CaloPointFlow II

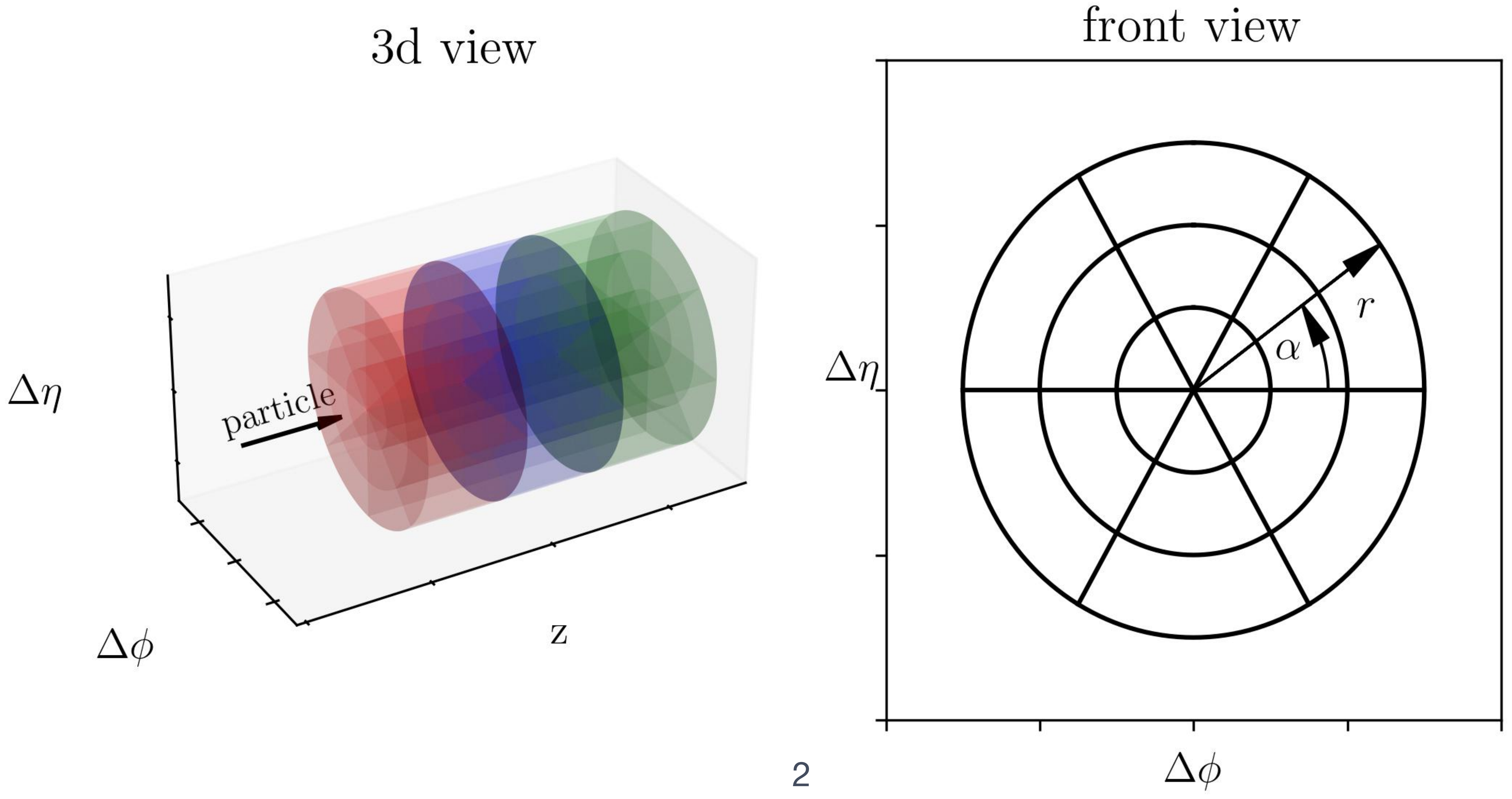
Updates for the CaloChallenge

Kerstin Borras, Dirk Krücker, Benno Käch,
Moritz Scham, Simon Schnake
07.11.23
ML4Jets 2023

CaloChallenge

Dataset 3

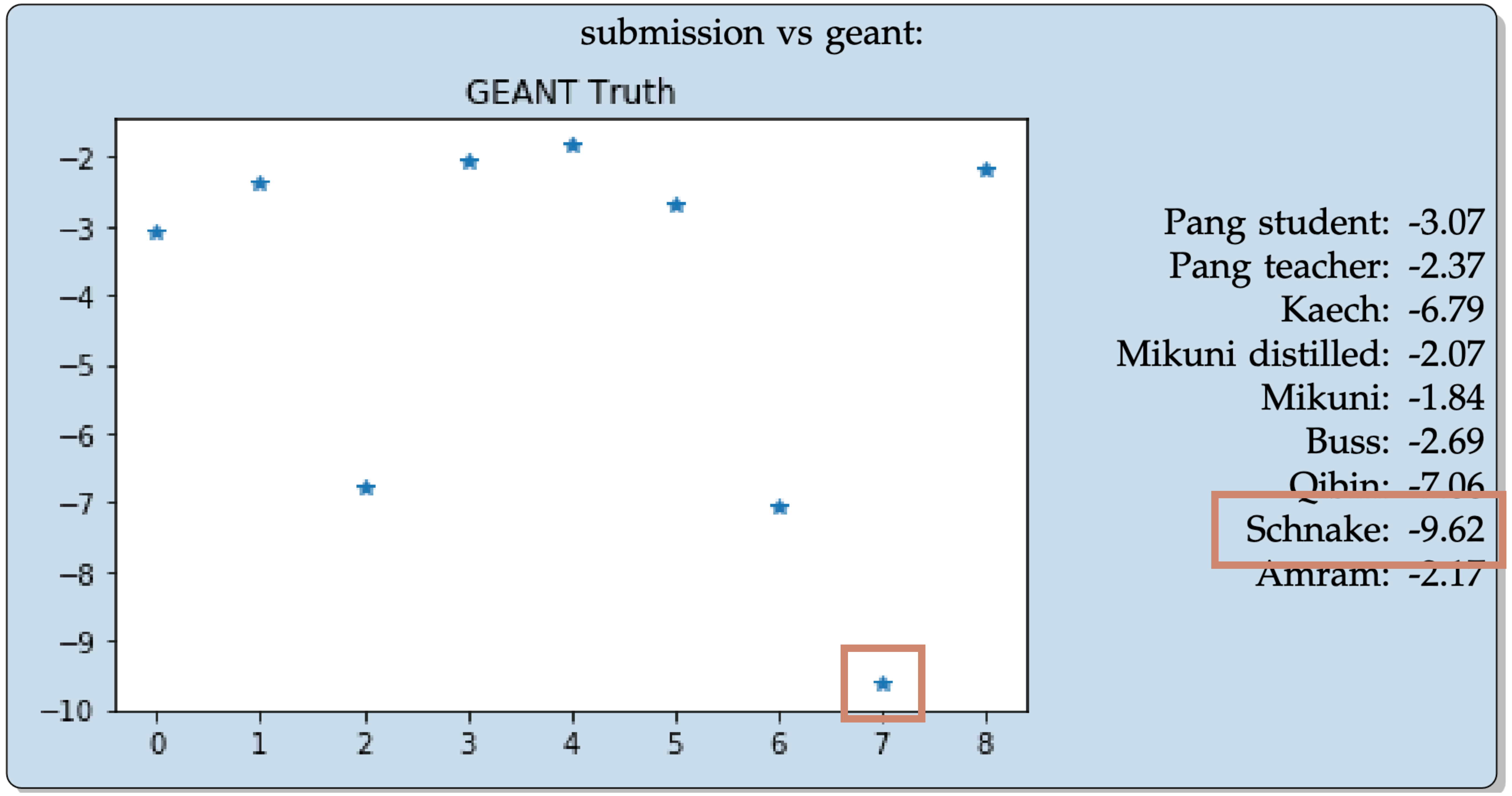
- 200k GEANT4 electron showers
- Energy Distribution: 1 GeV to 1 TeV using log-uniform distribution
- 45 layers with 18 radial and 50 angular bins, totaling 40,500 voxels



https://calochallenge.github.io/homepage/

The performance of CaloPointFlow I

Log-posterior ds2



Claudius Krause - CaloChallenge (preliminary) Results - CaloChallenge Workshop

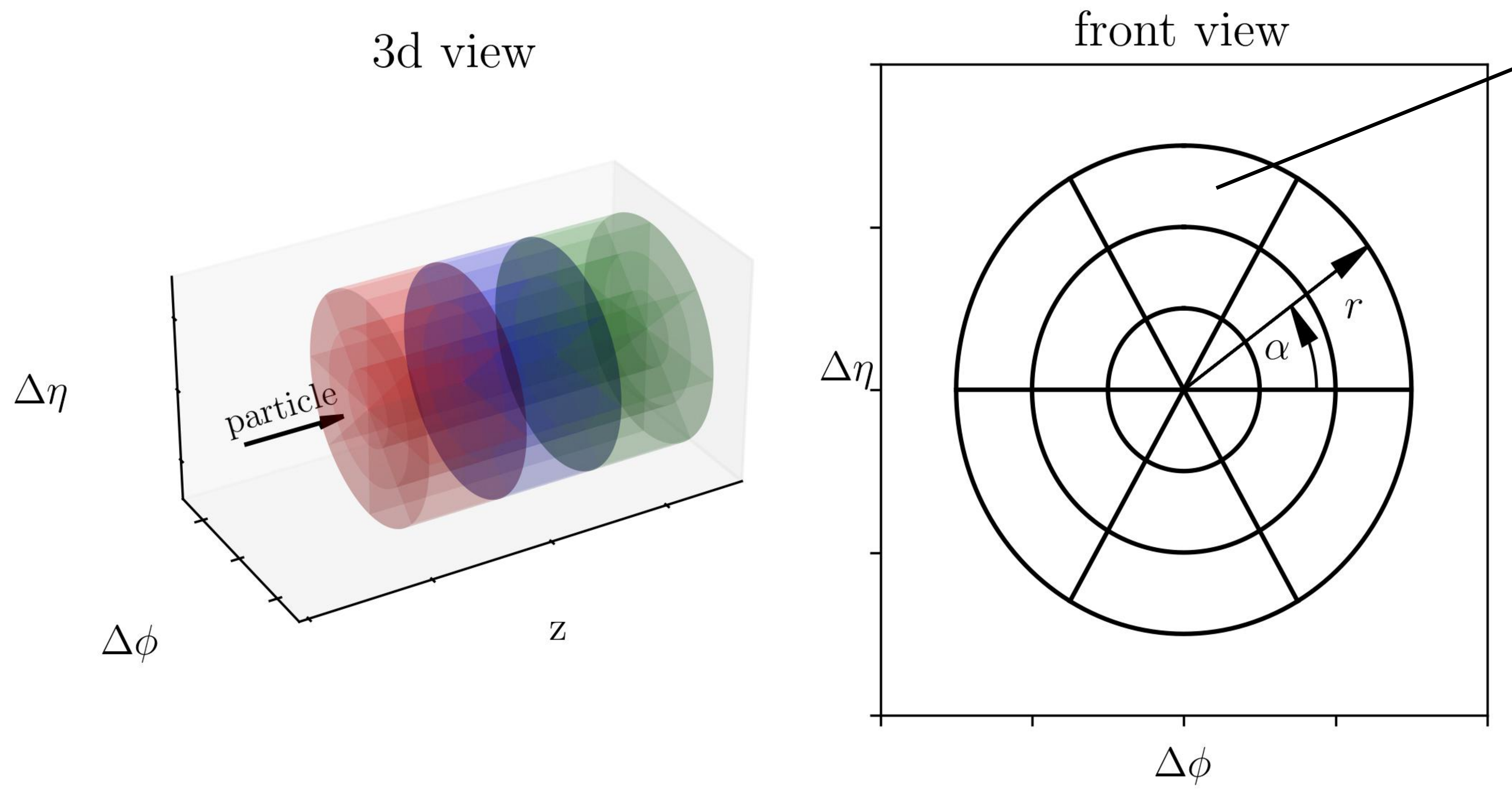
Results for Dataset 3 were not ready at that time.

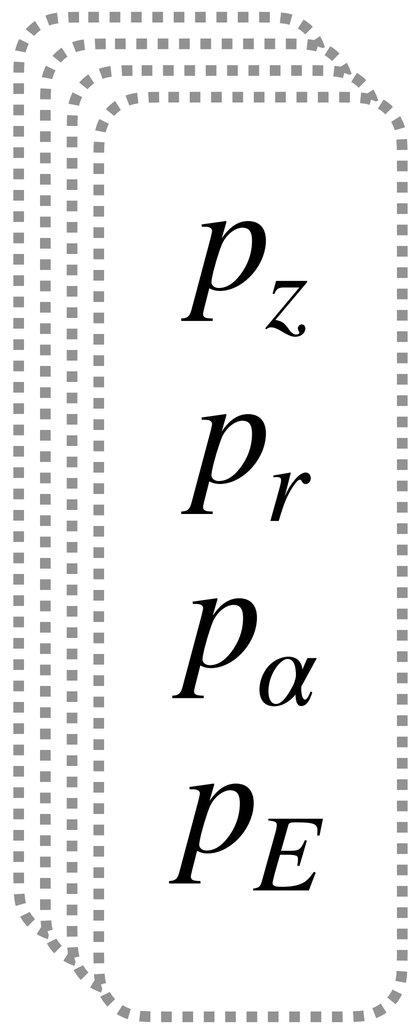
CaloPointFlow I was performing poorly.

