QVAE w/ Pegasus

Dec 1st

- QVAE
  - Architectures
    - CNN
    - FCN
  - Energy incidence
    - Condition on encoder and decoder
    - Condition on encoder
    - Unconditionalized
    - Modulated energy => Can lead to learning how to modulate more features, position of voxels, angles, etc.
  - Results/metrics
    - Energy histogram
    - Sparsity histogram
    - Conditionalized energy and sparsity histogram (NOT GOOD)
- RBM
  - Topology
    - Chimera-like
    - Pegasus
  - Metrics
    - Energy distribution for encoded and RBM Gibbs samples
    - Zais and Zrais estimates for partition function => log-likelihood of model
  - Dwave
    - Sehmi’s method
    - Fast stein. Not robust but could be helpful?
    - Hao’s method
- Theory. Work in progress

- In the previous slides (Nov 18th), model super-valley-246 seem to check every aspect decently well (RBM energy overlap between encoded data and Gibb samples, good recon, good samples, robust RBM).

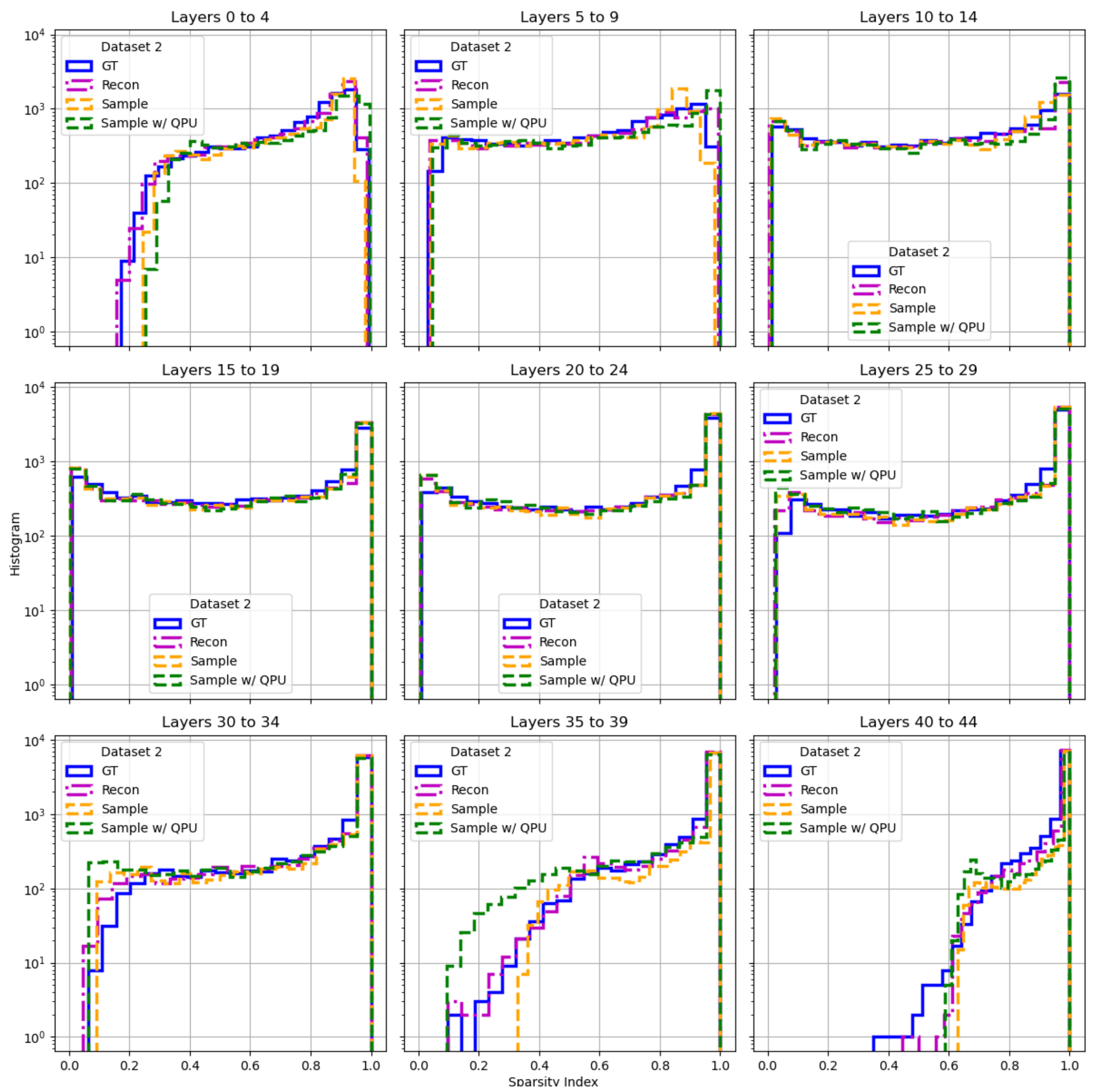
- Here we tried conditioning the energy by ***positional embedding***. Basically, instead of concatenating the “incident energy”, we simply add it to the torch tensor in the encoder and the decoder.

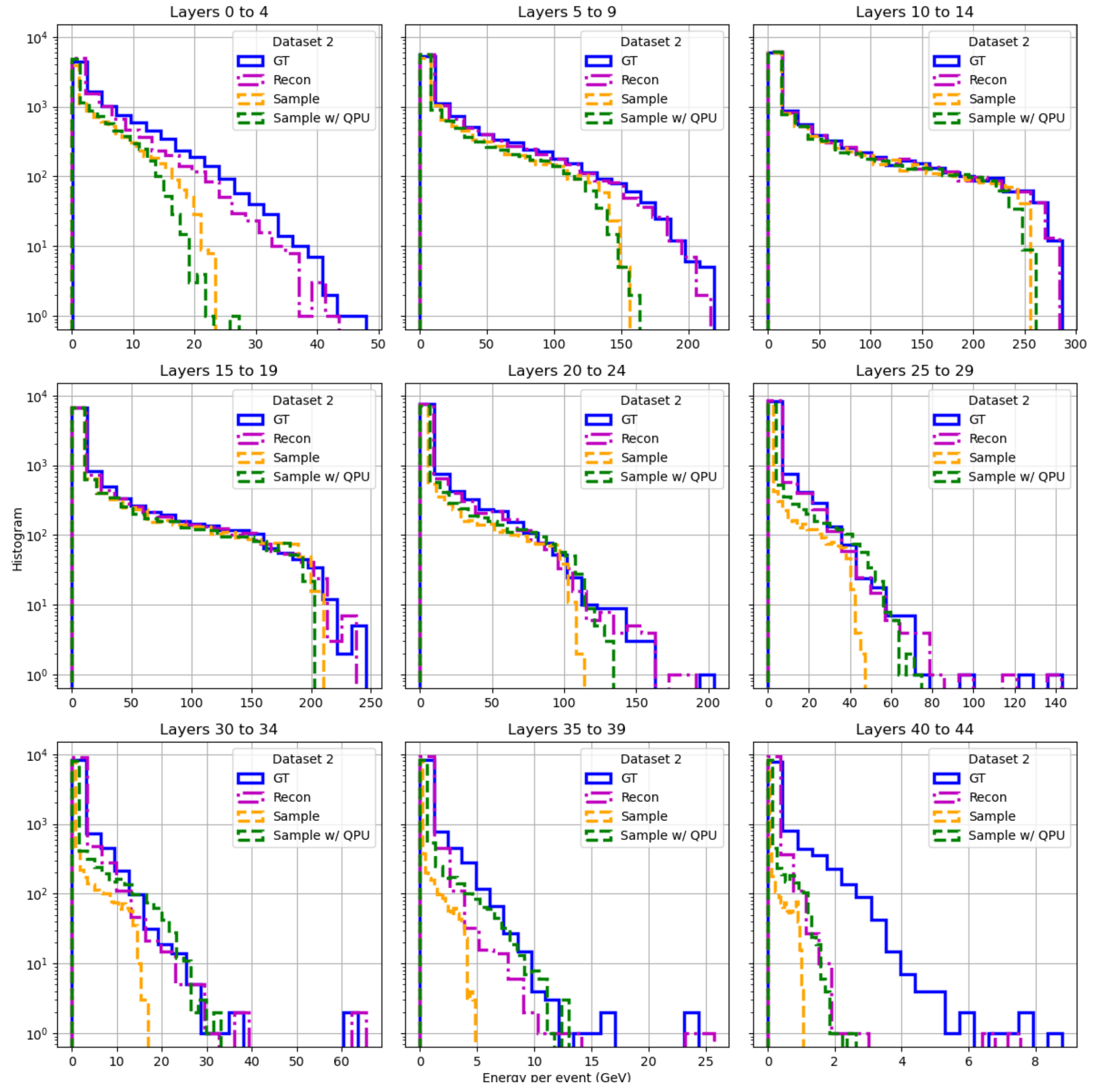
- We also tried the ***positional encoding*** of the voxels by adapting the positional encoding done in transformers for LLM. In LLM we have 1D vectors, but in our case, we have a cylinder, so we applied positional encoding *a la Attention is all you need* in (r,\phi,z).

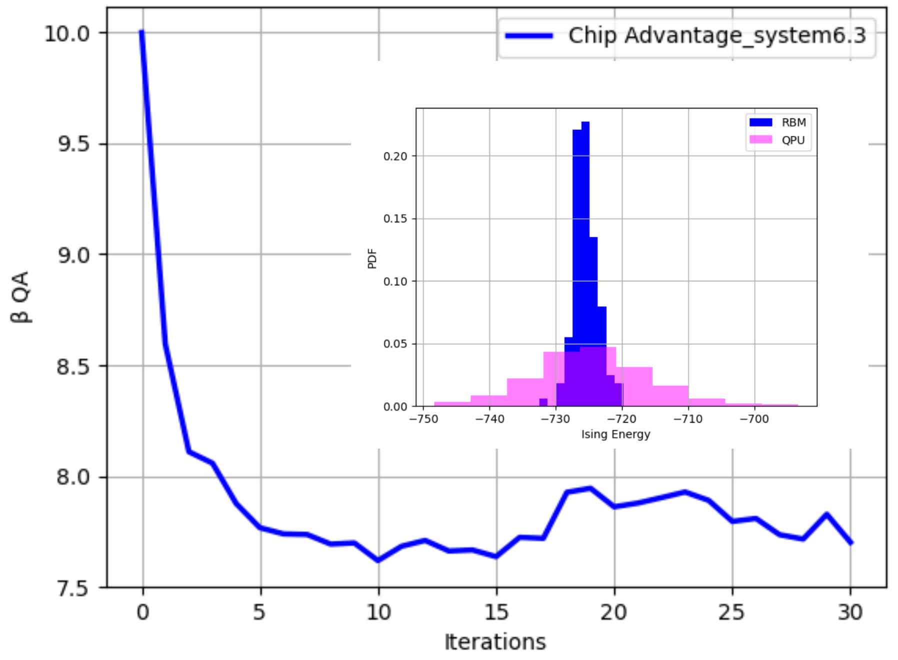
- For the latter model, there were some instabilities during training. So the results are not conclusive.

