\mathcal{Q}(z; \mathcal{C}) = \underset{k \in \{1,\ldots,K\}}{\arg\min} ||z - e_k||_2, \tag{1}$$

The fidelity of this representation is limited by the codebook size K: driving down the quantization error requires a finer partition of the latent space. In practice, large codebooks are difficult to train and suffer from codebook collapse, in which a small fraction of entries are ever selected and the effective capacity saturates below K.

By representing z with an ordered sequence of M codes, residual quantization (Zeghidour et al., 2021) avoids this tradeoff. The effective codebook size increases to K^M while avoiding having to scale K as much. The latent is encoded by quantizing the running residual: starting from $r_0 = z$, at each depth $m = 1, \ldots, M$ we select the code that best approximates the current residual vector and subtract it. In other words, at each depth m:

$$k_m = \mathcal{Q}(r_{m-1}; \mathcal{C}_m), \qquad r_m \;=\; r_{m-1} - e^m_{k_m}, \tag{2}$$

where $\mathcal{C}_m = \{e^m_k\}_{k=1}^{K}$ is the codebook at such depth. The input is thus mapped to the sequence of indices $(k_1, \ldots, k_M)$, and the quantized latent is recovered as the cumulative sum $\hat{z} = \sum_{m=1}^{M} e^m_{k_m}$. The reconstructed input $\hat{x} = G(\hat{z})$ is produced by a decoder G. The tokenizer consists of a self-attention transformer encoder E (Vaswani et al., 2023), the residual-quantization module, and a transformer decoder G, and is trained end-to-end using the reconstruction objective. Further details on architecture and training methods are available in the Appendix A.2.

Given the frozen tokenizer, each particle is mapped to its ordered sequence of M residual codes such that an event of N particles is flattened into a sequence of NM tokens, event generation thus becomes a sequence-modeling problem.

2.3 Autoregressive Generative Model

The core model is a decoder-only transformer with causal self-attention and gated feed-forward networks (Shazeer, 2020). A single linear head maps the final hidden state at each position to a categorical distribution over the code vocabulary. Further details on architecture specifications and training procedure are reported in Appendix B.

We model the distribution over *PFlow* tokens autoregressively with the transformer, conditioned on the *Truth*-level event: both *TR* and *PF* events are tokenized by the same frozen RQ-VAE. The model's input is the concatenation of the tokenized *TR* event and the tokenized *PF* event, with each section following its own `[SOS]` token; the *Truth* block thus acts as a prefix on which the target block is conditioned. The embedding of each token is the sum of three learned components: a shared embedding over the vocabulary of code indices, a depth embedding indexed by the residual level $m \in \{1, \ldots, M\}$, and a learnable linear projection of the corresponding codebook vector e^m_k itself.

Since events have variable particle multiplicity, the number of reconstructed particles N_{PF} is predicted directly from the pooled truth prefix by a feed forward network that is trained to predict logits for the different cardinality amounts. This target multiplicity is supplied to the model through an additional learned "countdown" embedding that encodes the number of particles still to be generated

at each position. The model is trained on next-token-prediction and cardinality objectives using a composite Cross-Entropy Loss function:

$$\mathcal{L} = \mathcal{L}^{\text{CE}} + \lambda_{\text{card}}\, \mathcal{L}^{\text{CE}}_{\text{card}} \quad , \quad \lambda_{\text{card}} = 0.1 \tag{3}$$

After training, to generate an event we tokenize the truth input, prefill it as the prefix, and sample the reconstructed multiplicity N_{PF} from the cardinality head's output distribution. We then autoregressively sample exactly $N_{\text{PF}} \cdot M$ tokens from the categorical distribution. The resulting token sequence is passed through the frozen RQ-VAE decoder to recover the particle-level quantities. Training and inference details are provided in Appendix B.

3 Results

We evaluate the reconstruction and generation performance on particle, event, and jet-level quantities. Jets are reconstructed from collimated sprays of particles by clustering nearby constituents in the $\eta - \phi$ plane into composite objects commonly used for downstream analysis in collider experiments.

We use both the held-out in-distribution test set (QCD + $t\bar{t}$) and out-of-distribution datasets ($H \to 4\ell$ and QCD 1000) to test the model's ability to generalize beyond physical processes seen during training. We examine feature distributions, and we also match generated (*FastSim*, *FS*) and *PFlow* particles and jets with *Truth* ones, using the Hungarian matching algorithm (Kuhn, 1955) with $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ as the cost function, and then evaluate the per-feature residual distributions.

The choice of evaluating on the residual sets is intentional: the conditional generation objective is to model the stochastic detector response and reconstruction. *FastSim* samples should be evaluated in terms of how well they reproduce the full simulation and reconstruction pipeline's response to the incoming detector-stable particles, rather than their ability to deterministically reproduce the exact *PFlow* features.

3.1 Tokenizer fidelity and codebook utilization

Achieving good performance on the reconstruction objective is a required test of the fidelity of the discrete representation learned by the tokenizer network. The latent is quantized with $M = 4$ residual stages (Eq. 2), and each codebook layer contains $K = 8192$ 16-dimensional vectors, more details can be found in Appendix A.2. For each continuous feature, the joint distribution, over both *TR* and *PF*-level events in the test dataset, of the original and reconstructed values is shown in Figure 1, together with the particle class confusion matrix.

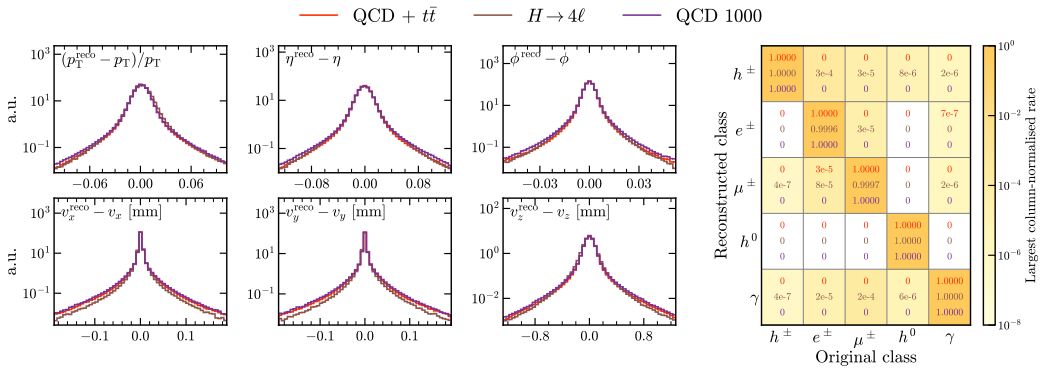


Figure 1: Reconstructed features residual distribution, and particle-class confusion matrix. Vertex residuals are only computed for charged particles.

Next we examine how the available discrete representation capacity is utilized across the residual-quantization levels. Figure 9 (Appendix D) shows the code distribution at each depth m. For each codebook, we also characterize its usage (i.e. the percentage of codes used at least once) and perplexity; metrics are reported in Table 2.

Table 2: Codebook usage and perplexity (PPL) at each residual depth m, for the three test datasets.

	$m=1$		$m=2$		$m=3$		$m=4$	
Dataset	Usage	Perplexity	Usage	Perplexity	Usage	Perplexity	Usage	Perplexity
QCD + $t\bar{t}$	58.5%	4042	99.9%	5575	99.9%	4894	99.9%	4723
$H \to 4\ell$	58.2%	3338	99.3%	5382	99.4%	4797	99.6%	4662
QCD 1000	58.5%	3897	99.9%	5652	100.0%	4981	100.0%	4871

3.2 Generation Performance

We show the performance of the largest model configuration, trained for four epochs on the full dataset. A performance comparison to other ablations is shown in Table 10, in the Appendix D.

Particle-level Residual distributions of particle-level features, for *PF* and *FS* particles matched to *TR*, are reported in Figure 2. The generated residual distributions closely track the corresponding *PF* distributions. The η and ϕ marginals in Figure 10 (Appendix D) also reproduce the detector-dependent structure across the tested datasets.

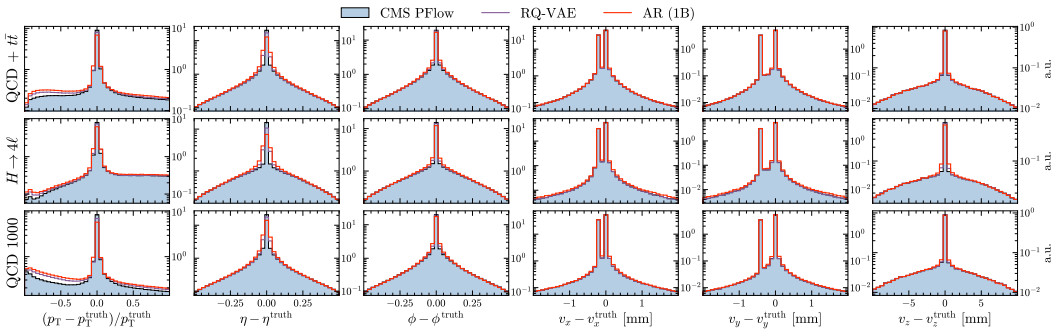


Figure 2: Spectra of particle residual features for all three testing datasets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

Event-level Event-level variables are a relevant test of the model's ability to capture global event correlations, in particular in an autoregressive setting where the bulk of the generated structure follows from a subset of initially generated hard particles. For event-level metrics, we consider number of particles, missing transverse energy in the x and y directions and the scalar sum of particle p_T. Residual distributions of these metrics are shown in Figure 3, and marginals are displayed in the appendix (Figure 12).

