CaloPointFlow

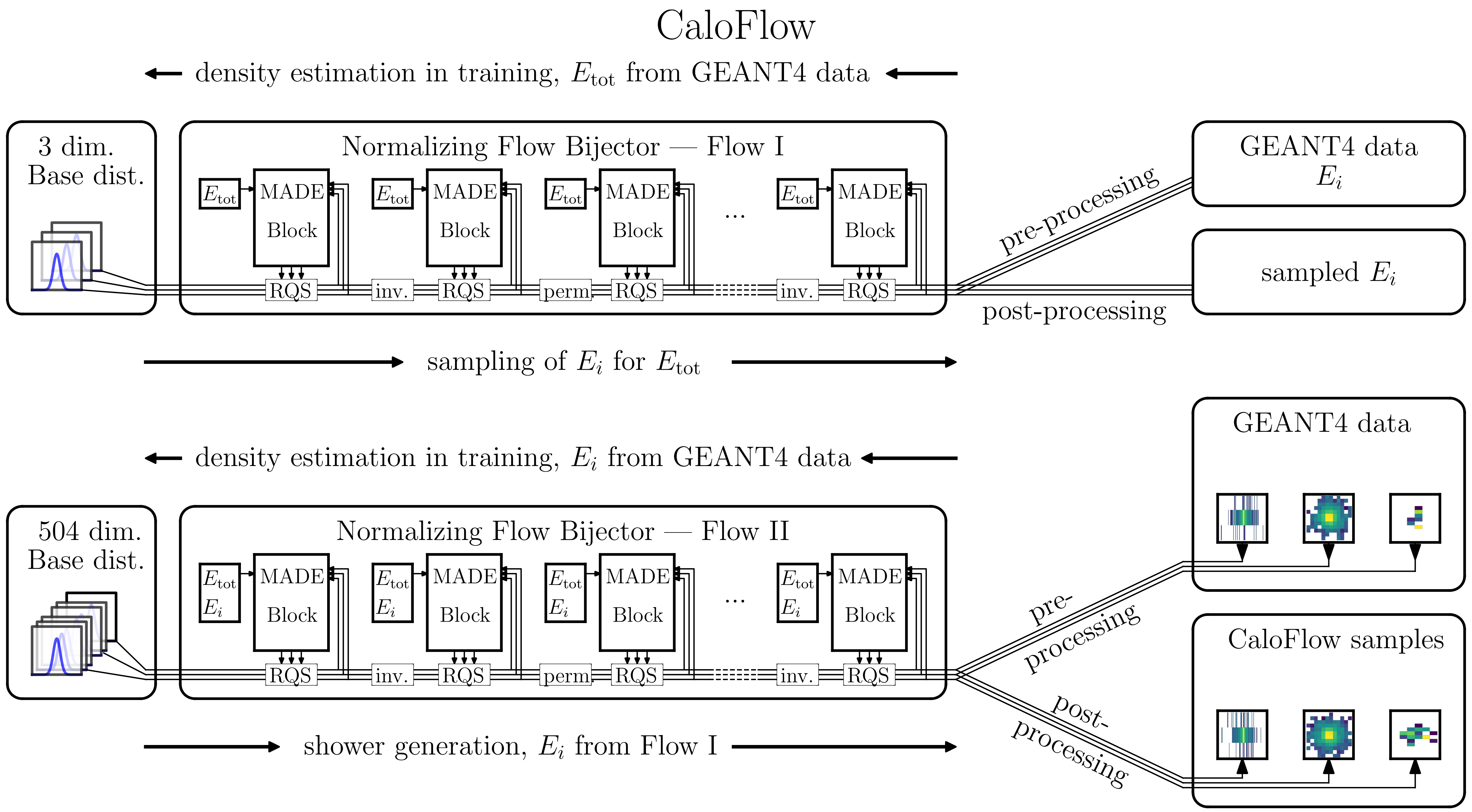
Results for the CaloChallenge Datasets

Kerstin Borras, Dirk Krücker, Simon Schnake
31.05.23
CaloChallenge Workshop

Motivation

Normalizing Flows are great but hard to scale

- Directly trainable by *max. likelihood*
- Fast and stable convergence
- CaloFlow passes classifier test
- Invertible property leads to $\mathcal{O}(n^2)$ scaling where n is the number of *input dimensions*



CaloFlow from *Krause et al.*: [2106.05285]

Motivation

Overcoming $\mathcal{O}(n^2)$ scaling

- Different possible routes
- Learn layer by layer (*L2LFlows* [2302.11594] / *Inductive CaloFlow* [2305.11934])
- Reduce voxelized calorimeter to point clouds
- Point cloud advantages
    - Calorimeter showers are sparse $\rightarrow$ lower n
    - Learn each point separately $n = 4$
    - Applicable to complex geometries

CaloPointFlow

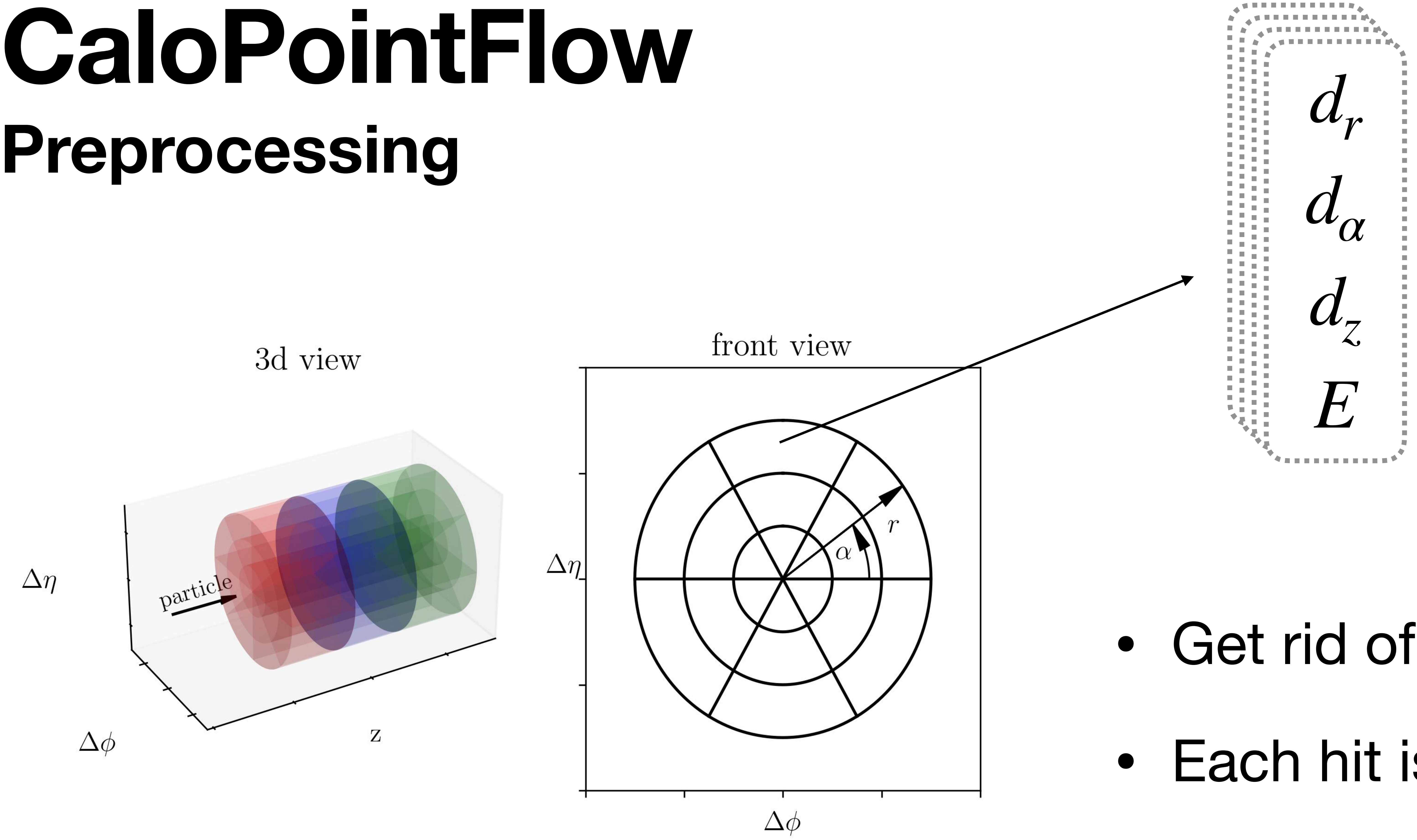
Approach

model has been published at
NeurIPS ML4PS
and evaluated on a different
dataset

- Interpret calorimeter showers as point clouds
- Generate shower shape information first
- Generate each point independently conditioned on the shower shape
- Inter-point-correlations ignored
- Based on PointFlow [1906.12320]

CaloPointFlow

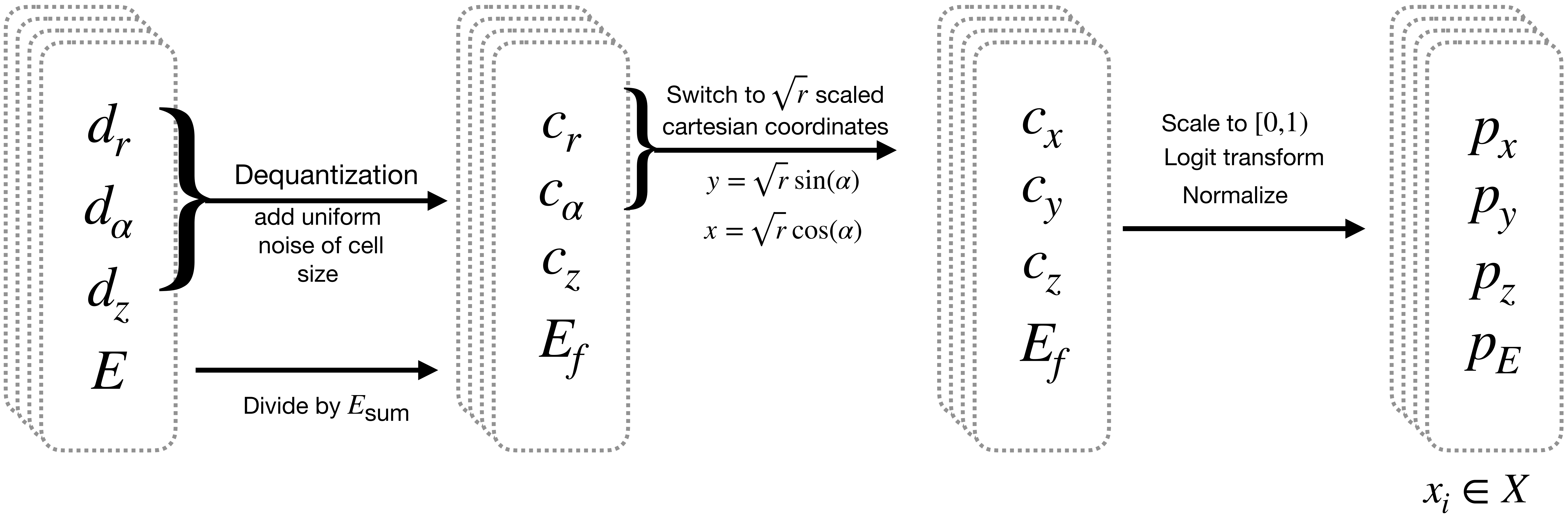
Preprocessing



- Get rid of empty cells
- Each hit is represented as point
- One shower equals to one point cloud

CaloPointFlow

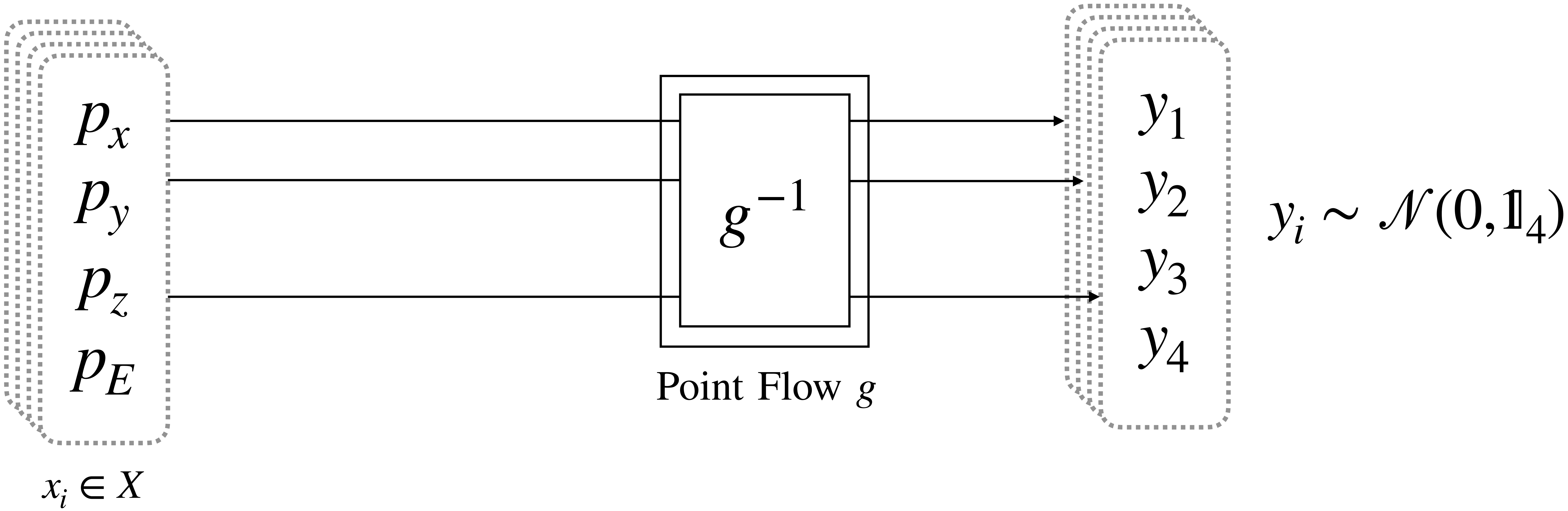
Preprocessing



CaloPointFlow

Learn each point separately

- Point Flow transforms each point independently
- $g^{-1}(x_i) = y_i$
- The flow is independent of the source of the points, and therefore, the shower from which they come

