Generating Accurate Showers in Highly Granular Calorimeters Using Normalizing Flows

Thorsten Buss, Sascha Diefenbacher, Frank Gaede,
Gregor Kasieczka, Claudius Krause, David Shih

thorsten.buss@uni-hamburg.de

Institut für Experimentalphysik
Universität Hamburg

May 31, 2023

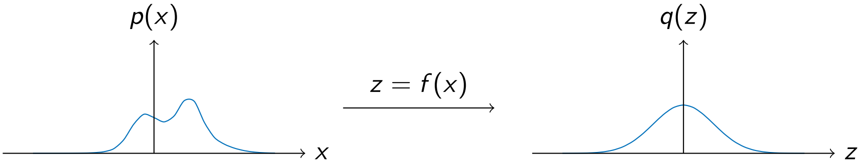
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

Normalizing Flows

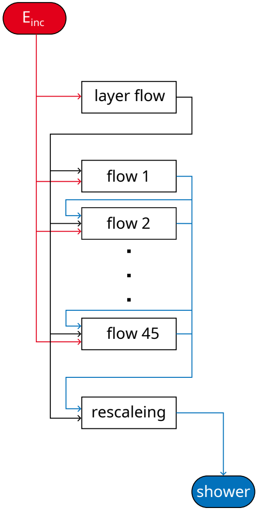
- diffeomorphism between physics space and latent space
- transform physics space distribution into a simple prior distribution
- change of variables formula allows for physics space density estimation
- training: minimize negative log-likelihood
- generation: sample from latent distribution and apply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

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



Architecture

- based on CaloFlow[1] and L2L Flow[2]
- one layer flow
  - learns distribution of layer energies
  - conditioned on incident energy
- 45 multiple flows
  - learn shower shape in layer
  - conditioned on
    - incident energy
    - layer energy
    - privies layers (up to five)
  - convolutional architecture
- generation
  - sample layer energies using layer flow
  - sample shower shape using multiple flows
  - rescale voxel energies



[1] Claudius Krause and David Shih. *CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows*. 2021. arXiv: 2106.05285.
[2] Sascha Diefenbacher et al. *L2LFlows: Generating High-Fidelity 3D Calorimeter Images*. 2023. arXiv: 2302.11594.

Layer Flow

pre/post-processing:

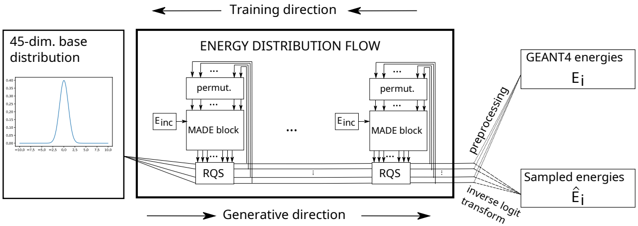
- $x \to \text{logit}(\frac{x}{x_{max}})$
- $c = 2\log_{10}(\frac{e_{inc}}{\text{GeV}}) - 3$
- add gaussian noise
- veto high energy ratios

MADE block:

- masked NN
- guarantees invertibility
- gives spline parameters

training:

- initial learning rate: 10^{-3}
- optimizer: Adam
- scheduler: exponential
- batch size: 1024
- epochs: 2000



graphic modified from [2110.11377 and 2302.11594]

Multiple Flows Components

pre/post-processing:

- $x \to \mathrm{logit}(\frac{x}{x_{max}})$
- $c_1 = 2\log_{10}(\frac{e_{inc}}{\mathrm{GeV}}) - 3$
- $c_2 = \log_{10}(\frac{e_i}{\mathrm{GeV}}) + 1$
- add log normal noise
- apply cut at 1.5×10^{-5} GeV

squeezes[3]:

- exchange spacial dimensions for channels by stacking pixels
- use asymmetrical squeezes
- squeezes factors are prime factors of #dimensions

activation norm[4]:

- normalize to $\mu = 0$ and $\sigma = 1$
- initialize on batch one
- then trainable parameters

spline coupling block[5]:

- split along channels
- cubic spline
- convolutional subnet

1×1 convolution[4]:

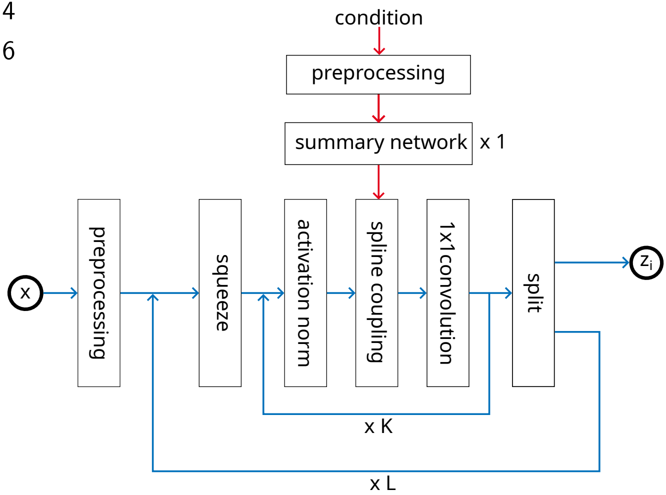
- replaces permutation
- LU decomposed linear transformation

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017. arXiv: 1605.08803.
[4] Diederik P. Kingma and Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. 2018. arXiv: 1807.03039.
[5] Conor Durkan et al. *Cubic-Spline Flows*. 2019. arXiv: 1906.02145.

Multi-scale Architecture

$$K = 4$$

$$L = 6$$



Subnets

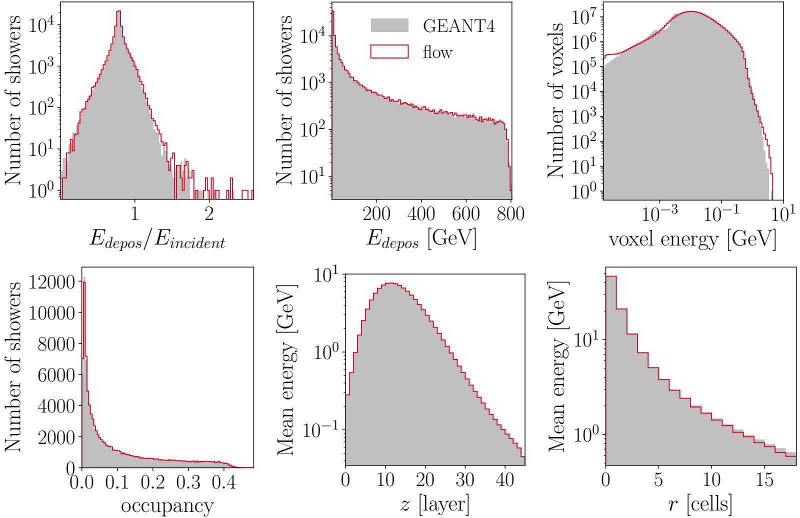
summary network:

- input
  - five previous layers
  - incident energy
  - deposited energy in layer i
- output
  - one output per squeeze
  - shape: $8 \times n_\alpha \times n_r$
- network layer
  - 1×1 convolution
  - additional per squeeze
    - 3×3 convolution (cylindrical padding)
    - down sampling (max pooling or stride)
    - 3×3 convolution (cylindrical padding)
  - activation: LeakyReLu

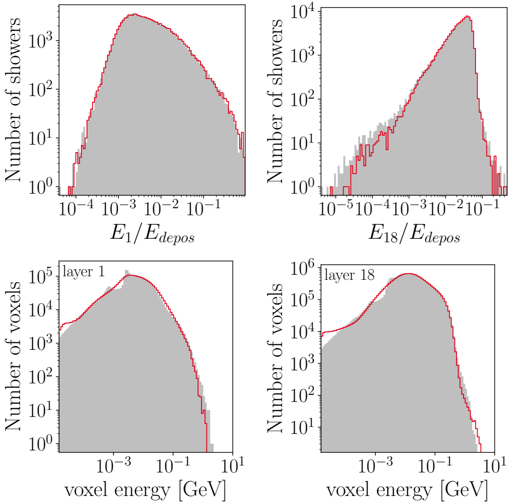
subnets:

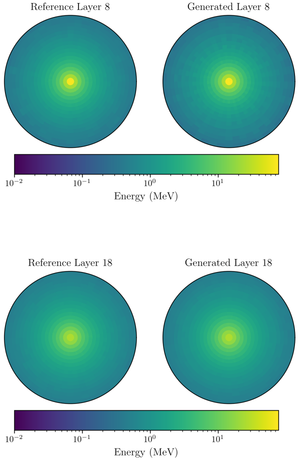
- input
  - untransformed channels
  - summarized condition
- output
  - spline parameters for transformed channels
- layer
  - 1×1 convolution
  - 3×3 convolution
    - cylindrical padding
  - 3×3
    - cylindrical padding
  - activation: LeakyReLu