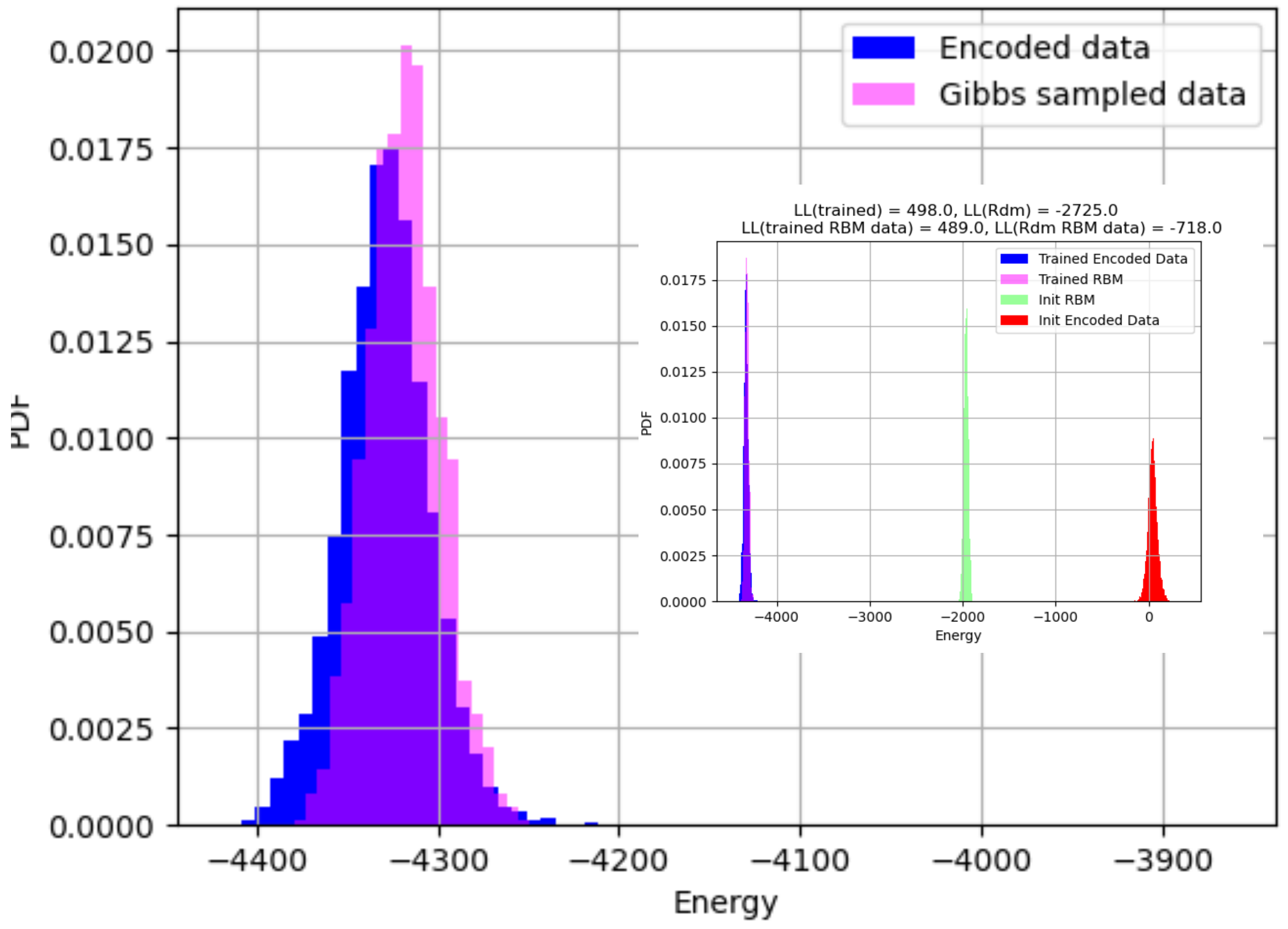
super-valley-246

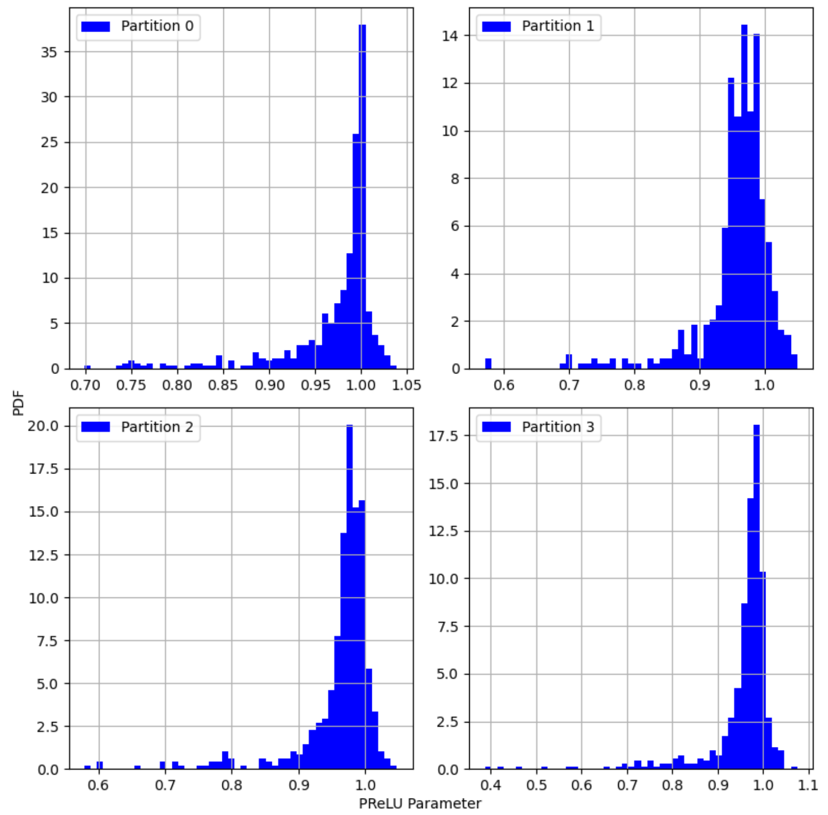
CNN + fully conditioned+scaled data

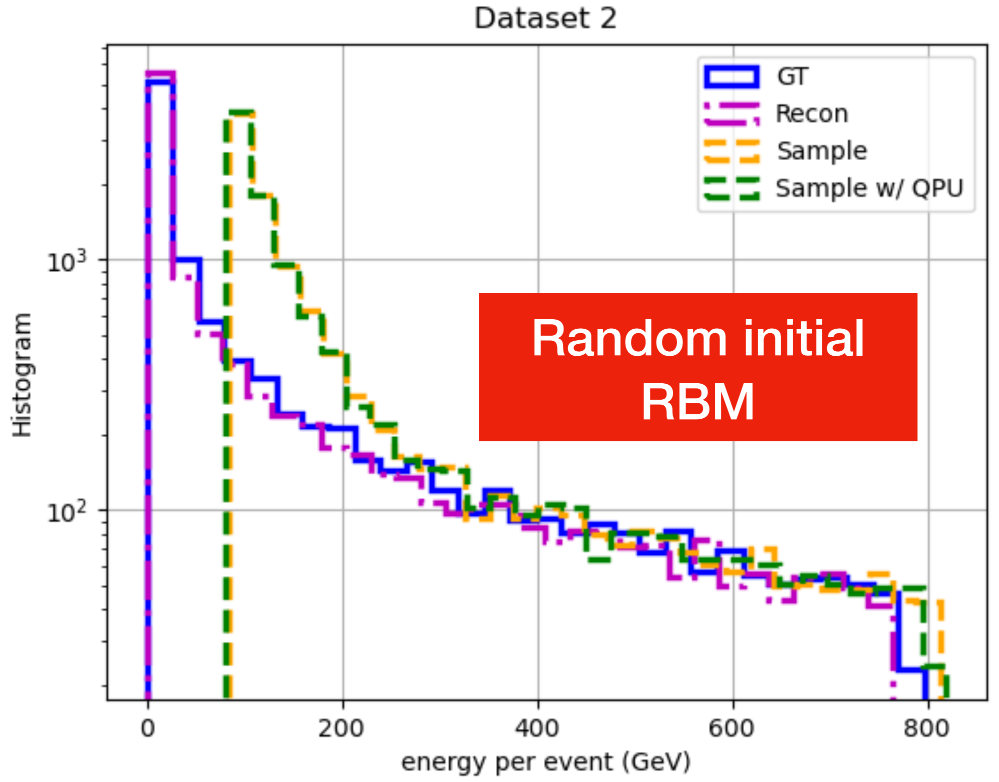


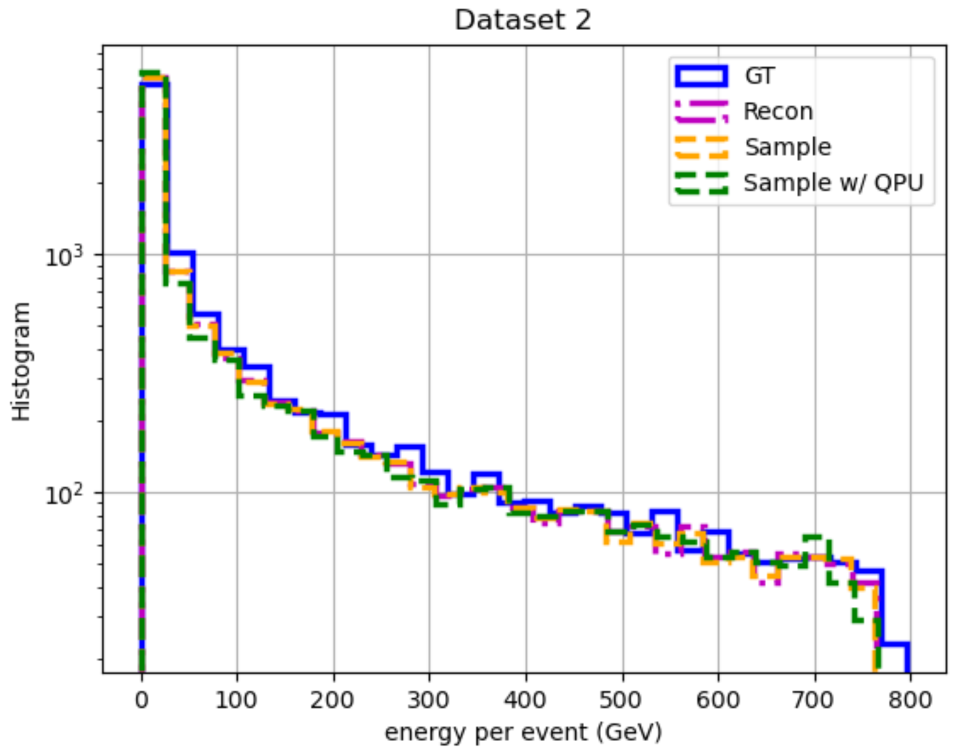


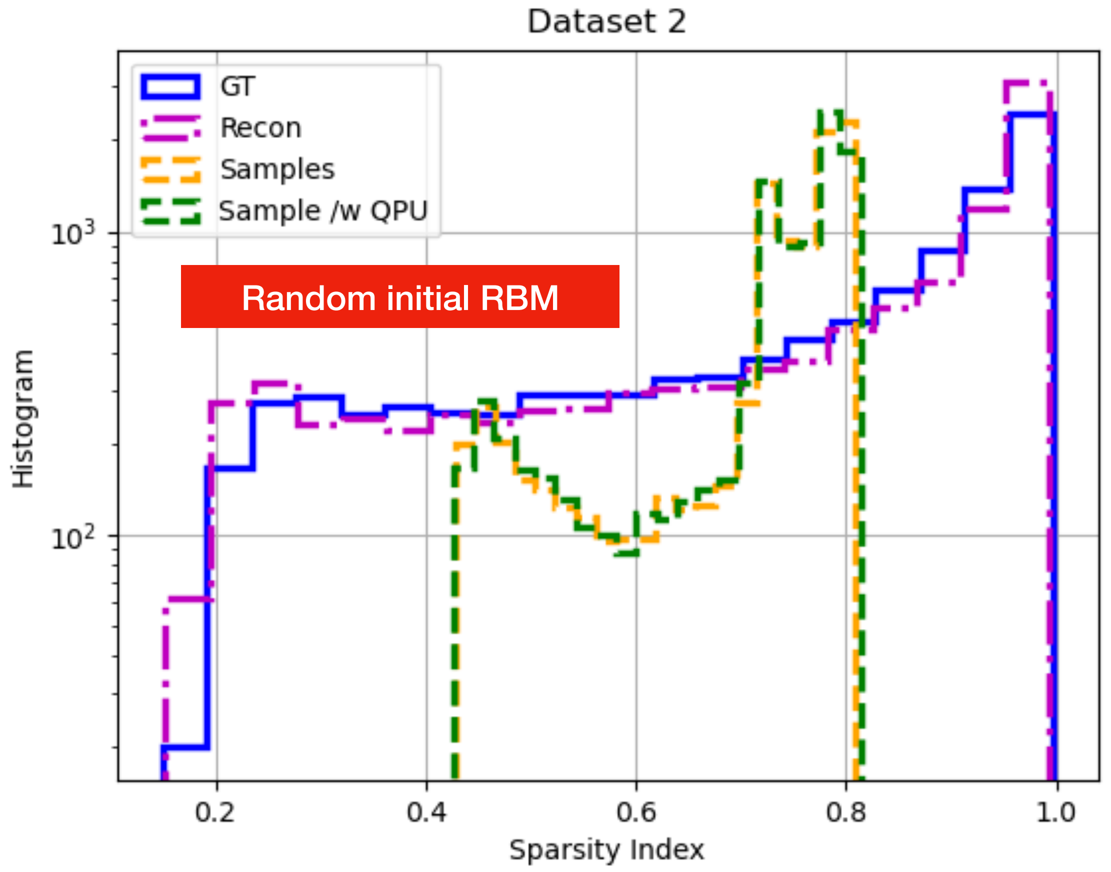