Jet-level We examine the kinematic properties of the clustered jets (p_T, η, ϕ) and the jet substructure quantities C_2 and D_2 (Larkoski et al., 2013; 2014). While inclusive marginal distributions test the overall kinematic phase-space coverage of the generated *FS* samples against target *PF*, residual distributions with respect to truth-matched jets evaluate the conditional response and resolution. In addition, the substructure observables test the effective modeling of intra-jet particle correlations and detector effects. Residual distributions for the selected variables are shown in Figure 4, and marginals in the appendix (Figure 11).

3.3 Scaling Study

We study how the generator model's performance changes with model size, training-set size, and repeated data exposure. We evaluate the model on the token cross-entropy loss on the statistically identical set of held-out events and fit its dependence on these training variables. We then evaluate how scaling affects the generated distributions of physical observables.
A full overview of training ablations used for the study is available in Tables 3 and 4.

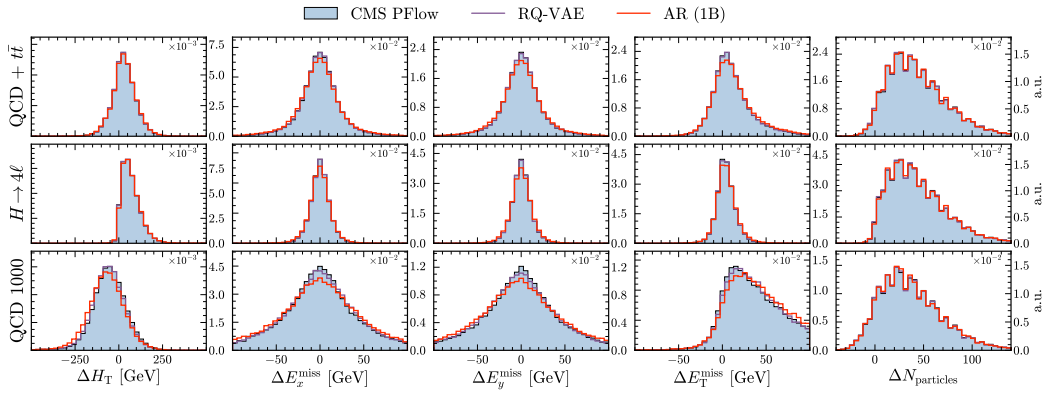


Figure 3: Spectra of event-level residual features for all three testing datasets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

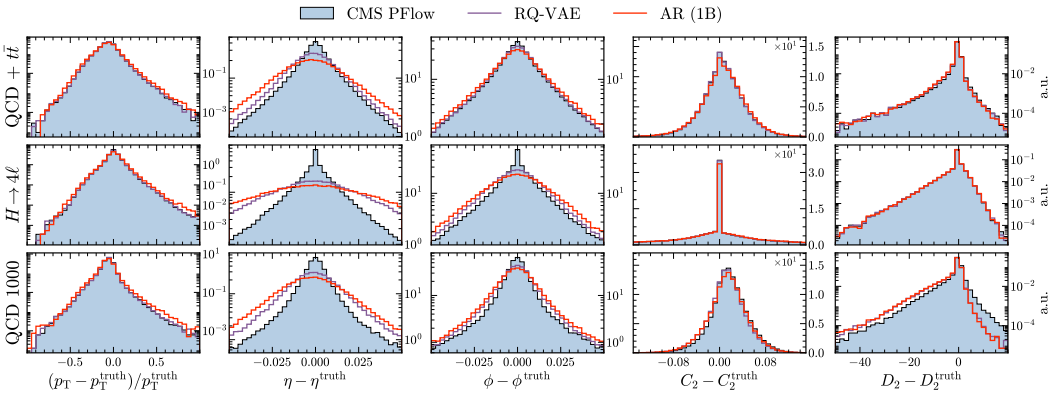


Figure 4: Spectra of residuals of jet-level features for all three testing datasets. The residuals are computed against matched truth jets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

3.3.1 Loss scaling

The fitted quantity is the held-out next-token-prediction (NTP) cross-entropy $\mathcal{L}$, in nats. This is the token component of the objective defined in Section 2.3; the cardinality term is excluded. First, we ignore data repetition and only consider fresh-data runs in which each event is only shown to the model once at most. In this regime, we fit the additive Chinchilla (Hoffmann et al., 2022) and coupled Skaling (Videau et al., 2026) forms that model the data-constraint free scaling behavior of the model family.

In settings such as detector simulation, generating MC training datasets can be an expensive endeavor. It is important to model the effects of data repetition on ML-based surrogates, and maximize the usefulness of the available synthetic datasets. To evaluate the effects of repeated data exposure, we also fit the "Prescriptive" repetition penalty introduced by Lovelace et al. (2026), and a custom combined form that applies this penalty to the Skaling coupled expression. We fit the measured losses by unweighted least squares. Table 5 reports the fitted α, β parameters and the exponents for compute-optimal allocation $a = \beta/(\alpha + \beta)$ and $b = \alpha/(\alpha + \beta)$ (Hoffmann et al., 2022), together with mean absolute percentage error (MAPE) and coefficient of determination R^2, to indicate goodness-of-fit. Although the fitted parameters vary, all models predict similar values for the a and b optimal-scaling parameters. The full parameters are reported in Table 9, Appendix C.1.

Figure 5 compares the measured held-out losses with the fitted curves and the implied compute-optimal scaling of N. Larger models overfit sooner on smaller datasets but achieve lower loss when

Table 3: Model configurations used in the scaling study. Each model is trained with eight data and repetition settings, giving 48 configurations.

Name	d_{model}	Layers	Heads	d_{head}	d_{ff}	Total Params
10M	256	8	8	32	512	9.83M
22M	384	10	8	48	768	21.67M
41M	512	12	8	64	1,024	40.75M
97M	768	14	12	64	1,536	96.71M
187M	1,024	16	16	64	2,048	186.87M
1B	2,048	24	32	64	4,096	1,046.92M

Table 4: Data and repetition settings used in the scaling study. Each setting is applied to all six model sizes. Data fractions are subsampled from the full dataset, in order to preserve the same statistical properties.

Data fraction	Unique events	Epochs	PF tokens	Total tokens
3.3%	546,523	1, 10, 30	324M	749M
33.3%	5,514,918	1, 3, 12	3.27B	7.56B
100%	16,561,317	1, 4	9.82B	22.7B

sufficient unique data are available. Over the compute range covered by the single-epoch runs, the fitted compute-optimal model size remains below the largest models in our grid, indicating that the study is data-constrained. The isoFLOP-fitted analysis yields $N \propto C^{0.53}$.

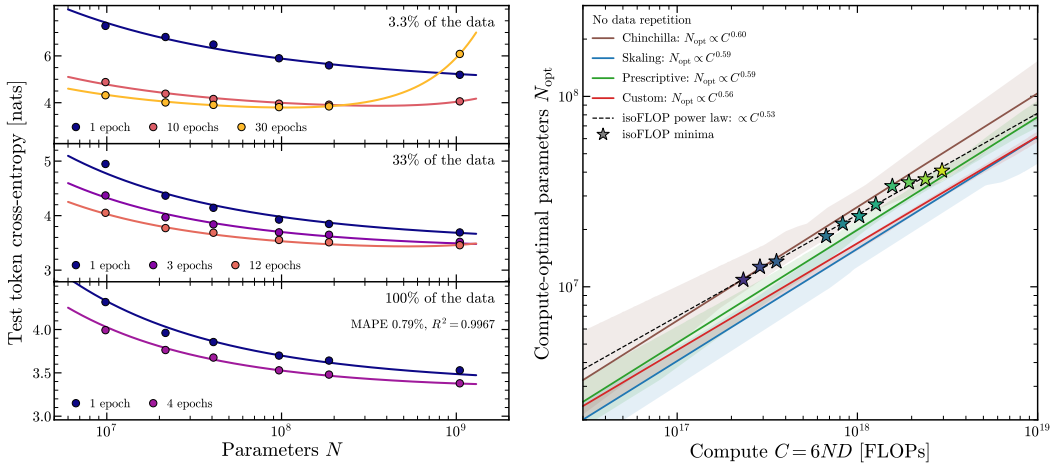


Figure 5: Left: each trained model's NTP loss against its parameter count. The lines show the custom fitted law's prediction for the given combination of dataset size and repetition. Right: compute-guided fit-predicted optimal scaling behaviors and isoFLOP curves minima. The methodology for recovering these minima is discussed in the Appendix C.2.

