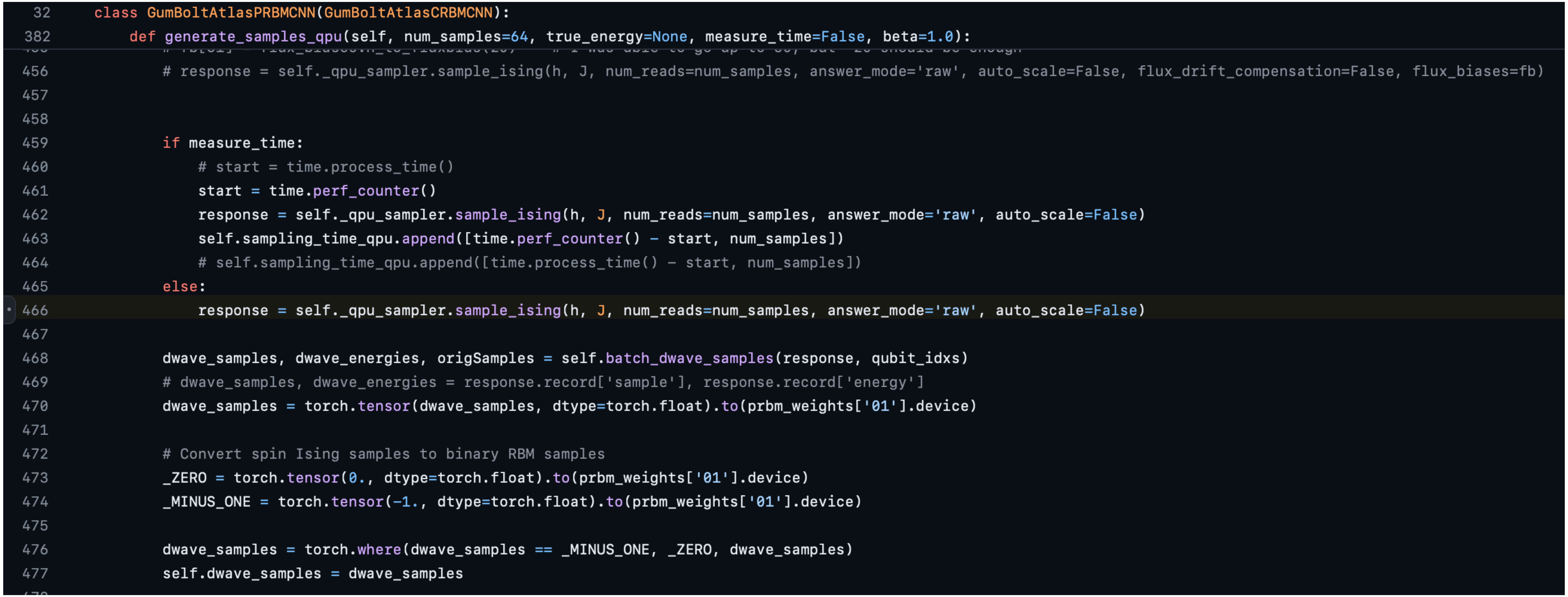
# QVAE w/ Pegasus & Zephyr

May 6

| | | Time/Shower [s] | |
|---|---|---|---|
| Dataset | Batch Size | CPU | GPU |
| 1 (photons)<br>(368 voxels) | 1 | 9.4 | 6.3 |
| | 10 | 2.0 | 0.6 |
| | 100 | 1.0 | 0.1 |
| 1 (pions)<br>(533 voxels) | 1 | 9.8 | 6.4 |
| | 10 | 2.0 | 0.6 |
| | 100 | 1.0 | 0.1 |
| 2 (electrons)<br>(6.5K voxels) | 1 | 14.8 | 6.2 |
| | 10 | 4.6 | 0.6 |
| | 100 | 4.0 | 0.2 |
| 3 (electrons)<br>(40.5K voxels) | 1 | 52.7 | 7.1 |
| | 10 | 44.1 | 2.6 |
| | 100 | - | 2.0 |