CaloPointFlow

Preprocessing

- Showers in high granular calorimeters are sparse
- Their geometry can be irregular
- Use point clouds to handle this





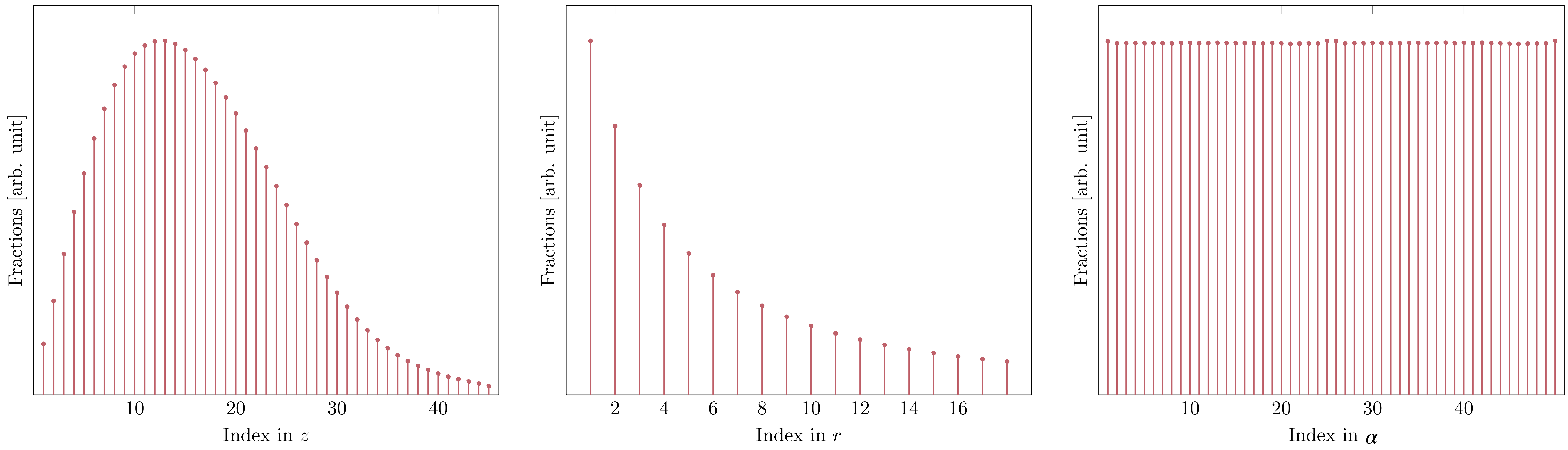
- Get rid of empty cells
- Each hit is represented as point
- One shower equals to one point cloud

https://calochallenge.github.io/homepage/

Dequantization

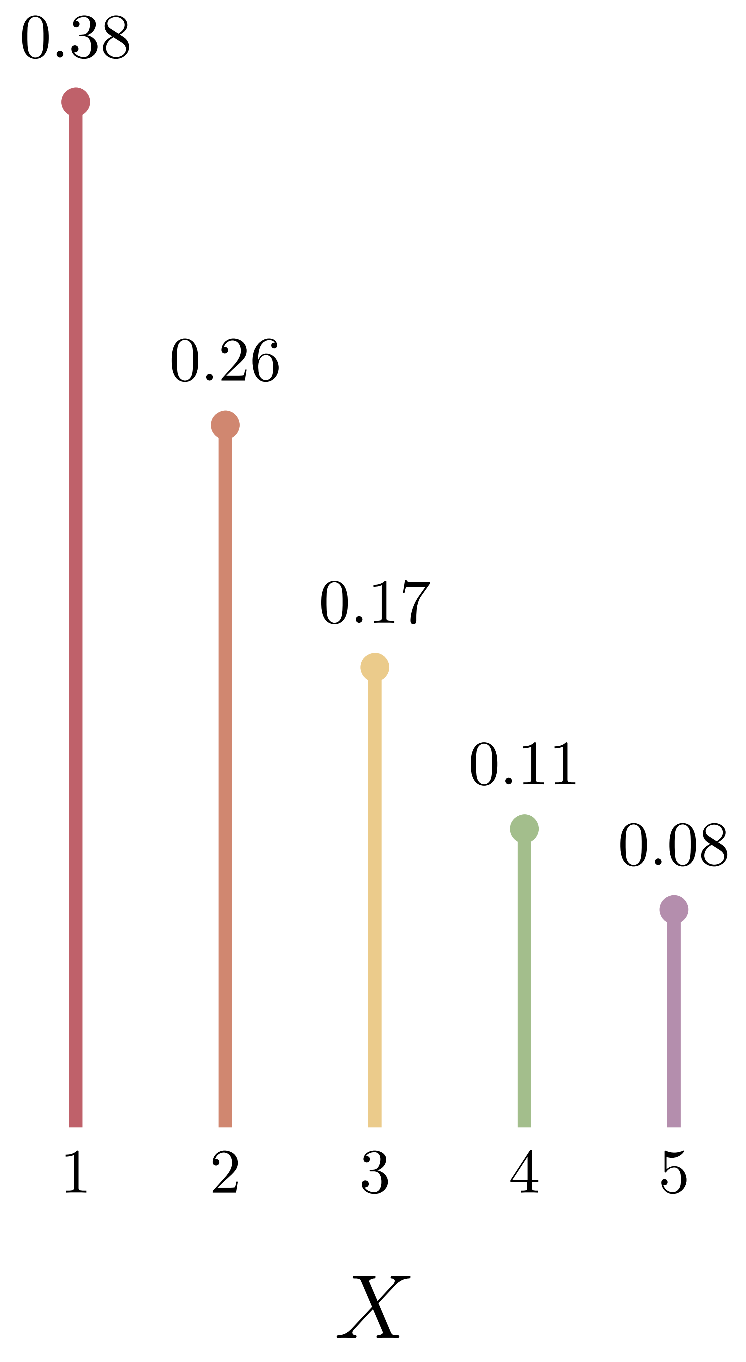
Marginal distributions

- The probability of different index has a large variation



Dequantization

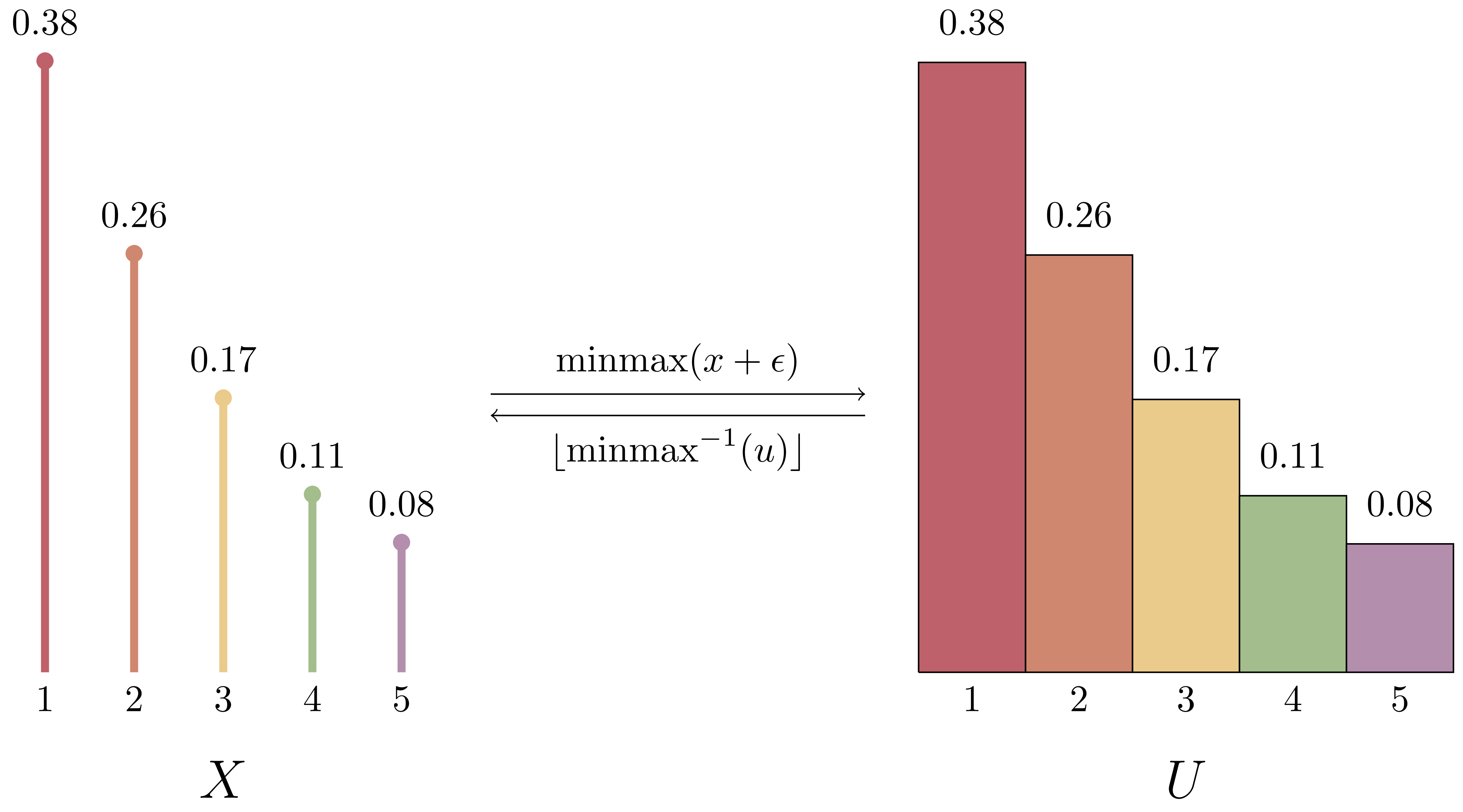
Standard dequantization



- A normalizing flow maps a input distribution to a normal distribution
- Discrete distributions are not directly mappable to a normal distribution
- We have to lift it to a continuous distribution
- This is called *dequantization*

Dequantization

Previous dequantization



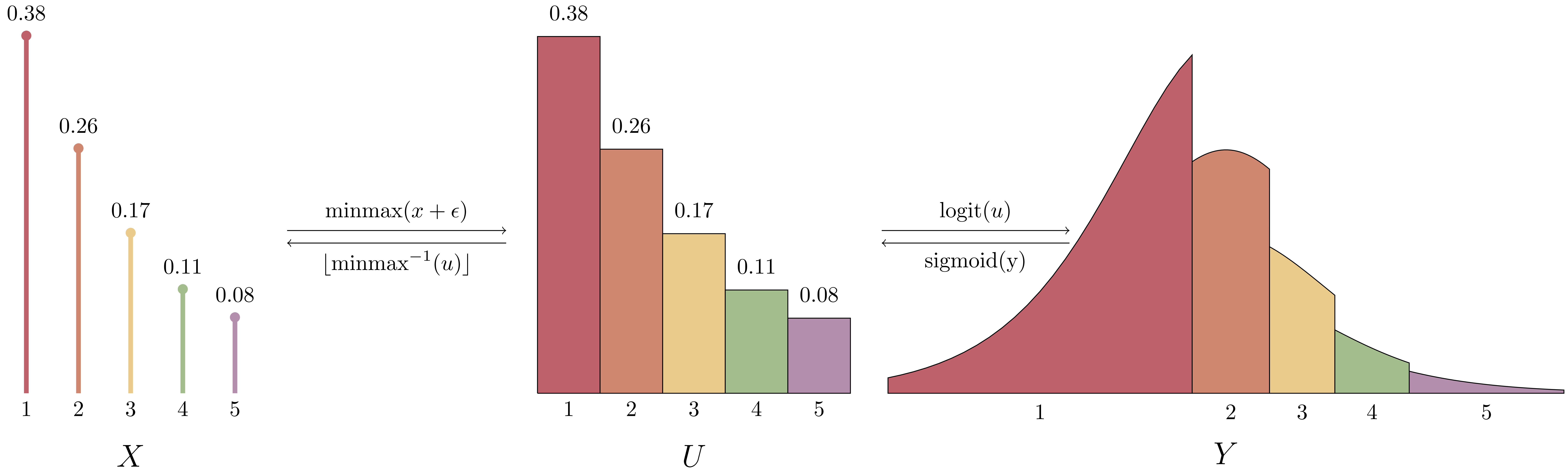
Uria et al [arxiv:1306.0186]

- Uniform noise is added
- The whole distribution is scaled to $[0,1]$

Dequantization

Previous dequantization

- As an invertible mapping $[0,1] \rightarrow [-\infty, \infty]$ the logit function is choosen
- The inverse is $\mathrm{sigmoid}$
- The resulting distribution has edges and long tails

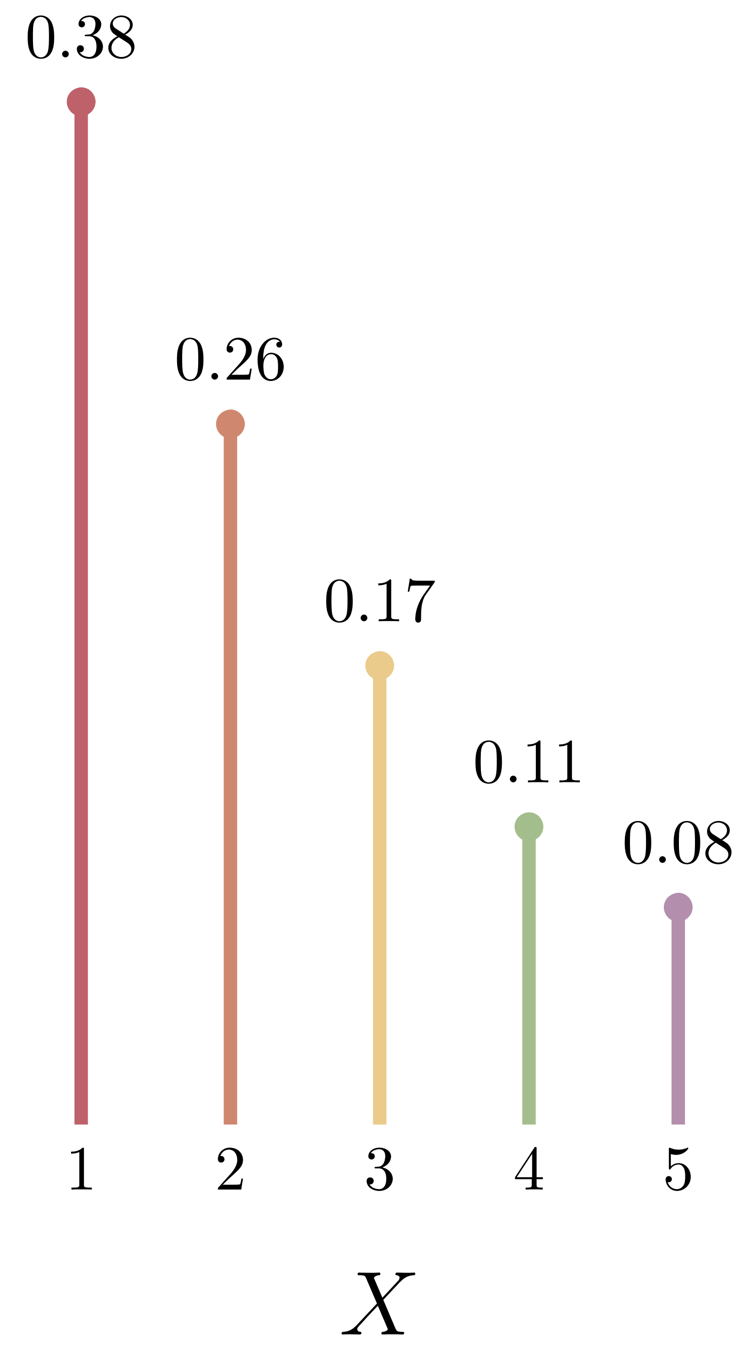


Uria et al - RNADE [`arxiv:1306.0186`]

