Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

Generative Modelling of Calorimeter Showers and Particle Jets

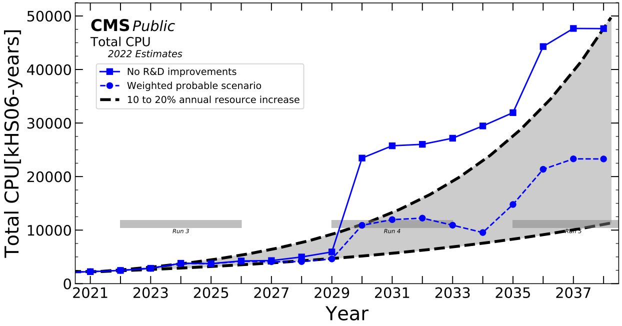
Erik Buhmann, Thorsten Buss, Sascha Diefenbacher, Engin Eren, Cedric Ewen, Frank Gaede, Gregor Kasieczka, William Korcari, Anatolii Korol, Katja Krüge, Peter McKeown, Martina Mozzanica, Lennard Rustige, Lorenzo Valente

Dec 14, 2023
Workshop on Machine Learning and High-Energy Physics

thorsten.buss@uni-hamburg.de

Detector Simulation

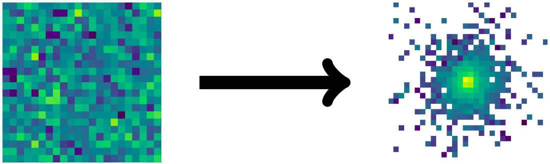
- monte carlo (MC) necessary to compare theory and measurements
- computational requirements expected to exceed available resources soon
- detector simulation most expensive part of simulation chain



[1] CMS Offline Software and Computing. *CMS Phase-2 Computing Model: Update Document*. 2022. URL: https://cds.cern.ch/record/2815292

Generative models

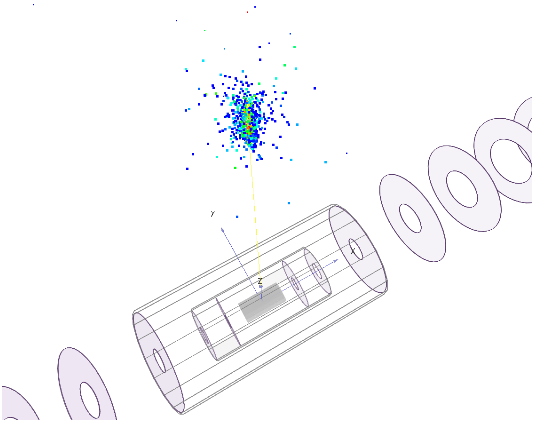
- generative neuronal networks learn distributions and can sample from them
- work flow:
  - simulate small amounts of data using slow monte carlo
  - train generative model on these data
  - draw large amounts of data from fast ML model



- a variety of generative models exists:
  - Generative Adversarial Networks (GAN)
  - Autoencoders (AE)
  - Normalizing Flows (NF)
  - Diffusion Models (DM)

International Large Detector (ILD)

- proposed detector for the ILC
- has two sampling calorimeters
- electromagnetic calorimeter (ECAL)
  - 30 layers, 5mm x 5mm cells
- hadronic calorimeter (HCAL)
  - 48 layers, 30mm x 30mm cells
- dataset[2]: photon showers in ECAL



[2] Erik Buhmann et al. *Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed*. 2021. arXiv: 2005.05334

[3] ILD Concept Group. *International Large Detector: Interim Design Report*. 2020. arXiv: 2003.01116

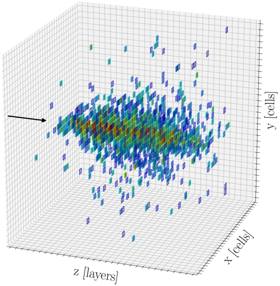
Data representation of showers

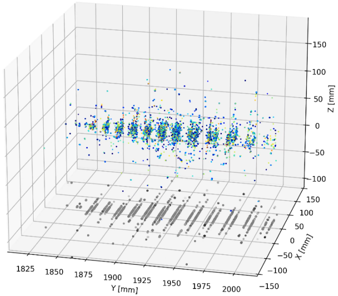
Fxed Grid

- 3D array filled with energy values
- entries correspond to calorimeter cells
- allows for convolutional networks

Point Clouds

- variable-length, permutation-invariant sets
- calorimeter showers are very sparse
- more economically represented
- only generation of non-zero points

