QVAE w/ Pegasus & Zephyr

May 27

From Ben Nachman talk at the MLCommons Research Working Group

State-of-the-art: CaloScore

(first diffusion model applied to high energy physics!

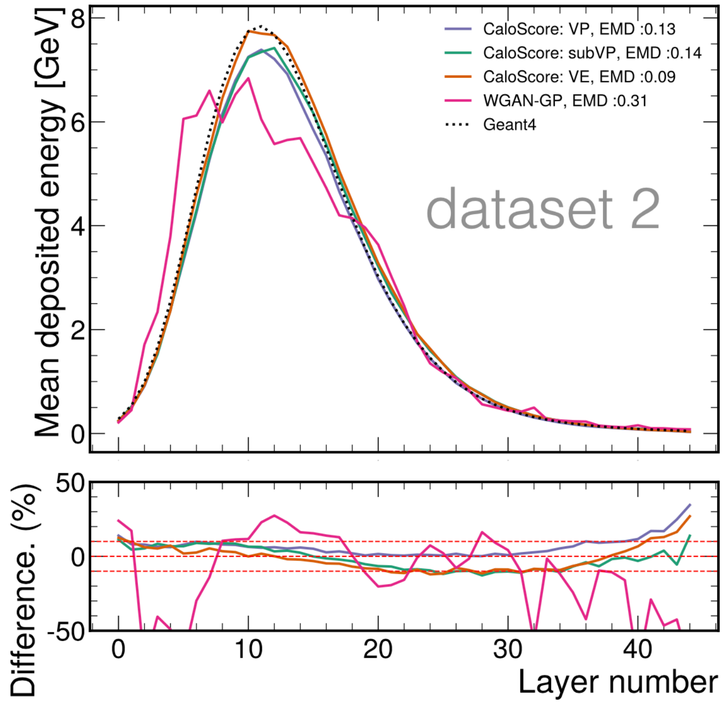
Dataset	N. of voxels	N. of weights	Time to 100 showers [s] CaloScore	WGAN-GP	Geant
dataset 1	384	32M	4.0	1.3	$\mathcal{O}(10^2 - 10^3)$
dataset 2	6480	1.4M	5.8	1.33	$\mathcal{O}(10^4)$
dataset 3	46080	1.7M	33.4	2.06	$\mathcal{O}(10^4)$

Sparse images; no smooth features as in natural images!

(train on slow simulation)
Input: 3-vector + particle type.
Output: image or point cloud

Model	# Parameters	Disk Size (Full)	Sample Time	AUC
Image	2,572,161	1016MB (62GB)	8036.19s	0.673
Point Cloud	620,678	509 MB	2631.41s	0.726

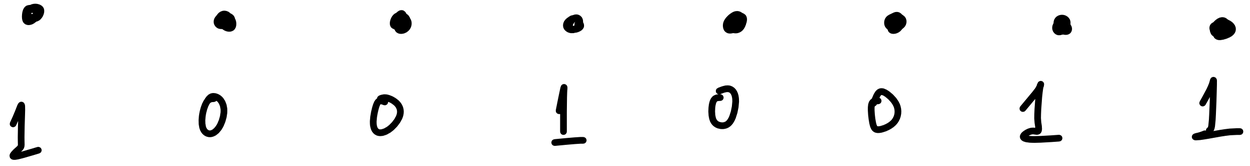
(on a slightly different dataset)



2308.03847, 2307.04780

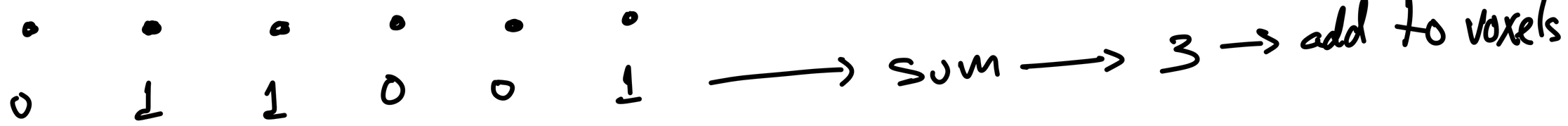
ENCODING PARTITION IN NEW ARCHITECTURE

Let us think of the partitions as a 1D vector



How to encode this ↑ to each encoder through the hierarchies?

Easiest most straightfwd way:



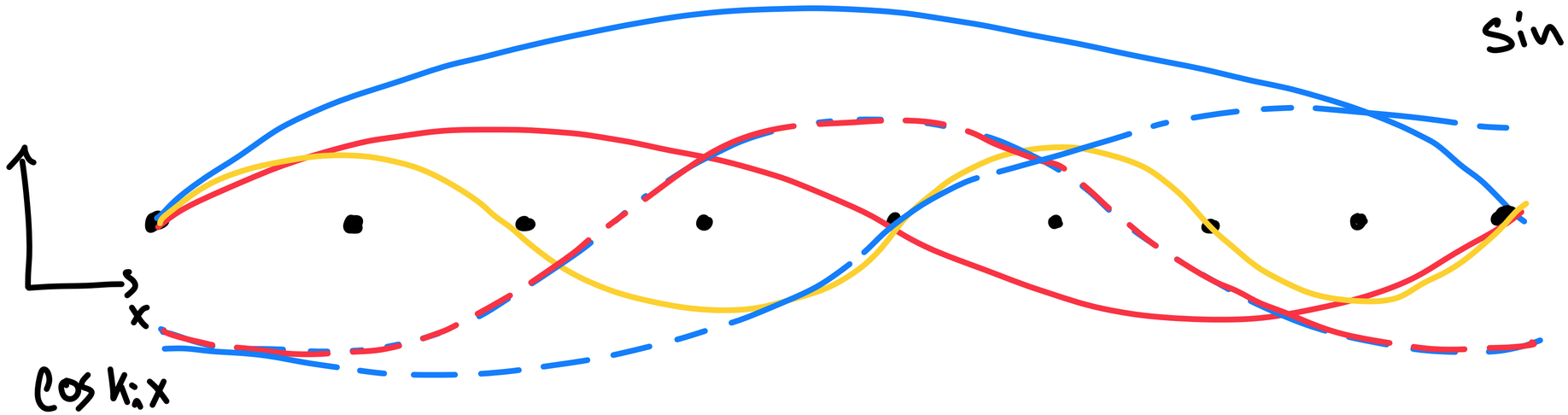
But this leads to degeneracy:

All zeros except
- 1 bit → deg → L
- 2 bits → deg → $\sum_{i=1}^{L-1} L-i$
- ⋮
- n bits → deg → $\frac{L!}{n_0!\, n_1!}$

0 0 1 1
0 1 0 1
1 0 0 1
0 1 1 0
1 0 1 0
1 1 0 0

$\frac{4!}{2!\,2!}$

How to avoid the degeneracy?