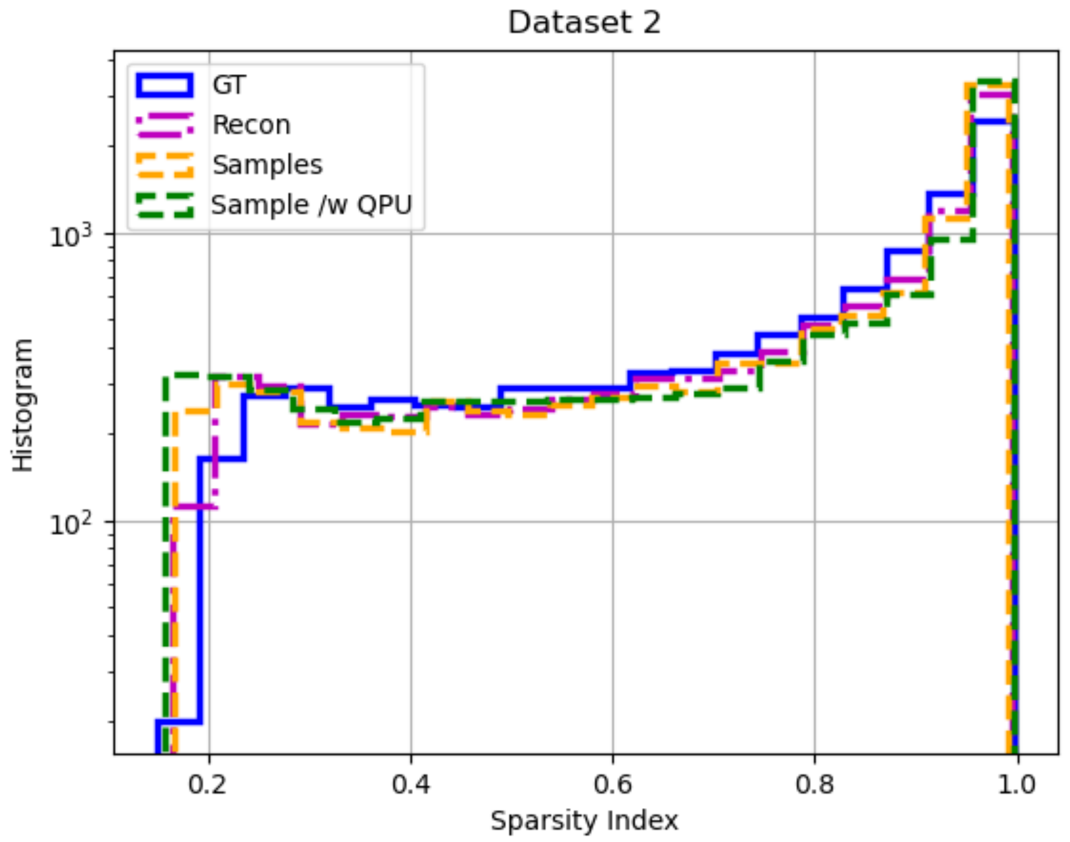




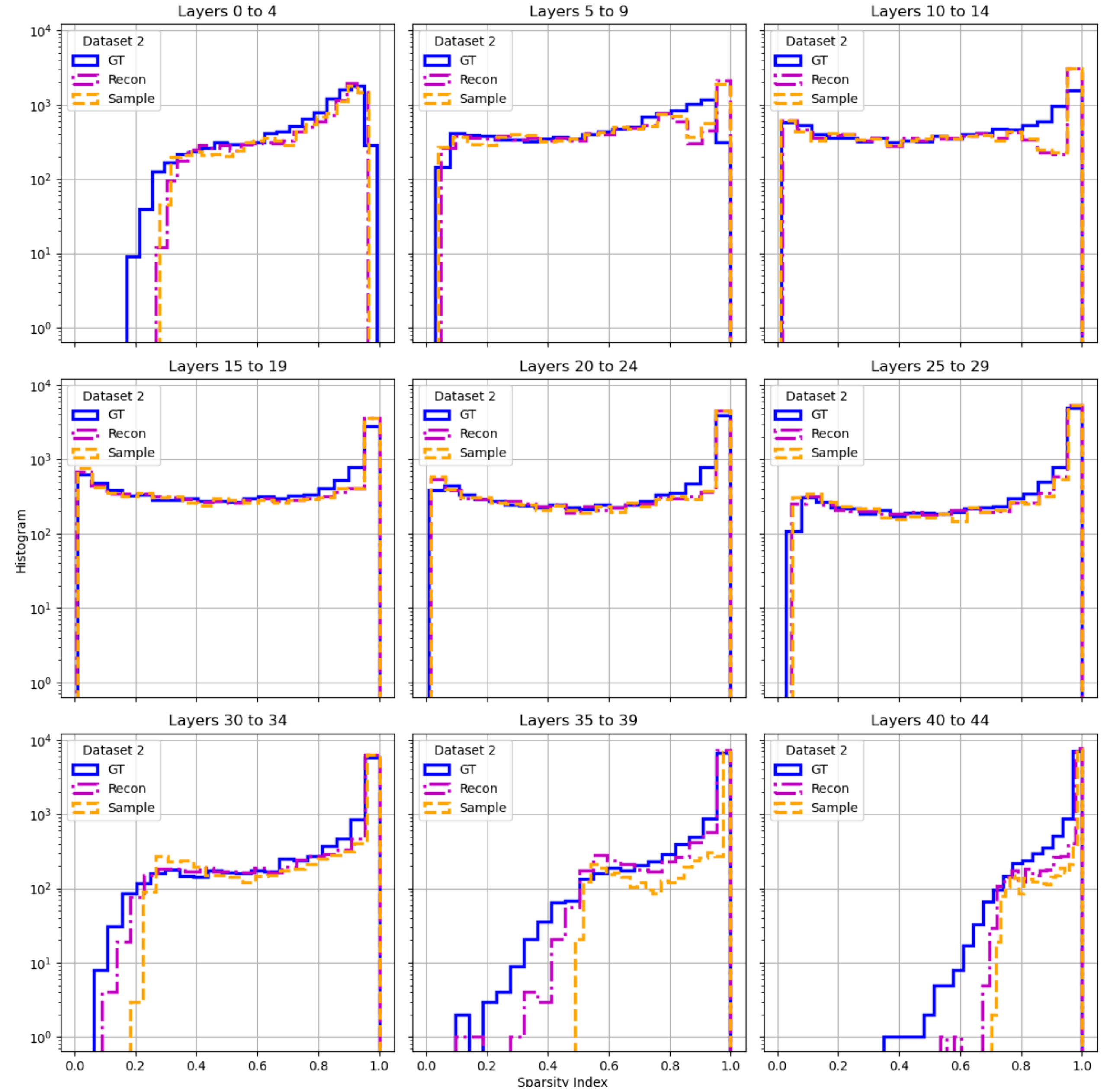






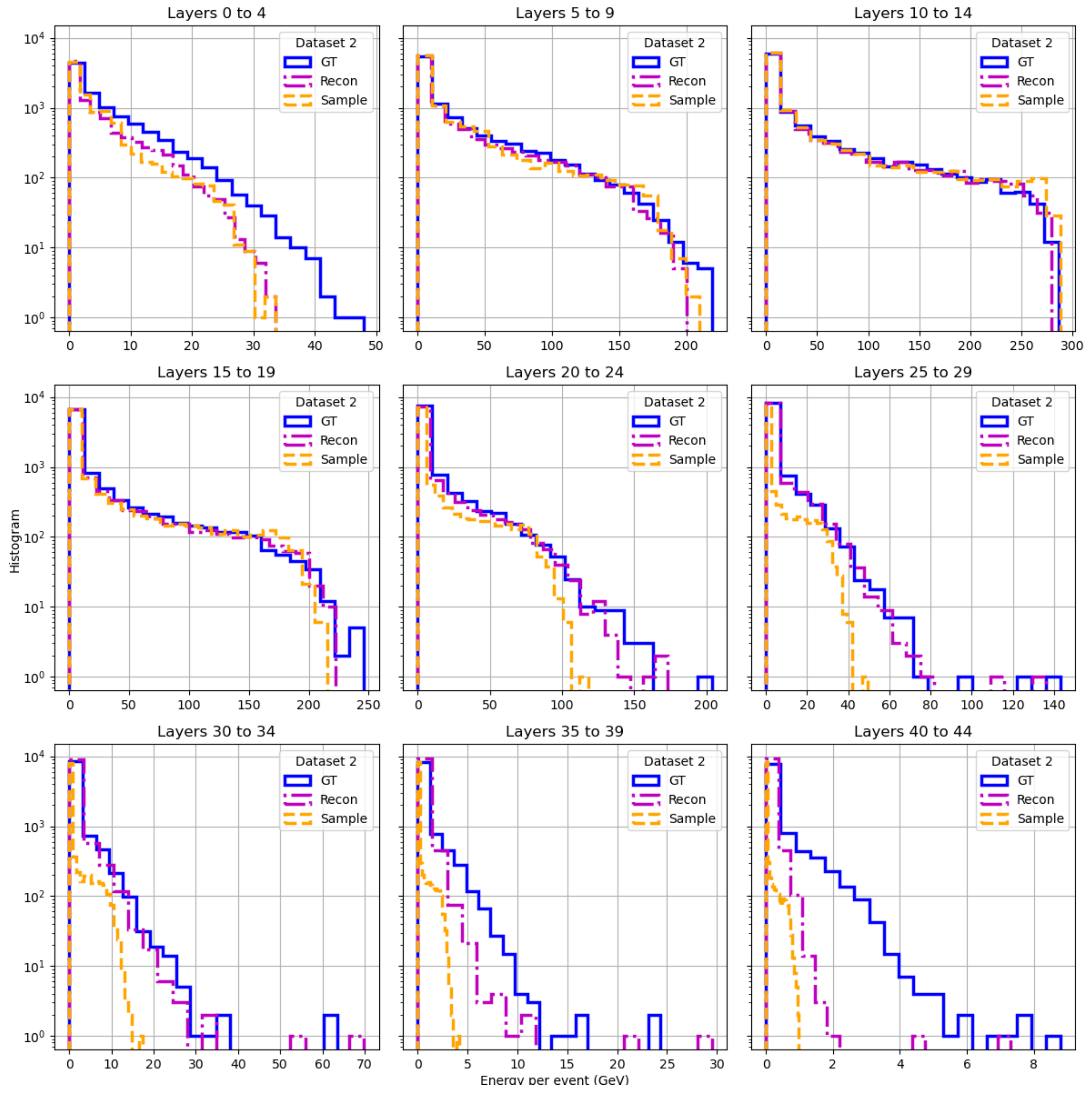


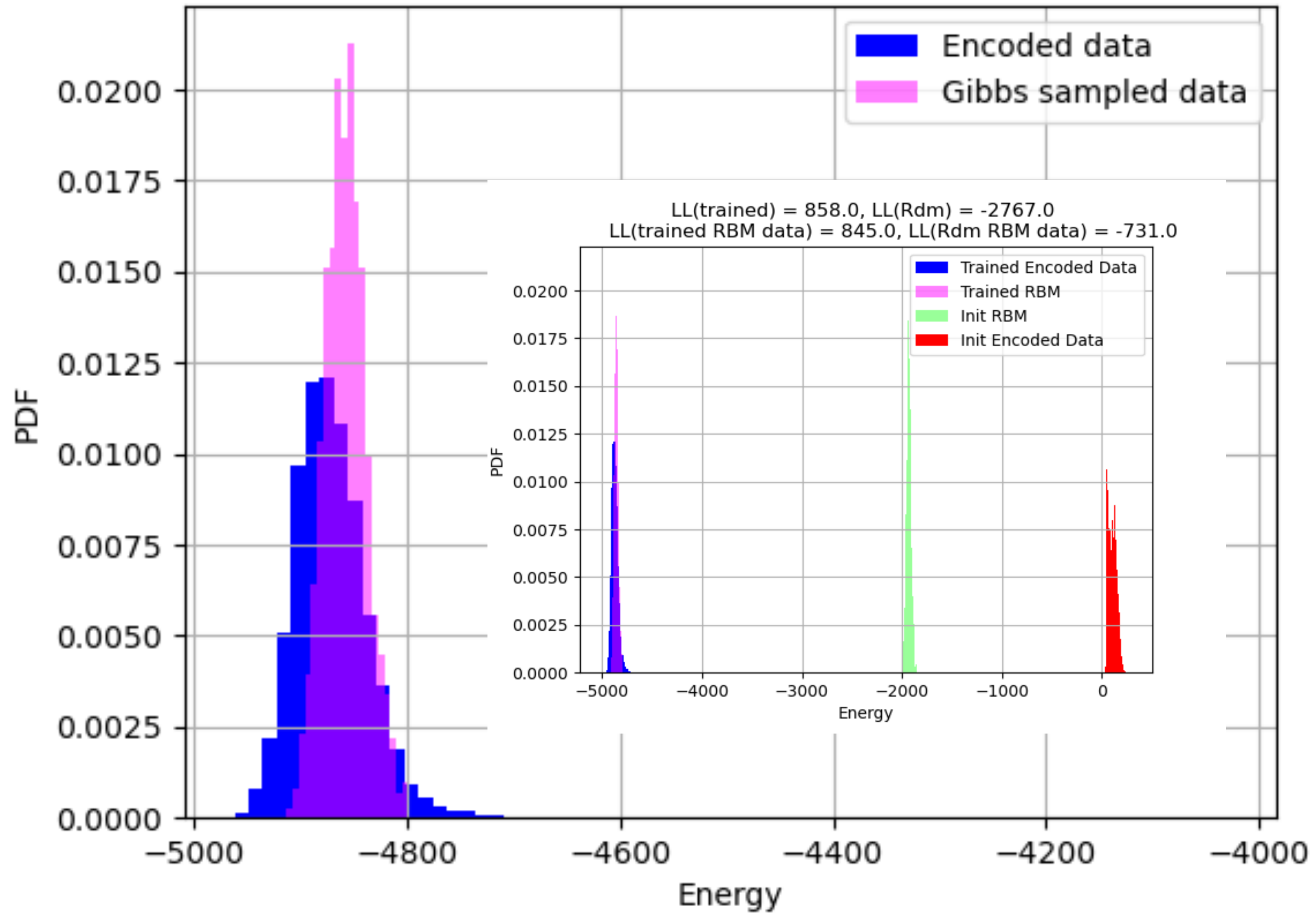
LL(Trained) = Trained model Log-likelihood evaluated on the encoded validation dataset.
LL(Trained RBM data) = Trained model Log-likelihood evaluated on Gibbs sampled data.
LL(Rdm) = Random RBM model Log-likelihood evaluated on the encoded validation dataset.
LL(Rdm RBM data) = Random RBM model Log-likelihood evaluated on on Gibbs sampled data.
We expect LL(trained) ~ LL(trained RBM data) > LL(Rdm RBM data) > LL(Rdm)