3.3.2 Scaling of physical metrics

It is not immediately obvious how modeling of the decoded physical variables correlates with a lower value of the NTP Loss. The model might produce token sequences that appear satisfactory on a surface level but produce nonsensical physical results once decoded. For this reason it is important to choose metrics that test both local and global features of the generated samples.

Table 5: Fits of the neural scaling laws. Chinchilla and Skaling are fitted to the 18 single-epoch runs, the repetition-aware laws to all 48 runs. Uncertainties are 1σ errors over refits with one run held-out.

Law	Functional form	α	β	a	b	MAPE [%]	R^2
Chinchilla	$L_\infty + AN^{-\alpha} + BU^{-\beta}$	0.42 ± 0.18	0.62 ± 0.15	0.60 ± 0.14	0.40 ± 0.14	3.55	0.9695
Skaling	$L_\infty + \left(AN^{-\alpha} + BU^{-\beta}\right)^k$	0.241 ± 0.052	0.345 ± 0.070	0.589 ± 0.048	0.411 ± 0.048	1.17	0.9961
Prescriptive	$L_\infty + AN^{-\alpha} + B[U(1+R)]^{-\beta} + PR^\delta \left(\frac{N}{U^\gamma}\right)^\kappa$	0.410 ± 0.092	0.598 ± 0.036	0.593 ± 0.074	0.407 ± 0.074	2.47	0.9690
Custom	$L_\infty + \left(AN^{-\alpha} + B[U(1+R)]^{-\beta}\right)^k + PR^\delta \left(\frac{N}{U^\gamma}\right)^\kappa$	0.273 ± 0.017	0.349 ± 0.016	0.561 ± 0.015	0.439 ± 0.015	0.79	0.9967

Global point-cloud metrics First, we adopt a procedure similar to that of Amram et al. (2026) by comparing the NTP loss with the Sliced Wasserstein Distance (SWD) (Rabin et al., 2011; Bonneel et al., 2015) between *FS* and *PF* samples, but we instead evaluate on the particle-, event-, and jet-level **residual** feature manifolds. For each feature group, we subsample the point clouds and average the W_1 Wasserstein Distance over 256 random projections after standardising every feature with the mean and standard deviation of the *PF* reference. We also include the statistical noise barrier obtained by computing the SWD between statistically identical samples.
In Figure 6 we plot the NTP loss for given checkpoints against the SWD on the given feature sets of the associated produced sample from the testing datasets. We find, by fitting a power law of the type:

$$\text{SWD} = a(\mathcal{L} - \mathcal{L}_0)^b + \text{SWD}_f \tag{4}$$

that the loss achieved on the held-out test dataset is a very strong performance predictor on this global metric. Fit results and a visualization are available in Figure 6. Details on the full procedure are available in Appendix C.3.

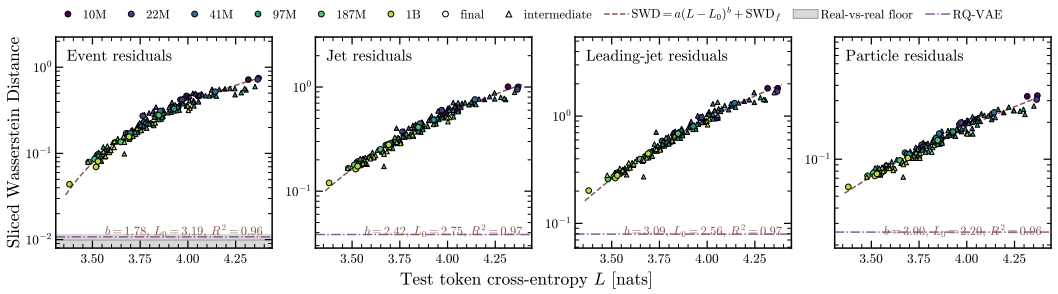


Figure 6: Measured Sliced Wasserstein Distance for all valid (selection criteria in Appendix C.3) runs, and fitted law in the form of Equation 4. We include both final and intermediate checkpoints and show the fitted power law parameters for each feature.

Individual metrics Observing scaling behavior on distributions of the individual output features is not a trivial task: modeling the marginal distributions of these is often easier and can result in differences that fall within two-sample noise. We find, however, that each one of the individual residual feature distributions exhibits scaling behavior, in the same form as Eq. 4, as the global feature manifolds, and that the achieved loss value is also a strong individual predictor of the fidelity of these generated physical metrics. In Figure 7 we check the one-dimensional W_1 Wasserstein Distance between the *FastSim* and *PFlow* residual distributions, and show the fit results across the different features. We find that improvements in achieved loss transfer to the individual features.

4 Conclusion

We apply discrete autoregressive modeling to conditional full-event simulation, demonstrating how residual-quantized representations of point-clouds enable generation of particle-level detector responses. By employing a frozen RQ-VAE tokenizer with an autoregressive transformer, we recover

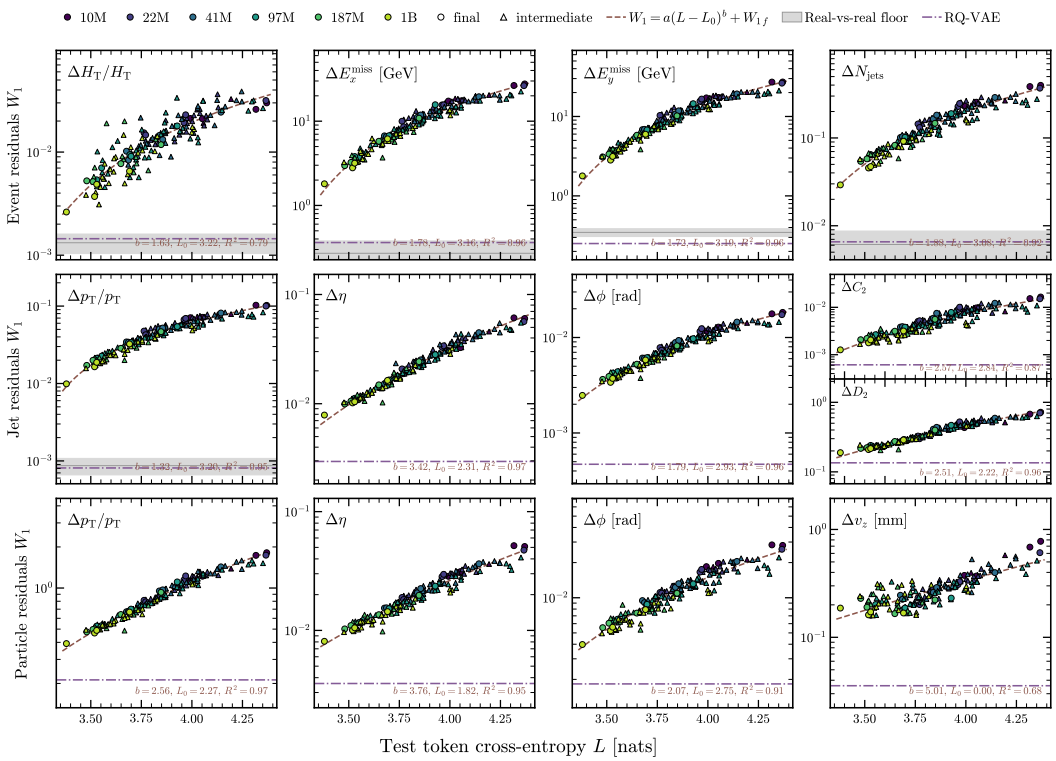


Figure 7: Per-feature W_1 Wasserstein Distance between the *PF* and *FS* valid (selection criteria in Appendix C.3) residual distributions. We include both final and intermediate checkpoints and show the fitted power law parameters for each feature.

continuous particle-level outputs from discrete sequences while preserving the flexibility of full-event generation. We demonstrate the fidelity of the learned representation and evaluate the generated events across particle, event, and jet-level observables, including on physical processes not encountered during training.

We then study the scaling behavior of this model family across a range of model and dataset sizes, find that the next-token prediction loss exhibits familiar scaling characteristics, and that the effects of data repetition can be captured by a repetition-aware scaling law.

Our empirical results demonstrate a direct link between token-level modeling performance and the fidelity of generated physical observables. We show that the achieved next-token prediction loss is a strong predictor of both global distributional distances and individual metrics, with their relationship well described by a simple power law. These findings suggest that token-level scaling laws may provide a useful proxy for estimating downstream physical performance, validate residual-quantized autoregressive architectures for scalable simulation in high-energy physics and demonstrate how empirical scaling laws can guide compute allocation for ML surrogate models.

