TRIUMF

THE UNIVERSITY OF BRITISH COLUMBIA

DTRC-NRC

UNIVERSITY of VIRGINIA

Calo4pQVAE: Progress and updates

Oct 24 2024

TRIUMF Canada's particle accelerator centre
Centre canadien d'accélération des particules

Summary

✓ Weight decay removal?

✓ Papers status

✓ Using the QPU w/ Pegasus and in Zephyr

✓ High temperature gradient approximation for trained RBMs

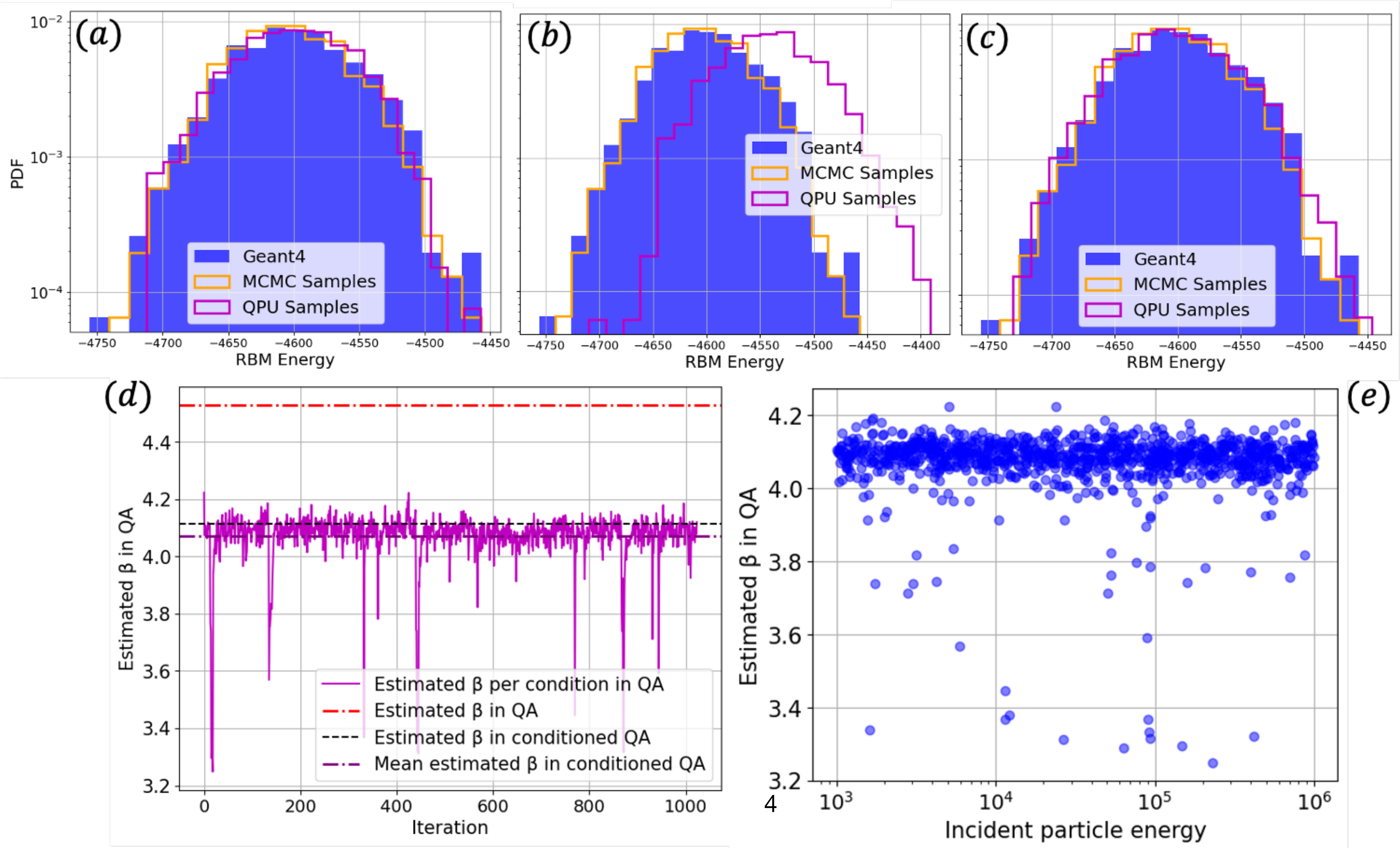
✦ RBM and Diffusion model equivalence.

✦ Relaxation time in RBMs

Papers status

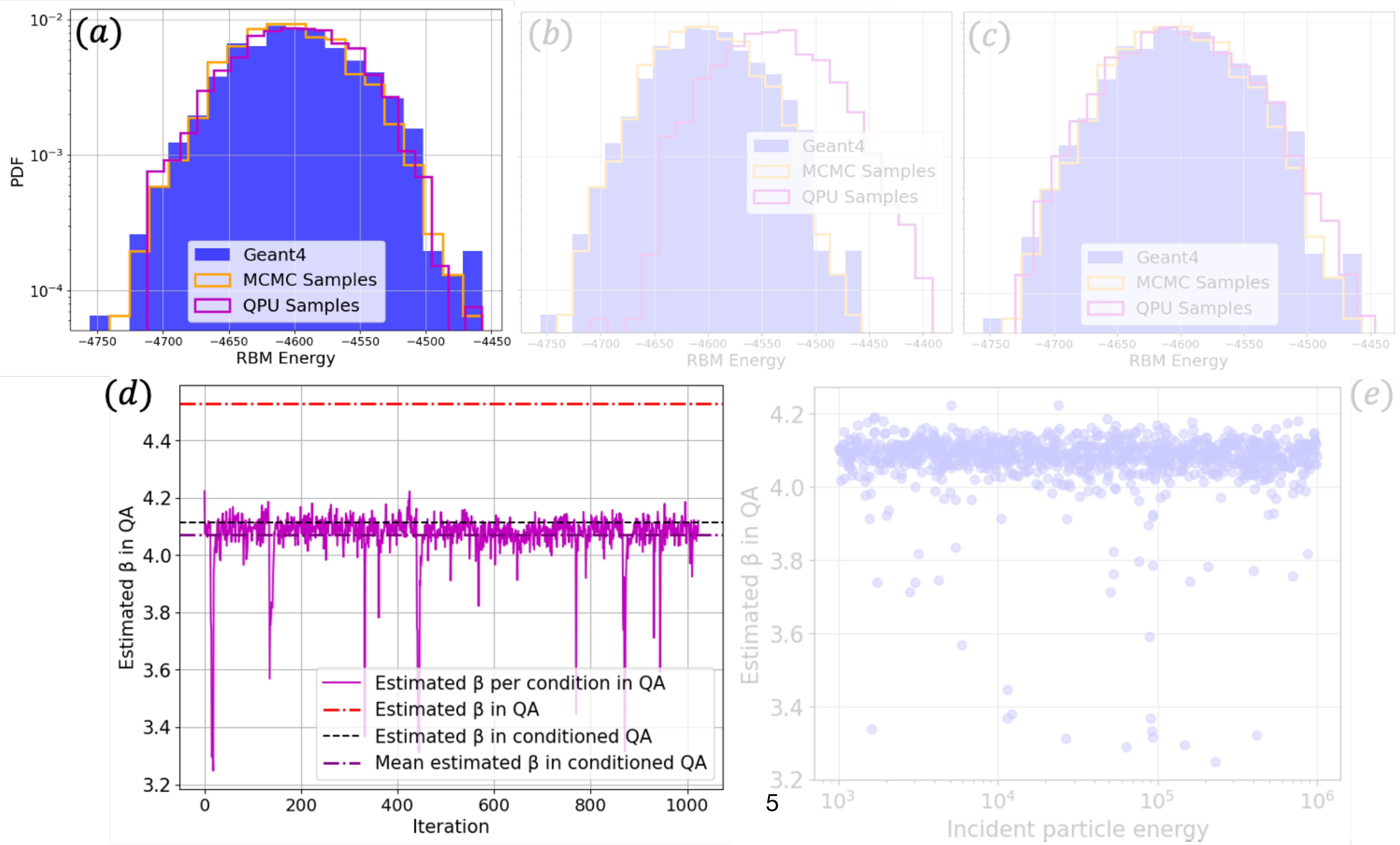
- EPJC -> Rebuttal submitted?
- iEEE QCE Conf -> haven't seen the proceedings online yet
- Neurips ML4Phys -> Got accepted!
- PRX draft -> on countdown for submission

Using QPU w/ Pegasus



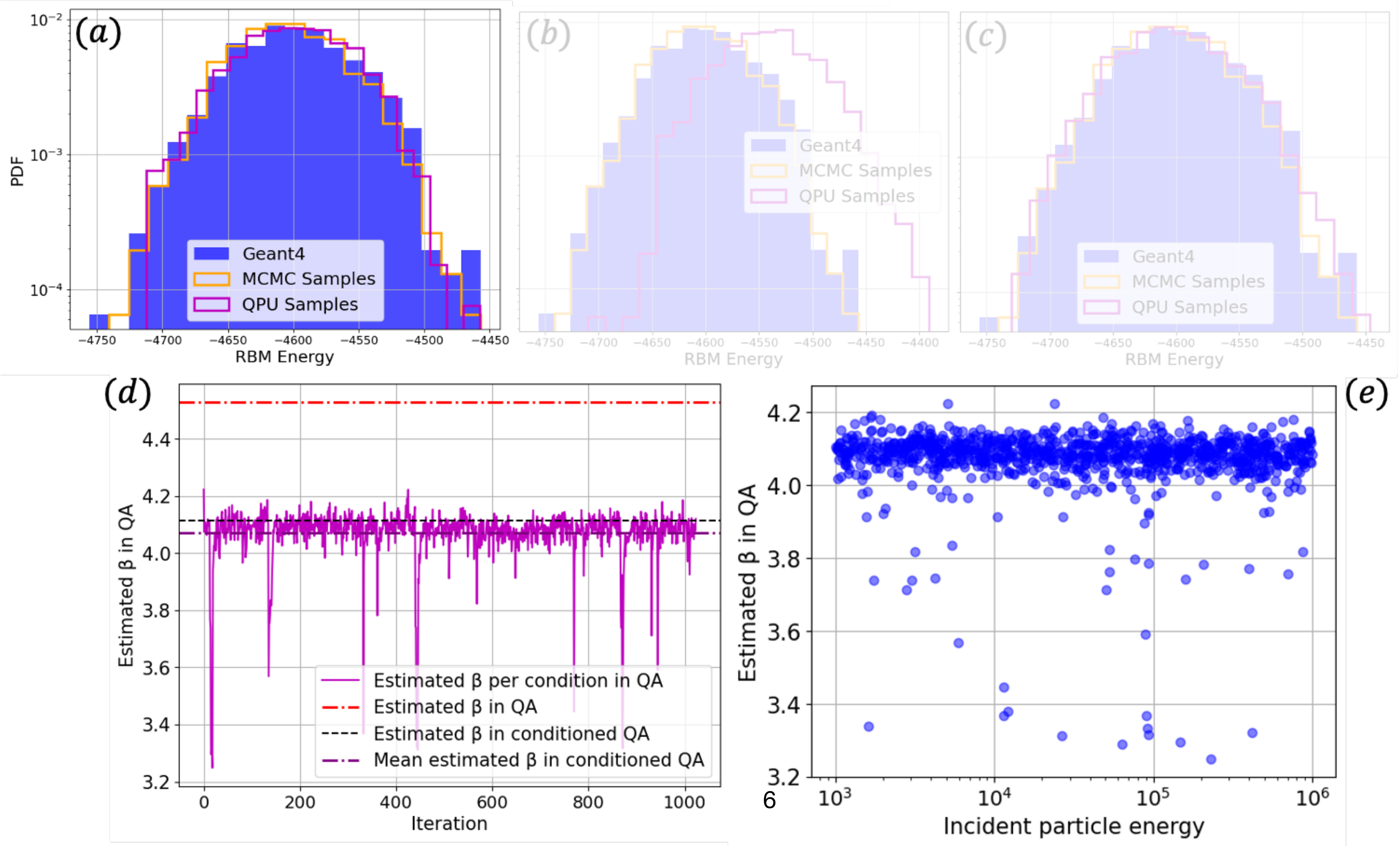
Using QPU w/ Pegasus

We estimate the QA inverse Temperature before generating each sample.



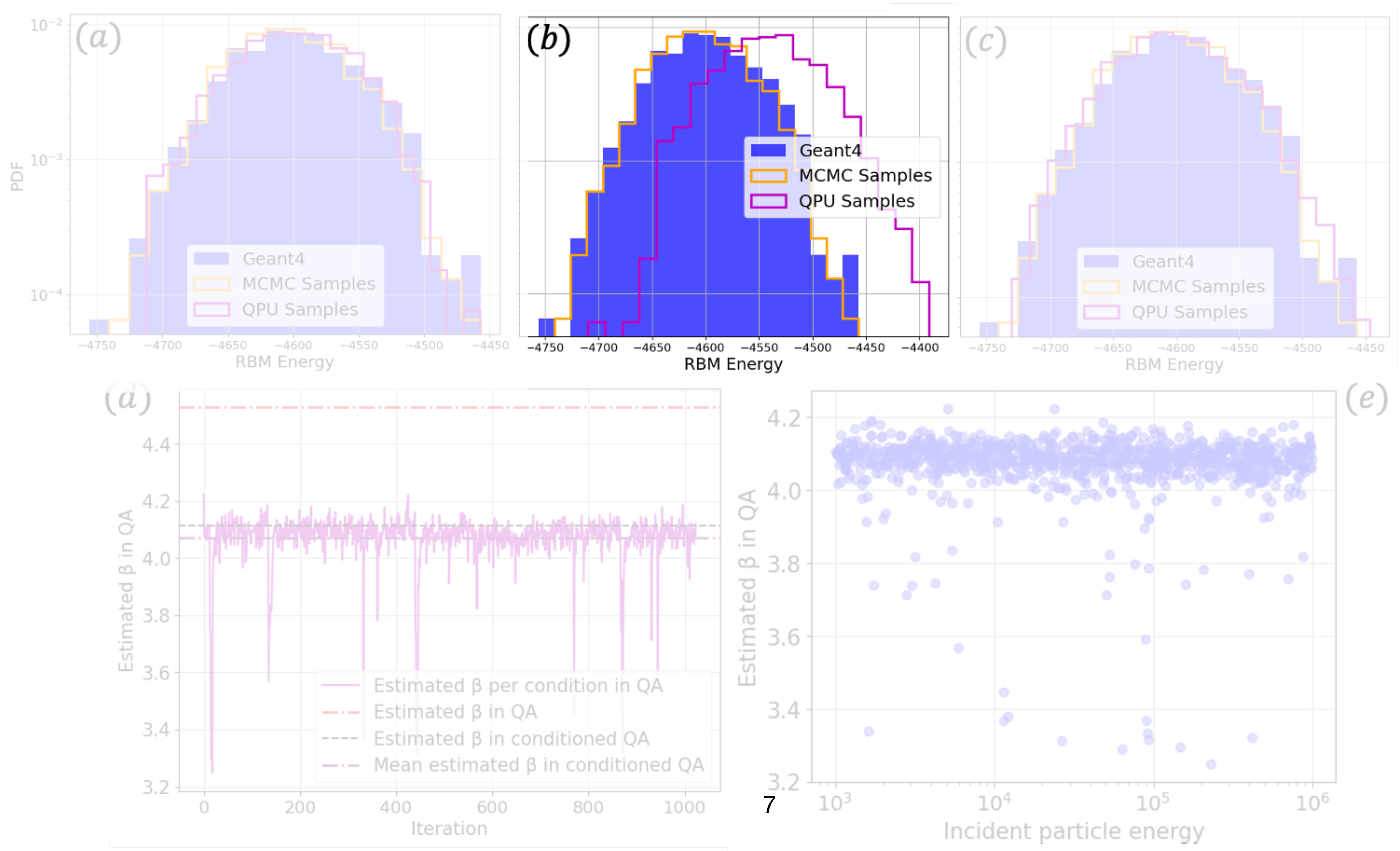
Using QPU w/ Pegasus

We estimate the QA inverse Temperature before generating each sample.



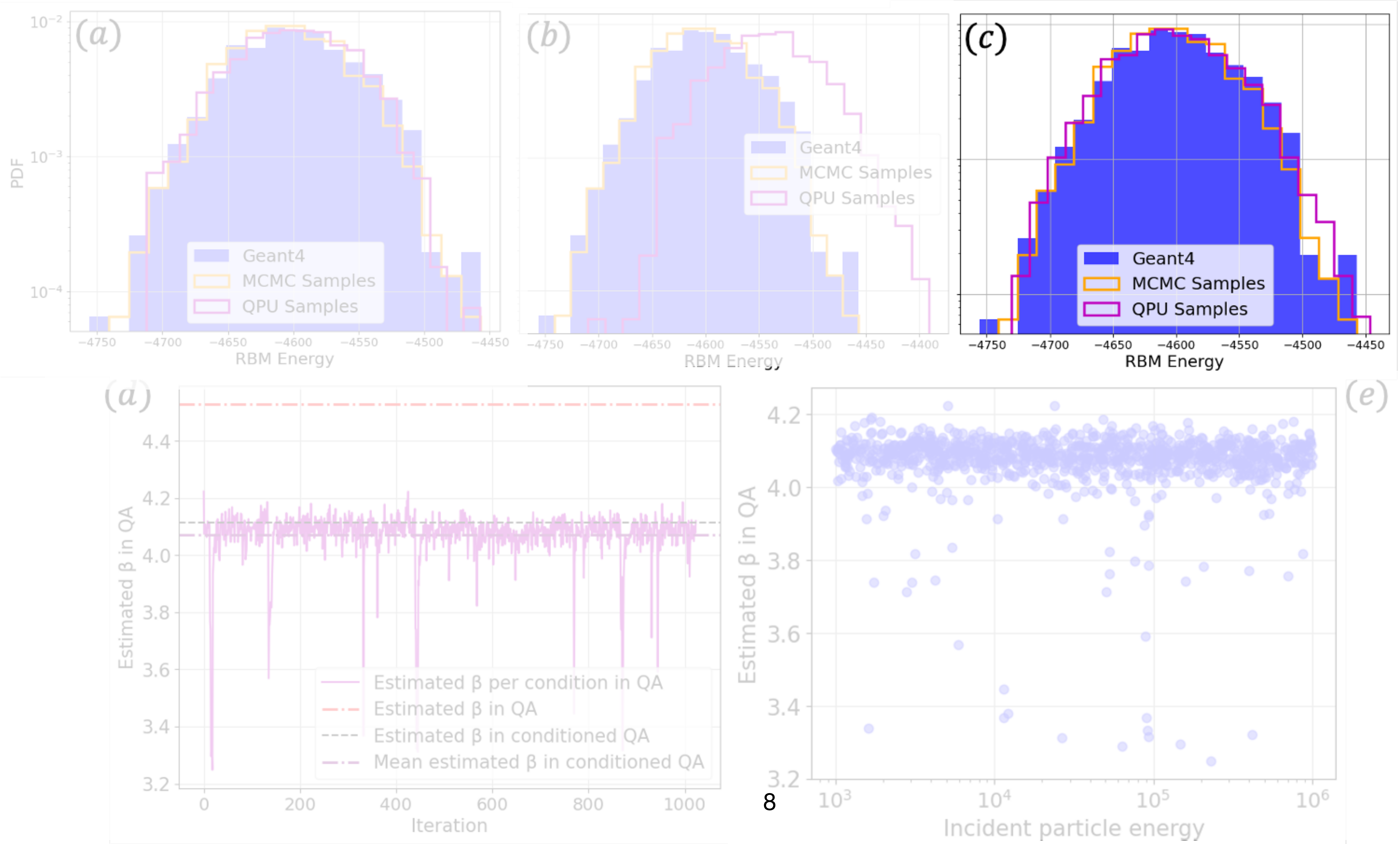
Using QPU w/ Pegasus

We estimate the QA inverse Temperature 1 time. Then generate all samples.



Using QPU w/ Pegasus

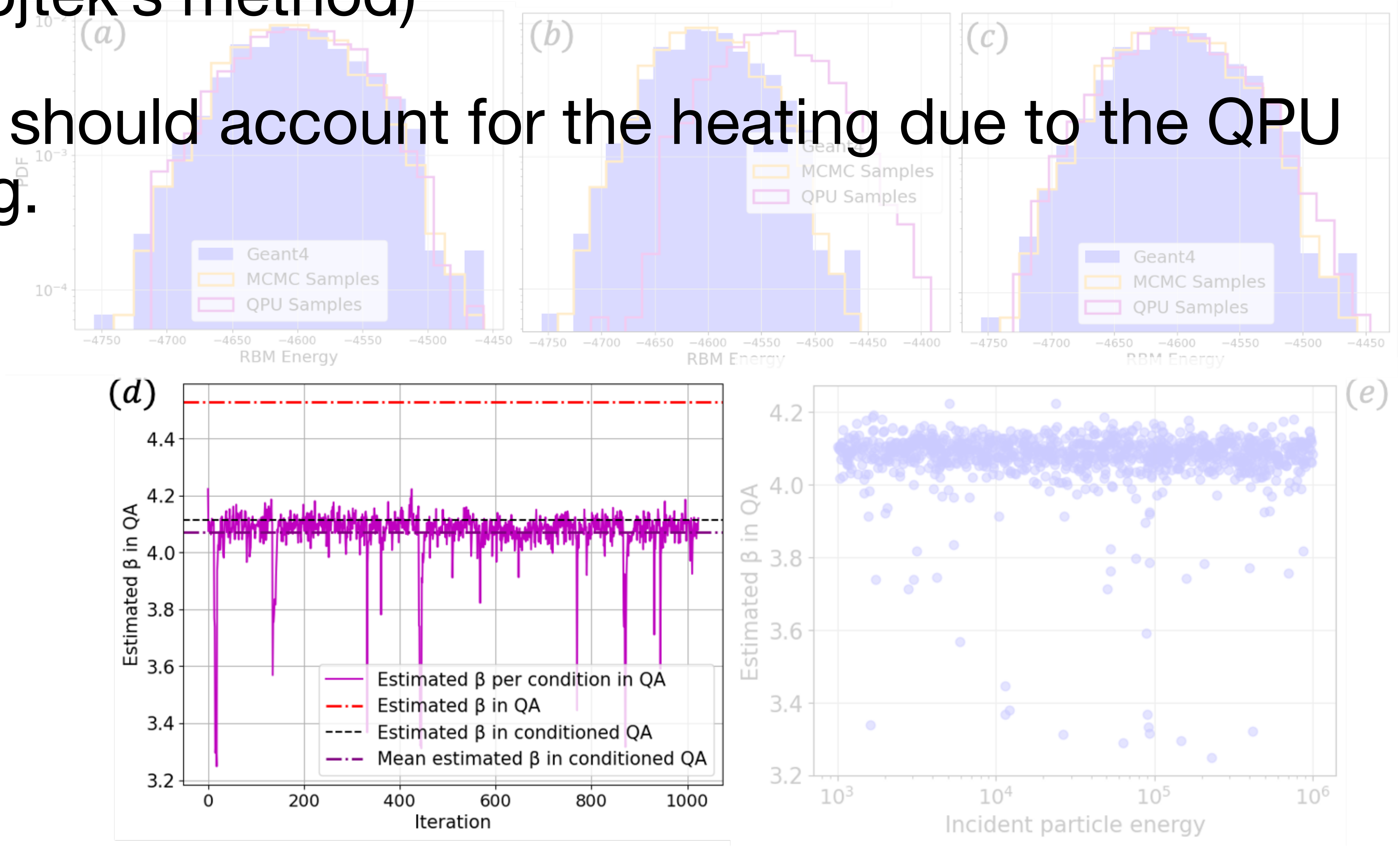
We estimate the QA inverse Temperature 1 time. Then generate all samples. Wait 2.5 seconds between samples



Using QPU w/ Pegasus

What if we estimate the QA inverse by generating 1 sample per API call? (Wojtek's method)

This way we should account for the heating due to the QPU programming.

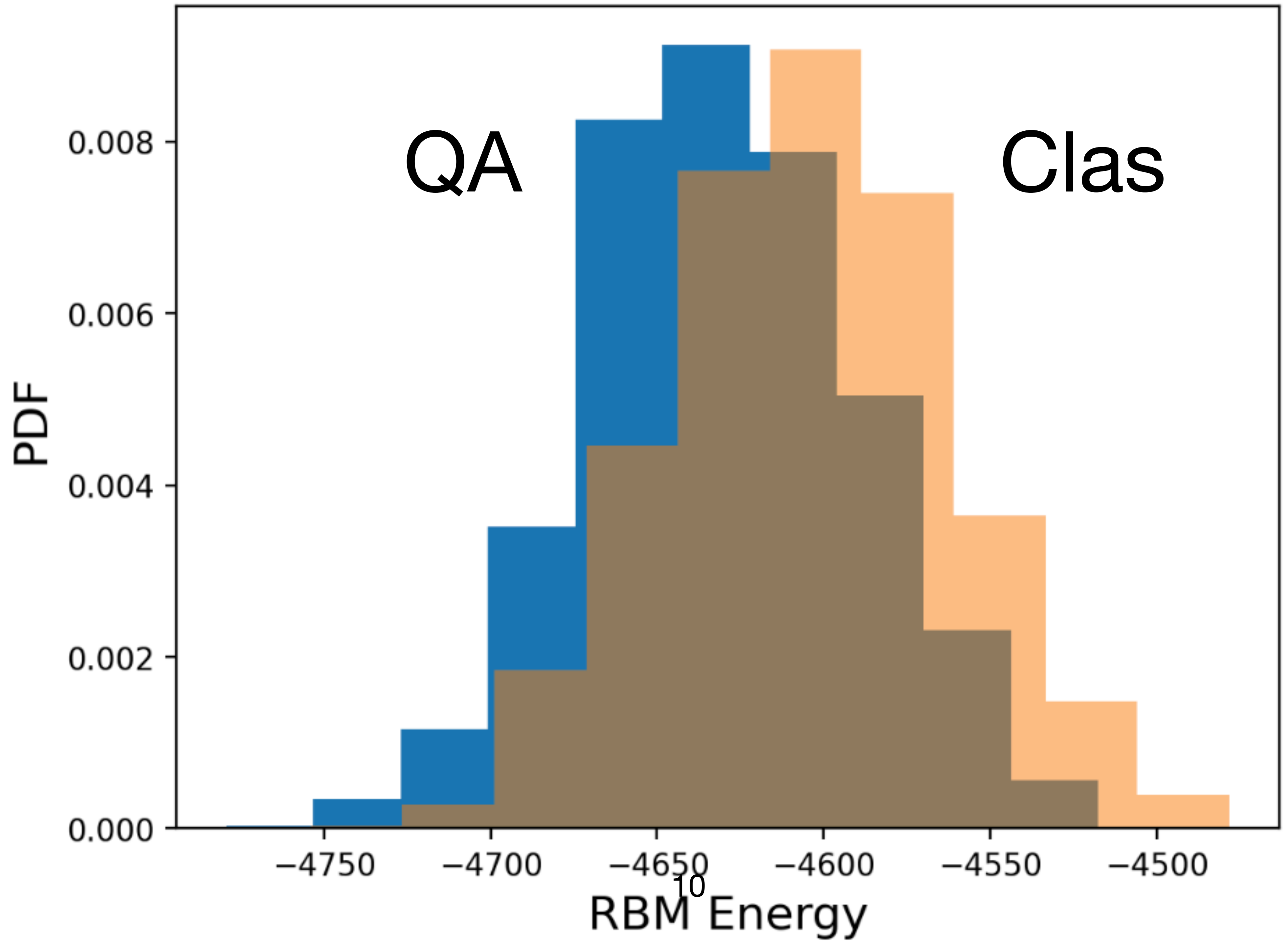


Using QPU w/ Pegasus

What if we estimate the QA inverse by generating 1 sample per API call?

This way we should account for the heating due to the QPU programming.

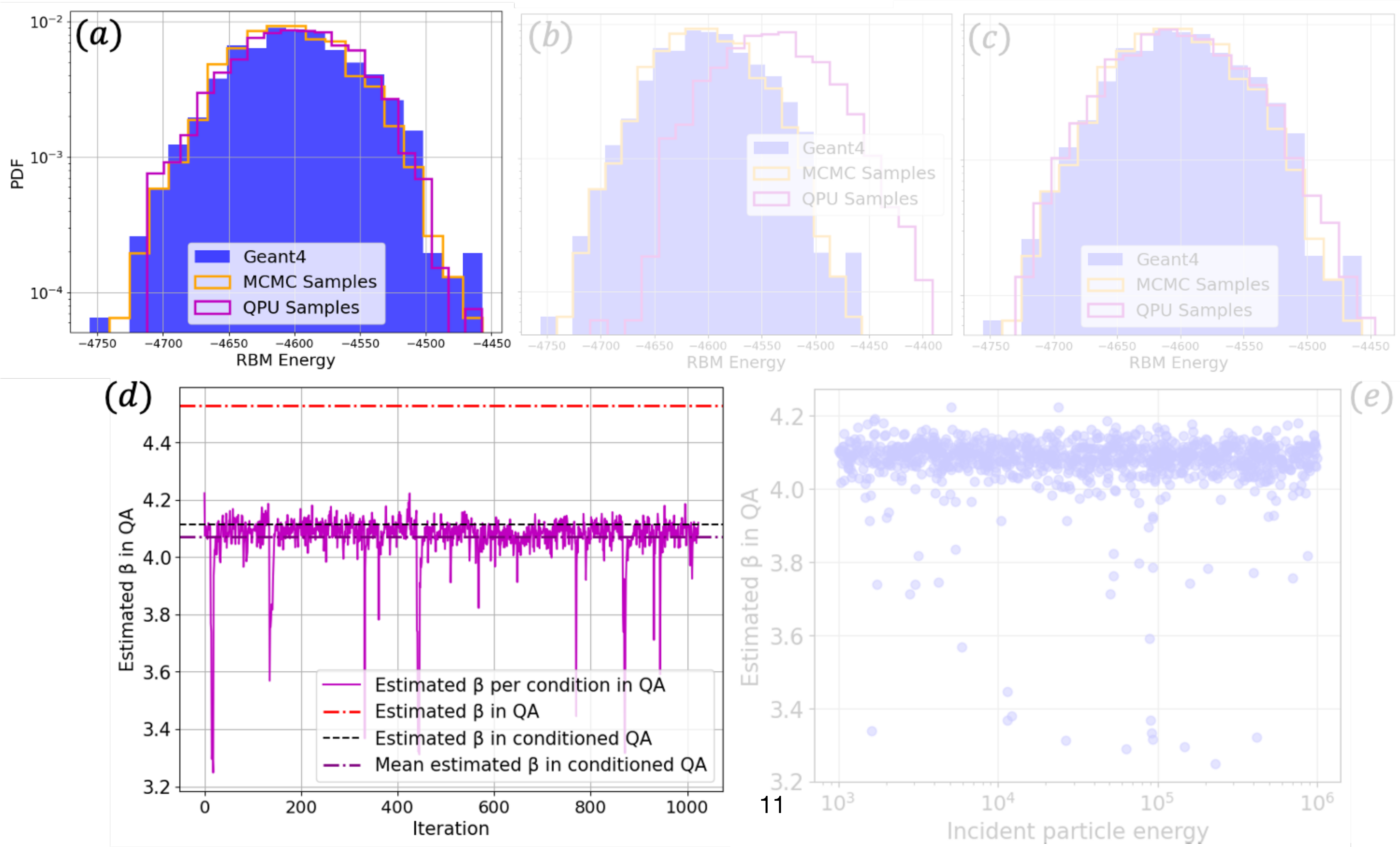
$$\beta_{QA} \approx 2.6$$



Using QPU w/ Pegasus

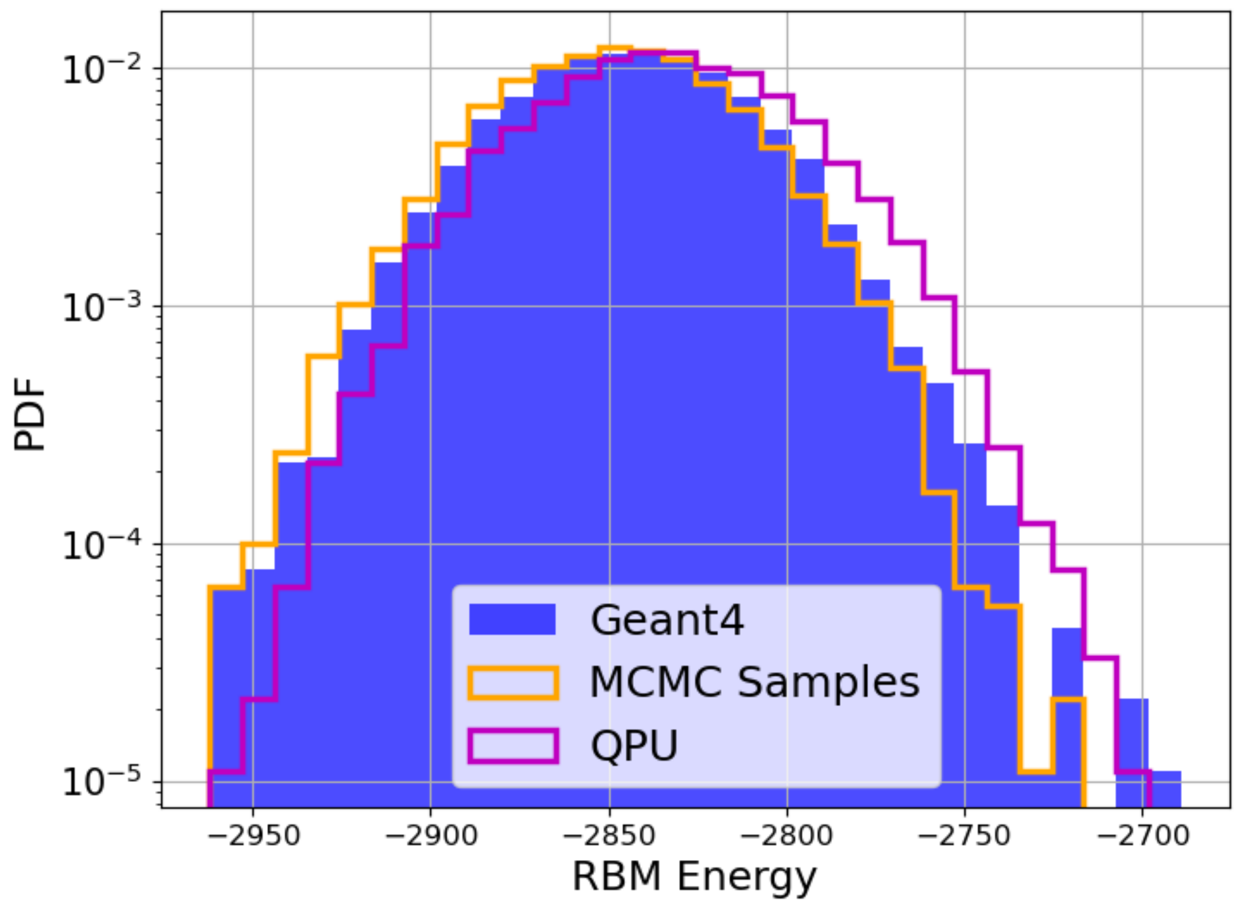
We estimate the QA inverse Temperature before generating each sample.

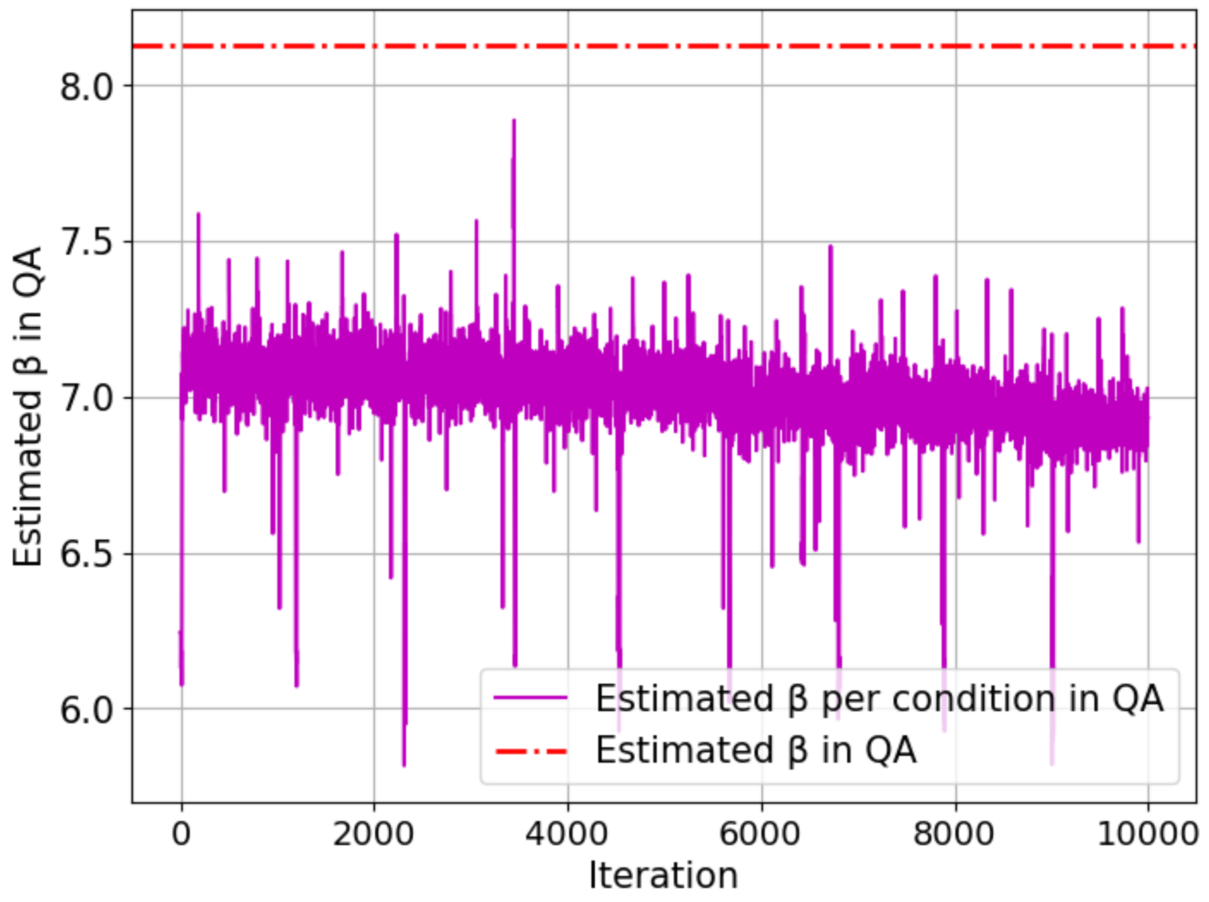
Winning method!