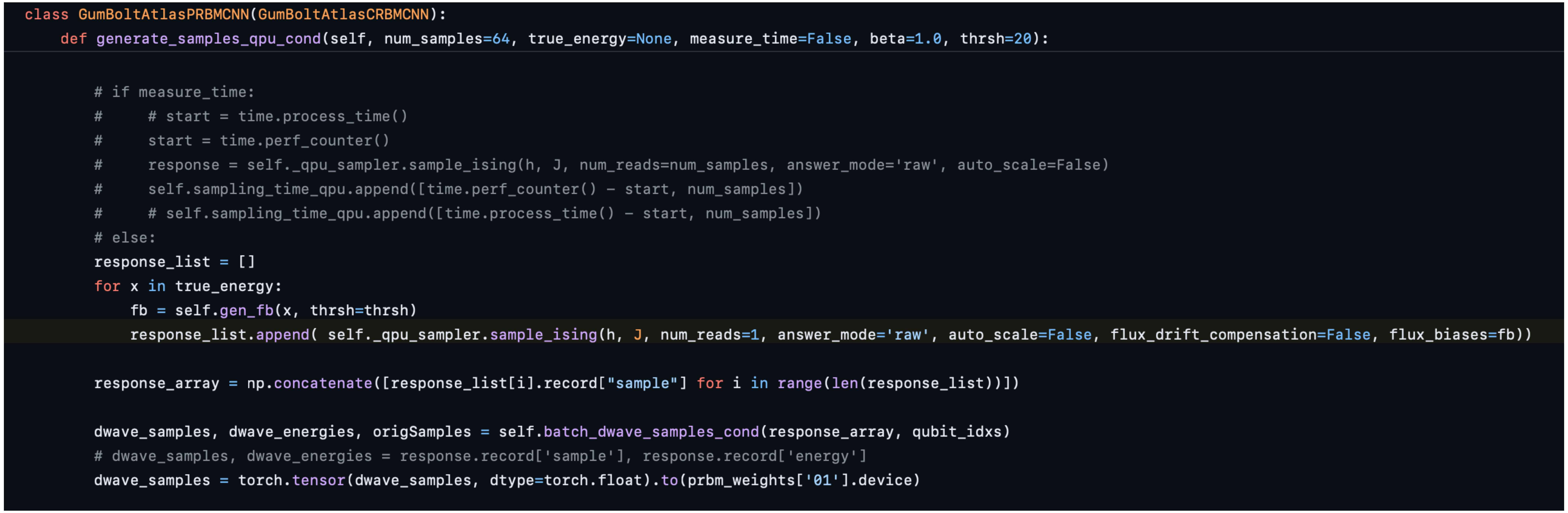
TABLE III. The shower generation time for **CaloDiffusion** on CPU and GPU for various batch sizes.