REPRODUCIBILITY STATEMENT

Full architectural specifications, data preprocessing pipeline, and experimental setups are detailed in Appendices A and B, along with exact hyperparameter choices and training configurations. All analyzed datasets are publicly accessible. The processed data artifacts, training scripts, and model checkpoints will be made publicly available upon paper publication.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, design experiments, clean and reformat datasets and refine hypotheses. We have not used generative AI tools to generate synthetic data sets, interpret results, help develop theoretical models or conceptual frameworks; and assisting with translation, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, edit software code. We have reviewed all AI-assisted work. LLM-generated code was verified for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

G. Aad, E. Aakvaag, B. Abbott, S. Abdelhameed, K. Abeling, N. J. Abicht, S. H. Abidi, M. Aboelela, A. Aboulhorma, H. Abramowicz, Y. Abulaiti, B. S. Acharya, A. Ackermann, C. Adam Bourdarios, L. Adamczyk, S. V. Addepalli, M. J. Addison, J. Adelman, A. Adiguzel, T. Adye, A. A. Affolder, Y. Afik, M. N. Agaras, A. Aggarwal, C. Agheorghiesei, F. Ahmadov, S. Ahuja, X. Ai, G. Aielli, A. Aikot, M. Ait Tamlihat, B. Aitbenchikh, M. Akbiyik, T. P. A. Åkesson, A. V. Akimov, D. Akiyama, N. N. Akolkar, S. Aktas, G. L. Alberghi, J. Albert, U. Alberti, P. Albicocco, G. L. Albouy, S. Alderweireldt, Z. L. Alegria, M. Aleksa, I. N. Aleksandrov, C. Alexa, T. Alexopoulos, F. Alfonsi, M. Algren, M. Alhroob, B. Ali, H. M. J. Ali, S. Ali, S. W. Alibocus, M. Aliev, G. Alimonti, W. Alkakhi, C. Allaire, B. M. M. Allbrooke, J. S. Allen, J. F. Allen, P. P. Allport, A. Aloisio, F. Alonso, C. Alpigiani, Z. M. K. Alsolami, A. Alvarez Fernandez, M. Alves Cardoso, M. G. Alviggi, M. Aly, Y. Amaral Coutinho, A. Ambler, C. Amelung, M. Amerl, C. G. Ames, T. Amezza, D. Amidei, B. Amini, K. Amirie, A. Amirkhanov, S. P. Amor Dos Santos, K. R. Amos, D. Amperiadou, S. An, C. Anastopoulos, T. Andeen, J. K. Anders, A. C. Anderson, A. Andreazza, S. Angelidakis, A. Angerami, A. V. Anisenkov, A. Annovi, C. Antel, E. Antipov, M. Antonelli, F. Anulli, M. Aoki, T. Aoki, M. A. Aparo, L. Aperio Bella, M. Apicella, C. Appelt, A. Apyan, M. Arampatzi, S. J. Arbiol Val, C. Arcangeletti, A. T. H. Arce, J-F. Arguin, S. Argyropoulos, J.-H. Arling, O. Arnaez, H. Arnold, G. Artoni, H. Asada, K. Asai, S. Asai, S. Asatryan, N. A. Asbah, R. A. Ashby Pickering, A. M. Aslam, K. Assamagan, R. Astalos, K. S. V. Astrand, S. Atashi, R. J. Atkin, H. Atmani, P. A. Atmasiddha, K. Augsten, A. D. Auriol, V. A. Austrup, G. Avolio, K. Axiotis, A. Azzam, D. Babal, H. Bachacou, K. Bachas, A. Bachiu, E. Bachmann, M. J. Backes, A. Badea, T. M. Baer, P. Bagnaia, M. Bahmani, D. Bahner, K. Bai, J. T. Baines, L. Baines, O. K. Baker, E. Bakos, D. Bakshi Gupta, L. E. Balabram Filho, V. Balakrishnan, R. Balasubramanian, E. M. Baldin, P. Balek, E. Ballabene, F. Balli, L. M. Baltes, W. K. Balunas, J. Balz, I. Bamwidhi, E. Banas, M. Bandieramonte, A. Bandyopadhyay, S. Bansal, L. Barak, M. Barakat, E. L. Barberio, D. Barberis, M. Barbero, M. Z. Barel, T. Barillari, M-S. Barisits, T. Barklow, P. Baron, D. A. Baron Moreno, A. Baroncelli, A. J. Barr, J. D. Barr, F. Barreiro, J. Barreiro Guimarães da Costa, M. G. Barros Teixeira, S. Barsov, F. Bartels, R. Bartoldus, A. E. Barton, P. Bartos, A. Basan, M. Baselga, S. Bashiri, A. Bassalat, M. J. Basso, S. Bataju, R. Bate, R. L. Bates, S. Batlamous, M. Battaglia, D. Battulga, M. Bauce, M. Bauer, P. Bauer, L. T. Bayer, L. T. Bazzano Hurrell, J. B. Beacham, T. Beau, J. Y. Beaucamp, P. H. Beauchemin, P. Bechtle, H. P. Beck, K. Becker, A. J. Beddall, V. A. Bednyakov, C. P. Bee, L. J. Beemster, M. Begalli, M. Begel, J. K. Behr, J. F. Beirer, F. Beisiegel, M. Belfkir, G. Bella, L. Bellagamba, A. Bellerive, C. D. Bellgraph, P. Bellos, K. Beloborodov, D. Benchekroun, F. Bendebba, Y. Benhammou, K. C. Benkendorfer, L. Beresford, M. Beretta, E. Bergeaas Kuutmann, N. Berger, B. Bergmann, J. Beringer, G. Bernardi, C. Bernius, F. U. Bernlochner, F. Bernon, A. Berrocal Guardia, T. Berry, P. Berta, A. Berthold, A. Berti, R. Bertrand, S. Bethke, A. Betti, A. J. Bevan, L. Bezio, N. K. Bhalla, S. Bharthuar, S. Bhatta, P. Bhattarai, Z. M. Bhatti, K. D. Bhide, V. S. Bhopatkar, R. M. Bianchi, G. Bianco, O. Biebel, M. Biglietti, C. S. Billingsley, Y. Bimgdi, M. Bindi, A. Bingham, A. Bingul, C. Bini,

G. A. Bird, M. Birman, M. Biros, S. Biryukov, T. Bisanz, E. Bisceglie, J. P. Biswal, D. Biswas, I. Bloch, A. Blue, U. Blumenschein, J. Blumenthal, V. S. Bobrovnikov, L. Boccardo, M. Boehler, B. Boehm, D. Bogavac, A. G. Bogdanchikov, L. S. Boggia, V. Boisvert, P. Bokan, T. Bold, M. Bomben, M. Bona, M. Boonekamp, A. G. Borbély, I. S. Bordulev, G. Borissov, D. Bortoletto, and The ATLAS Collaboration. Transforming jet flavour tagging at ATLAS. *Nature Communications*, 17(1):541, January 2026. ISSN 2041-1723. doi: 10.1038/s41467-025-65059-6. URL https://www.nature.com/articles/s41467-025-65059-6.

S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J. J. Gómez Cadenas, I. González, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F. W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampén, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M. G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J. P. Wellisch, T. Wenaus, D. C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, July 2003. ISSN 0168-9002. doi: 10.1016/S0168-9002(03)01368-8. URL https://www.sciencedirect.com/science/article/pii/S0168900203013688.

Oz Amram, Luca Anzalone, Joschka Birk, Darius A. Faroughy, Anna Hallin, Gregor Kasieczka, Michael Krämer, Ian Pang, Humberto Reyes-Gonzalez, and David Shih. Aspen Open Jets: Unlocking LHC Data for Foundation Models in Particle Physics, December 2024. URL http://arxiv.org/abs/2412.10504. arXiv:2412.10504 [hep-ph].

Oz Amram, Darius A. Faroughy, Tjarko Gerdes, Anna Hallin, Gregor Kasieczka, Michael Krämer, Humberto Reyes-Gonzalez, and David Shih. Neural scaling laws for jet generation, 2026. URL https://arxiv.org/abs/2605.28940.

ATLAS Collaboration. ATLAS Software and Computing HL-LHC Roadmap. Technical report, CERN, Geneva, 2022. URL https://cds.cern.ch/record/2802918.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization, July 2016. URL http://arxiv.org/abs/1607.06450. arXiv:1607.06450 [stat.ML].

Wahid Bhimji, Chris Harris, Vinicius Mikuni, and Benjamin Nachman. OmniLearned: A Foundation Model Framework for All Tasks Involving Jet Physics. *Physical Review D*, 113(3): 032020, February 2026. ISSN 2470-0010, 2470-0029. doi: 10.1103/knmd-f5jm. URL http://arxiv.org/abs/2510.24066. arXiv:2510.24066 [hep-ph].

Joschka Birk, Anna Hallin, and Gregor Kasieczka. OmniJet-$$: The first cross-task foundation model for particle physics. *Machine Learning: Science and Technology*, 5(3):035031, September 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad66ad. URL http://arxiv.org/abs/2403.05618. arXiv:2403.05618 [hep-ph].