Dinh et al - RealNVP [`arxiv:1605.08803`]

Dequantization

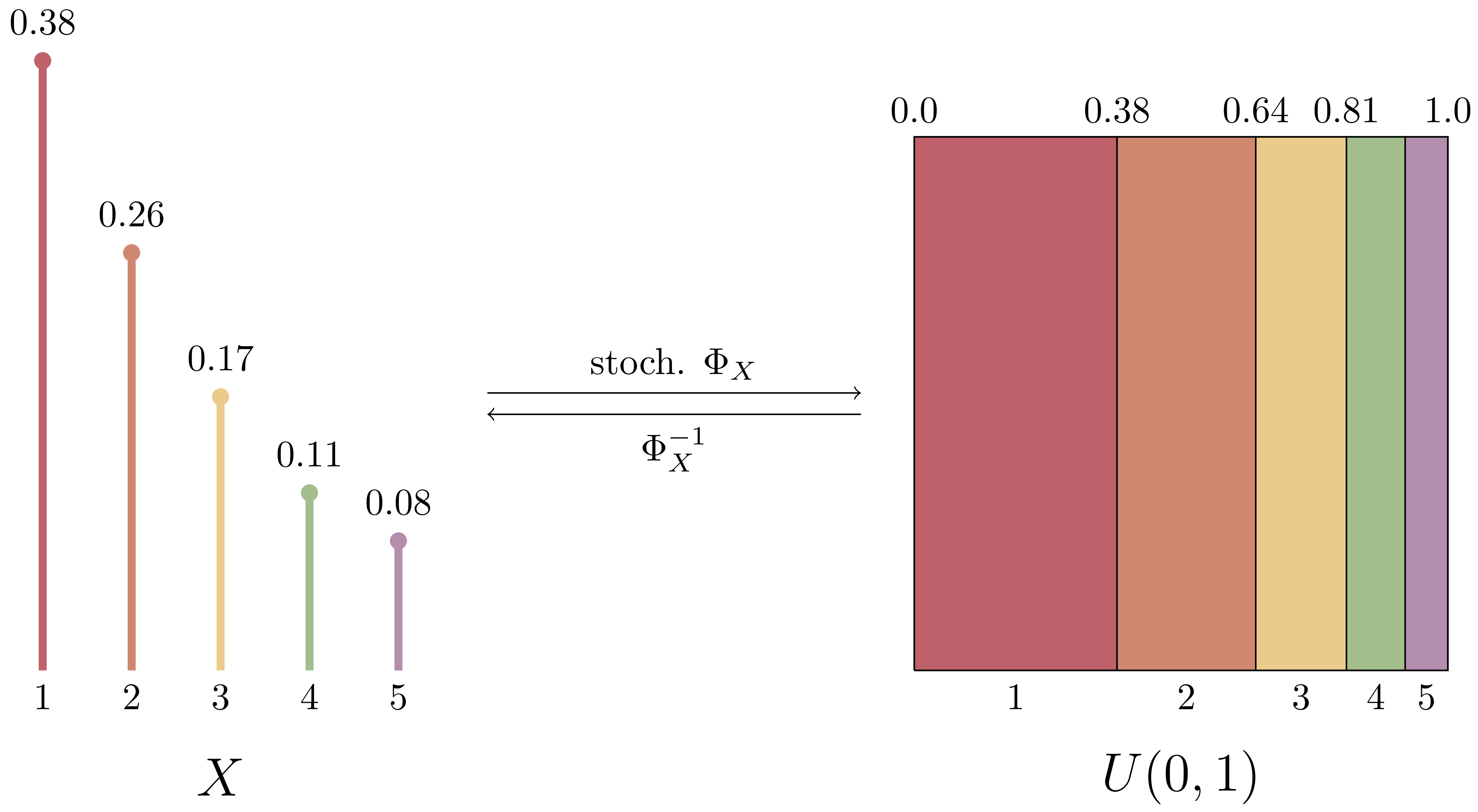
CDFDequantization



- Can we do better?

Dequantization

CDFDequantization



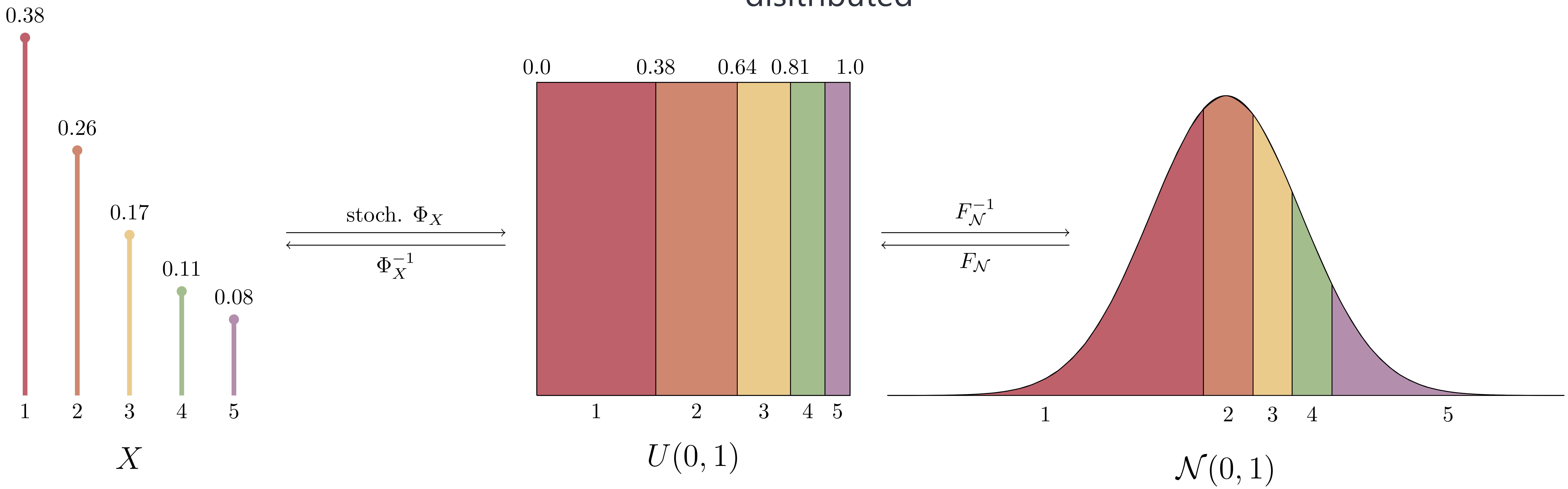
- We transform each index i to a random value between $\mathrm{cdf}(x)$ and $\mathrm{cdf}(x+1)$
- $\Phi_X(x) = \mathrm{cdf}(x) + \mathrm{pmf}(x) \cdot \epsilon$
- $\epsilon \sim U(0,1)$
- Inverse $\Phi^{-1}(u) = \min\{x \quad u \leq \mathrm{cdf}(x+1)\}$

Dequantization

CDFDequantization

Improvement No 1

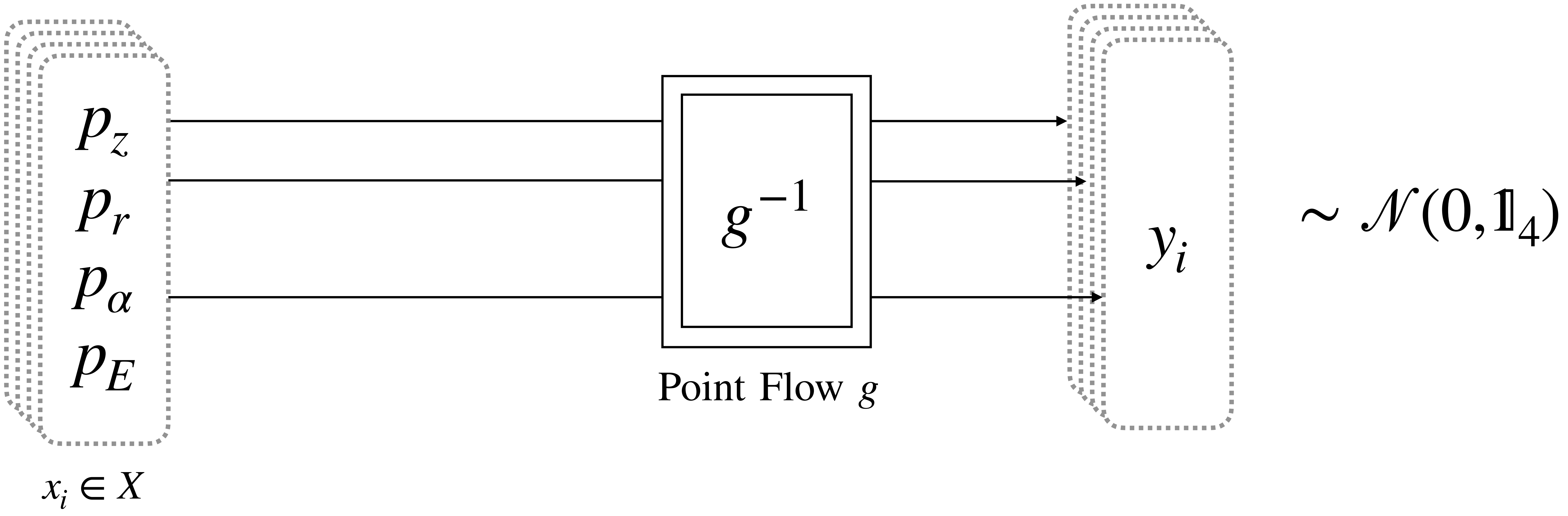
- Instead of logit we use the quantile function of $\mathcal{N}(0,1)$
- Our transformed distribution is normal disitributed



CaloPointFlow

Learn each point separately

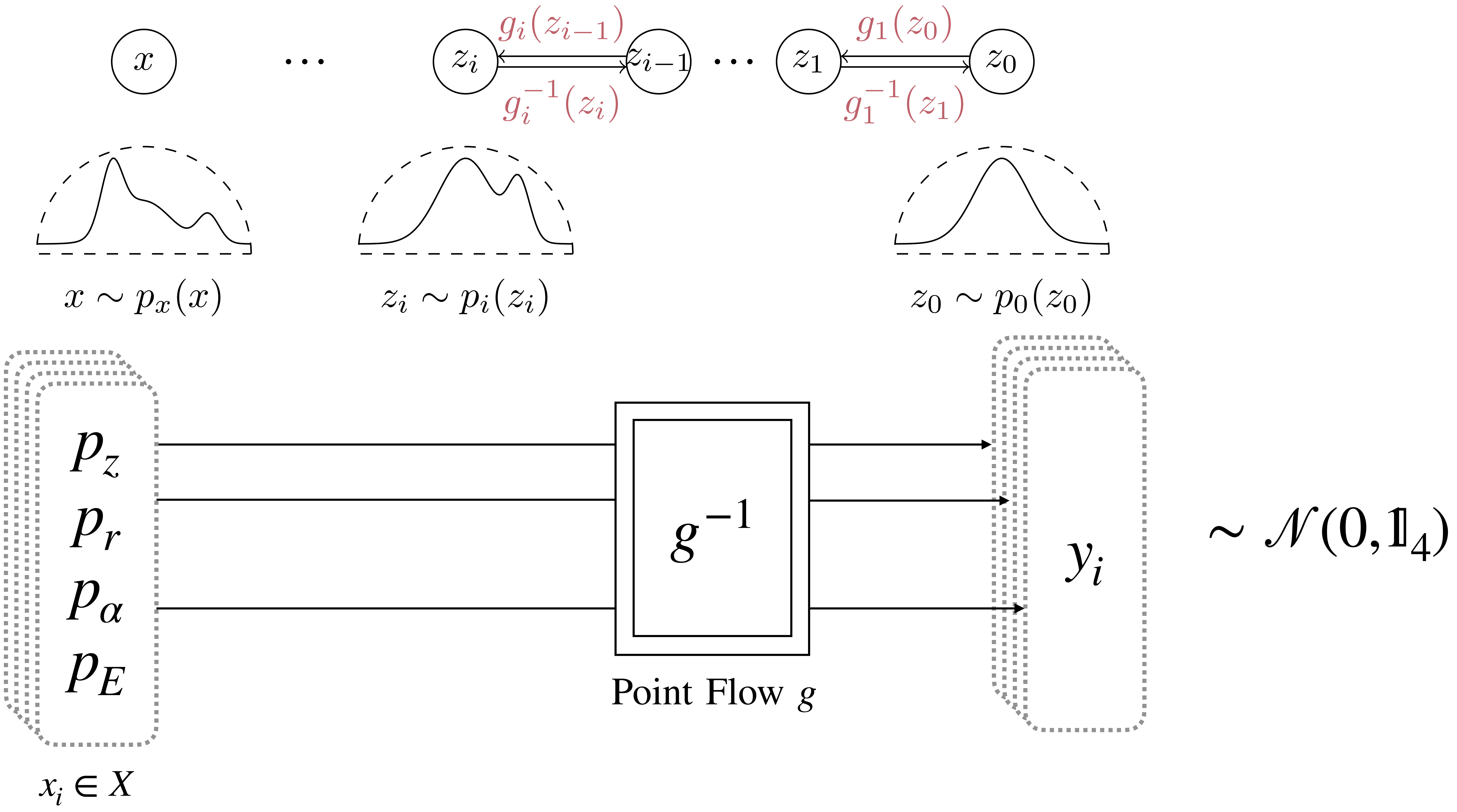
- Point Flow is a point-wise transformation