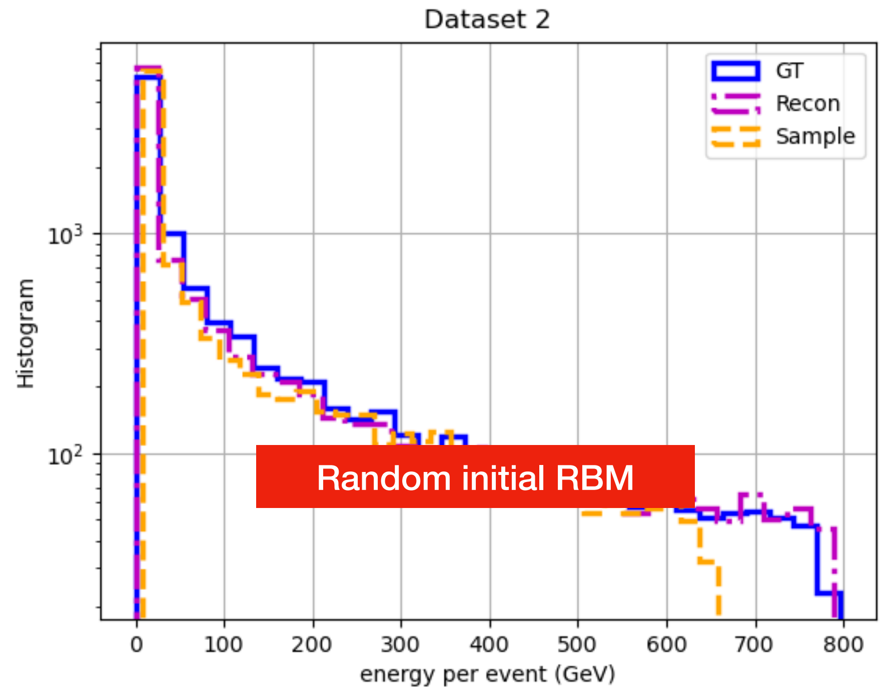


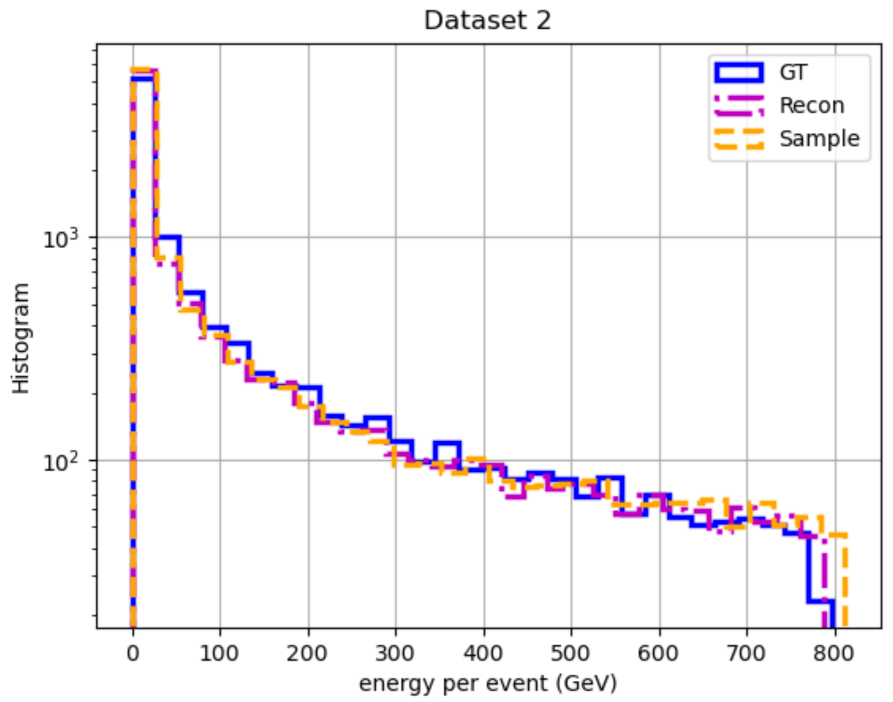
confused-sponge-256

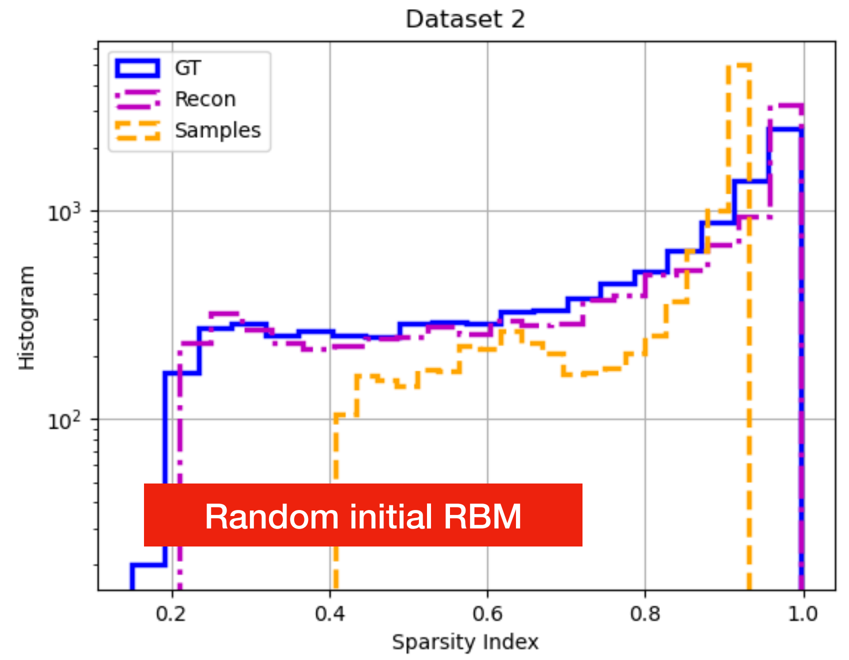
CNN+posEnc cond VAE+scaled data

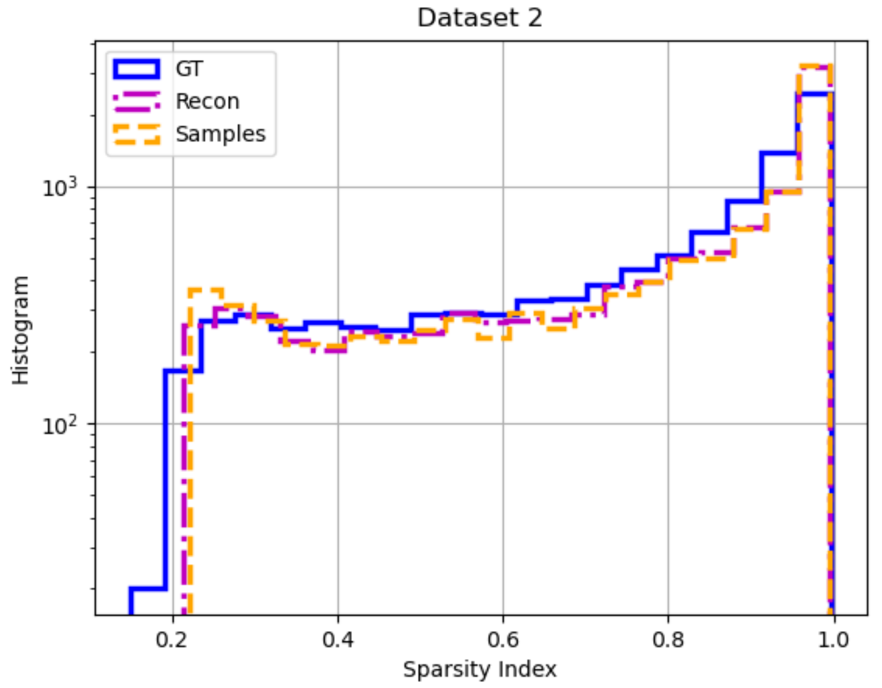


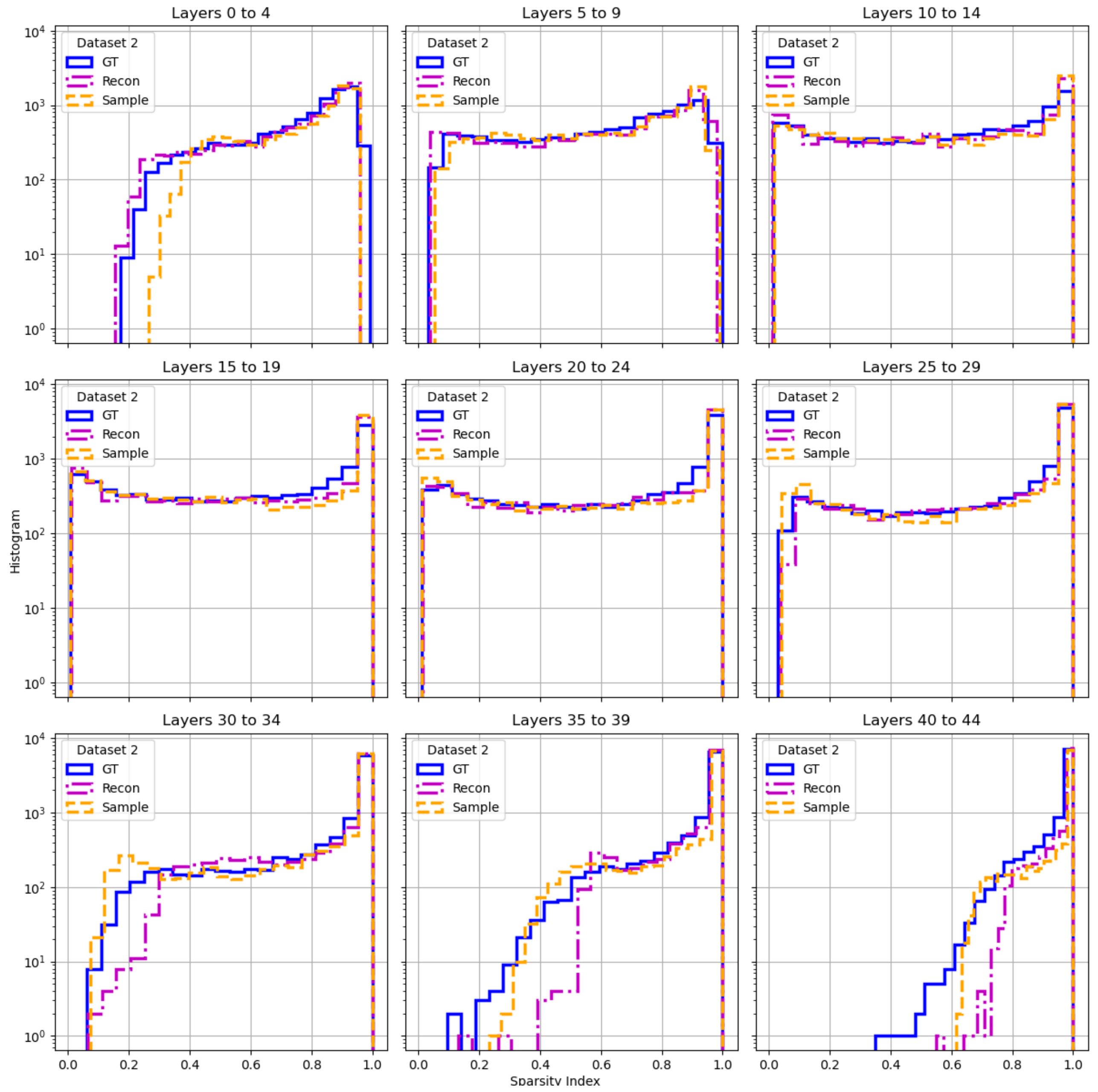






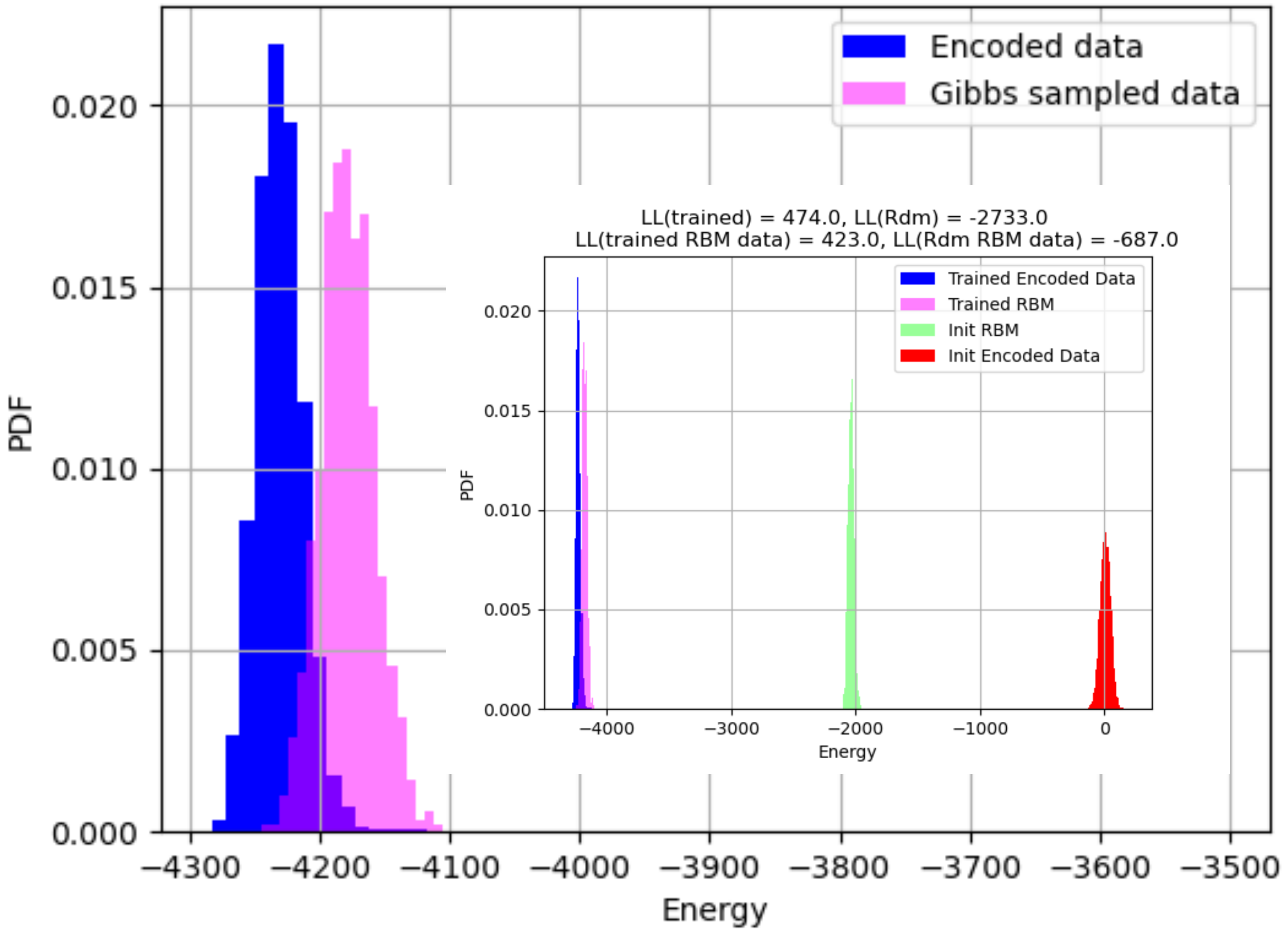


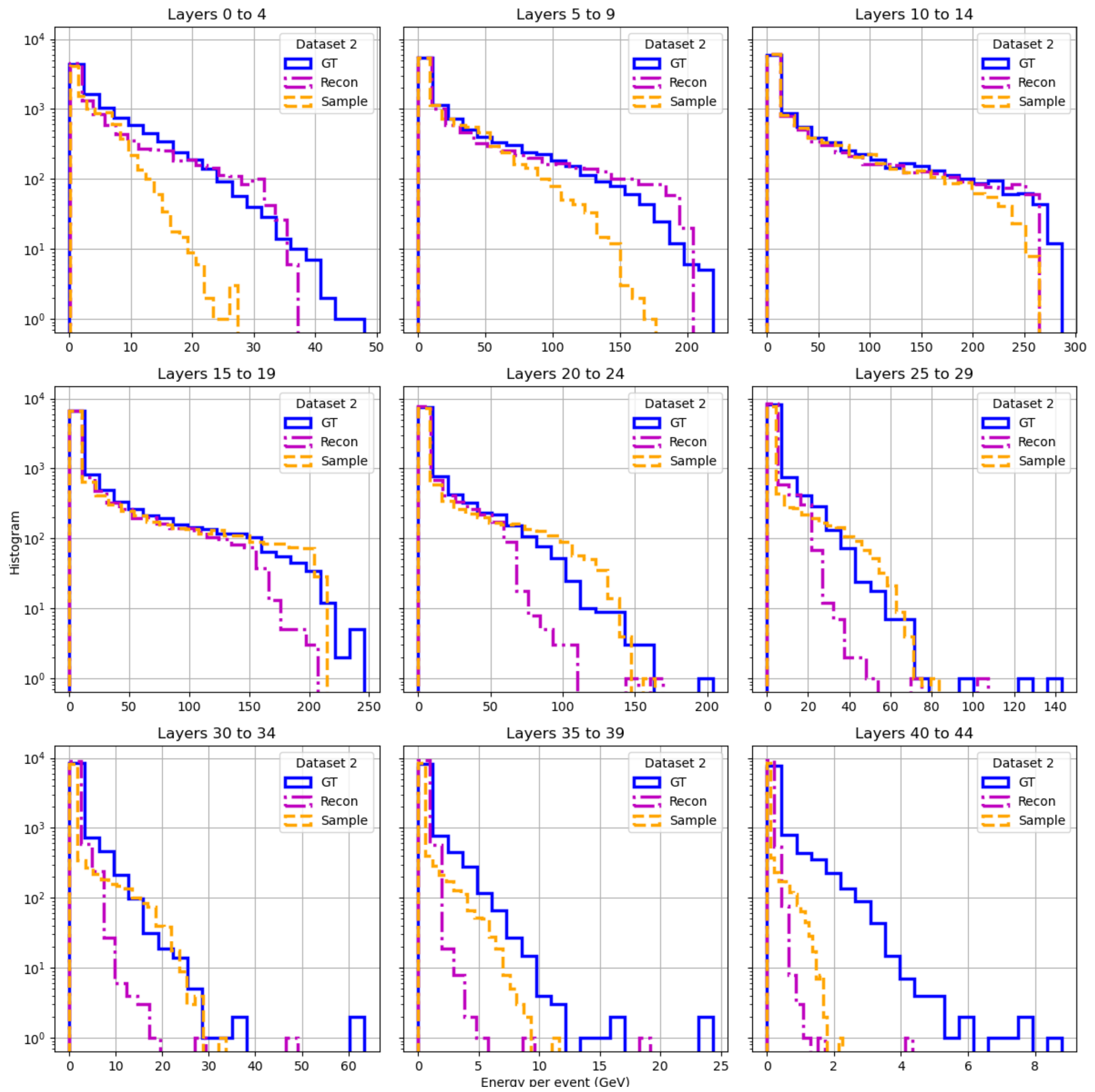