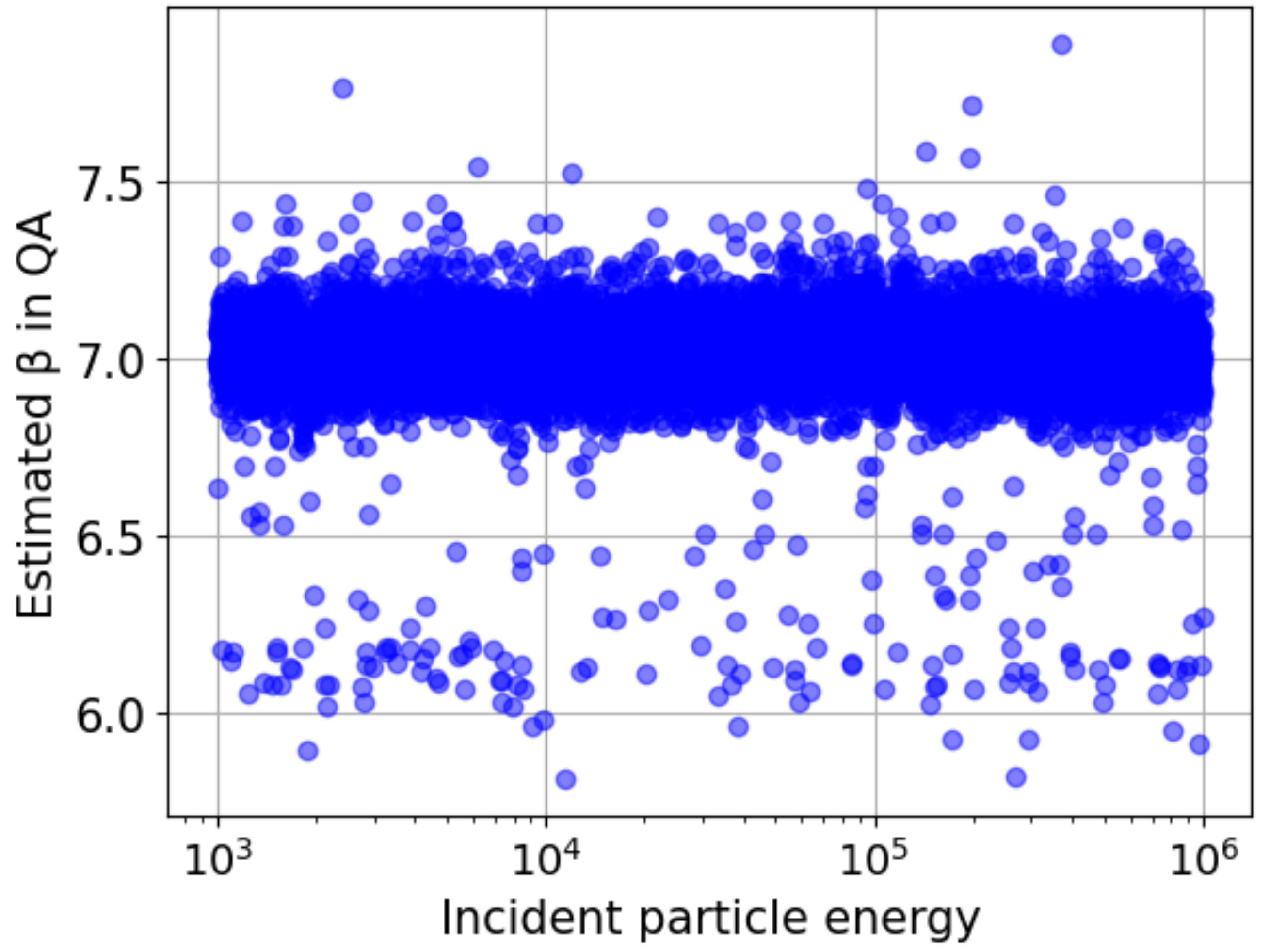


Using QPU w/ Zephyr :: Winning method

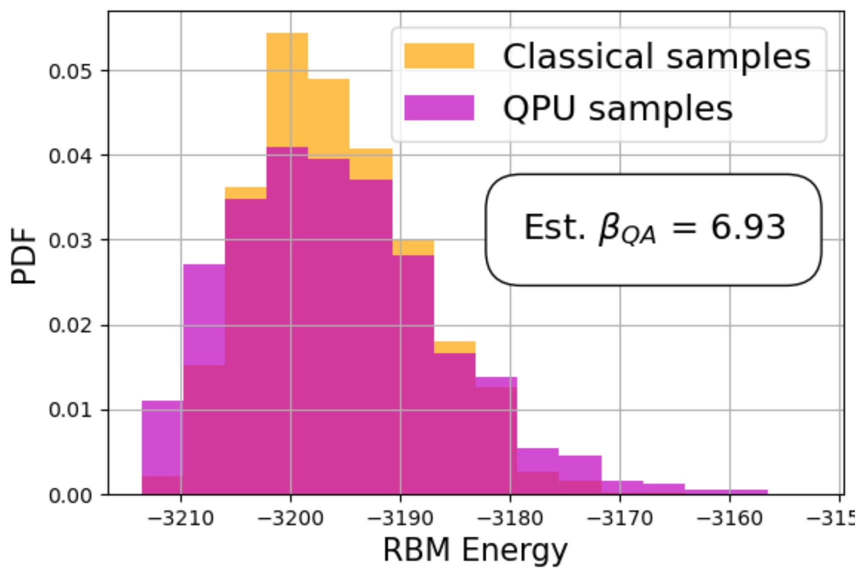
Model A

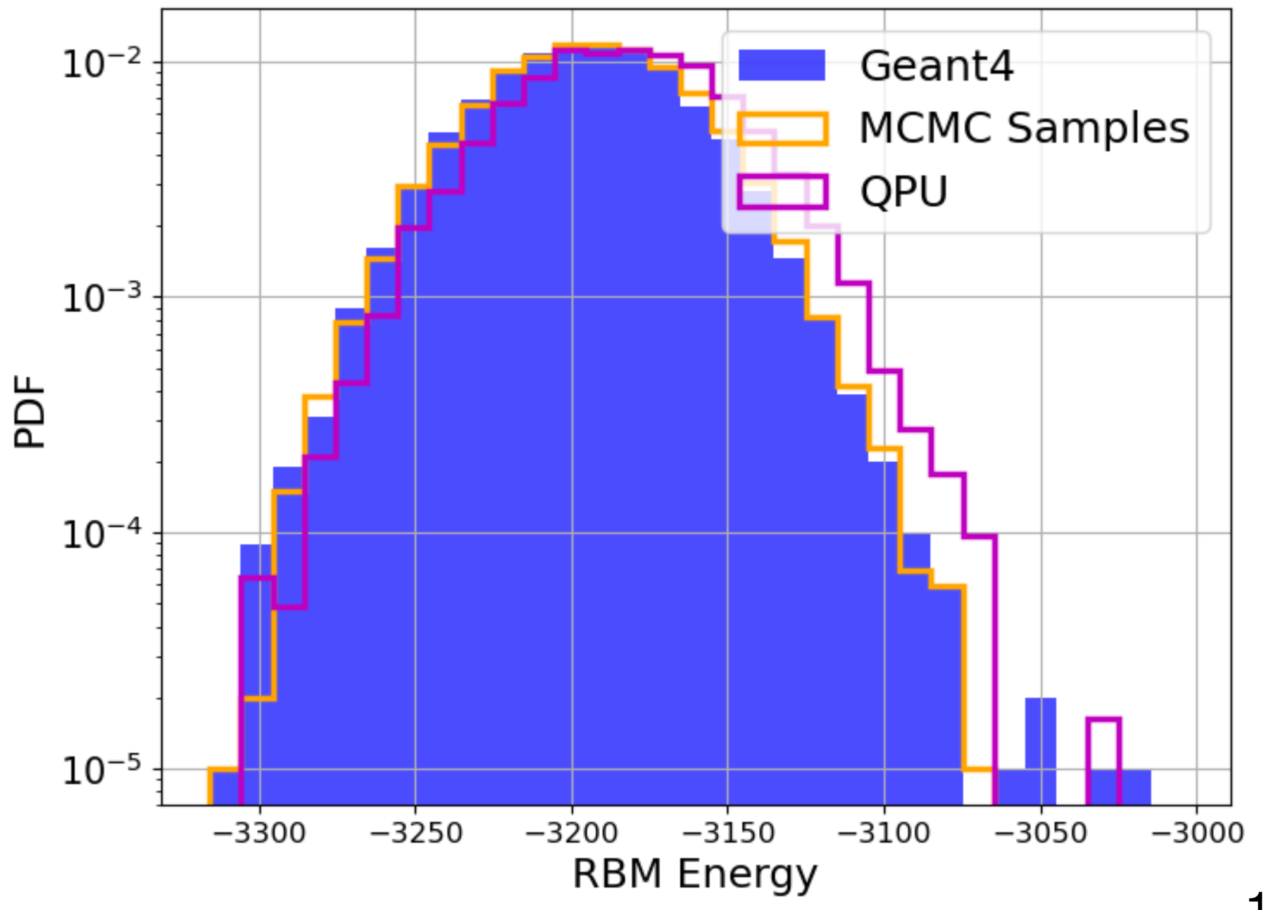


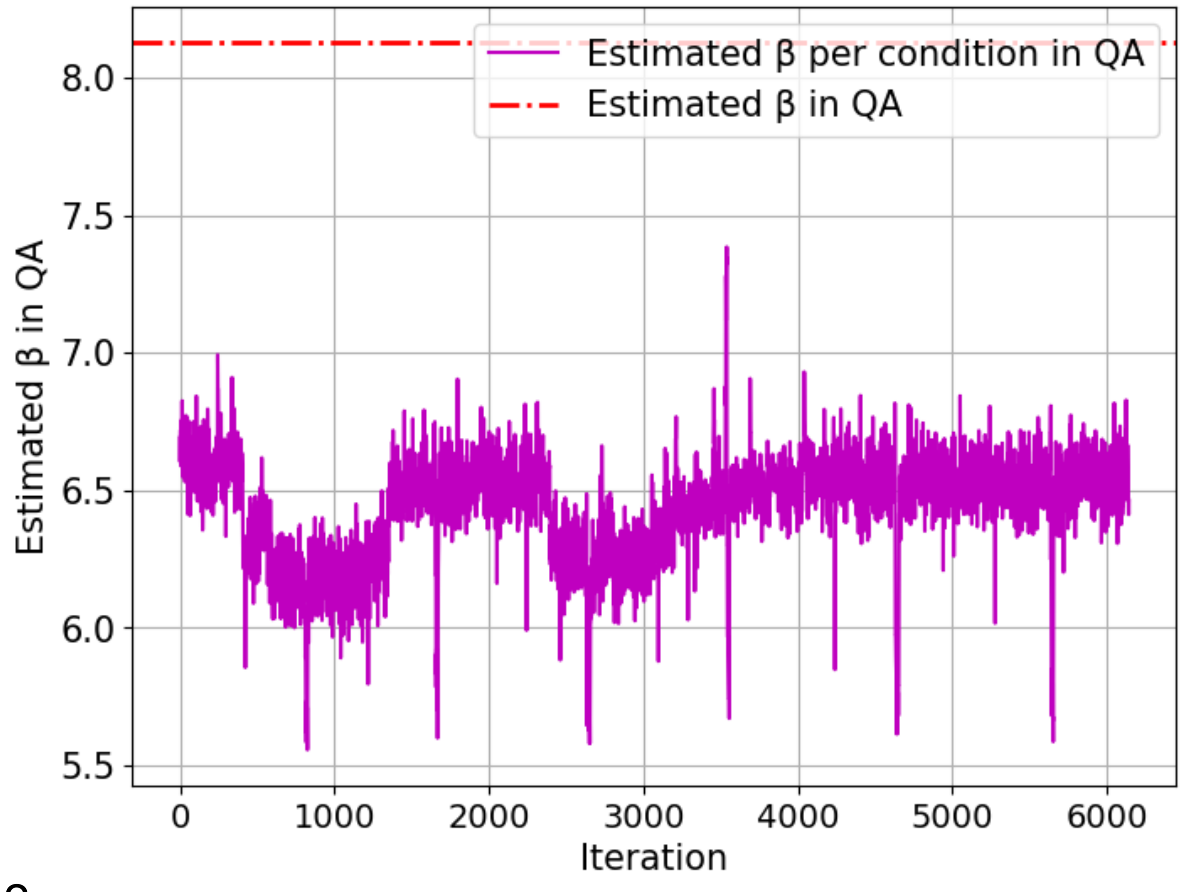


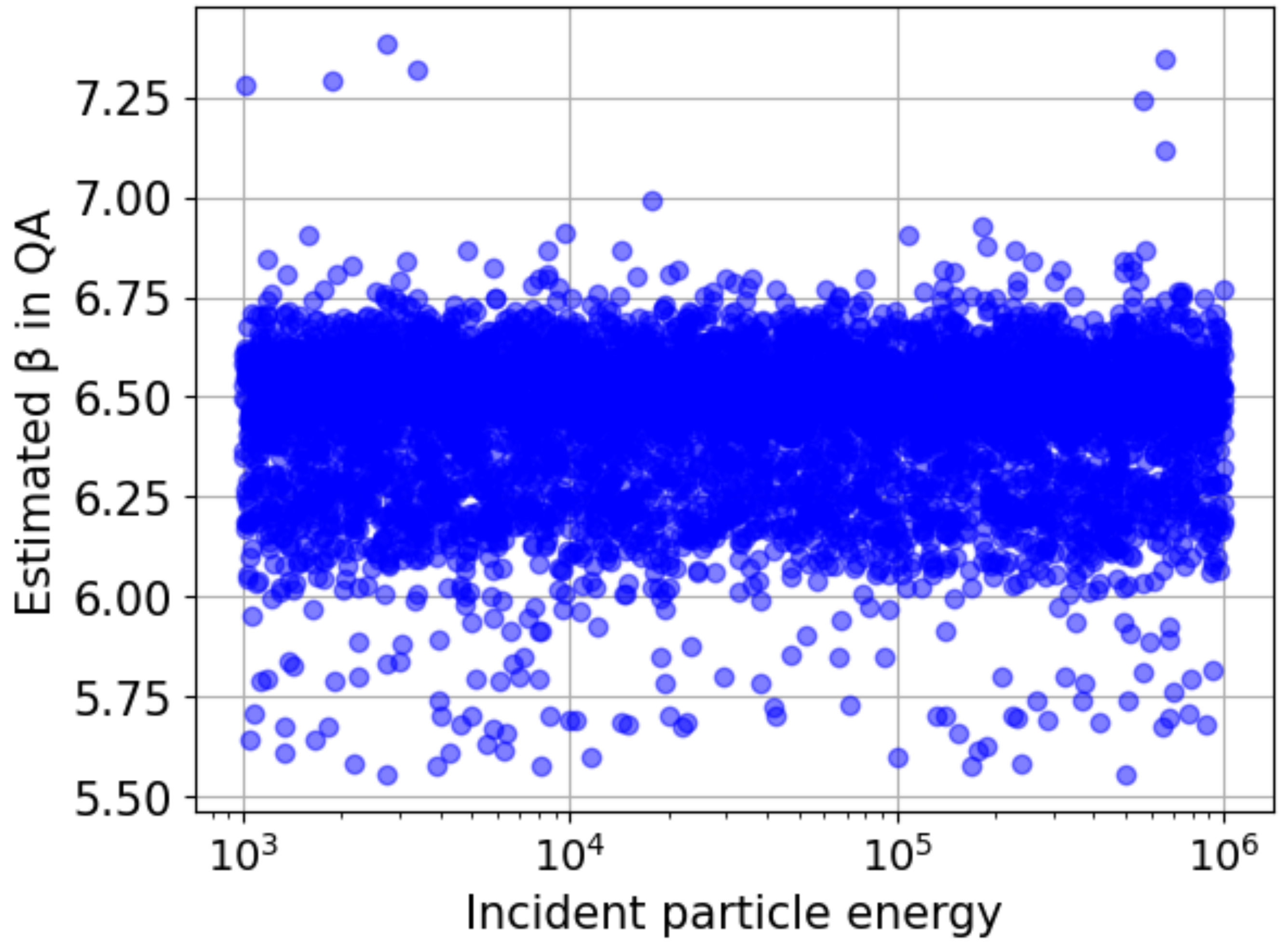


Model B



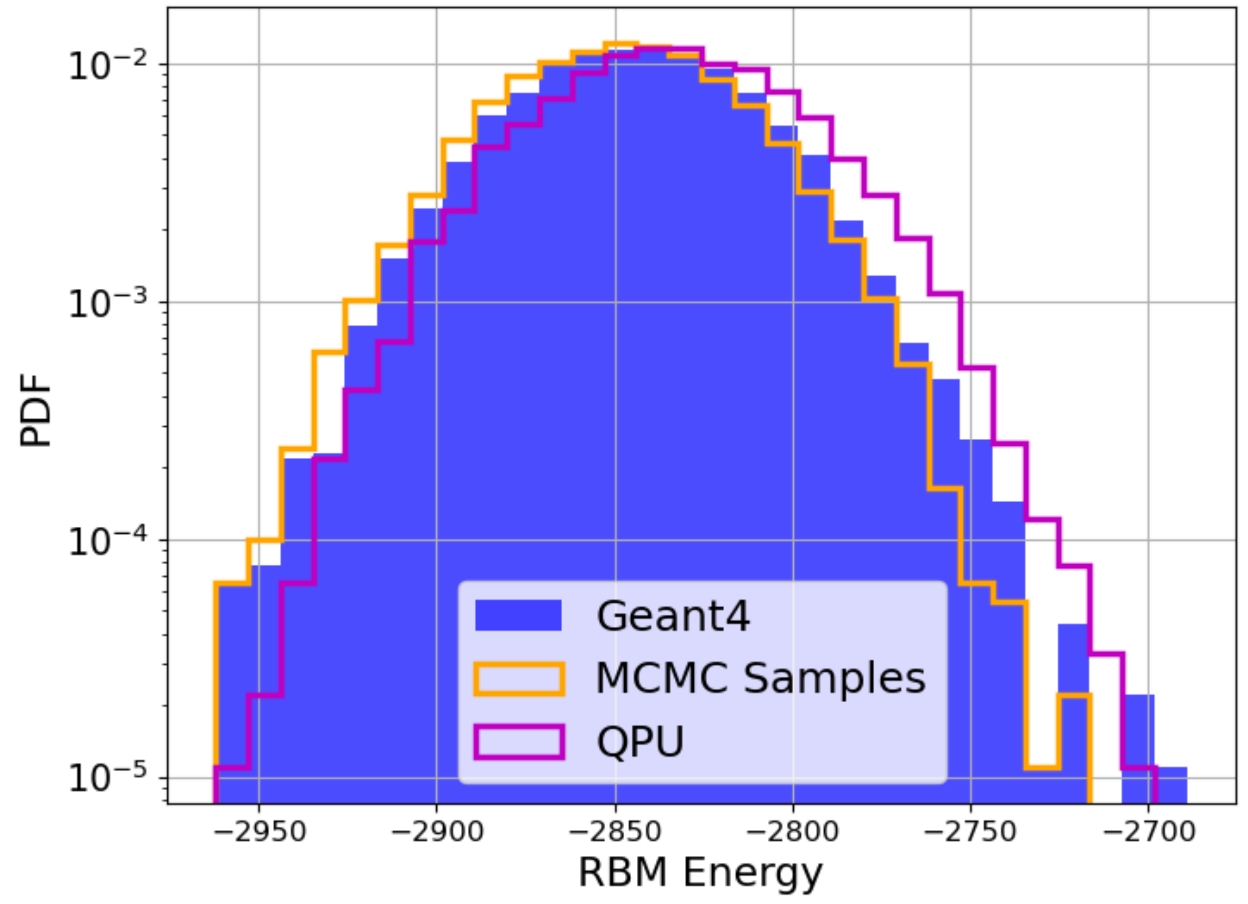


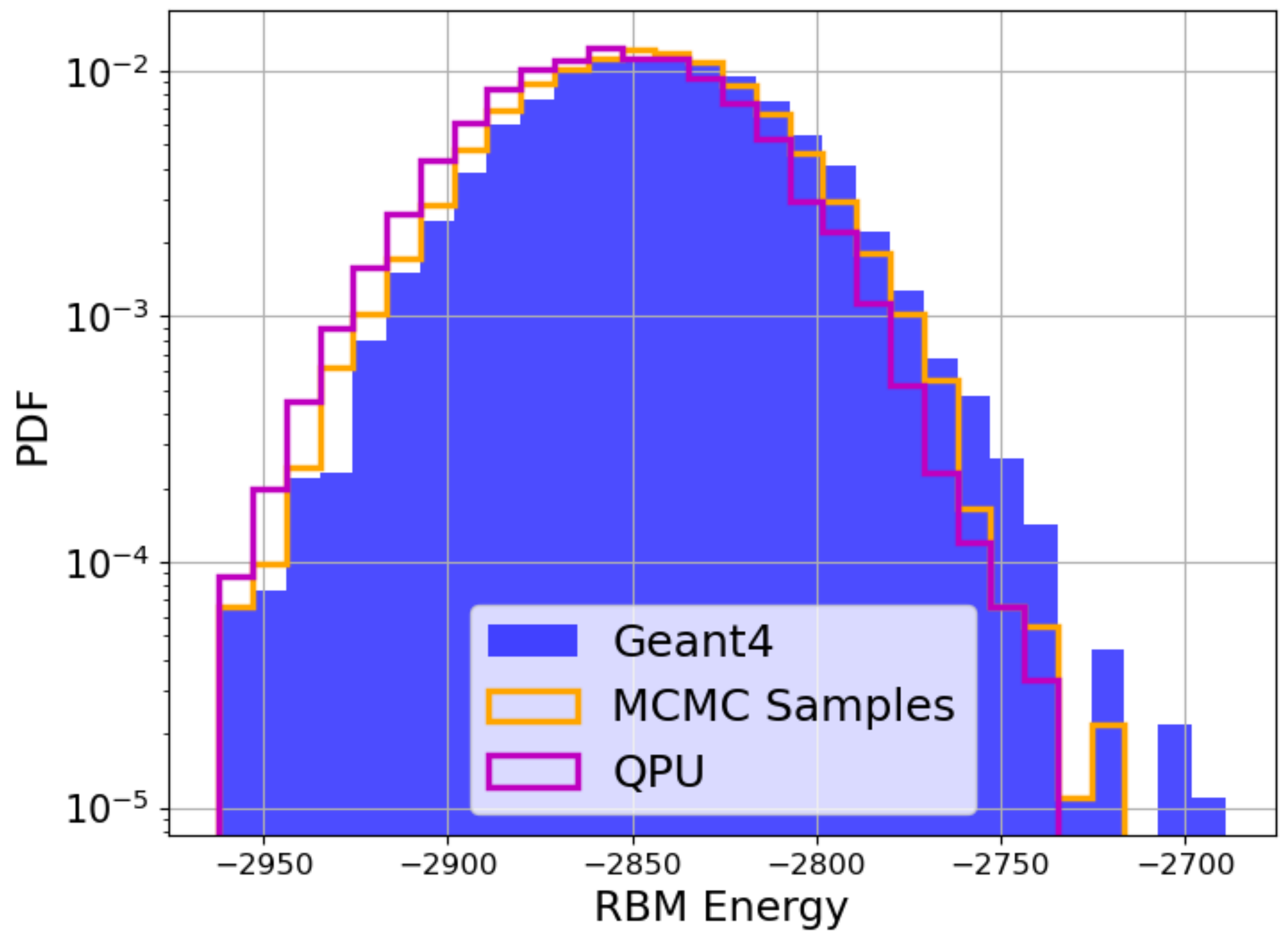




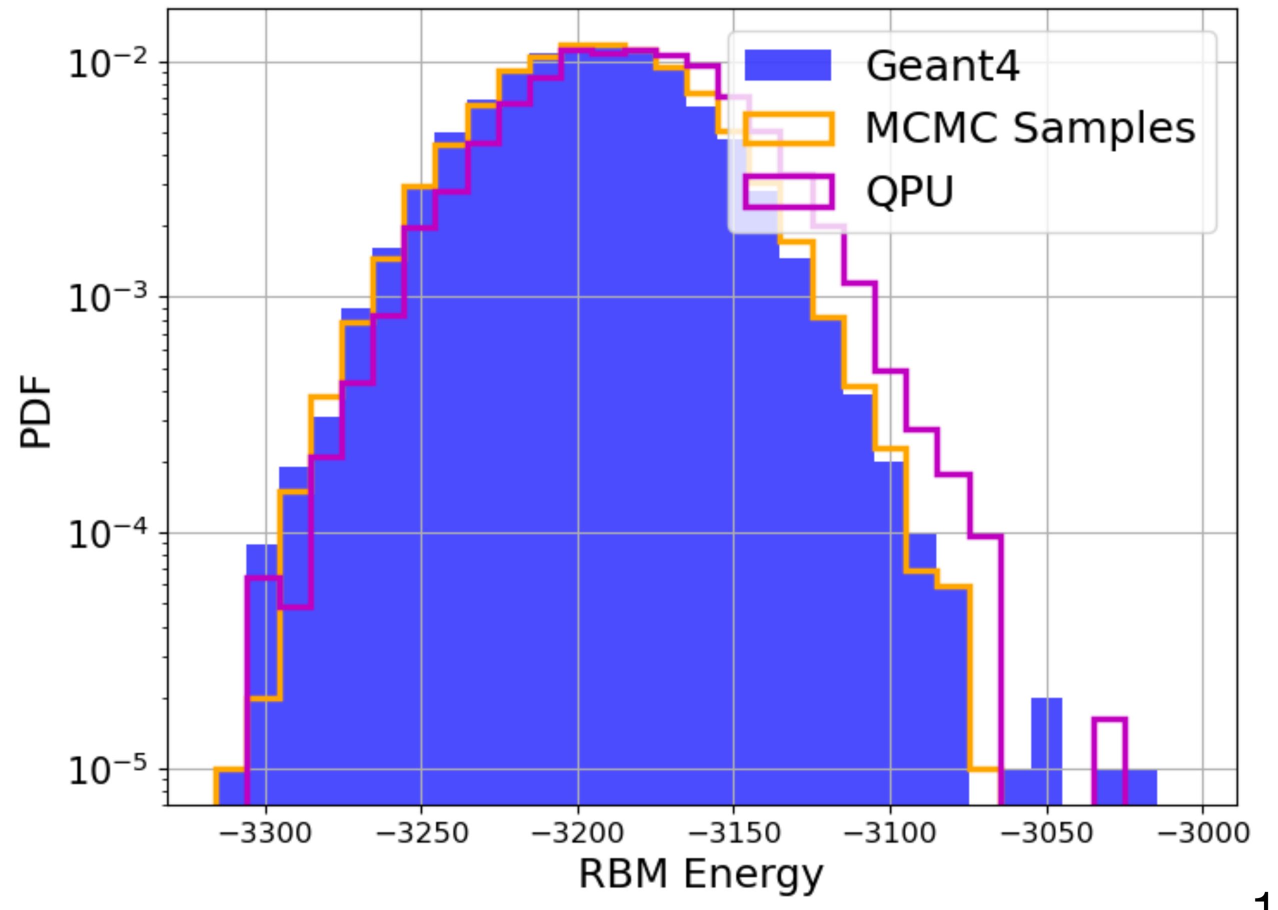
Using QPU w/ Zephyr :: vs Wojtek's method

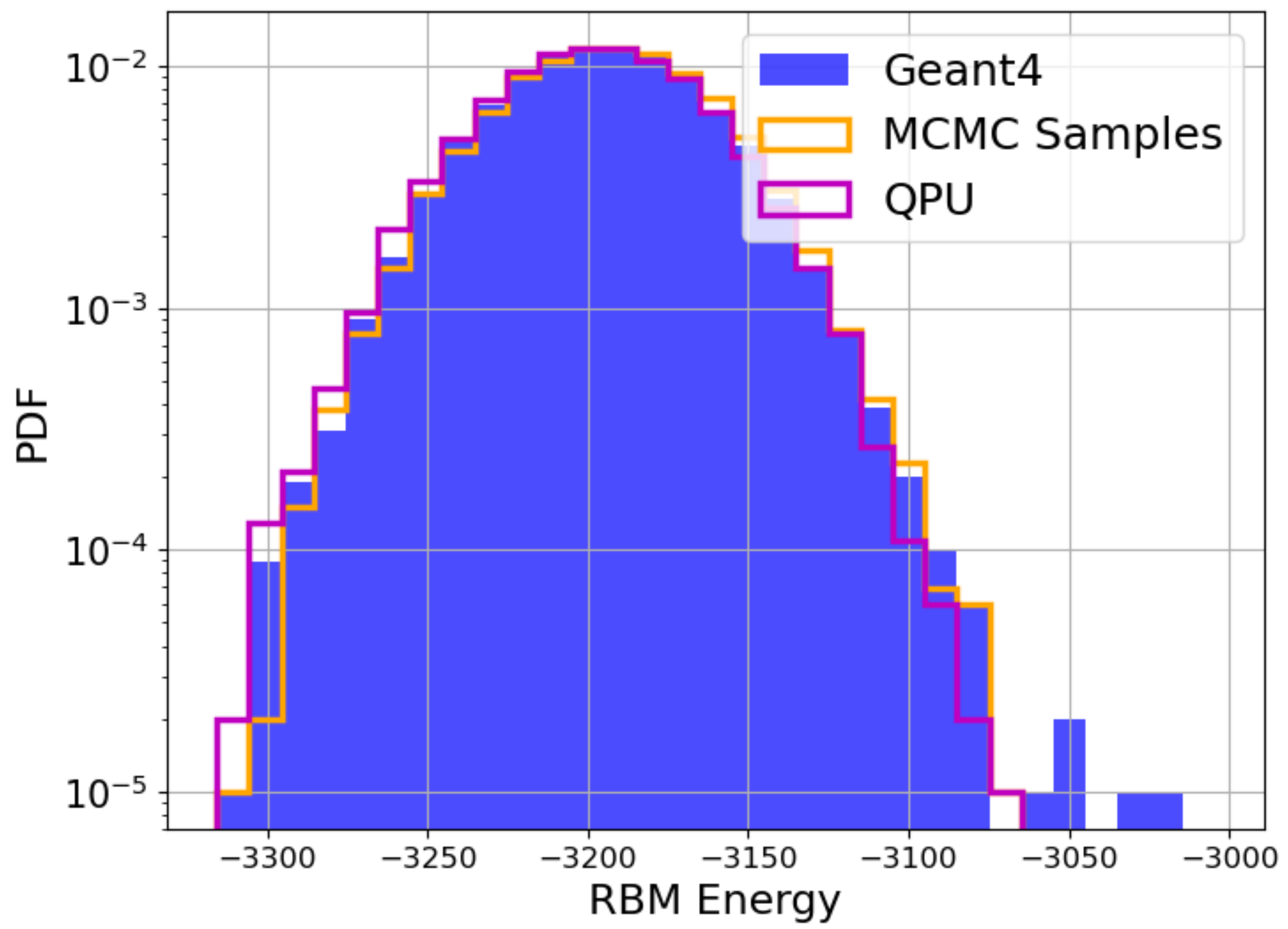
Model A





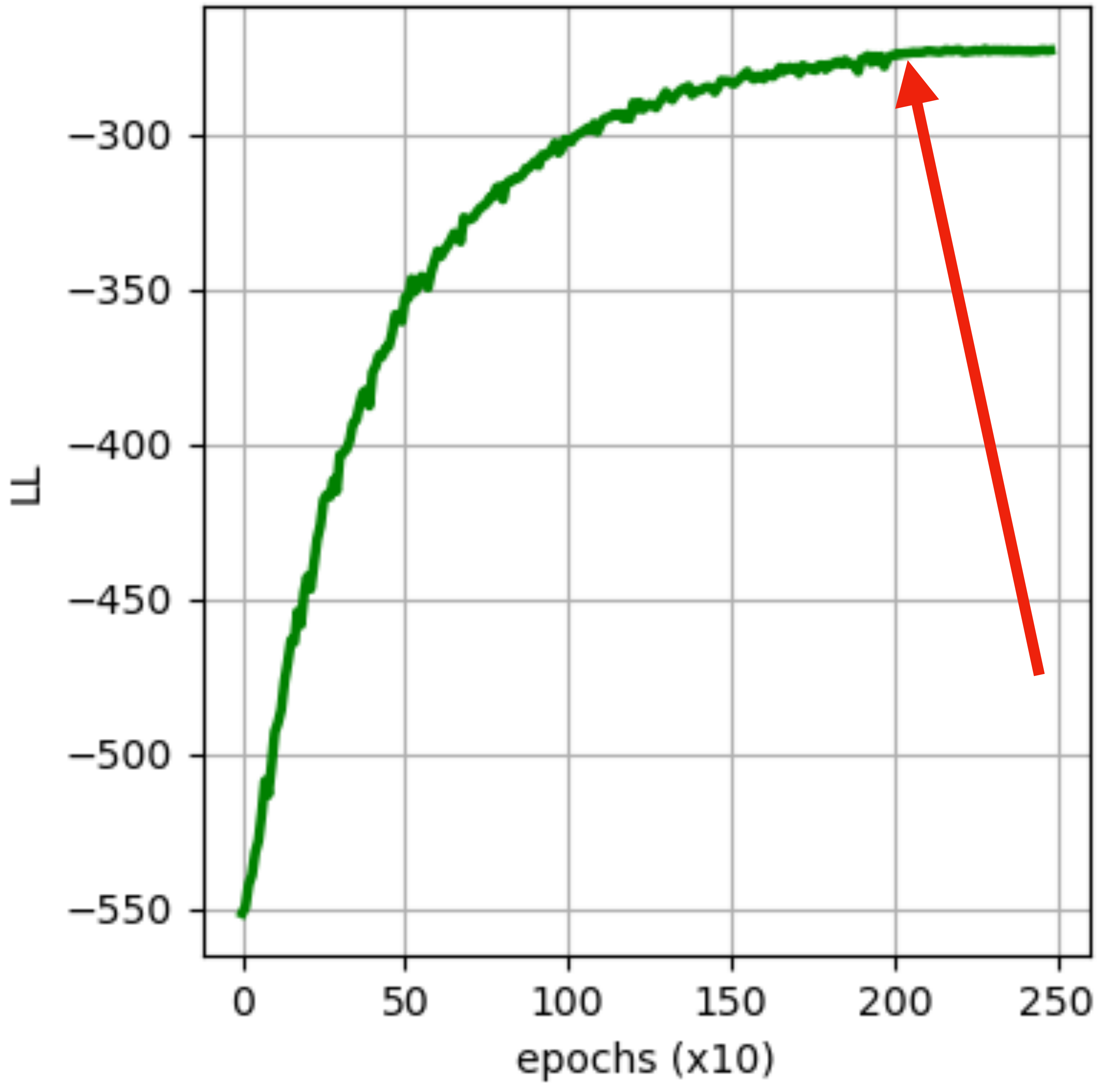
Model B

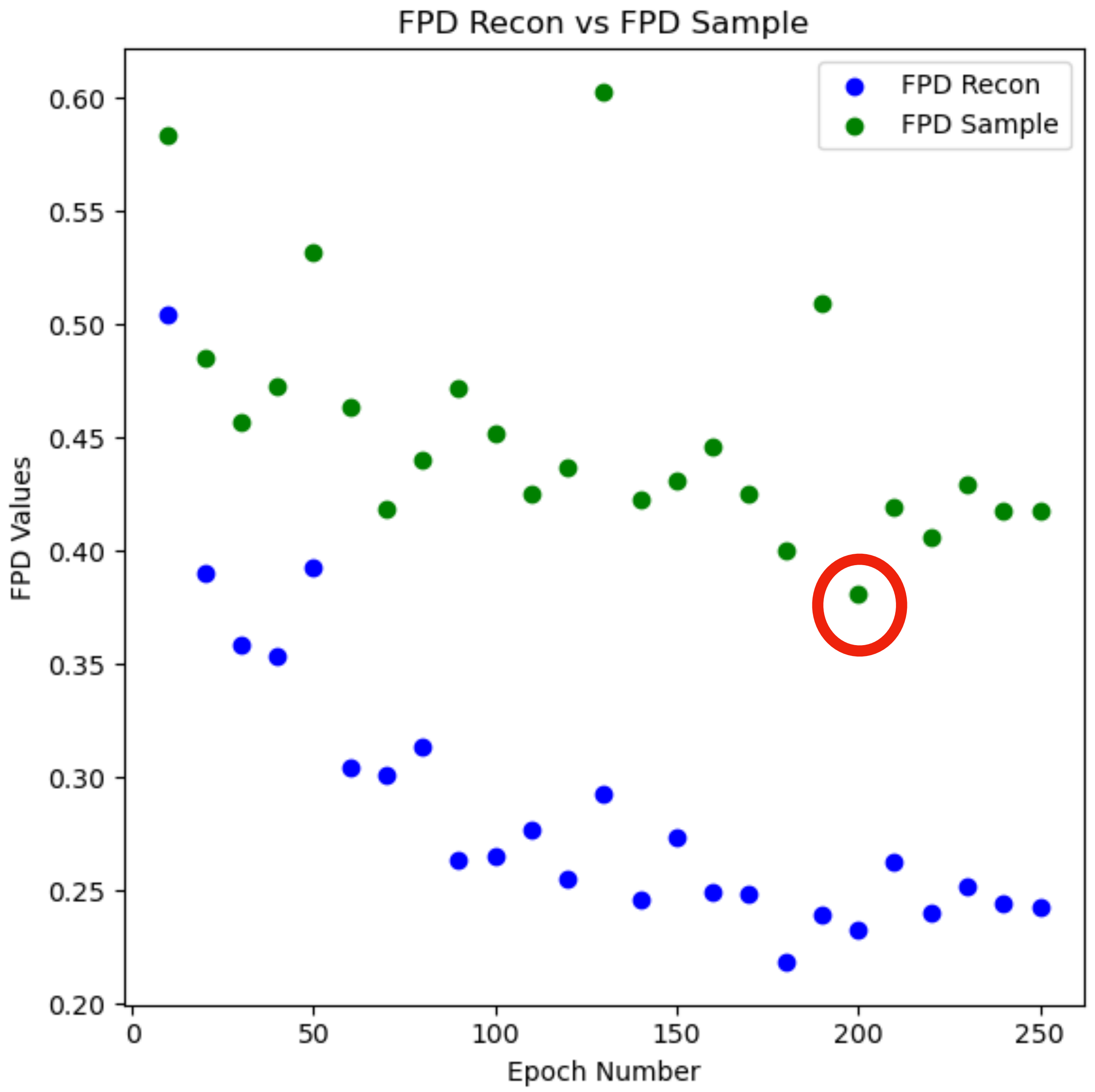


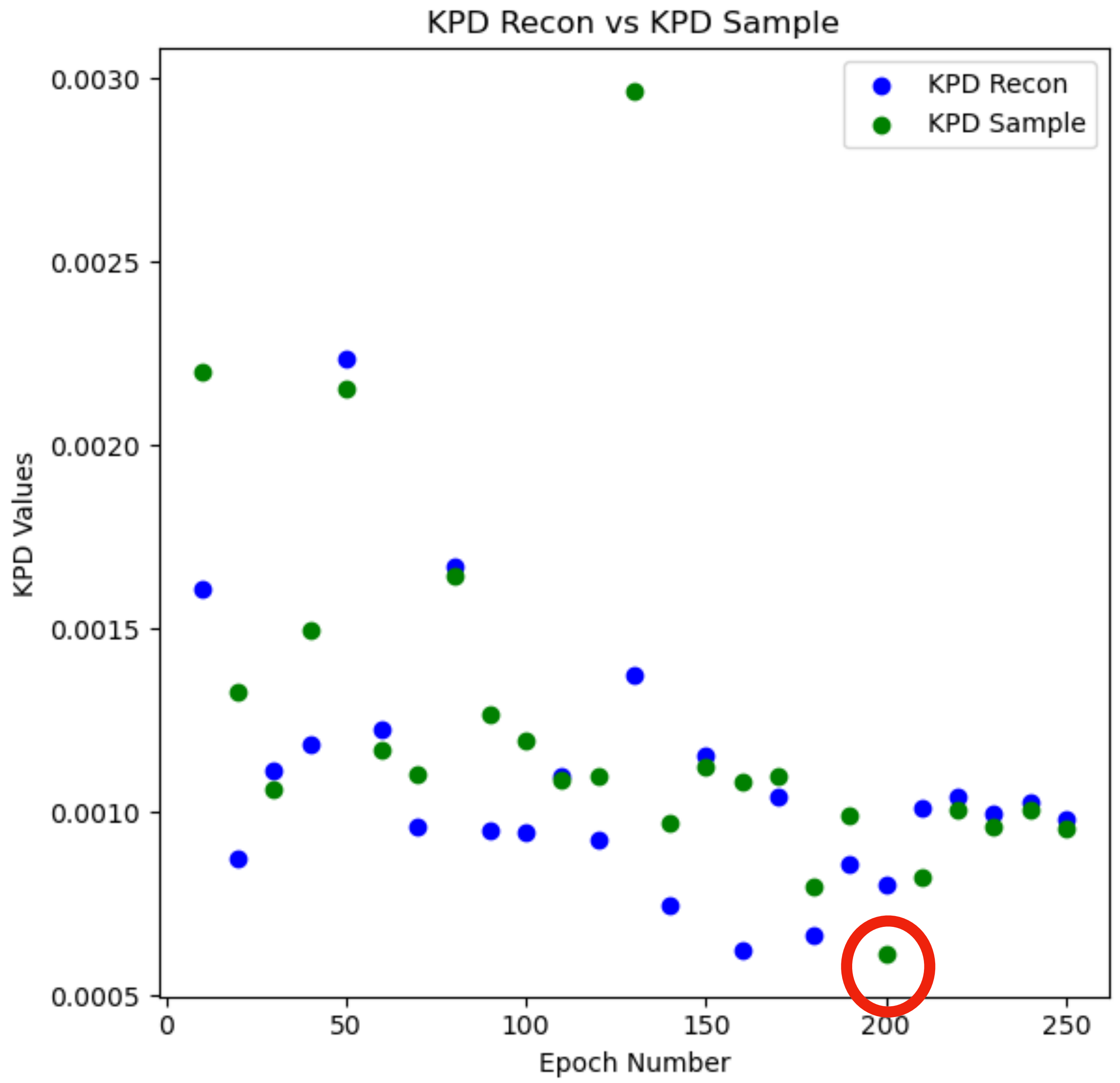


Using QPU w/ Zephyr :: vs Wojtek's method

Model B

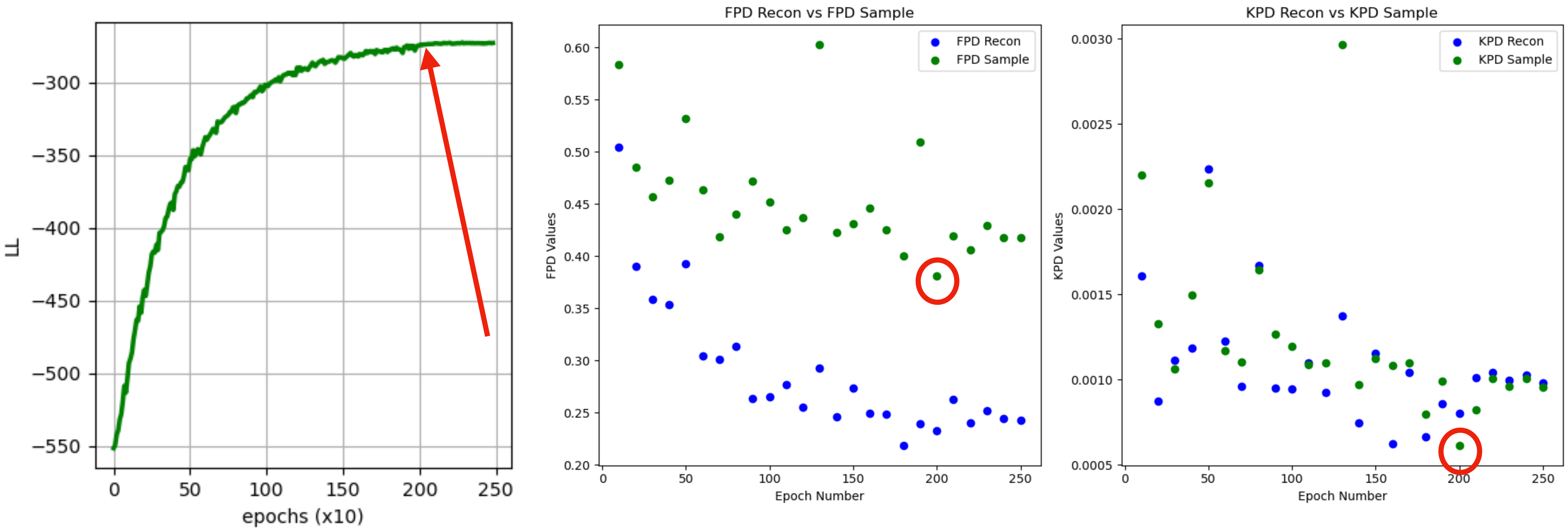






Using QPU w/ Zephyr :: vs Wojtek’s method

Model B



Time required to train using QPU:

(1 sample generation time)X(# of samples)X(epochs)

(20ms)X(100k)X(200) = 111.1 hrs

Using QPU w/ Zephyr :: vs Wojtek’s method

- Train Enc and Decoder and train QPU afterwards with a smaller sample.
- Discuss with dwave options and roadmaps

High Temperature gradient approximation

$$\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle = 0$$

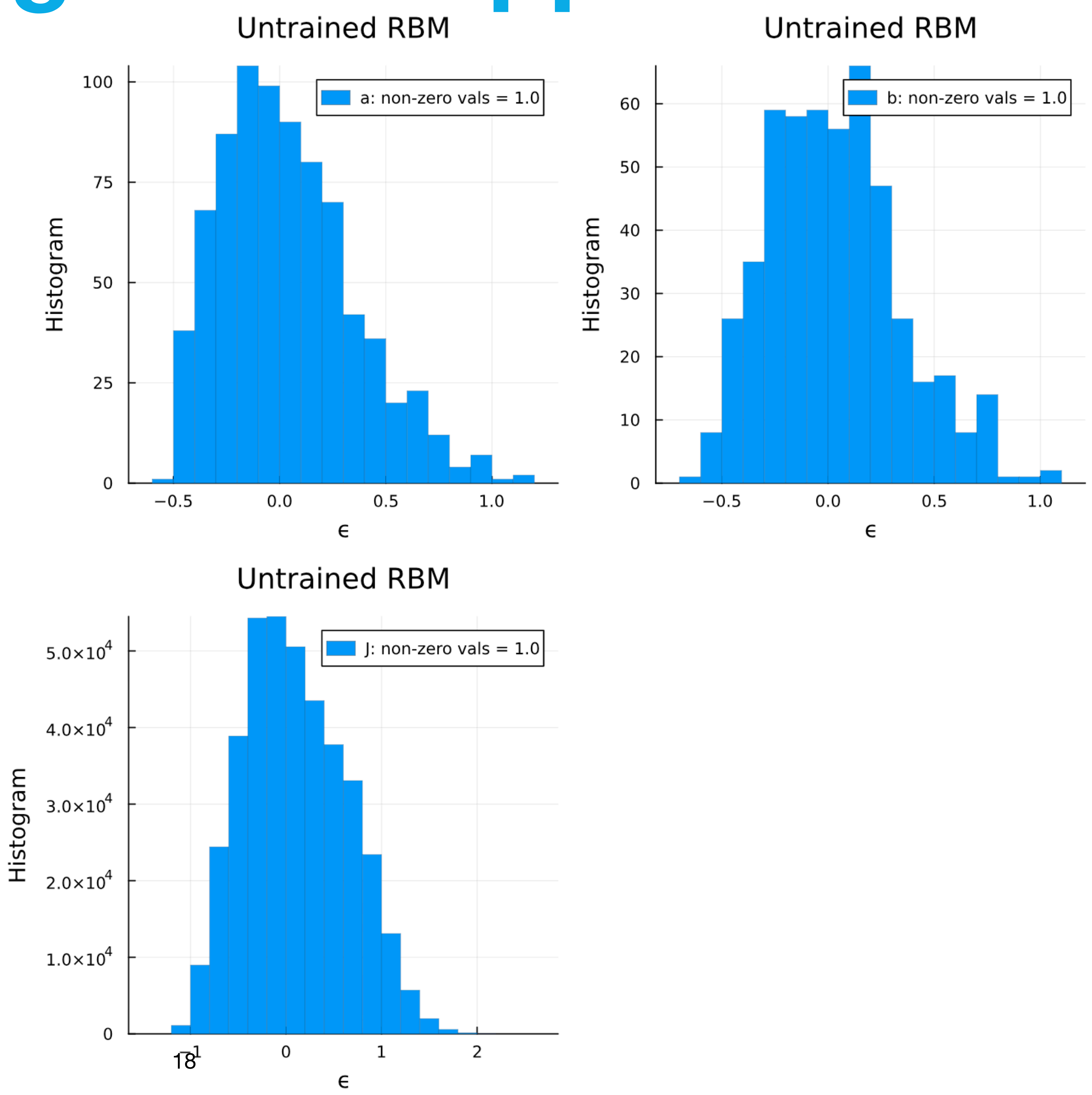
Condition

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

1000 Gibbs sampling steps

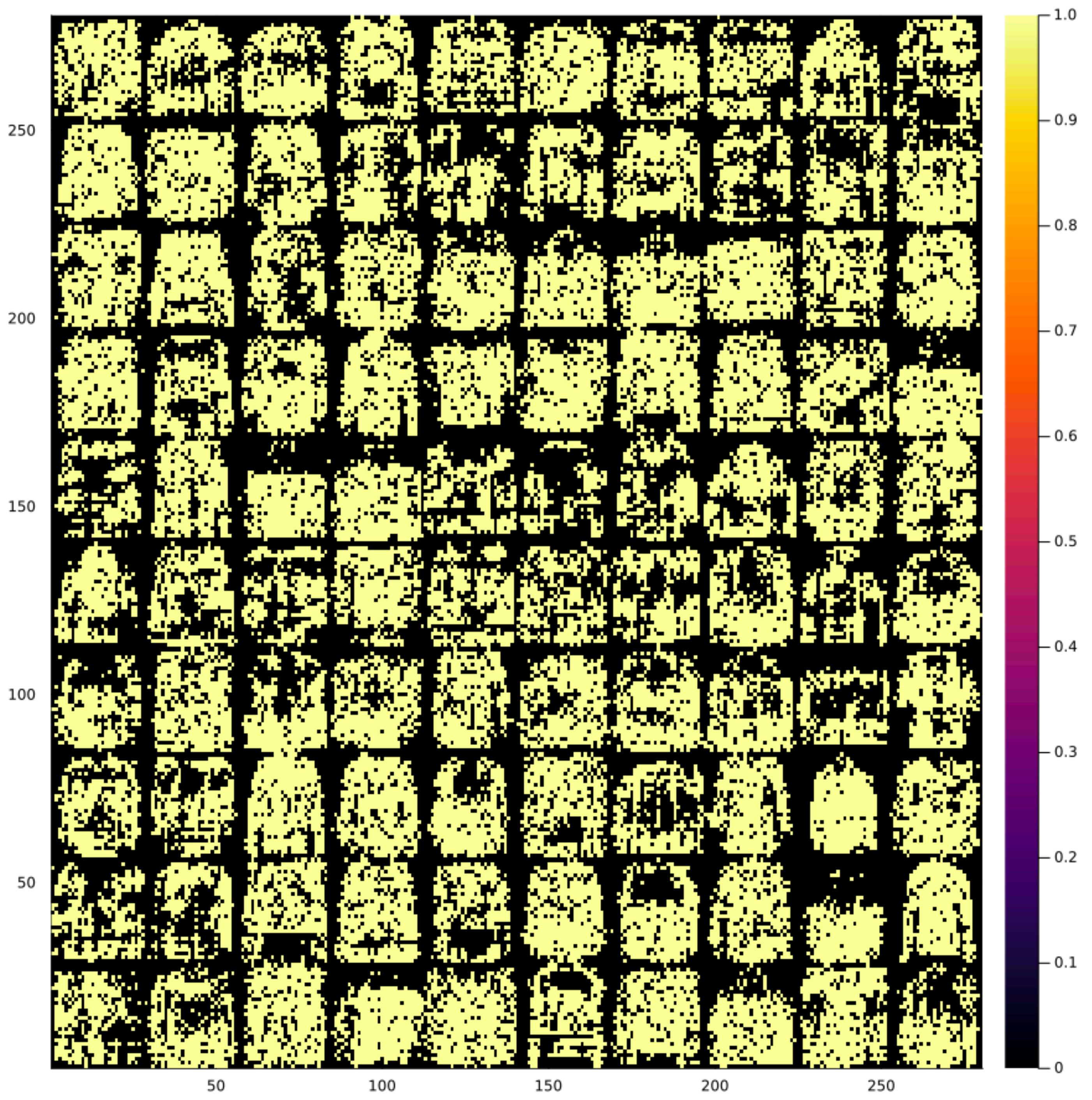
High Temperature gradient approximation

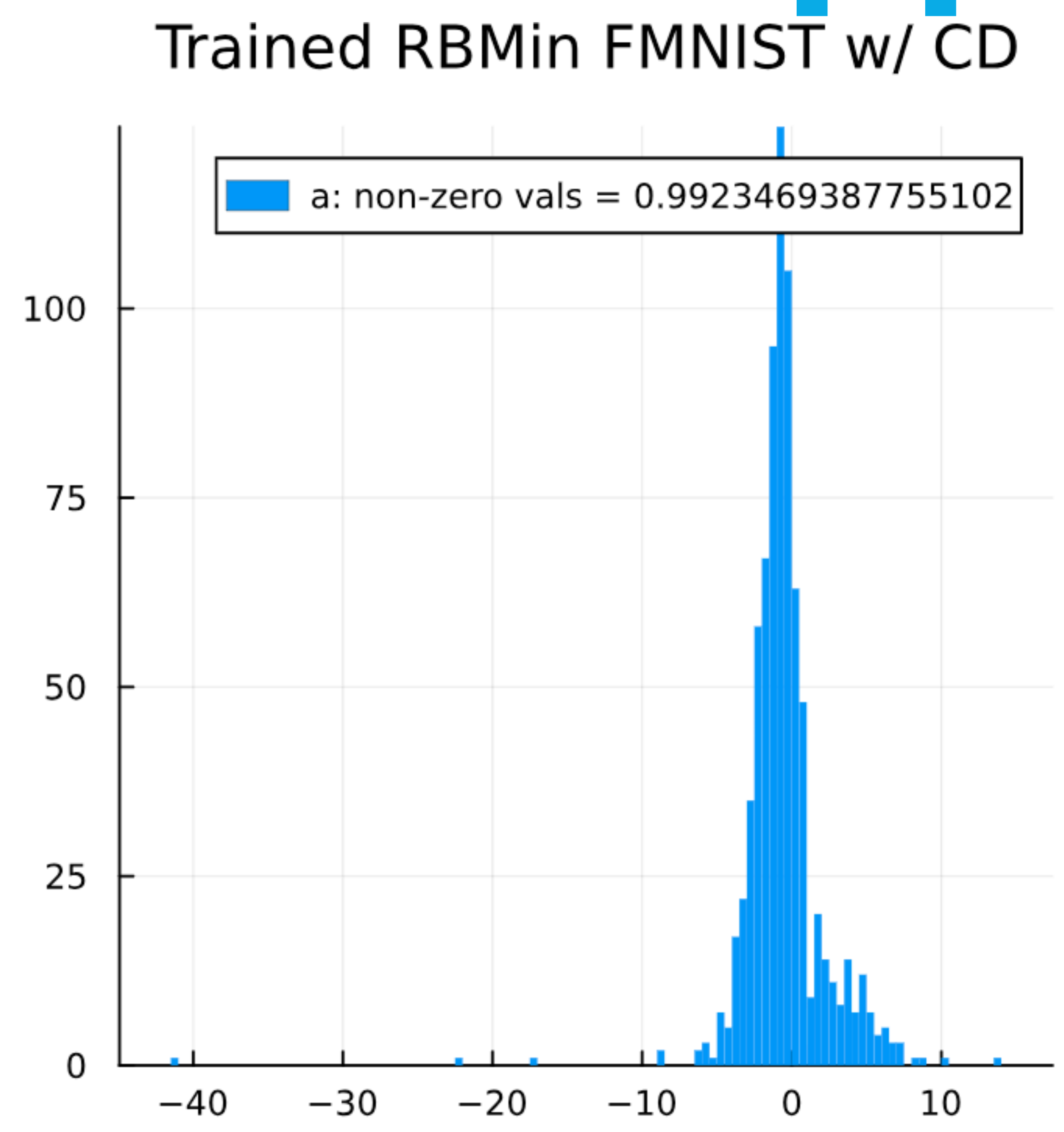
$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