CaloPointFlow

Point Flow g architecture

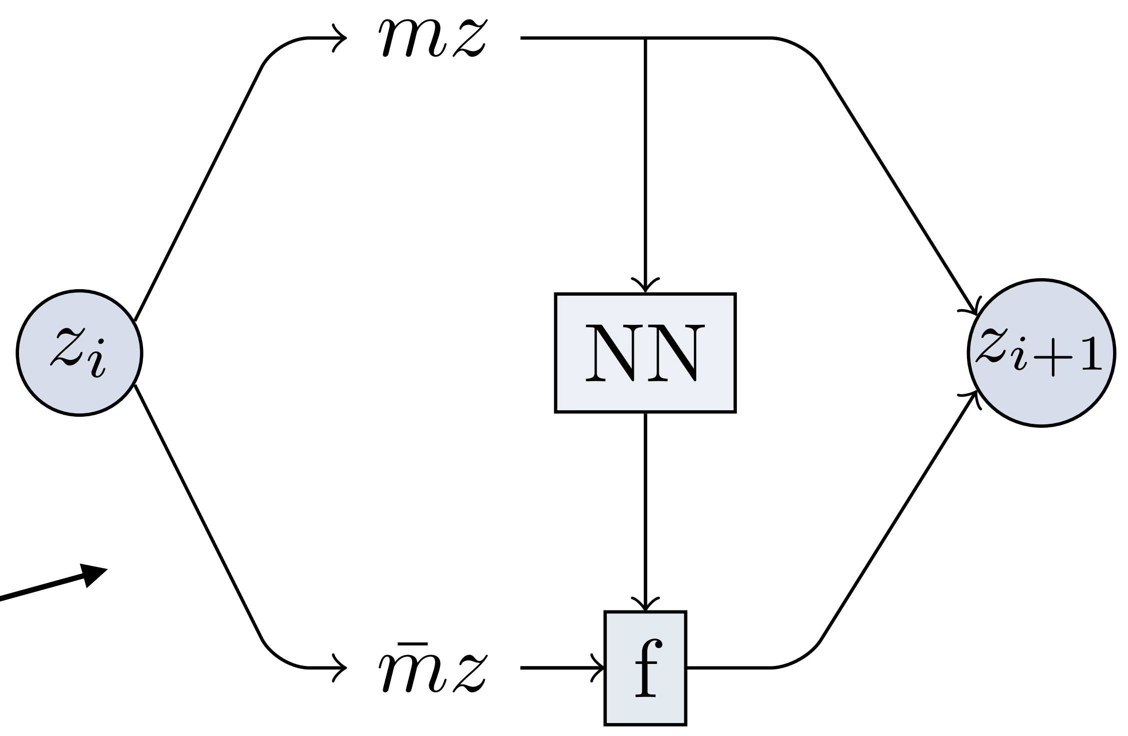
The Point Flow g is a Normalizing Flow

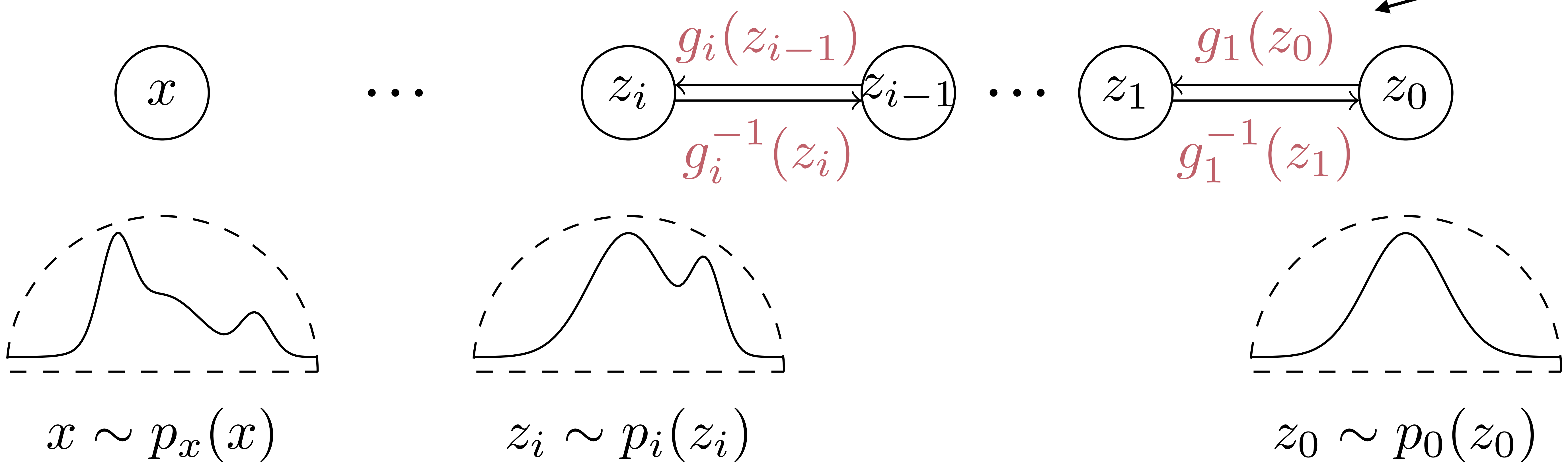


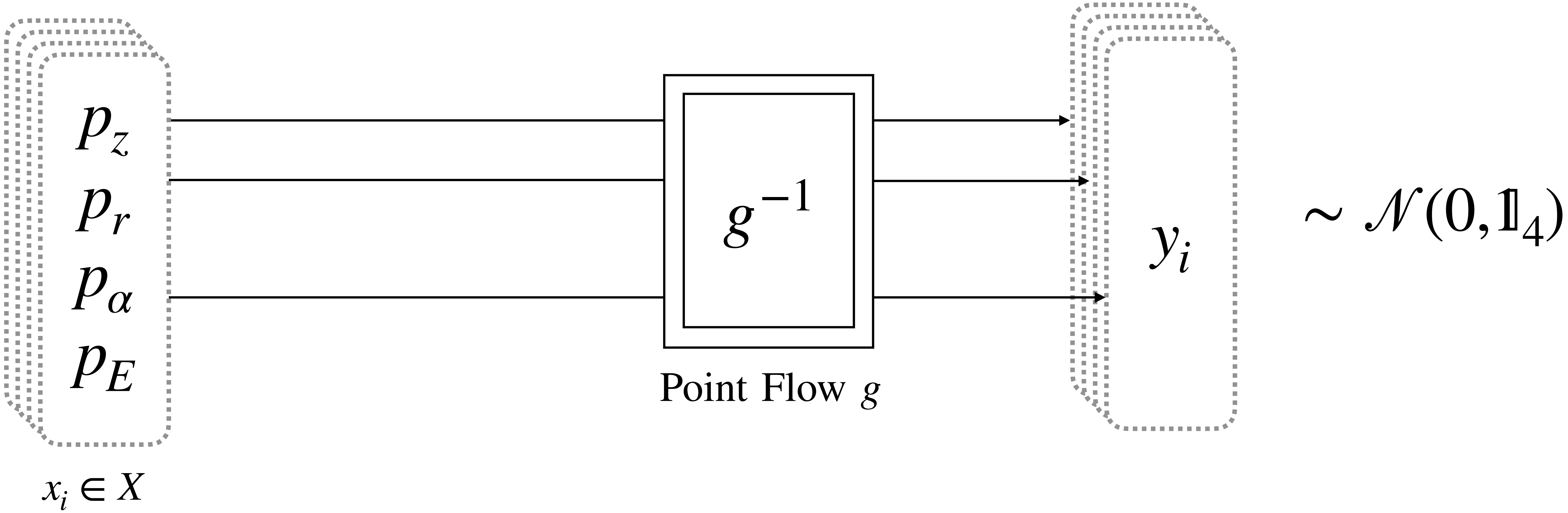
CaloPointFlow

Point Flow g architecture

The g_is are coupling layers



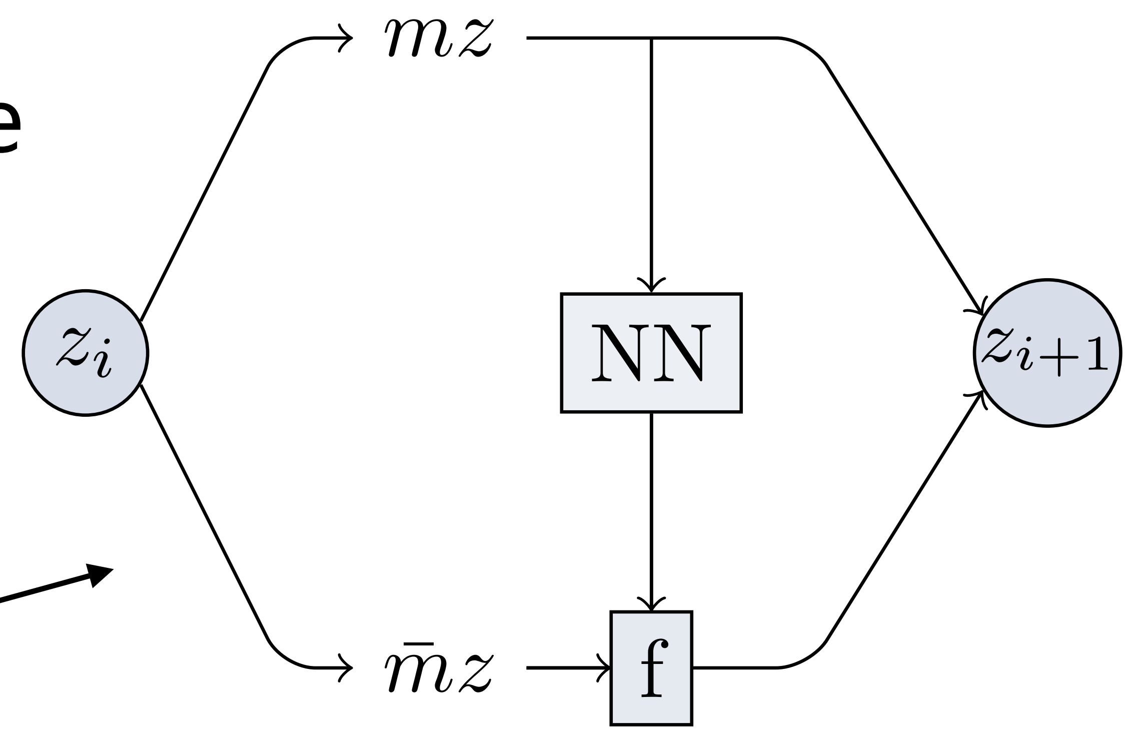


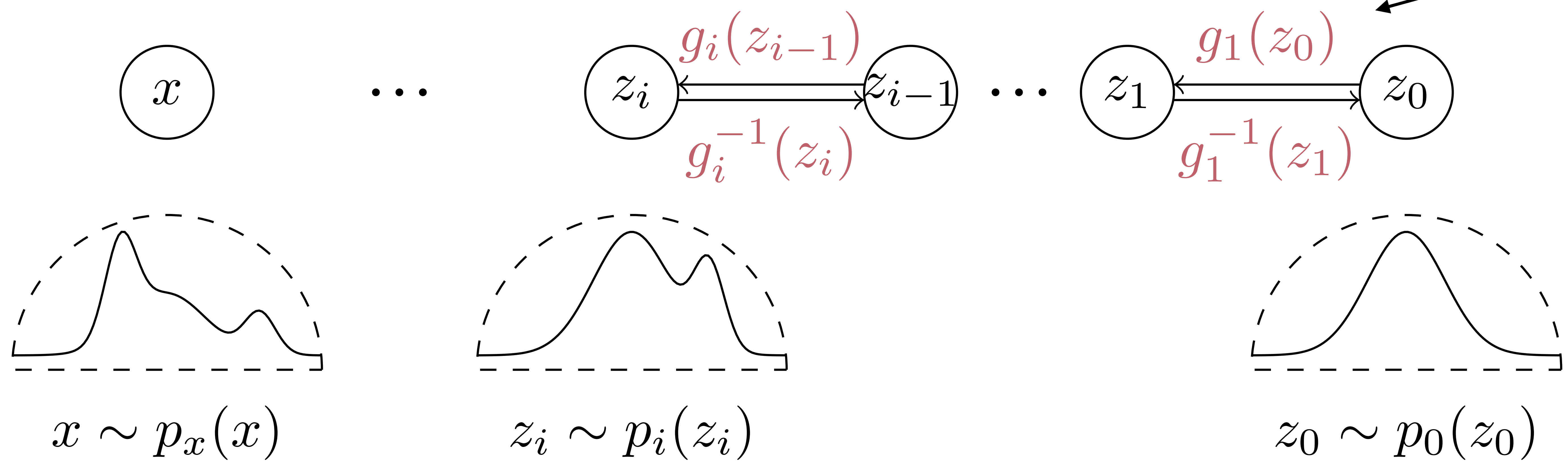


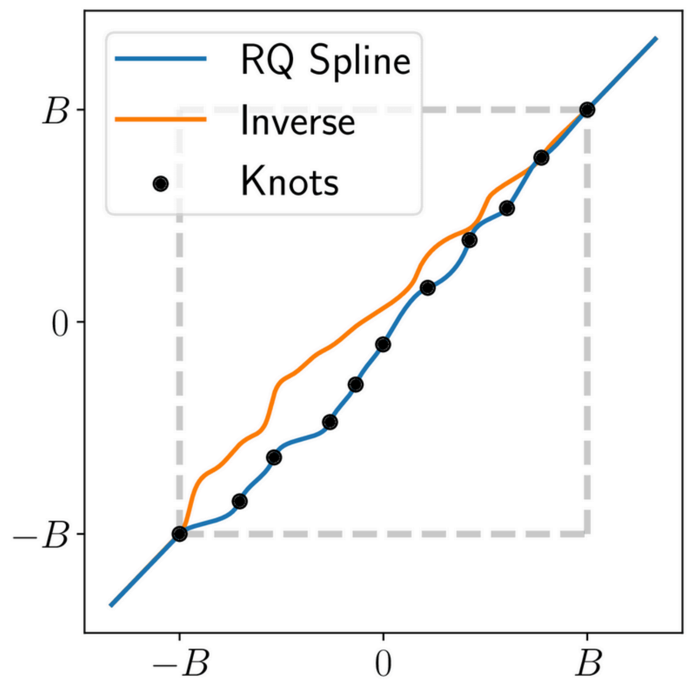
CaloPointFlow

Point Flow g architecture

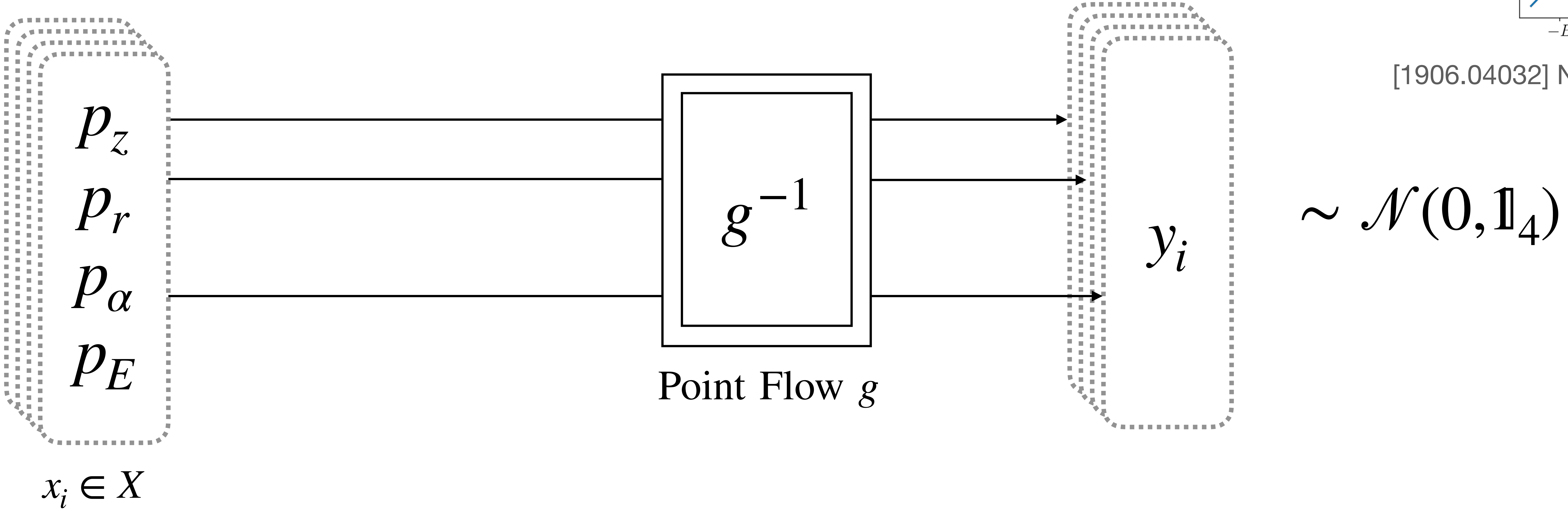