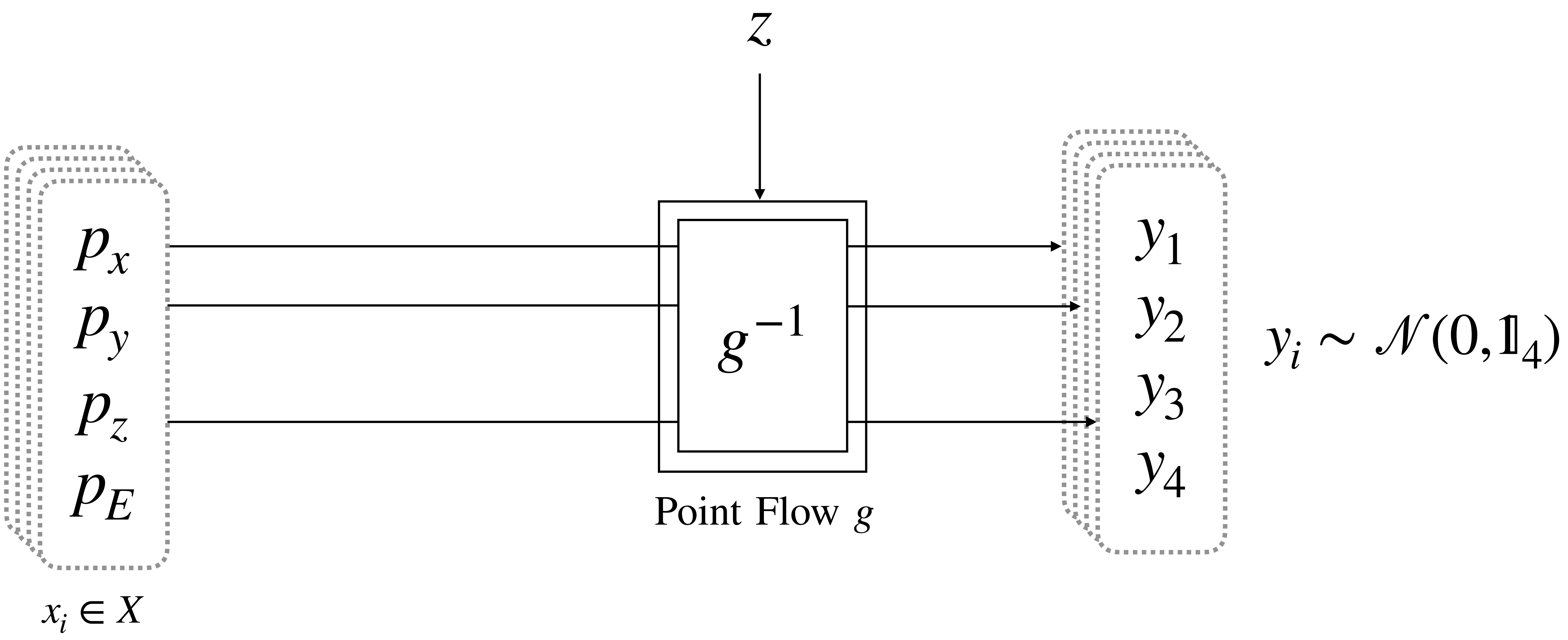


CaloPointFlow

Learn each point separately

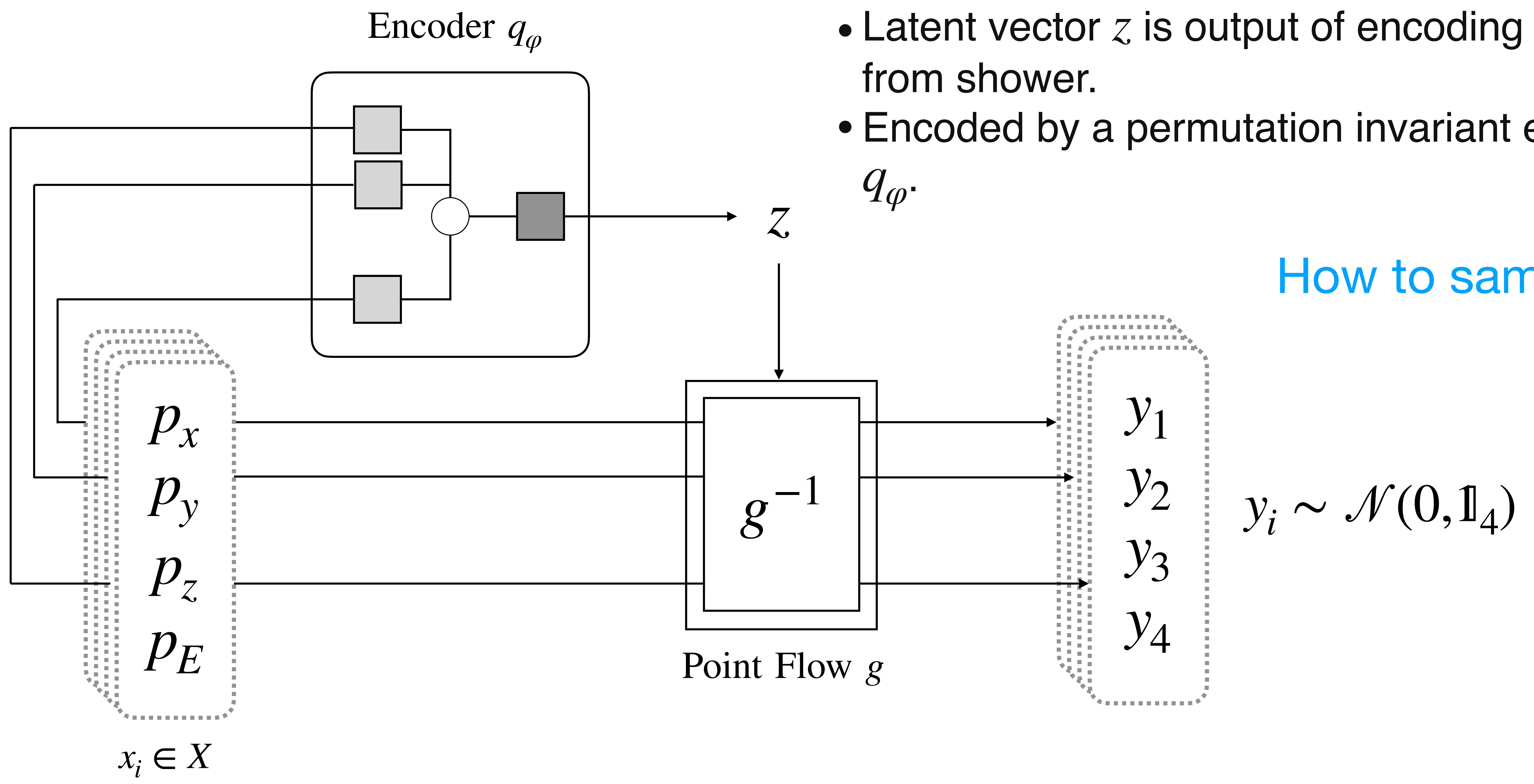
- Latent variable z contains the shower information
- Point Flow is conditioned on z

How we get z?



CaloPointFlow

Learn each point separately

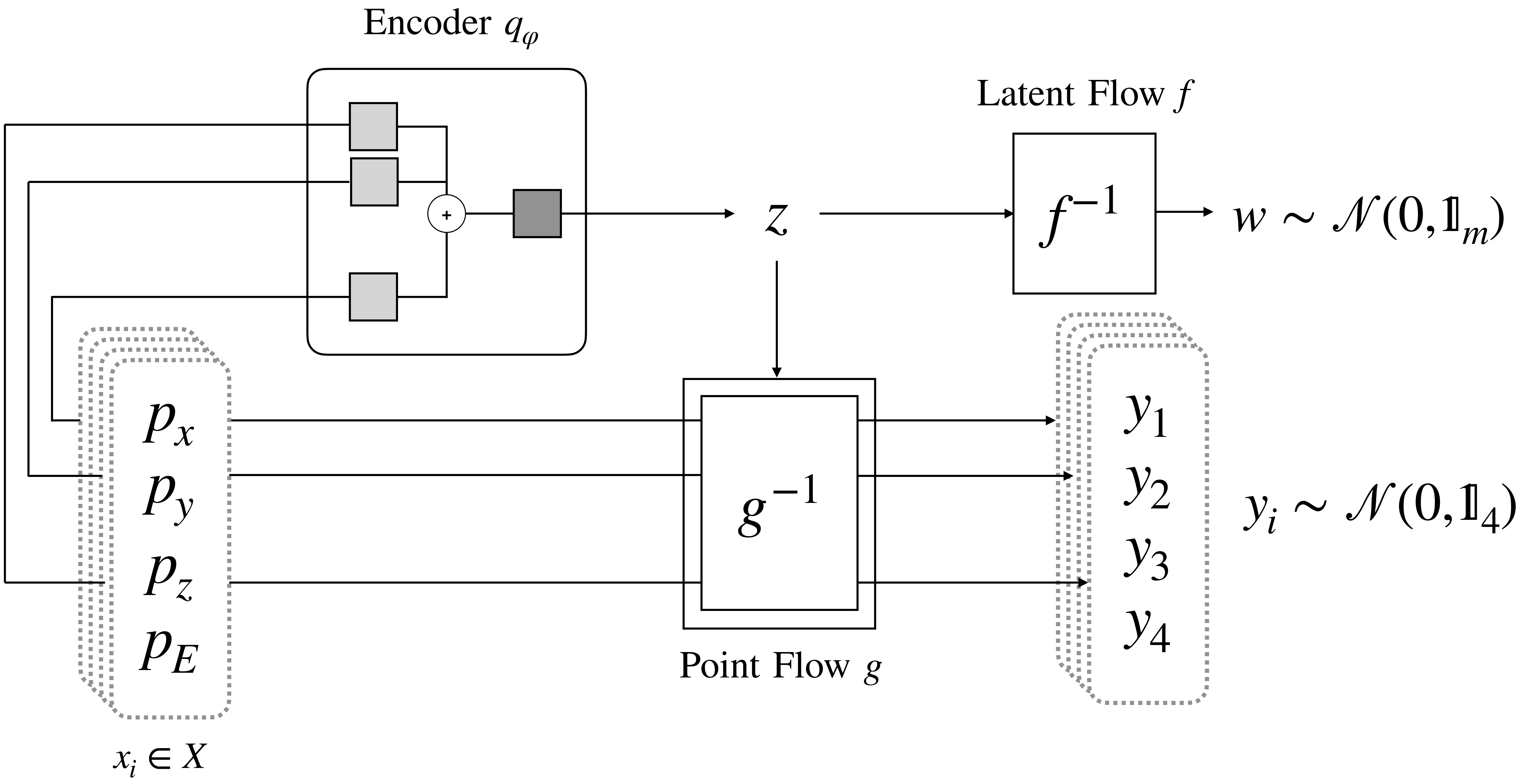


- Latent vector z is output of encoding all points from shower.
- Encoded by a permutation invariant encoder q_φ.

How to sample z?