	1	0	0	1	1	1	0	0	0
pos	0	1	2	3	4	5	6	7	8

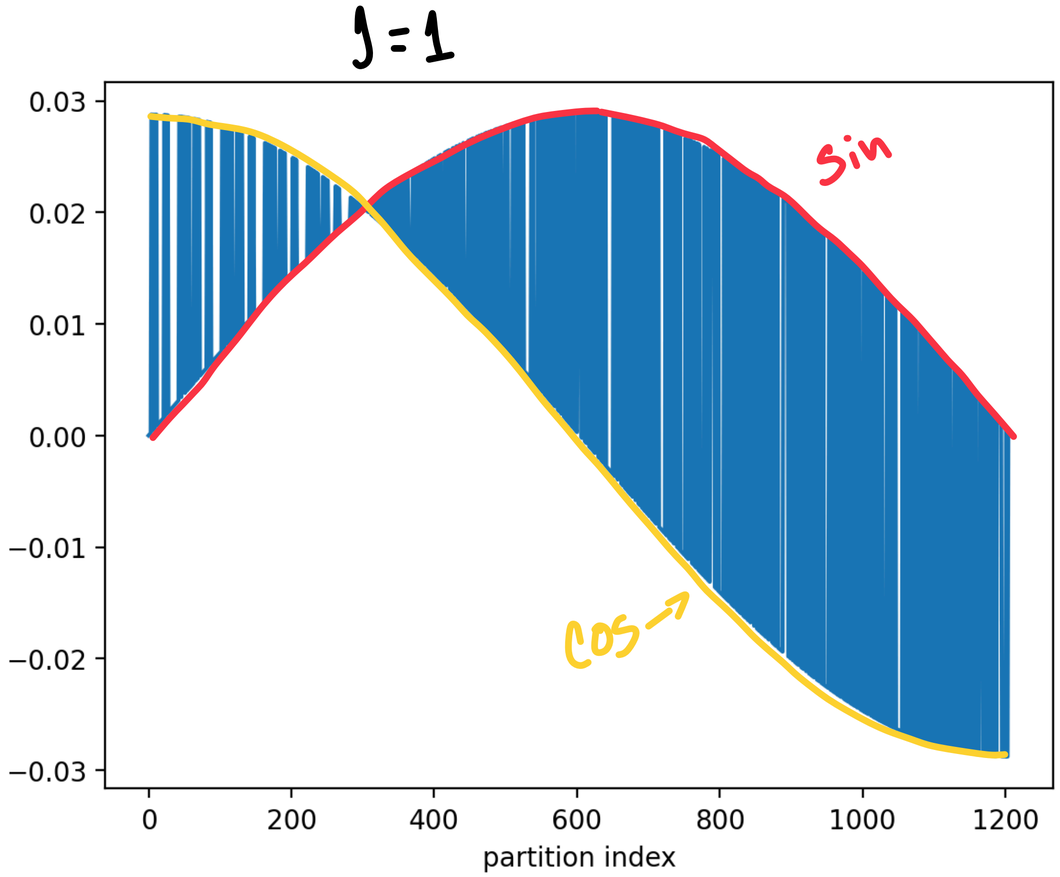
$$f_j(x_i) = \begin{cases} \cos[K_j\, pos(x_i)] & \text{if } x_i = 1 \\ \sin[K_j\, pos(x_i)] & \text{if } x_i = 0 \end{cases}$$