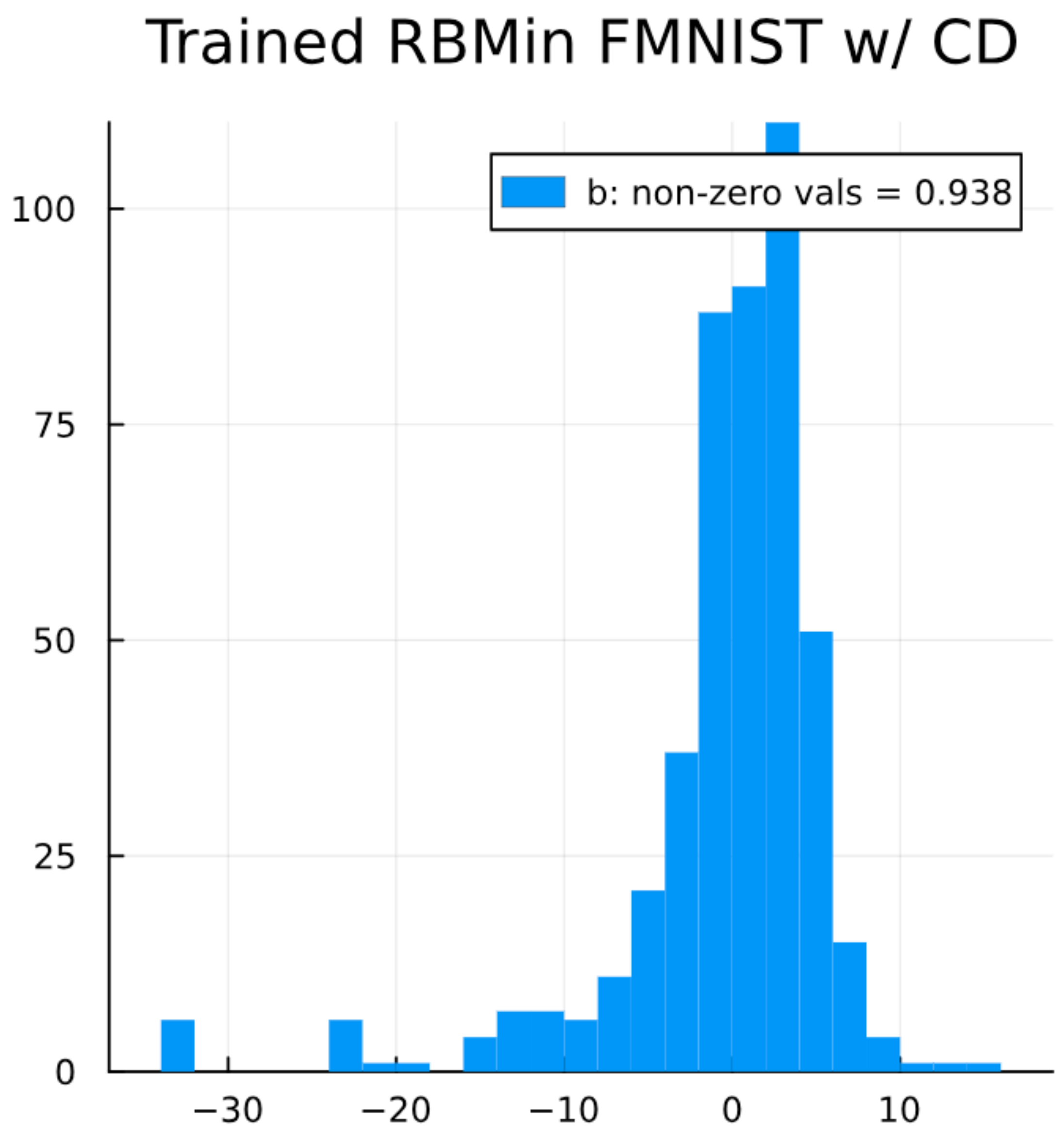


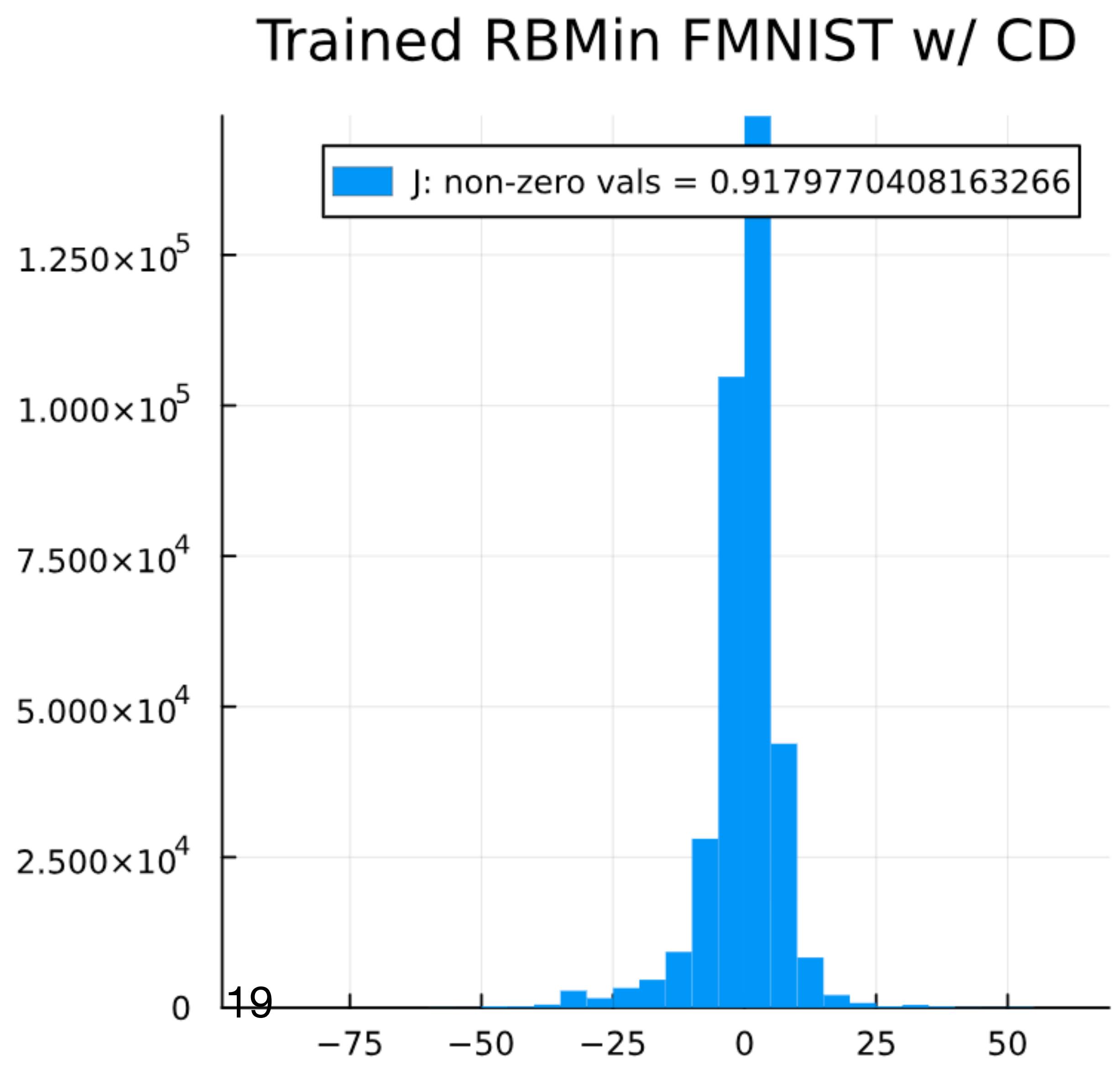
High Temperature gradient approximation

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$



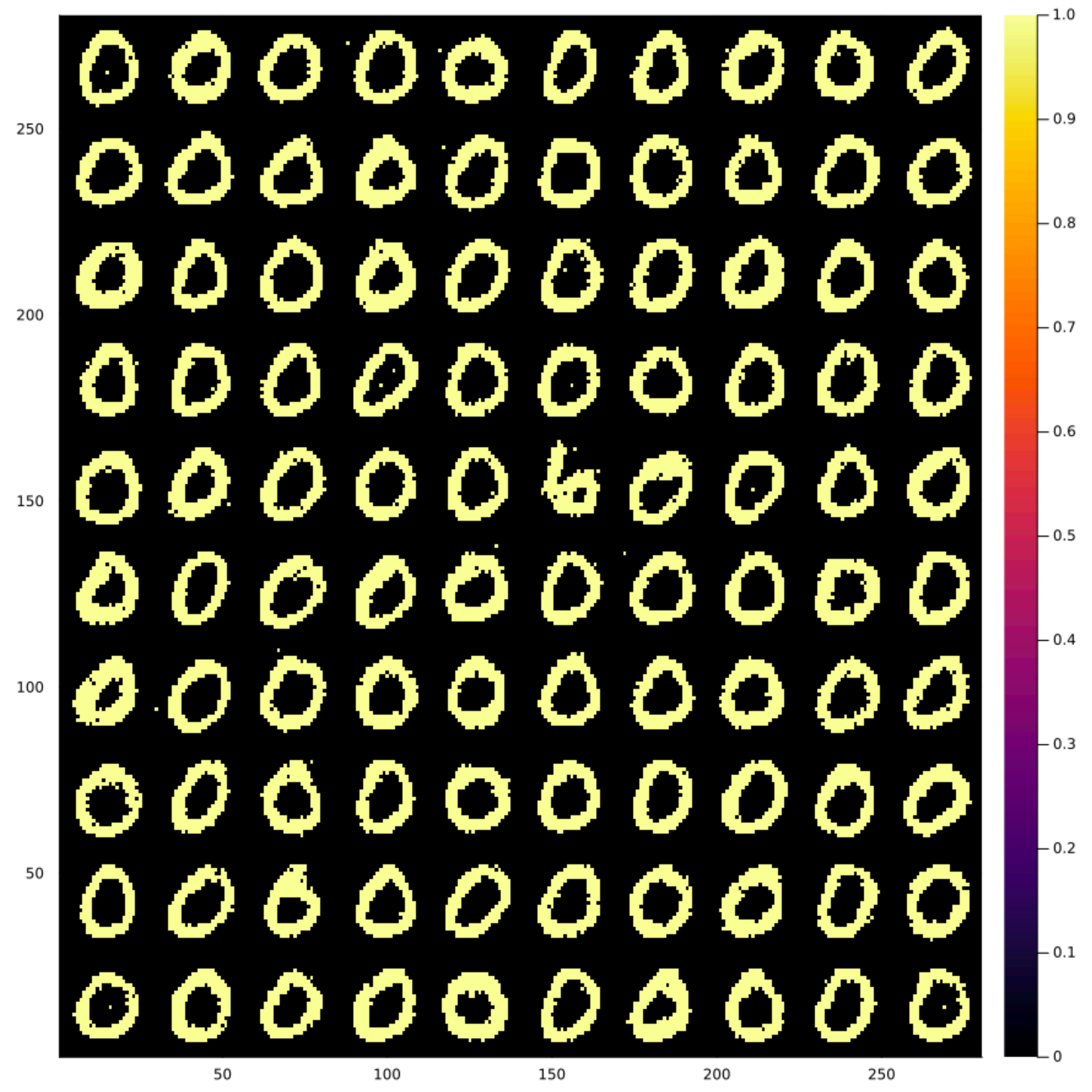


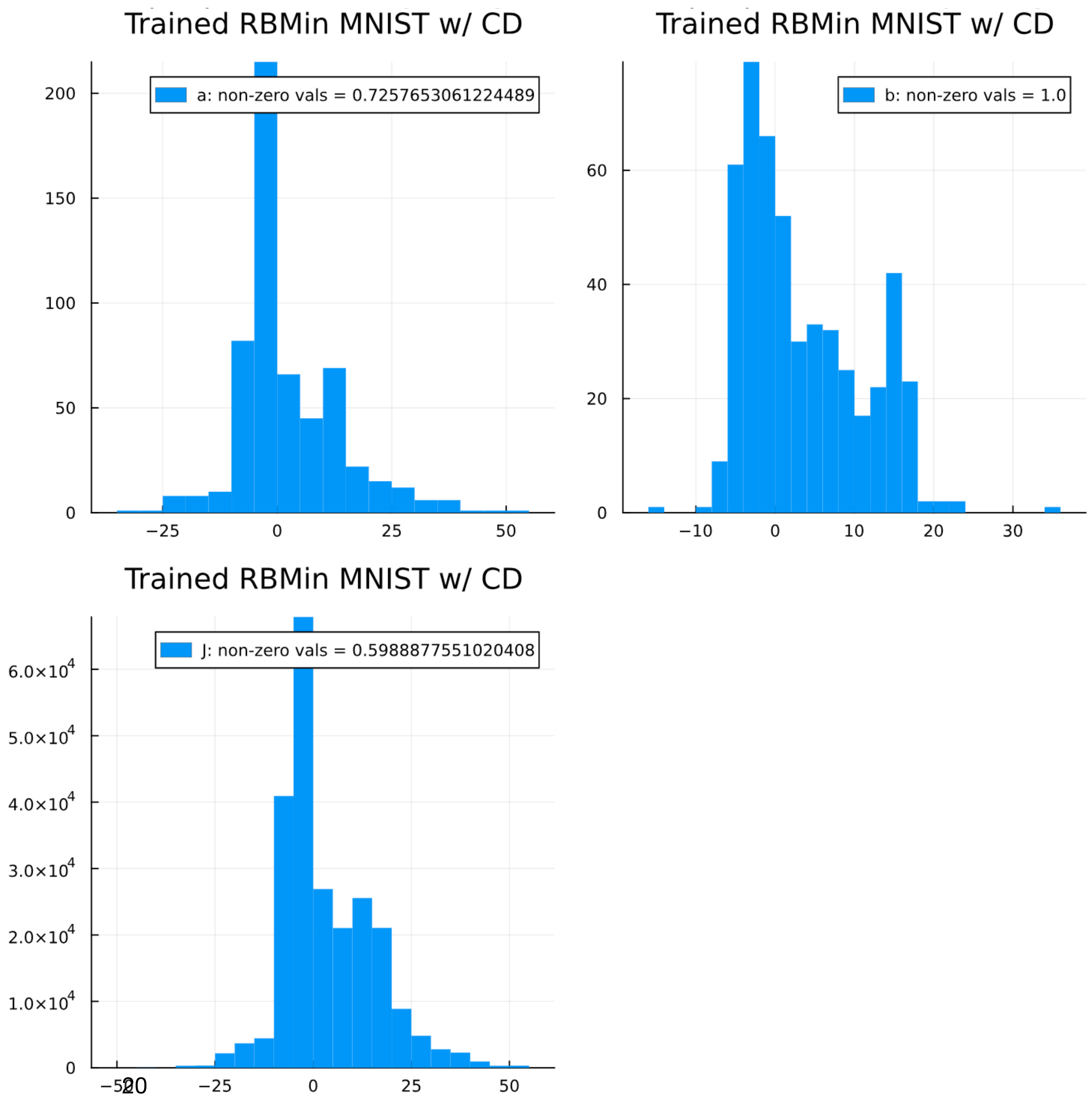




High Temperature gradient approximation

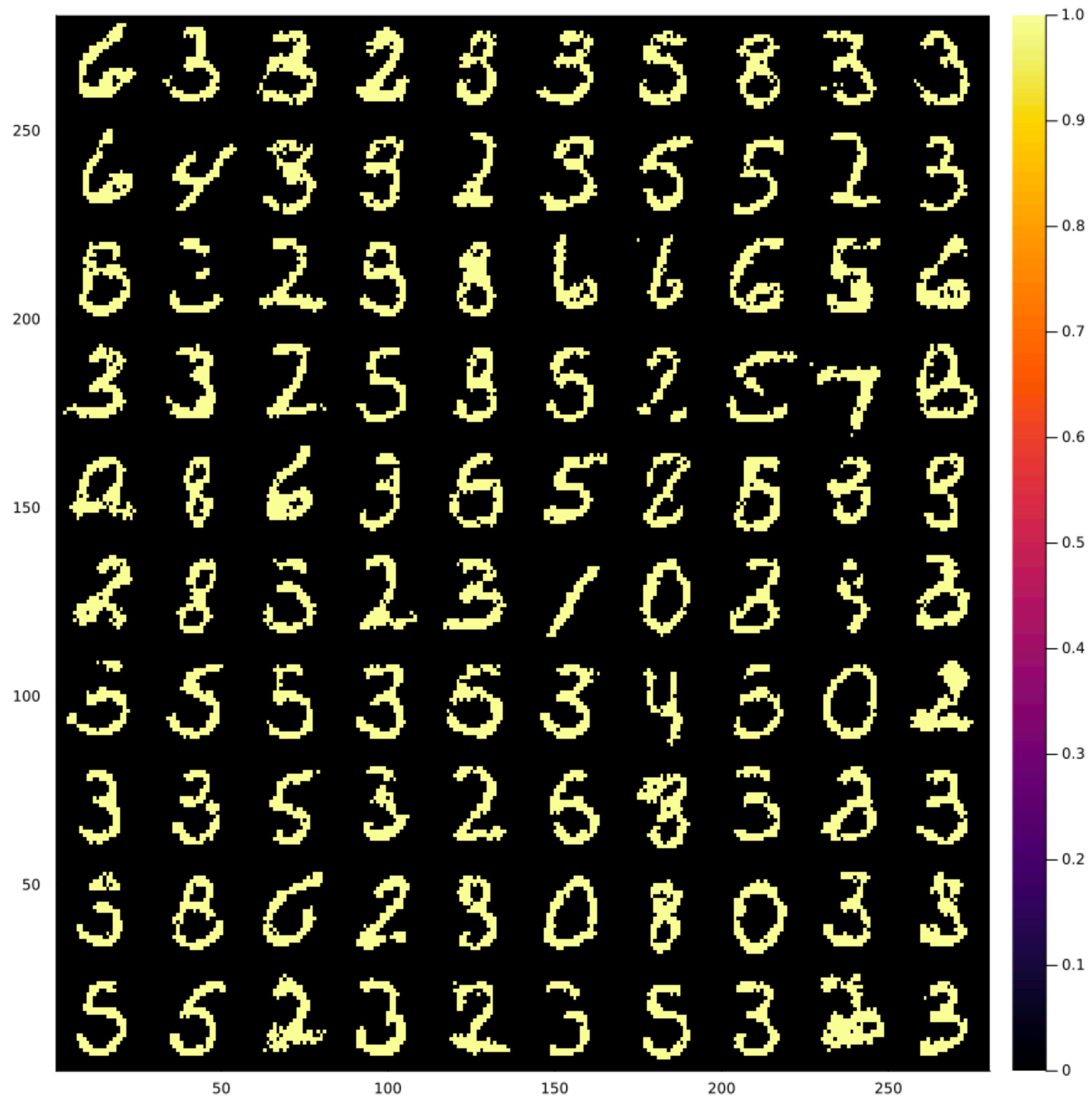
$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

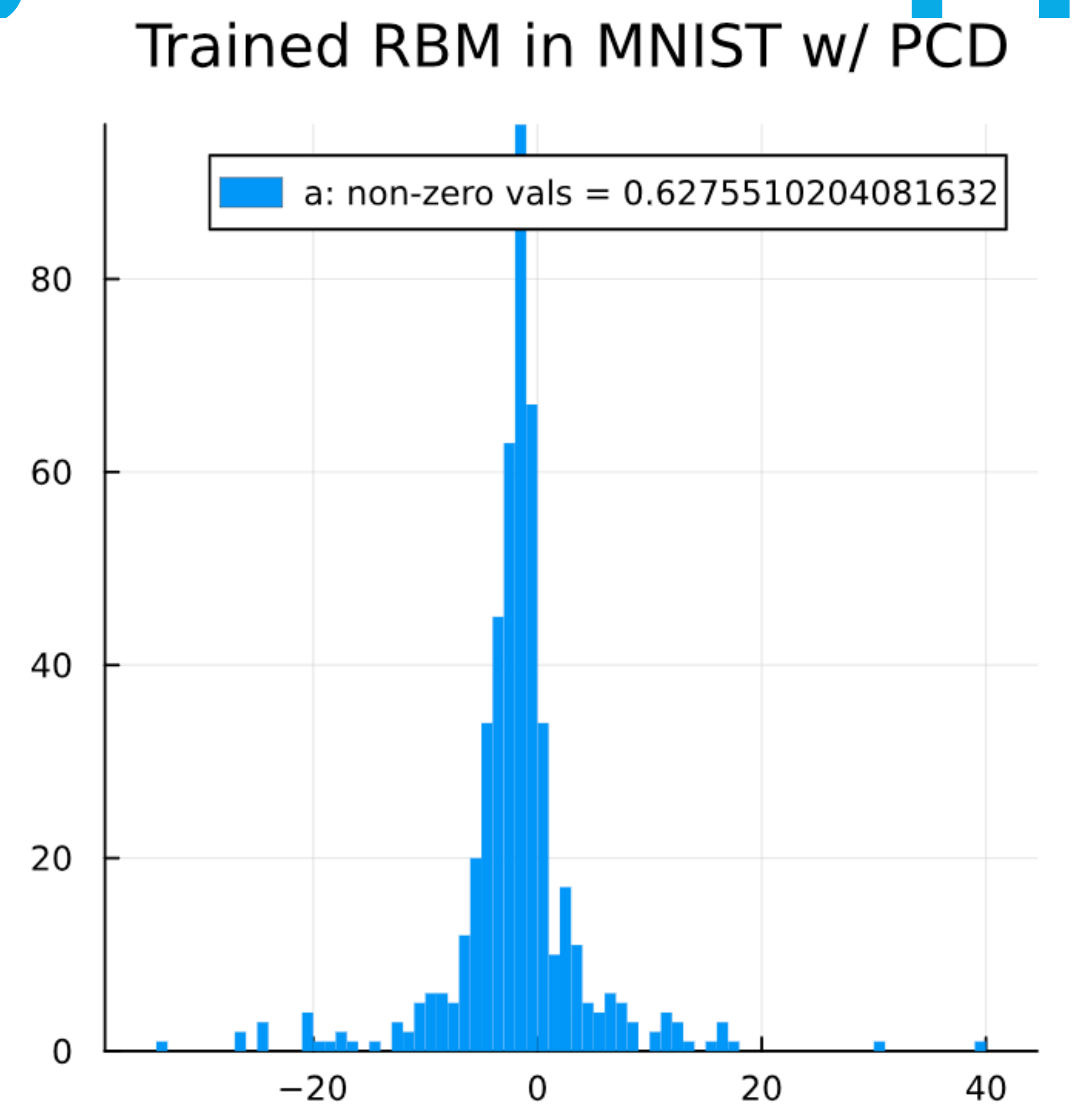


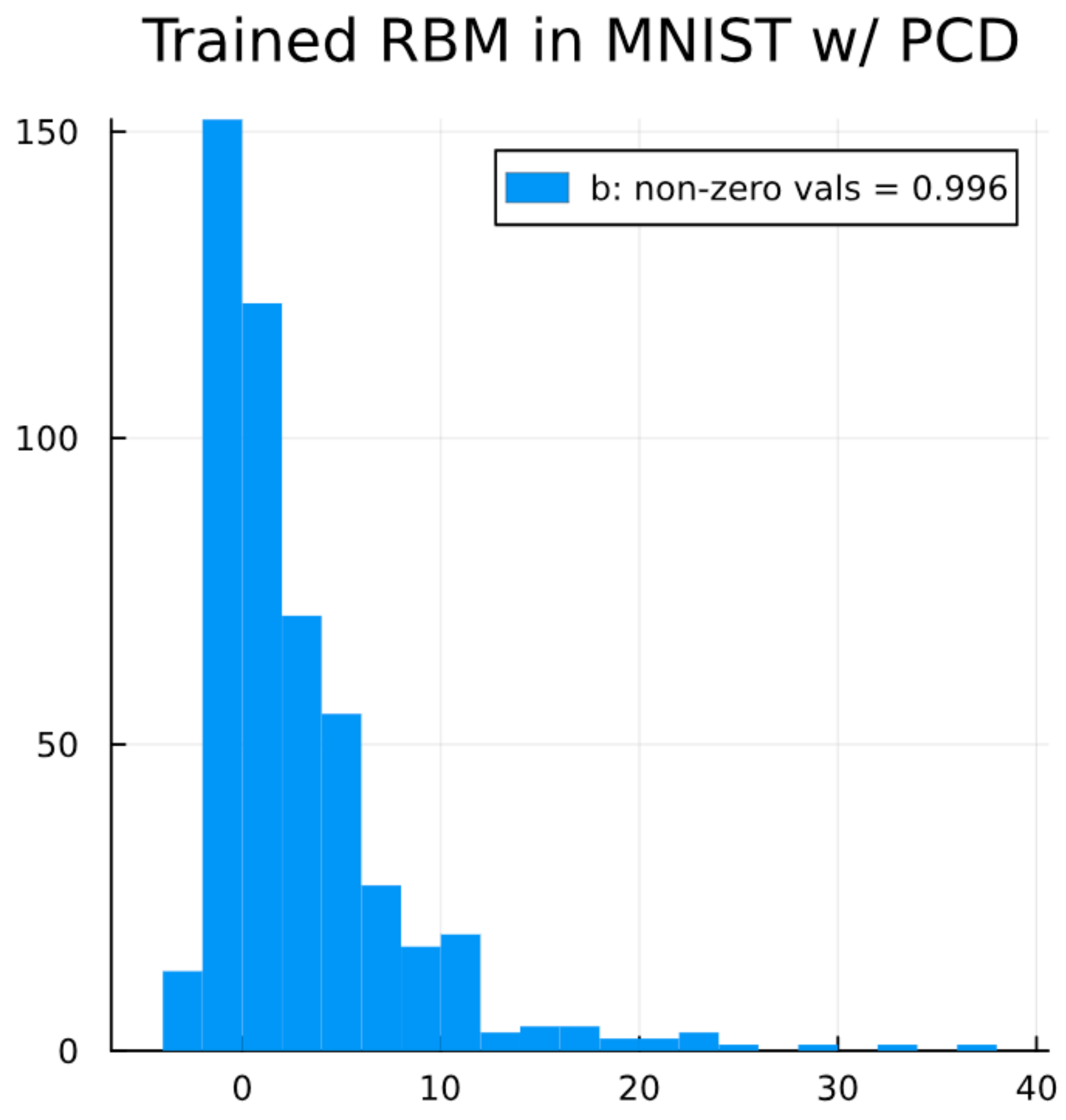


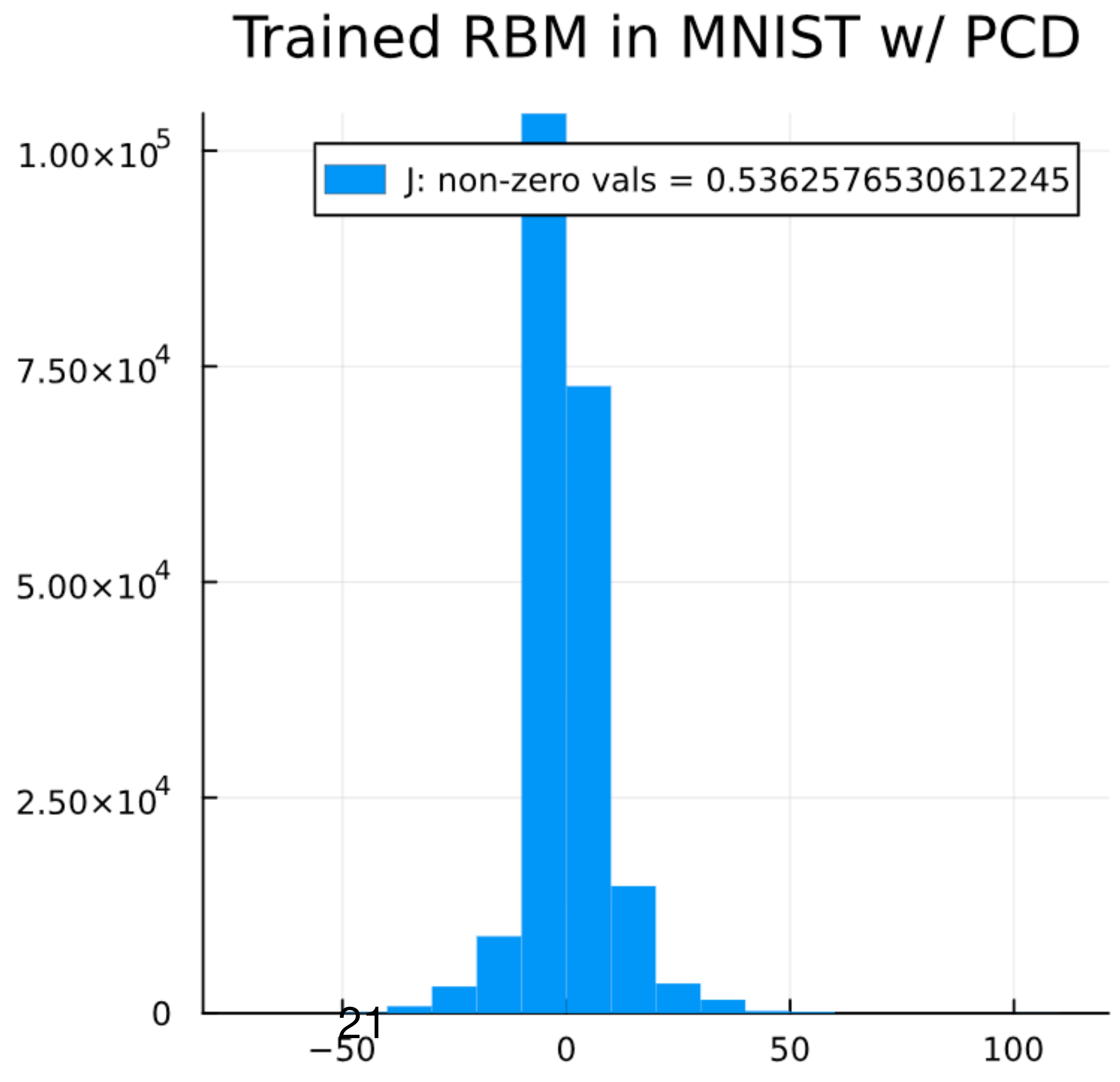
High Temperature gradient approximation

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$







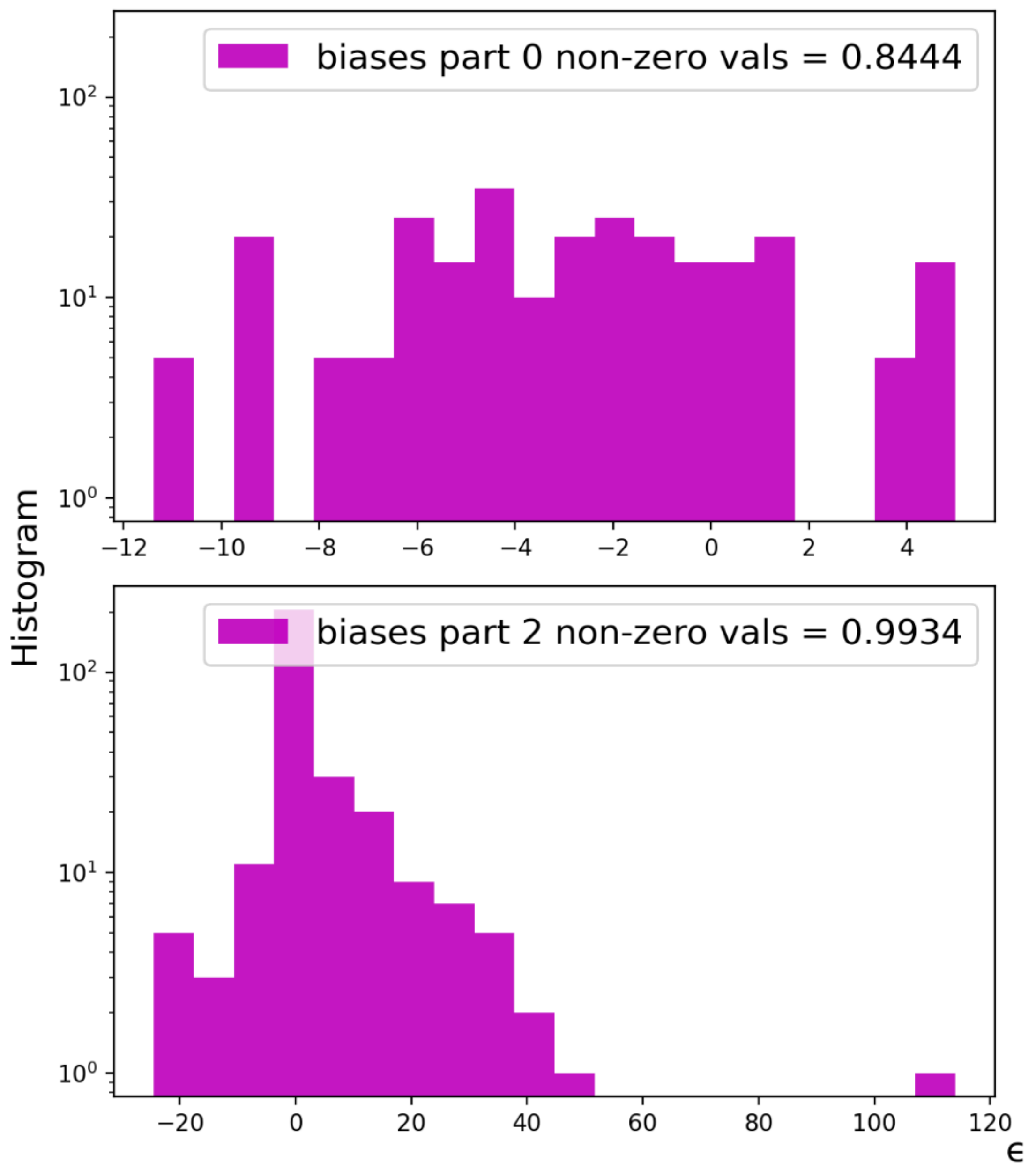


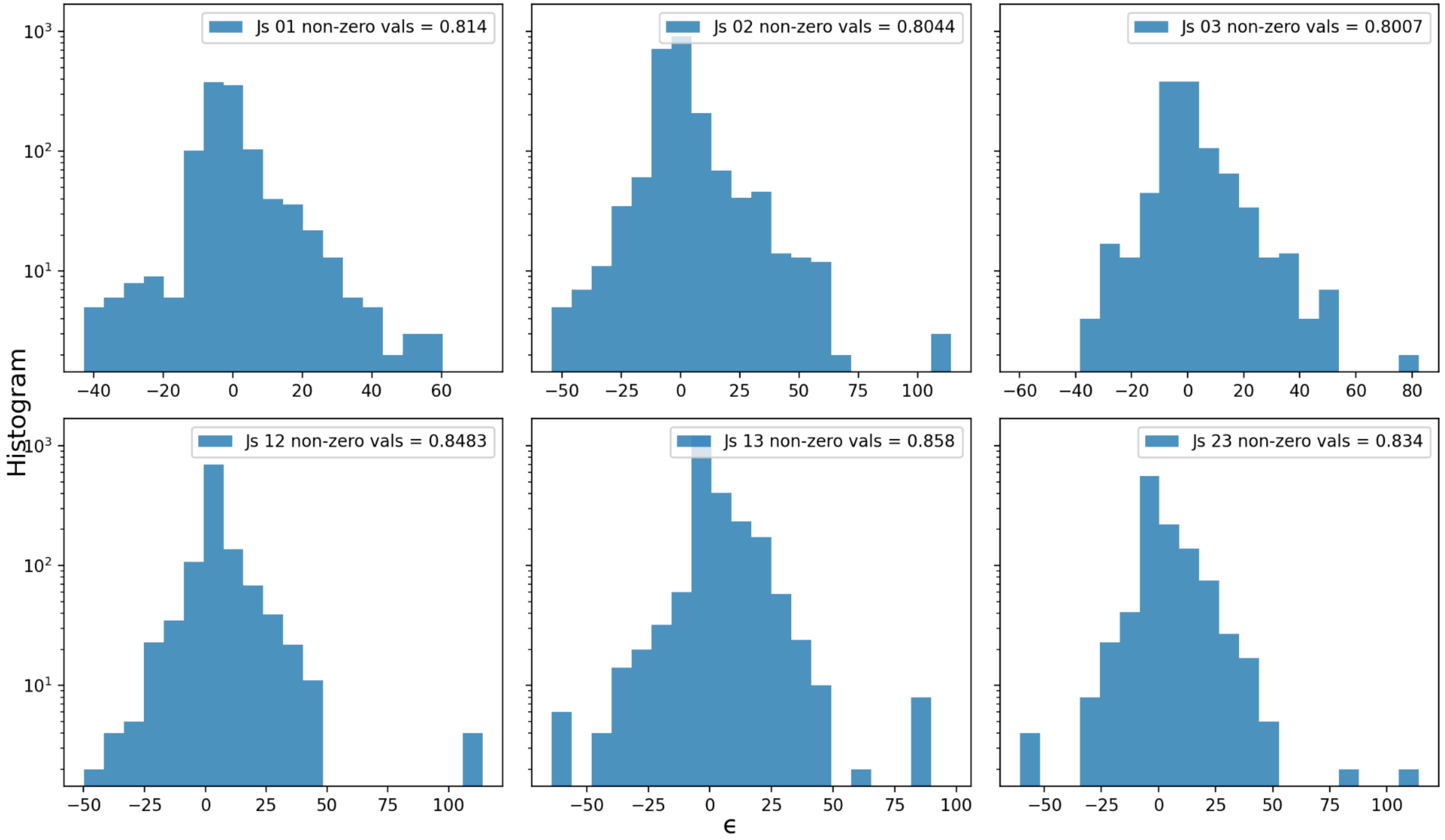
High Temperature gradient approximation

Each point in the histogram corresponds to a parameter being updated *separately* from the rest.

High Temperature gradient approximation

CaloQVAE Model B





High Temperature gradient approximation

Small RBMs

Let's assume an RBM w/ 10 visible and 10 hidden nodes.

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

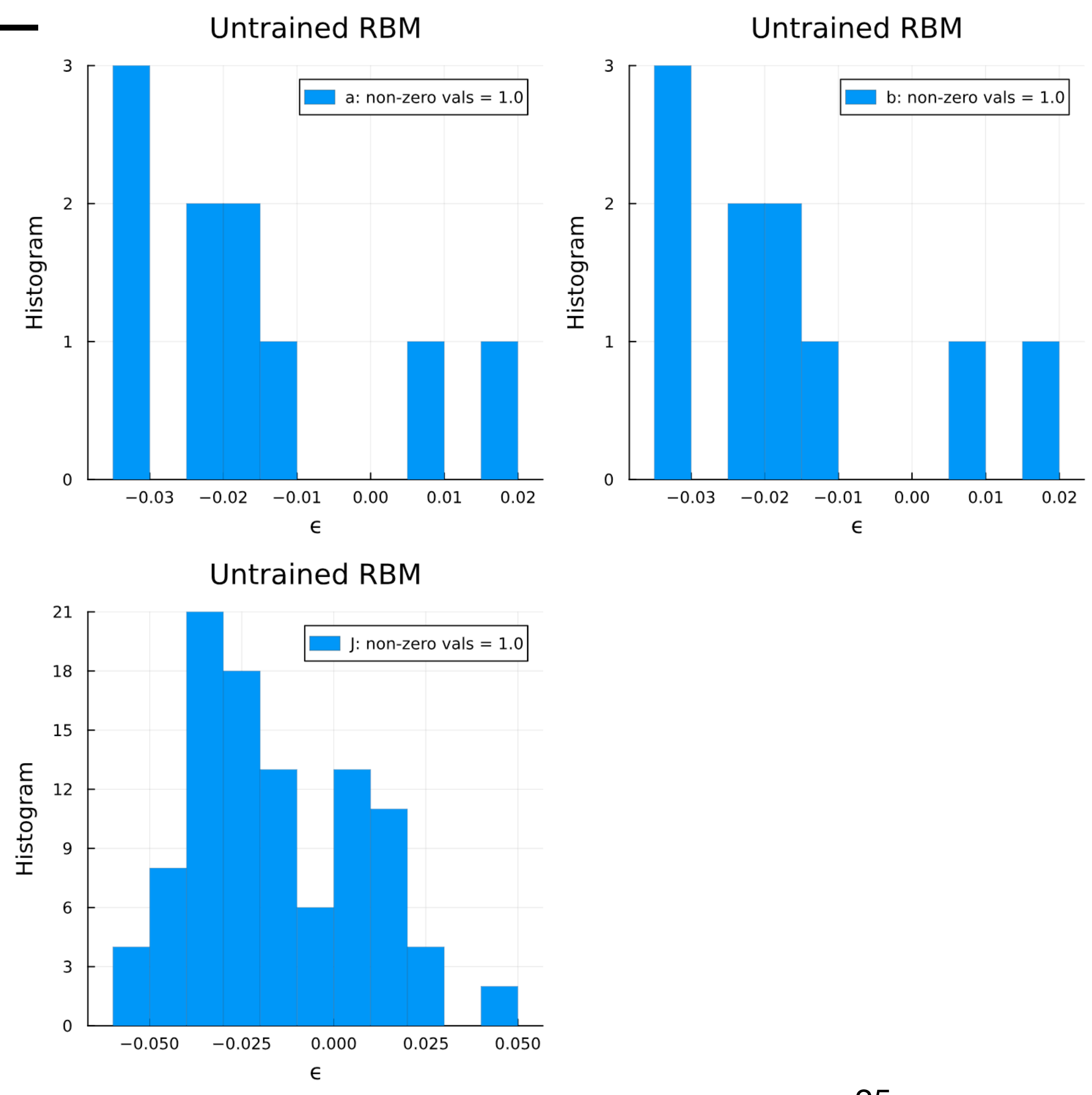
High Temperature gradient approximation

Small RBMs

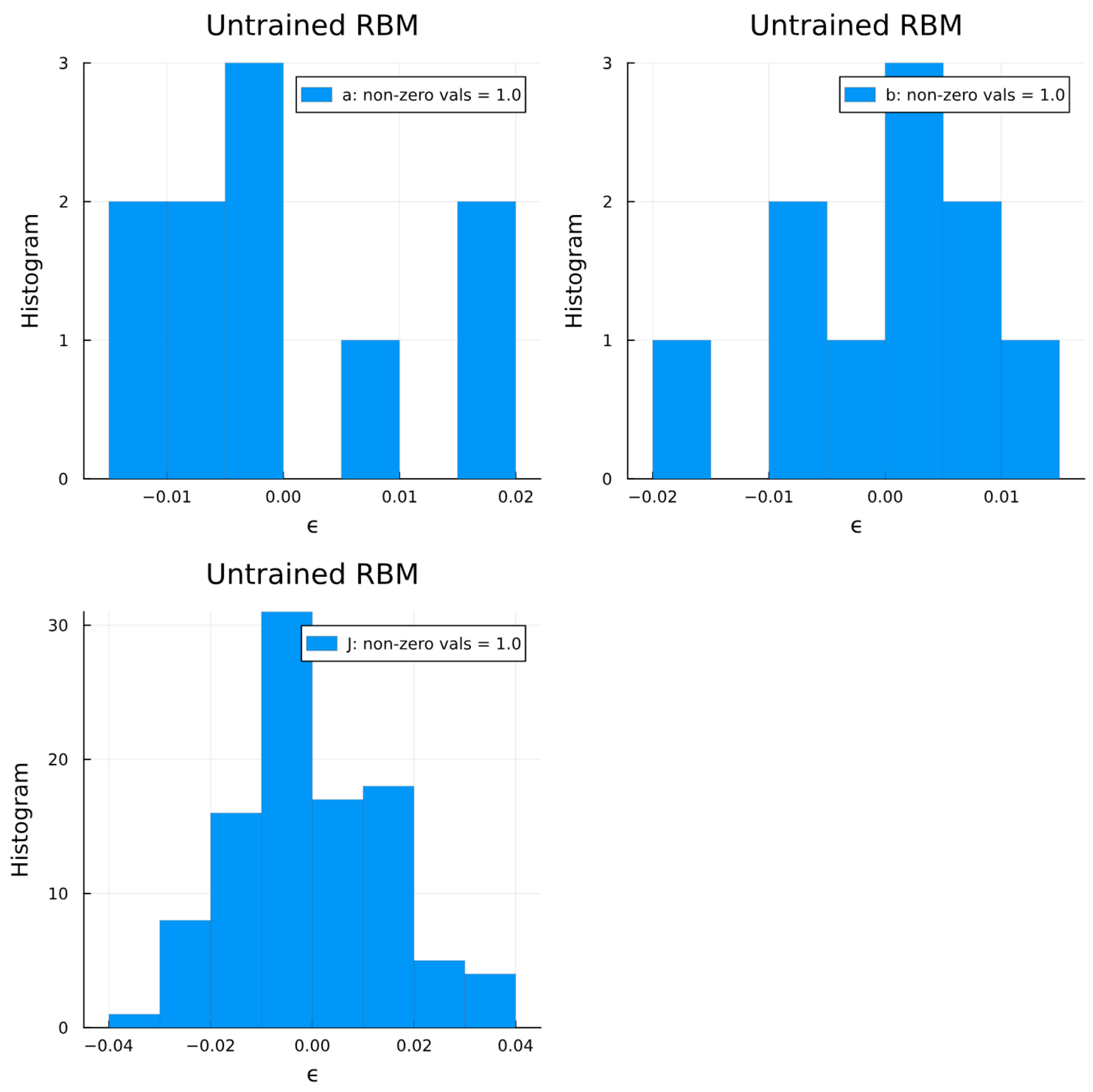
Let's assume the weights and biases are sampled from a normal N(0,0.1)