CaloPointFlow

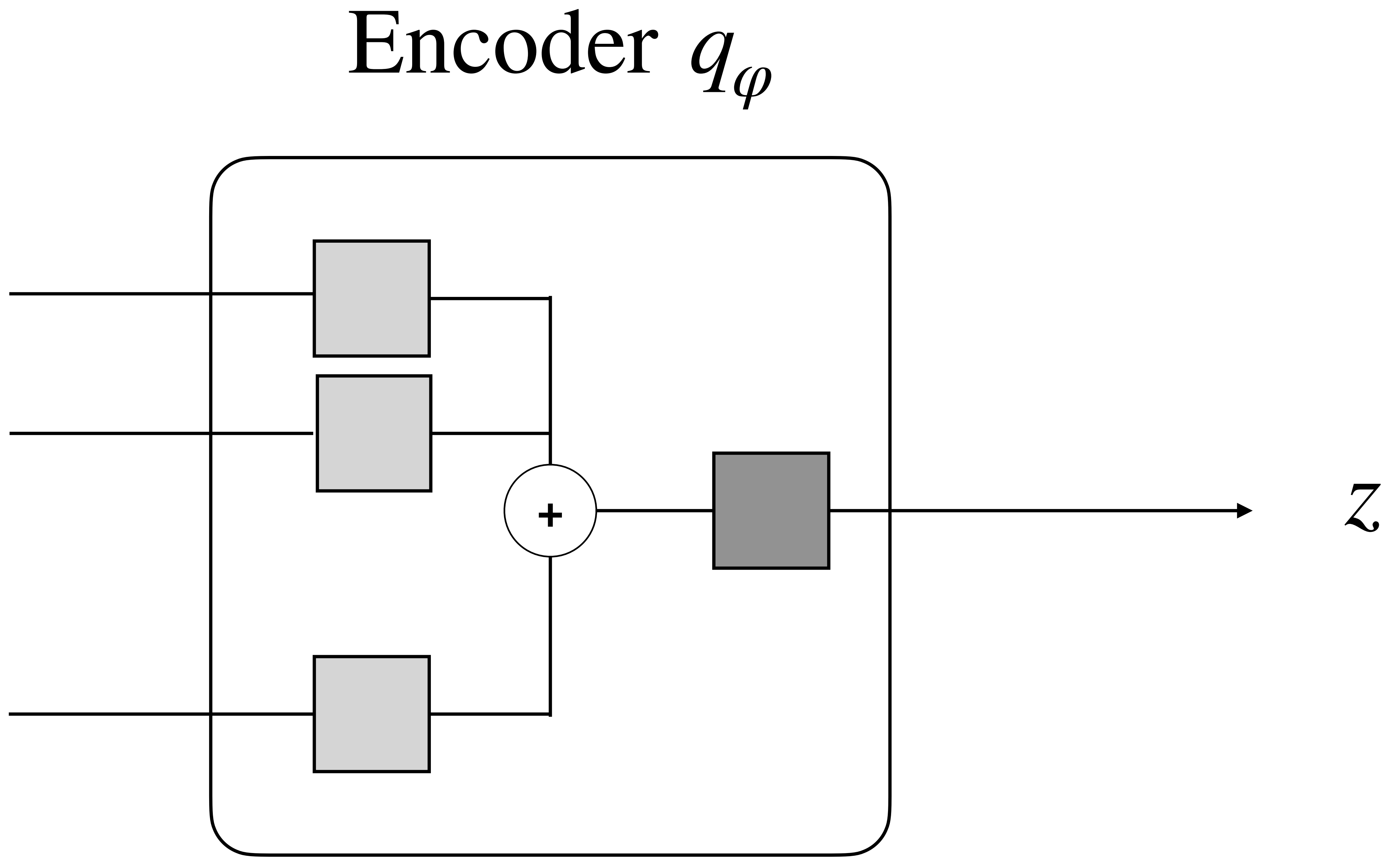
Learn each point separately



CaloPointFlow

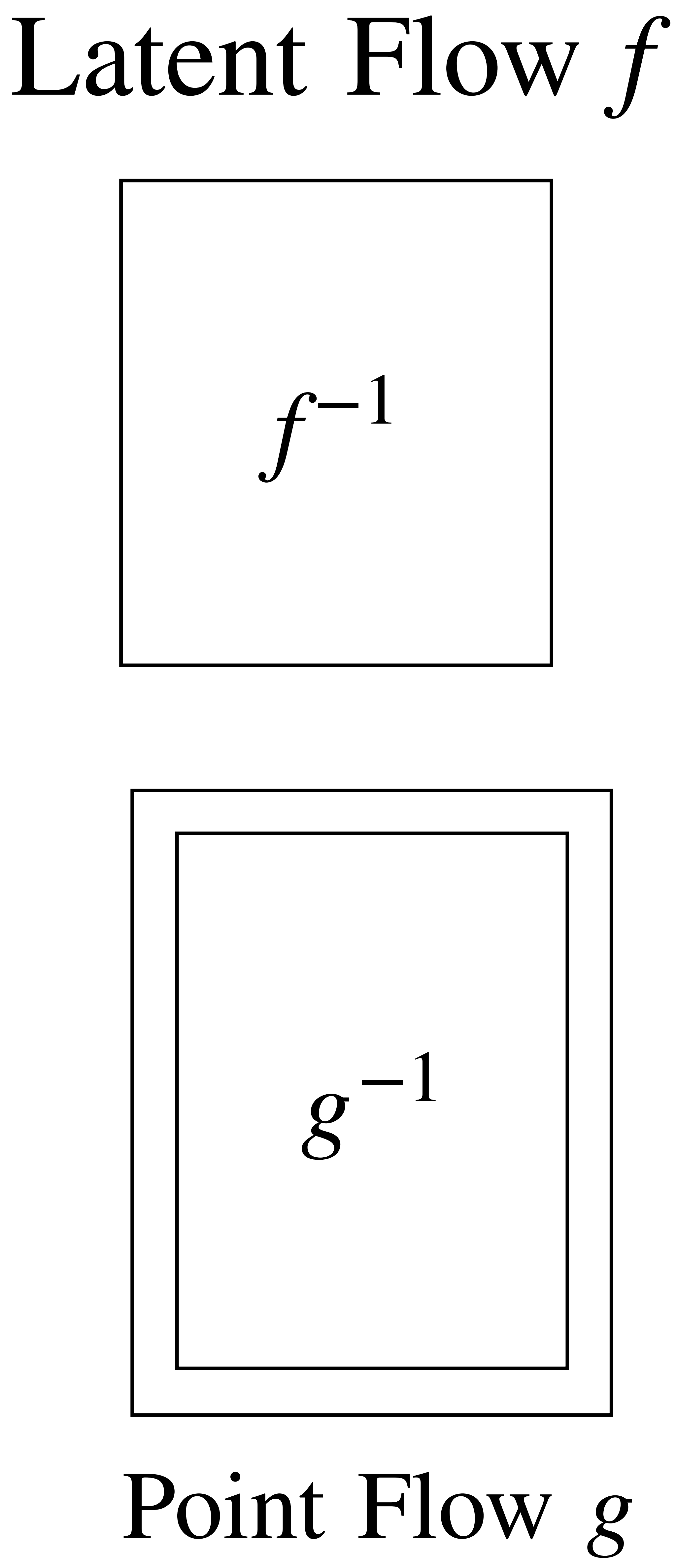
Encoder q_φ

- Encoder q_φ is permutation invariant
- Transform each point to a higher dim. space
- Average over all points in higher space
- Transform averaged higher space to latent space z
- Based on Deep Sets [arxiv:1703.06114]



CaloPointFlow
Flows

- Both flows are rational quadratic spline autoregressive flows
- Latent flow f is conditioned on $E_{\text{in}}, n_{\text{hits}}, E_{\text{sum}}$
- Point flow g is conditioned on $z, E_{\text{in}}, n_{\text{hits}}, E_{\text{sum}}$



Loss Function

Can be derived from the ELBO

$$\mathscr{L} = \mathbb{E}_{q_\varphi(z|X)}\left[\sum_{x_i \in X} \ln p_\theta(g(x,z)) + \ln\left|\det \frac{\partial g(x,z)}{\partial x}\right|\right] + \mathbb{E}_{q_\varphi(z|X)}\left[\ln p_\theta(f(z)) + \ln\left|\det \frac{\mathrm{d}f(z)}{\mathrm{d}z}\right|\right] - \mathscr{H}(q_\varphi(z\,|\,X))$$

Point Flow loss $\mathscr{L}_{\text{point}}$

Latent Flow loss $\mathscr{L}_{\text{latent}}$

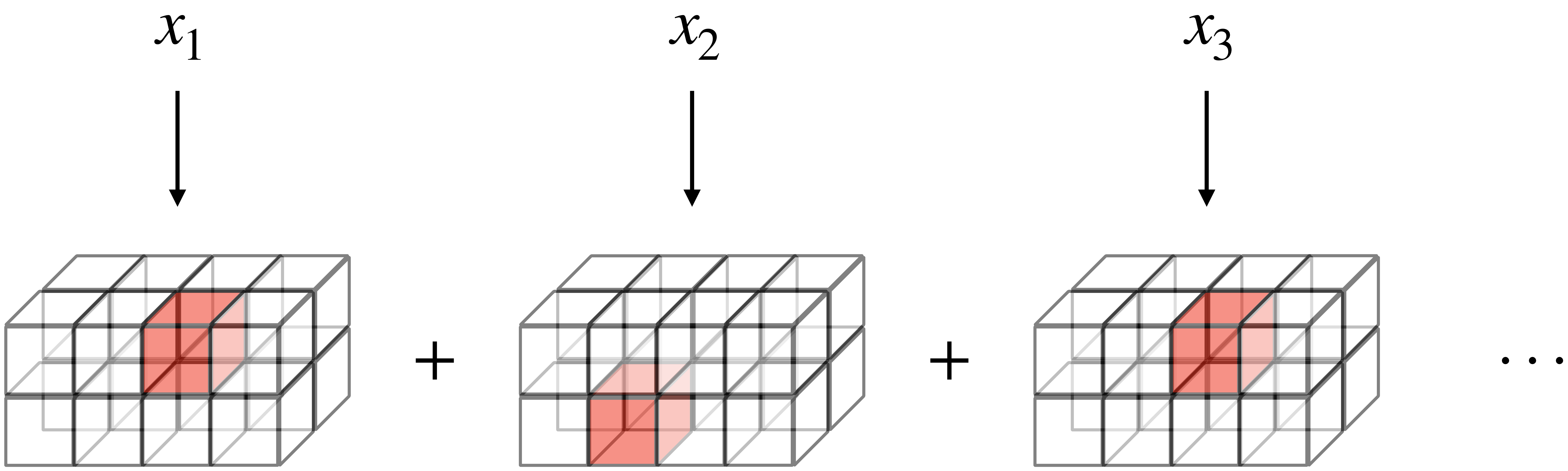
entropy loss $\mathscr{L}_{\text{entr}}$

$$\mathscr{L} = \mathscr{L}_{\text{point}} + \mathscr{L}_{\text{latent}} + \mathscr{L}_{\text{entr}}$$

Sampling

Two problems

- Number of points not defined by CaloPointFlow
- Multiple generated points can belong to the same calorimeter cell



Sampling

- Sample z from latent flow f conditioned on $E_{\text{in}}, n_{\text{hits}}, E_{\text{sum}}$
- Sample points from point flow g conditioned on $z, E_{\text{in}}, n_{\text{hits}}, E_{\text{sum}}$
- Post-process points to cell coordinate and E_f
- Continue sampling until we have n_{hits} different hit cells
- Overwrite previously hit cells
- Scale energy back