Rational quadratic splines (RQS) are used as the univariate transformation





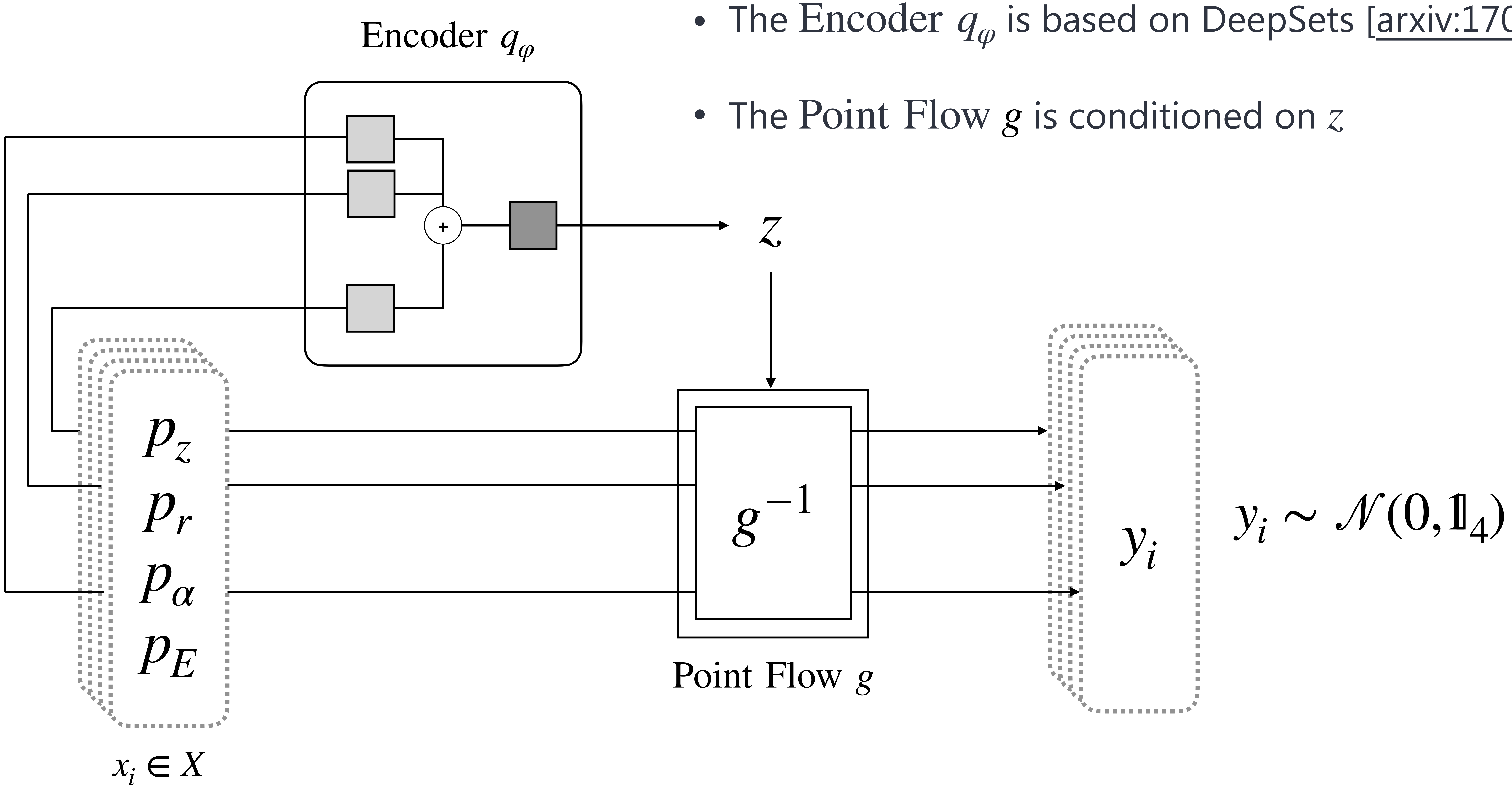


[1906.04032] Neural Spline Flows



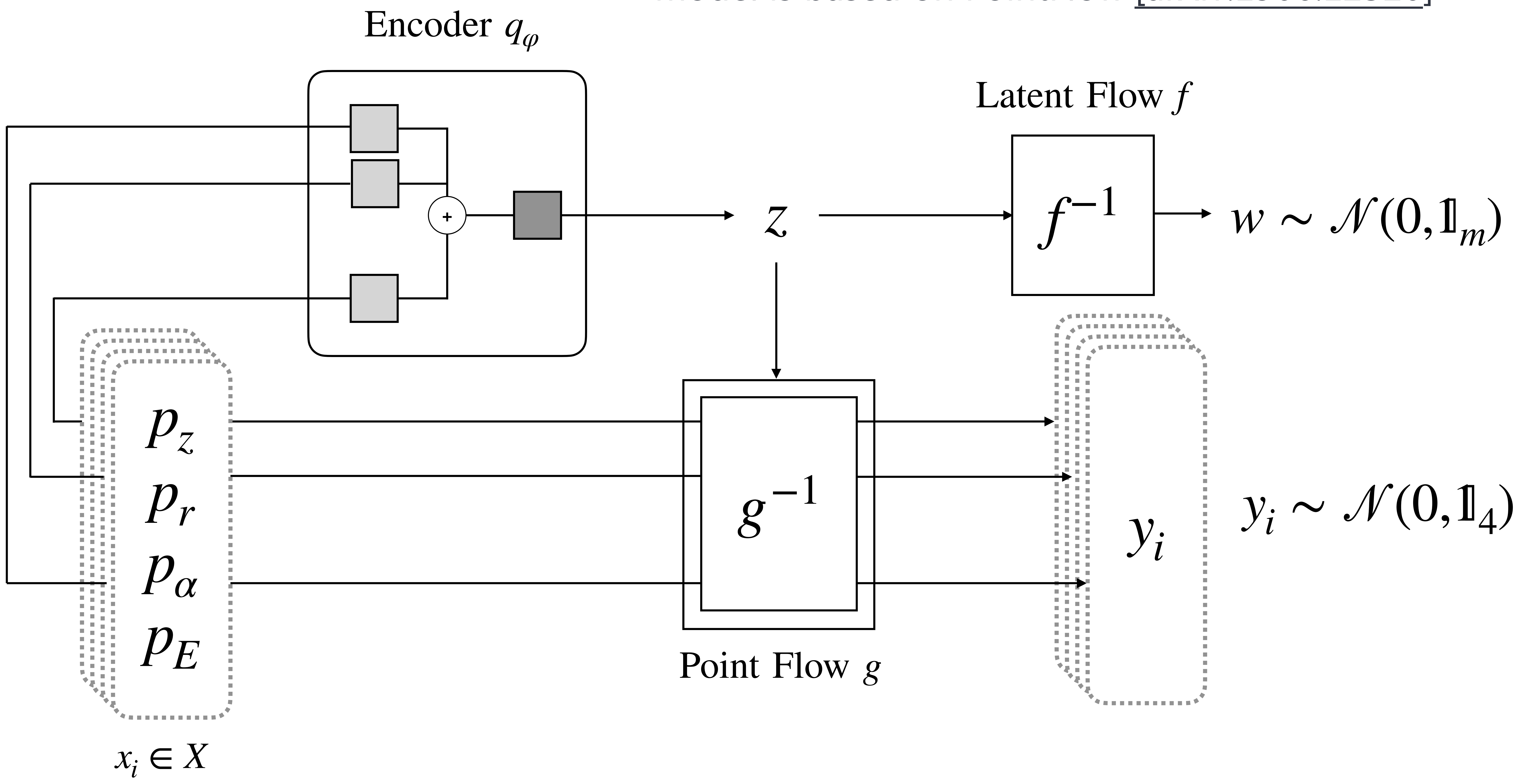
CaloPointFlow

- The shower information is encoded in the latent variable z
- The Encoder q_φ is based on DeepSets [arxiv:1703.06114]
- The Point Flow g is conditioned on z



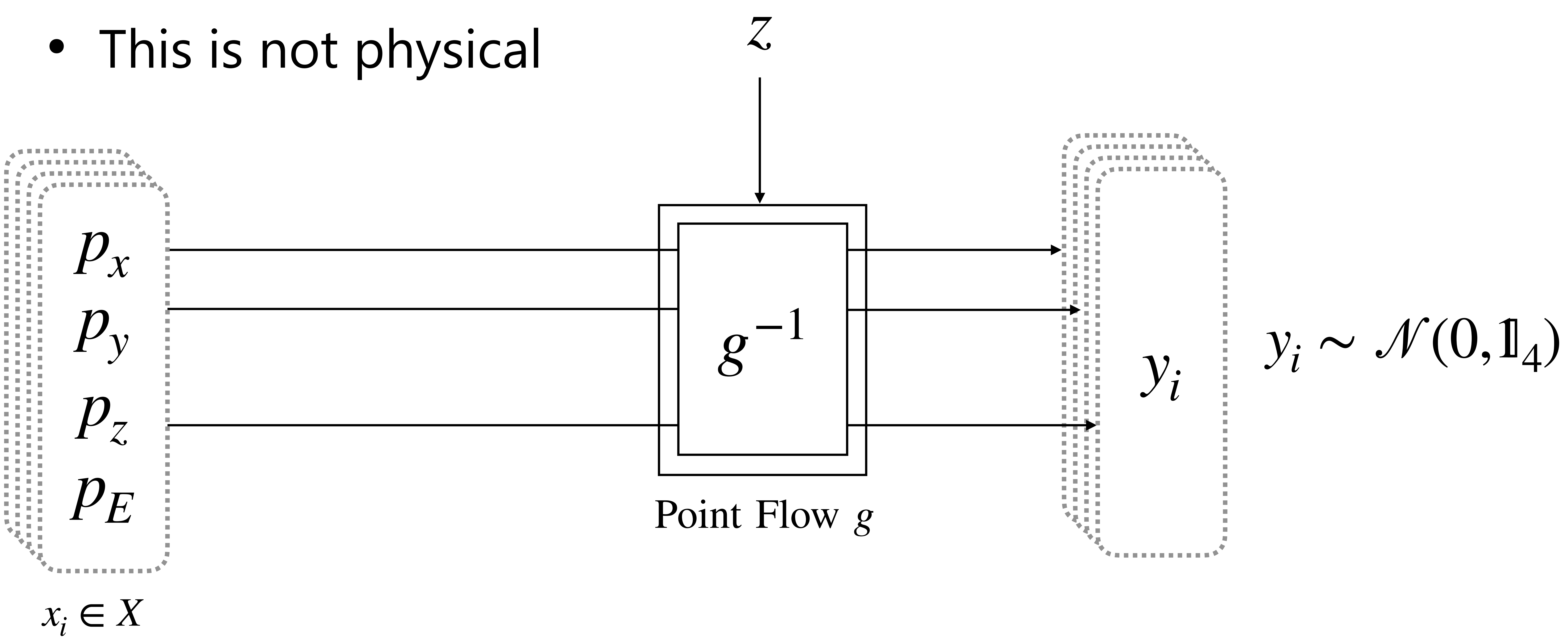
CaloPointFlow

- z is generated by the Latent Flow f
- f has the same structure as g
- model is based on PointFlow [arXiv:1906.12320]