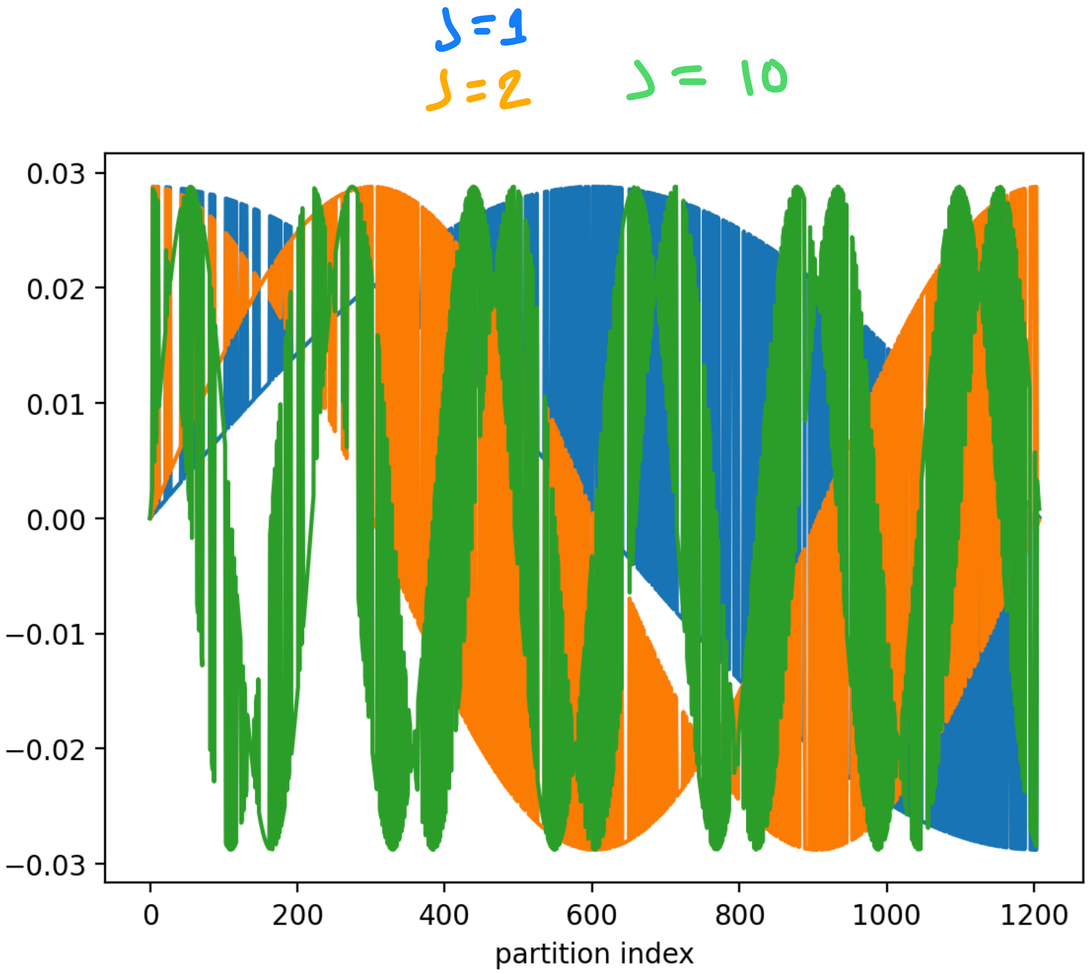
Boundary conditions

$\text{sen}\, Kx = 0$ for $x = 0$ and $x = M$

implies $K_j = \frac{j\pi}{M}$ w/ $j = 1, \dots, M-1$

$$F = \begin{bmatrix} 1 & \text{sen}K_1 & \text{sen}2K_1 & \cos 3K_1 & \cos 4K_1 & \cos 5K_1 & \text{sen}6K_1 & \text{sen}7K_1 & \text{sen}8K_1 \\ 1 & \text{sen}K_2 & \text{sen}2K_2 & \cos 3K_2 & \cos 4K_2 & \cos 5K_2 & \text{sen}6K_2 & \text{sen}7K_2 & \text{sen}8K_2 \\ \vdots & & & & & & & & \end{bmatrix}$$



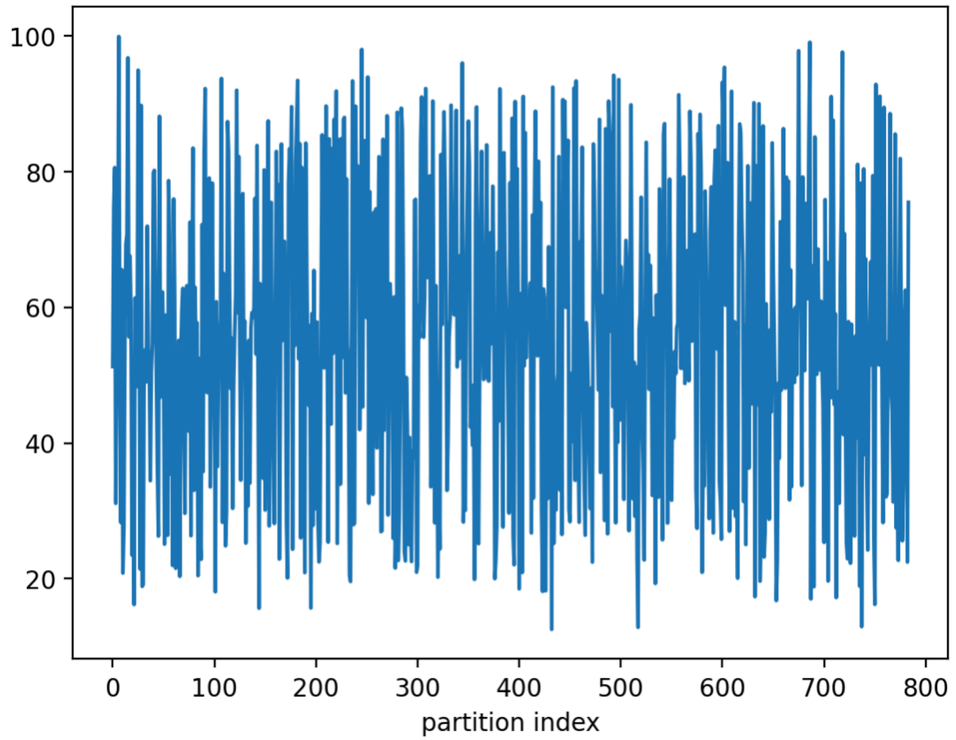


We still need to somehow encode partition to a single number to be added to the voxels.

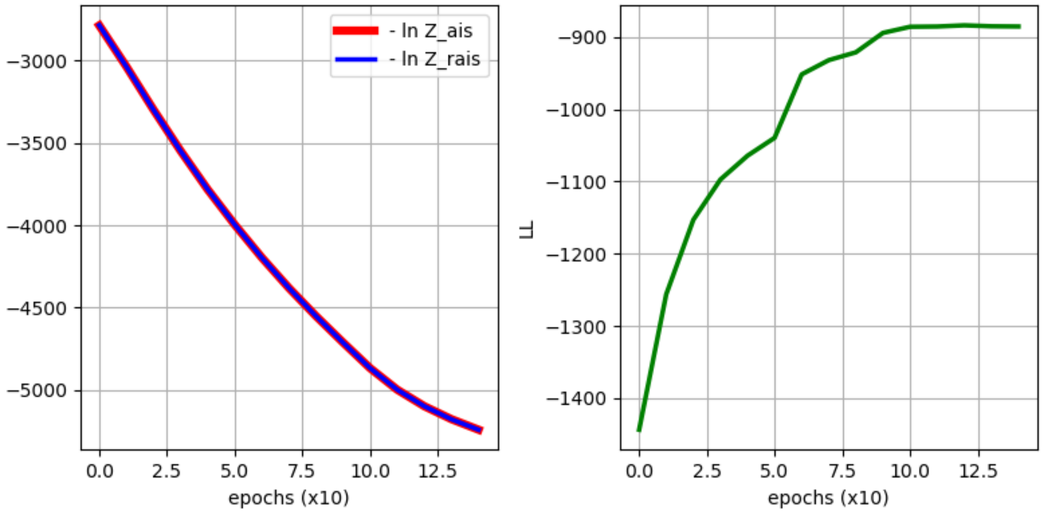
For each partition and each sample we have a matrix