$$E = \frac{E_{\text{sum}} \cdot E_f}{\sum E_f}$$

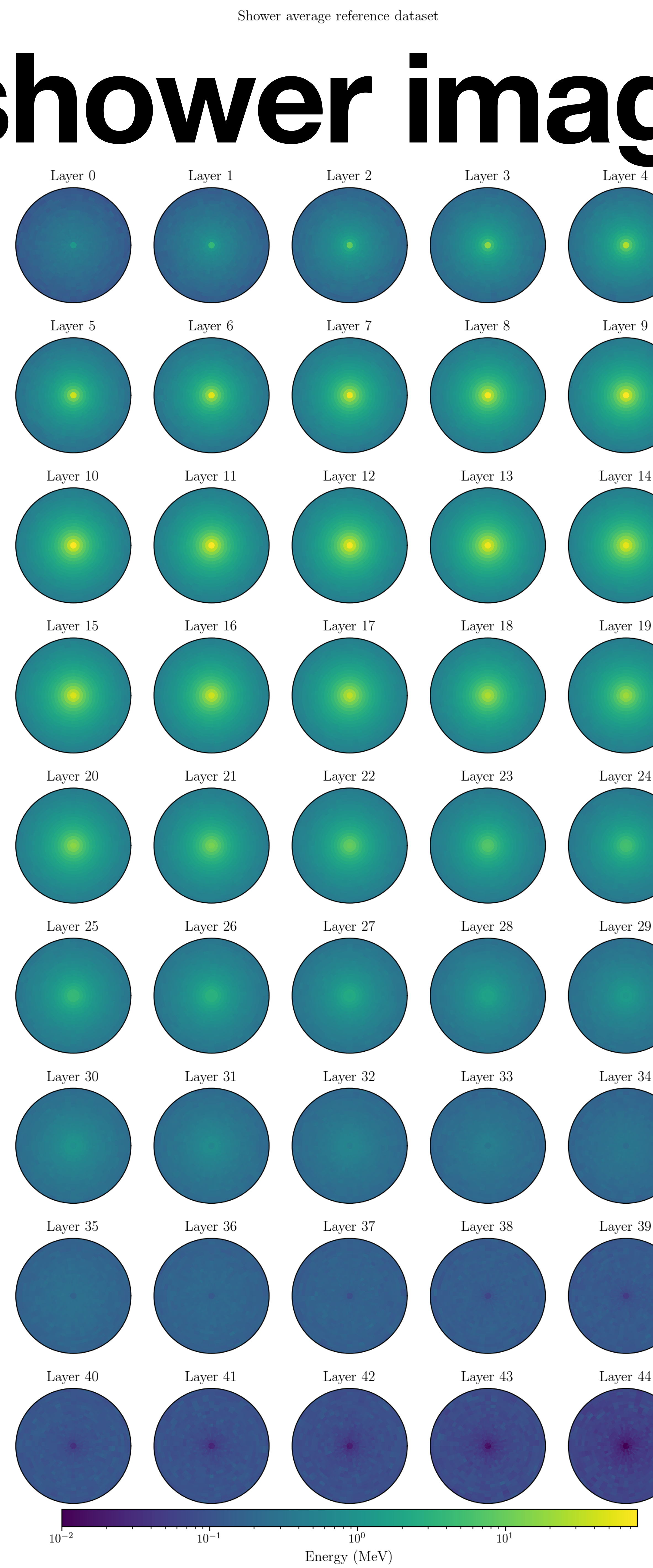
Evaluation

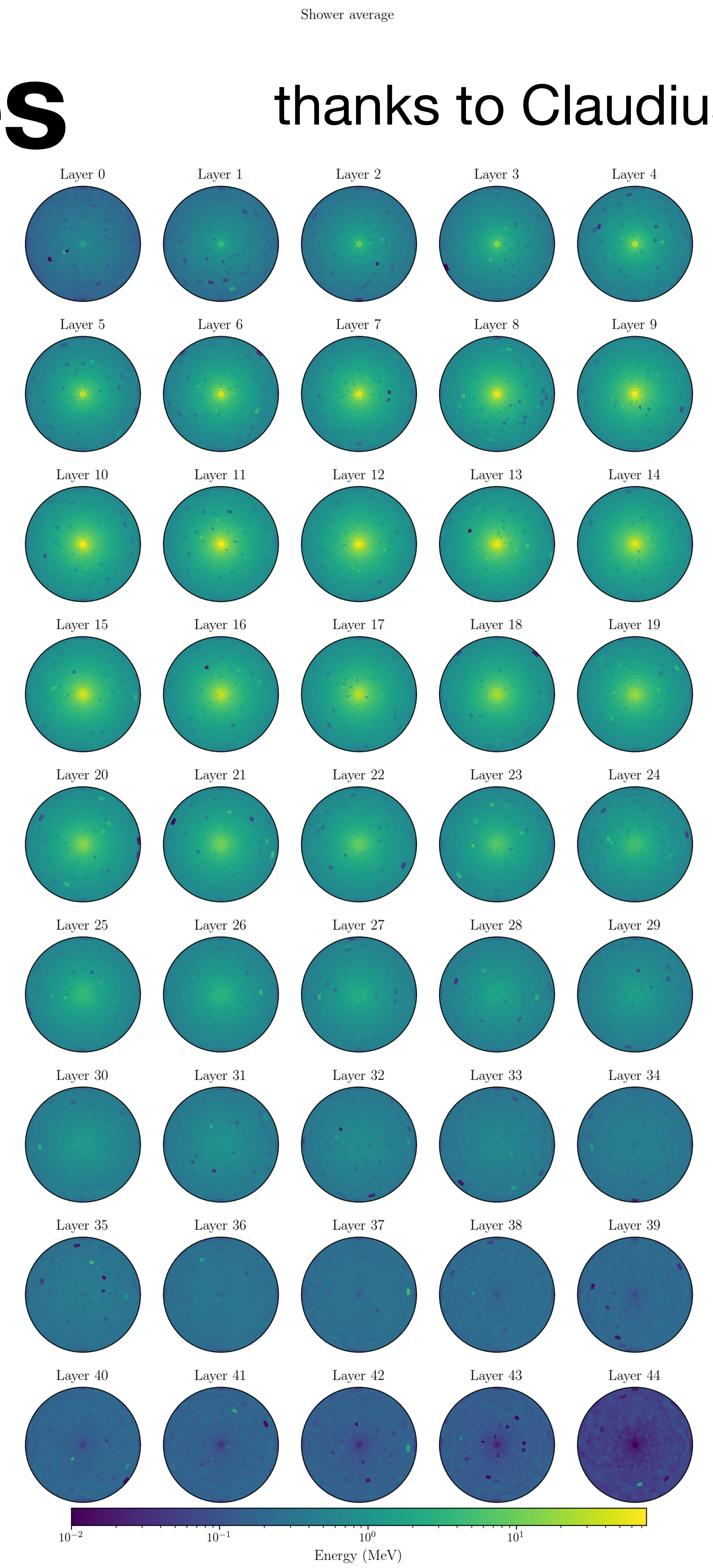
- We show results for CaloChallenge Dataset 3
- All results for Dataset 2 are in the appendix and are very similar
- Dataset 1 pions and photons has been generated
  but there are no evaluations ready

Average shower images

thanks to Claudius for the nice visualization

Geant4

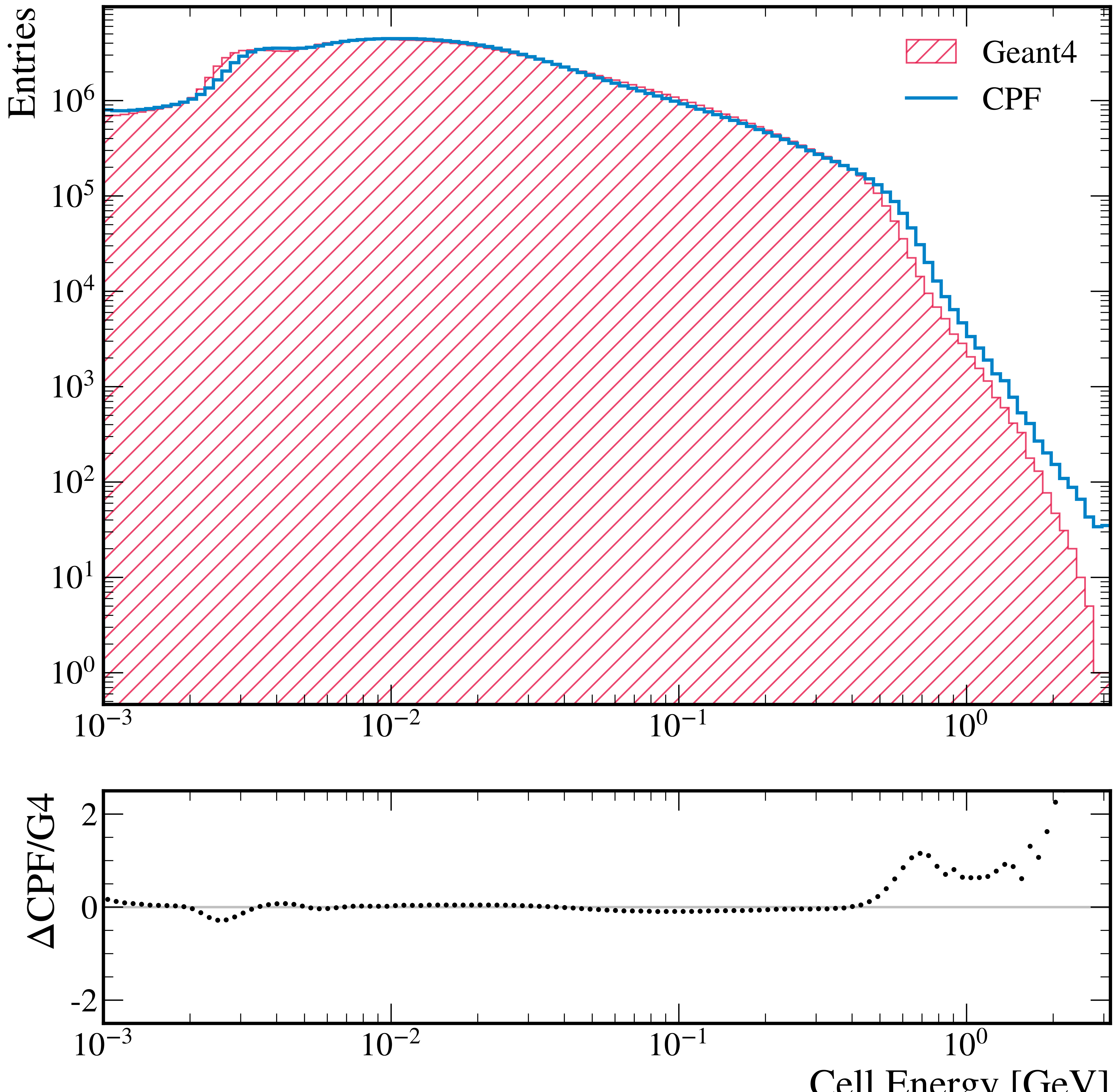




CPF

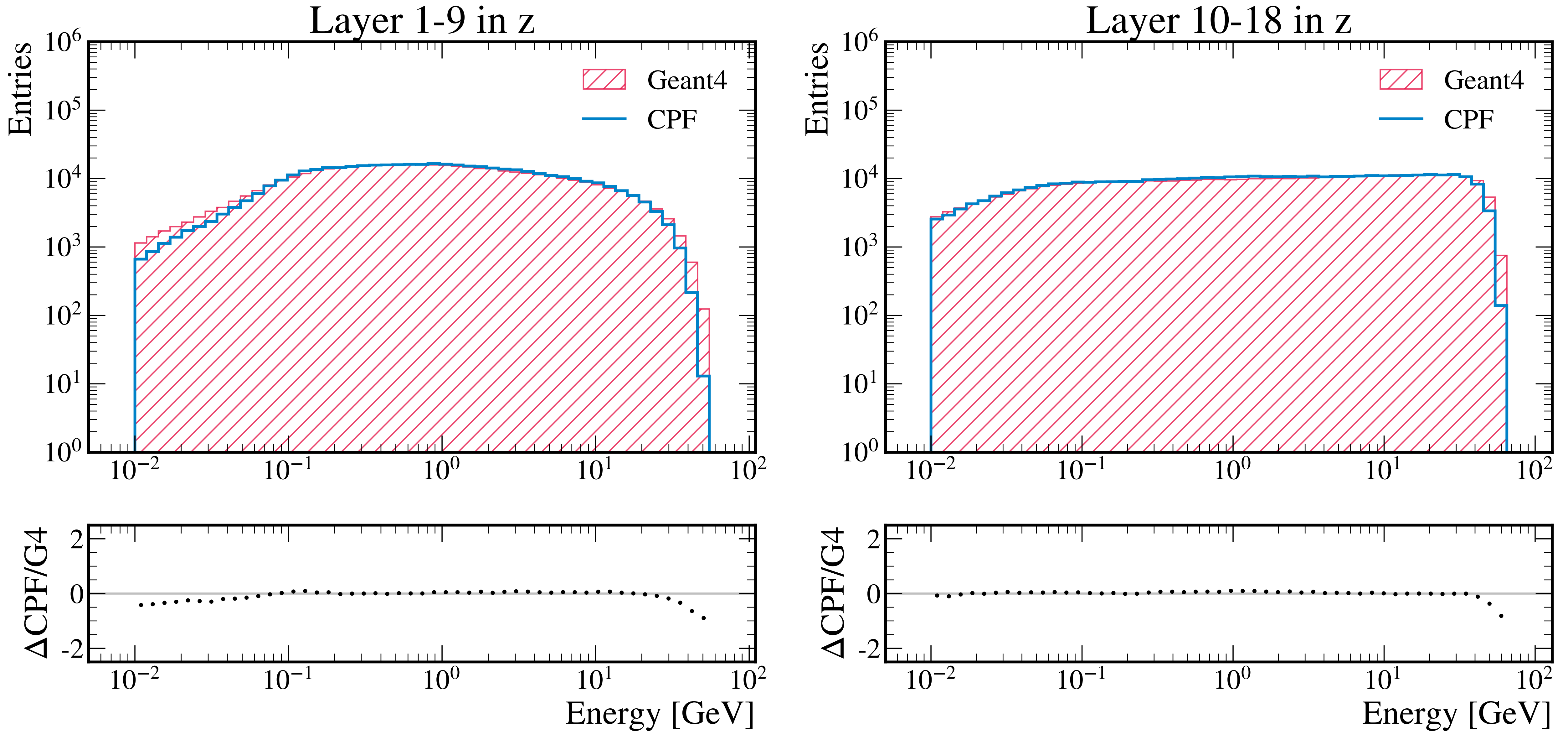
Cell Energy Distribution

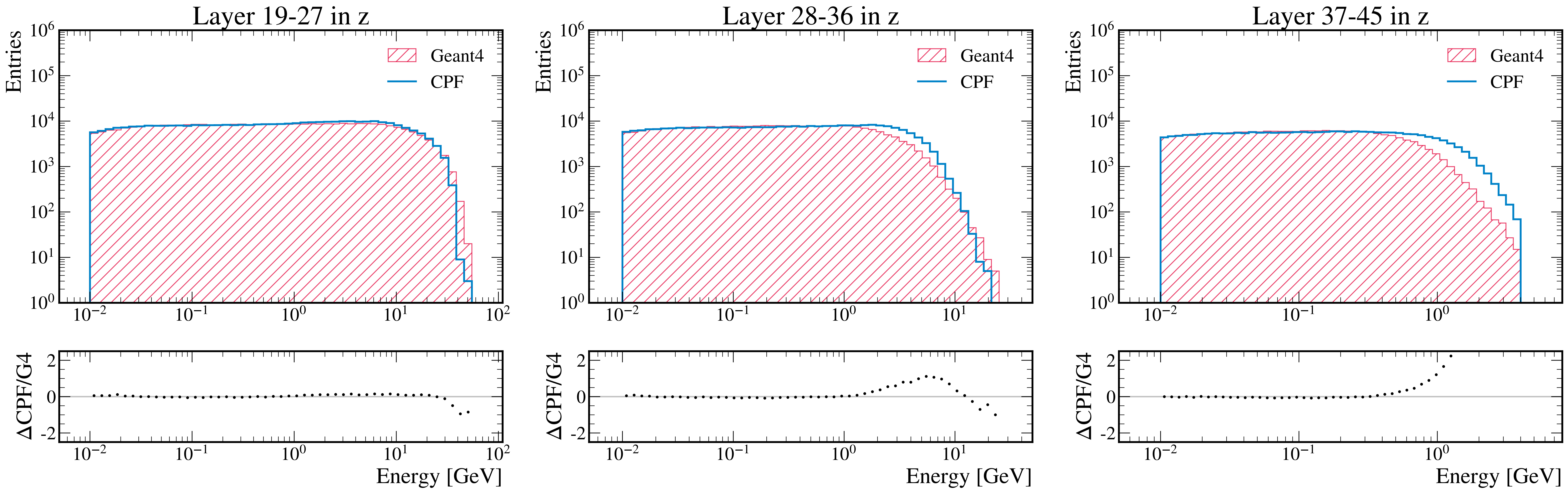
- Agreement in high statistics area
- Differences in tails