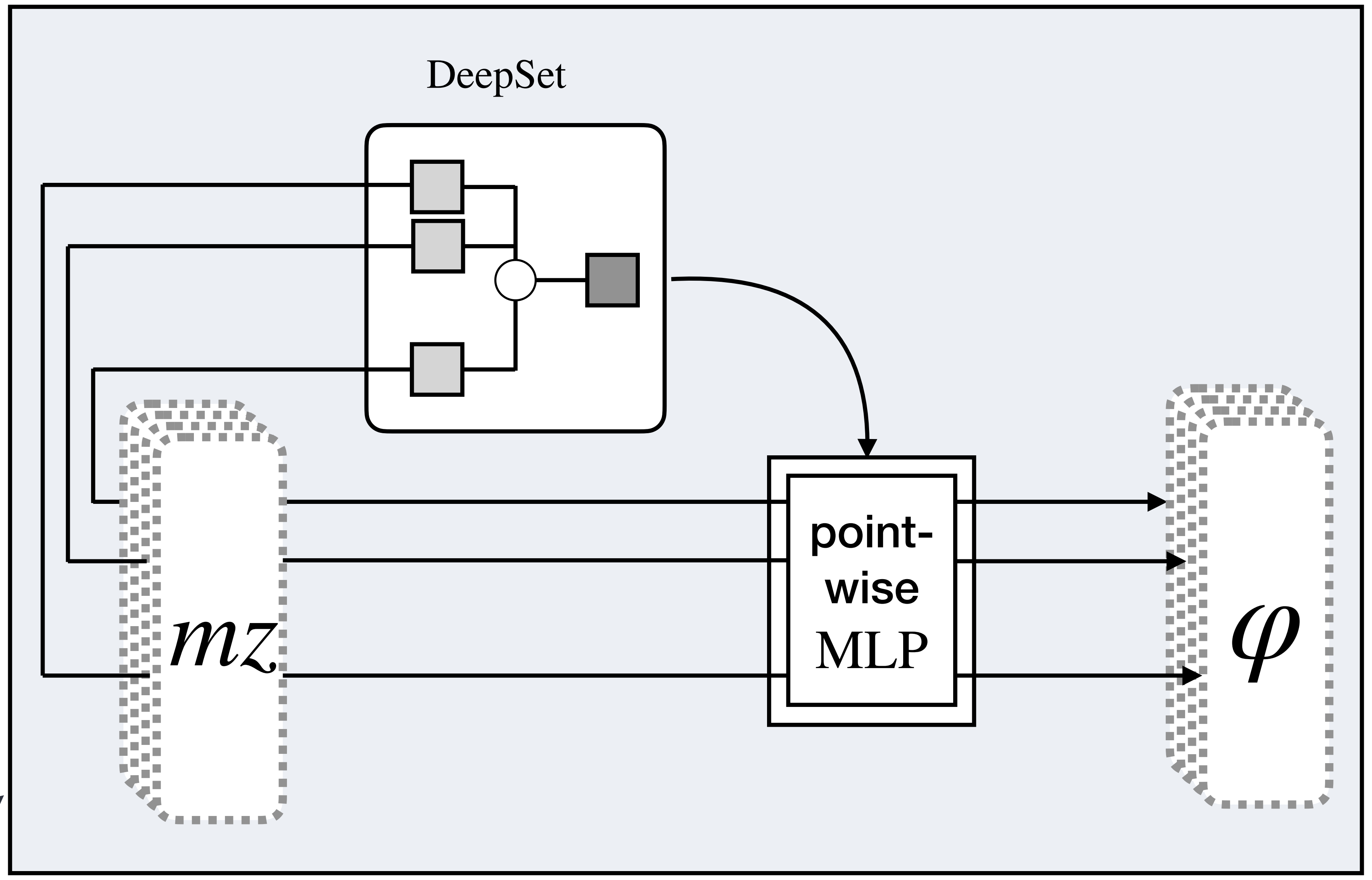


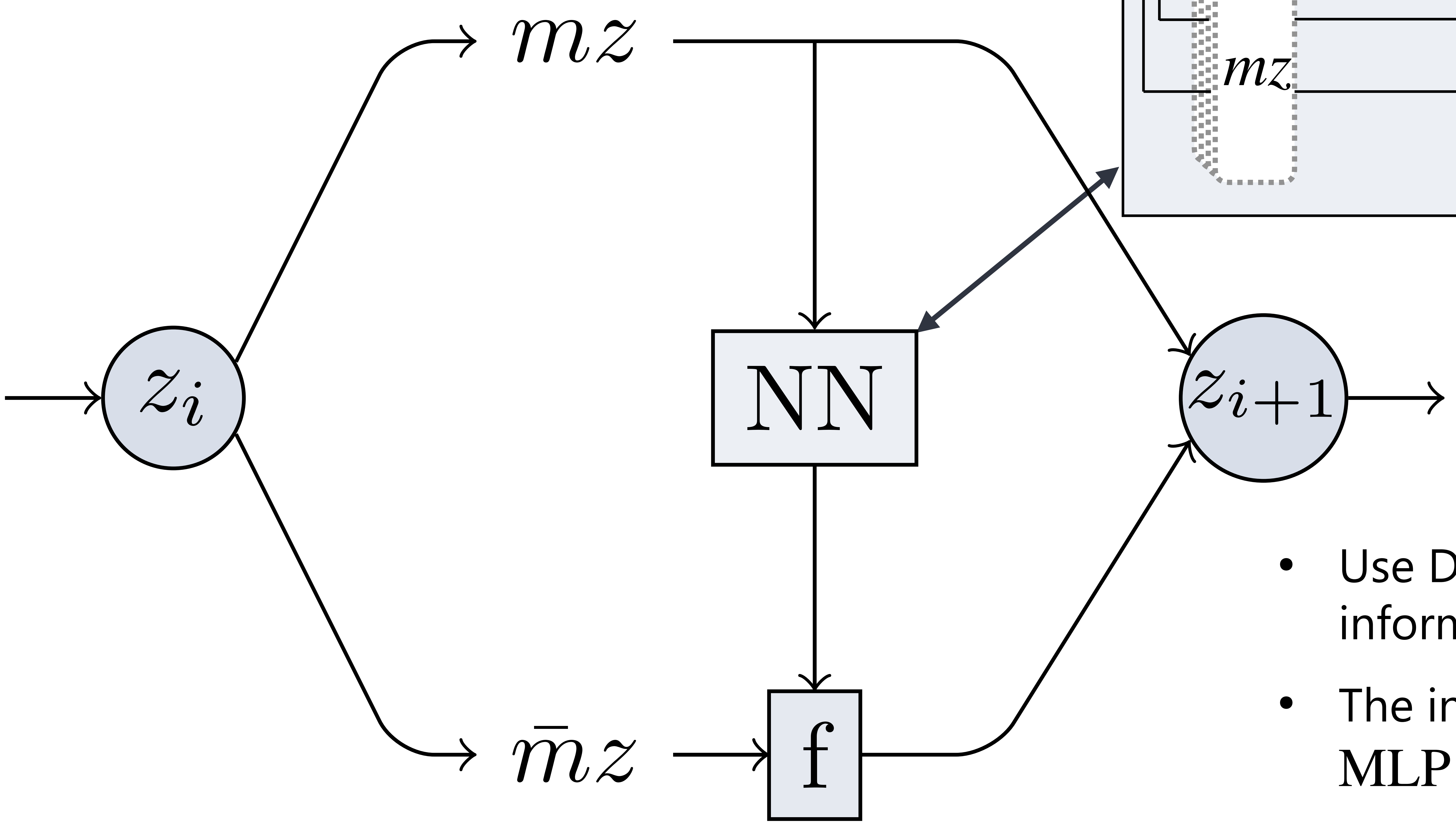
CaloPointFlow

- By construction there is no point-to-point communication, while transformation
- Therefore, the points are conditional independent ($x_i \perp\!\!\!\perp x_j \;\; z$)
- This is not physical