Joschka Birk, Frank Gaede, Anna Hallin, Gregor Kasieczka, Martina Mozzanica, and Henning Rose. OmniJet-$_c$: Learning point cloud calorimeter simulations using generative transformers, June 2025. URL http://arxiv.org/abs/2501.05534. arXiv:2501.05534 [hep-ph].

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 1(51):22–45, 2015. doi: 10.1007/s10851-014-0506-3. URL https://hal.science/hal-00881872.

Anja Butter, Tilman Plehn, Steffen Schumann, Simon Badger, Sascha Caron, Kyle Cranmer, Francesco Armando Di Bello, Etienne Dreyer, Stefano Forte, Sanmay Ganguly, Dorival Gonçalves, Eilam Gross, Theo Heimel, Gudrun Heinrich, Lukas Heinrich, Alexander Held, Stefan Höche, Jessica N. Howard, Philip Ilten, Joshua Isaacson, Timo Janßen, Stephen Jones, Marumi Kado, Michael Kagan, Gregor Kasieczka, Felix Kling, Sabine Kraml, Claudius Krause, Frank Krauss, Kevin Kröninger, Rahool Kumar Barman, Michel Luchmann, Vitaly Magerya, Daniel Maitre, Bogdan Malaescu, Fabio Maltoni, Till Martini, Olivier Mattelaer, Benjamin Nachman, Sebastian Pitz, Juan Rojo, Matthew Schwartz, David Shih, Frank Siegert, Roy Stegeman, Bob Stienen, Jesse Thaler, Rob Verheyen, Daniel Whiteson, Ramon Winterhalder, and Jure Zupan. Machine Learning and LHC Event Generation. *SciPost Physics*, 14(4):079, April 2023. ISSN 2542-4653. doi: 10.21468/SciPostPhys.14.4.079. URL http://arxiv.org/abs/2203.07460. arXiv:2203.07460 [hep-ph].

Carlos Cardona-Giraldo, Cristiano Fanelli, James Giroux, Cole Granger, Benjamin Nachman, and Gerald Sabin. Generalizable Foundation Models for Calorimetry via Mixtures-of-Experts and Parameter Efficient Fine Tuning, March 2026. URL http://arxiv.org/abs/2603.28804. arXiv:2603.28804 [physics.ins-det].

CMS Collaboration. Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET. Technical report, CERN, Geneva, 2009. URL https://cds.cern.ch/record/1194487.

CMS Collaboration. Simulated dataset GluGluToHToZZTo4L_M-125_7TeV-minloHJJ-pythia6-tauola in AODSIM format for 2011 collision data (SM Higgs), 2016a. URL http://opendata.cern.ch/record/1443.

CMS Collaboration. Simulated dataset QCD_Pt-1000to1400_TuneZ2_7TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), 2016b. URL http://opendata.cern.ch/record/1342.

CMS Collaboration. Simulated dataset QCD_Pt-470to600_TuneZ2_7TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), 2016c. URL http://opendata.cern.ch/record/1553.

CMS Collaboration. Simulated dataset QCD_Pt-600to800_TuneZ2_7TeV_pythia6 in AODSIM format for 2011 collision data (SM Exclusive), 2016d. URL http://opendata.cern.ch/record/1558.

CMS Collaboration. Simulated dataset TTJets_MSDecays_central_TuneZ2_7TeV-madgraph-tauola in AODSIM format for 2011 collision data (SM Inclusive), 2016e. URL http://opendata.cern.ch/record/1662.

CMS Collaboration. CMS Offline Software and Computing for HL-LHC Conceptual Design Report. Technical report, CERN, Geneva, 2026. URL https://cds.cern.ch/record/2957472.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić,

Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling Vision Transformers to 22 Billion Parameters, February 2023. URL http://arxiv.org/abs/2302.05442. arXiv:2302.05442 [cs].

Etienne Dreyer, Eilam Gross, Dmitrii Kobylianskii, Vinicius Mikuni, and Benjamin Nachman. Conditional deep generative models for simultaneous simulation and reconstruction of entire events. *Physical Review D*, 113(3):032005, February 2026. doi: 10.1103/14ph-482n. URL https://link.aps.org/doi/10.1103/14ph-482n.

Christopher Fifty, Ronald G. Junkins, Dennis Duan, Aniketh Iyengar, Jerry W. Liu, Ehsan Amid, Sebastian Thrun, and Christopher Ré. Restructuring Vector Quantization with the Rotation Trick, October 2024. URL https://arxiv.org/abs/2410.06424v2.

Tobias Golling, Lukas Heinrich, Michael Kagan, Samuel Klein, Matthew Leigh, Margarita Osadchy, and John Andrew Raine. Masked Particle Modeling on Sets: Towards Self-Supervised High Energy Physics Foundation Models. *Machine Learning: Science and Technology*, 5(3):035074, September 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad64a8. URL http://arxiv.org/abs/2401.13537. arXiv:2401.13537 [hep-ph].

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Claudius Krause, Michele Faucci Giannelli, Gregor Kasieczka, Benjamin Nachman, Dalila Salamani, David Shih, Anna Zaborowska, Oz Amram, Kerstin Borras, Matthew R. Buckley, Erik Buhmann, Thorsten Buss, Renato Paulo Da Costa Cardoso, Anthony L. Caterini, Nadezda Chernyavskaya, Federico A. G. Corchia, Jesse C. Cresswell, Sascha Diefenbacher, Etienne Dreyer, Vijay Ekambaram, Engin Eren, Florian Ernst, Luigi Favaro, Matteo Franchini, Frank Gaede, Eilam Gross, Shih-Chieh Hsu, Kristina Jaruskova, Benno Käch, Jayant Kalagnanam, Raghav Kansal, Taewoo Kim, Dmitrii Kobylianskii, Anatolii Korol, William Korcari, Dirk Krücker, Katja Krüger, Marco Letizia, Shu Li, Qibin Liu, Xiulong Liu, Gabriel Loaiza-Ganem, Thandikire Madula, Peter McKeown, Isabell-A. Melzer-Pellmann, Vinicius Mikuni, Nam Nguyen, Ayodele Ore, Sofia Palacios Schweitzer, Ian Pang, Kevin Pedro, Tilman Plehn, Witold Pokorski, Huilin Qu, Piyush Raikwar, John A. Raine, Humberto Reyes-Gonzalez, Lorenzo Rinaldi, Brendan Leigh Ross, Moritz A. W. Scham, Simon Schnake, Chase Shimmin, Eli Shlizerman, Nathalie Soybelman, Mudhakar Srivatsa, Kalliopi Tsolaki, Sofia Vallecorsa, Kyongmin Yeo, and Rui Zhang. CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation. *Reports on Progress in Physics*, 88(11):116201, November 2025. ISSN 0034-4885, 1361-6633. doi: 10.1088/1361-6633/ae1304. URL http://arxiv.org/abs/2410.21611. arXiv:2410.21611 [physics.ins-det].

Harold. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2:83–97, 1955.

Andrew J. Larkoski, Gavin P. Salam, and Jesse Thaler. Energy correlation functions for jet substructure. *Journal of High Energy Physics*, 2013(6):108, June 2013. ISSN 1029-8479. doi: 10.1007/JHEP06(2013)108. URL https://link.springer.com/10.1007/JHEP06(2013)108. arXiv:1305.0007.

Andrew J. Larkoski, Ian Moult, and Duff Neill. Power Counting to Better Jet Observables. *Journal of High Energy Physics*, 2014(12):9, December 2014. ISSN 1029-8479. doi: 10.1007/JHEP12(2014)009. URL http://arxiv.org/abs/1409.6298. arXiv:1409.6298 [hep-ph].

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs.LG].

Justin Lovelace, Christian Belardi, Srivatsa Kundurthy, Shriya Sudhakar, and Kilian Q. Weinberger. Prescriptive scaling laws for data constrained training, 2026. URL https://arxiv.org/abs/2605.01640.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning, May 2018. URL http://arxiv.org/abs/1711.00937. arXiv:1711.00937 [cs.LG].

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein Barycenter and its Application to Texture Mixing. In *Proc. SSVM'11*, pp. 435–446, Israel, 2011. Springer. URL https://hal.science/hal-00476064.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2, June 2019. URL http://arxiv.org/abs/1906.00446. arXiv:1906.00446 [cs.LG].

Noam Shazeer. GLU Variants Improve Transformer, February 2020. URL http://arxiv.org/abs/2002.05202. arXiv:2002.05202 [cs.LG].

Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A brief introduction to PYTHIA 8.1. *Computer Physics Communications*, 178(11):852–867, June 2008. ISSN 0010-4655. doi: 10.1016/j.cpc.2008.01.036. URL https://www.sciencedirect.com/science/article/pii/S0010465508000441.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, November 2023. URL http://arxiv.org/abs/2104.09864. arXiv:2104.09864 [cs.CL].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs.CL].

Mathurin Videau, Badr Youbi-Idrissi, David Lopez-Paz, and Kartik Ahuja. Skaling: Chinchilla's exponents meet kaplan's coupling, 2026. URL https://arxiv.org/abs/2608.07222.