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

Exact



BGS



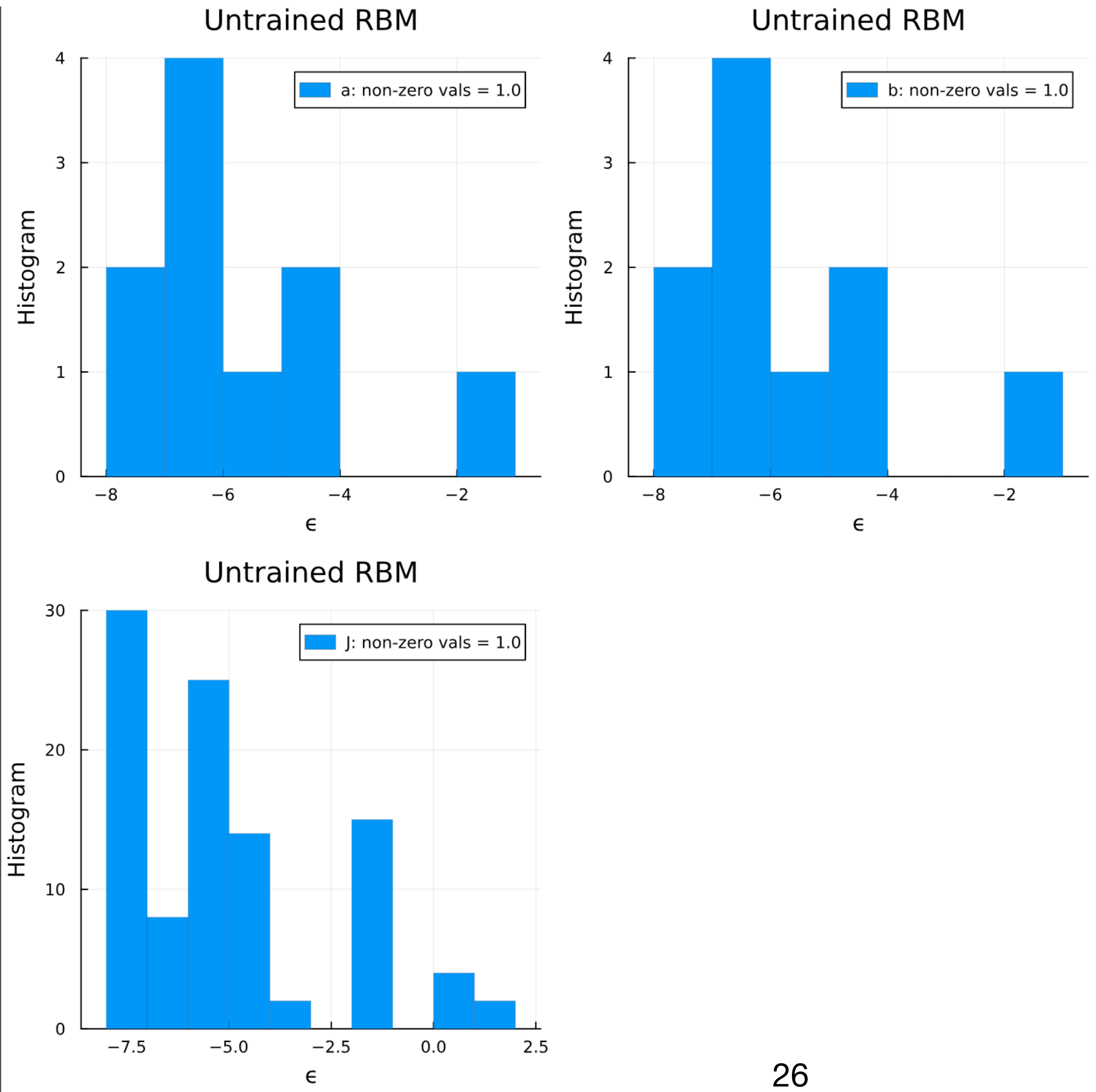
High Temperature gradient approximation

Small RBMs

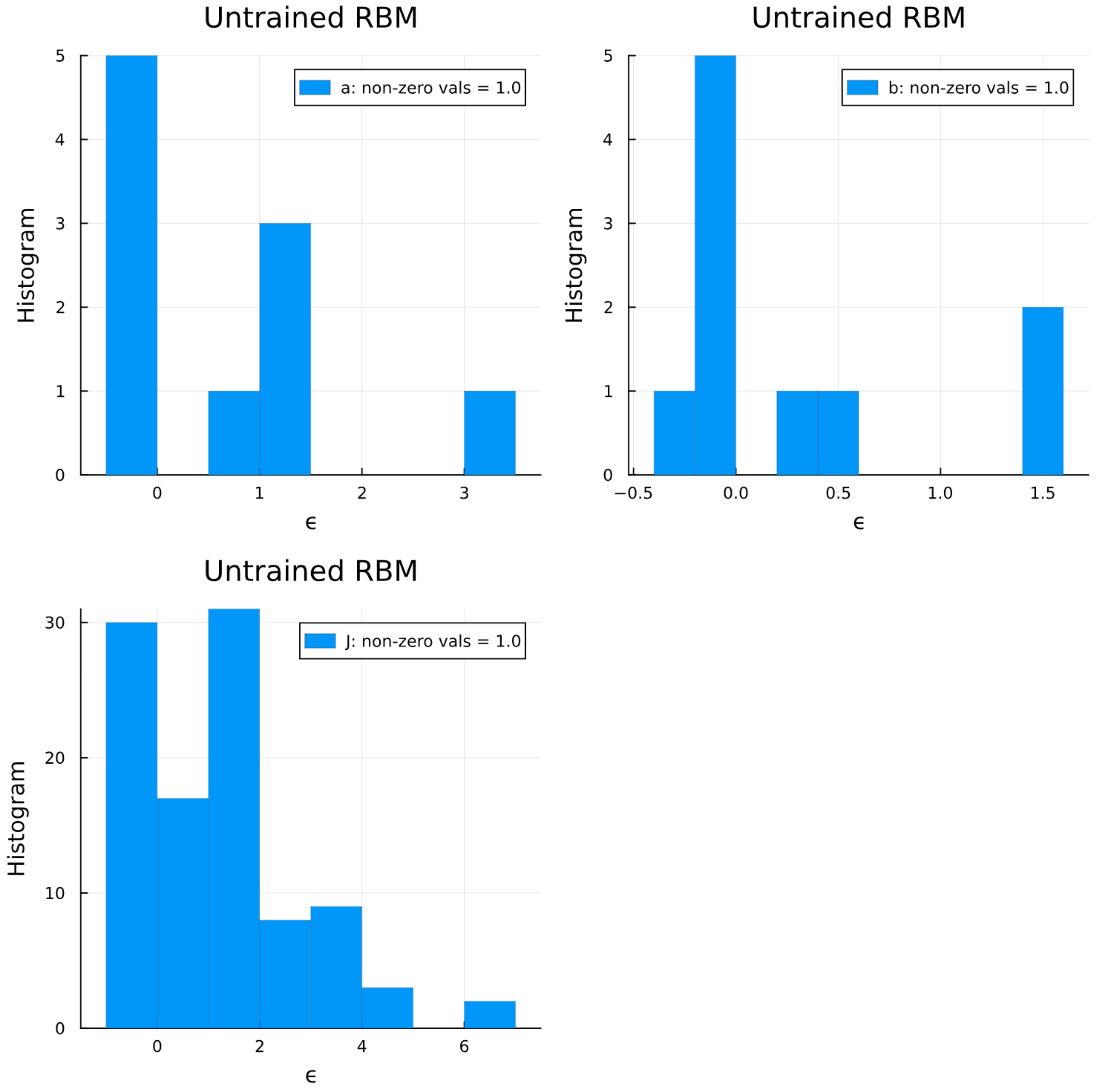
Let’s assume the weights and biases are sampled from a Uni(-1,1)

$$\epsilon = \frac{\langle E \frac{\partial E}{\partial \Theta} \rangle - \langle E \rangle \langle \frac{\partial E}{\partial \Theta} \rangle}{\langle \frac{\partial E}{\partial \Theta} \rangle}$$

Exact



BGS



High Temperature gradient approximation

Small RBMs

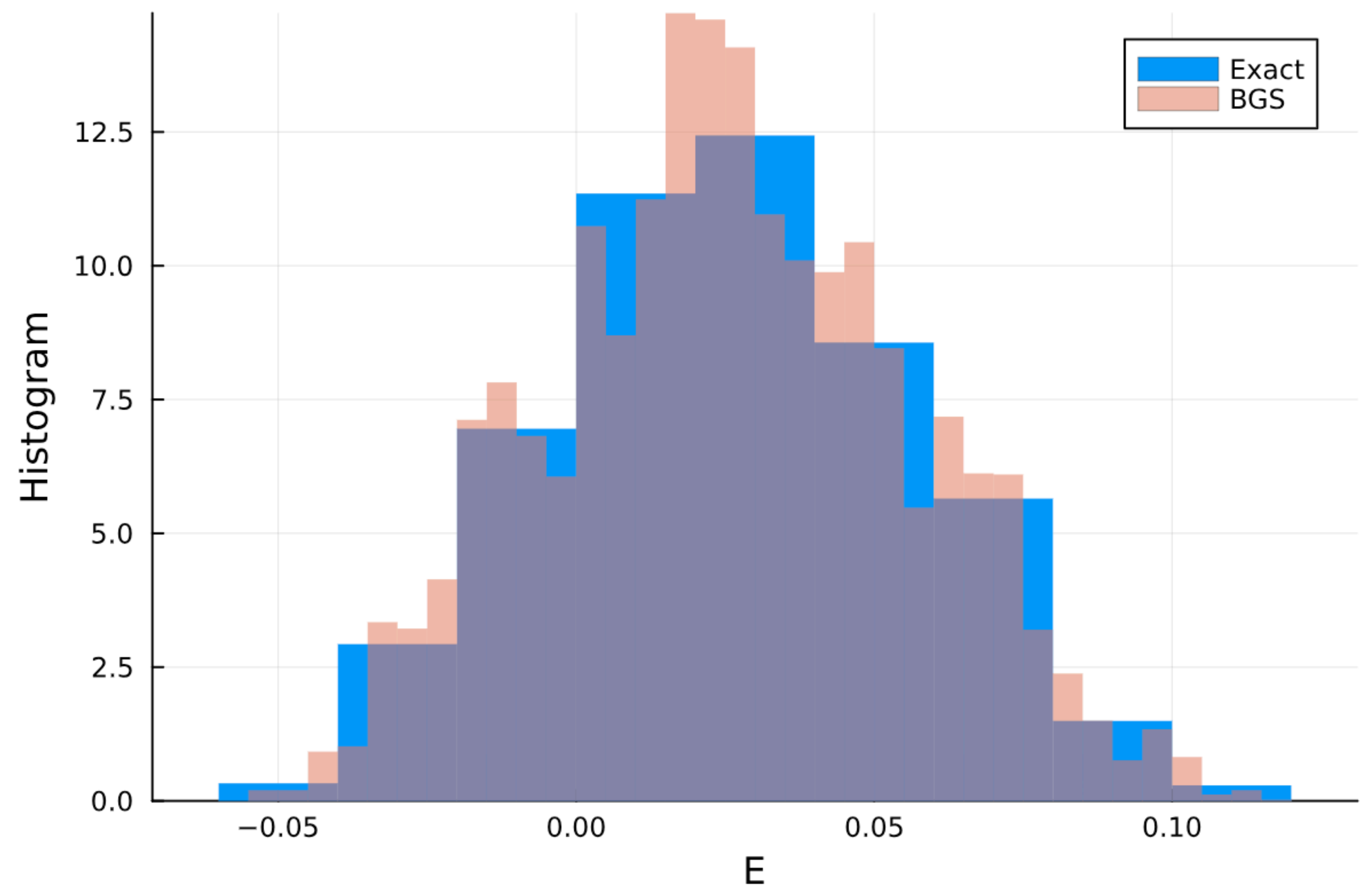
For Uni(-1,1) distributed weights and biases, the range of epsilons do not match between exact and BGS.

High Temperature gradient approximation

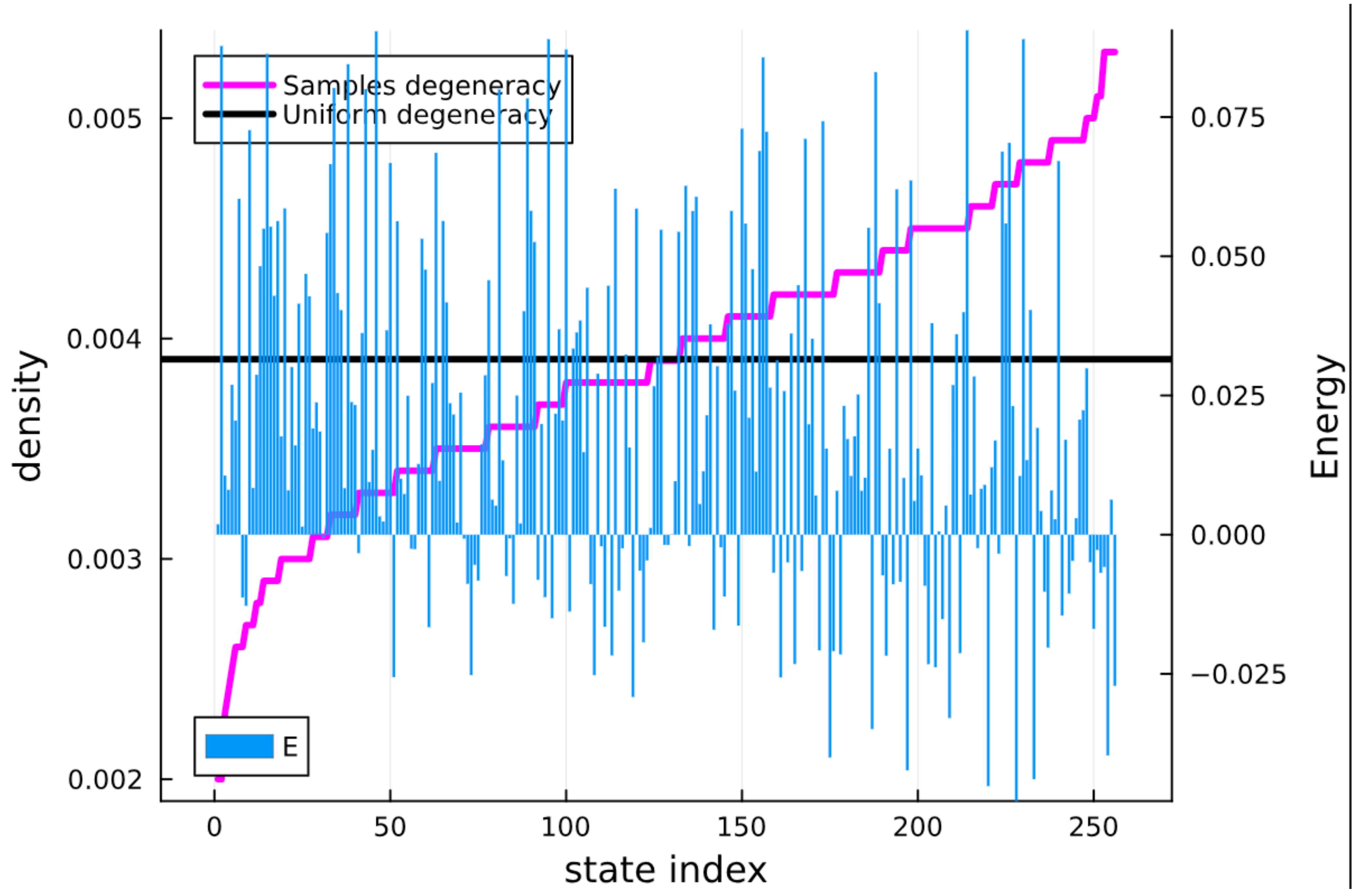
Small RBMs

Let’s look at the energy histograms

N(0,1) weights and biases



10k states sampled via BGS.
We measure the state degeneracy

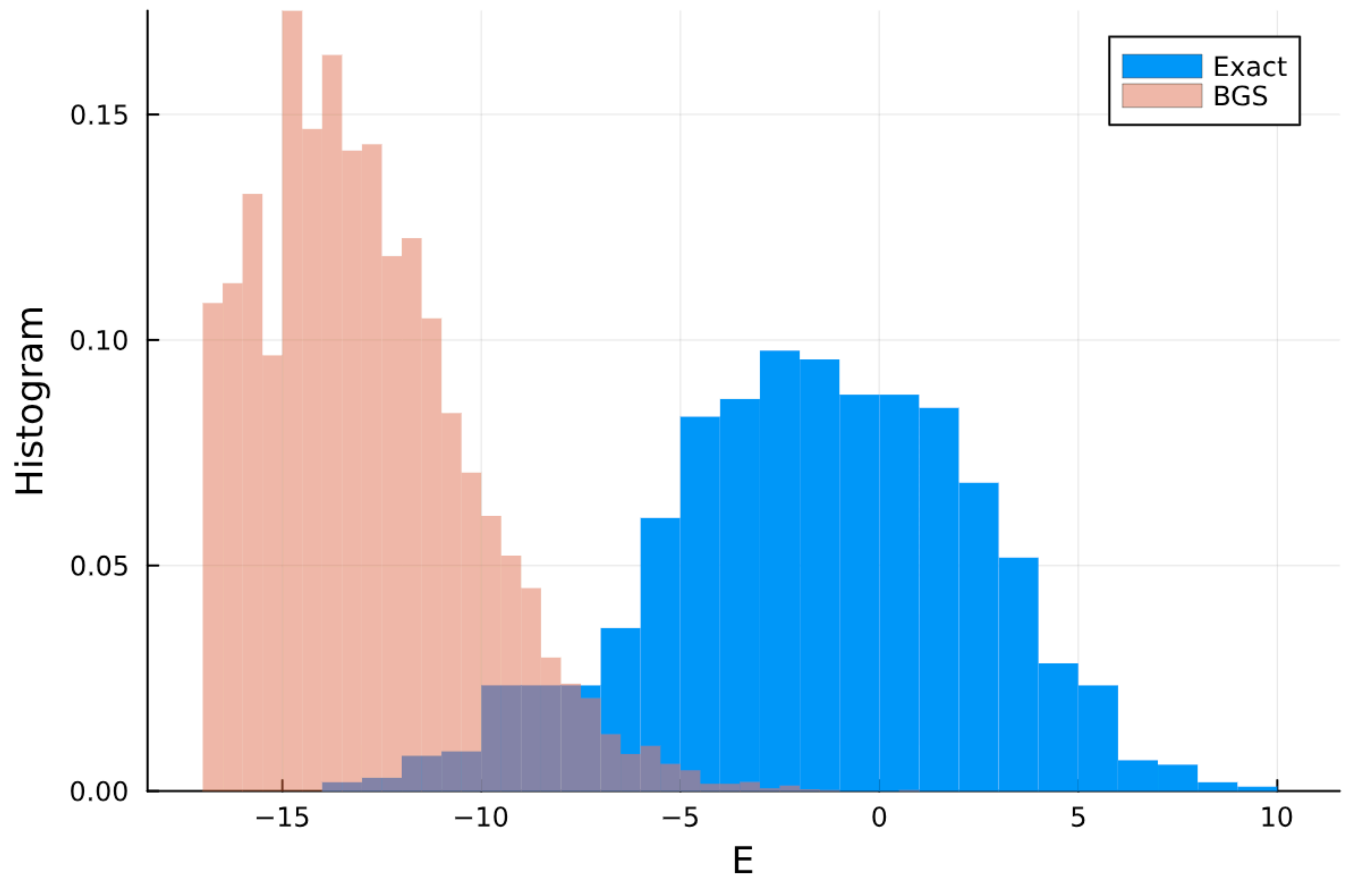


High Temperature gradient approximation

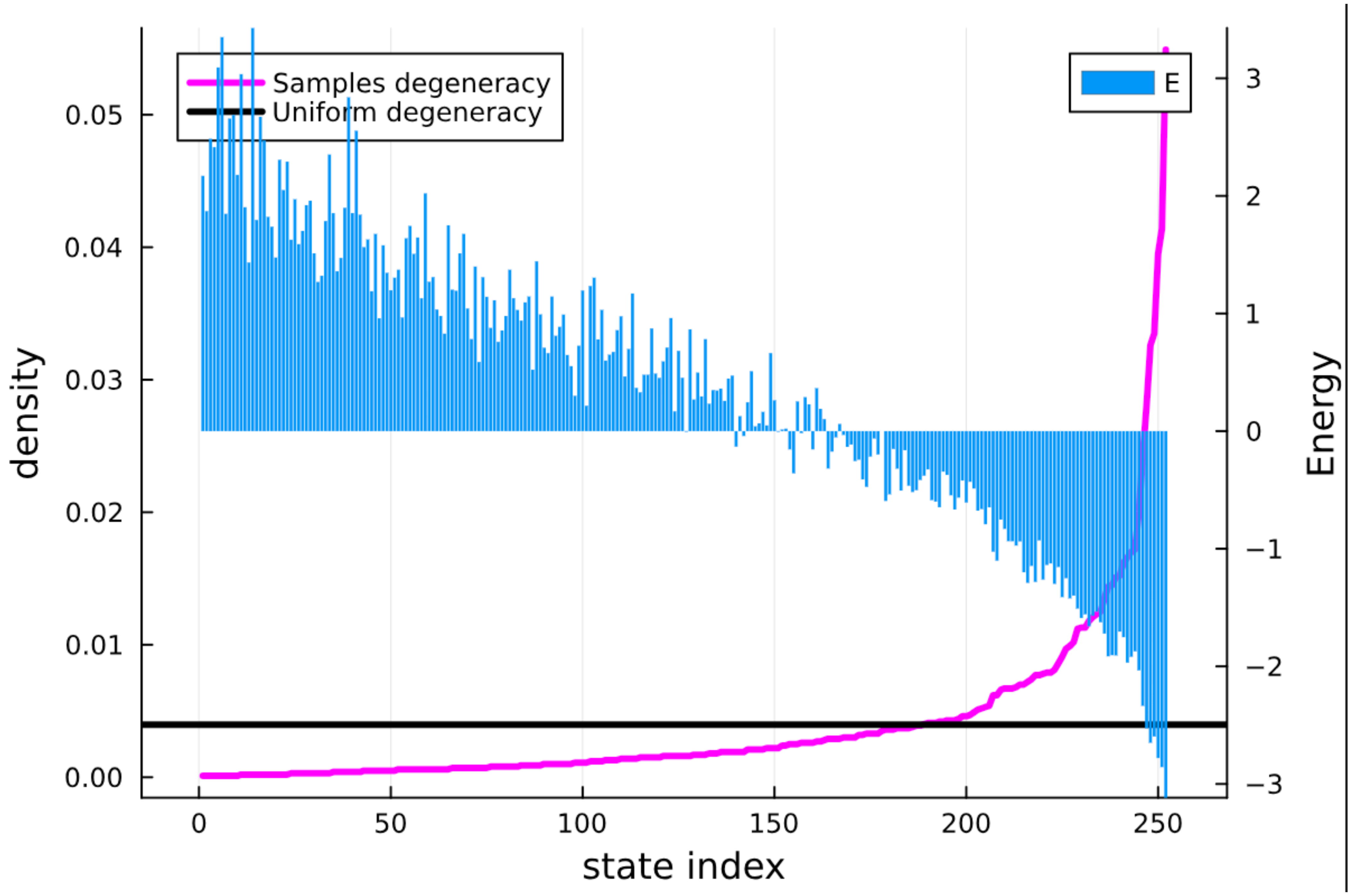
Small RBMs

Let's look at the energy histograms

U(-1,1) weights and biases



10k states sampled via BGS.
We measure the state degeneracy



High Temperature gradient approximation

Small RBMs

$$\mathcal{N}(J_0, J_1)$$

$$\tilde{J}_0 = NJ_0 \rightarrow 20 \cdot \delta$$

$$\tilde{J} = N^{1/2}J \rightarrow \sqrt{20} \cdot 1$$

$$kT = 1$$

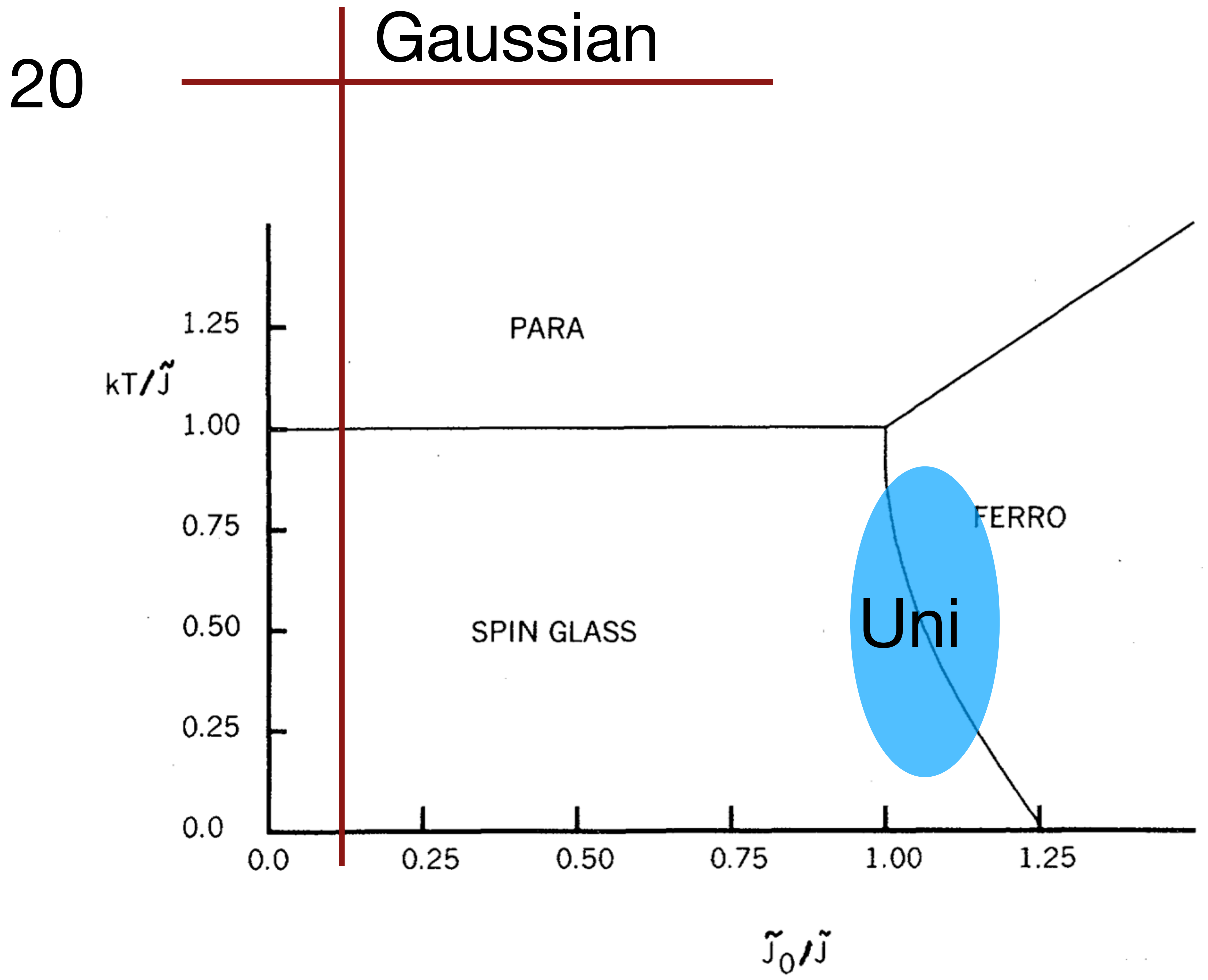


FIG. 1. Phase diagram of spin-glass ferromagnet.

Solvable Model of Spin glass, Kirkpatrick, Sherrington

After fixing the periodicity bug

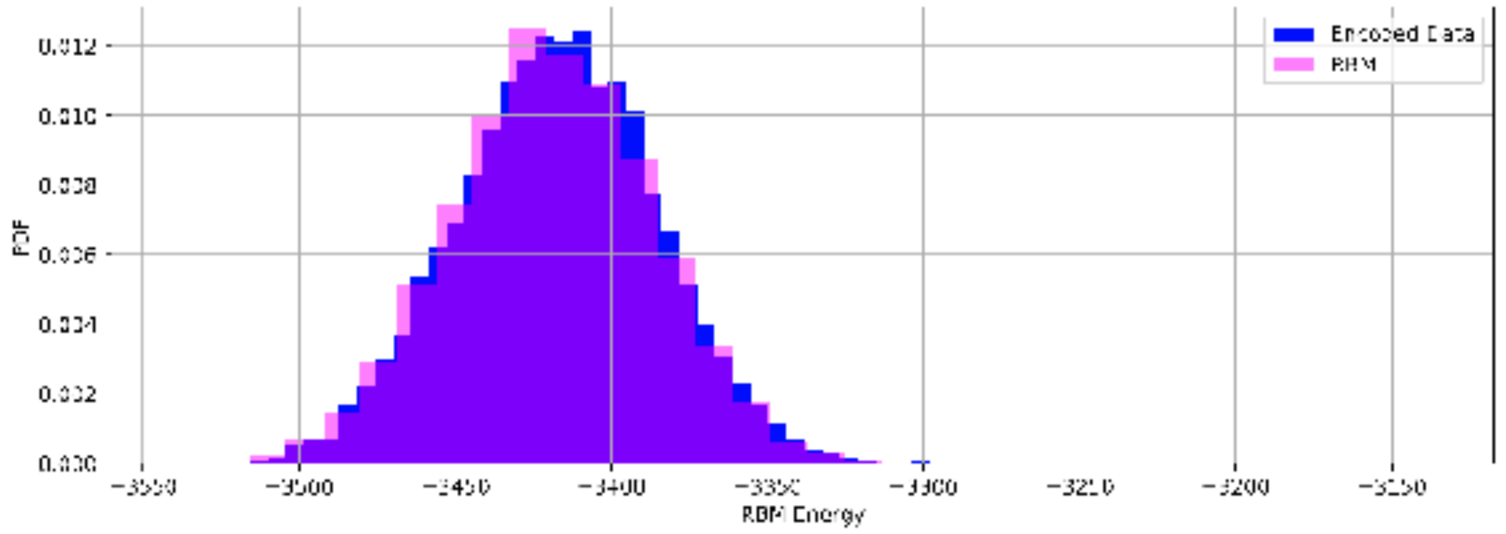
We trained several models

- ✦*Lively-grass-519* — zeph after bug fixed + non high-T approx
- ✦*Worthy-paper-518* – zeph after bug fixed + linear attention
- ✦*Devoted-lion-515* – zeph after bug fixed
- ✦*Hearty-moon-514* – zeph after bug fixed + linear attention + mask removed from activation during training
- ✦*Divine-dream-509* – Zeph after bug fixed + mask removed from activation during training

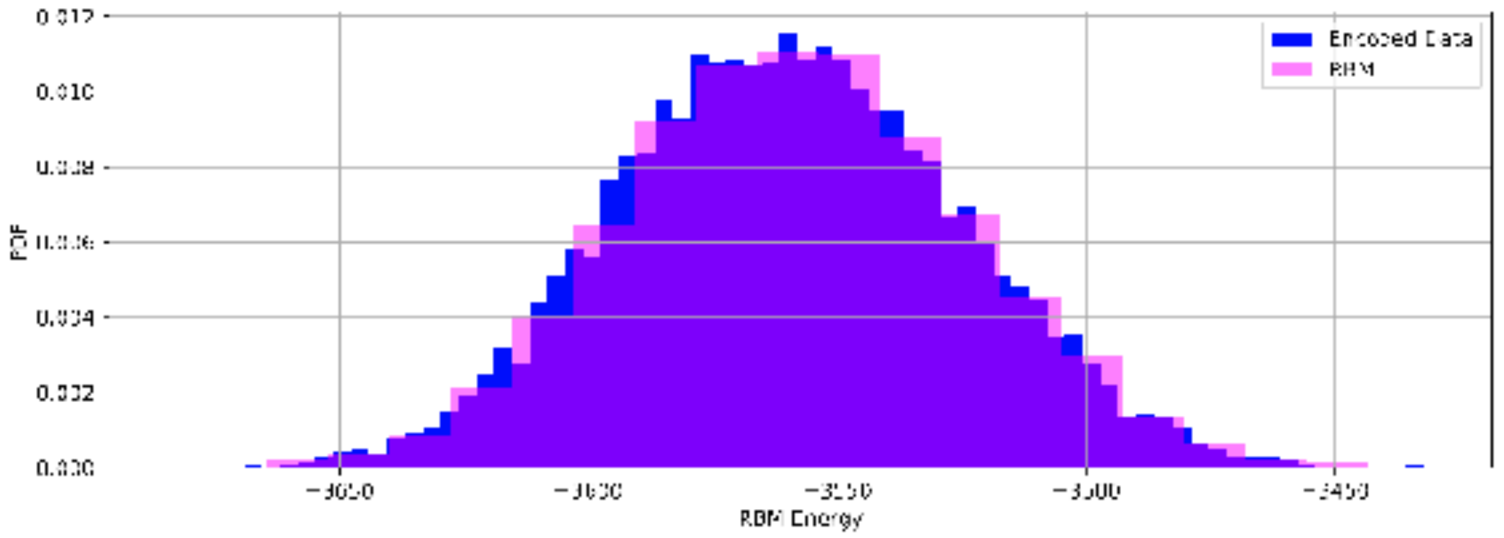
After fixing the periodicity bug

We trained several models

lively-grass-519

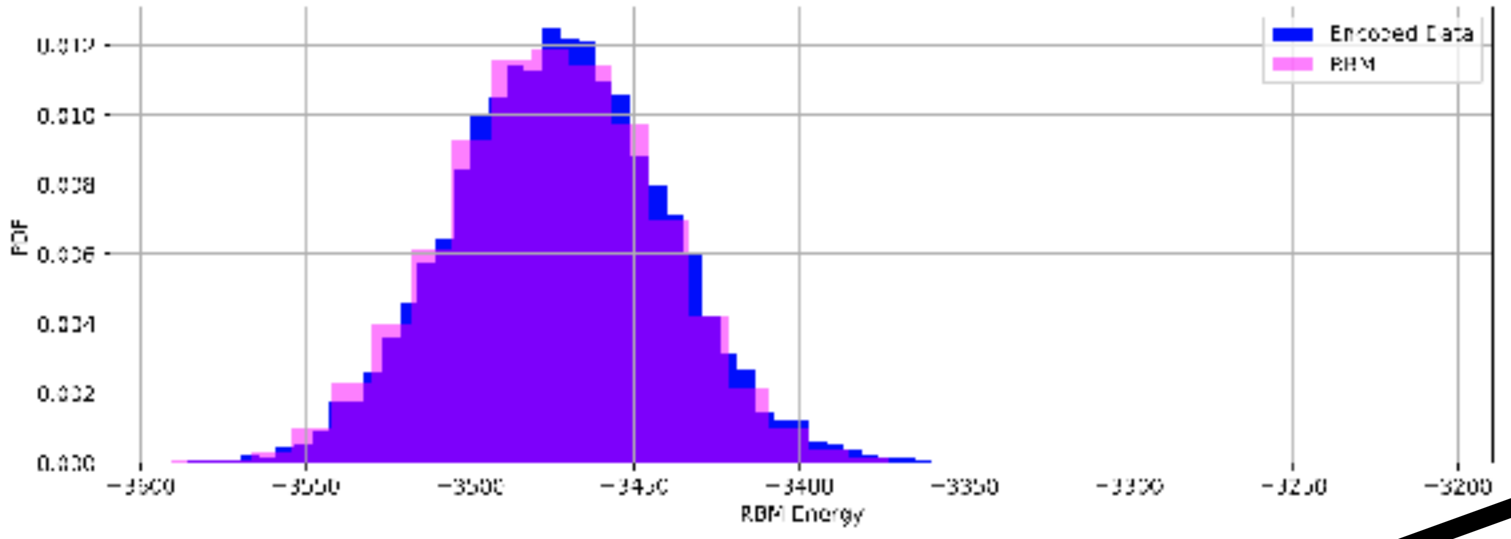


worthy-paper-518

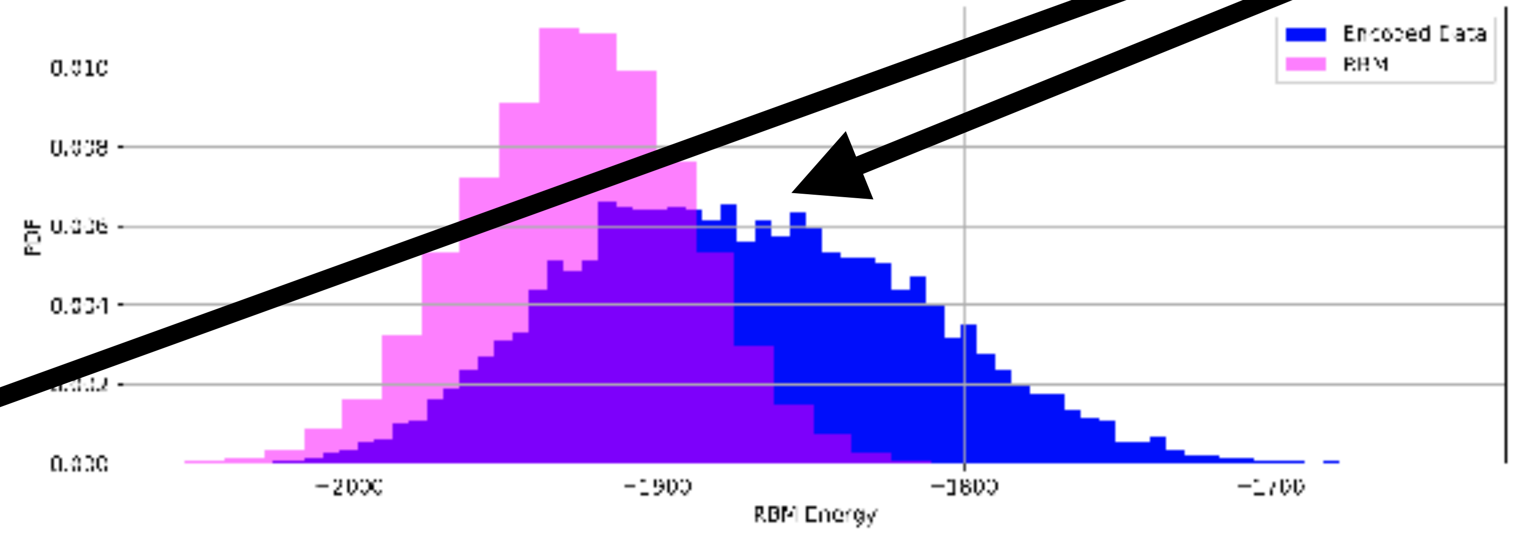


Mask removed from activation during training

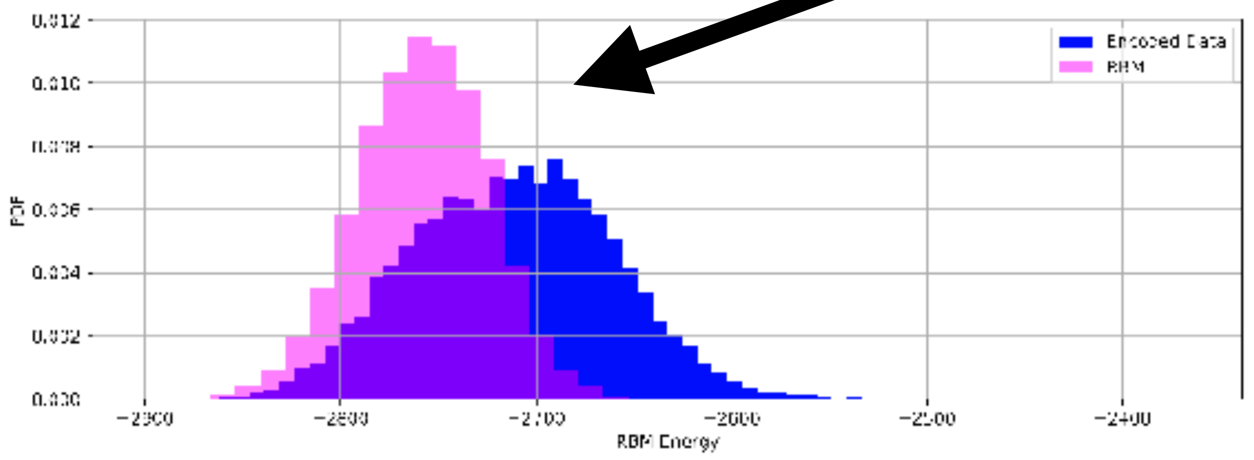
devoted-lion-515



hearty-moon-514

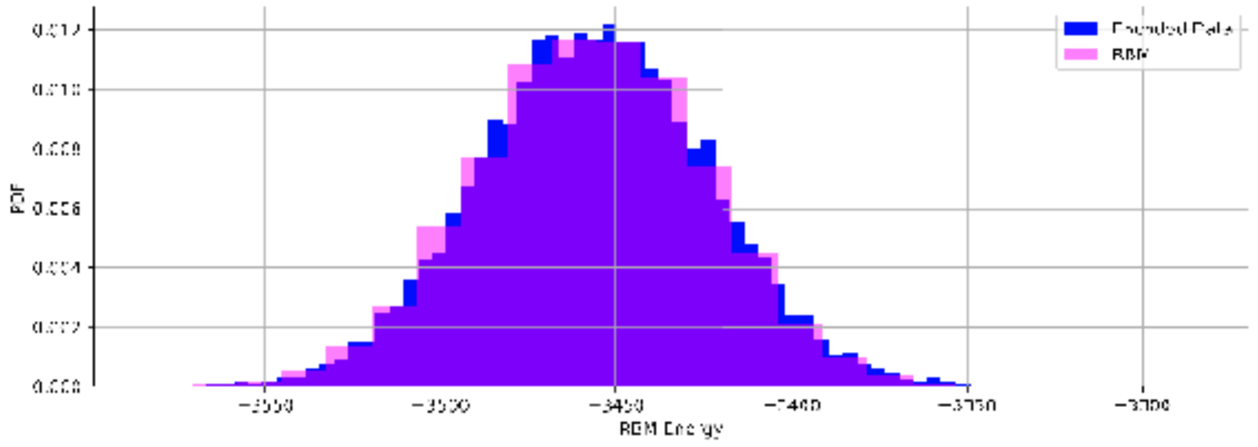


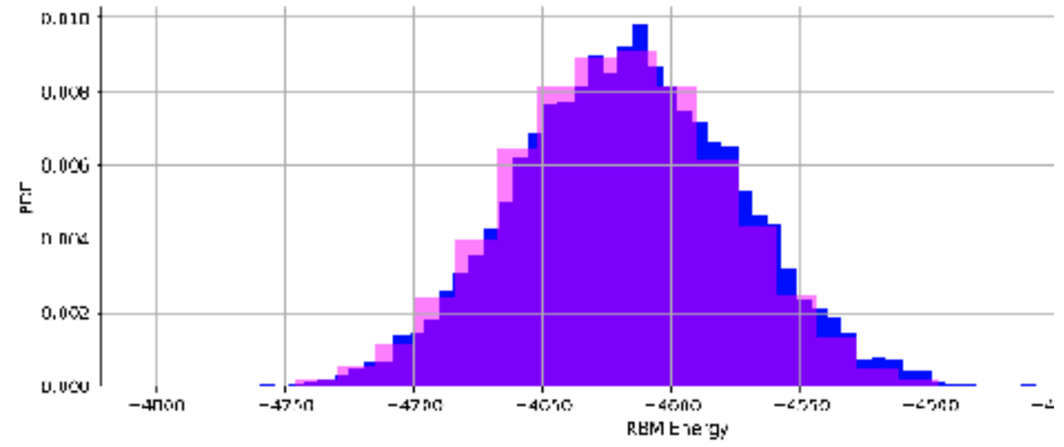
divine-dream-509



Peg and Zeph before fix

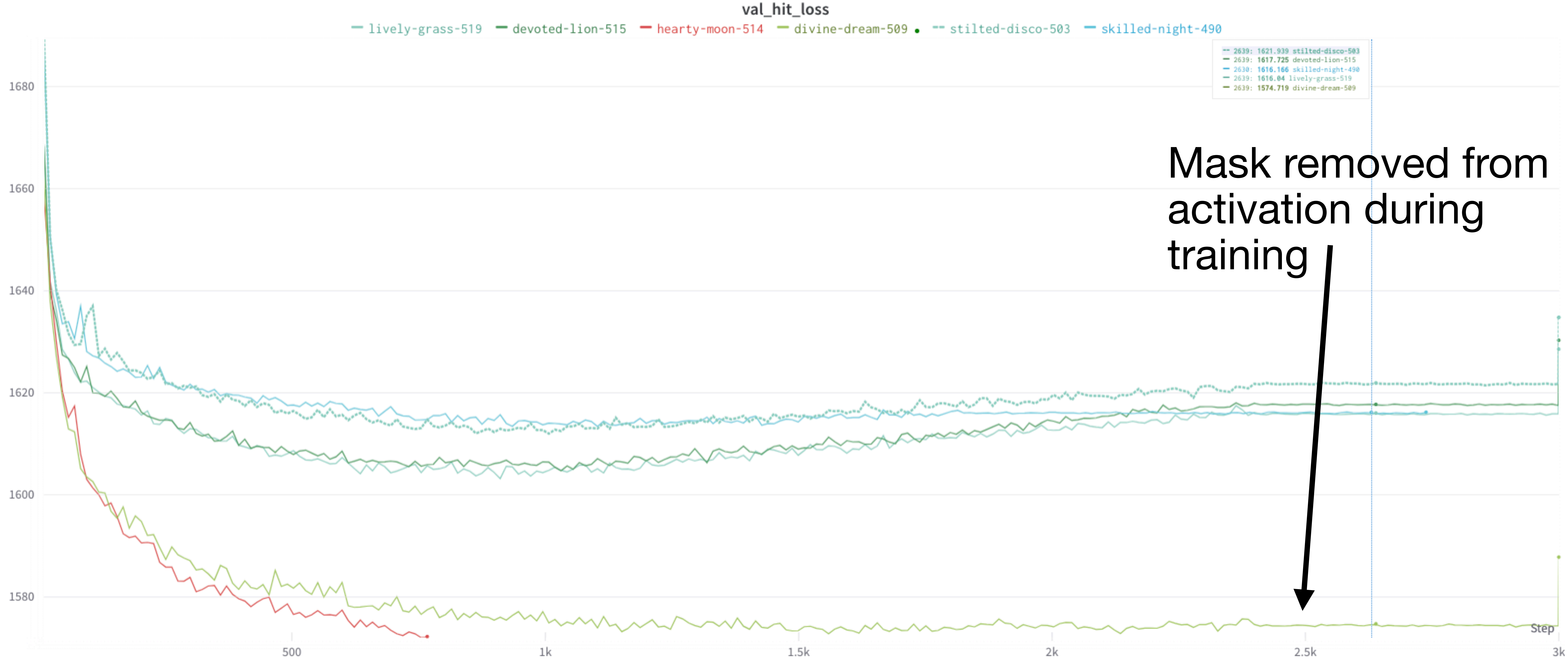
stilted-disco-503





After fixing the periodicity bug

We trained several models



After fixing the periodicity bug

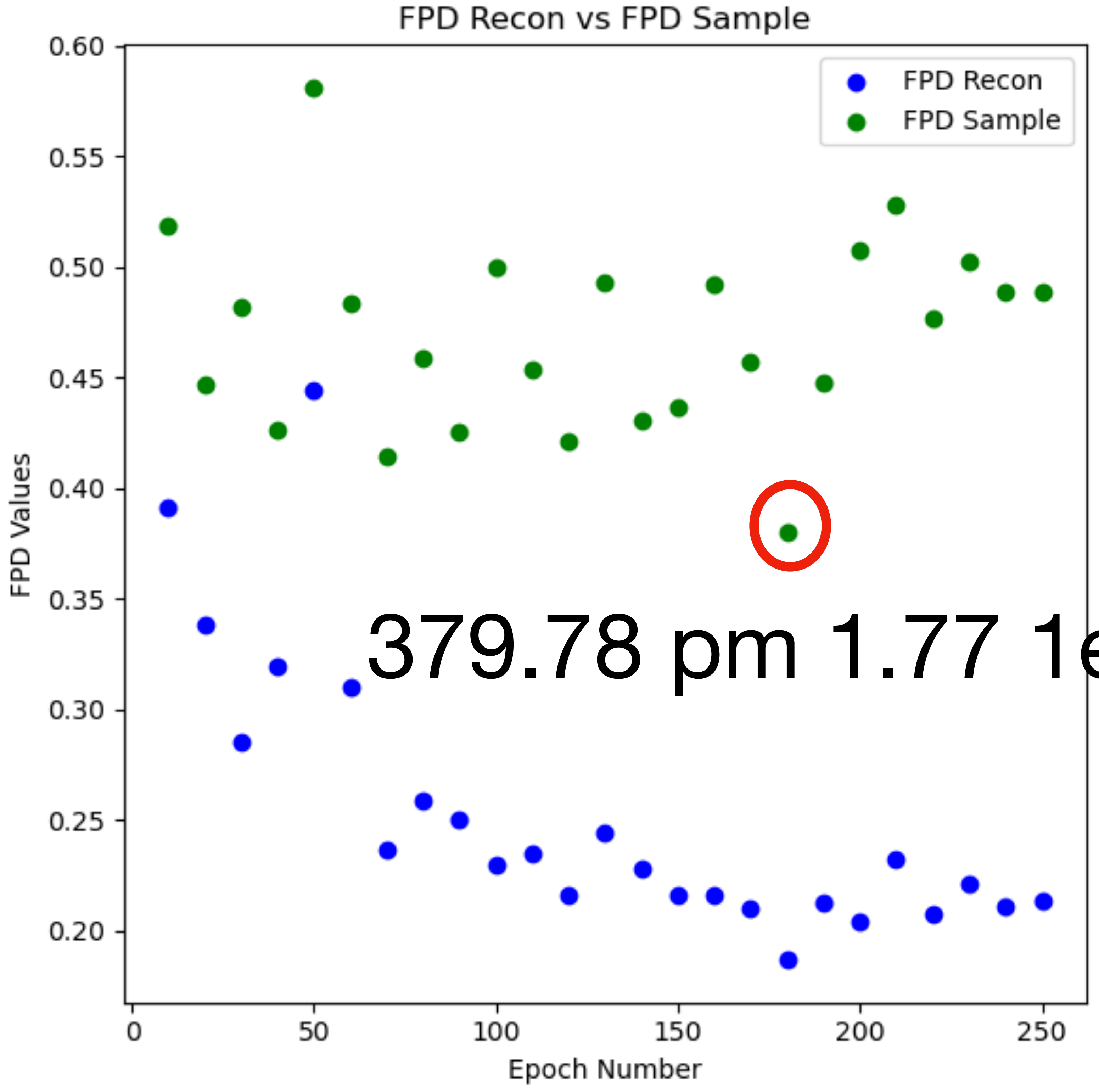
We trained several models

- ✦*Lively-grass-519* — zeph after bug fixed + non high-T approx
- ✦*Worthy-paper-518* – zeph after bug fixed + linear attention
- ✦*Devoted-lion-515* – zeph after bug fixed
- ✦~~*Hearty-moon-514* – zeph after bug fixed + linear attention + mask removed from activation during training **_=> Bad RBM_**~~
- ✦*Divine-dream-509* – Zeph after bug fixed + mask removed from activation during training **_=> Bad RBM_**