Energy Distribution in different layer areas

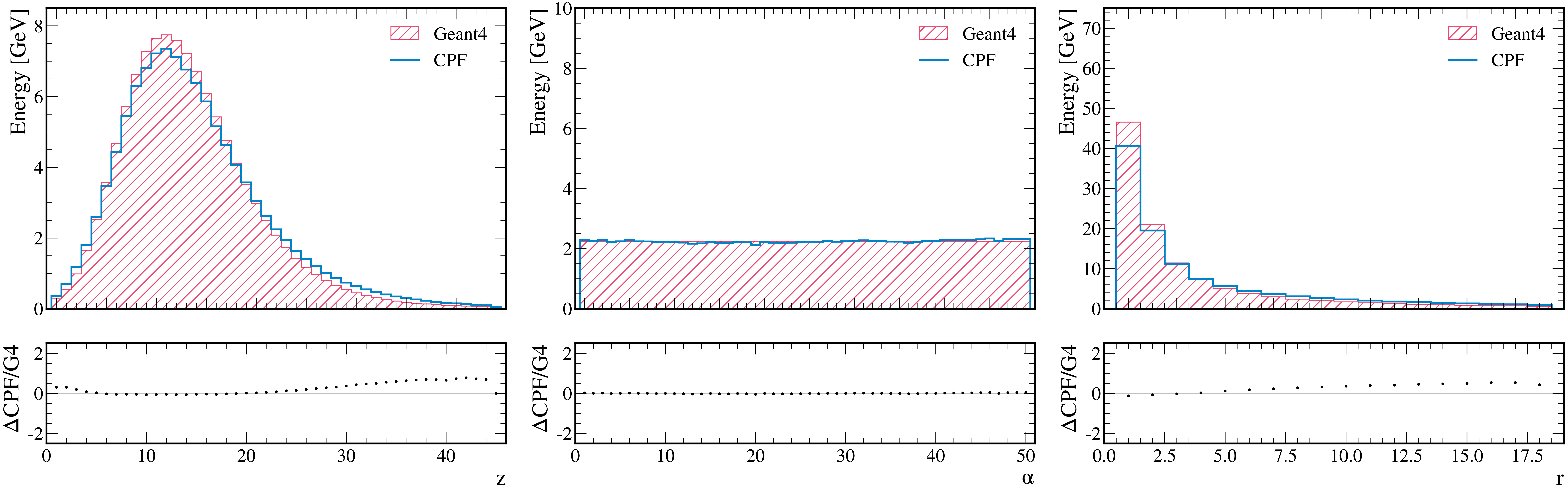
- Overall good agreement
- Also problems in tails





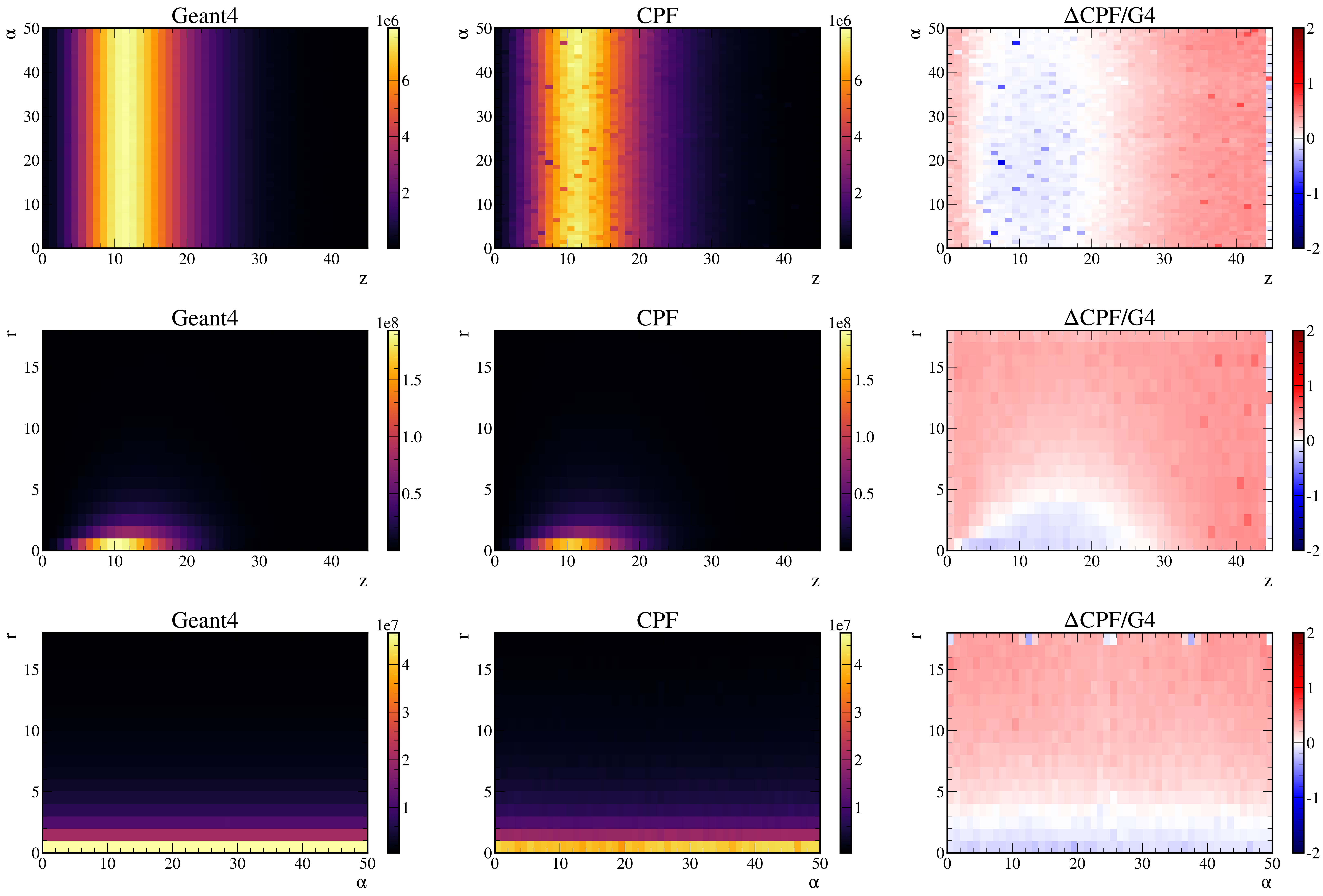
Shower profiles

- To low energy in center
- To high energy in tails



Shower profiles in 2D

- No structural differences
- High density too low
- Low density too high

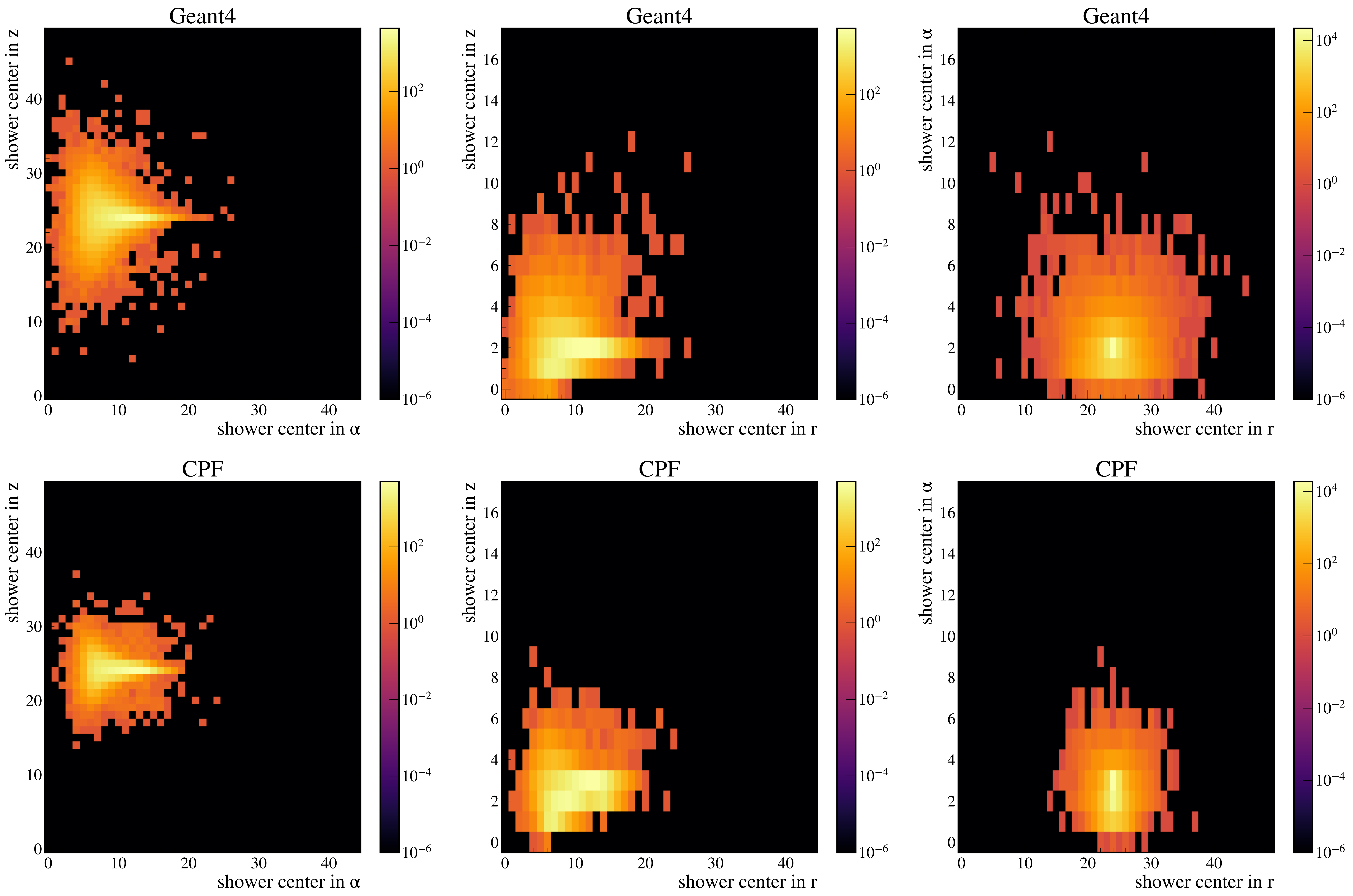


Shower means

- Agreement wit in z and α with small differences.
- Huge shift in r. Overall the shower have a too large radial distributions.

Shower means in 2D

- Same features
- structural morphing



Eigenvalues of covariance matrix

- Calculated unbiased energy weighted sample covariance matrix for each shower

$$C = \frac{1}{\sum_{i=1}^{n} E_i - 1} \sum_{i=1}^{n} E_i (x_i - \mu^*)^T (x_i - \mu^*) \text{ with } \mu^* = \frac{1}{\sum_{i=1}^{n} E_i} \sum_{i=1}^{n} E_i x_i$$