Matthias Vigl, Nicole Hartman, Michael Kagan, and Lukas Heinrich. Neural scaling laws for boosted jet tagging, 2026. URL https://arxiv.org/abs/2602.15781.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec, July 2021. URL http://arxiv.org/abs/2107.03312. arXiv:2107.03312 [cs].

Zhijian Zhuo, Yutao Zeng, Ya Wang, Sijun Zhang, Jian Yang, Xiaoqing Li, Xun Zhou, and Jinwen Ma. HybridNorm: Towards Stable and Efficient Transformer Training via Hybrid Normalization, March 2025. URL http://arxiv.org/abs/2503.04598. arXiv:2503.04598 [cs].

A METHODS DETAILS

A.1 INPUT REPRESENTATION

Every input particle is described by a 12-dimensional vector:

$$x = \big(\tilde{p}_T,\ \tilde{\eta},\ \sin\phi,\ \cos\phi,\ \tilde{v}_x,\ \tilde{v}_y,\ \tilde{v}_z,\ \mathrm{onehot}_5(\mathrm{class})\big), \tag{5}$$

where $\tilde{p}_T = (\log(1+p_T)-\mu)/\sigma$ and the pseudorapidity and vertex coordinates are normalized with fixed means and standard deviations computed once on the training sample (Table 6). The vertex of neutral *PF* candidates is set to zero (Section 2). We also apply a pseudorapidity cut of $|\eta| < 2.7$, and during evaluation we restrict the final generated *PF* particles with a $|\eta| < 2.5$ cut to remain inside the tracker's acceptance region. This allows the generator to model how particles close to the tracker's acceptance region affect the detector response and reconstruction of those inside. Particle sequences are p_T sorted and zero-padded to a maximum of 400 particles and a boolean mask marks valid entries.

Table 6: Fixed feature standardization. p_T is transformed as $\log(1+p_T)$ before standardization; ϕ is replaced by $(\sin\phi, \cos\phi)$.

	$\log(1+p_T)$	η	v_x	v_y	v_z
mean μ	1.13	0.00	0.206	0.331	0.34
std. σ	0.83	1.29	0.691	0.702	5.73

A.2 RQ-VAE TOKENIZER

Architecture The tokenizer is a set-to-set autoencoder: both the encoder E and the decoder G act on the p_T-ordered set of particles with bidirectional self-attention. The encoder and decoder are mirrored and consist of:

1. a per-particle input MLP $12 \to 32 \to 128$ (ReLU, with biases);
2. a 5-layer transformer at width $d = 128$ with 8 attention heads;
3. a per-particle output MLP $128 \to 32 \to 16$ (ReLU), producing the latent $z \in \mathbb{R}^{16}$.

Each transformer layer is pre-norm, with affine-free LayerNorm (Ba et al., 2016). Queries and keys are RMS-normalized per head with a learnable scale (QK-norm, Dehghani et al., 2023). The feed-forward network is a SwiGLU unit (Shazeer, 2020) with hidden width 256 and biases. Attention is computed with the variable-length FlashAttention-2 kernel (Dao et al., 2022; Dao, 2023) on the unpadded particles of each event. The encoder and decoder parameter count is 0.85M each, 1.70M in total.

Codebooks are initialized by k-means (100 iterations) on the first training batch (Zeghidour et al., 2021) and then updated by exponential moving averages of the assigned residuals (Oord et al., 2018; Razavi et al., 2019) with decay 0.8. Gradients reach the encoder through the quantizer via the rotation trick (Fifty et al., 2024).

Training The loss function is the sum of the mean square error (cross-entropy for classes) between the input and its reconstruction and a small per-codebook commitment loss (Oord et al., 2018).
The tokenizer is trained on the first 16.5 M events of the training dataset. Each event contributes its *Truth* and *PFlow* particle sets, for a total of 33M point-clouds per epoch. We use the AdamW optimizer (Loshchilov & Hutter, 2019) ($\beta = (0.9, 0.999)$, weight decay 0.01), gradient-norm clipping at 1.0, and a cosine-decay-schedule learning rate with a brief warmup. After training, the tokenizer is frozen.

B AUTOREGRESSIVE MODEL

Vocabulary and sequence layout The index space contains $V = K + 3 = 8195$ entries, namely a padding token (0), the code indices $1, \ldots, 8192$, a *PF* start token and a *TR* start token. Code index

k at depth m and the same index at depth $m' \neq m$ map to the same entry, and the depth embedding tells them apart. For an event with N_{TR} *Truth* and N_{PF} *PFlow* particles the input is

$$\big[\,\texttt{SOS}_{\text{TR}},\ t_{1,1},\ldots,t_{1,4},\ \ldots,\ t_{N_{\text{TR}},4},\ \texttt{pad}\ldots\ \big|\ \texttt{SOS}_{\text{PF}},\ p_{1,1},\ldots,p_{1,4},\ \ldots,\ p_{N_{\text{PF}},4},\ \texttt{pad}\ldots\big], \quad (6)$$

where each block is padded to $400 \times 4 = 1600$ code positions, for a total length of 3202. Padding never enters computation and does not receive gradients.

Backbone The backbone model is a decoder-only transformer with L layers of width d_{model} (Table 3), causal self-attention with $d_{\text{head}} = 64$ (except $d_{\text{head}} = 32$ and 48 for the two smallest models), and no biases. Each layer follows the HybridNorm arrangement (Zhuo et al., 2025). Queries, keys and values are each RMS-normalized per head with a learnable scale (QKV-norm) before rotary position embeddings (Su et al., 2023) are applied to queries and keys. Positions are the absolute indices in the padded 3202-long layout, as a result the *PF* block always starts at position 1601, whatever the truth multiplicity. The feed-forward block is SwiGLU with hidden width $d_{\text{ff}} = 2d_{\text{model}}$, and a final LayerNorm follows the last layer.

The NTP head is an untied linear map $\mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^V$ with bias. The cardinality head pools the final hidden states over the truth block by a masked mean. The head then applies $\text{Linear}(d_{\text{model}}, d_{\text{model}}) \to \text{GELU} \to \text{Linear}(d_{\text{model}}, 401)$, returning logits over $N_{\text{PF}} \in \{0, \ldots, 400\}$. Its gradient flows into the shared backbone.

Training Throughout the scaling study, N follows the convention of Hoffmann et al. (2022) and counts all parameters in the model, including embedding tables, it excludes the frozen tokenizer. A more detailed breakdown of the composition of the models in the study is available in Table 7

Table 7: Parameter breakdown of the generator models (in millions). N is the total used for fitting.

Model	Transformer	Code proj.	Token head	Card. head	Embedding tables	N
10M	5.244	0.004	2.106	0.169	2.305	9.827
22M	14.747	0.007	3.155	0.302	3.457	21.668
41M	31.460	0.009	4.204	0.468	4.609	40.750
97M	82.578	0.013	6.302	0.899	6.914	96.706
187M	167.775	0.017	8.400	1.461	9.218	186.871
1B	1006.638	0.035	16.792	5.018	18.436	1046.918

The loss is applied to the *PF* block only: truth-block targets are set to the padding index. During training the countdown embedding uses the true N_{PF}. All generator runs share one recipe, with no per-size tuning. We employ the AdamW optimizer with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-5}$ and weight decay 0.01 applied to all parameters, and a cosine-decay-schedule learning rate with a brief warmup. The decay is tuned to finish at the designed end of the run, and arrives at a tenth of the peak value (Hoffmann et al., 2022). The global batch-size is tuned per run to 256 events, using gradient accumulation when needed, with bf16 mixed precision training and gradient-norm clipping at 1.0.

Inference The truth event is tokenized with the frozen encoder and runs through the model once (prefill) to fill the key/value cache. N_{PF} is then sampled from the cardinality head, and exactly $4N_{\text{PF}}$ codes are sampled autoregressively. Padding and start tokens are masked out from the output's categorical distribution at every step. The sampled codes are summed across depths to $\hat{z}$ and decoded by the frozen RQ-VAE decoder over the whole generated event. KV-caching is used at inference to reduce wall-time. To reduce padding waste, events are first prefilled through the model once, and the cardinality is sampled from the corresponding head. The events are then reshuffled such that events in the same batch have the same predicted cardinality and thus we avoid wasting decode steps on padding tokens. A further decoding option, since our cardinality prediction is independent of token sampling (no `[EOS]` token) is forcing the token cardinality of the target *PFlow* event corresponding to the fed *Truth* conditioning, though we find that this *perfect cardinality* method does not yield measurably different results from regularly sampling the cardinality during the prefill phase.
Inference-time generation speeds are available in Table 8.

Table 8: Generation throughput of the best run of each model size, measured on 50,000 held-out events. Times include quantization, prefill, autoregressive decode and the RQ-VAE decoding. Each model is tested at the biggest power-of-two batch size B to fit in GPU memory with an 8% margin. The models are tested on NVIDIA A100 SXM4 (80 GB) and NVIDIA H200 (141 GB) GPUs.