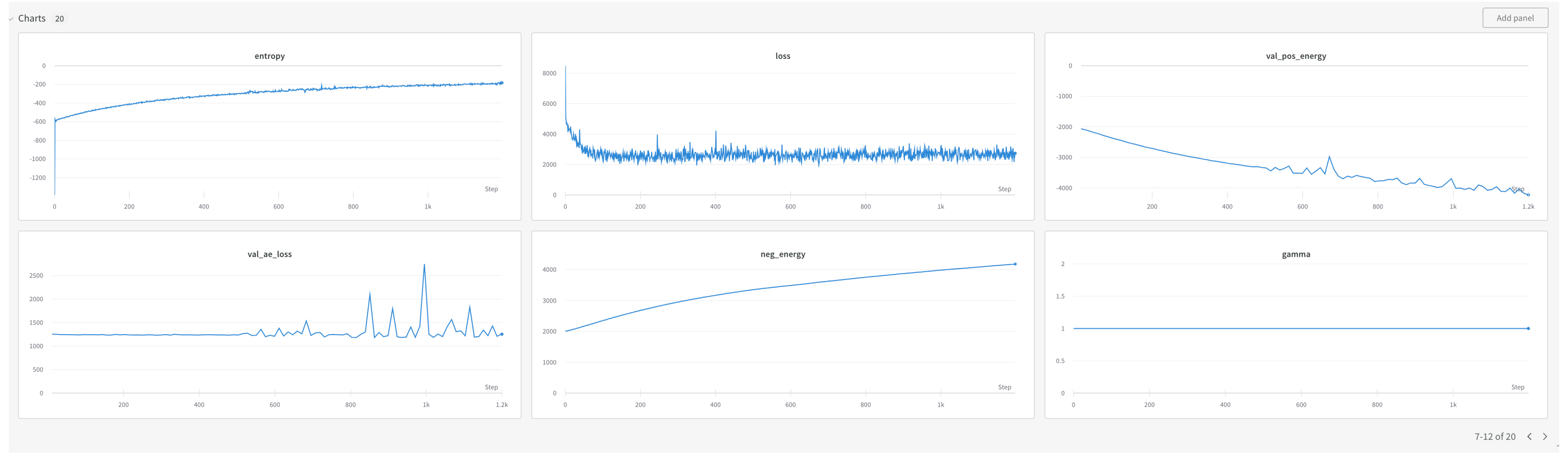
zany-cloud-260

CNN+ cond VAE+posEnc on voxels+scaled data

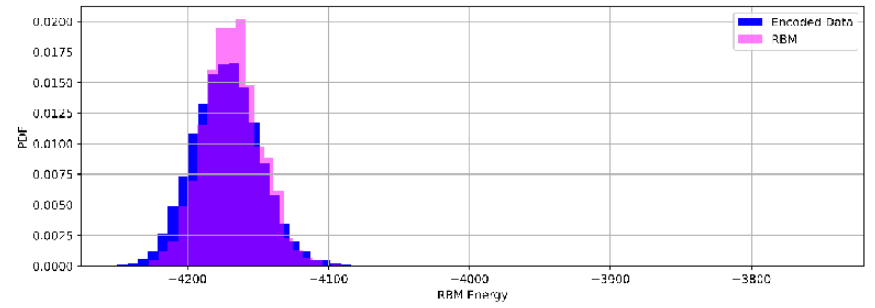


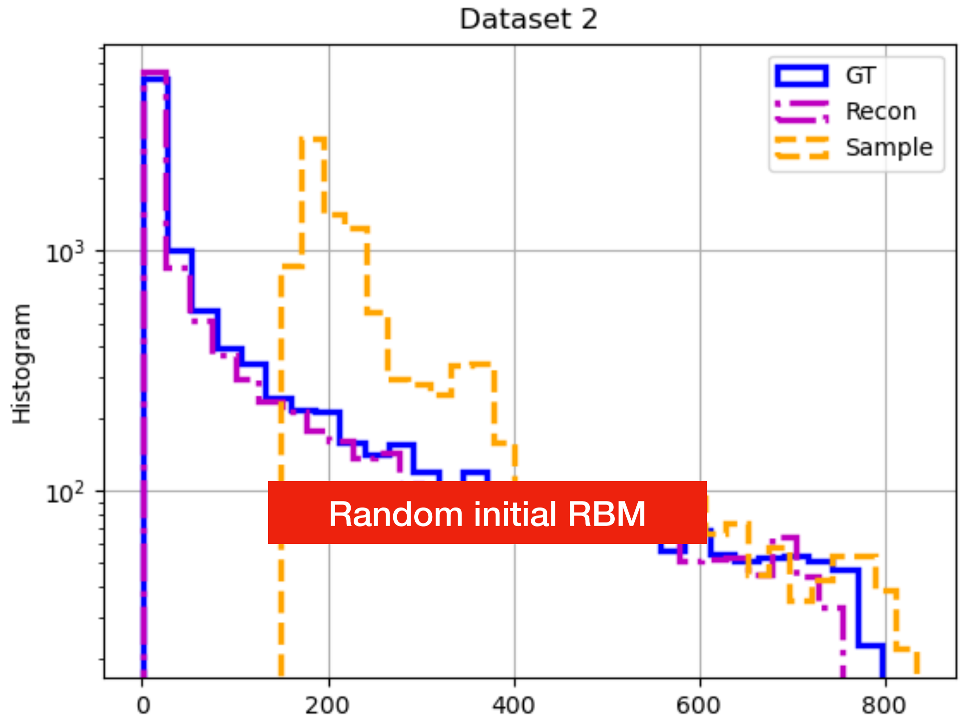


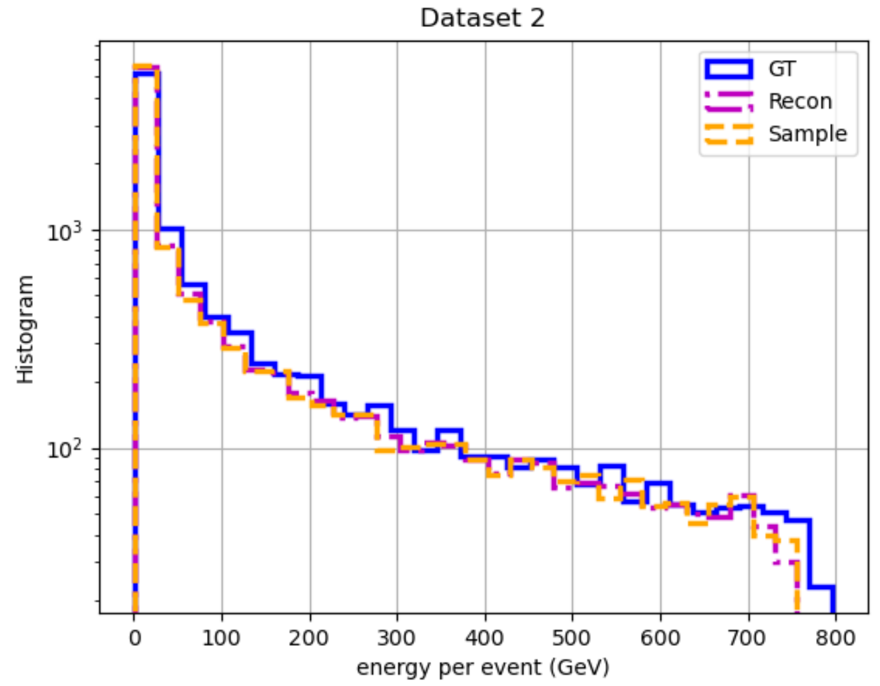
Somewhat unstable model

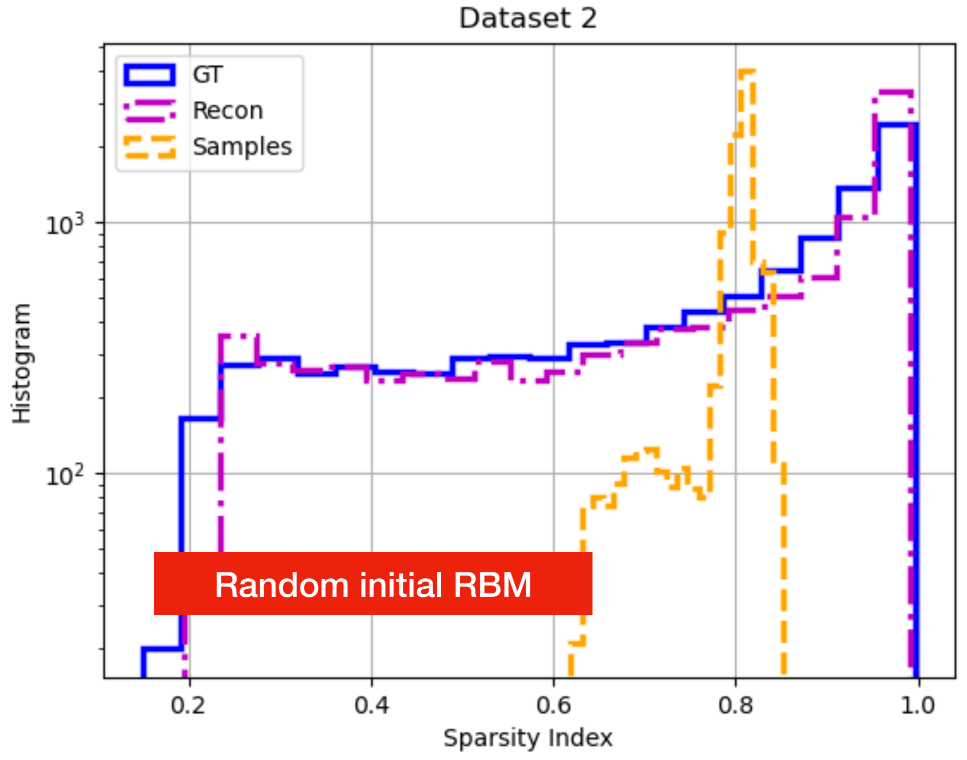


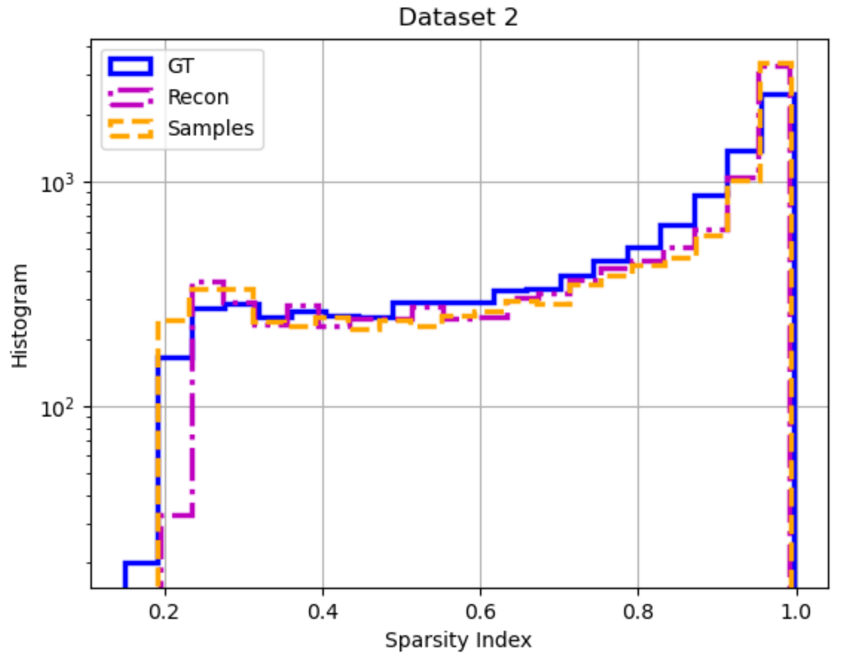
RBM energy