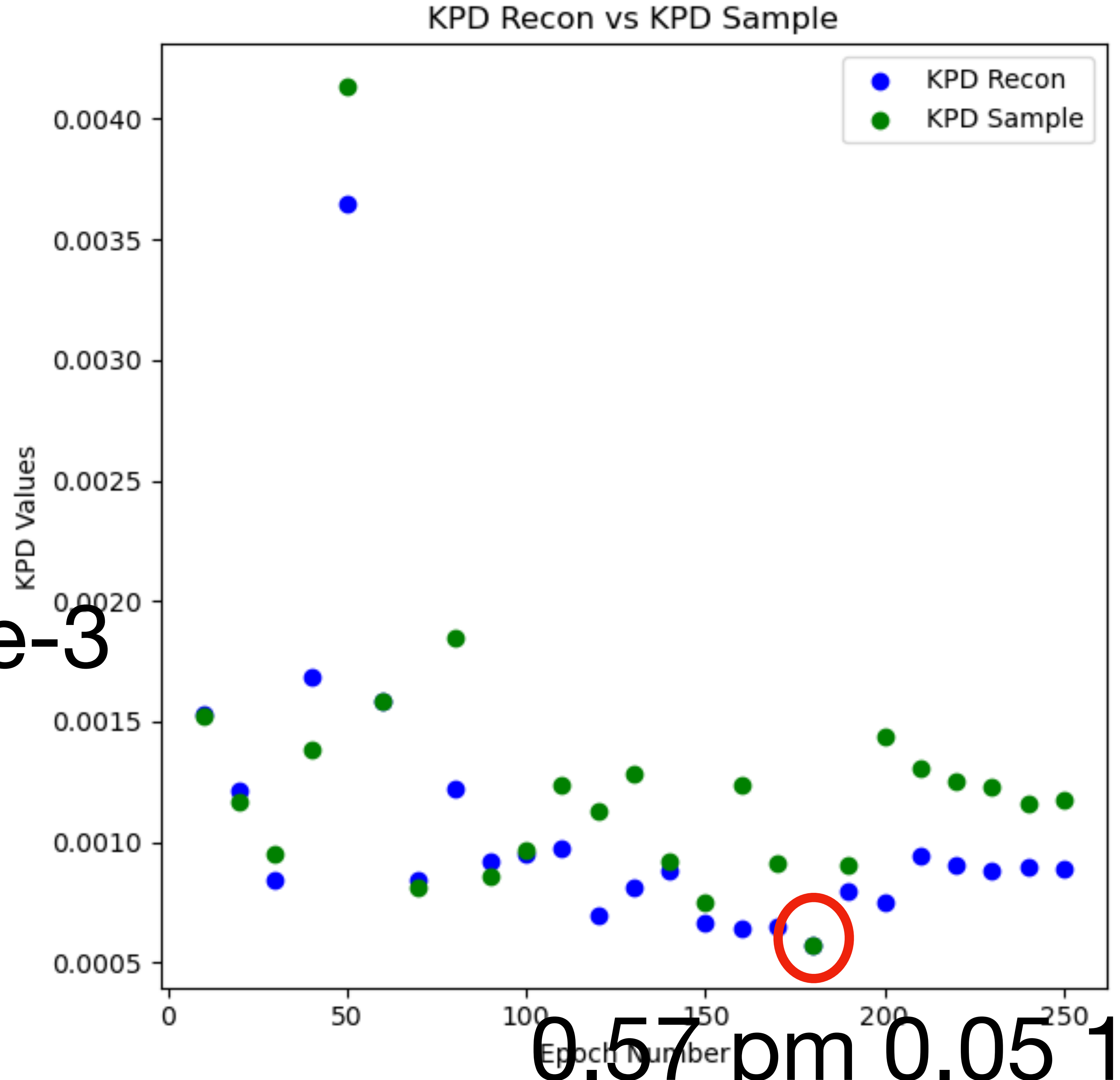
After fixing the periodicity bug

We trained several models

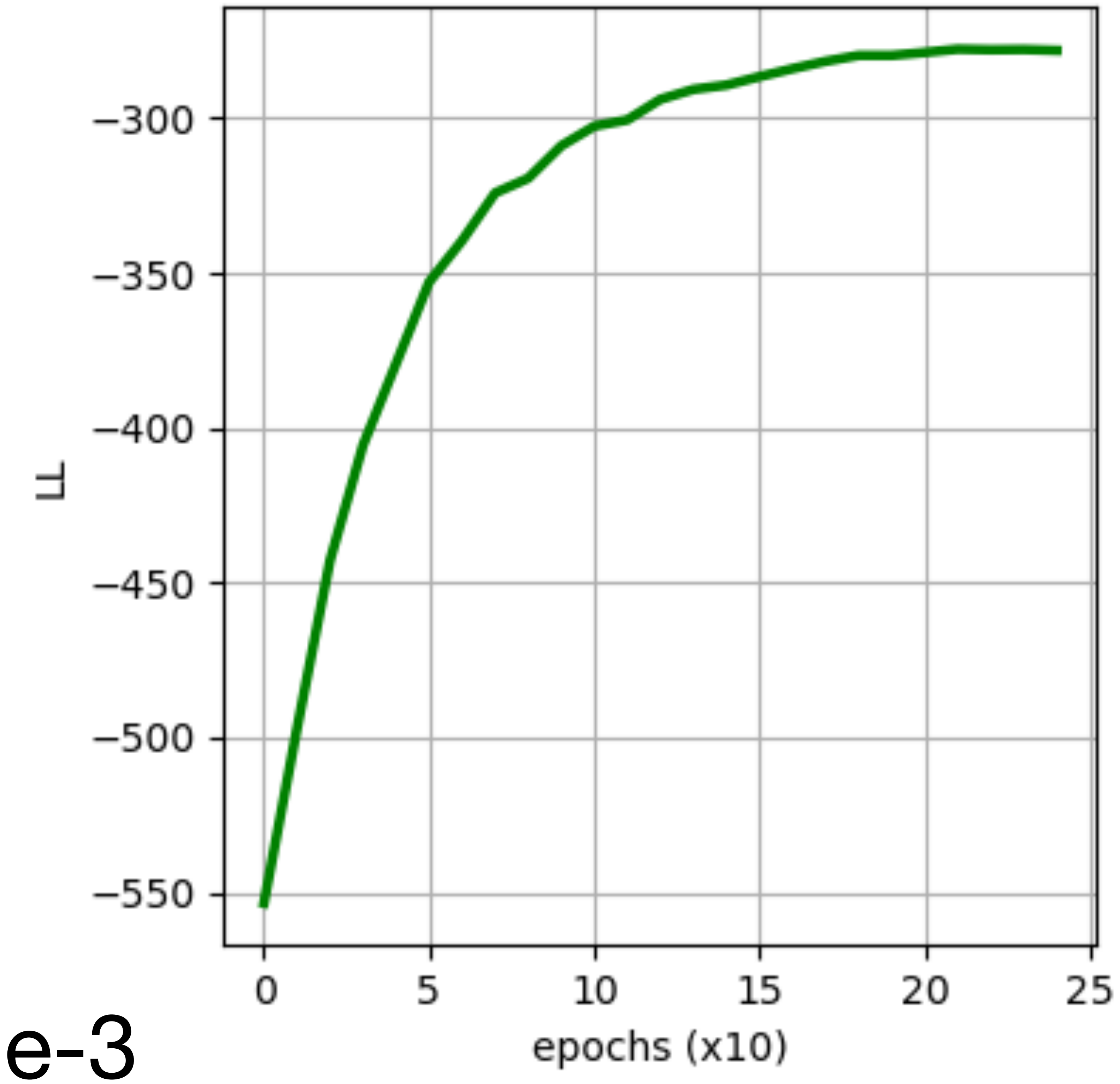
✦ *Devoted-lion-515* – zeph after bug fixed



379.78 pm 1.77 1e-3



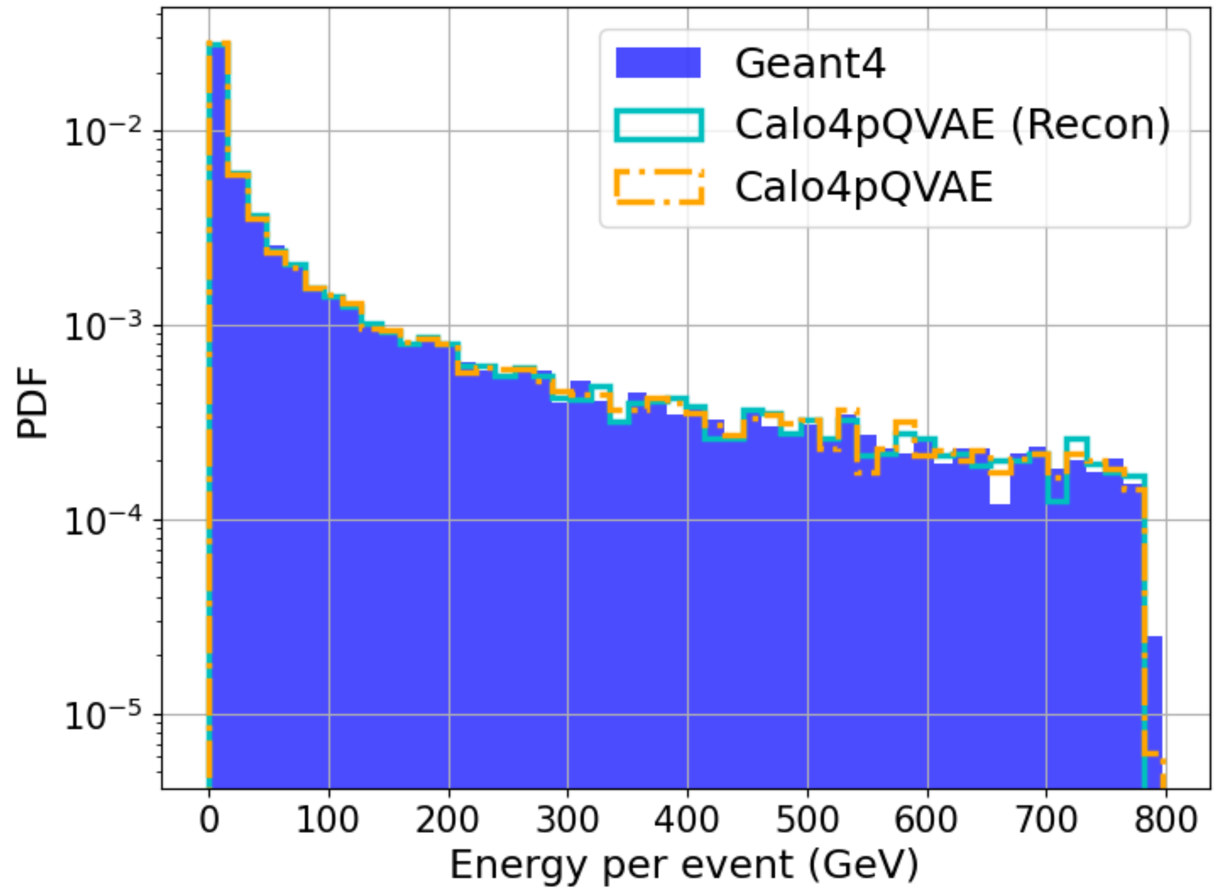
0.57 pm 0.05 1e-3

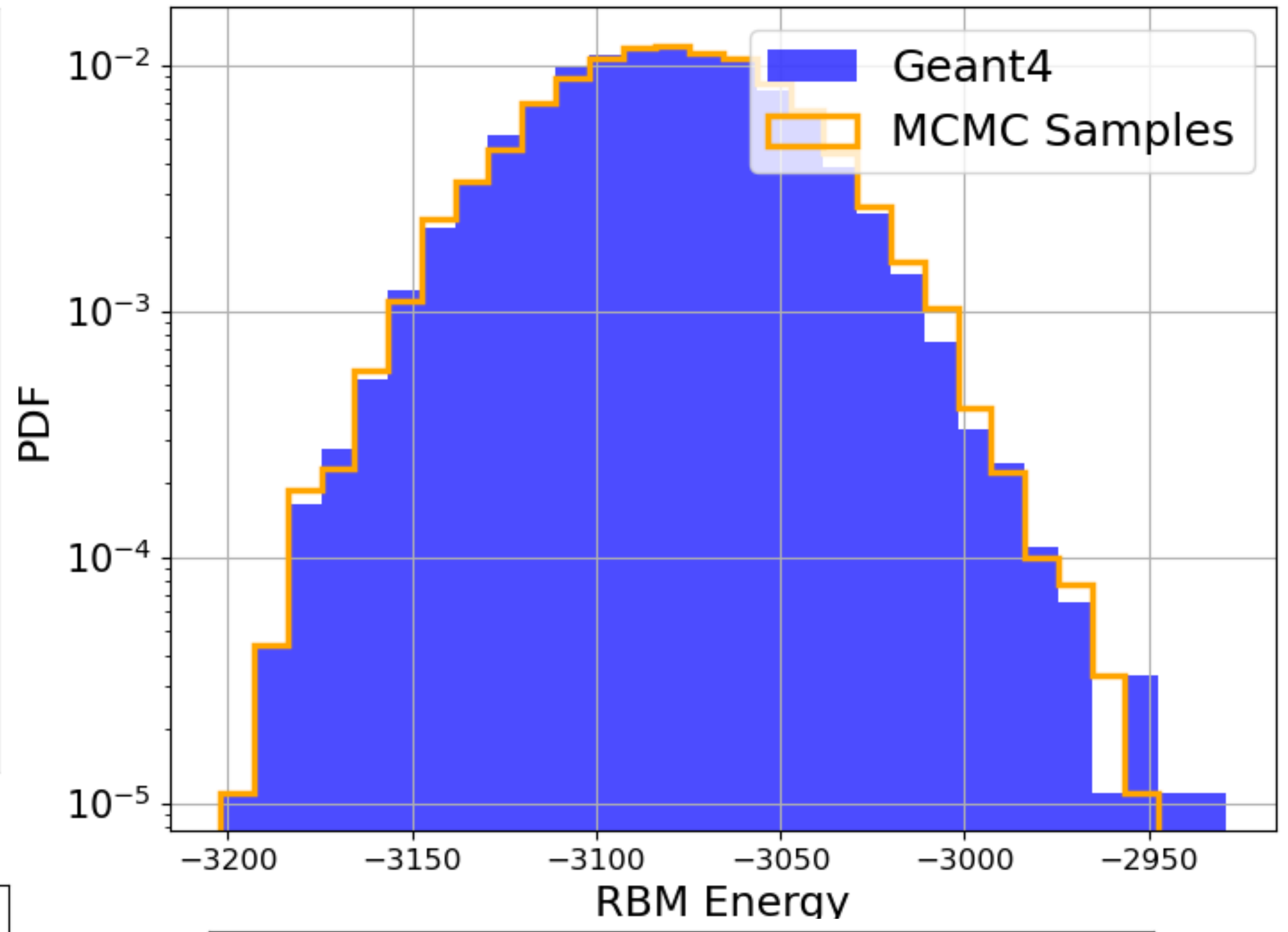


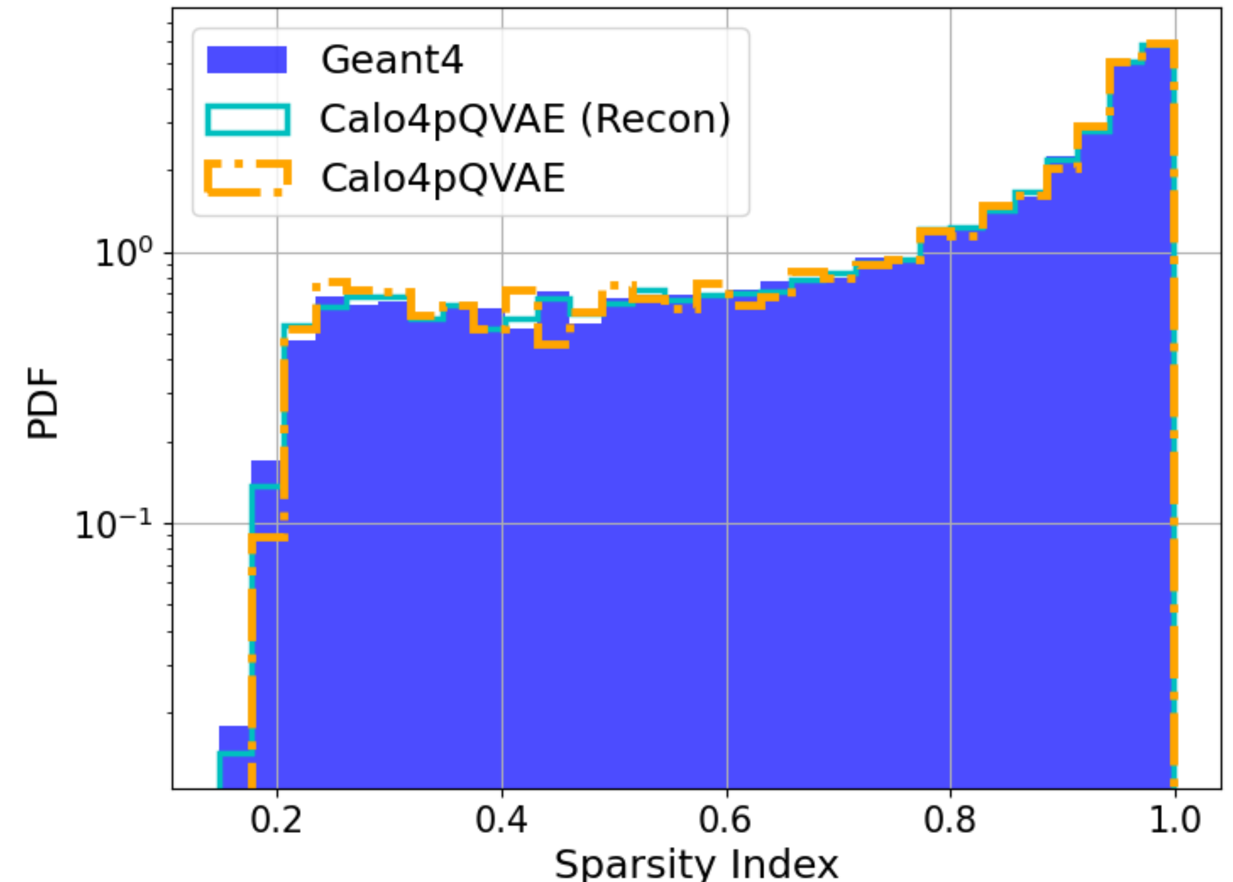
After fixing the periodicity bug

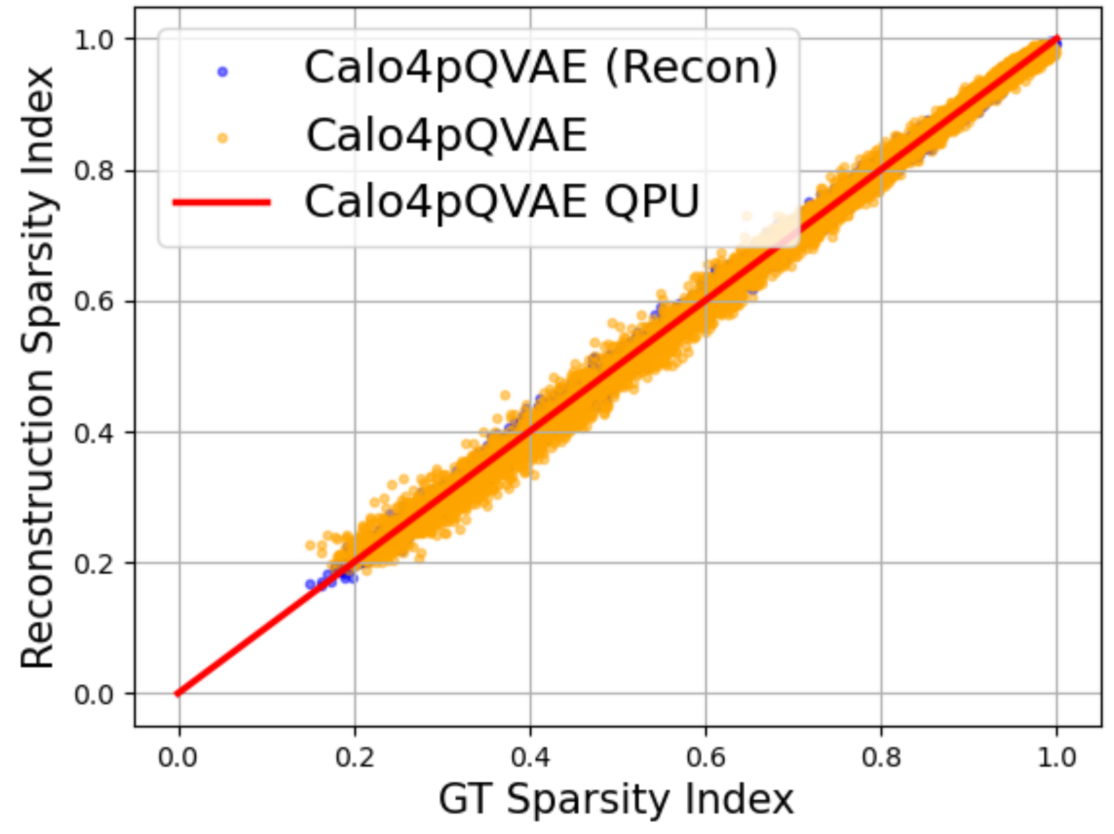
We trained several models

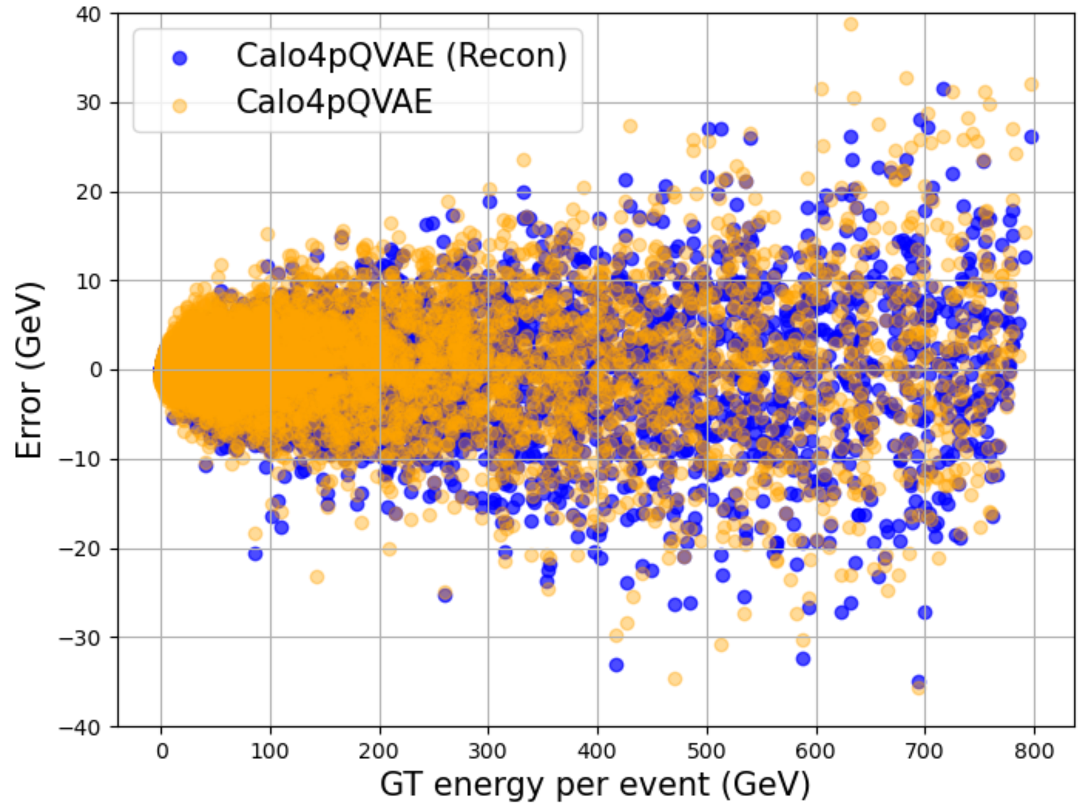
✦ *Devoted-lion-515* – zeph after bug fixed