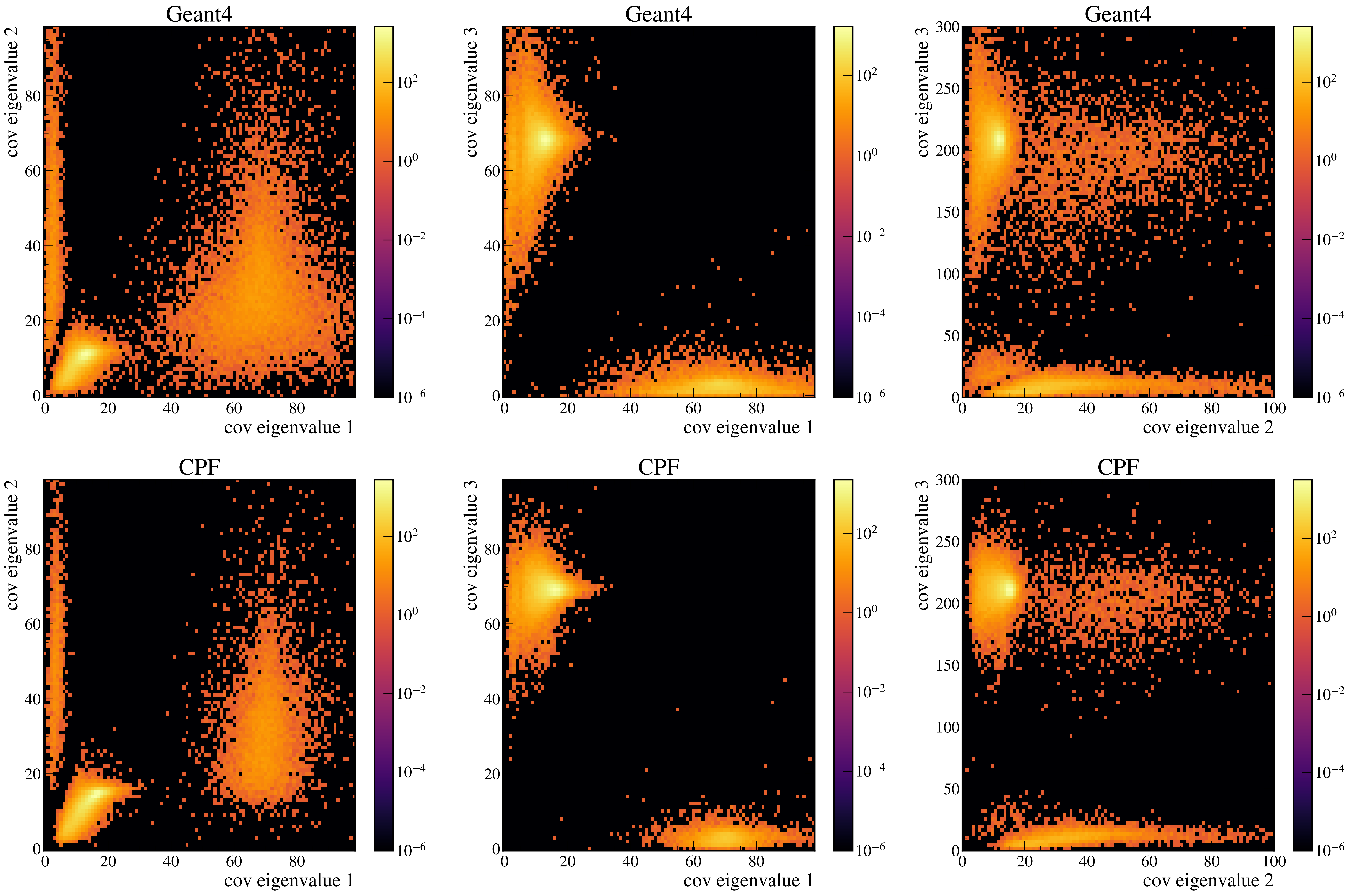
- Eigenvalue decomposition of C give as the widths of the shower base on the principal components of the shower

Eigenvalues of covariance matrix

- Structural agreement between Geant and CPF
- Shifts and differences visible

Eigenvalues of covariance matrix 2D

- Same sub-distributions visible
- Structural morphing visible
- Good proxy look of the differences between CPF and G4 shower



Classifier Scores and Sampling Time

- CaloPointFlow does not pass Claudius classifier test

CaloChallenge Classifier	low	low-normed	high
AUC	0.9868	0.9854	0.9664
JSD	0.8006	0.7765	0.6656

- Relativ fast sampling time (including multiple sampling due to double hits)

number showers	sampling time	time per shower
50,000	548.26s	10.9652ms

- Also fast training time (≈ 5min/epoch)

Conclusion & Outlook

- Interpret calorimeter showers as point clouds
- Tested the possibilities of a linear model without point-to-point relations
- Can handle high granular datasets
- Shower structure is overall good resembled
- Their are some structural deviations
- Possible future research areas are
  - including point to point correlations
  - refine the output with a model that introduces point to point relations
- Next steps
  - Analyse results of dataset 1