# Sampling CRBM w/ Pegasus

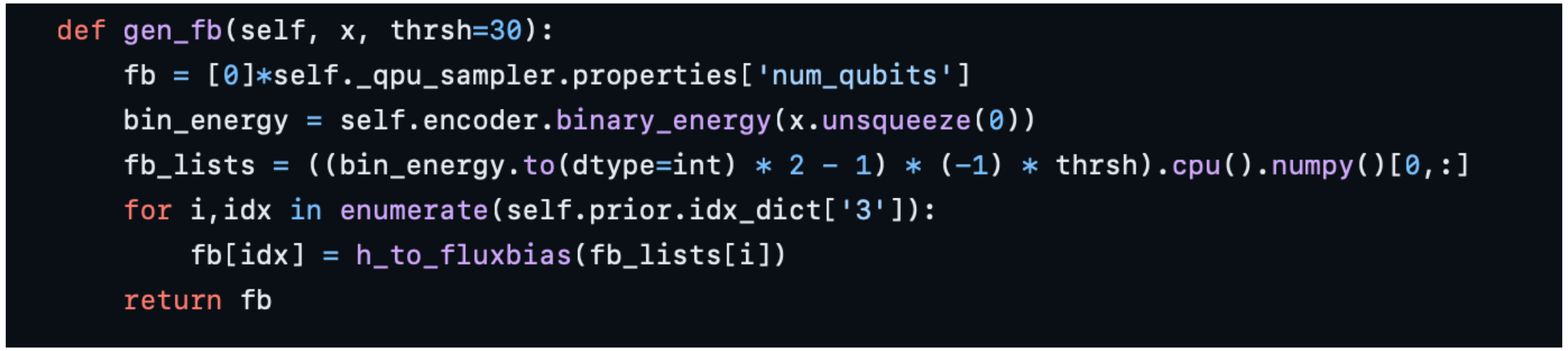


```python
class GumBoltAtlasPRBMCNN(GumBoltAtlasCRBMCNN):
    def generate_samples_qpu(self, num_samples=64, true_energy=None, measure_time=False, beta=1.0):
        # response = self._qpu_sampler.sample_ising(h, J, num_reads=num_samples, answer_mode='raw', auto_scale=False, flux_drift_compensation=False, flux_biases=fb)


        if measure_time:
            # start = time.process_time()
            start = time.perf_counter()
            response = self._qpu_sampler.sample_ising(h, J, num_reads=num_samples, answer_mode='raw', auto_scale=False)
            self.sampling_time_qpu.append([time.perf_counter() - start, num_samples])
            # self.sampling_time_qpu.append([time.process_time() - start, num_samples])
        else:
            response = self._qpu_sampler.sample_ising(h, J, num_reads=num_samples, answer_mode='raw', auto_scale=False)

        dwave_samples, dwave_energies, origSamples = self.batch_dwave_samples(response, qubit_idxs)
        # dwave_samples, dwave_energies = response.record['sample'], response.record['energy']
        dwave_samples = torch.tensor(dwave_samples, dtype=torch.float).to(prbm_weights['01'].device)

        # Convert spin Ising samples to binary RBM samples
        _ZERO = torch.tensor(0., dtype=torch.float).to(prbm_weights['01'].device)
        _MINUS_ONE = torch.tensor(-1., dtype=torch.float).to(prbm_weights['01'].device)

        dwave_samples = torch.where(dwave_samples == _MINUS_ONE, _ZERO, dwave_samples)
        self.dwave_samples = dwave_samples
```

# Sampling CRBM w/ Pegasus



```
class GumBoltAtlasPRBMCNN(GumBoltAtlasCRBMCNN):
    def generate_samples_qpu_cond(self, num_samples=64, true_energy=None, measure_time=False, beta=1.0, thrsh=20):

        # if measure_time:
        #     # start = time.process_time()
        #     start = time.perf_counter()
        #     response = self._qpu_sampler.sample_ising(h, J, num_reads=num_samples, answer_mode='raw', auto_scale=False)
        #     self.sampling_time_qpu.append([time.perf_counter() - start, num_samples])
        #     # self.sampling_time_qpu.append([time.process_time() - start, num_samples])
        # else:
        response_list = []
        for x in true_energy:
            fb = self.gen_fb(x, thrsh=thrsh)
            response_list.append( self._qpu_sampler.sample_ising(h, J, num_reads=1, answer_mode='raw', auto_scale=False, flux_drift_compensation=False, flux_biases=fb))

        response_array = np.concatenate([response_list[i].record["sample"] for i in range(len(response_list))])

        dwave_samples, dwave_energies, origSamples = self.batch_dwave_samples_cond(response_array, qubit_idxs)
        # dwave_samples, dwave_energies = response.record['sample'], response.record['energy']
        dwave_samples = torch.tensor(dwave_samples, dtype=torch.float).to(prbm_weights['01'].device)
```



```
def gen_fb(self, x, thrsh=30):
    fb = [0]*self._qpu_sampler.properties['num_qubits']
    bin_energy = self.encoder.binary_energy(x.unsqueeze(0))
    fb_lists = ((bin_energy.to(dtype=int) * 2 - 1) * (-1) * thrsh).cpu().numpy()[0,:]
    for i,idx in enumerate(self.prior.idx_dict['3']):
        fb[idx] = h_to_fluxbias(fb_lists[i])
    return fb
```