	A100			H200		
Model	B	**evt/s**	**ms/evt**	B	**evt/s**	**ms/evt**
10M	1,024	126.5	7.9	2,048	265.1	3.8
22M †	512	15.1	66.1	1,024	26.8	37.4
41M	256	62.5	16.0	512	132.5	7.5
97M	256	38.2	26.2	256	72.7	13.8
187M	128	22.0	45.4	256	48.8	20.5
1B	32	4.4	229.1	64	12.0	83.2

†The 22M model is slower than the parameter count would suggest because of its attention head dimension, $d_{\text{head}} = 48$, for which the inference-running system had no optimized FlashAttn kernel.

C Scaling study methods

C.1 Loss scaling

The fitted $\mathcal{L}(N, U, R)$ law uses events as the data indicator: at a fixed test distribution, average tokens per event converge to a constant which the amplitude B absorbs. Only the size, the data fraction and the epoch count change across the grid. The architecture family, optimizer, peak learning rate, batch size and schedule shape are the same for all runs.

We fit the four laws of Table 5 directly in the parameter count N and the number of unique events U, with the amplitudes A, B and P as free parameters. Every law is built from the same two terms:

$$F(N, U, R) = A\,N^{-\alpha} + B\,[U(1+R)]^{-\beta}, \tag{7}$$

$$\text{penalty}(N, U, R) = P\,R^{\delta}\left(N/U^{\gamma}\right)^{\kappa}. \tag{8}$$

Chinchilla is $L_\infty + F$, Skaling is $L_\infty + F^k$, Prescriptive is $L_\infty + F +$ penalty, and Custom is $L_\infty + F^k +$ penalty. The penalty is zero at $R = 0$. The laws without a repetition term are only defined for single-epoch data and are fitted to the 18 single-epoch runs. The two repetition-aware laws are fitted to all 48 runs. We minimize the unweighted sum of squared residuals in $\mathcal{L}$ with a least-squares solver. The used loss values are the means on the held-out in-distribution testing dataset. In Table 9 we report the fitted parameters, the in-sample Mean Absolute Percentage Error (MAPE) and the coefficient of determination (R^2).

C.2 IsoFLOP analysis

Conventionally, for studies at a fixed compute budget C (introduced by Hoffmann et al. (2022)), several model sizes are trained, and their learning-rate schedules end exactly at C. Our grid was designed for the (N, U, R) fits instead. Each single-epoch run (repetition is not modeled in these studies) covers one compute value at its endpoint and a range of smaller compute values along its training curve. We therefore build isoFLOP slices from the monitored training curves of the 18 single-epoch runs, following the steps below. We use the $C \approx 6ND$ (Kaplan et al., 2020) approximation, with D being the number of total tokens processed, i.e. $C \approx 6N[4(n_{\text{TR}} + n_{\text{PF}}) + 2]n_{\text{Evts}}$. The main difficulty is that for an intermediate checkpoint of a run, its learning rate has not yet fully annealed. Results should be taken with this consideration in mind. The specific retrieval methods are explained in the following section.

Every single-epoch run saves a checkpoint up to every $1/16$ of an epoch, giving 8 to 16 logs per run. Every saved checkpoint's loss is then evaluated on the held out testing dataset. For a run whose logged range contains the desired compute, we estimate its loss at that compute value by local quadratic interpolation in $\log t$, where t is the optimizer step count: we fit a parabola in $\log t$

Table 9: Fitted parameters of the loss scaling laws. Chinchilla and Skaling are fitted to the 18 single-epoch runs, Prescriptive and Custom to all 48 runs; N and U are in parameters and unique events. Uncertainties are 1σ errors over refits with one held-out run.

	Chinchilla	**Skaling**	**Prescriptive**	**Custom**
L_∞	2.67 ± 0.40	3.191 ± 0.082	2.80 ± 0.11	3.255 ± 0.015
A	1300 ± 4100	34 ± 31	900 ± 1200	55 ± 14
α	0.42 ± 0.18	0.241 ± 0.052	0.410 ± 0.092	0.273 ± 0.017
B	$(0.9 \pm 2.2) \times 10^4$	106 ± 93	7400 ± 3900	114 ± 24
β	0.62 ± 0.15	0.345 ± 0.070	0.598 ± 0.036	0.349 ± 0.016
k	–	2.39 ± 0.32	–	2.379 ± 0.072
P	–	–	$(4.5 \pm 7.9) \times 10^{-13}$	$(3.6 \pm 3.8) \times 10^{-11}$
κ	–	–	1.260 ± 0.084	1.411 ± 0.033
γ	–	–	0.136 ± 0.041	0.531 ± 0.041
δ	–	–	1.641 ± 0.092	1.660 ± 0.028
MAPE [%]	3.55	1.17	2.47	0.79
R^2	0.9695	0.9961	0.9690	0.9967

through the three logged points closest to $t_w(C)$ (the step at which the run reaches C) and evaluate it there.

For each C budget, in order to find the minimum achievable loss value we fit a parabola through all the available points, with the criteria that at the specified budget at least 4 different values be available. Fig. 8 shows the selected isoFLOP slices and associated loss points.

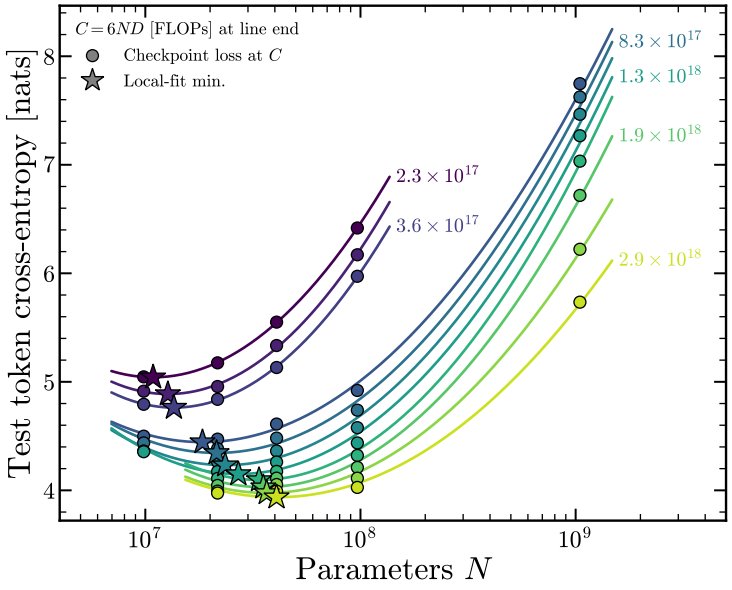


Figure 8: IsoFLOP slices fit through the measured loss values. For every slice we show the minima and the compute value.

C.3 Scaling in the physical metrics

For every run we generate 50,000 *PFlow* events conditioned on the same test *Truth* dataset (temperature 1, sampled cardinality) and decode it with the frozen tokenizer. Before computing any observable, generated and reference *PF* events pass the same acceptance, $p_T > 1\,\text{GeV}$ and $|\eta| < 2.5$, thresholds.

Residual feature sets. We compare distributions of residuals with respect to the truth input rather than raw marginals:

- *Event residuals*: for every *FS* and *PF* event, we compute the difference with respect to the corresponding *TR* event of the scalar sum of p_T, the missing transverse energies on the x and y axes, and the difference in particle cardinality.
- *Particle residuals*: generated particles are Hungarian-matched to the *TR* set, with cost ΔR. Pairs with $\Delta R > 0.6$ are discarded. Particle continuous features are then considered for the residuals.
- *Jet-residuals*: anti-k_T jets ($R = 0.5$) are clustered at both levels and matched in the same way, with $\Delta R < 0.2$. The residuals are those of the jet kinematic and substructure features of Section 3.2.

The full generated manifolds for the different feature sets are subsampled to at most 2×10^5 rows (this is relevant for the particle-level features). The computed distance is averaged over 256 random projections of the one-dimensional W_1 between the samples. The *noise floor* is the same metric computed between the reference test sample and another statistically identical sample. The RQ-VAE reference is the distance between the reference *PF* events and the reconstruction of the second sample from the residual-quantized representation. For the one-dimensional local metrics we use the exact W_1 between residual distributions.

The loss-metric relation (Eq. 4) is fitted by least squares in $\ln$ SWD, only on runs that are converged and not overfitted. We require (i) $C = 6ND \geq 3 \times 10^{17}$, (ii) no overfitting of NTP loss and (iii) no overfitting on the cardinality prediction objective, (iv) more than 2 training tokens per model parameter. Intermediate checkpoints are included to probe the loss–metric relationship across training dynamics.

D Additional material

D.1 Quantization results

In Figure 9 we report the per-quantizer level codebook usages and perplexities, evaluated across all three testing datasets.