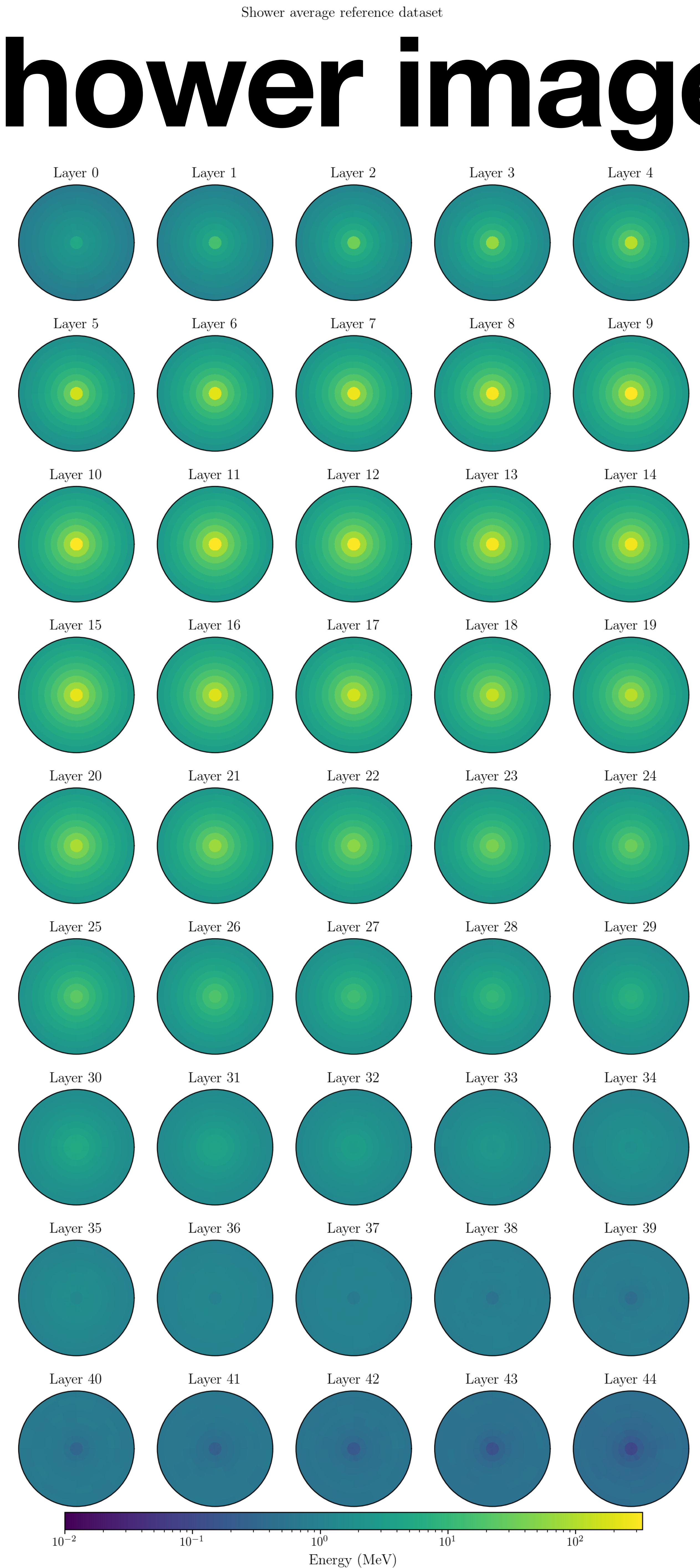
BACKUP

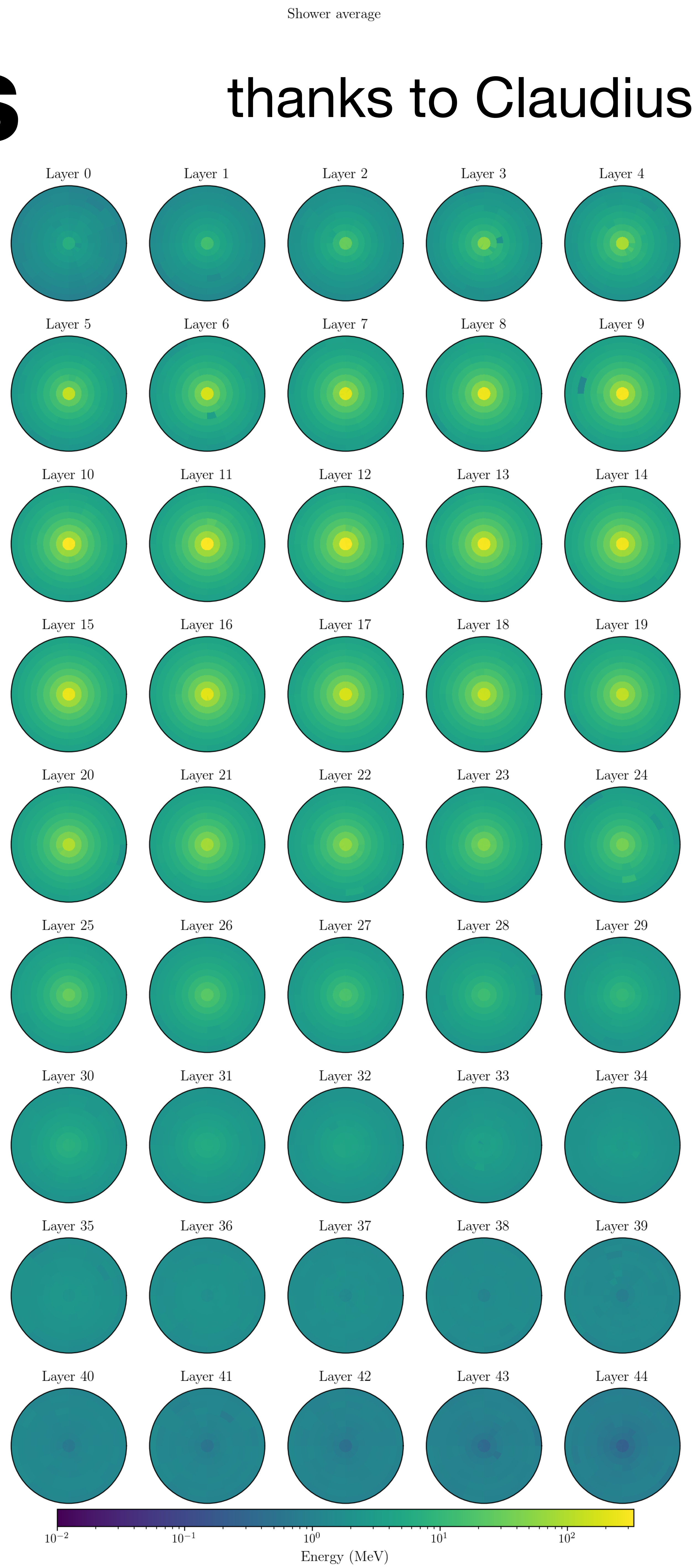
DATASET 2 PLOTS

Average shower images

thanks to Claudius for the nice visualization

Geant4

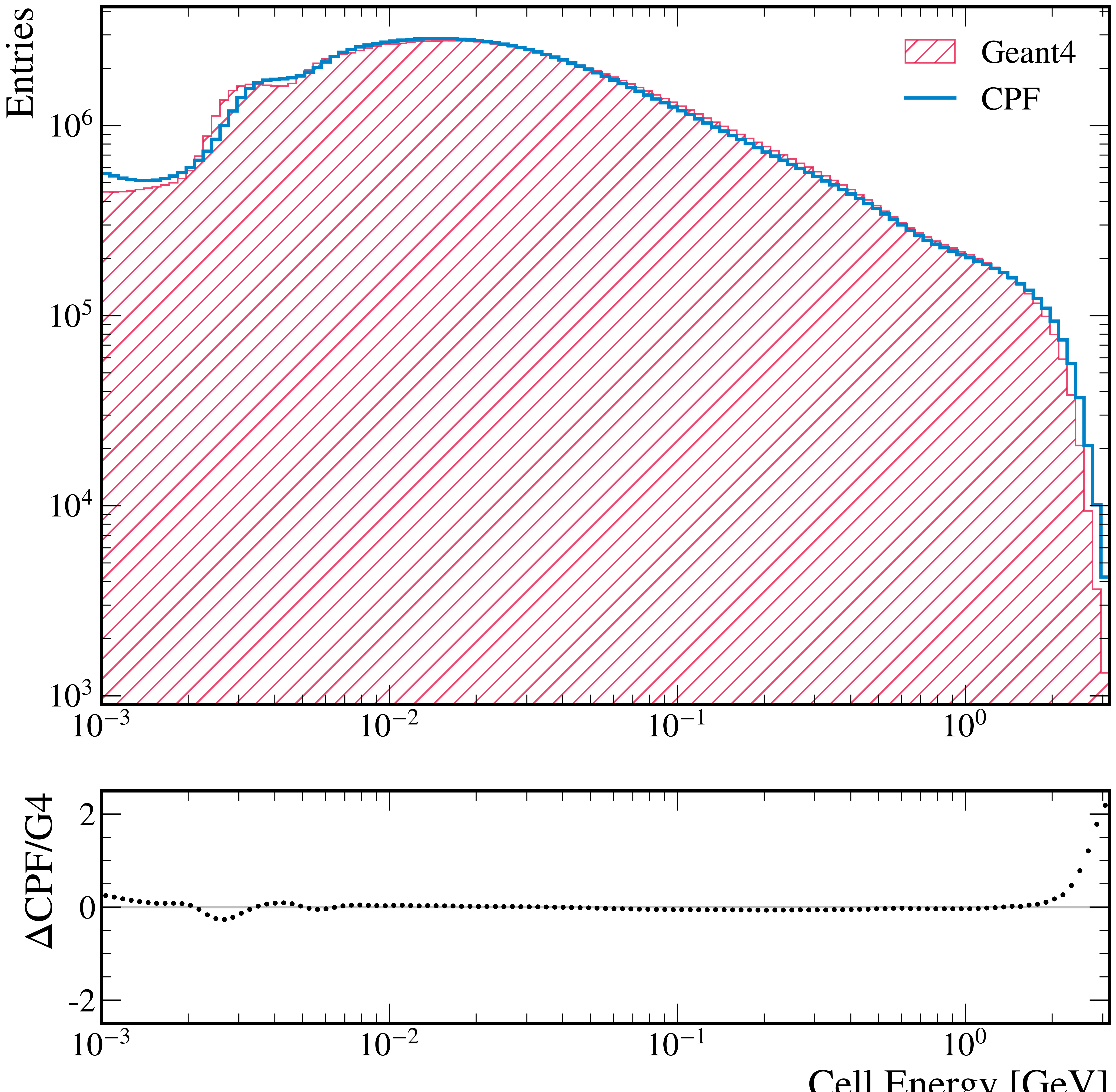




CPF

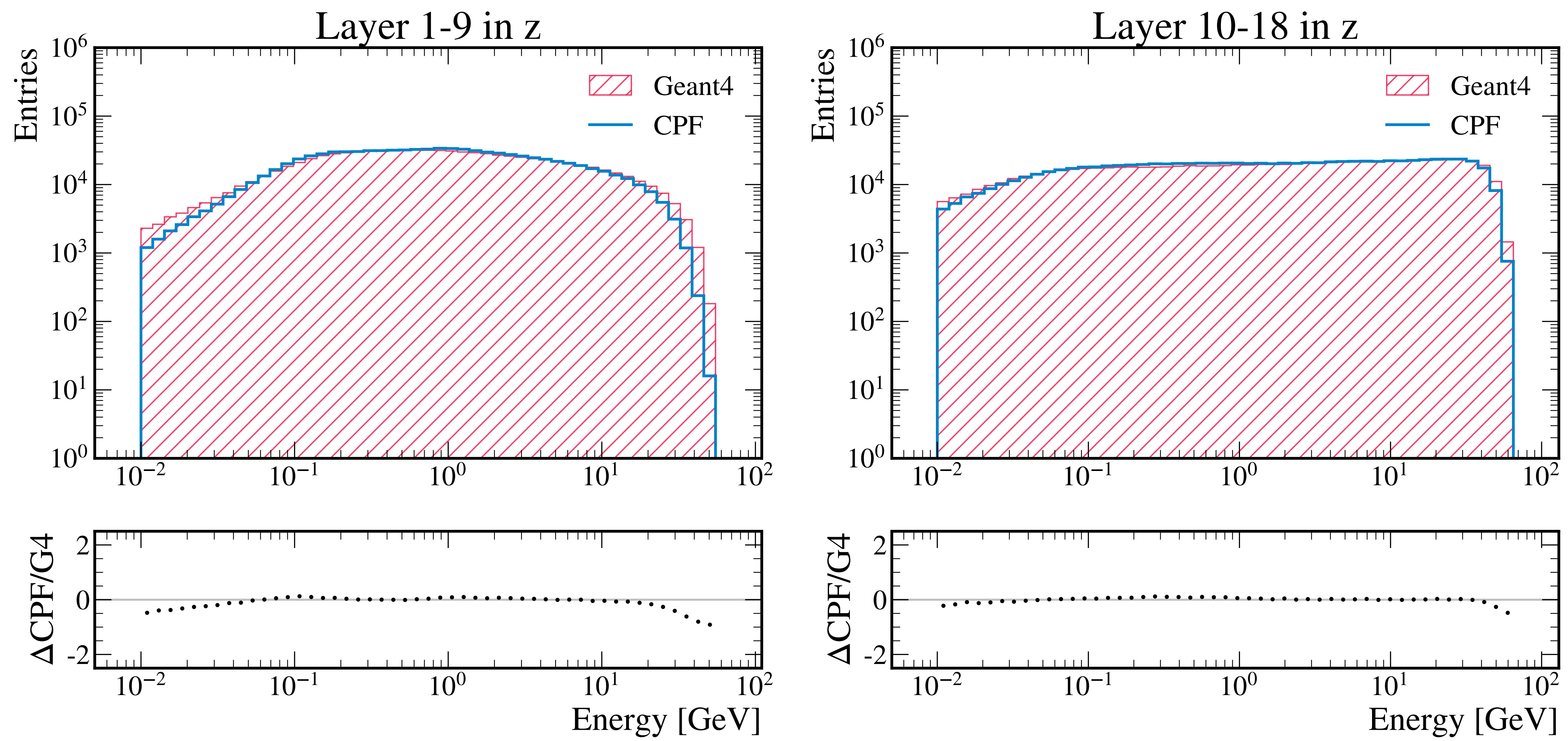
Cell Energy Distribution

- Agreement in high statistics area
- Differences in tails



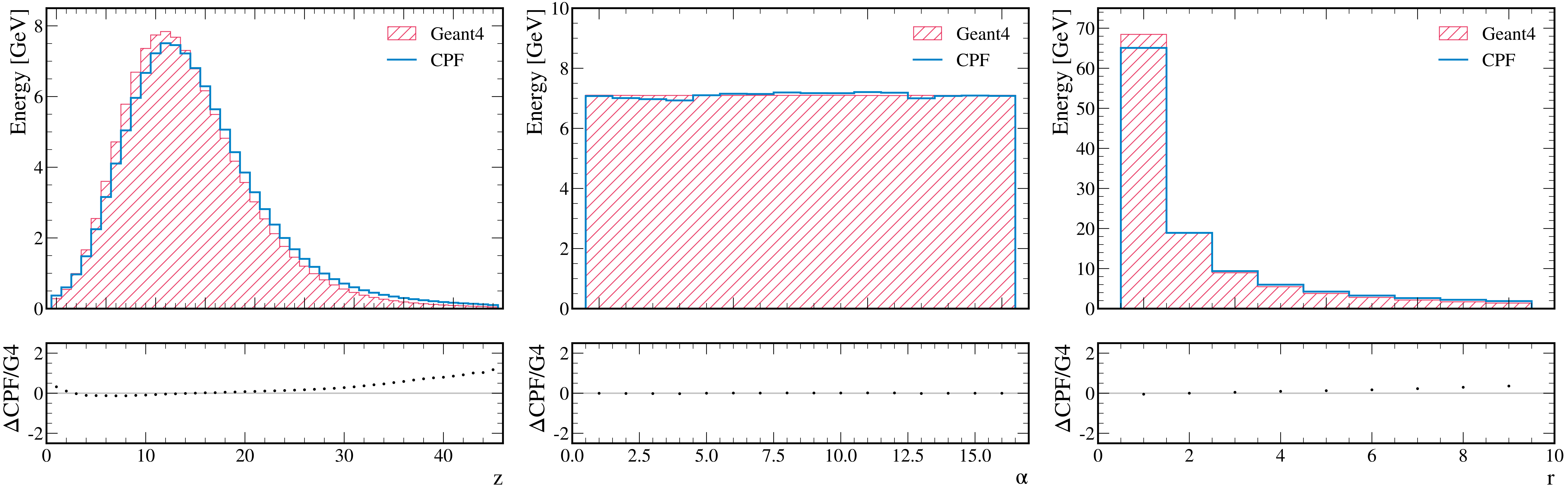
Energy Distribution in different layer areas

- Overall good agreement
- Also problems in tails



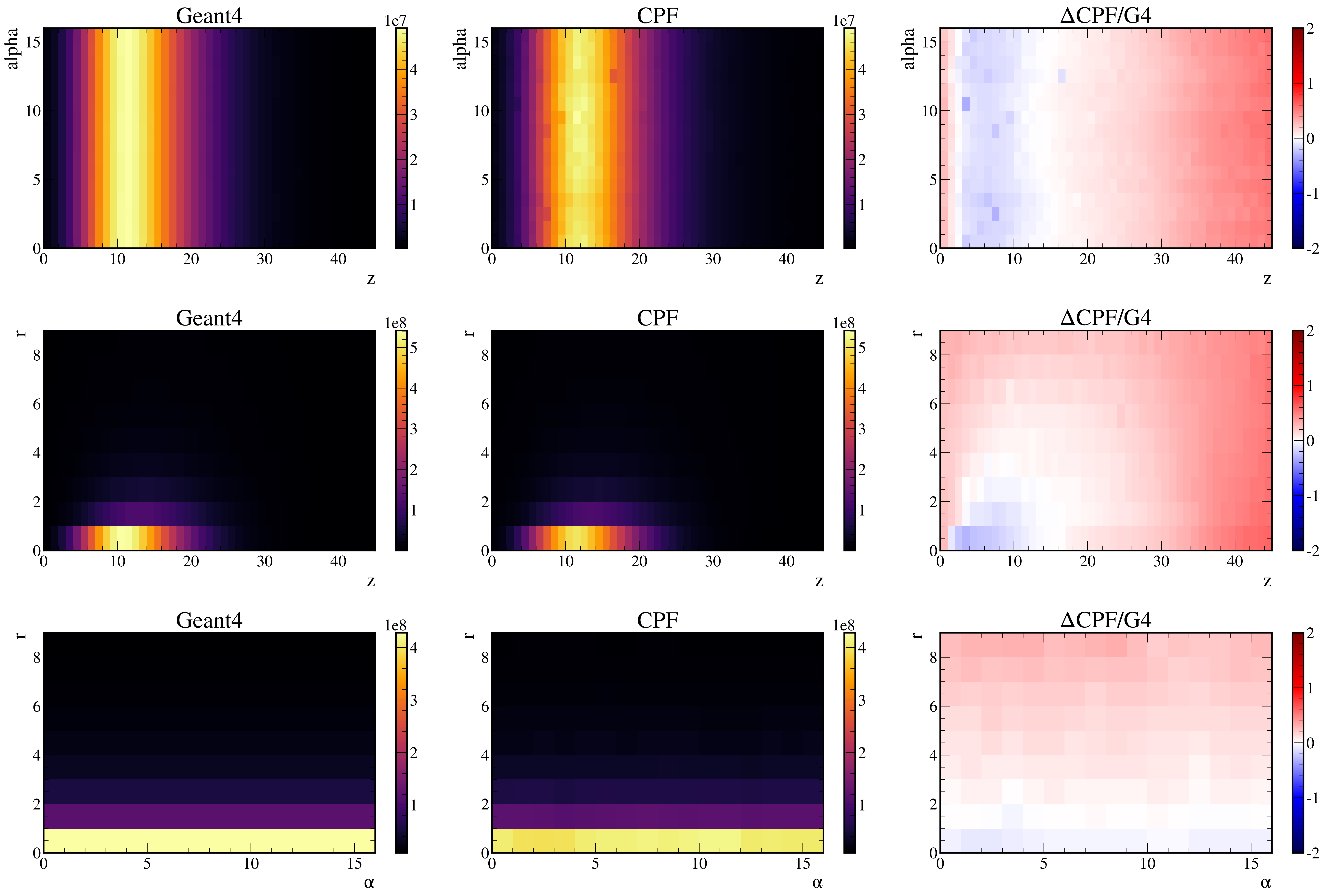
Shower profiles

- To low energy in center
- To high energy in tails



Shower profiles in 2D

- No structural differences
- High density too low
- Low density too high

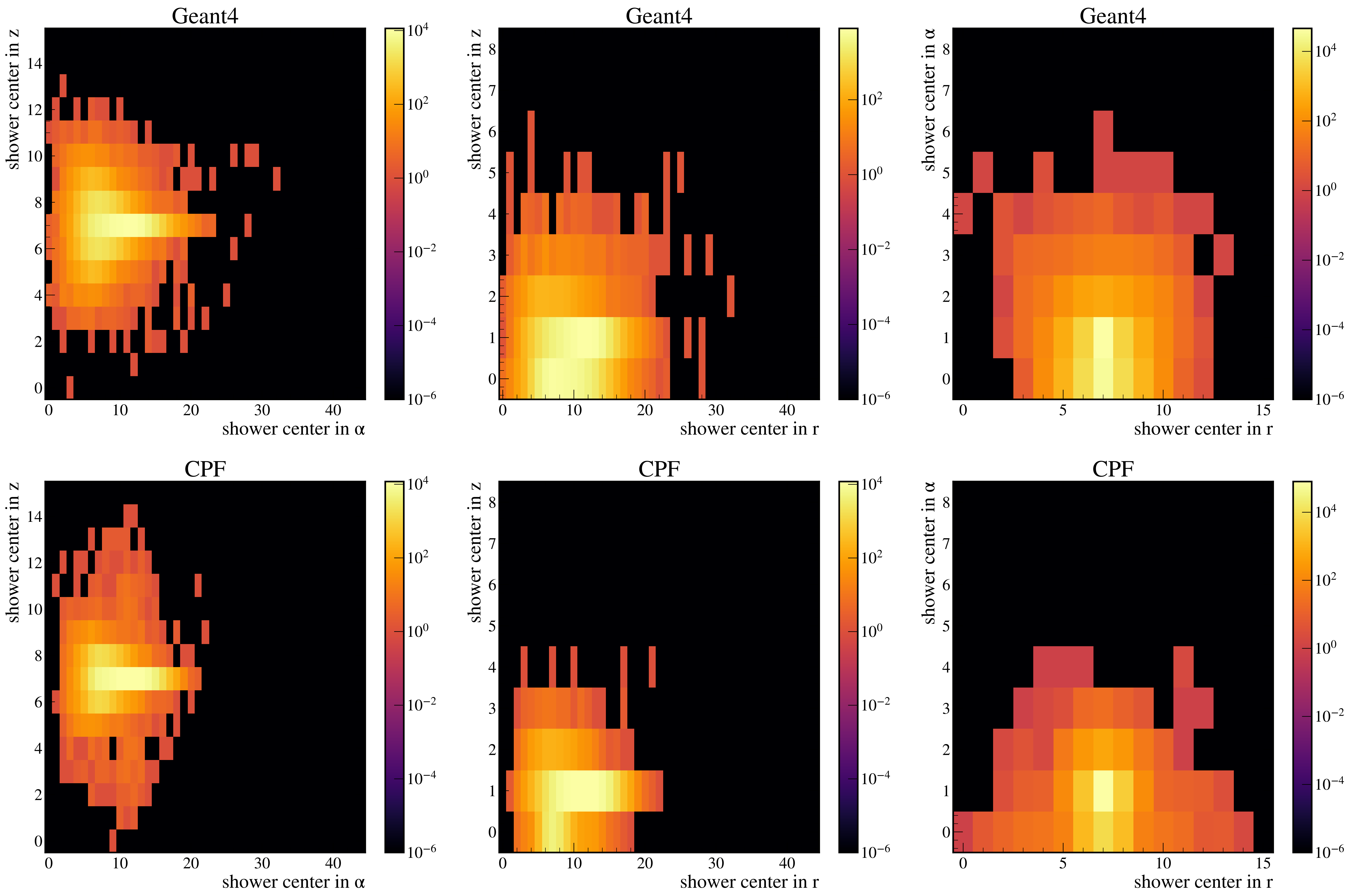


Shower means

- Agreement wit in z and α with small differences.
- Huge shift in r. Overall the shower have a too large radial distributions.