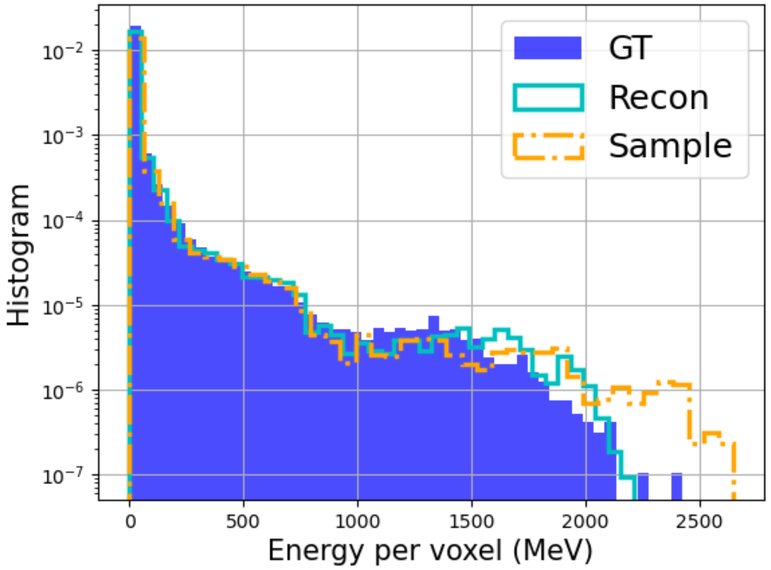
F where $F_{ji} = f_j(X_i)$

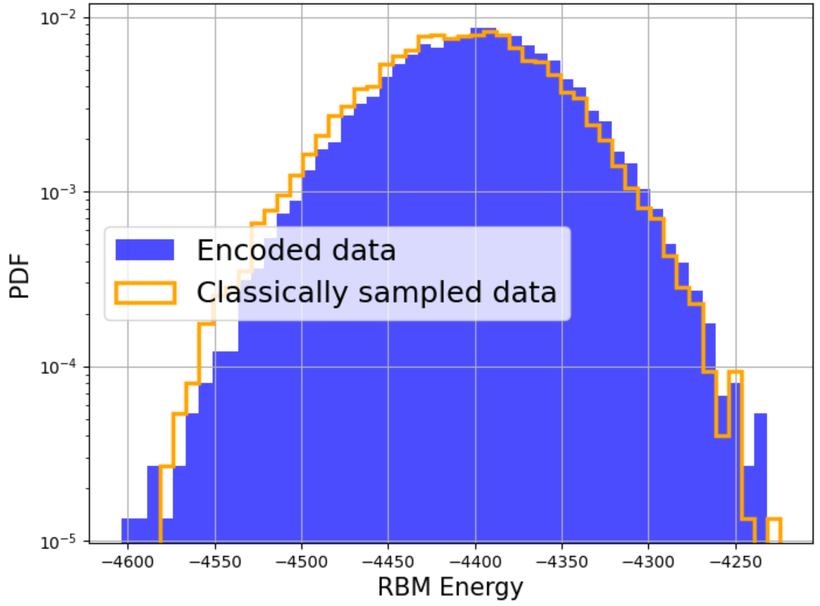
Then we define the scalar $S = \sum_{j,i} F_{ji}$

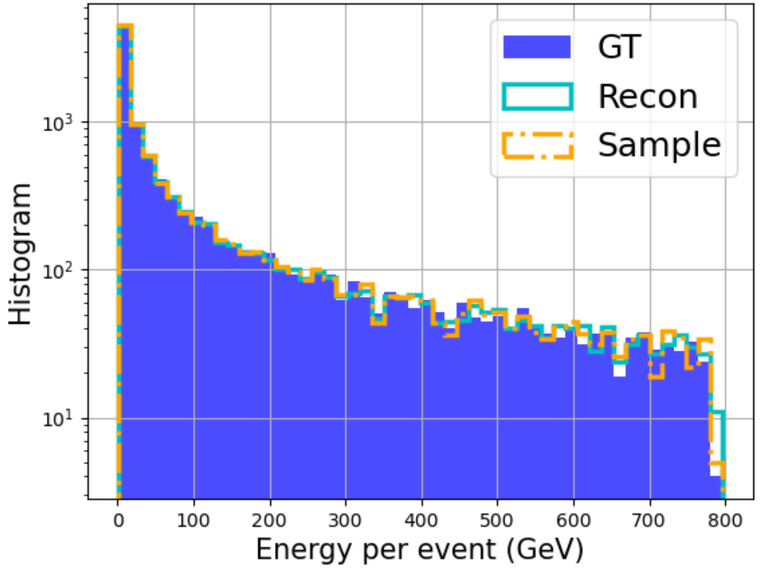


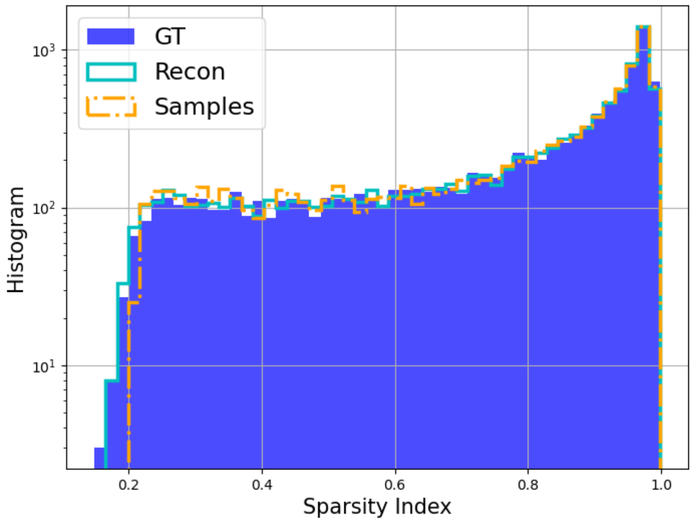
PEGASUS


- ln Z_ais
- ln Z_rais
epochs (x10)
LL
epochs (x10)