DeepSetFlow

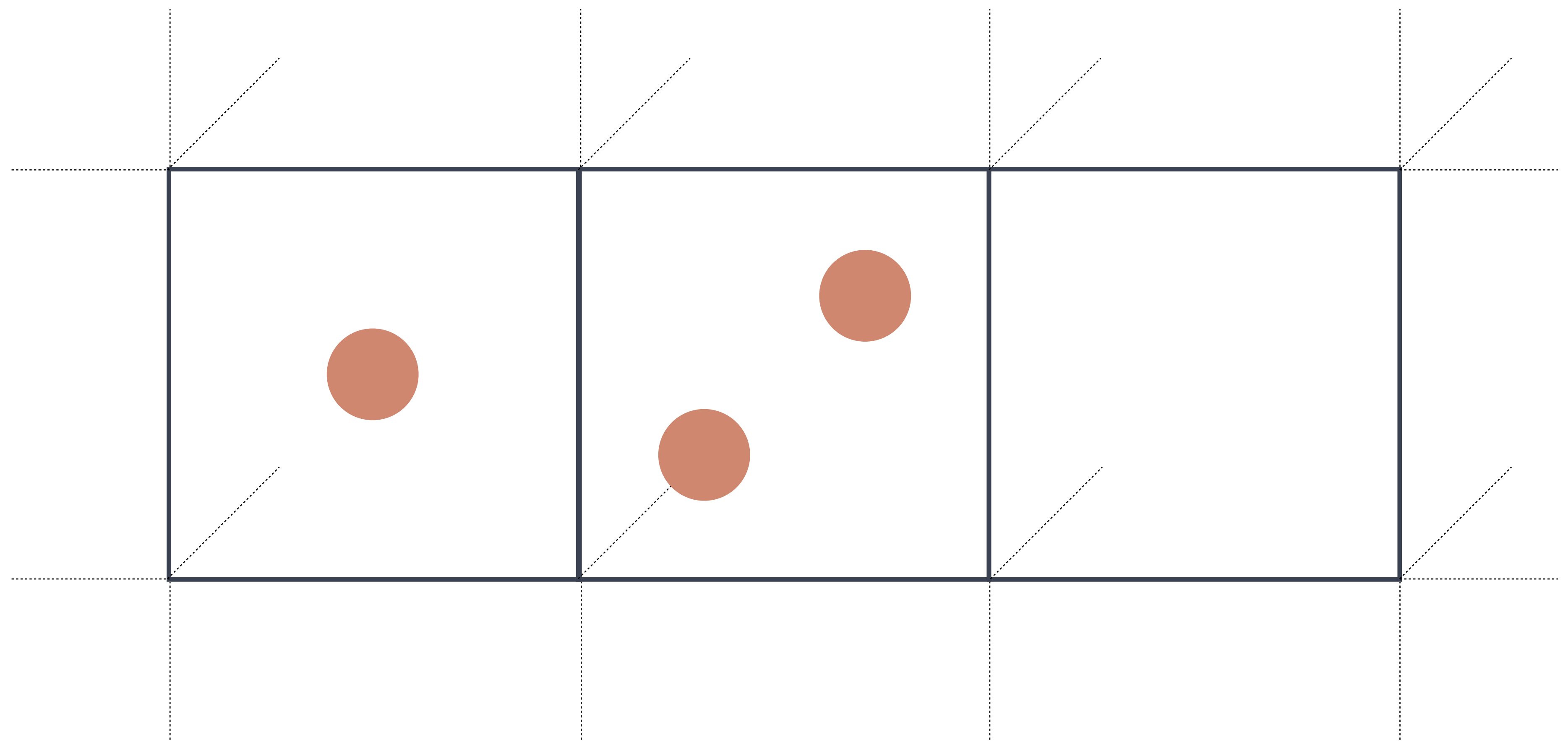




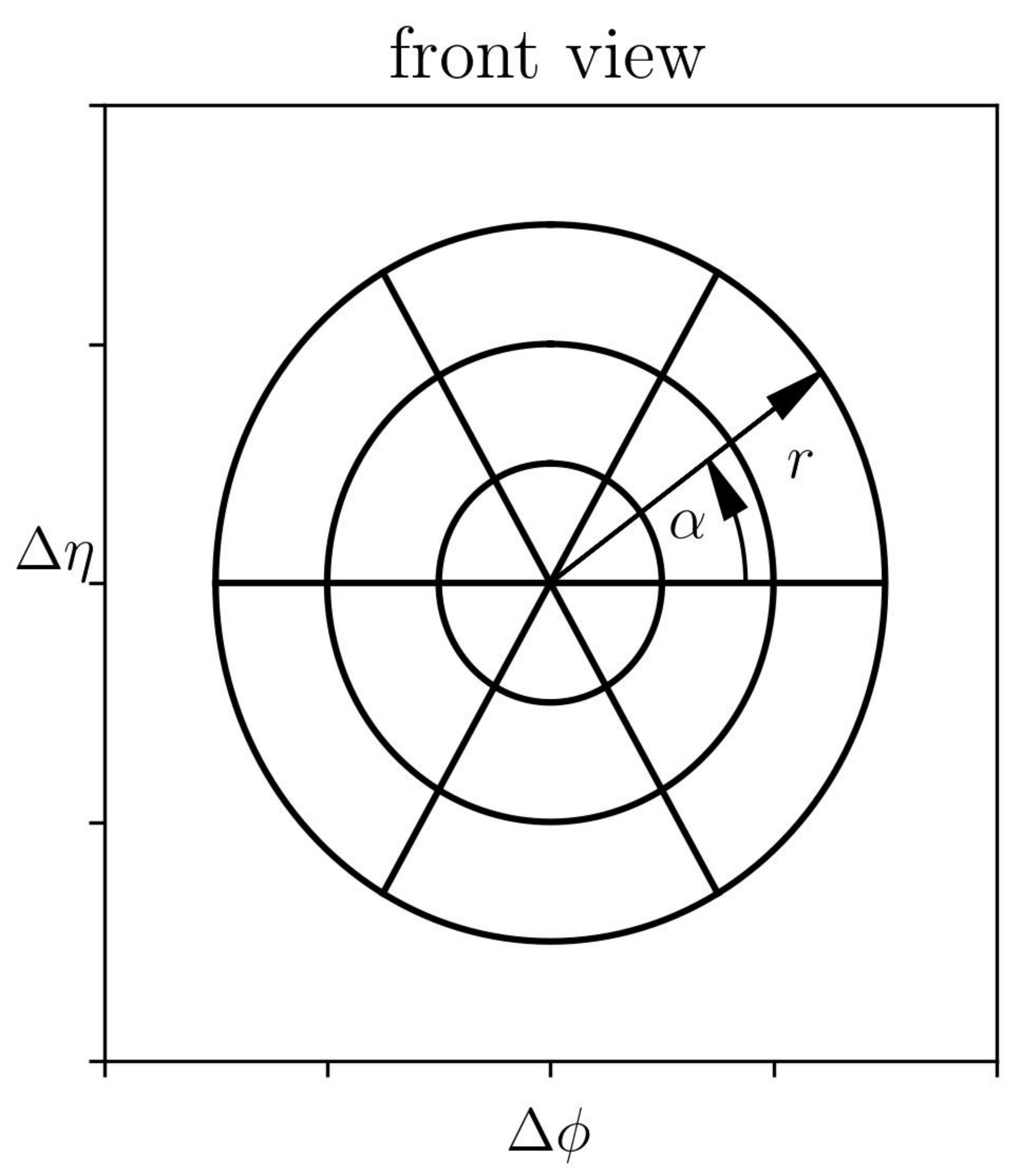
- Use DeepSet in each NN to aggregate the information of the point cloud
- The information is added to the point-wise MLP input
- inspired by EPiC-GAN [arXiv:2301.08128]

Multiple Hits Problem

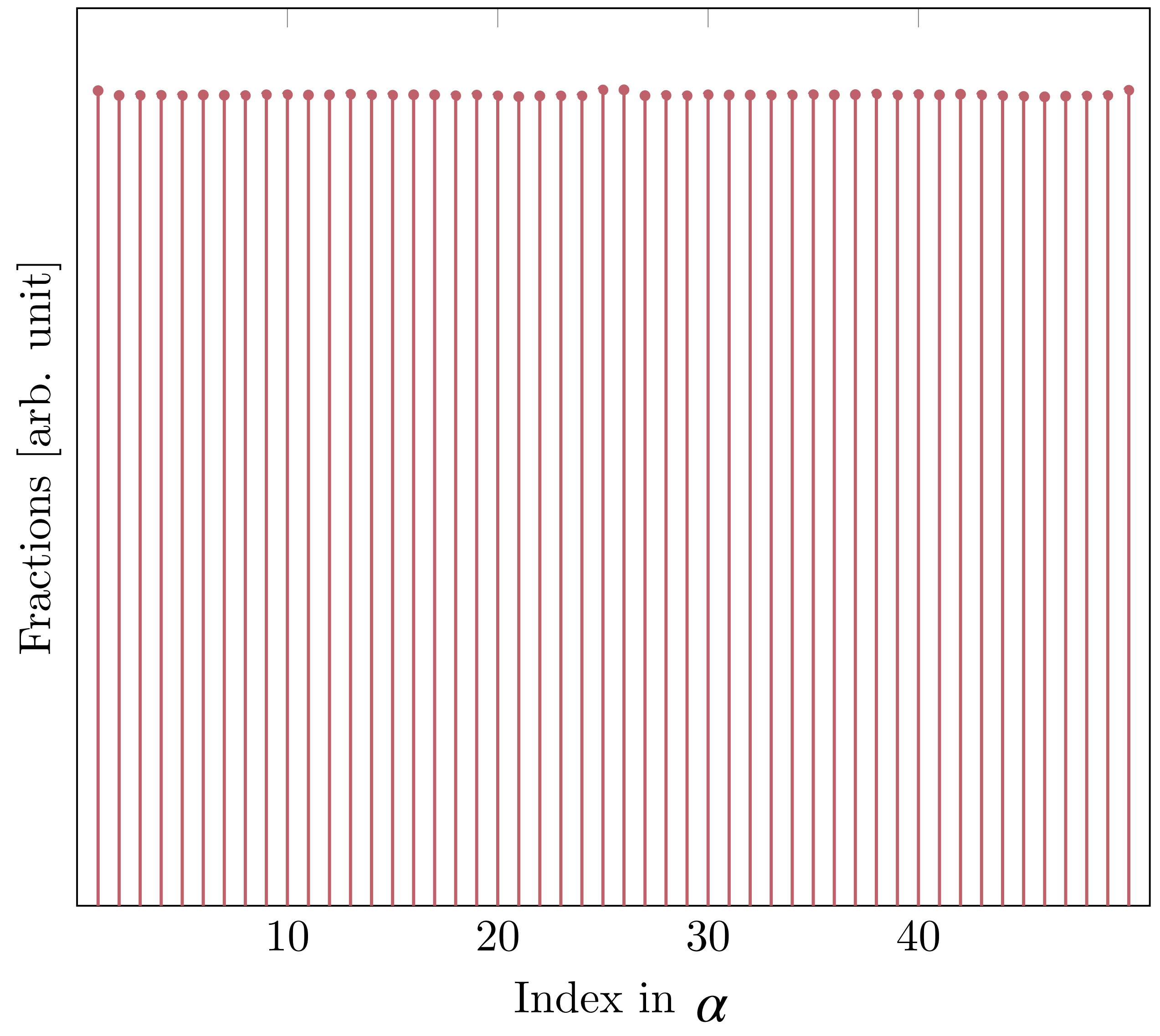
- The points are generated on continuous space
- Mapped to discrete indices
- Multiple hits can have the same index
- Multiple hits cannot occur in the real data



Rotation invariance



- The data is rotation invariant
- Therefore, the marginal distribution in α is flat
- One could generate the α dimension, by random sampling
- This does not preserve shower substructure in α

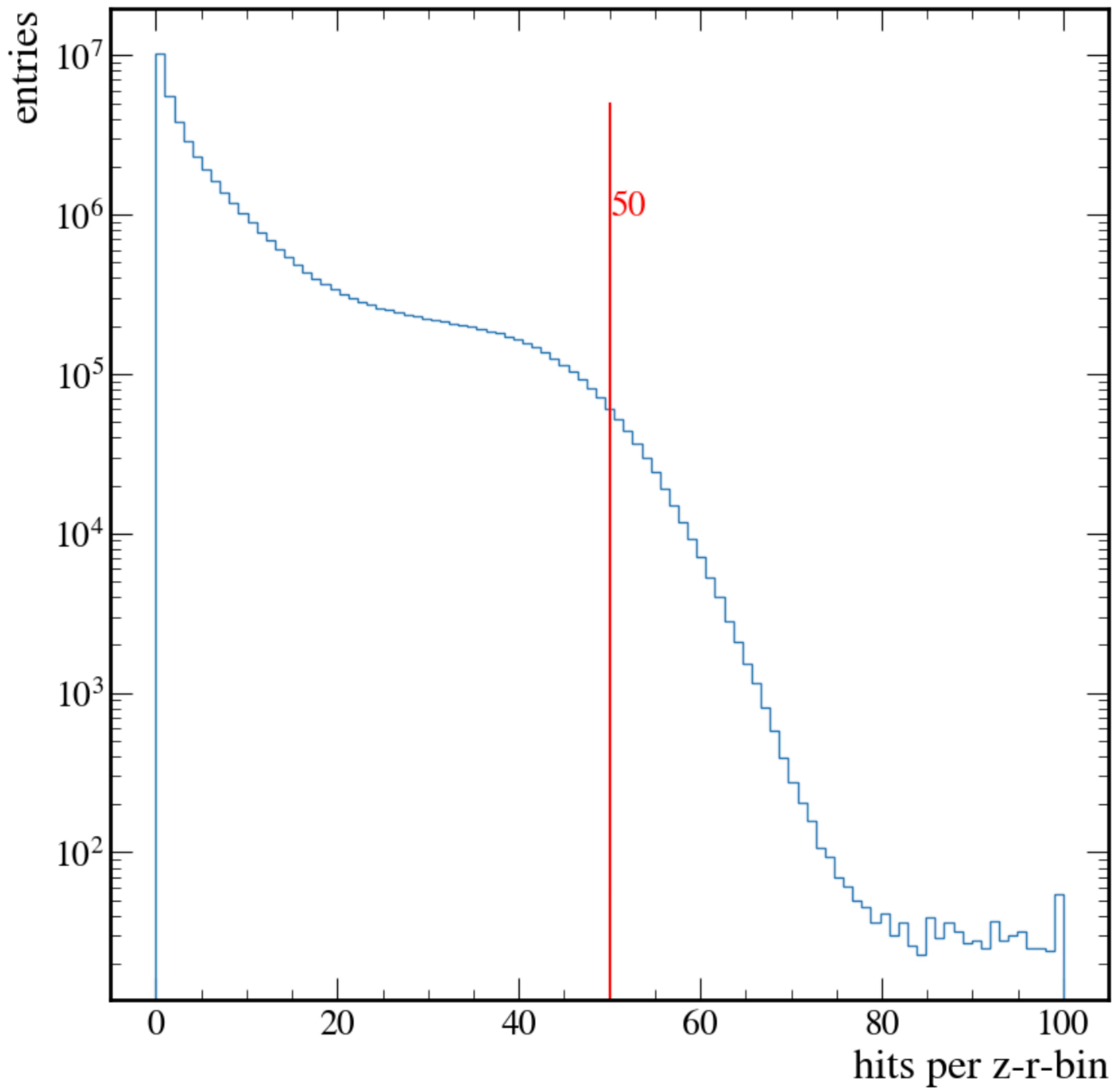


Generate without α

p_z
p_r
p_E

Improvement No 3

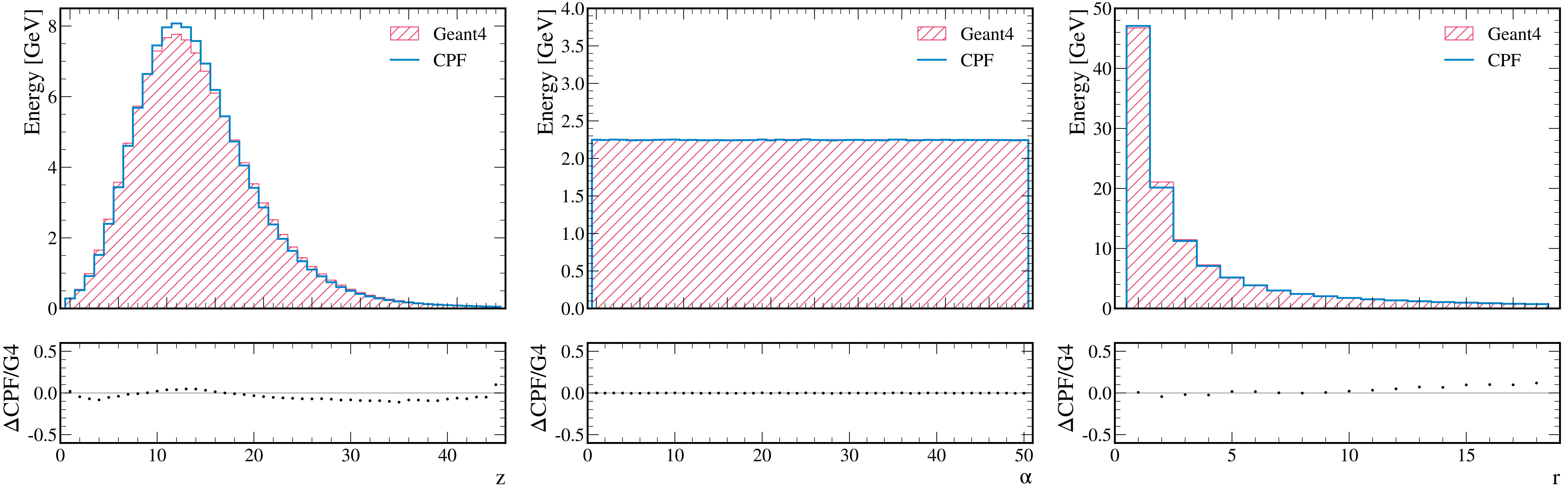
- Generate points without p_α
- This allows 50 hits per (z, r)-bin
- Distribute randomly uniformly in α
- If there are more than 50 hits, we also add them to random α bins