Dataset 3


Number of showers
10^0 10^1 10^2 10^3 10^4
1 2
$E_{depos}/E_{incident}$
GEANT4
flow
200 400 600 800
E_{depos} [GeV]
Number of voxels
10^0 10^1 10^2 10^3 10^4 10^5 10^6 10^7
10^{-3} 10^{-1} 10^1
voxel energy [GeV]
0 2000 4000 6000 8000 10000 12000
0.0 0.1 0.2 0.3 0.4
occupancy
Mean energy [GeV]
10^{-1} 10^0 10^1
0 10 20 30 40
z [layer]
0 5 10 15
r [cells]

Dataset 3


Number of showers
E_1/E_{depos}
Number of showers
E_{18}/E_{depos}
layer 1
Number of voxels
voxel energy [GeV]
layer 18
Number of voxels
voxel energy [GeV]


Reference Layer 8
Generated Layer 8
Energy (MeV)
Reference Layer 18
Generated Layer 18
Energy (MeV)

Metrics

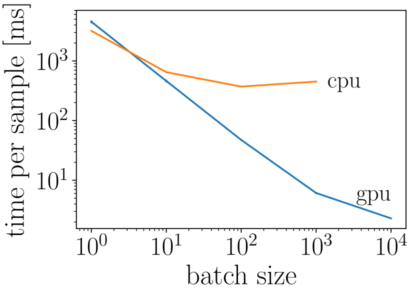
LL-Classifier	AUC	JSD
dataset 2	0.81	0.22
dataset 3	0.80	0.22

HL-Classifier	AUC	JSD
dataset 2	0.79	0.20
dataset 3	0.87	0.35

χ^2	dataset 2	dataset 3
Voxel spectrum	2.8×10^{-3}	4.6×10^{-3}
E_{tot}/E_{inc}	1.6×10^{-3}	1.7×10^{-3}
E_1	1.4×10^{-3}	8.7×10^{-4}
E_{10}	1.6×10^{-4}	3.5×10^{-4}
E_{30}	5.5×10^{-3}	7.5×10^{-3}
E_{45}	9.8×10^{-3}	1.7×10^{-2}
$\mu_{\eta 1}$	3.0×10^{-3}	2.9×10^{-3}
$\mu_{\eta 10}$	1.9×10^{-3}	2.1×10^{-3}
$\mu_{\eta 30}$	5.2×10^{-3}	8.8×10^{-3}
$\mu_{\eta 45}$	5.5×10^{-3}	5.9×10^{-3}

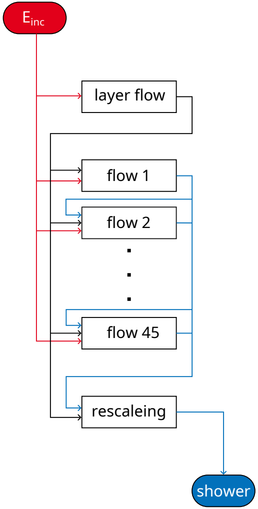
Timing

- results for dataset 3
- comparison of generation times
- hardware:
  - cpu: Intel® Xeon® E5-2640
  - gpu: NVIDIA® A100®
- #threads for cpu case: 1

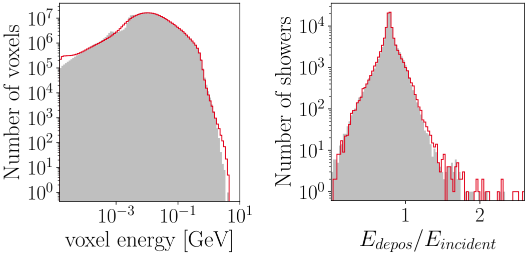


Hardware	Batch size	time [ms]
CPU	1	3194.46
	10	647.45
	100	370.86
GPU	1	4531.97
	1000	6.11

Conclusion



- convolutional flows scale well with input dimensions
- evaluation of inverse in almost same time
- flows can generate highly accurate showers



Regular ILD Dataset