Shower means in 2D

- Same features
- structural morphing

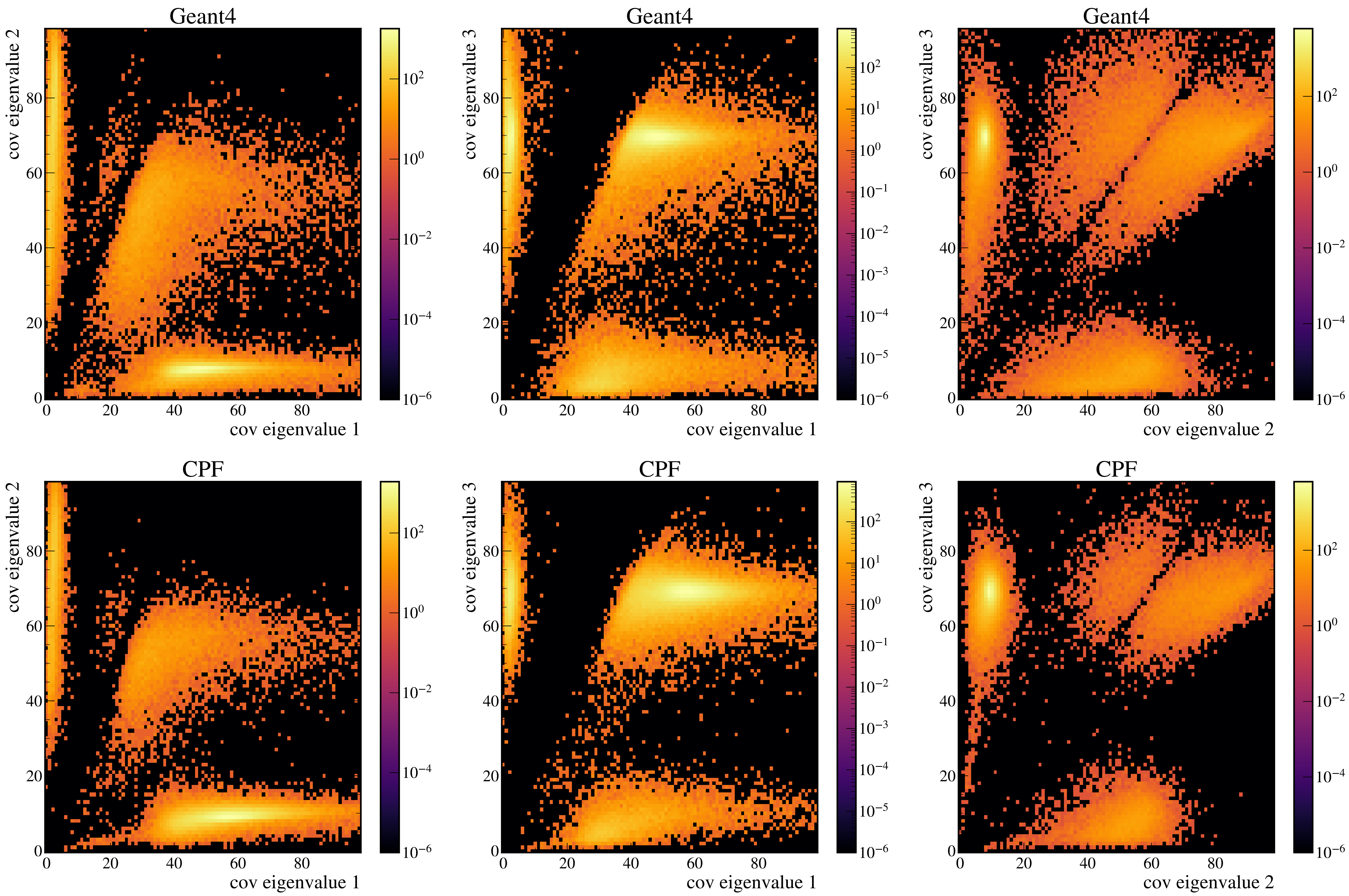


Eigenvalues of covariance matrix

- Structural agreement between Geant and CPF
- Shifts and differences visible

Eigenvalues of covariance matrix 2D

- Same sub-distributions visible
- Structural morphing visible
- Good proxy look of the differences between CPF and G4 shower

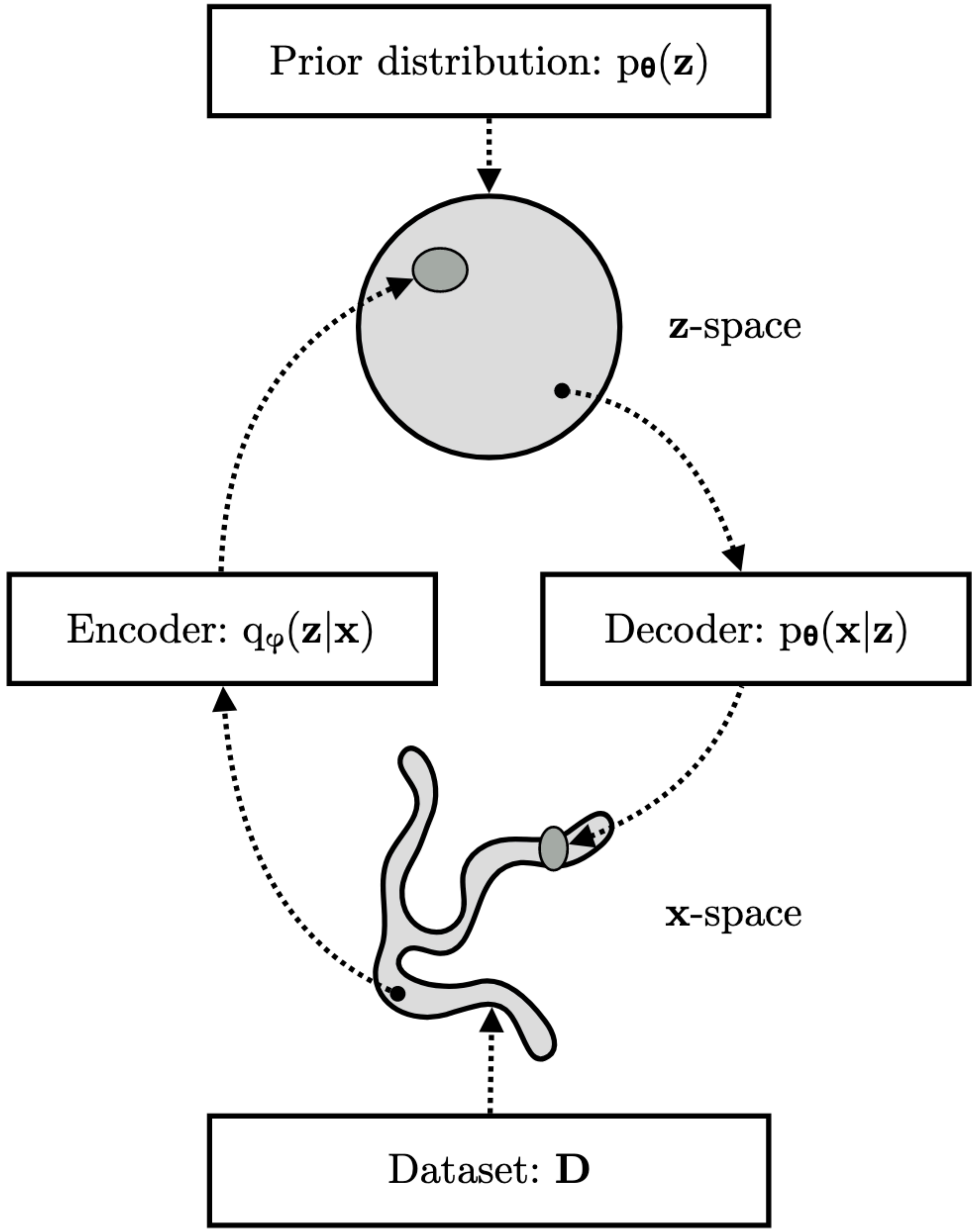


VAE
Variational Autoencoders

ELBO $\mathscr{L} = \mathbb{E}_{q_\varphi(z|x)}[\ln p_\theta(x|z)] - D_{KL}(q_\varphi(z|x)||p_\theta(z))$

- If we assume that the data is gaussian distributed the first term is the MSE and the last term is a regularisation that keeps the latent gaussian
- The Encoder predicts (μ, σ)
- To a differentiable point is sampled by $z = \mu + \epsilon \odot \sigma$.

here $\epsilon \sim \mathrm{N}(0,1)$ (reparametrization trick)



Durk Kingma PhD Thesis

Encoding

VAE with an NF Prior

ELBO $\mathscr{L} = \mathbb{E}_{q_\varphi(z|X)}[\ln p_\theta(X|z)] - D_{KL}(q_\varphi(z|X)||p_\theta(z)) = \mathbb{E}_{q_\varphi(z|X)}[\ln p_\theta(X|z) + \ln p_\theta(z) - \ln q_\varphi(z|X)]$

Bijective transformation (NF) $w = f(z)$ with $w \sim \mathrm{N}(0,1)$

$$
\begin{aligned}
\mathscr{L} &= \mathbb{E}_{q_\varphi(z|X)}\left[\ln p_\theta(X|z) + \ln p_\theta(z) - \ln q_\varphi(z|X)\right] \\
&= \mathbb{E}_{q_\varphi(z|X)}\left[\ln p_\theta(X|z) + \log p_\theta(f(z)) + \log\left|\det\frac{\mathrm{d}f(z)}{\mathrm{d}z}\right| - \ln q_\varphi(z|X)\right] \\
&= \mathbb{E}_{q_\varphi(z|X)}\left[\ln p_\theta(X|z)\right] + \mathbb{E}_{q_\varphi(z|X)}\left[\log p_\theta(f(z)) + \log\left|\det\frac{\mathrm{d}f(z)}{\mathrm{d}z}\right|\right] - \mathscr{H}(q_\varphi(z|X))
\end{aligned}
$$

Decoding

Using a second Normalizing Flow

$$\ln p_\theta(X|z) = \ln \prod_{x_i \in X} p_\theta(x_i|z) = \sum_{x_i \in X} \ln p_\theta(x_i|z)$$

(NF) $y_i = g(x_i, z)$ with $y_i \sim \mathrm{N}(0,1)$

$$\ln p_\theta(X|z) = \sum_{x_i \in X} \ln p_\theta(x_i|z)$$

$$= \sum_{x_i \in X} \ln p_\theta(g(x_i, z)) + \log \left| \det \frac{\partial g(x_i, z)}{\partial x} \right|$$

The Algorithm
How to tame the beast

for $t = 1,2,...,T$ **do**

$\mu, \sigma \leftarrow q_\varphi(X_t)$ where d is the dimension of μ

and X_t is a point cloud sample

$$\mathcal{L}_{\text{entr}} = \frac{d}{2}(1 + \ln(2\pi)) + \sum_{i=1}^{d} \ln \sigma_i$$

$z = \epsilon \odot \sigma + \mu$ (Reparametrization)

$w \leftarrow f(z)$

$$\mathcal{L}_{\text{prior}} = \mathrm{N}(w; 0, I) + \ln \left| \det \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|$$

$L \leftarrow 0$

for $x_i \in X_t$ **do**

$y_i \leftarrow g(x_i, z)$

$$L_i \leftarrow \log \mathrm{N}(y_i; 0, I) + \log \left| \det \frac{\partial g(x_i, z)}{\partial x} \right|$$

$L \leftarrow L + L_i$

end for

$$\mathcal{L}_{\text{recon}} = \frac{L}{n_{X_t}}$$

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{entr}}$$

Adam($-\mathcal{L}$)

end for