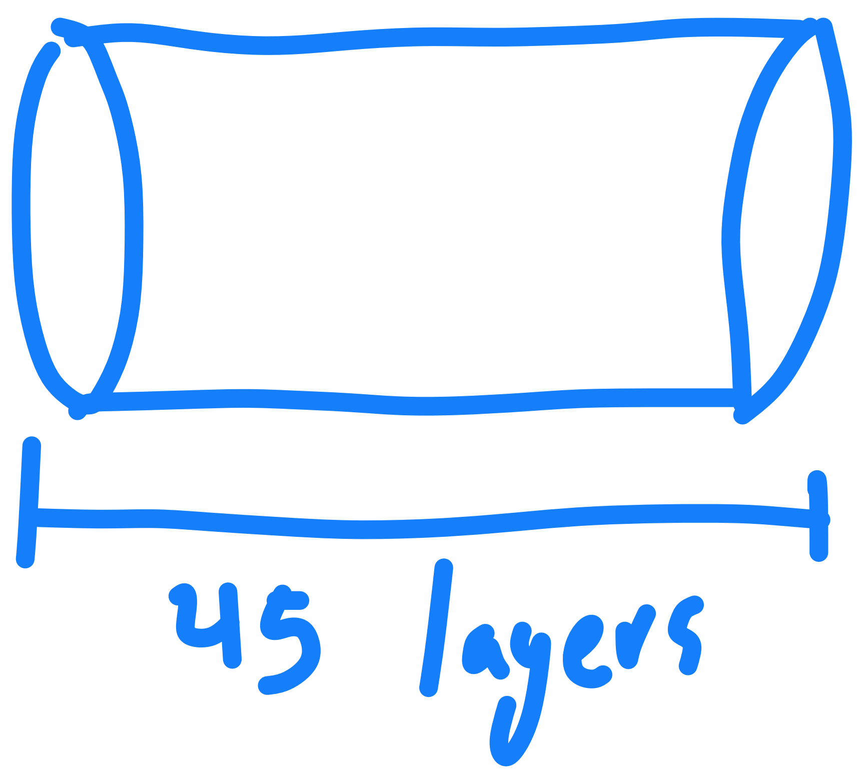


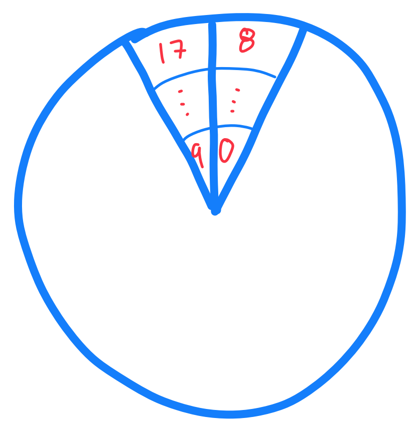










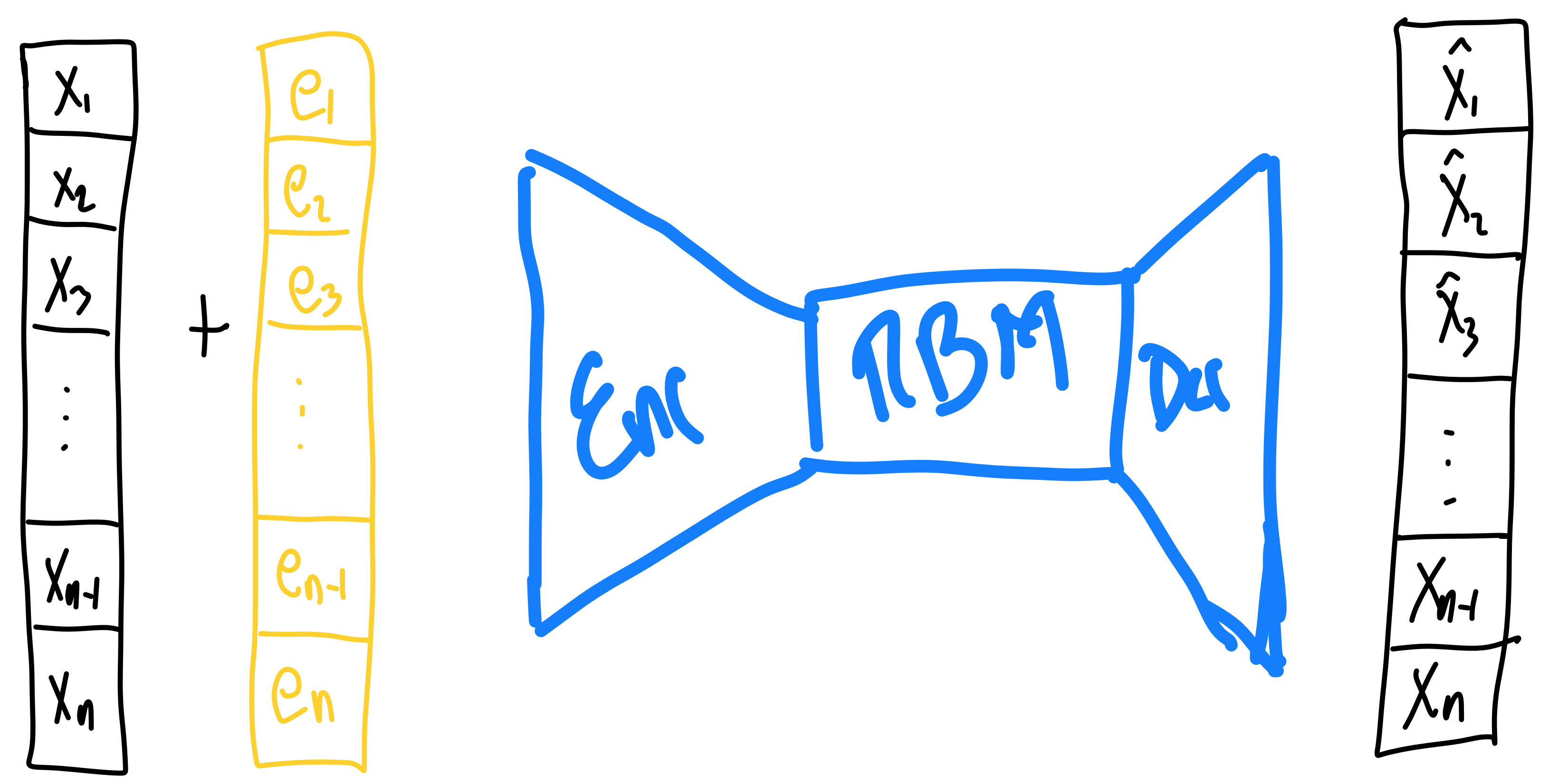


	0	1	…	8	9	…	6480
z	0	0	…	0	0		44
r	0	1	…	8	0		8
φ	0	0	…	0	1		15

$$\Downarrow$$

$$\boxed{i} \longrightarrow \begin{cases} z = i \,//\, 144 \\ \varphi = (i - 144\cdot z)\,//\,9 \\ r = (i - 144\cdot z)\,\%\,9 \end{cases}$$