Results

Shower Shapes

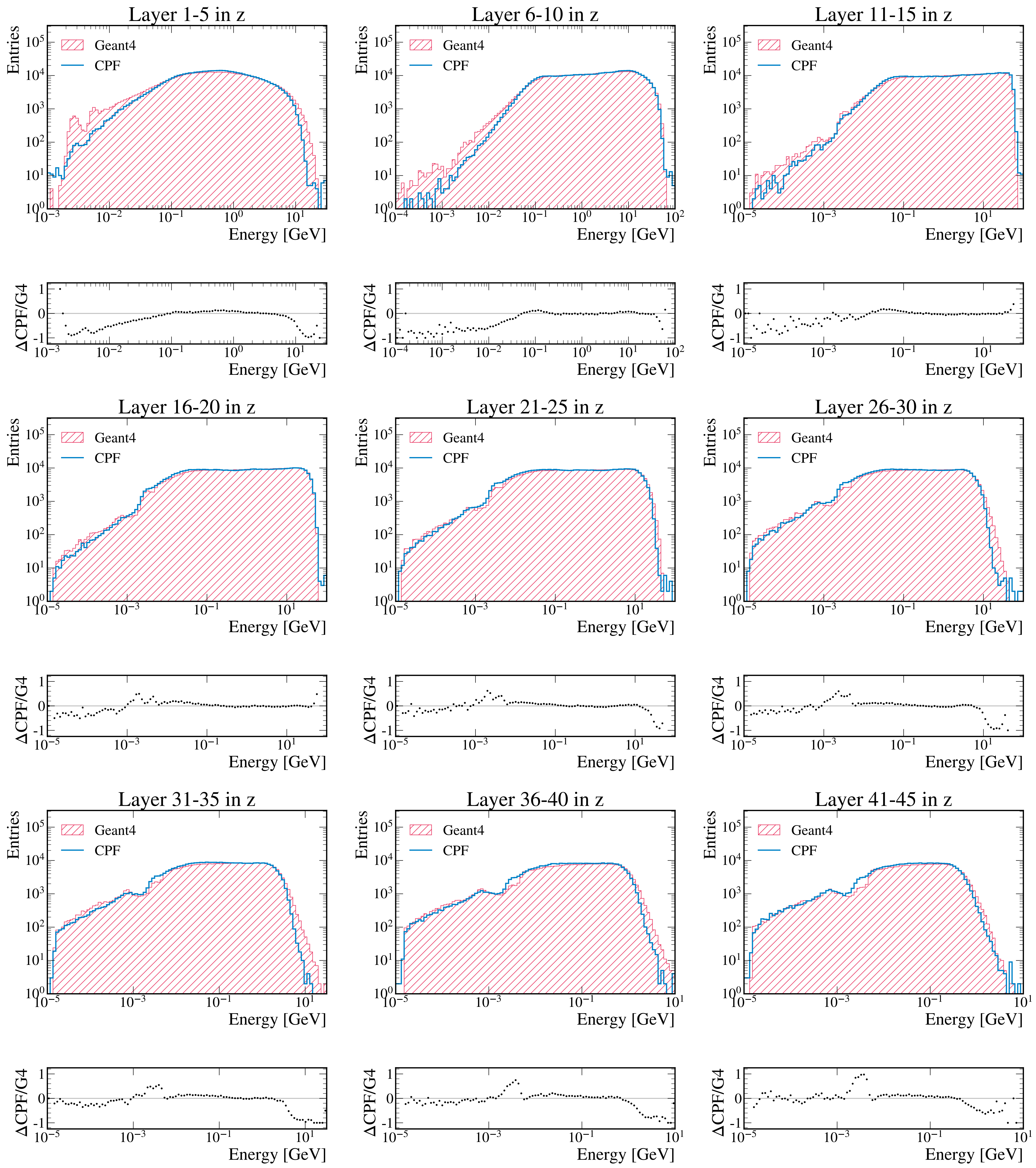
Dataset 3



⇒ Shower shapes are modelled well

Energy distributions

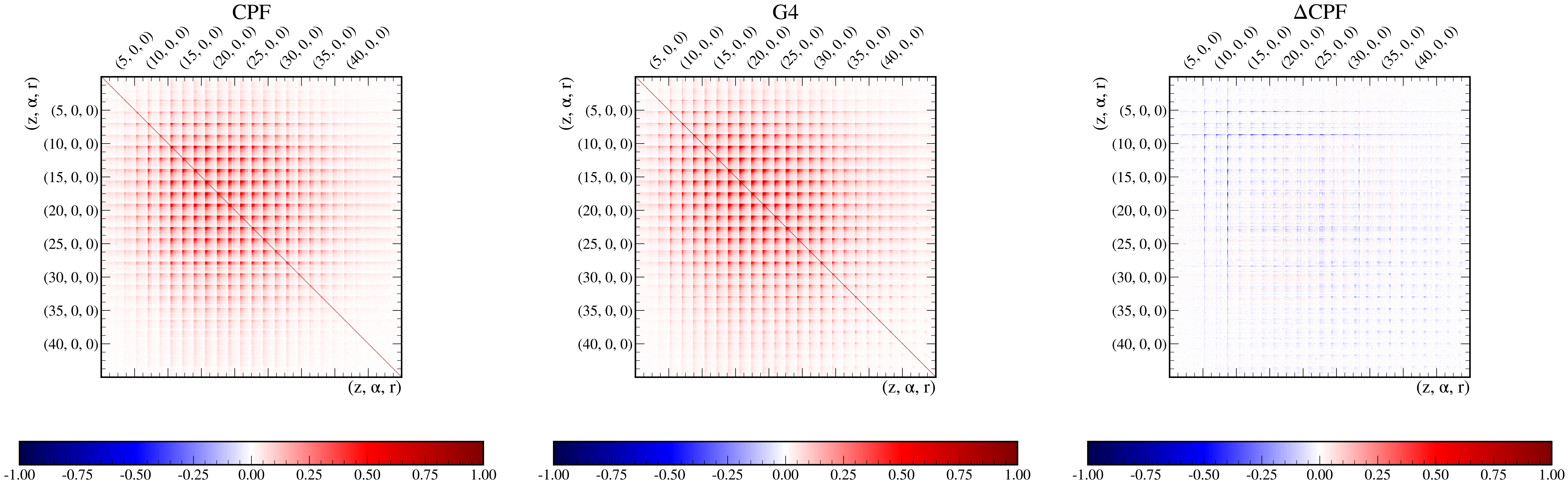
Dataset 3



- Bulk area is well modelled in each layer
- Tails are not always well matching

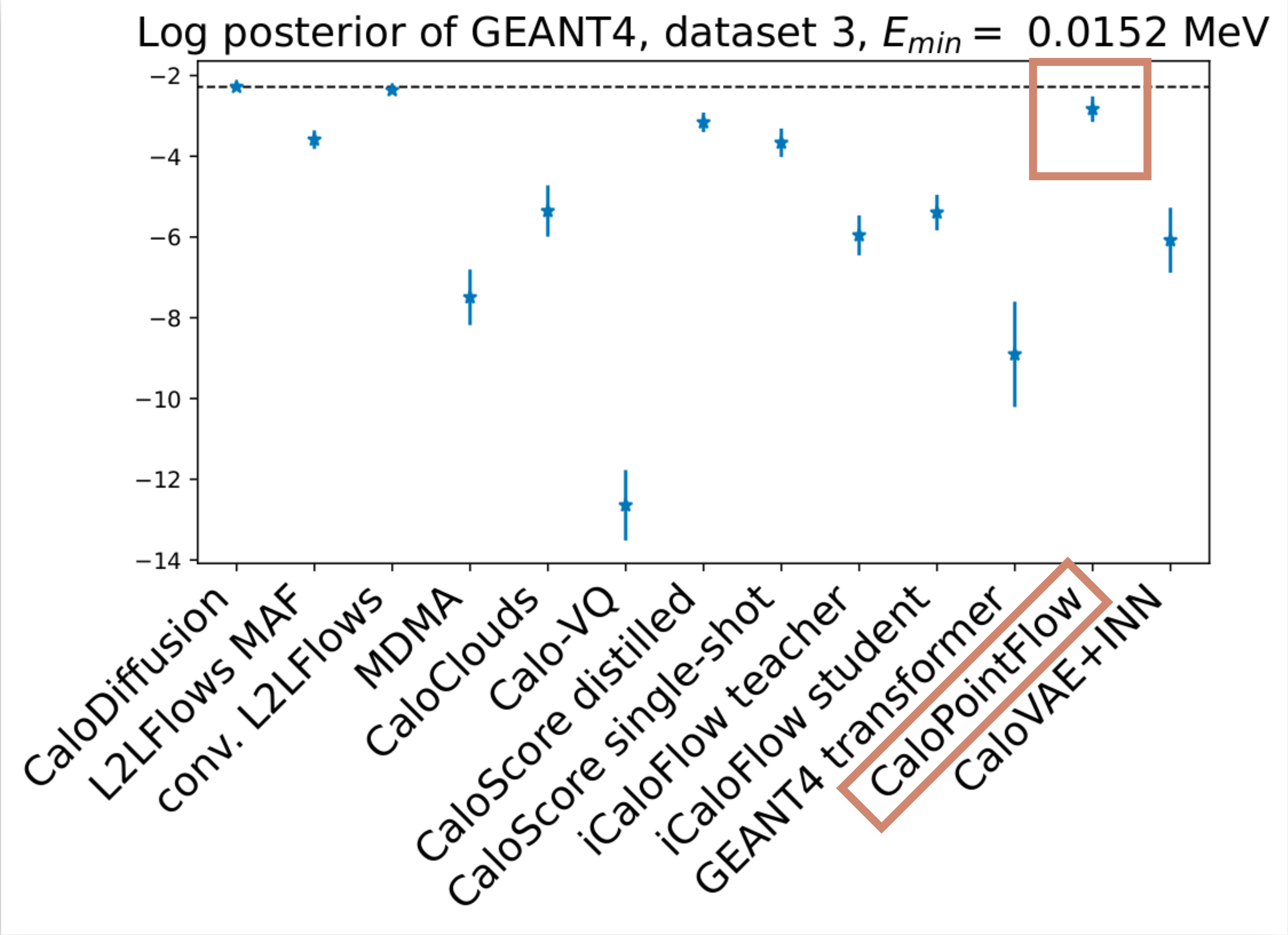
Correlation coefficients

Dataset 3



⇒ Correlations are modelled well

Log posterior



CaloPointflow II ranks third on dataset 3

From Claudius Krause, for more details ⇒ talk on Thurday

Conclusion

- CaloPointFlow I
- Improvements of CaloPointFlow II
  - CDFDequantization
  - DeepSetFlow
  - Mitigate multiple hit problem by randomly assigning α
- CaloPointflow II ranks third on dataset 3

Conclusion

- CaloPointFlow I
- Improvements of CaloPointFlow II
  - CDFDequantization
  - DeepSetFlow
  - Mitigate multiple hit problem by randomly assigning α
- CaloPointflow II ranks third on dataset 3

Thank you for your attention!

BACKUP

DATASET 2 PLOTS

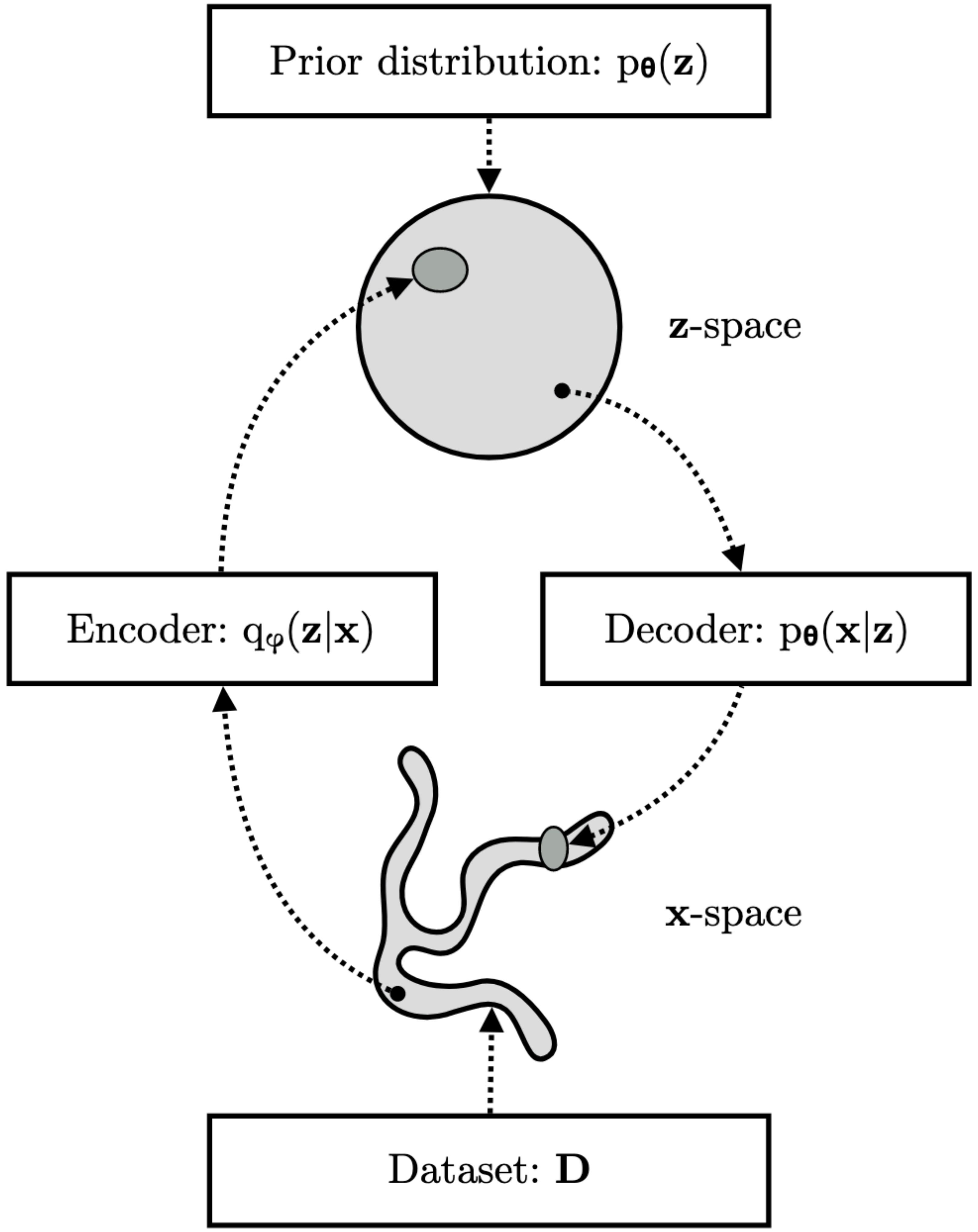
VAE

Variational Autoencoders

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

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

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

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

Decoding

Using a second Normalizing Flow

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

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

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

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

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

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

$w \leftarrow f(z)$

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

$L \leftarrow 0$

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

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

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

Adam($-\mathscr{L}$)

end for