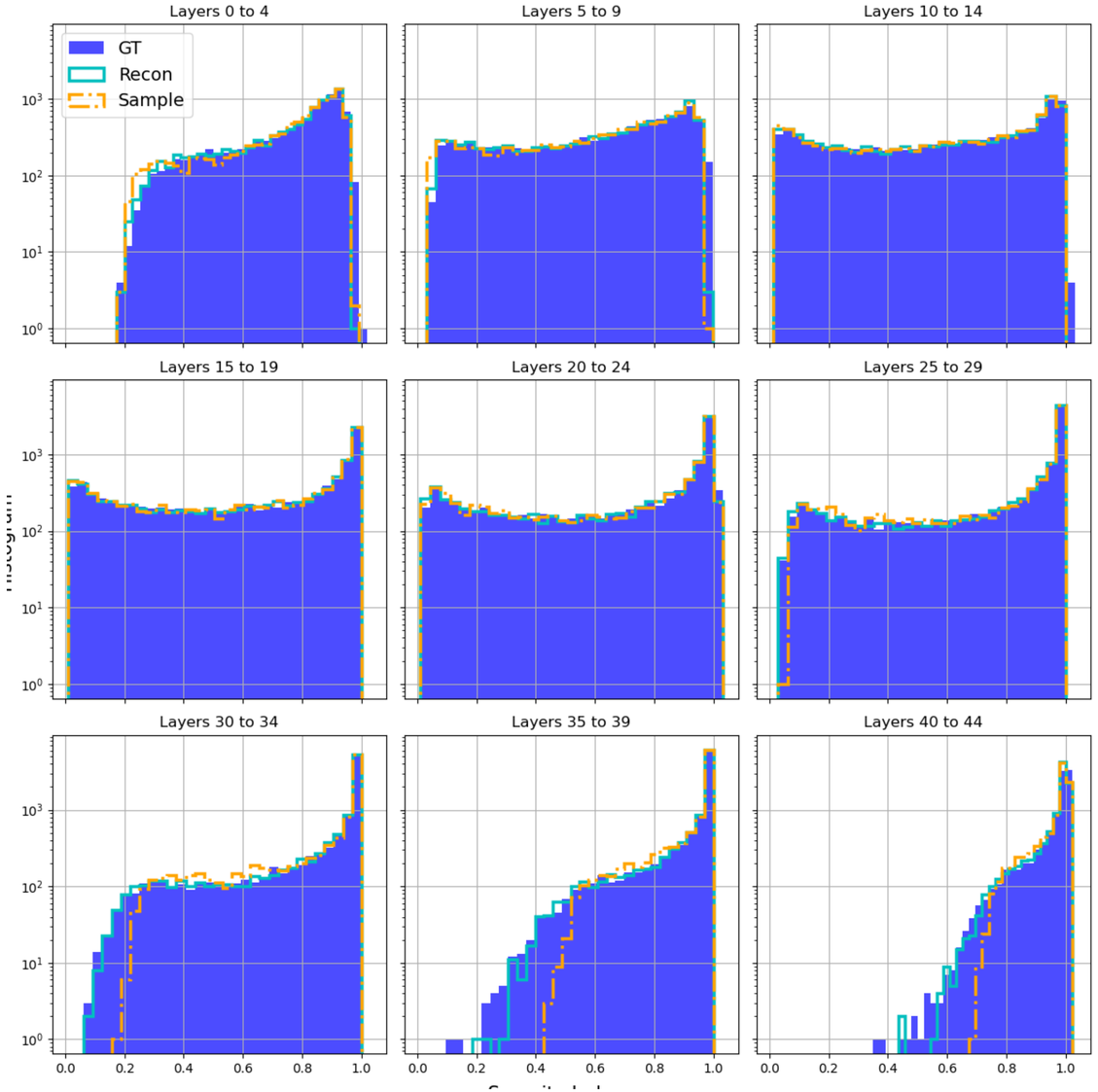

GT
Recon
Sample
Histogram
Energy per voxel (MeV)