Positional encoding



How to obtain e_u?

$$PE(u, i) = \begin{cases} \sin k_i u & \text{if } i \text{ is even} \\ \cos k_i u & \text{if } i \text{ is odd} \end{cases} \quad \text{w/ } i = 0, \dots, d$$

$$k_i = 10^{-4i/d}$$

$$\Downarrow$$

$$\lambda_i \in [2\pi, 2\pi \cdot 10^4)$$

$$e_u = \sum_i PE(u, i)$$

$w \sim k$

$w \in [w_{min}, w_{max}]$

$\sim$ Debye

zany-cloud-260

CNN+ cond
VAE+posEnc on
voxels+scaled data

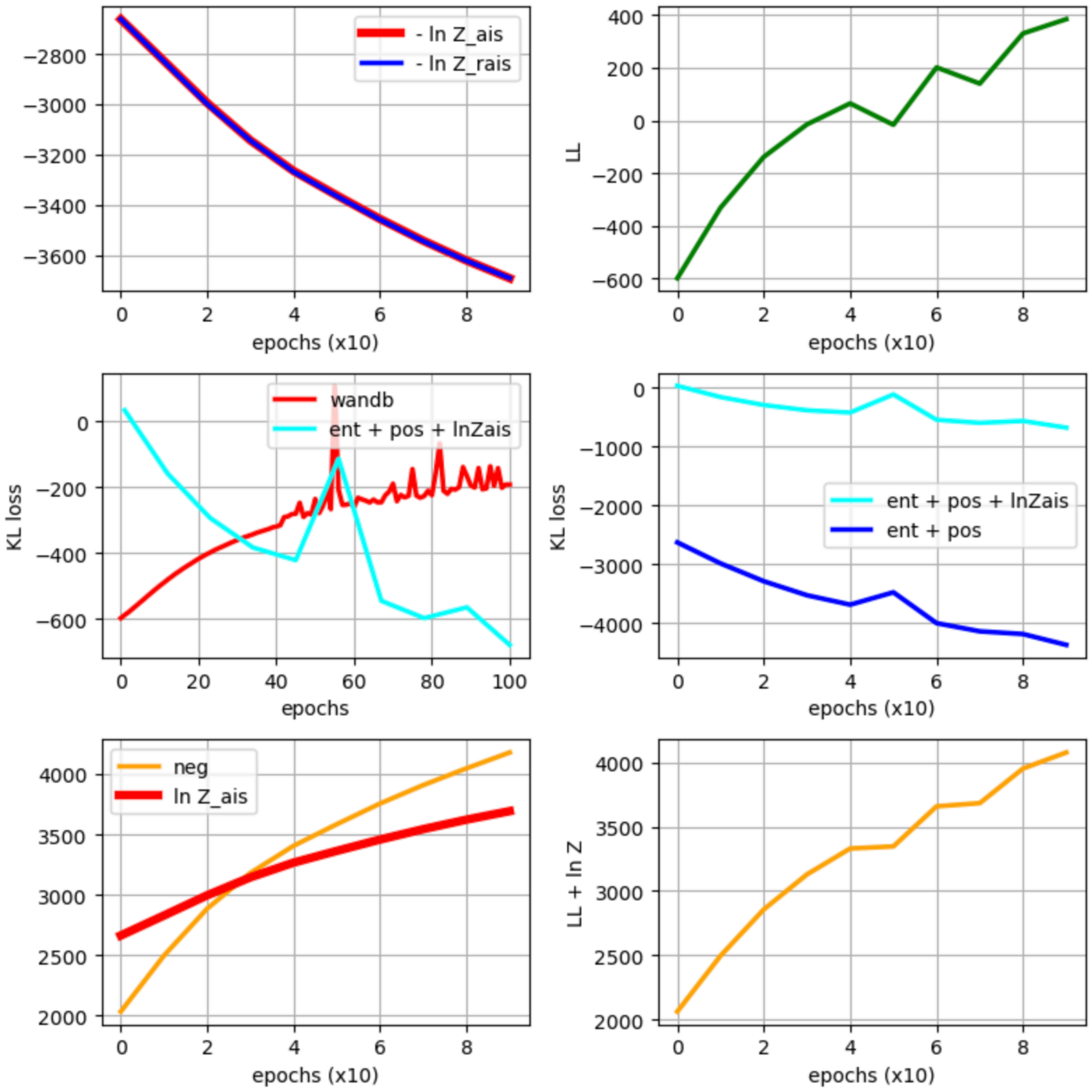
$\{z_i\}_{i=1}^N$: validation dataset

L: $p(\{z_i\}) = \prod_{i=1}^N p(z_i) = \frac{1}{Z^N} \prod_{i=1}^N e^{-E(z_i)}$

Z: partition function

$$LL = \langle \ln p(\{z_i\}) \rangle = \frac{1}{N}\left[-\sum_{i=1}^N E(z_i) - N \ln Z \right]$$

$$= -\langle E(z) \rangle_{\{z_i\}} - \ln Z$$



confused-
sponge-256

CNN+posEnc cond
VAE+scaled data

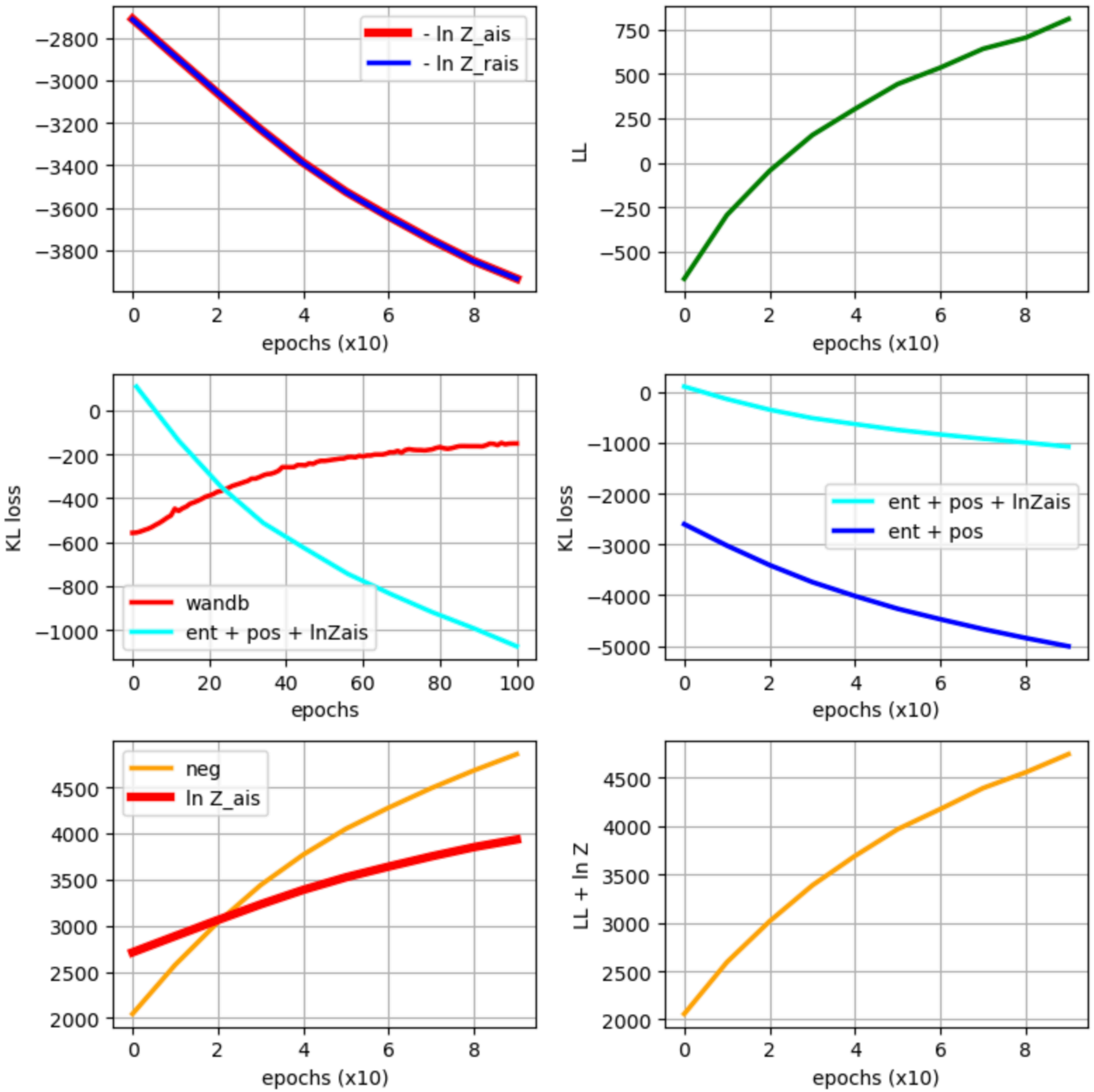
$\{z_i\}_{i=1}^{N}$: validation dataset

L: $p(\{z_i\}) = \prod_{i=1}^{N} p(z_i) = \frac{1}{Z^N} \prod_{i=1}^{N} e^{-E(z_i)}$

Z: partition function

$$LL = \langle \ln p(\{z_i\}) \rangle = \frac{1}{N}\left[-\sum_{i=1}^{N} E(z_i) - N \ln Z\right]$$

$$= -\langle E(z) \rangle_{\{z_i\}} - \ln Z$$



Things to do

- Make code lighter
- More work needed on the Positional Encoding
- I suggest increasing annealing rates.
- Save loss function data
- Save backup model after each epoch.
- What defines our best model during training?