D.2 Generation results

We present the marginal feature distribution corresponding to the residual distributions presented in Section 3.2. The spectra of particle features are available in Figure 10, those of jets in Figure 11, and finally the distributions of event-level features are shown in Figure 12. A comparison of the generation performance of different ablations is available in Table 10

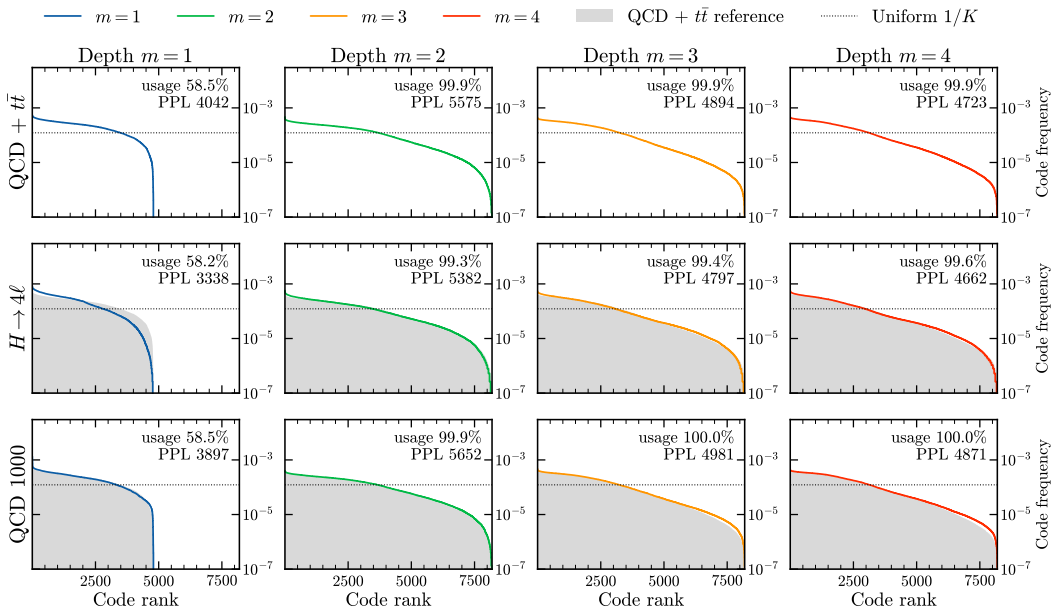


Figure 9: Usage distributions across the testing datasets. Codes are ordered on the x axis according to their usage rank, from most to least used.

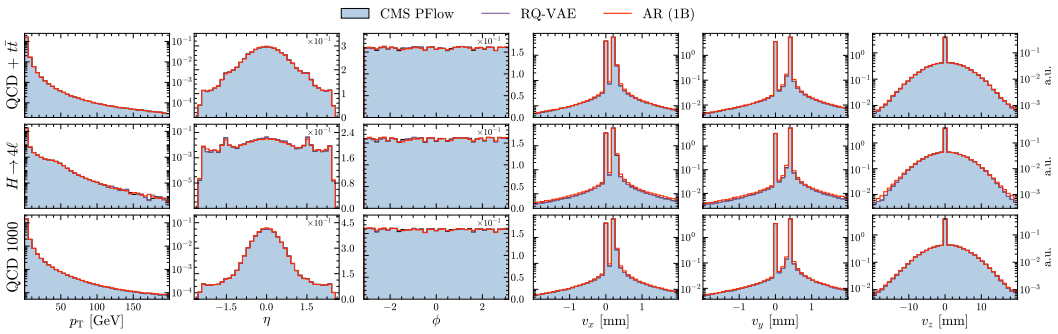


Figure 10: Spectra of particle-level features for all three testing datasets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

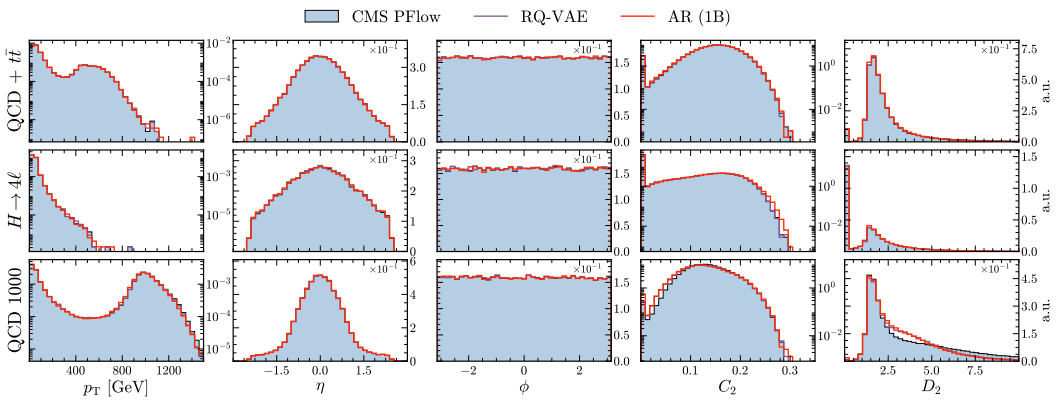


Figure 11: Spectra of jet-level features for all three testing datasets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

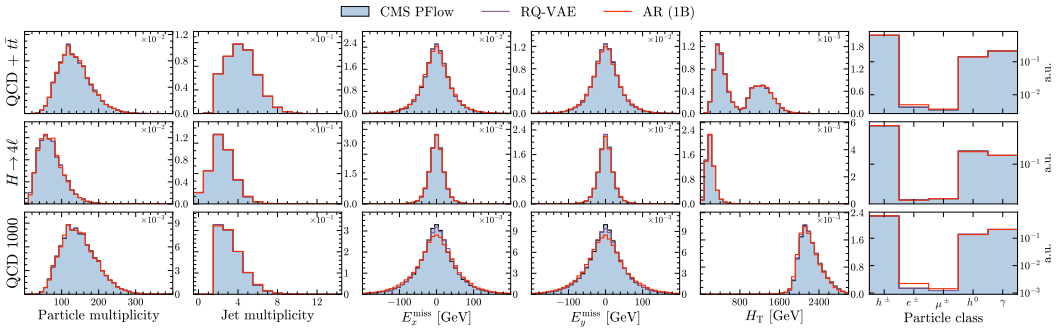


Figure 12: Spectra of event-level features for all three testing datasets. All histograms are normalized to unity; rows correspond to different datasets/processes while columns correspond to different observables.

Table 10: Per-feature W_1 distance between the *FS* and *PF* residual distributions, in the units of each feature, for the models trained on the full dataset for 4 epochs, on the held-out in-distribution test set (50,000 events). The RQ-VAE column is the tokenizer reconstruction of an independent 50,000-event sample with the same process mix, and the noise column the distance between two such independent *PF* samples. The best model in each row is in bold.

Residual	**Unit**	**10M**	**22M**	**41M**	**97M**	**187M**	**1B**	**RQ-VAE**	**Noise**
Event residuals									
$\Delta H_T/H_T$	–	0.0212	0.0154	0.0102	0.00538	0.00526	**0.00262**	0.00144	0.00132
ΔE_x^{miss}	GeV	17.3	9.98	6.69	3.81	2.97	**1.81**	0.363	0.271
ΔE_y^{miss}	GeV	17.1	10.3	6.82	3.77	3.09	**1.79**	0.251	0.347
ΔE_T^{miss}	GeV	22.9	13.2	8.60	4.62	3.78	**2.09**	0.439	0.491
$\Delta N/N$	–	0.0195	0.0131	0.00940	0.0130	0.00629	**0.00513**	0.00793	0.00509
ΔN_{jets}	–	0.232	0.144	0.101	0.0646	0.0541	**0.0292**	0.00656	0.00612
Jet residuals									
$\Delta p_T/p_T$	–	0.0708	0.0475	0.0344	0.0212	0.0171	**0.00990**	0.000811	0.000869
$\Delta\eta$	–	0.0306	0.0208	0.0152	0.0112	0.00994	**0.00787**	0.00296	0.000114
$\Delta\phi$	rad	0.0117	0.00783	0.00609	0.00424	0.00364	**0.00248**	0.000470	0.000105
ΔC_2	–	0.00792	0.00504	0.00382	0.00237	0.00207	**0.00127**	0.000612	0.000198
ΔD_2	–	0.433	0.327	0.271	0.240	0.218	**0.189**	0.135	0.0112
Particle residuals									
$\Delta p_T/p_T$	–	1.12	0.813	0.670	0.526	0.477	**0.392**	0.212	0.00564
$\Delta\eta$	–	0.0285	0.0189	0.0154	0.0113	0.0103	**0.00809**	0.00357	0.000163
$\Delta\phi$	rad	0.0185	0.0111	0.00946	0.00625	0.00555	**0.00398**	0.00183	0.000225
Δv_z	mm	0.375	0.232	0.245	**0.160**	0.229	0.187	0.0356	0.0138
Δv_x	mm	**0.0292**	0.0414	0.0461	0.0328	0.0449	0.0309	0.00145	0.00143
Δv_y	mm	0.0310	0.0429	0.0513	0.0349	0.0458	**0.0287**	0.00178	0.00250