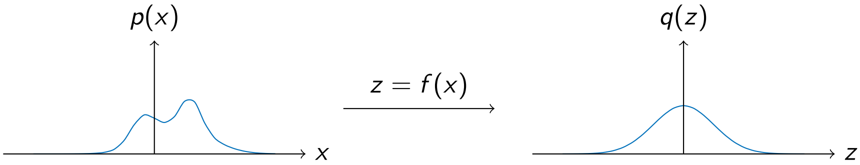




Normalizing Flows

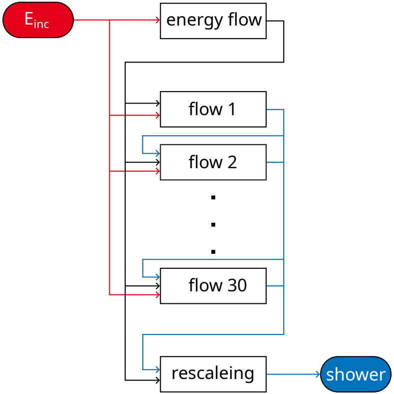
- diffeomorphism between physics space and latent space
- transform physics space distribution into a simple prior distribution
- change of variables formula allows for physics space density estimation
- training: minimize negative log-likelihood
- generation: sample from latent distribution and apply inverse of function

$$p(x) = q(f(x))\,|J_f(x)| \qquad \mathcal{L} = -\log q(f(x)) - \log|J_f(x)|$$



Convolutional L2LFlows

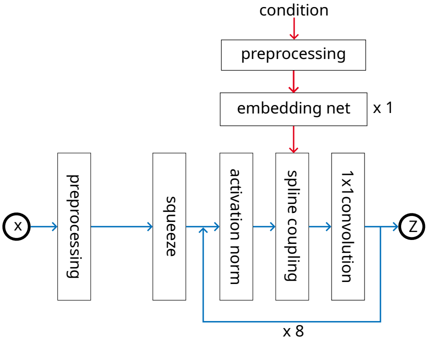
- based on CaloFlow[4] and L2LFlows[5]
- one energy distribution flow
  - learns distribution of layer energies
  - conditioned on incident energy
- 30 causal flows
  - learn shower shape in layer
  - conditioned on
    - incident energy
    - layer energy
    - previous layers
- generation
  - sample layer energies using energy distribution flow
  - sample shower shape using causal flows
  - rescale voxel energies



4 Claudius Krause and David Shih. *CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows*. 2021. arXiv: 2106.05285

5 Sascha Diefenbacher et al. *L2LFlows: Generating High-Fidelity 3D Calorimeter Images*. 2023. arXiv: 2302.11594

Flow Architecture



- energy distribution flow
  - masked autoregressive flow[6]
- causal flows
  - spline coupling flow[7]
  - convolutional U-Nets[8] as sub networks
  - architecture similar to Glow[9]
- features in energy spectrum are smeared out
  - → apply element-with function to get them back

---

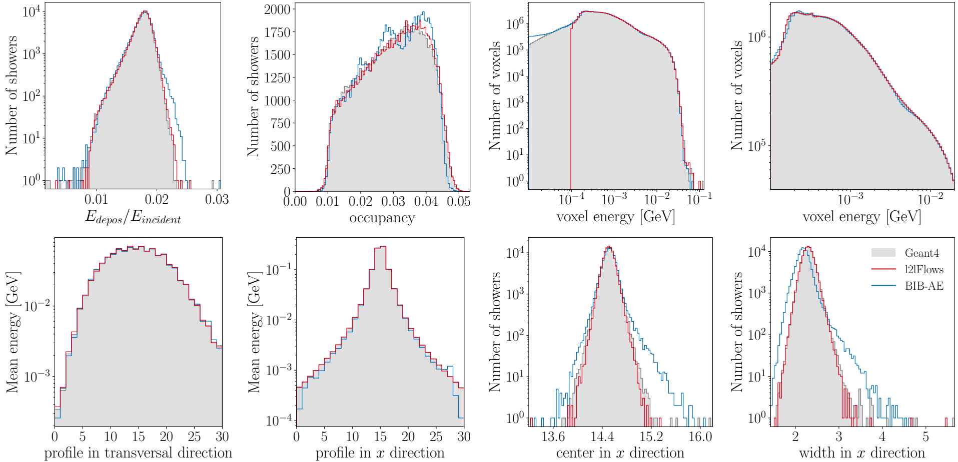
6 Mathieu Germain et al. *MADE: Masked Autoencoder for Distribution Estimation*. 2015. arXiv: 1502.03509