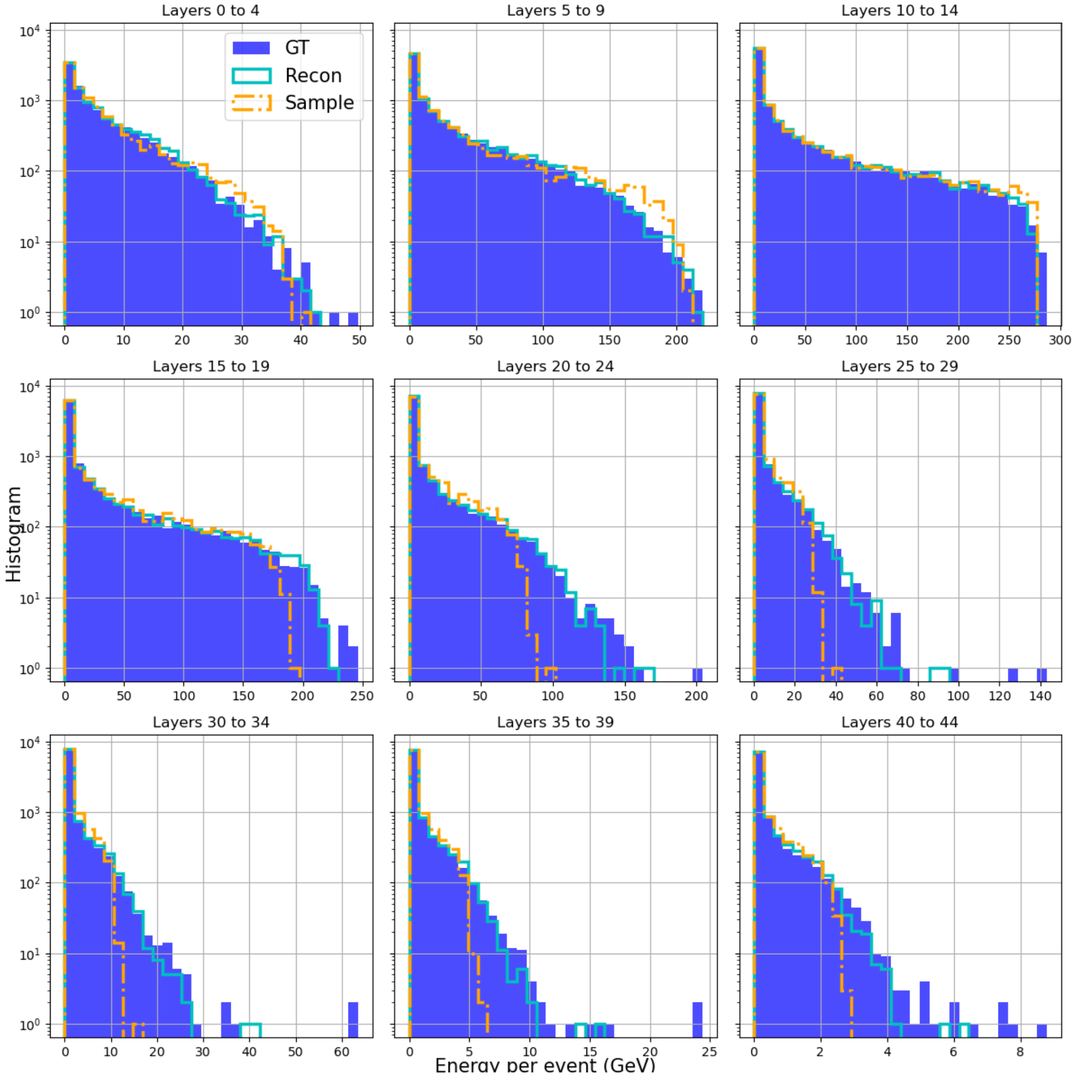

Encoded data
Classically sampled data
PDF
RBM Energy


GT
Recon
Sample
Histogram
Energy per event (GeV)

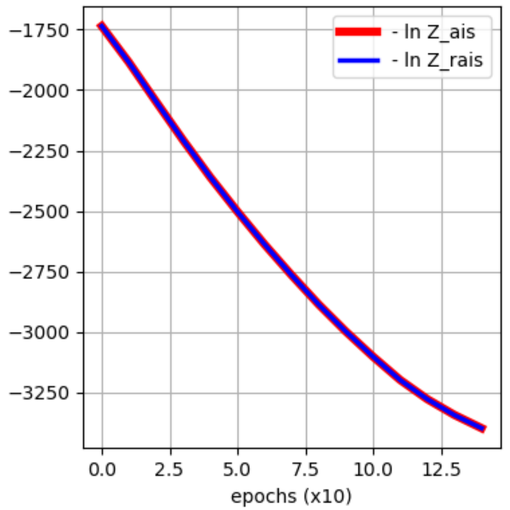

GT
Recon
Samples
Histogram
Sparsity Index

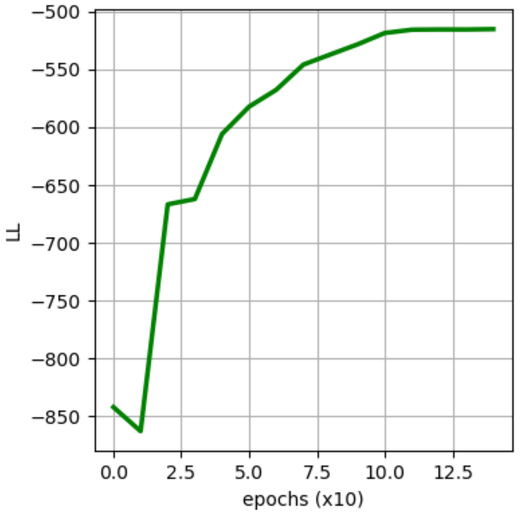
PEGASUS

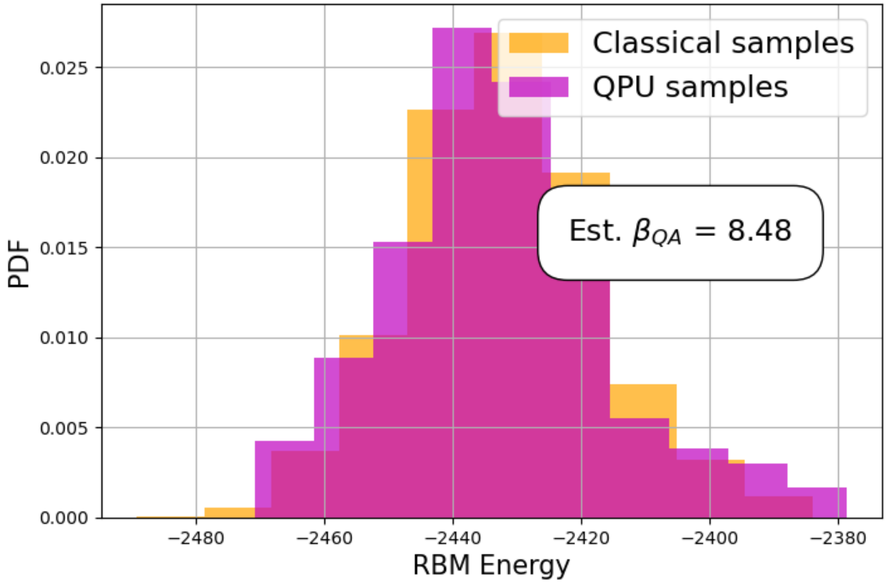

Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
GT
Recon
Sample

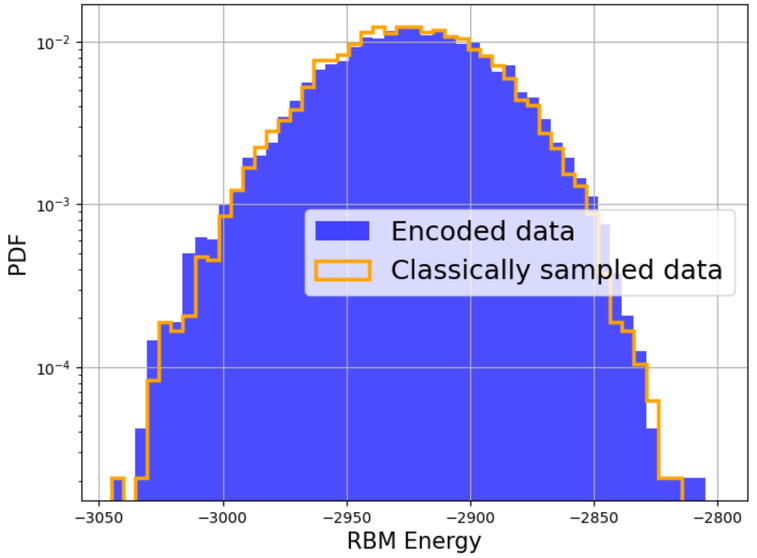

Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
GT
Recon
Sample
Histogram
Energy per event (GeV)