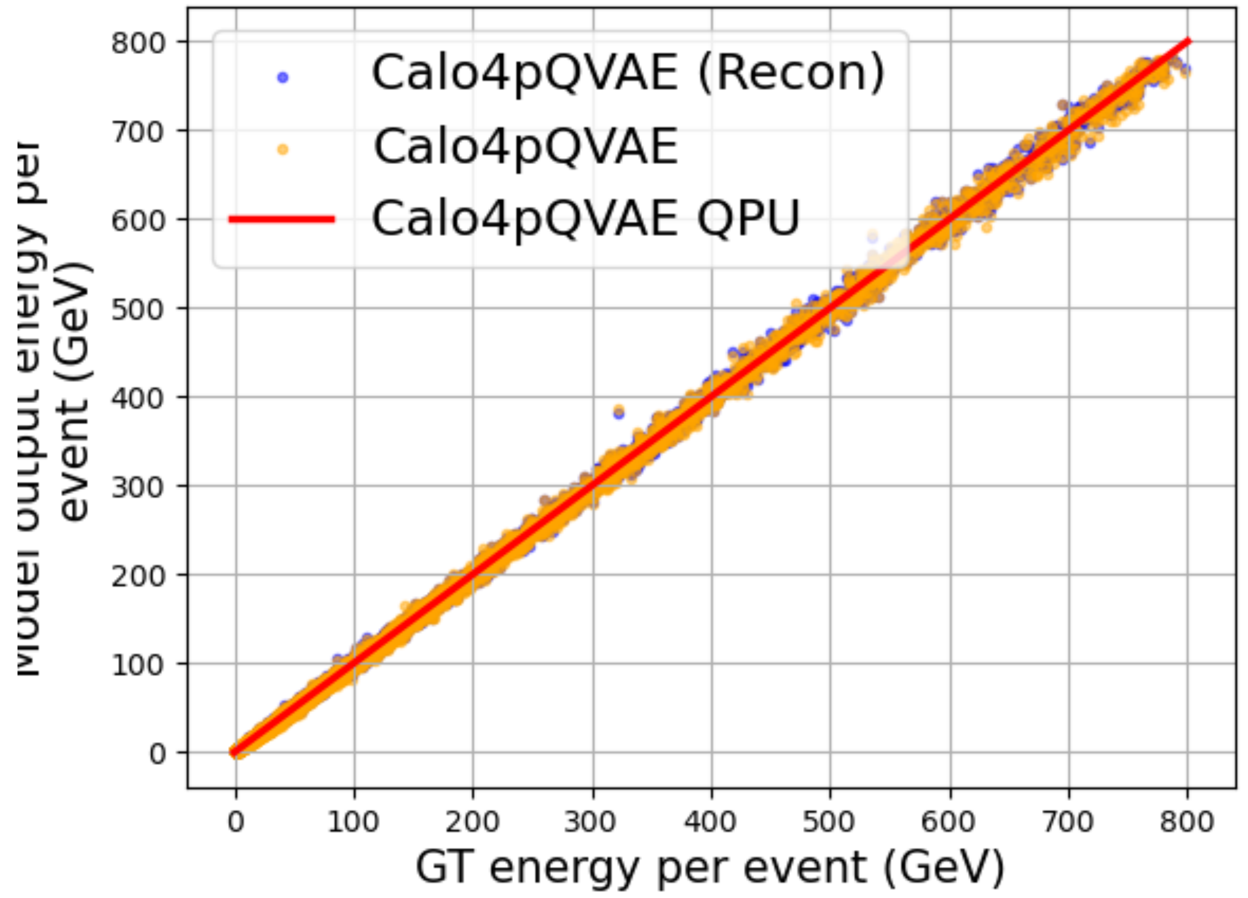


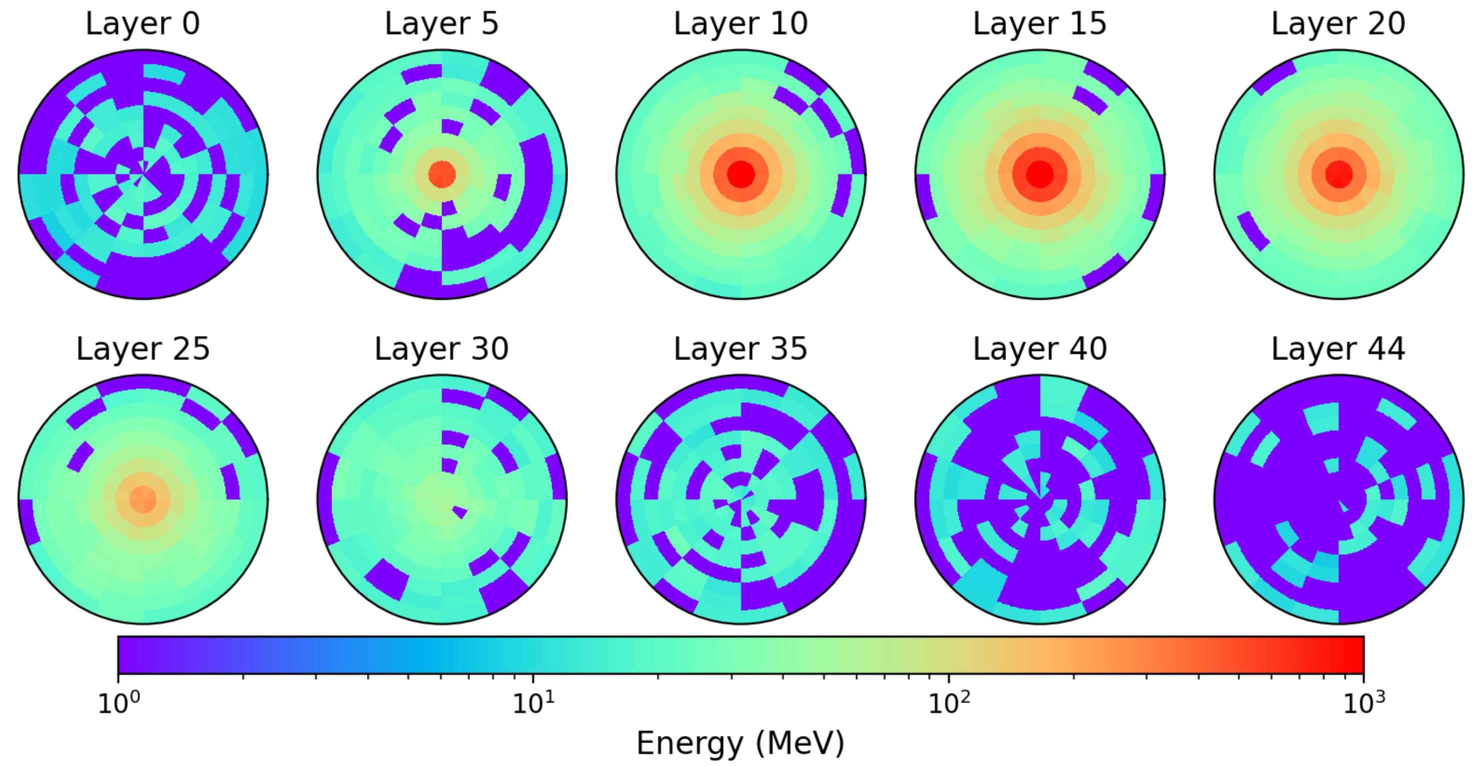


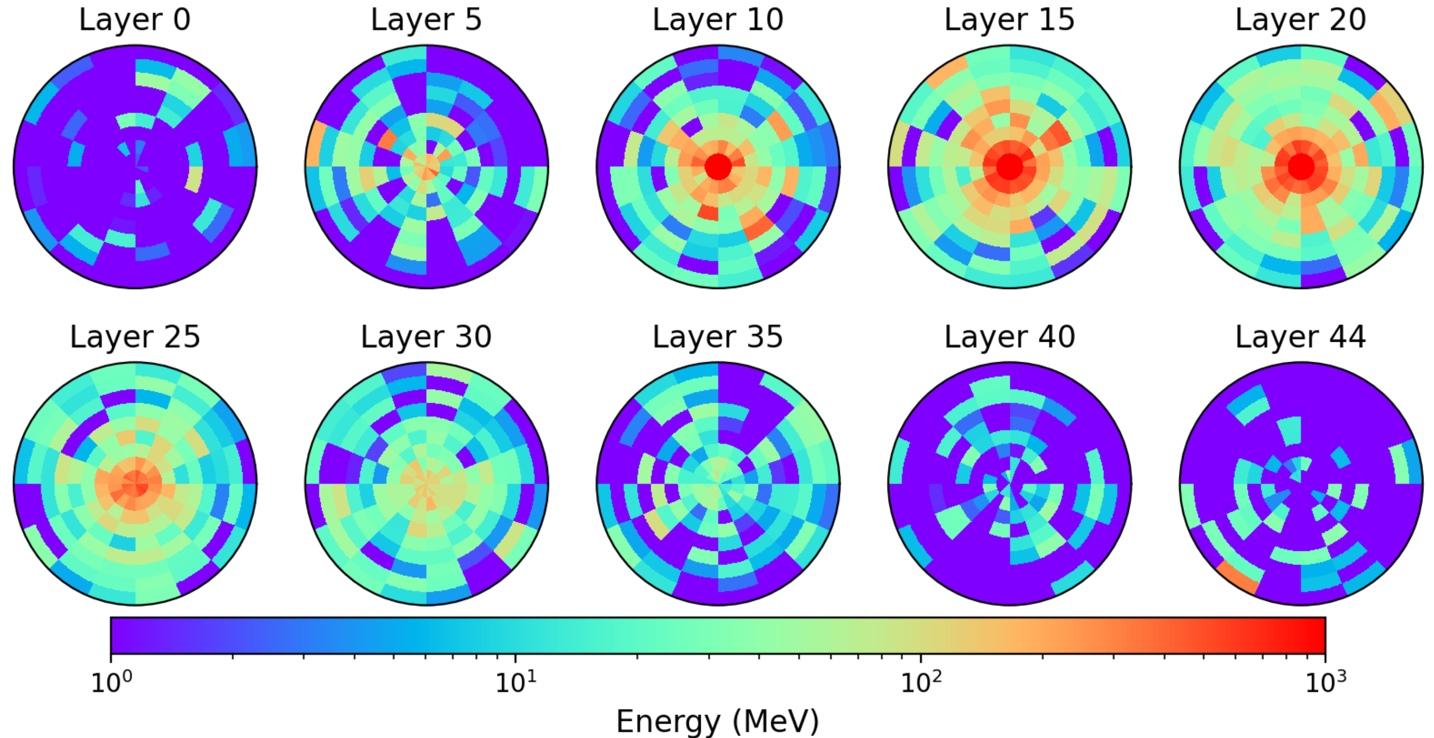








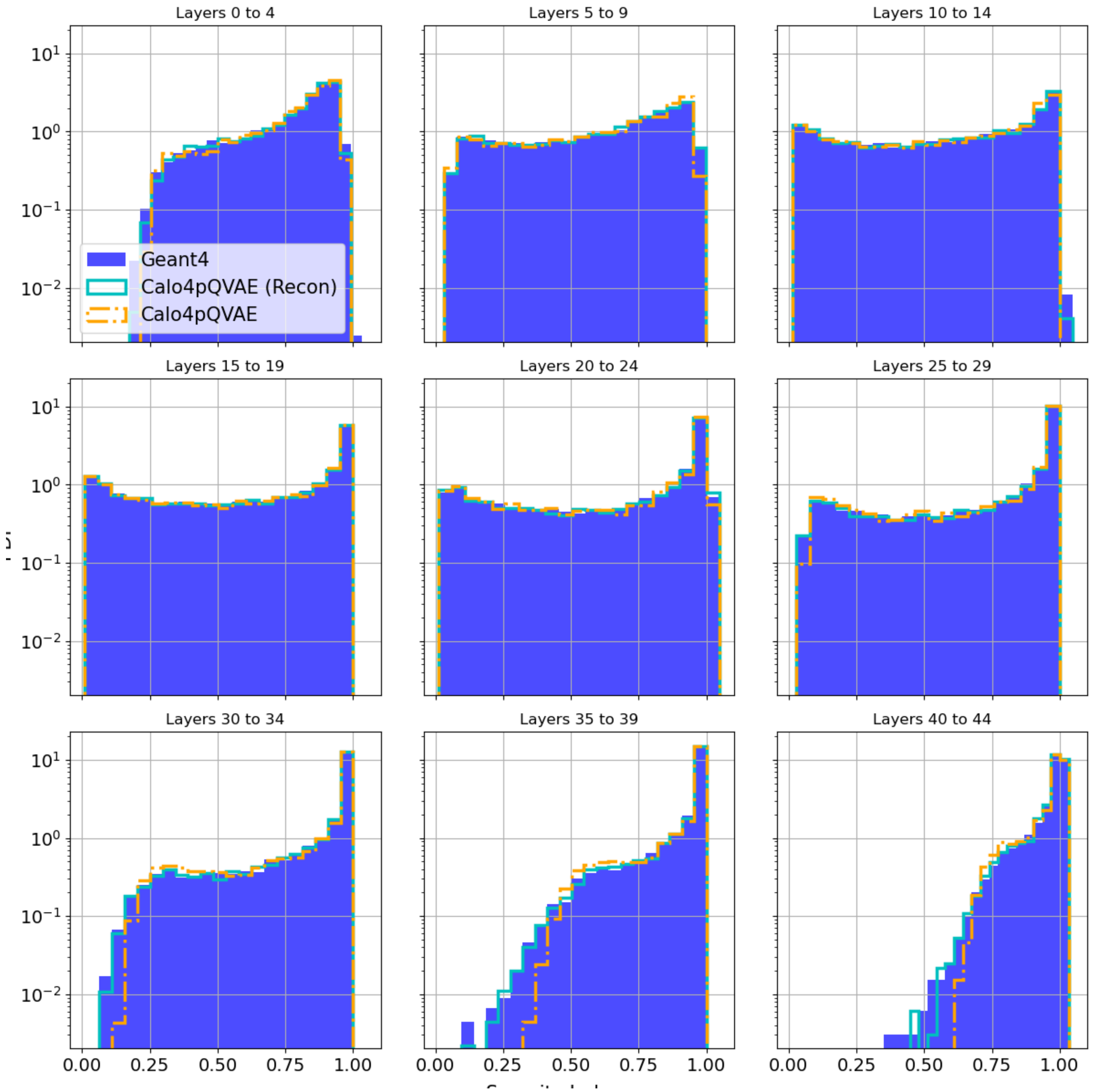


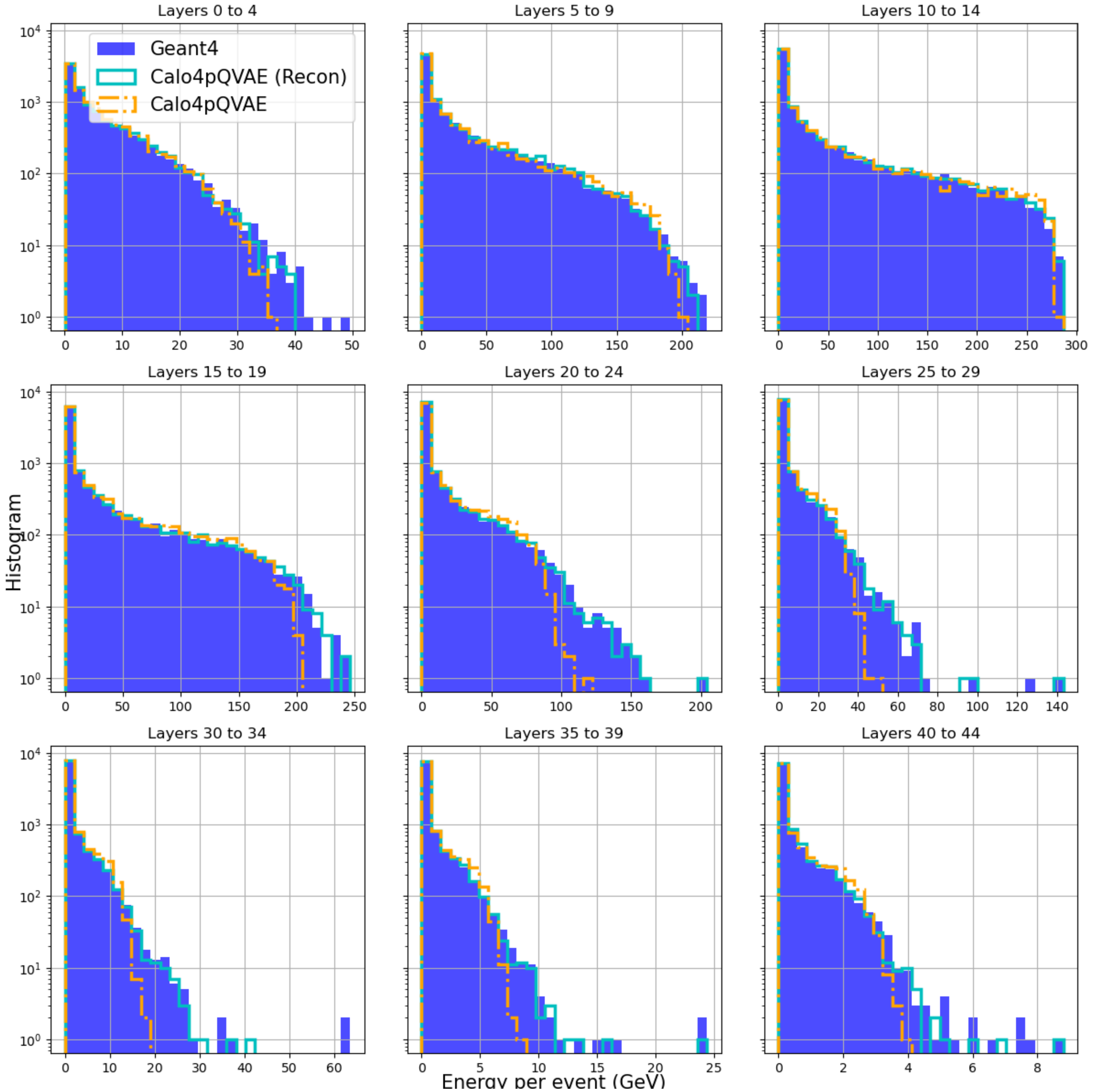


After fixing the periodicity bug

We trained several models

- *Devoted-lion-515* – zeph after bug fixed

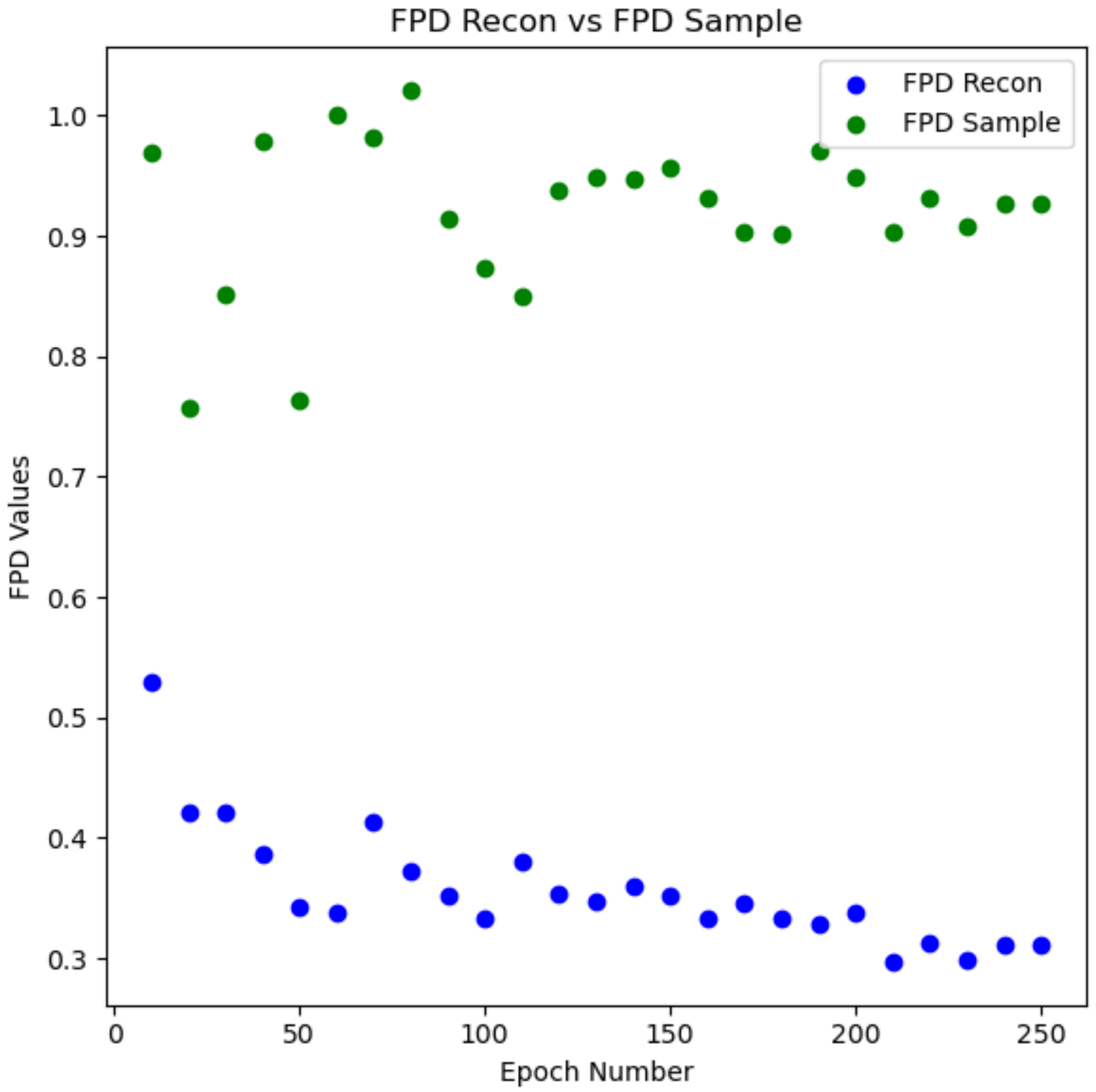


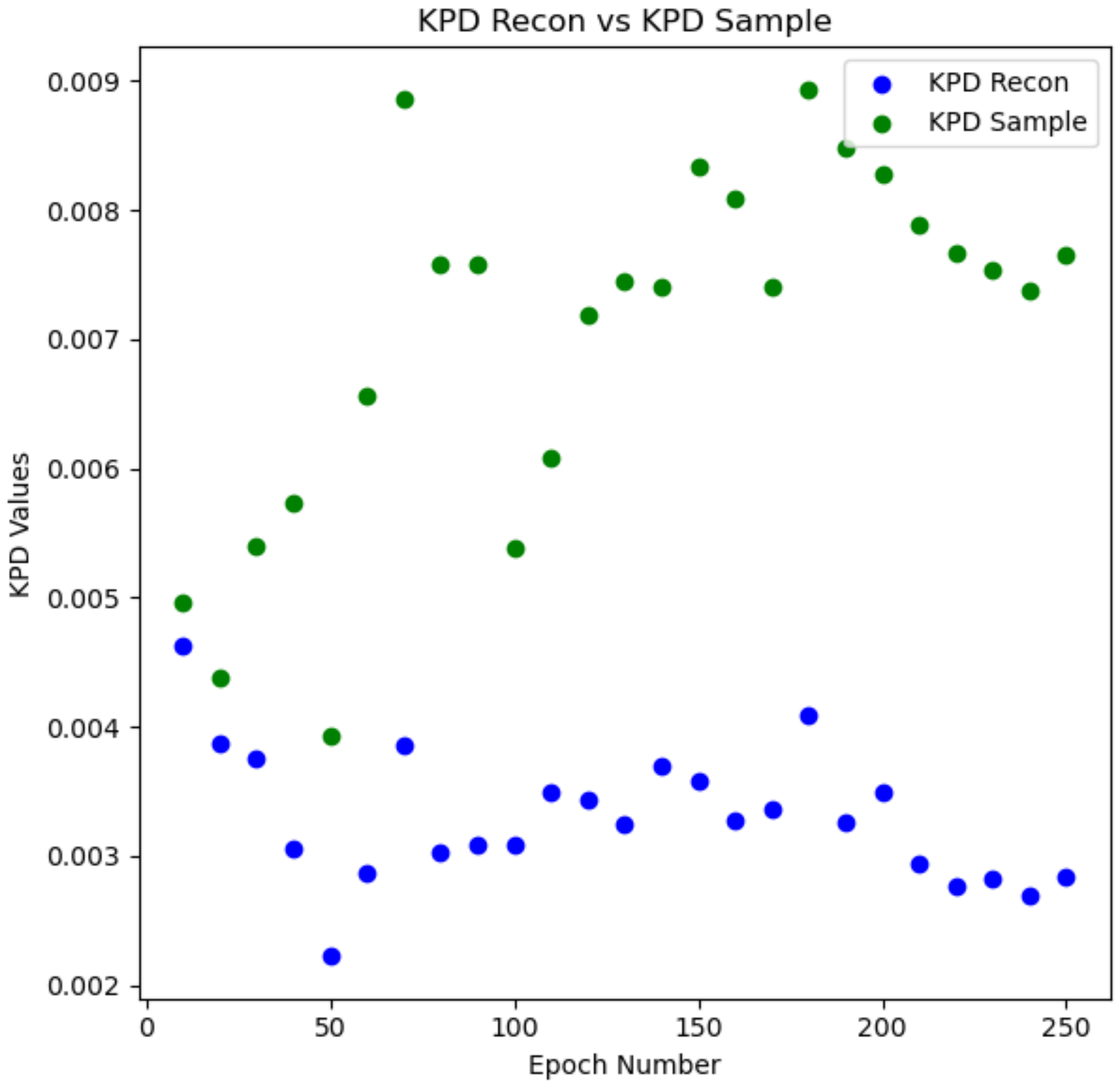


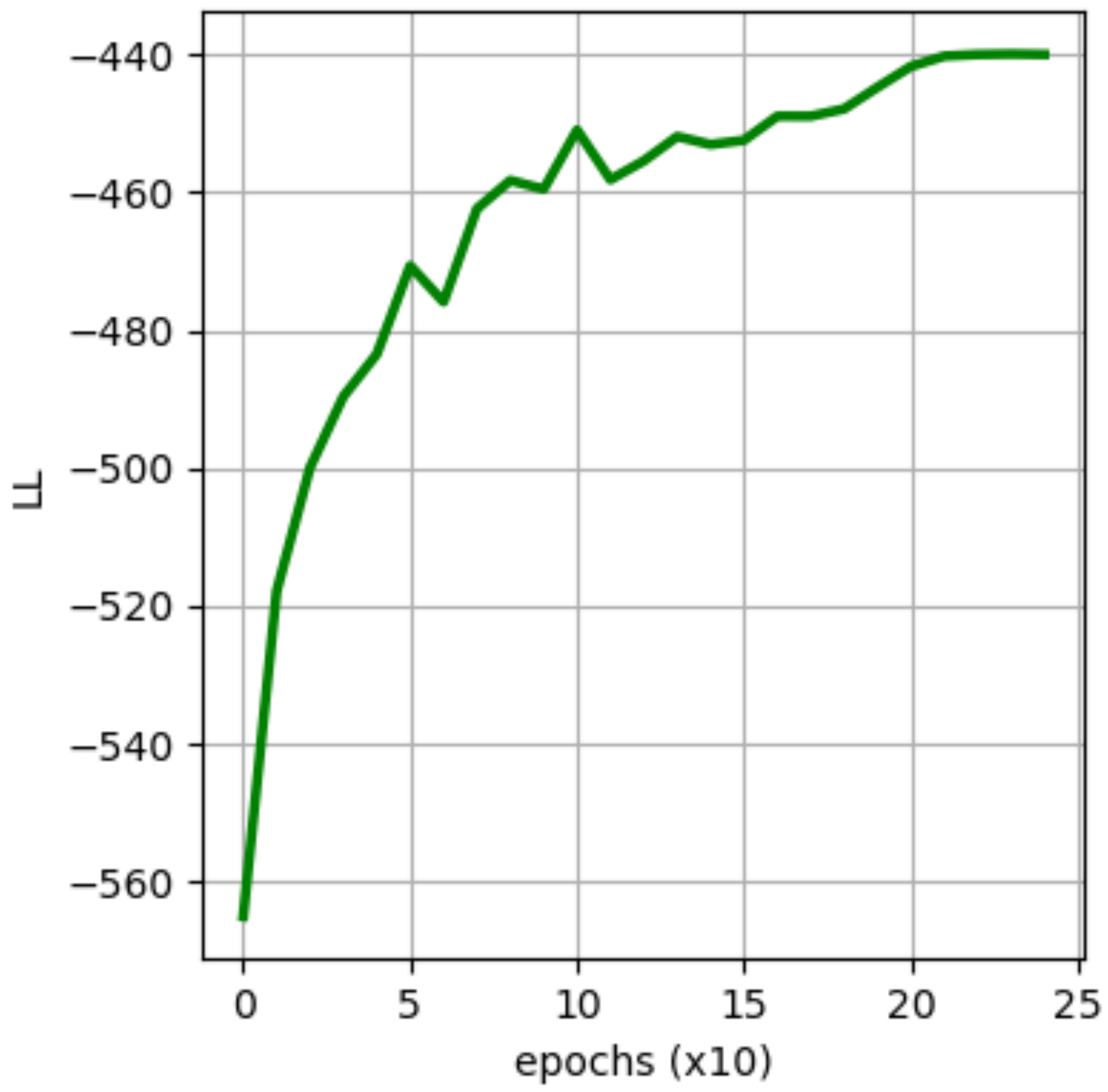
After fixing the periodicity bug

We trained several models

- *Divine-dream-509* – Zeph after bug fixed + mask removed from activation during training



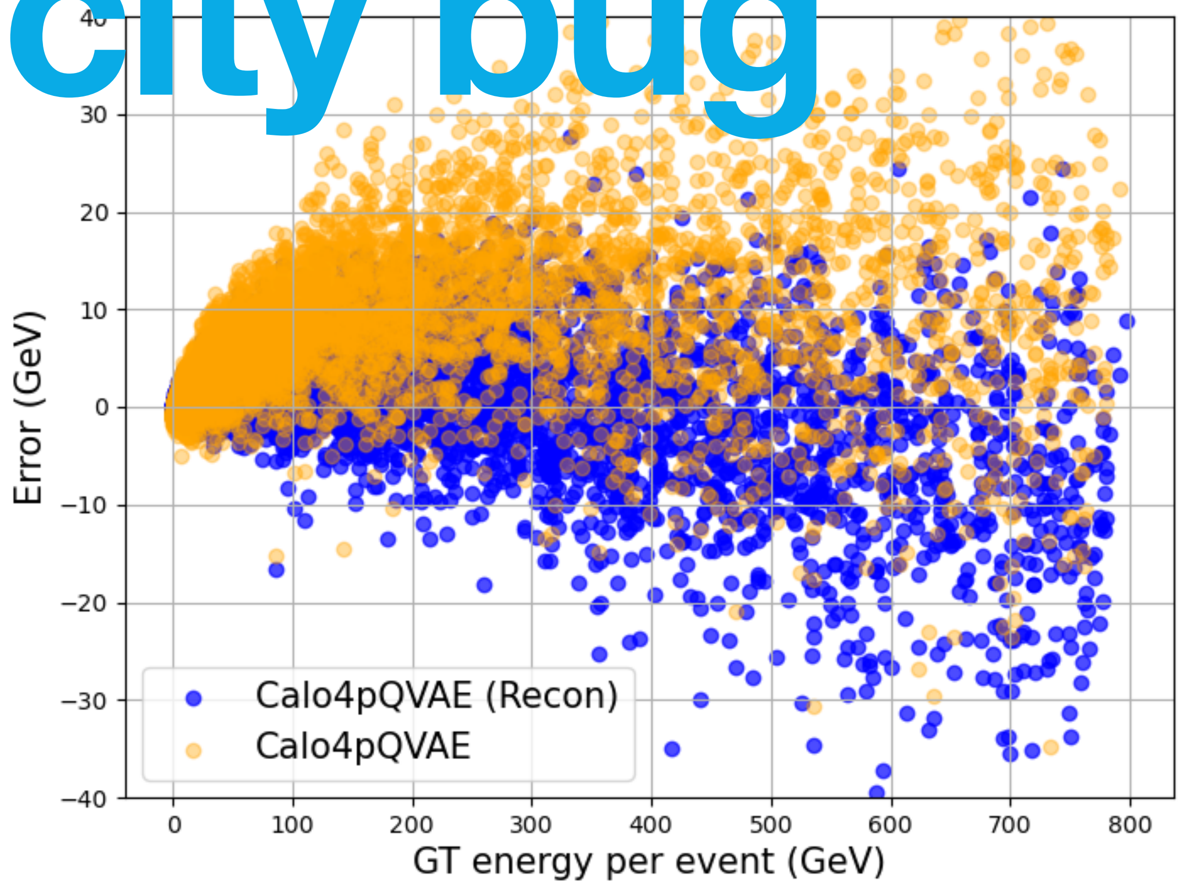


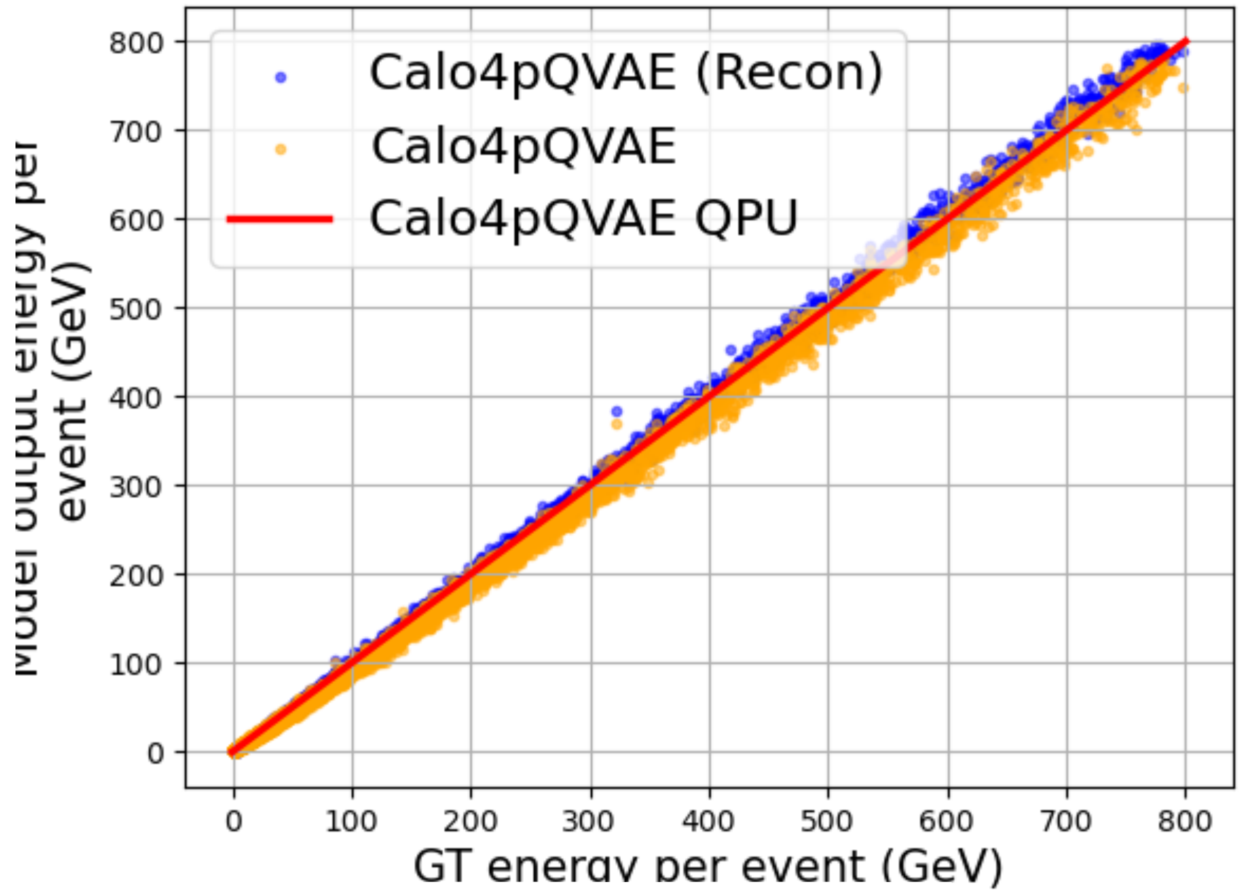


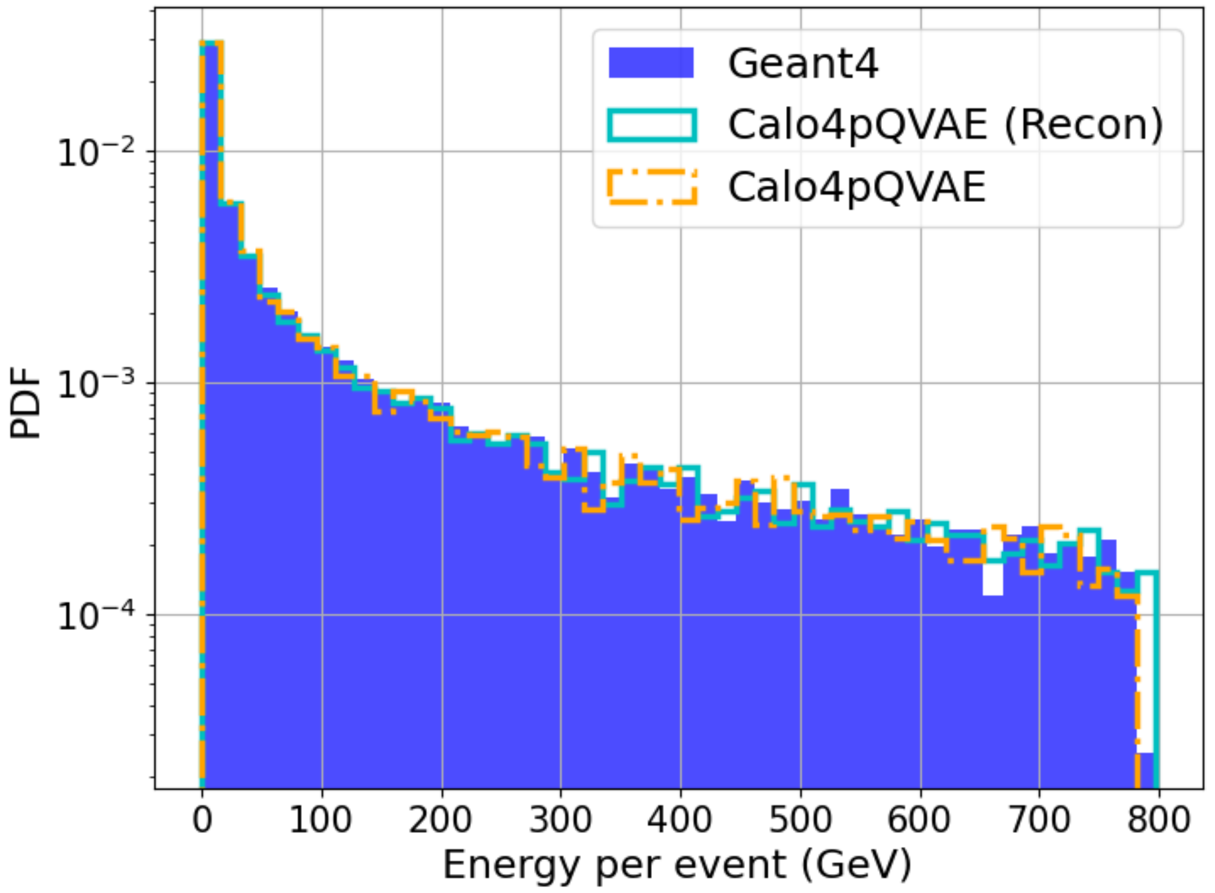
After fixing the periodicity bug

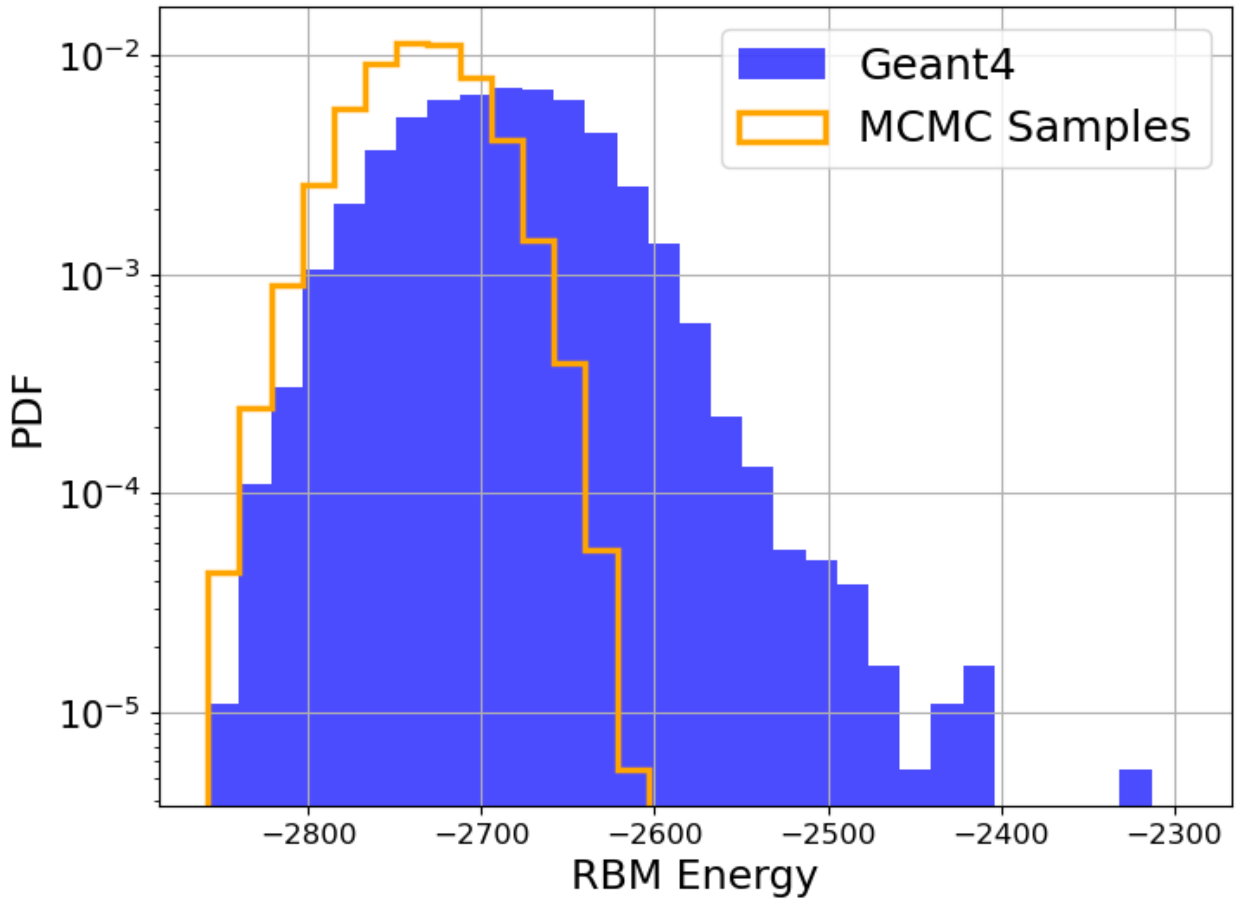
We trained several models

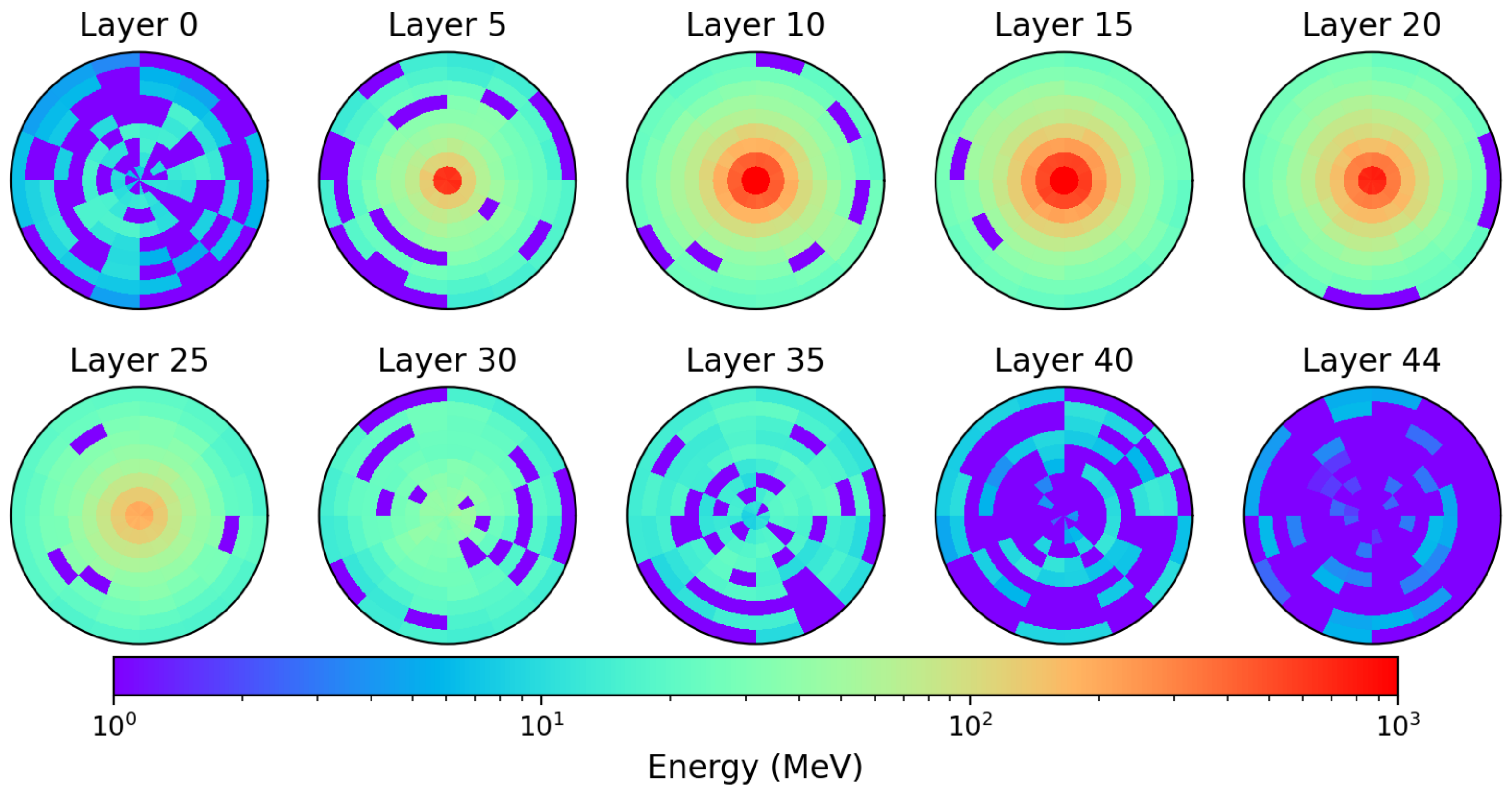
✦ *Divine-dream-509*

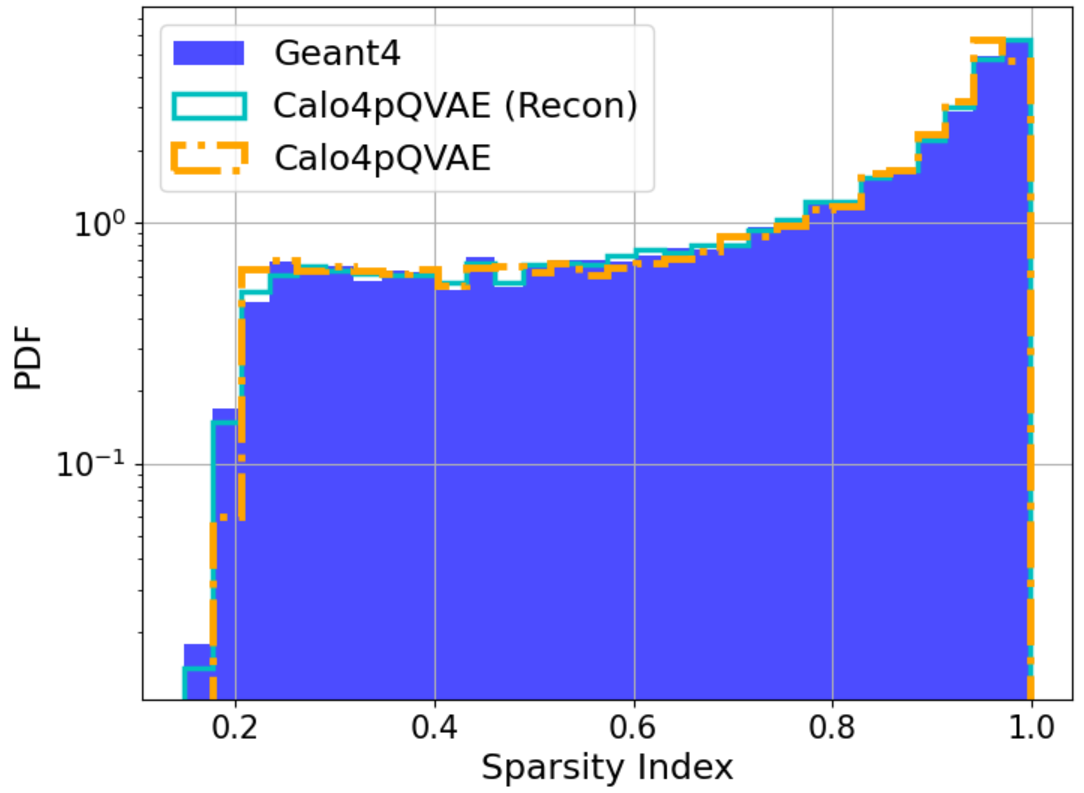


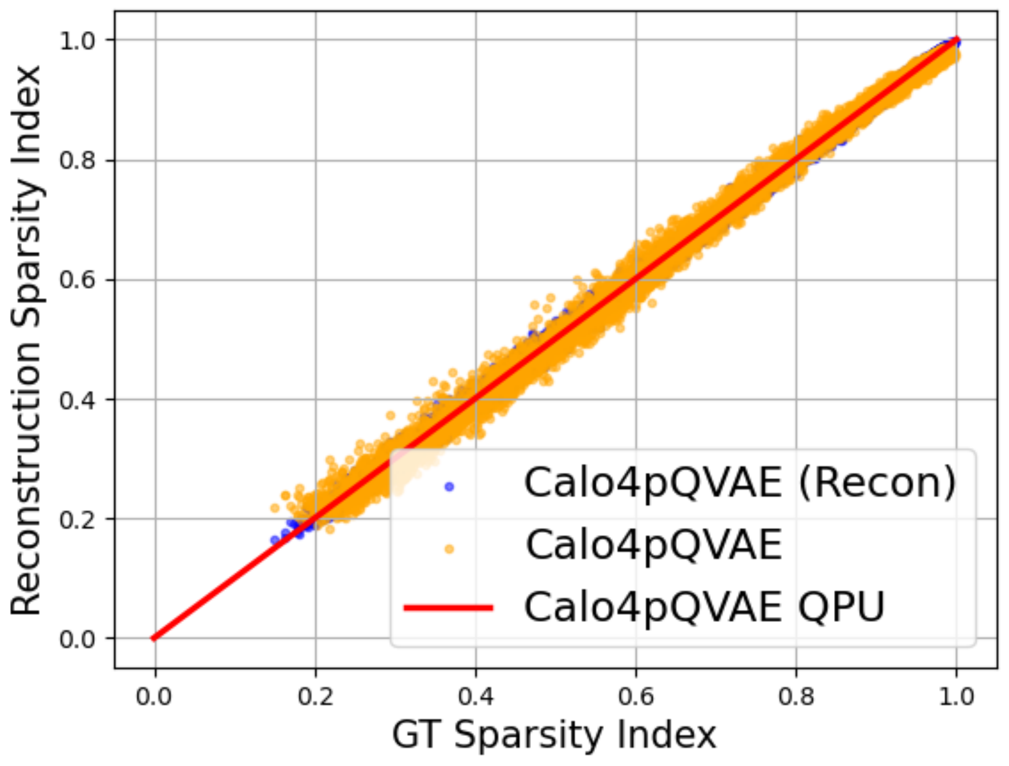


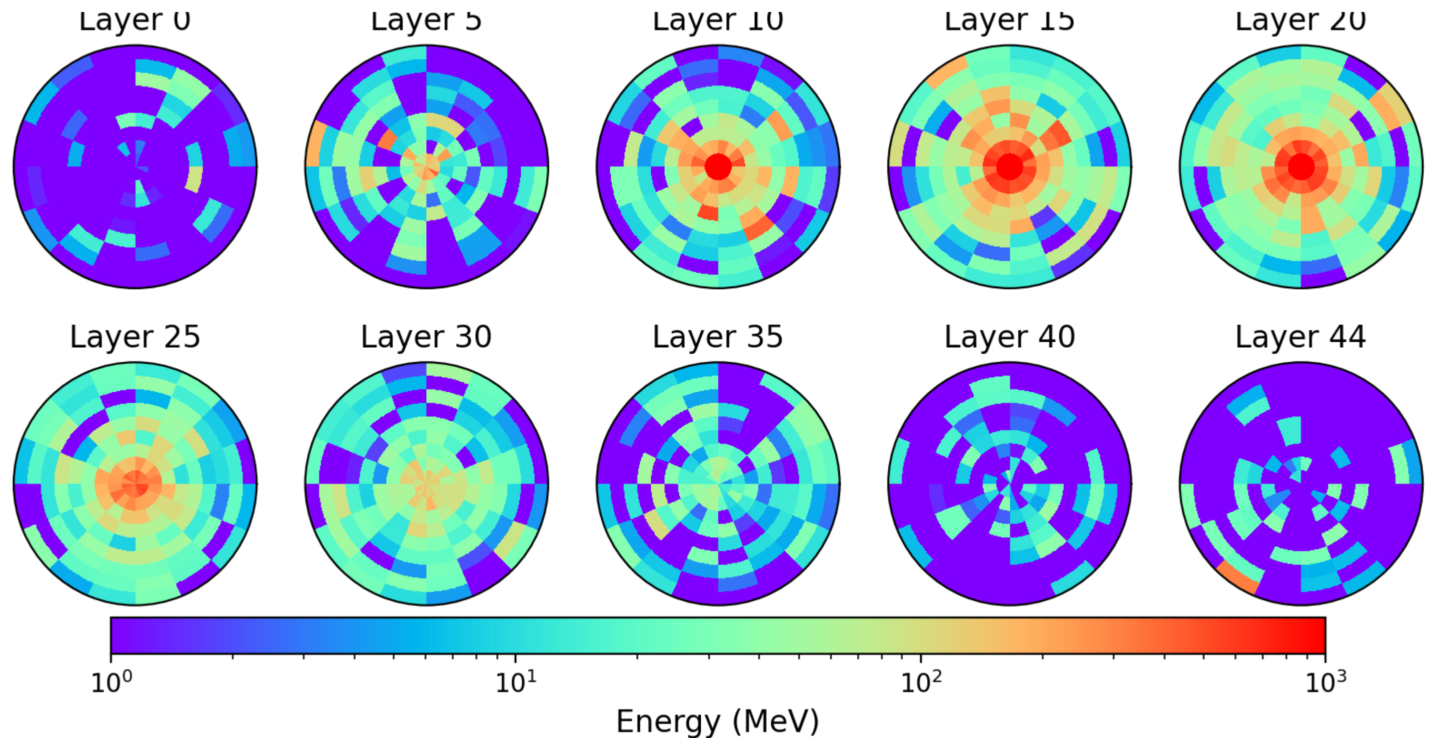








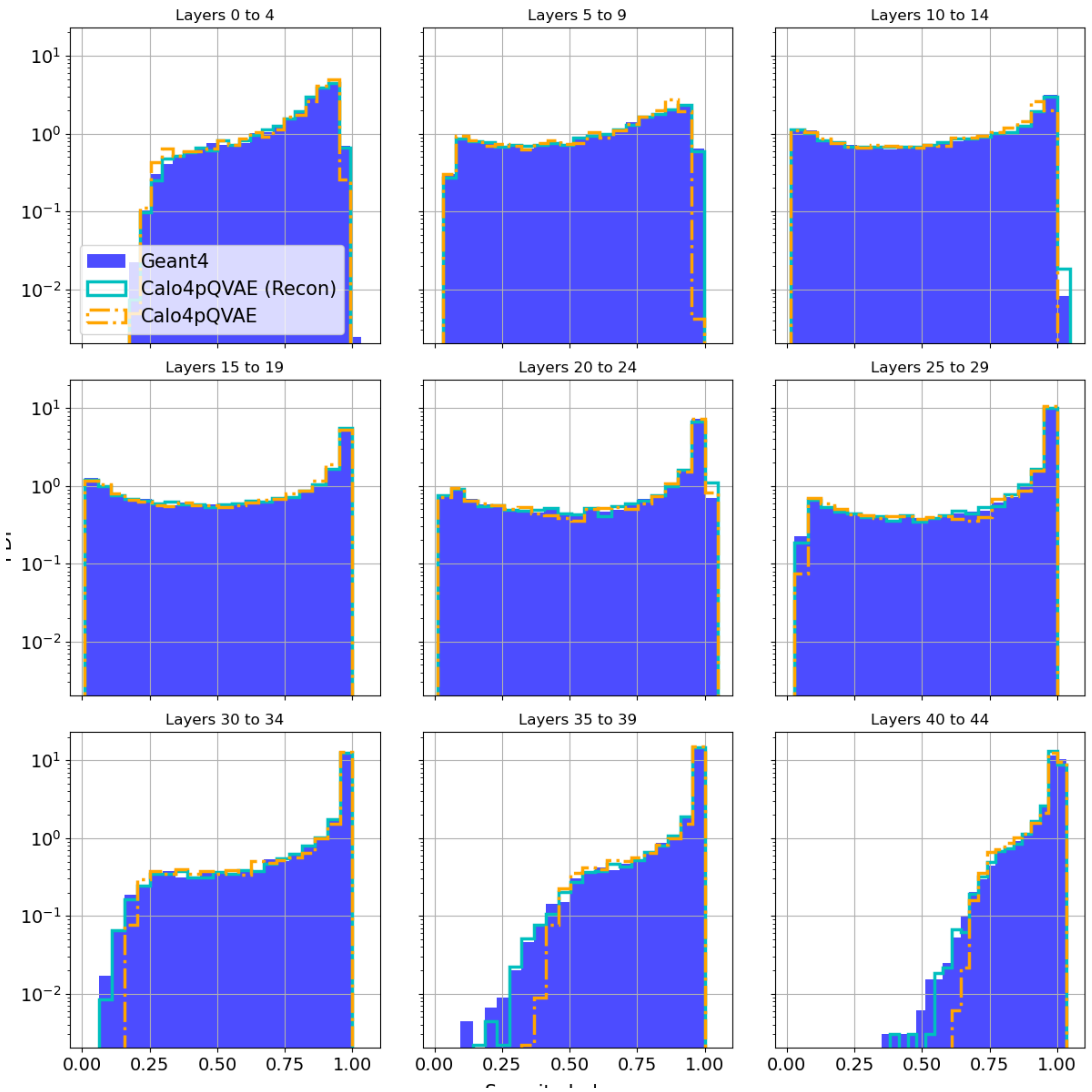


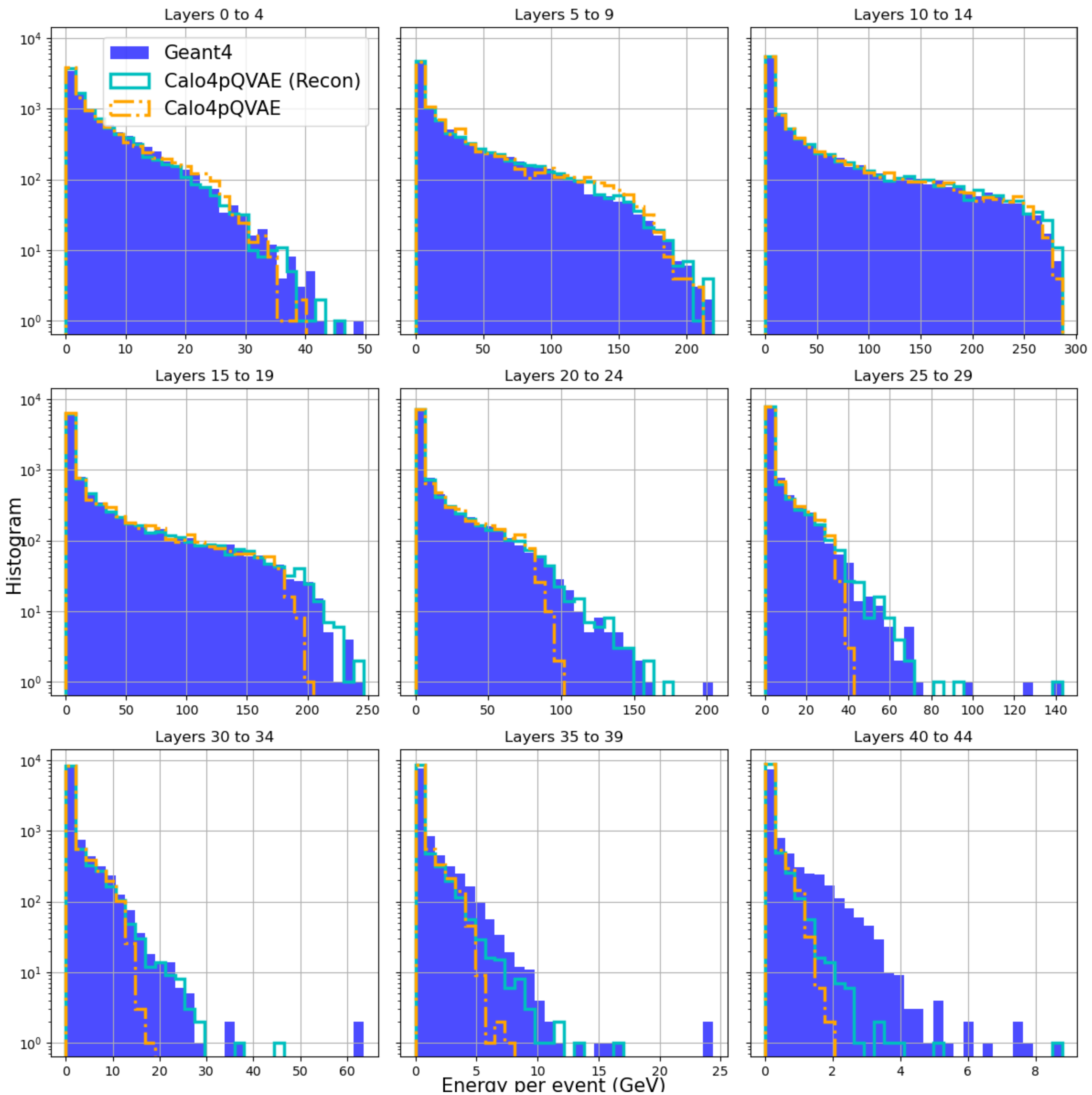


After fixing the periodicity bug

We trained several models

Divine-dream-509

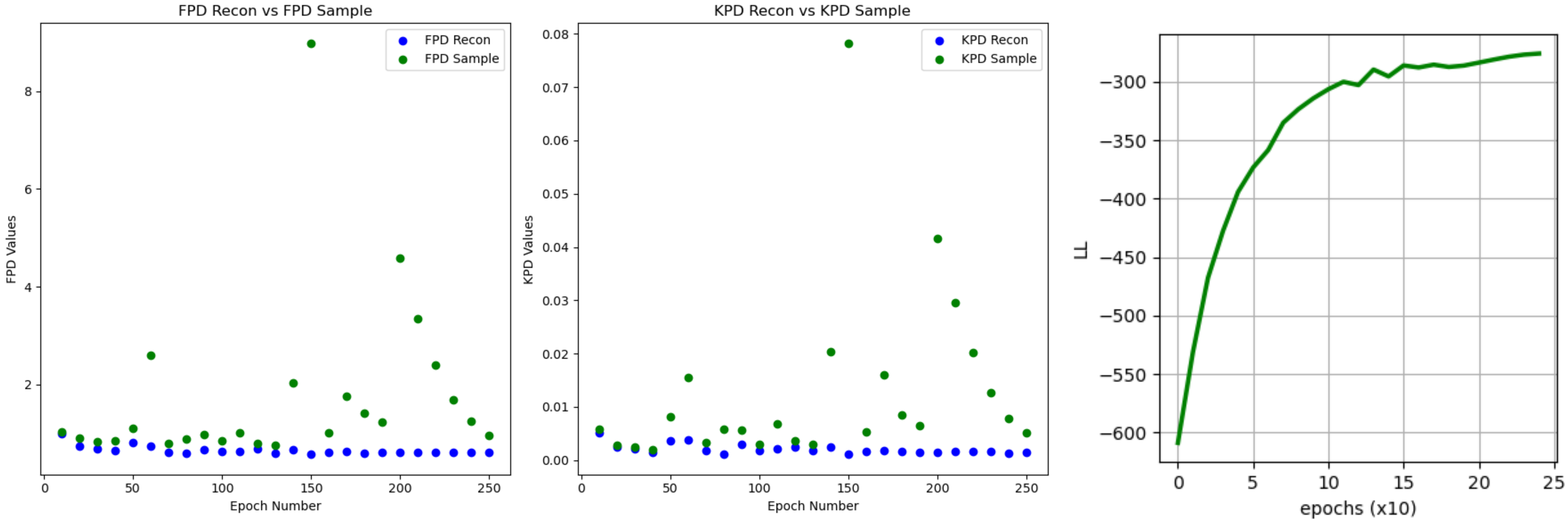




After fixing the periodicity bug

We trained several models

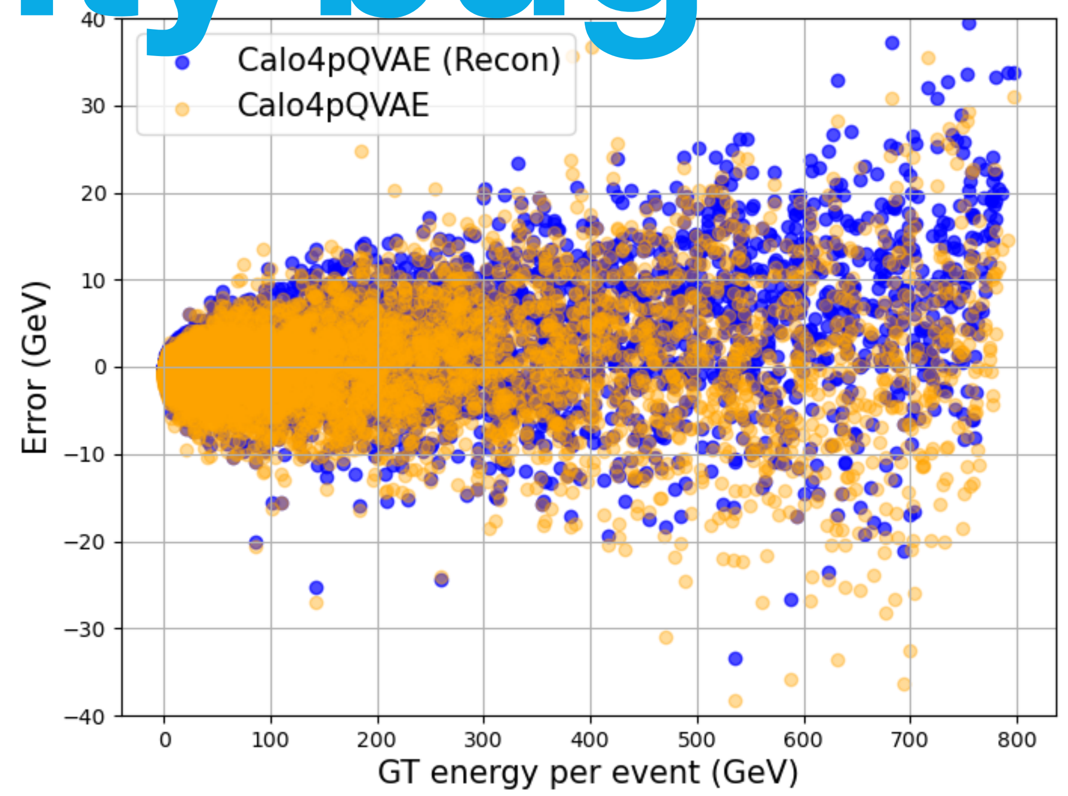
✦*Worthy-paper-518* – zeph after bug fixed + linear attention