7 Conor Durkan et al. *Neural Spline Flows*. 2019. arXiv: 1906.04032

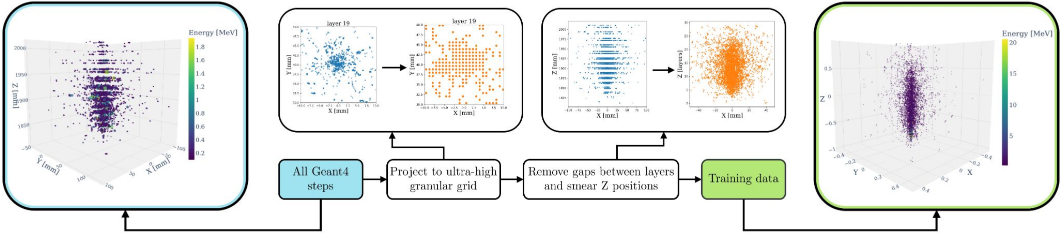
8 Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597

9 Diederik P. Kingma and Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. 2018. arXiv: 1807.03039

L2LFlows Results



Point Cloud Representation Pre-Processing

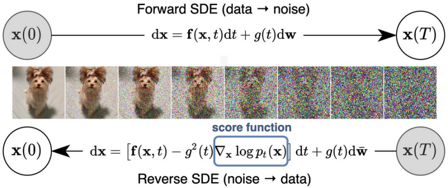


- point clouds of clustered Geant4 steps
- 36x higher resolution than detector cells
- 7x fewer points than full Geant4 steps

	points per shower
all Geant4 steps	40 000
clustered Geant4 steps	6 000
hits in calorimeter grid	1 500

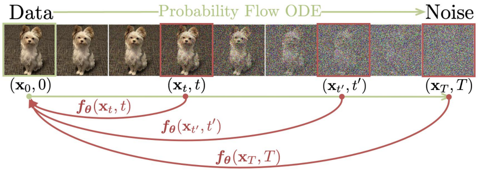
Diffusion Models

- denoising diffusion model[10]
  - discrete time diffusion process
  - train to predict noise vector
  - number of time steps is fixed
- score based model[11]
  - continuous time diffusion process
  - stochastic differential equation (SDE)
  - sample by solving reverse SDE
- probability flow ODE
  - remove stochasticity
  - SDF $\to$ ODE
- consistency model distillation[12]
  - allows for single step sampling



$$\mathcal{L} = \|s_\theta(x_t, t) - \nabla_x \log p_t(x_t)\|_2^2$$

$$dx = [f(x, t) - \frac{1}{2} g(x, t)^2 \nabla_x \log p_t(x)]dt$$



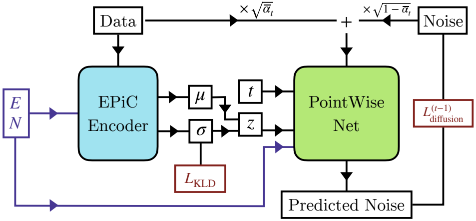
10 Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239

11 Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456

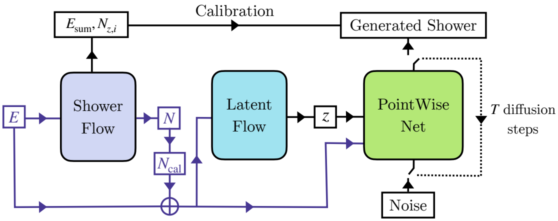
12 Yang Song et al. *Consistency Models*. 2023. arXiv: 2303.01469

Calo Clouds I

training



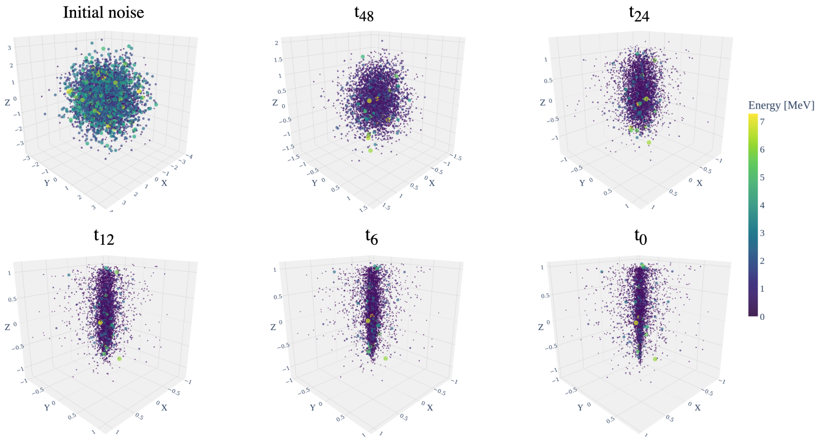
sampling



- point cloud denoising model
  - discrete time diffusion process
  - 100 time steps
- post-denoising calibration
  - visible deposited energy
  - center of gravity in X and Y-direction