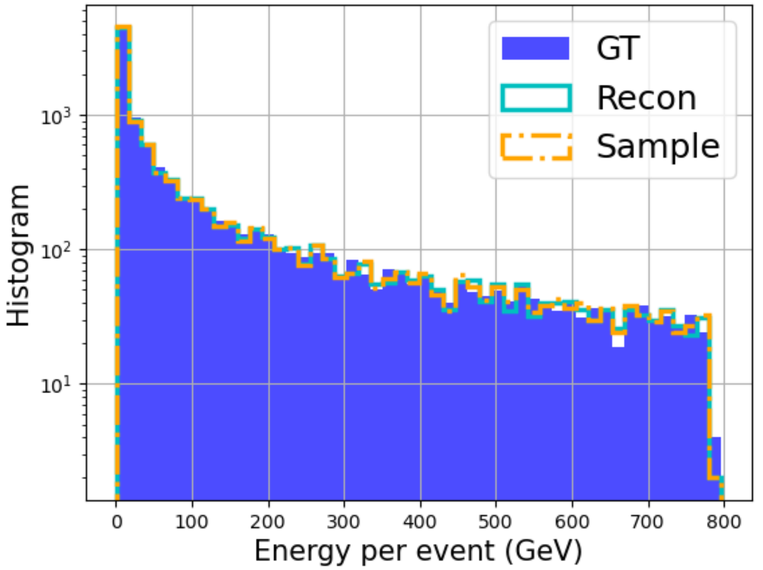
ZEPHYR

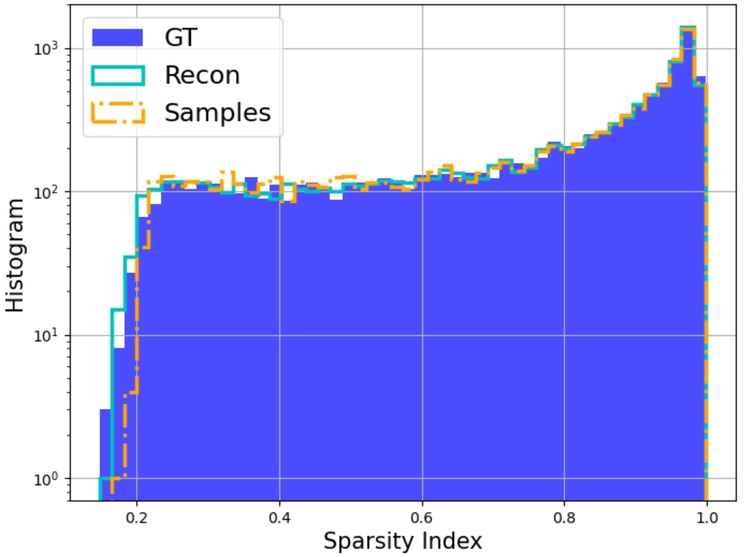

- ln Z_ais
- ln Z_rais
epochs (x10)


LL
epochs (x10)