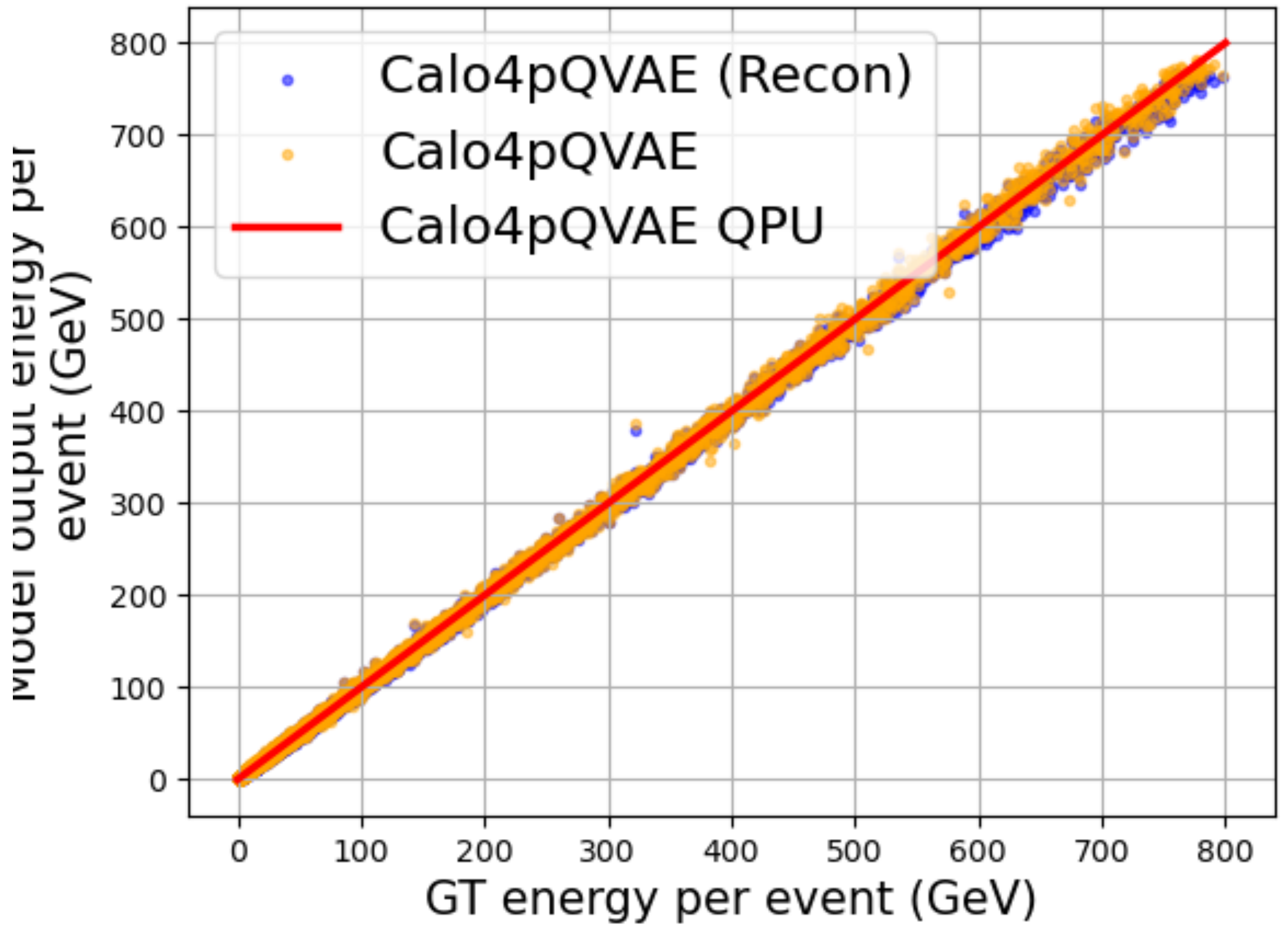


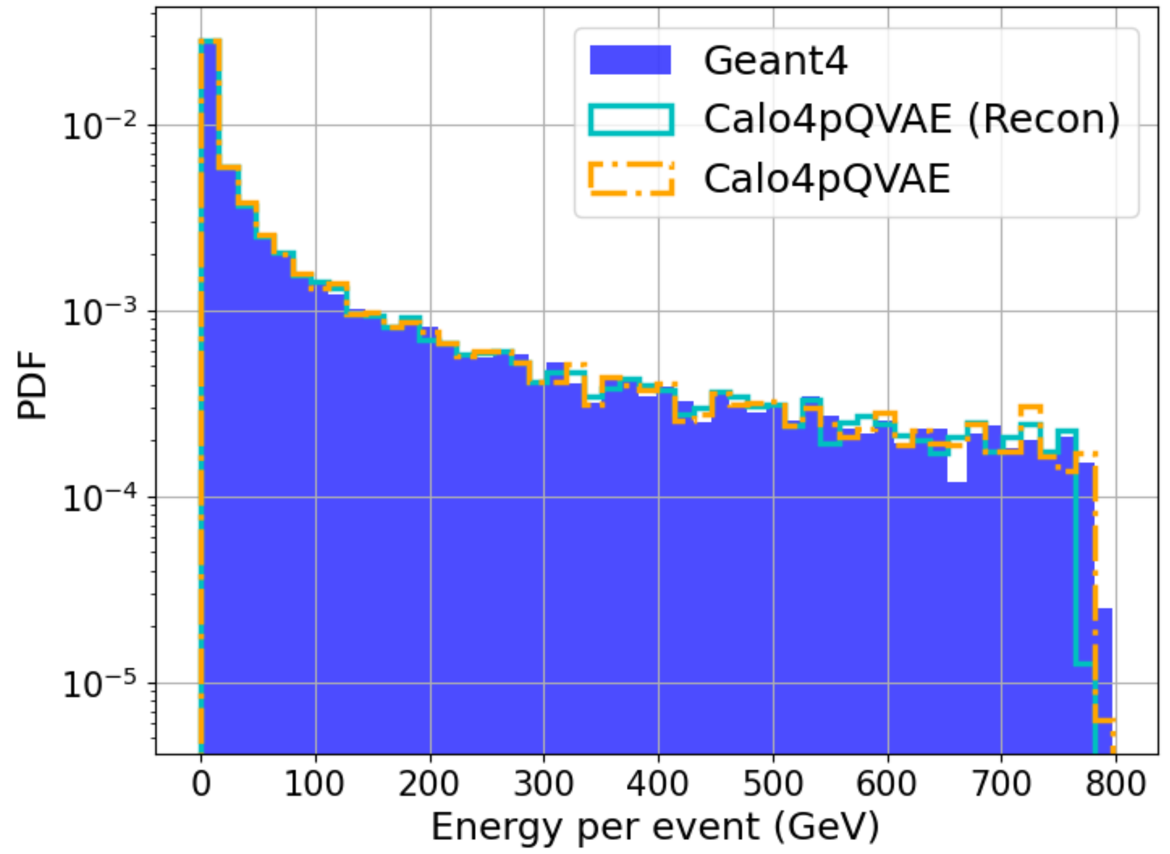
After fixing the periodicity bug

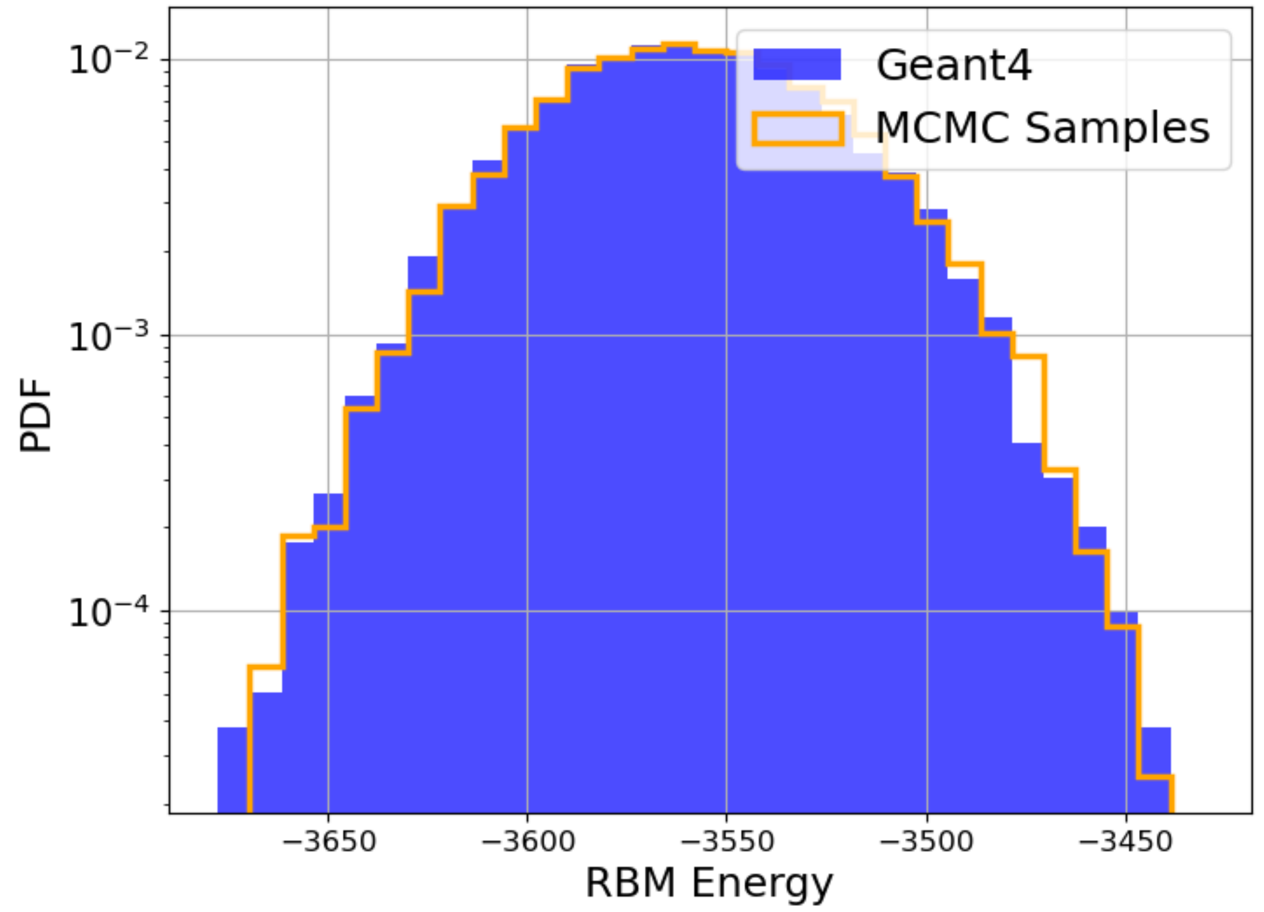
We trained several models

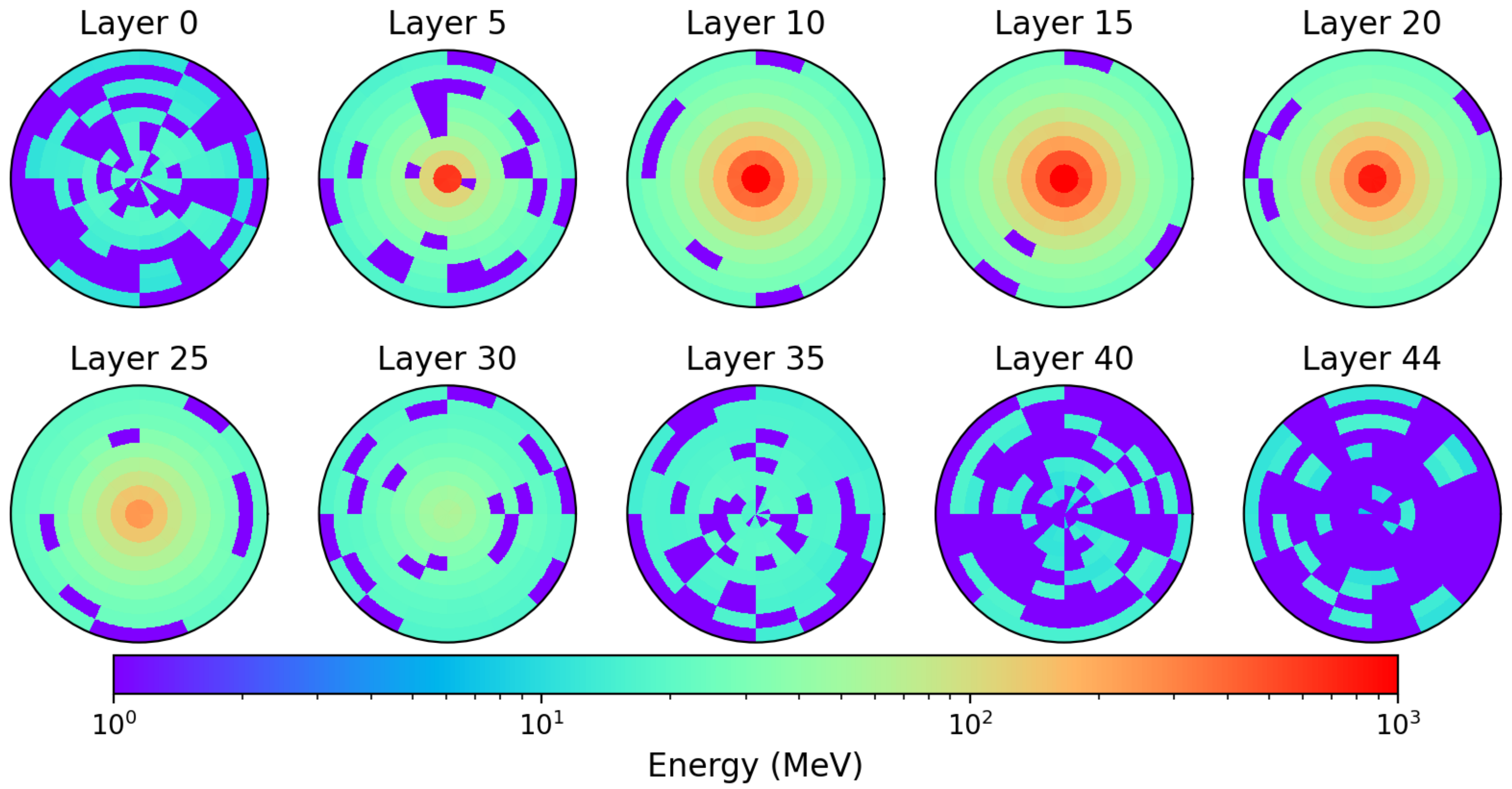
✦ *Worthy-paper-518*

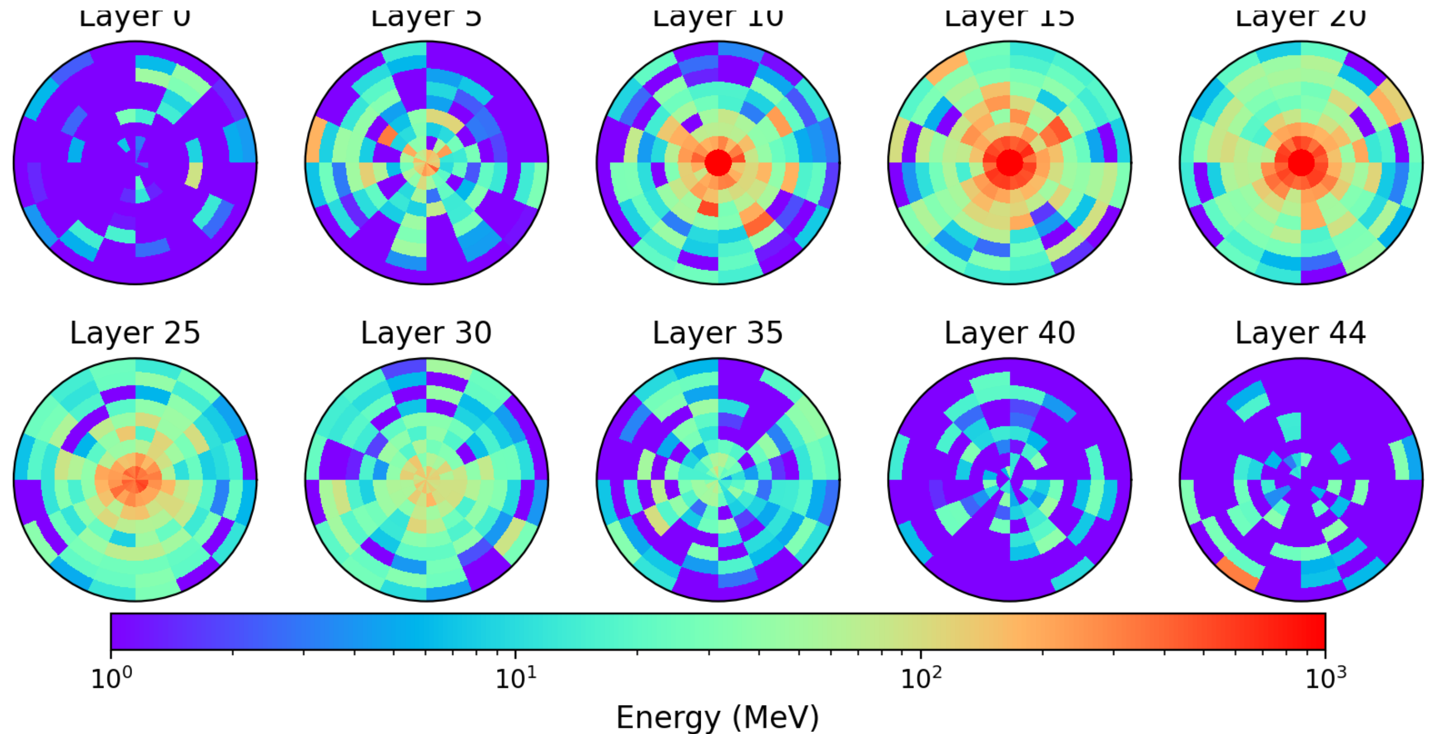


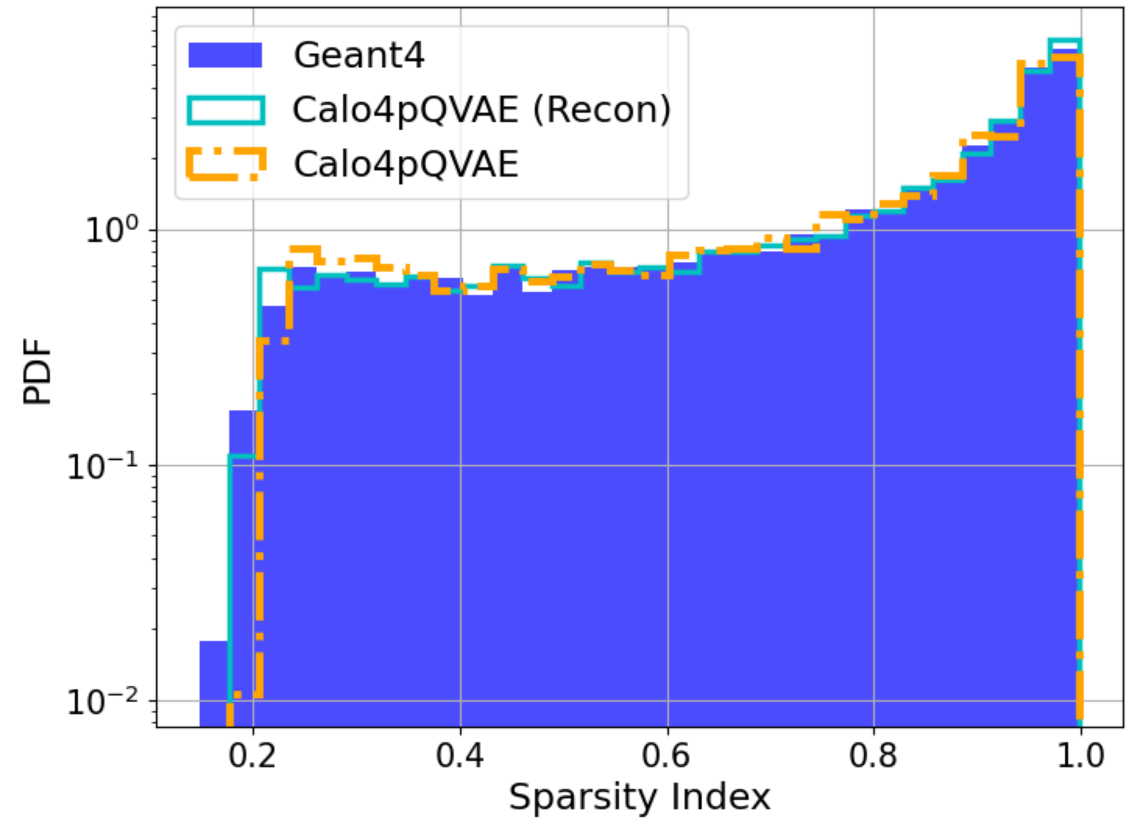








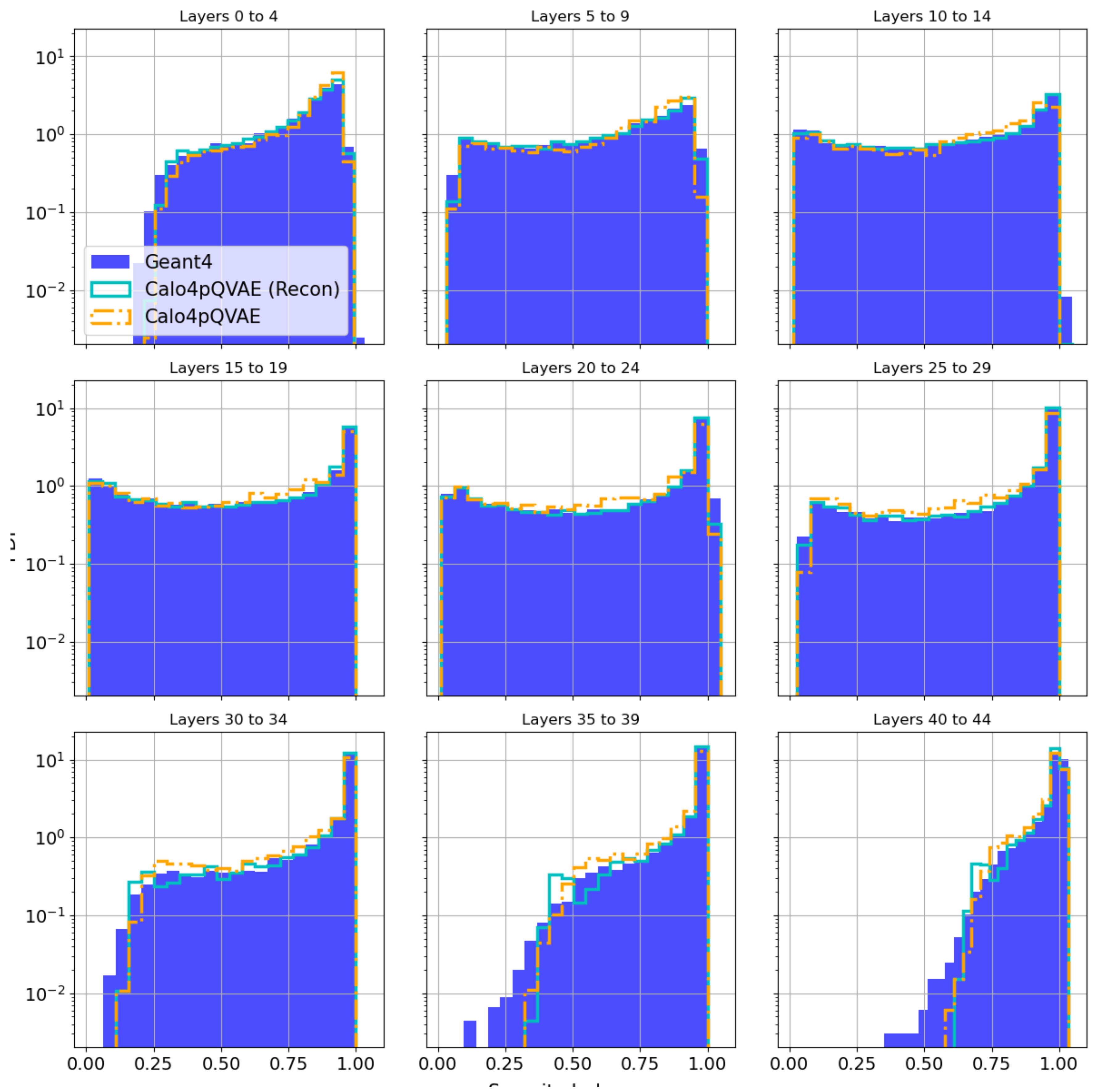


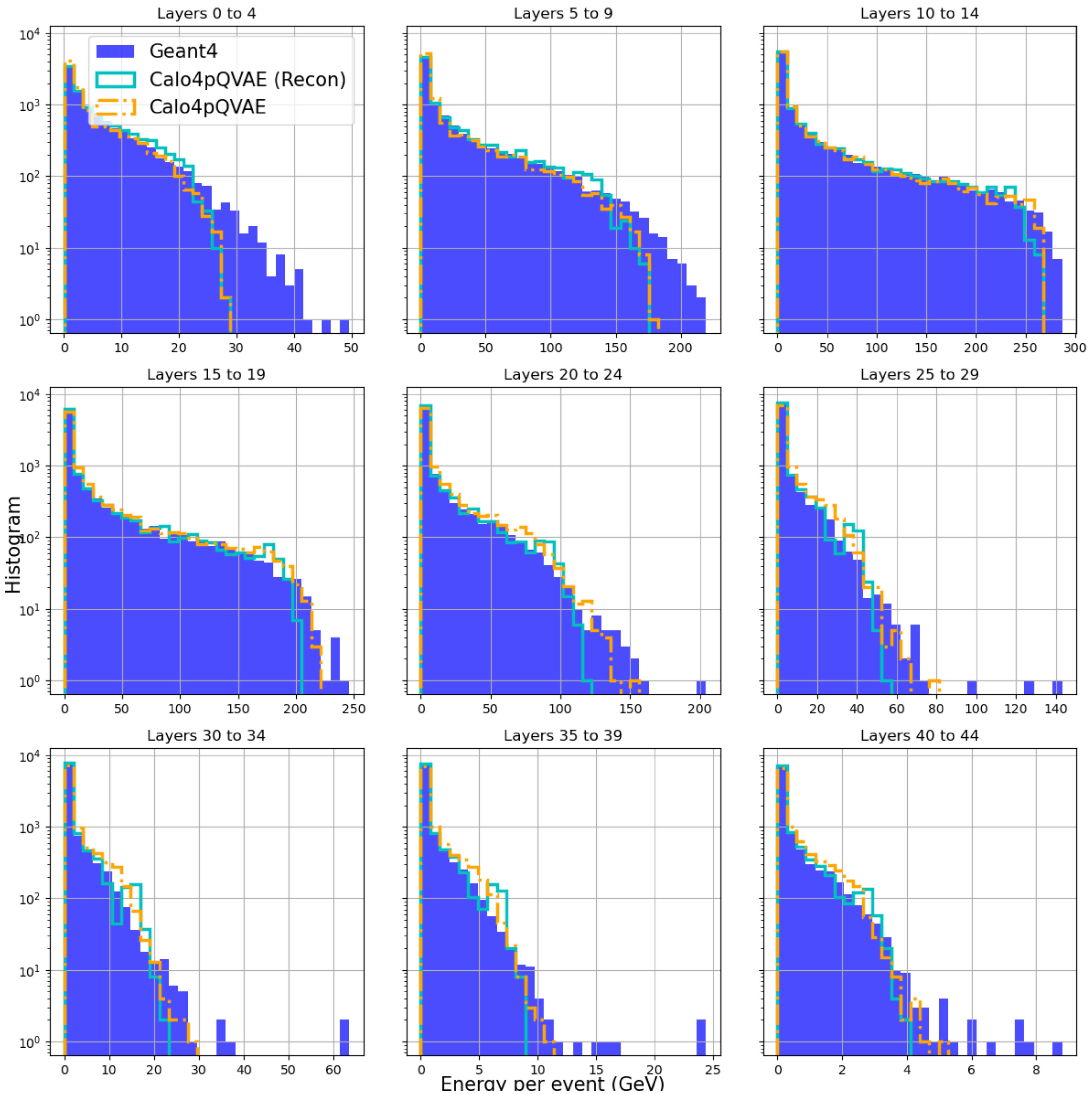


After fixing the periodicity bug

We trained several models

✦*Worthy-paper-518*

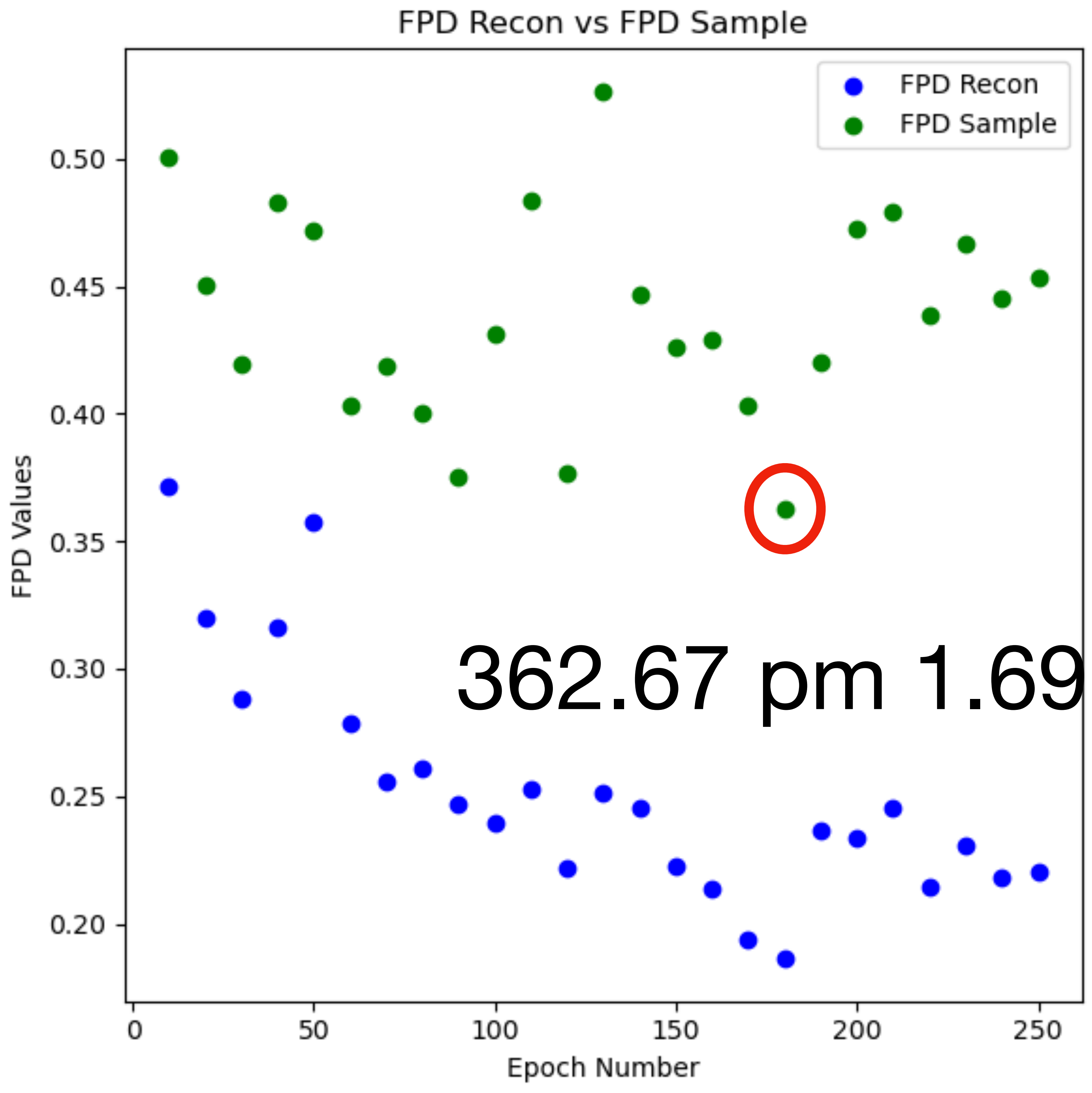




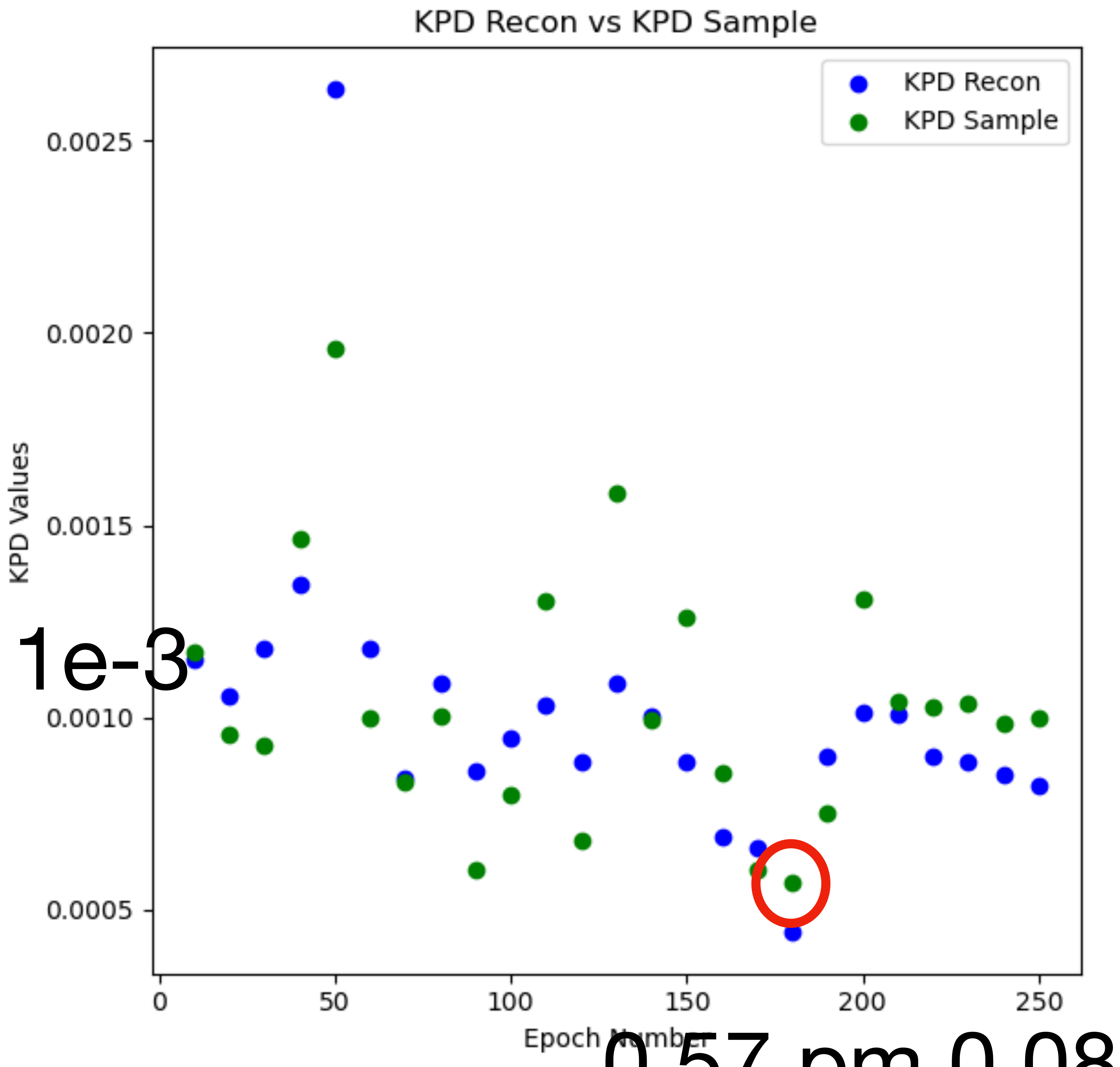
After fixing the periodicity bug

We trained several models

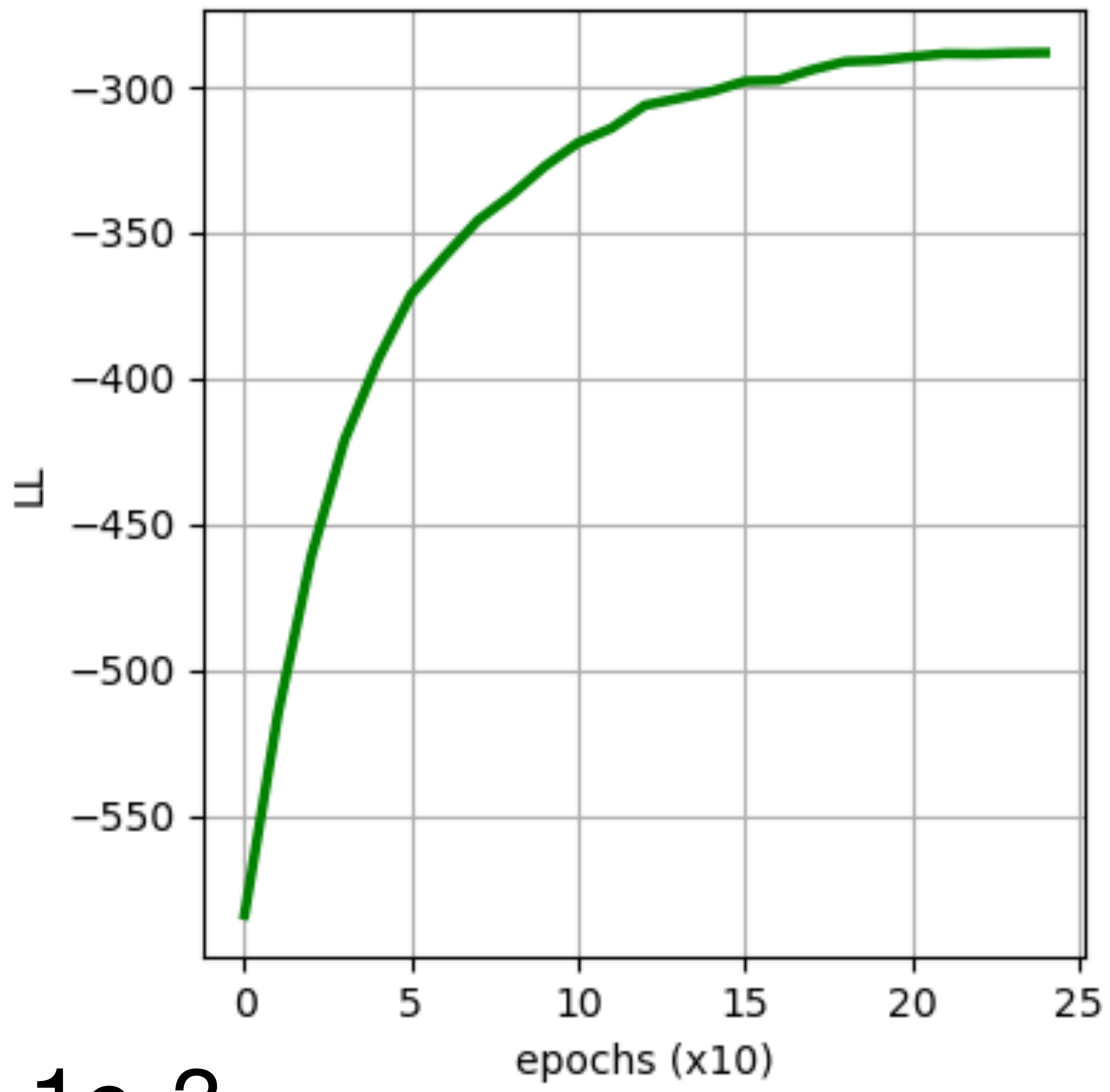
✦*Lively-grass-519* — zeph after bug fixed + non high-T approx