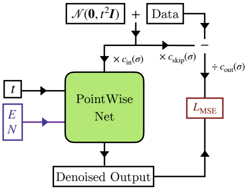
[13] Erik Buhmann et al. *CaloClouds: fast geometry-independent highly-granular calorimeter simulation*. 2023. arXiv: 2305.04847

Reverse Diffusion Process

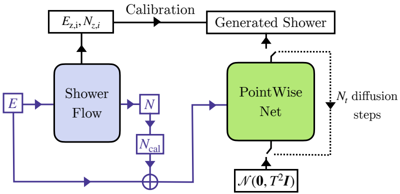


Calo Clouds II

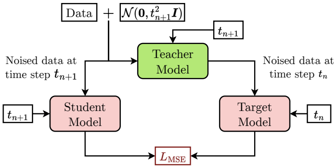
training



sampling



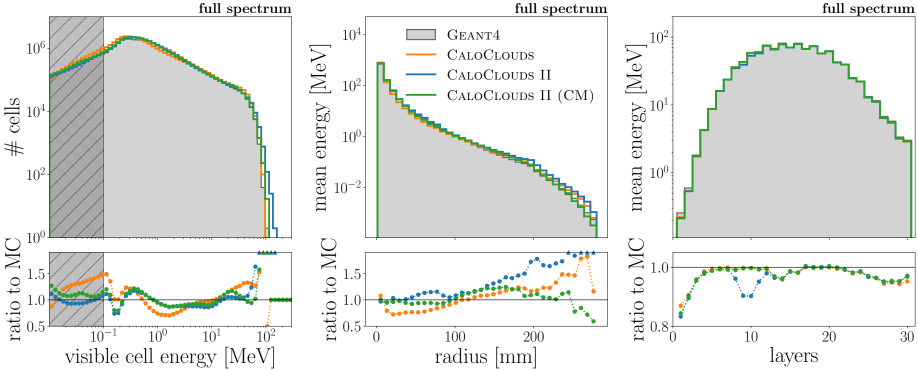
distillation



- score based model
  - continuous time diffusion process
  - probability flow ODE
  - 25 network evaluations
- distillation into a consistency model
  - allows for single step sampling

14 Erik Buhmann et al. *CaloClouds II: Ultra-Fast Geometry-Independent Highly-Granular Calorimeter Simulation*. 2023. arXiv: 2309.05704

CaloClouds Results



Timing

Simulator	Hardware	Batch size	time [ms]
GEANT4	CPU	1	4081.53
L2LFlows		1	1202.66
BIB-AE		1	426.32
L2LFlows	GPU	100	12.34
BIB-AE		100	1.42

timing on Getting High dataset

Simulator	Hardware	Batch size	time [ms]
GEANT4	CPU	1	3914.80
CaloClouds I		1	3146.71
CaloClouds II		1	651.68
CM		1	84.35
CaloClouds I	GPU	64	24.91
CaloClouds II		64	6.12
CM		64	2.09

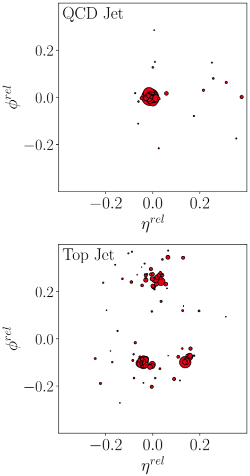
timing on CaloClouds dataset

Particle Jets Dataset

- benchmark dataset: JetNet30[15]
- simulated jets from proton-proton collisions
- anti-k_T clustered with $R = 0.8$
- maximum particle multiplicity $N = 30$
- constituents coordinates normalized and centered

$$p_T^{\text{rel}} = \frac{p_T}{p_T^{\text{jet}}} \qquad \eta_T^{\text{rel}} = \eta_T - \eta_T^{\text{jet}} \qquad \phi_T^{\text{rel}} = \phi_T - \phi_T^{\text{jet}}$$