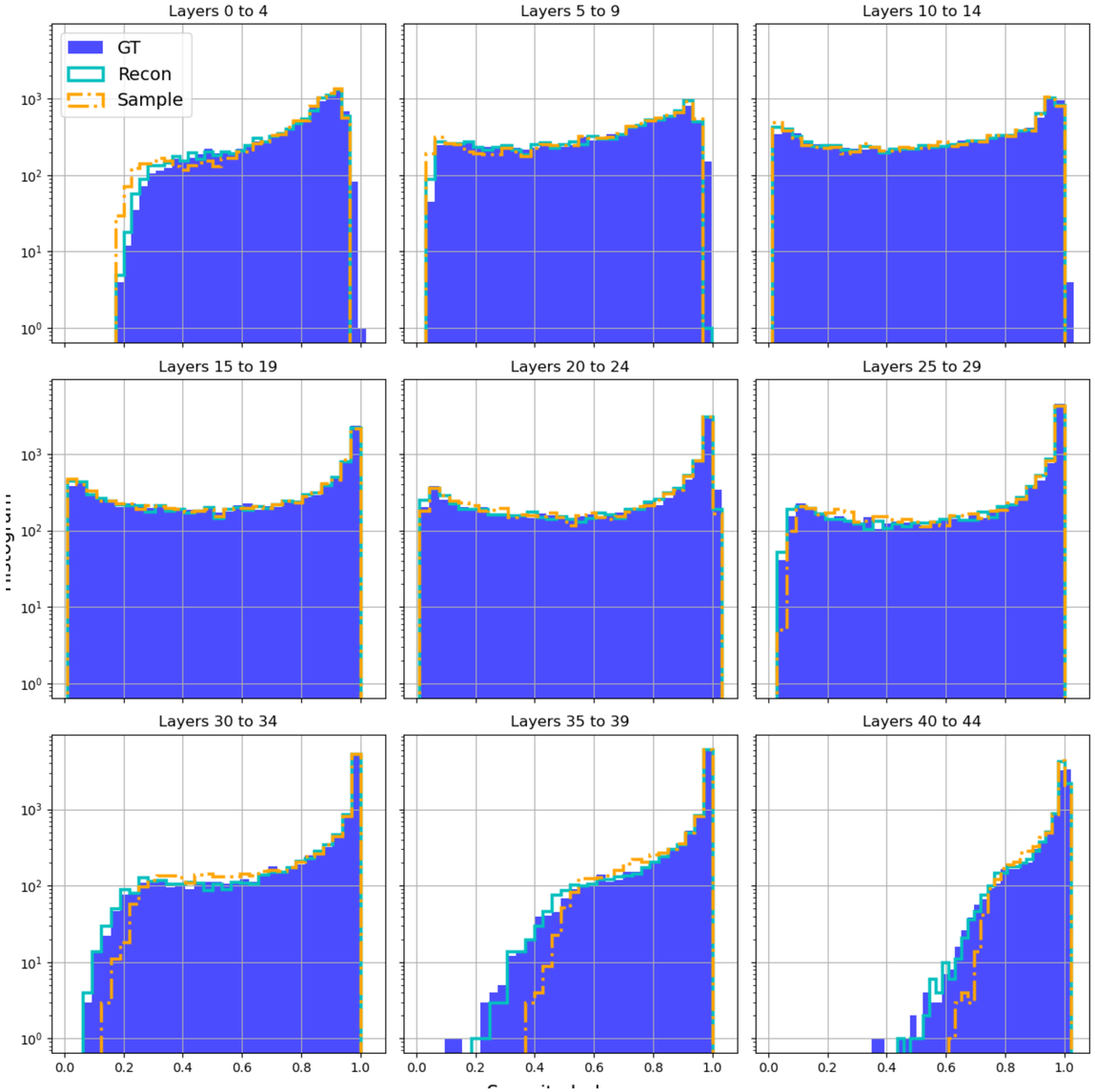

Classical samples
QPU samples
Est. β_{QA} = 8.48
PDF
RBM Energy

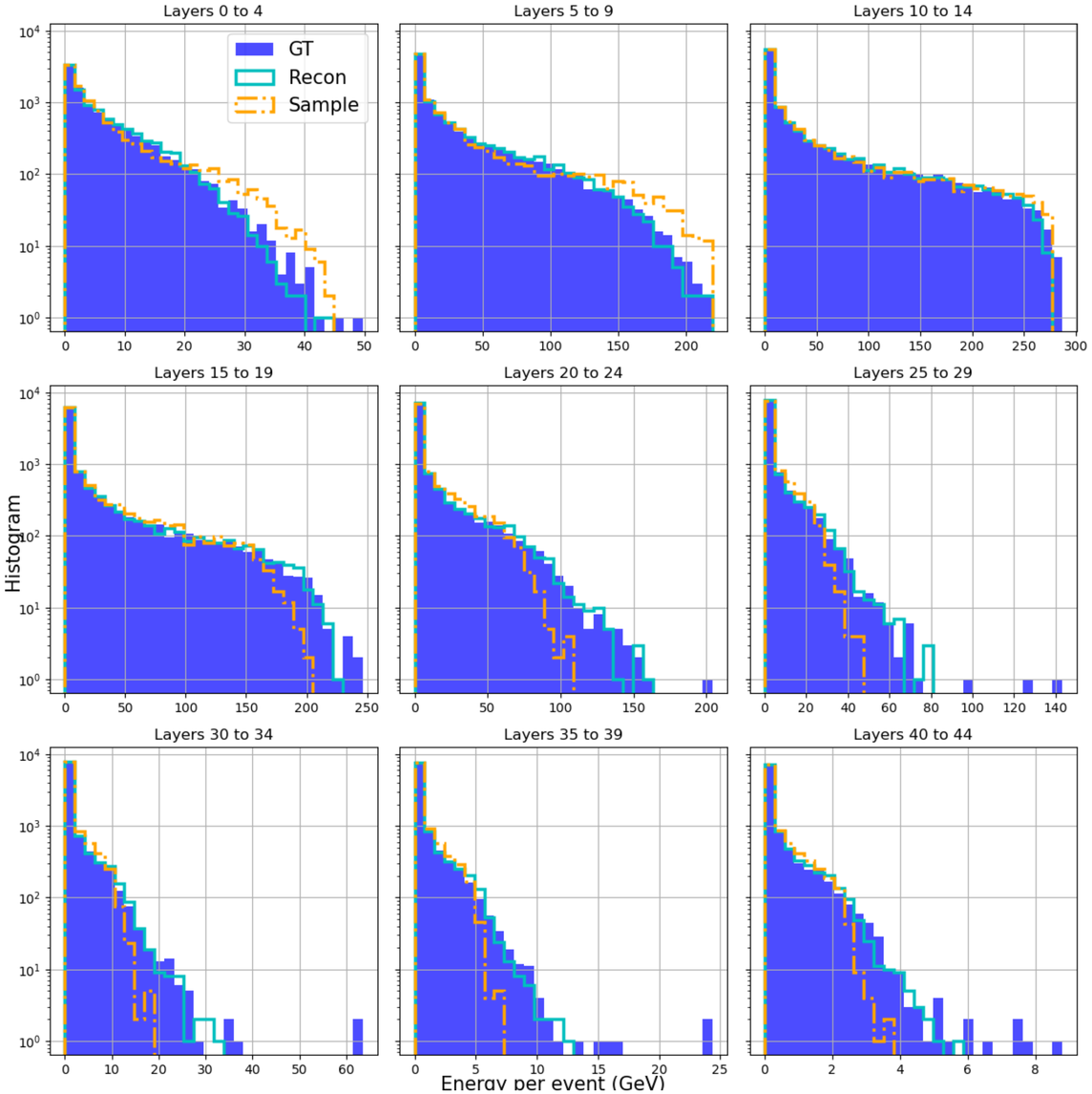

Encoded data
Classically sampled data
PDF
RBM Energy


GT
Recon
Sample
Histogram
Energy per event (GeV)


GT
Recon
Samples
Histogram
Sparsity Index

ZEPHYR


Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
GT
Recon
Sample
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44


Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
GT
Recon
Sample
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
Histogram
Energy per event (GeV)

Data preprocessing

$\{v_i\}_{i=1}^{6480}$: shower

$\epsilon_i = \frac{v_i}{E_{inc}}$: reduced deposited energy in voxel ith

$x_i = \delta + (1 - 2\delta)\epsilon_i \qquad \delta = 10^{-6}$

$u_i = \log \frac{x_i}{1 - x_i}$

$u_i' = \frac{u_i - \langle u_i \rangle}{\sigma_{u_i}}$

Metrics

- RBM LL vs Epochs
- Time benchmarks
- Histograms of energy, sparsity, etc.
- Jensen-Shannon Div, Hellinger entropy, Kulbach-Liebler Div