362.67 pm 1.69 1e-3



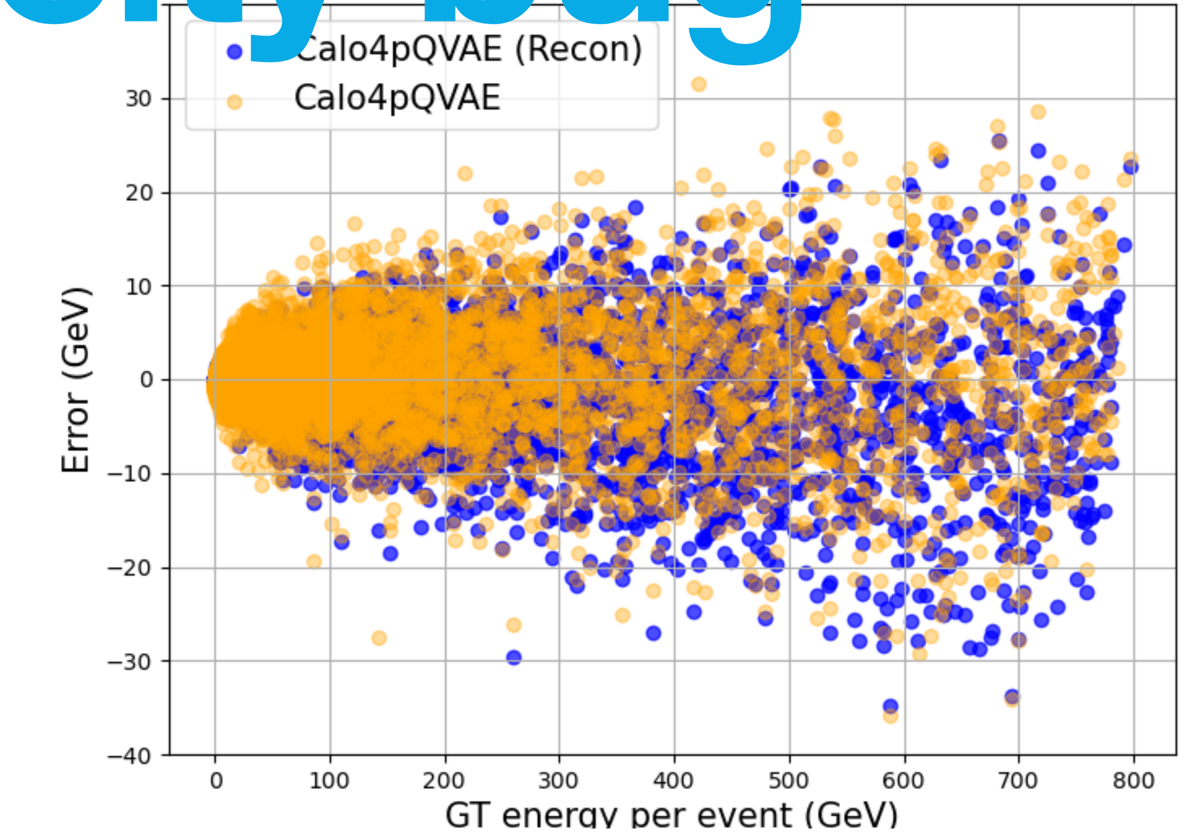
0.57 pm 0.08 1e-3

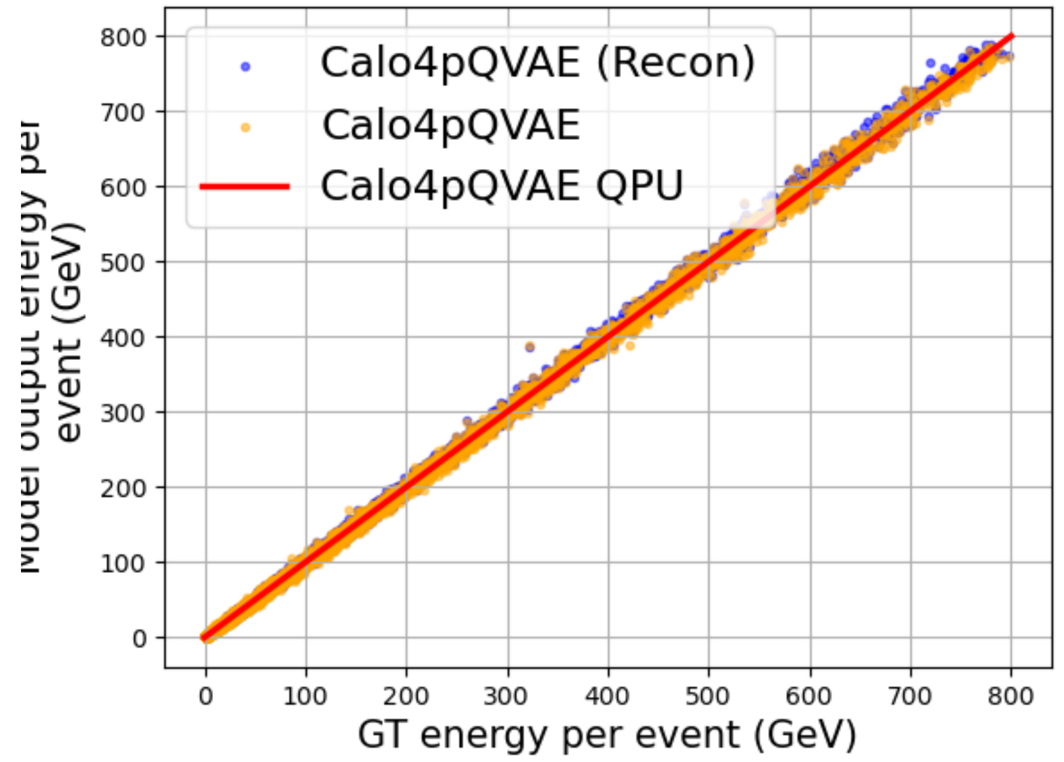


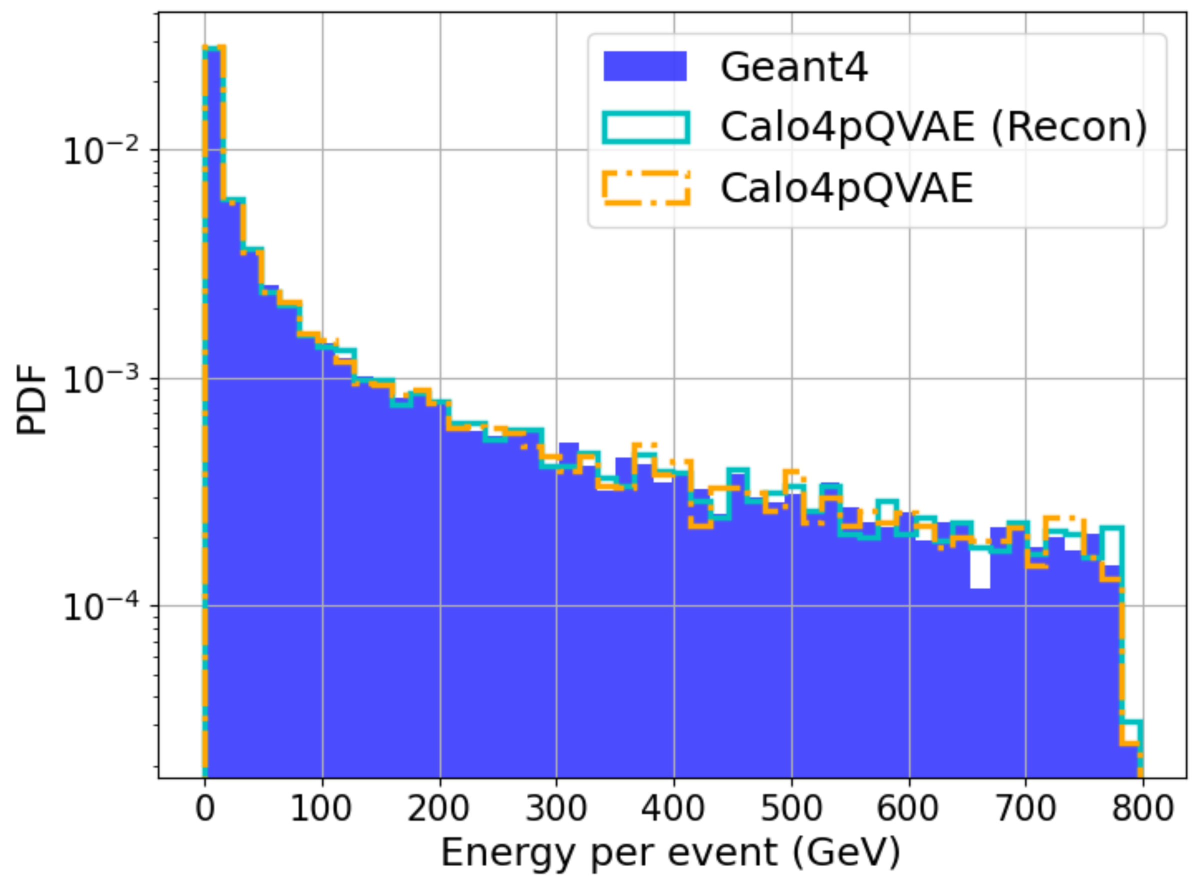
After fixing the periodicity bug

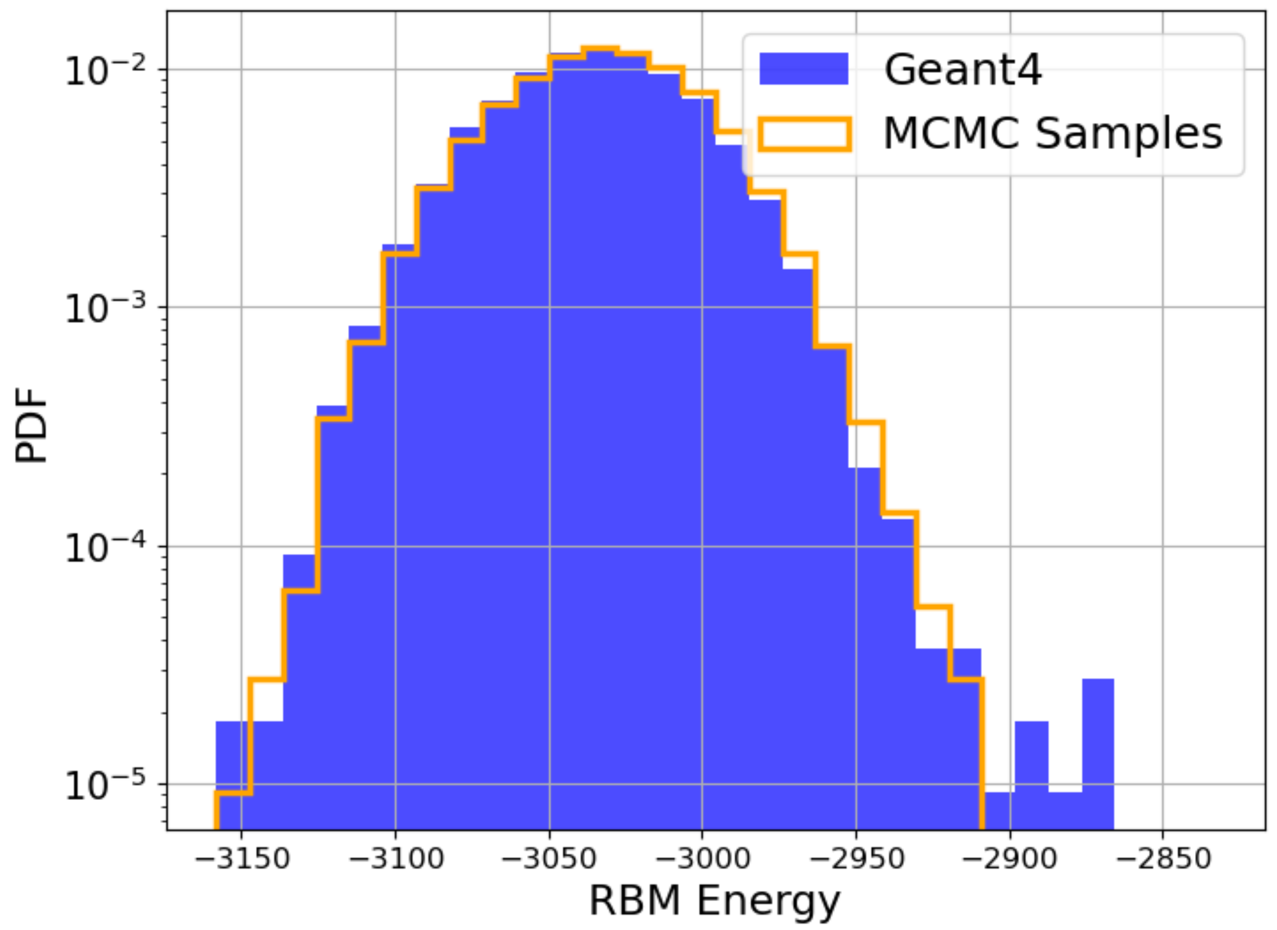
We trained several models

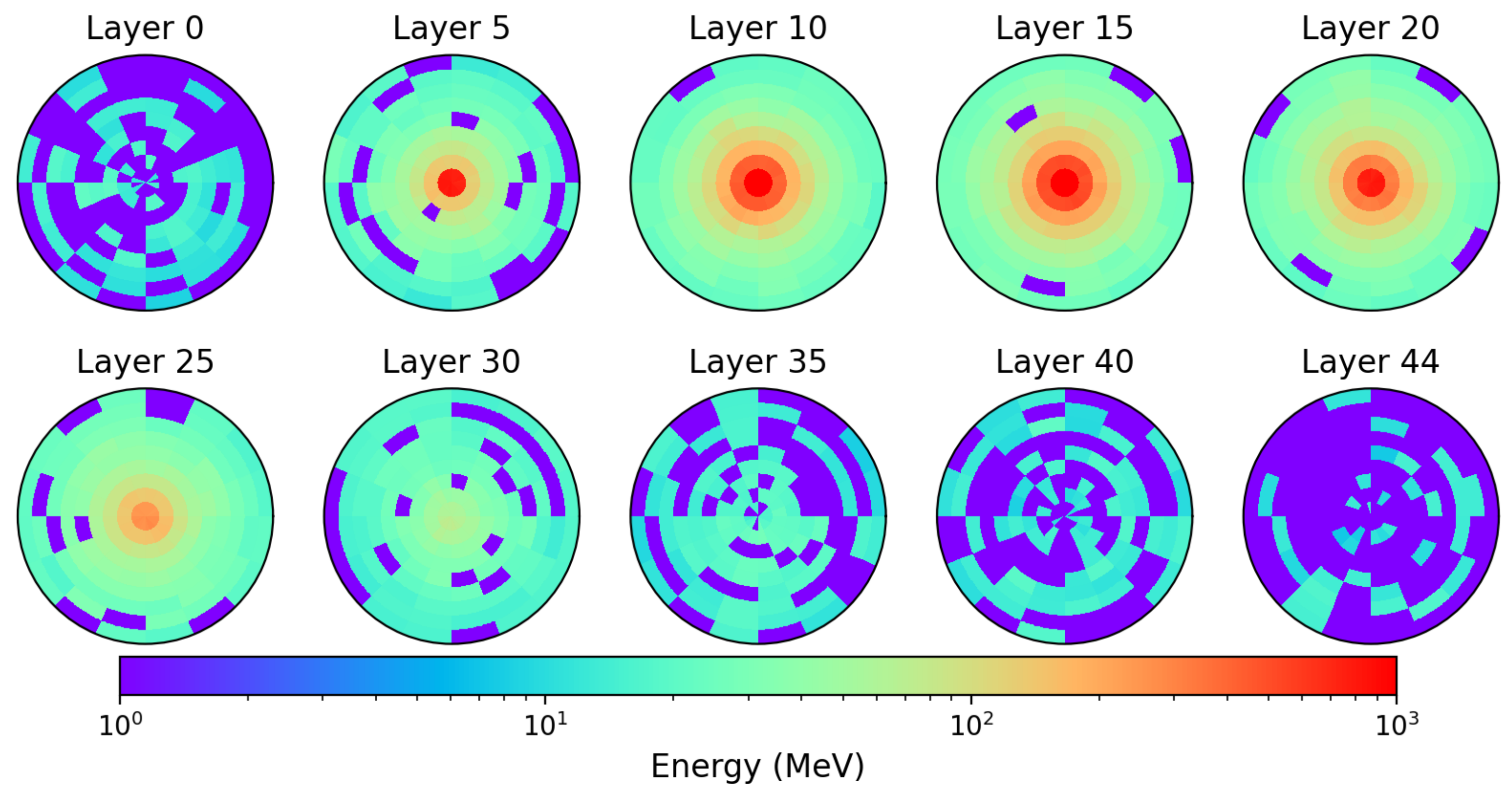
✦*Lively-grass-519*

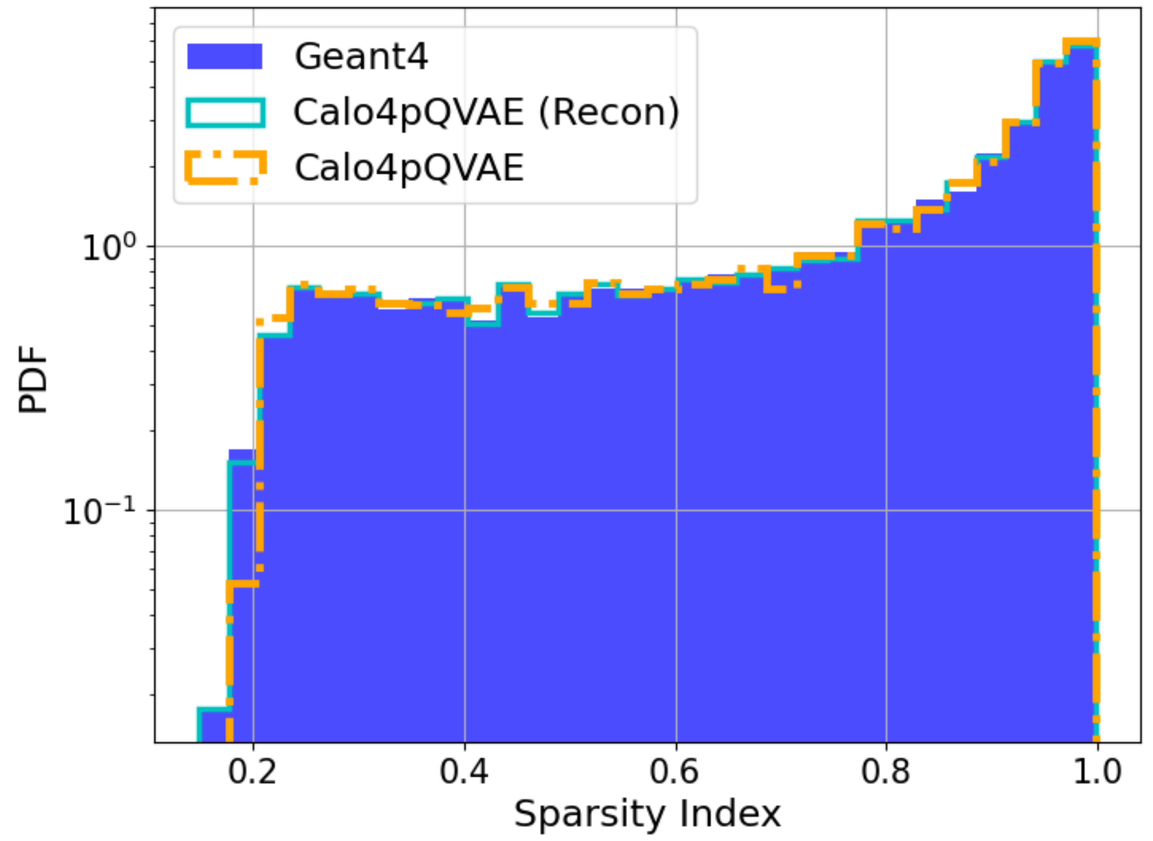


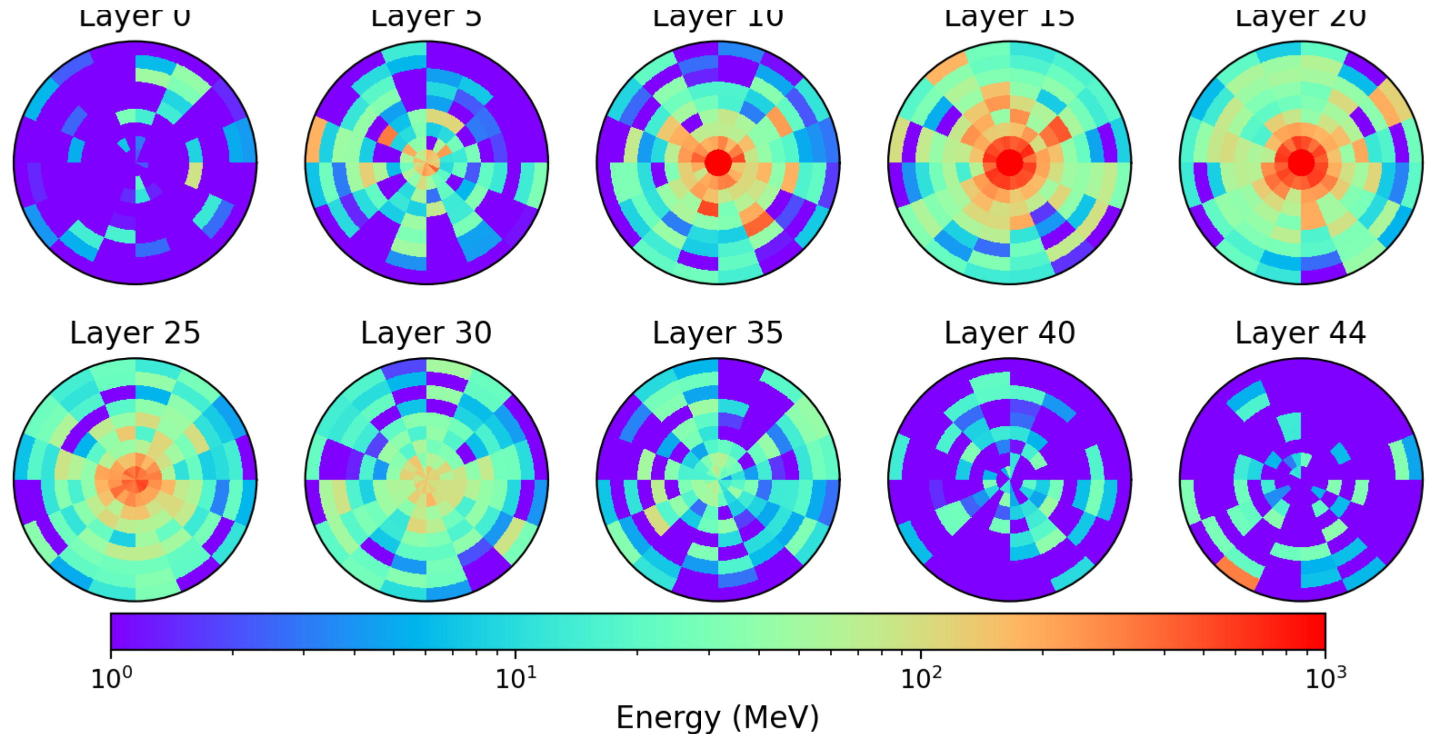








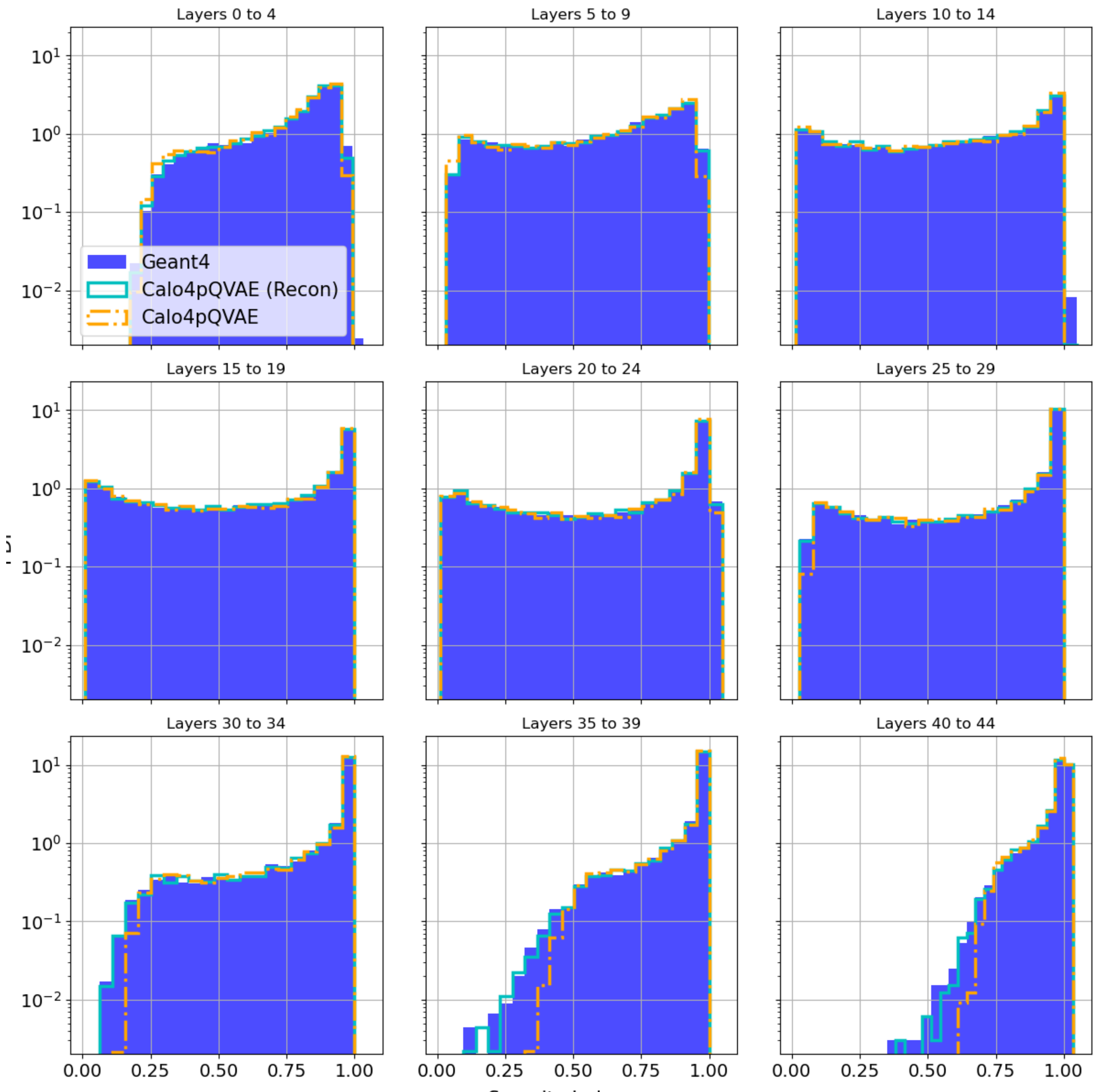


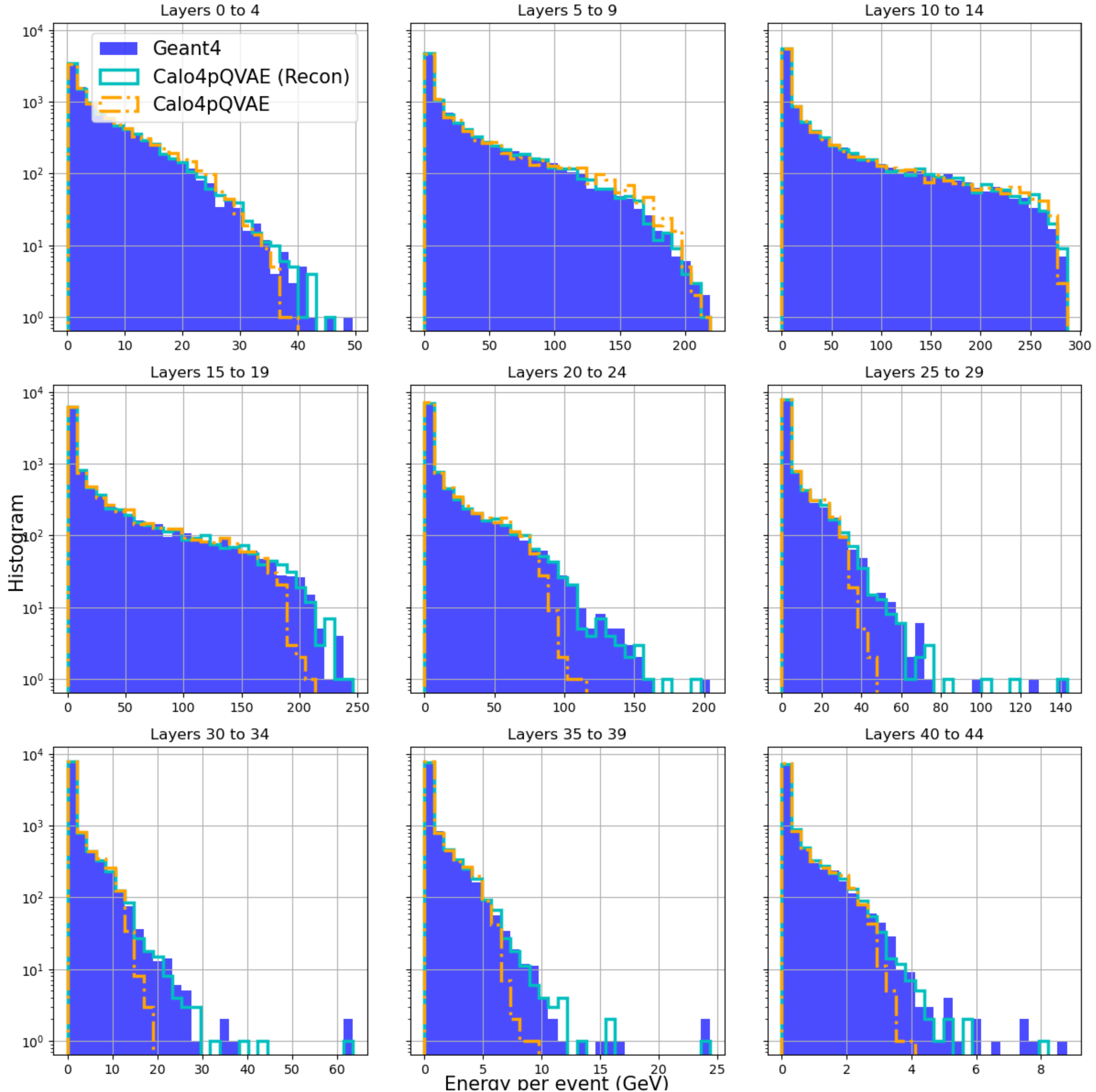


After fixing the periodicity bug

We trained several models

Lively-grass-519





The bug fix + the BGS exact gradient

Has led to better metrics

- **Before** FPD (x10^3): 380.6906 ± 1.1246 :: KPD (x10^3): 0.6147 ± 0.0649
- **After bug fix:** FPD (x10^3): 379.78 ± 1.77 :: KPD (x10^3): 0.57 ± 0.05
- **After bug fix + BGS exact gradient:** FPD (x10^3): 362.67 ± 1.69 :: KPD (x10^3): 0.57 ± 0.08

the BGS exact gradient

How was it coded?

- **Before**
  Loss = MSE + Hits + Entropy + Positive_Energy + Negative_Energy
  Backward
  Step

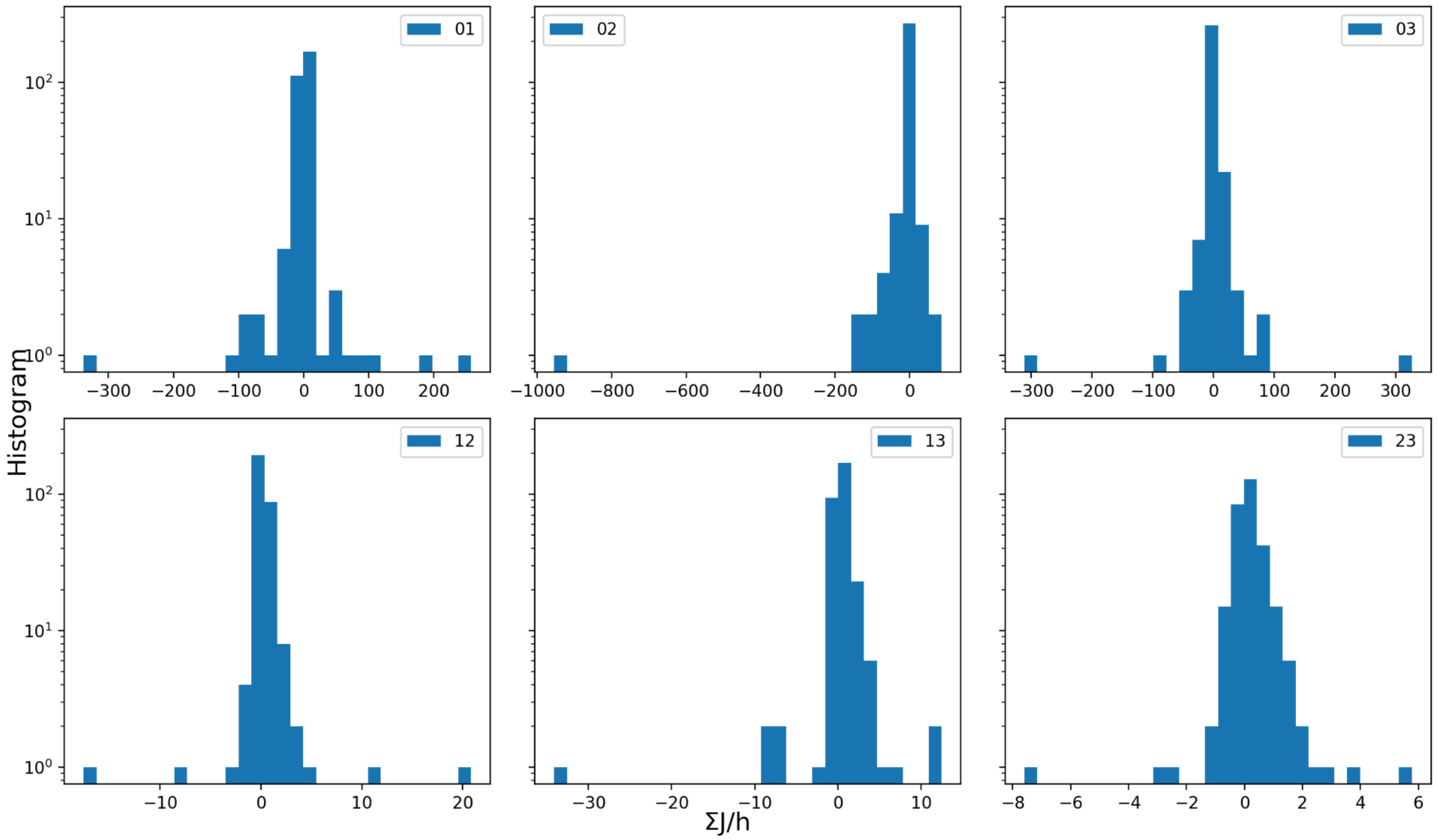
- **After**
  Loss = MSE + Hits + Entropy
  <u>Update RBM parameters (using SGD and LR=0.001)</u>
  Backward
  Step

Ratio between couplers and fields

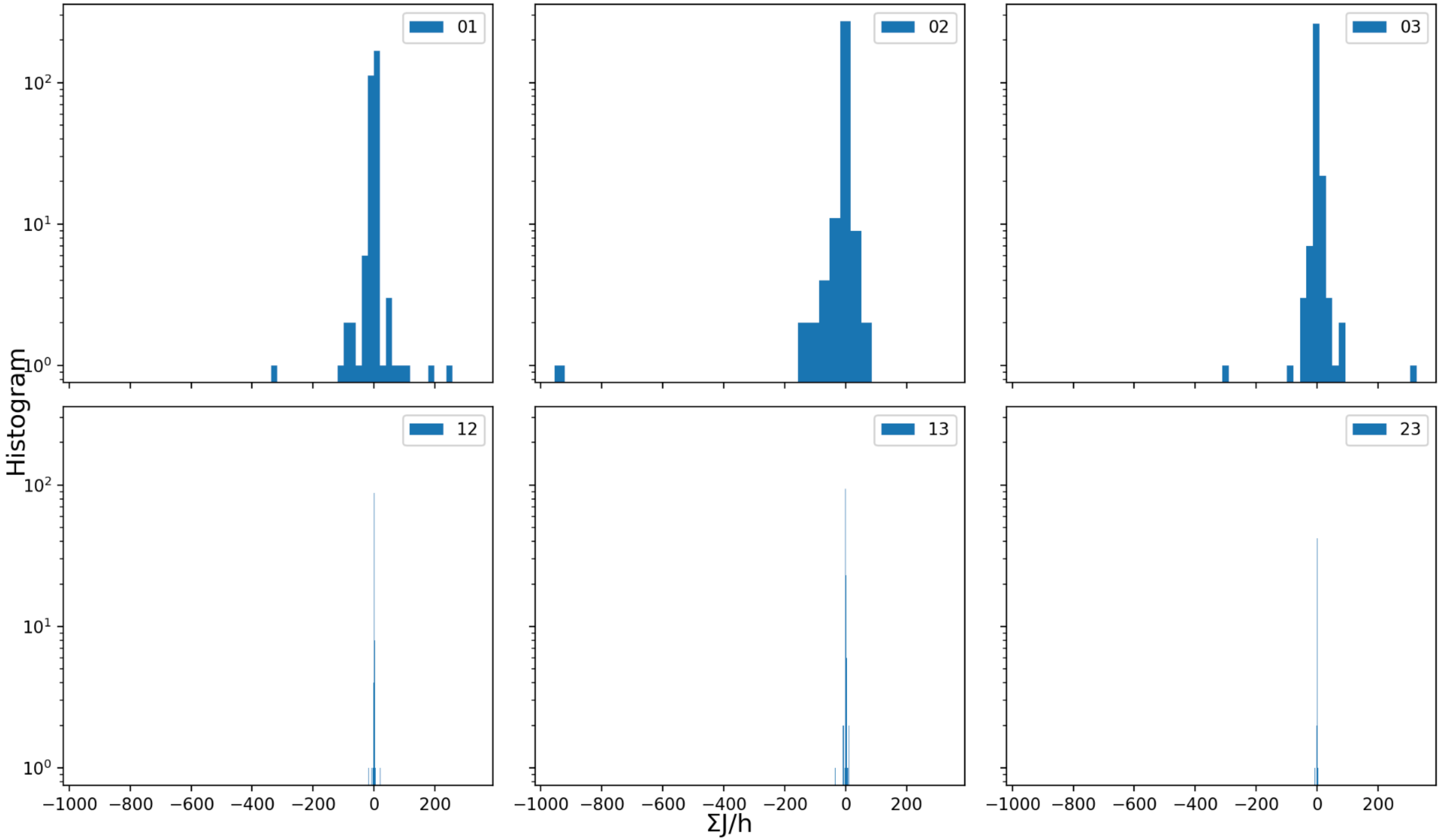
For a given spin i, what is the ratio between the some over its couplers and the self-field?

$$\frac{1}{h_i} \sum_j J_{ij}$$

Ratio between couplers and fields



Ratio between couplers and fields



RBM to Diffusion Model equivalence

$$Z = \sum_{v,h} e^{-\beta E(v,h)}$$

$$E(v,h) = -\langle v | a_0 \rangle - \langle b_0 | h \rangle - \langle v | W | h \rangle$$

$W = U \Sigma V^t$ (SVD)

$$|x\rangle = U|v\rangle, \qquad |y\rangle = V|h\rangle$$

$$Z = \sum_{x,y} e^{-\beta E(x,y)}$$

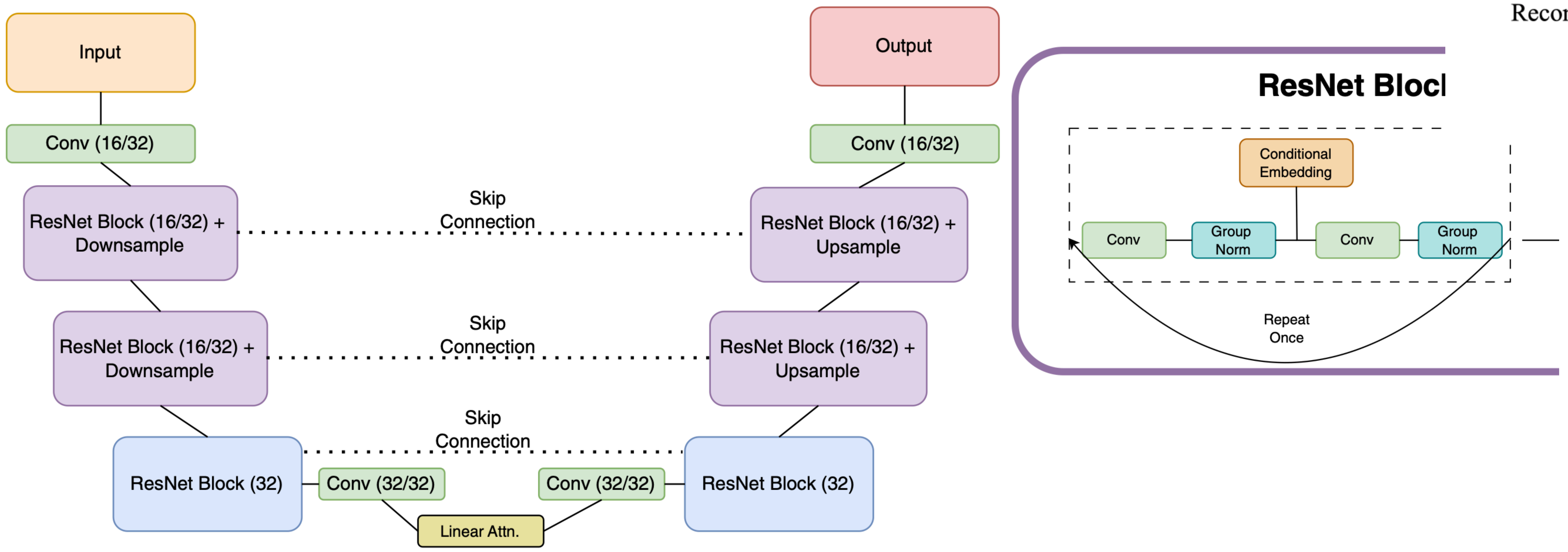
Diffusion models

How good are they?

CaloDiffusion

Kevin Pedro, Oz Amram

Denoising Model



CaloLatentDiffusion

Madula, Mikuni

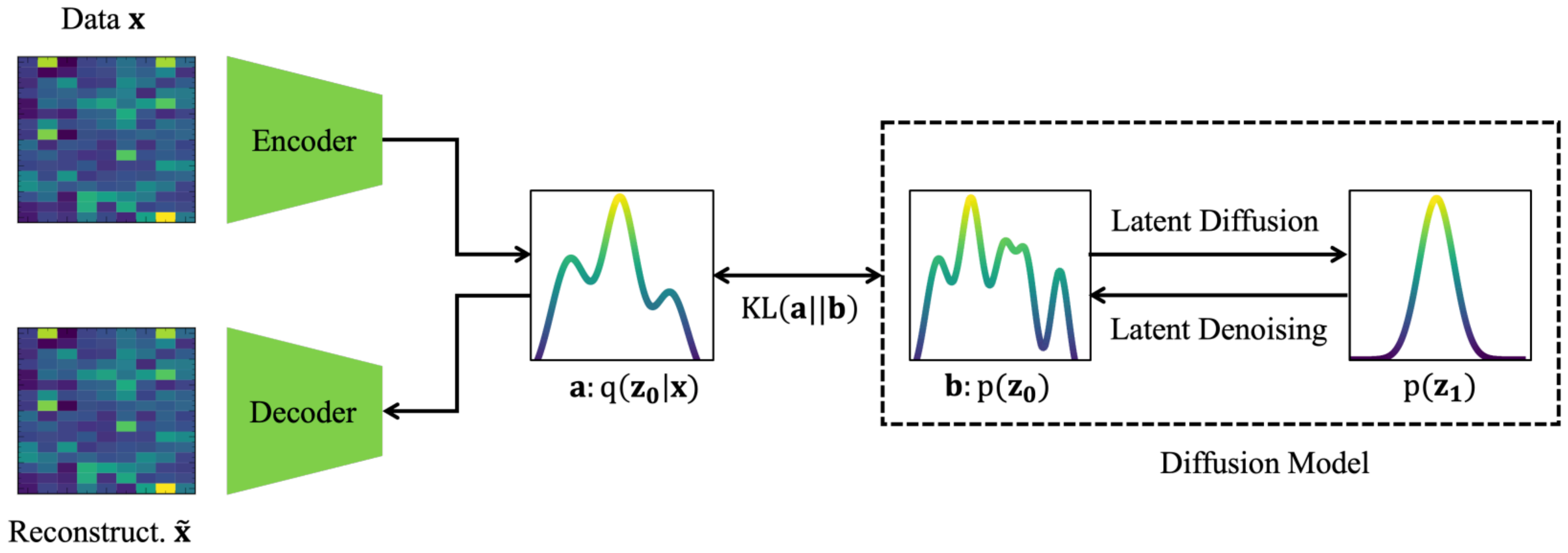
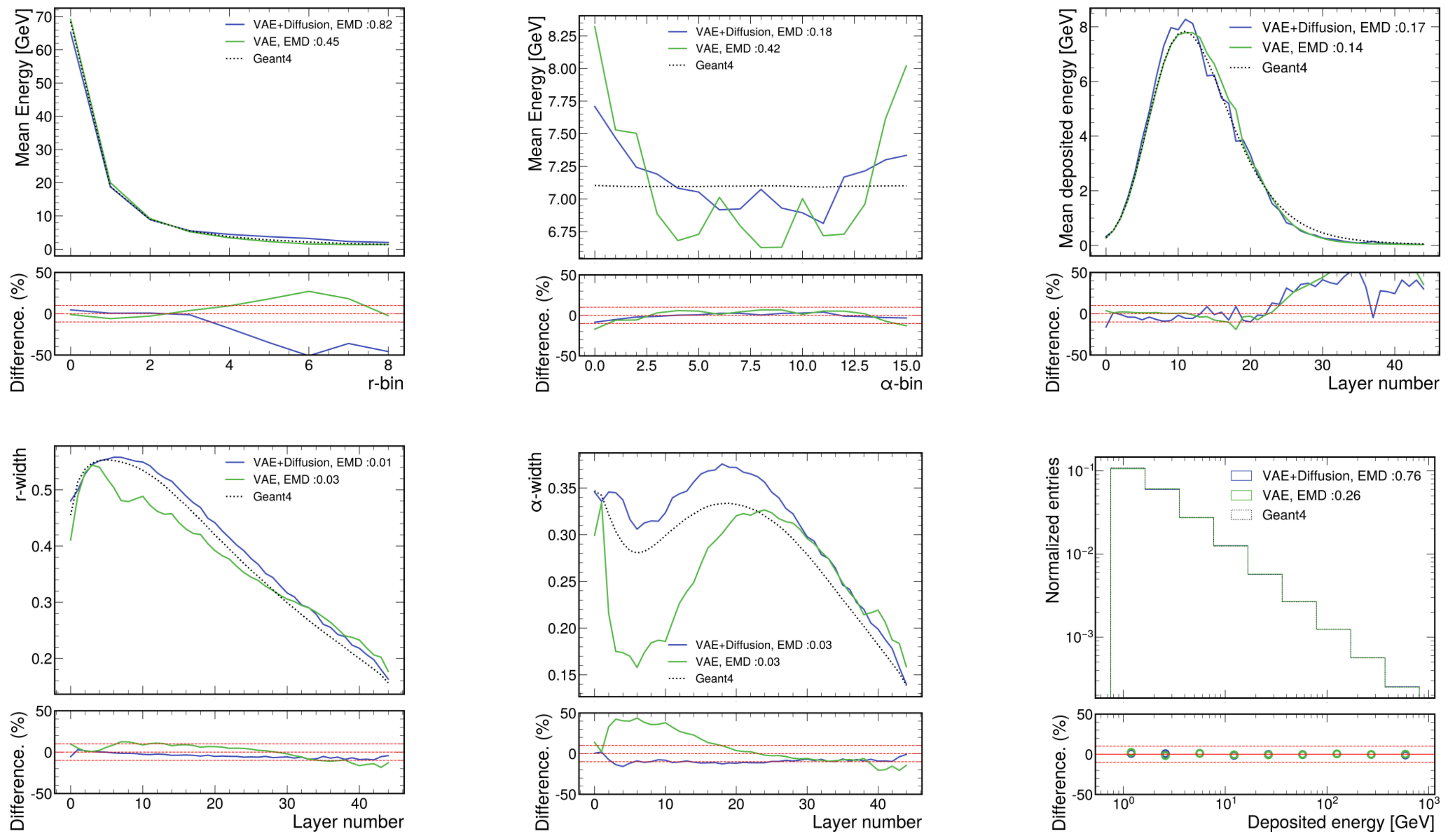


Figure 1: A schematic diagram of latent diffusion



ToDo

✓ UNet for CaloQVAE — Ian

✓ CaloQVAE w/ linear attention layers — further exploration needed

✓ Train current model with large RBM in Pegasus

✓ Associative mem in GAN — Coherent samples from Zephyr

✓ ATLAS dataset almost ready

✓ What if we add a UNet after our current CaloQVAE pipeline or after the encoder?

Work in progress

- ✓ Papers status
  - ➡PRX:
    - ➡ submit to arxiv after reviewing comments by Wojtek.
    - ➡Train 2 pegasus models 1) with fixed bug 2) with fixed bug + non-high T approximation. Discuss results and replcace in preprint.
    - ➡Evaluate model using CaloChallenge script and perhaps include in paper.
    - ➡Submit paper!
  - ➡Neurips (Not obliged to improve results)
    - ➡Train model with 9 decoders + fixed bug + non-high T approximation. Compare with results in Neurips paper.
    - ➡Evaluate model using CaloChallenge script
- ✓ Using the QPU w/ Pegasus and in Zephyr
  - ➡Dwave working on understanding better the T fluctuations due to flux biasing.
- ✓ High temperature gradient approximation for trained RBMs
  - ➡Removing this approximation improves results.
- ✦RBM and Diffusion model equivalence. Relaxation time in RBMs