- jet types: Gluons, light quarks, Top quarks



[15] Raghav Kansal et al. *Particle Cloud Generation with Message Passing Generative Adversarial Networks*. 2022. arXiv: 2106.11535

Continuous Normalizing Flow

Normalizing Flow

$$z := f_\theta(x) \qquad z \sim q$$

Training

$$\log p(x) = \log q(f(x)) - \log|J_f(x)|$$

- sampling
  - sample noise from prior distillation
  - map it to data distillation using f^{-1}
- f restrained to easy invertible functions

Continuous Normalizing Flow[16]

$$x_t := f(x_0, t) \qquad \partial_t x_t = v_\theta(x_t, t) \qquad x_1 \sim p_1$$

Training

$$\log p_0(x_0) = \log p_1(x_1) - \int_0^1 \mathrm{Tr}\left(\frac{\partial v_\theta}{\partial x_t}\right) dt$$

- sampling
  - sample noise from prior distillation
  - solve ODE given by the network
- v has no strong restrictions

[16] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. 2019. arXiv: 1806.07366

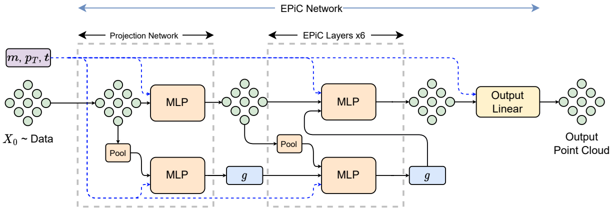
EPiC-FM & EPiC-JeDi

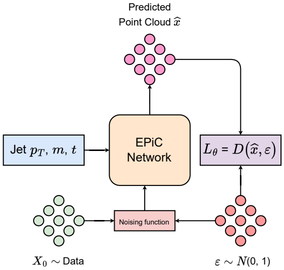
EPiC-FM: EPiC Architecture with Flow Matching

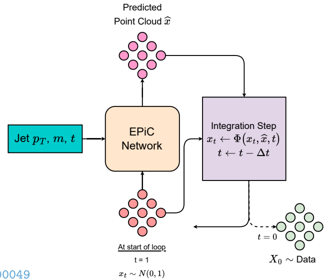
$$\mathcal{L}_{FM} = \| v_\theta(x_t, t) - ((1 - \sigma_{\min})\epsilon - x_0) \|_2^2$$

EPiC-JeDi: EPiC Architecture with JeDi[18]

$$\mathcal{L}_{JeDi} = \left(1 - \alpha \frac{\beta(t)}{\sigma(t)^2}\right) \| v_\theta(x_t, t) - \epsilon \|_2^2$$



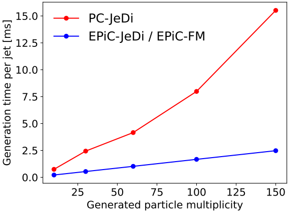


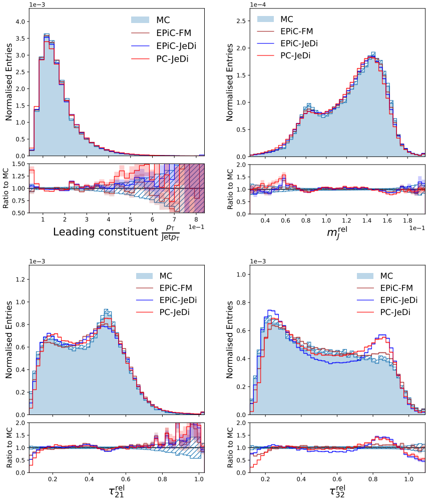


17 Erik Buhmann et al. *EPiC-ly Fast Particle Cloud Generation with Flow-Matching and Diffusion*. 2023. arXiv: 2310.00049

EPiC-FM & EPiC-JeDi Results

- conditioned version m^{jet} and p_T^{jet}
- unconditioned version
- generate conditioning with normalizing flow
- comparison to PC-JeDi[18]
- midpoint ODE solver with 200 model passes
- substructure most challenging to learn





[18] Matthew Leigh et al. *PC-JeDi: Diffusion for Particle Cloud Generation in High Energy Physics*. 2023. arXiv: 2303.05376

Summary

L2LFlows

- fixed grid representation
- improved performance
- good scaling behavior

CaloClouds I + II

- point cloud representation
- geometry independent
- very fast generation

EPiC-ly Flow Matching

- high fidelity jet generation
- good scaling with multiplicity