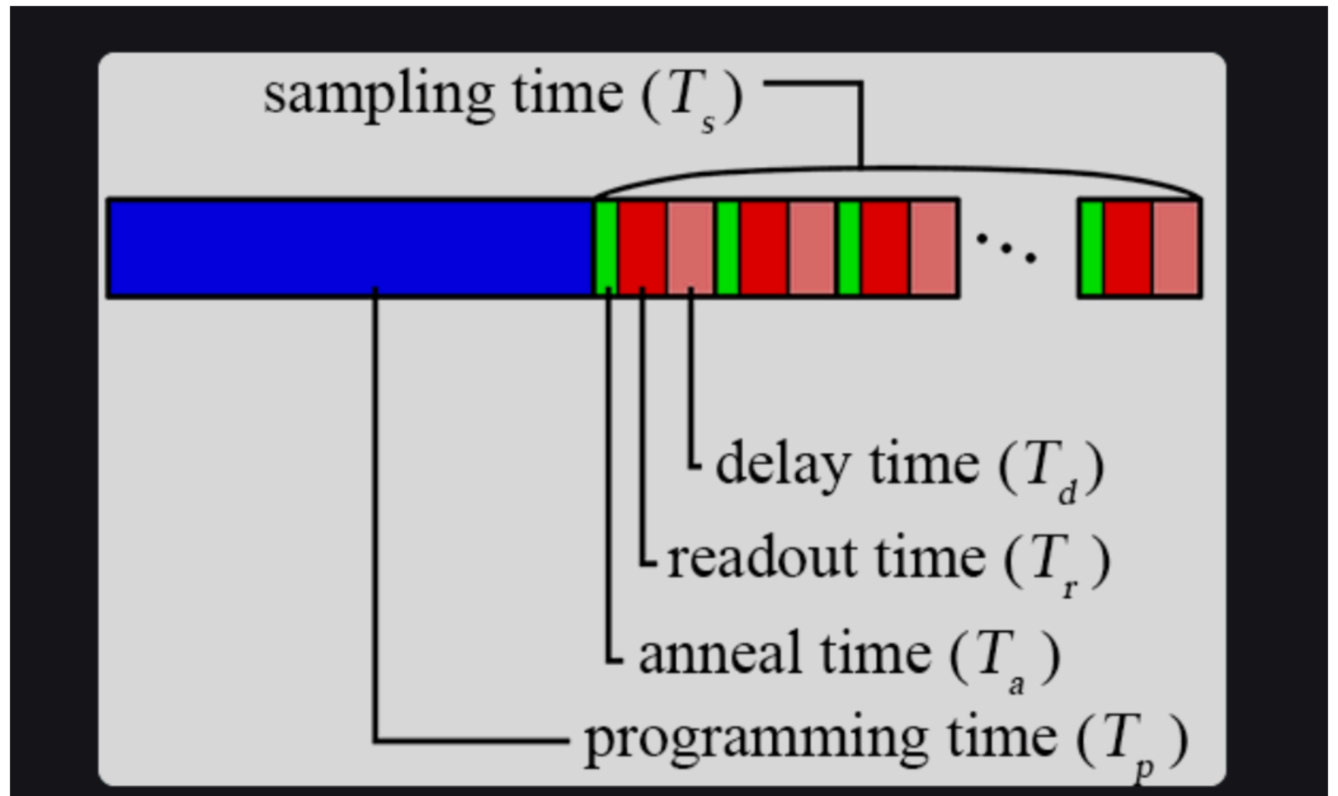
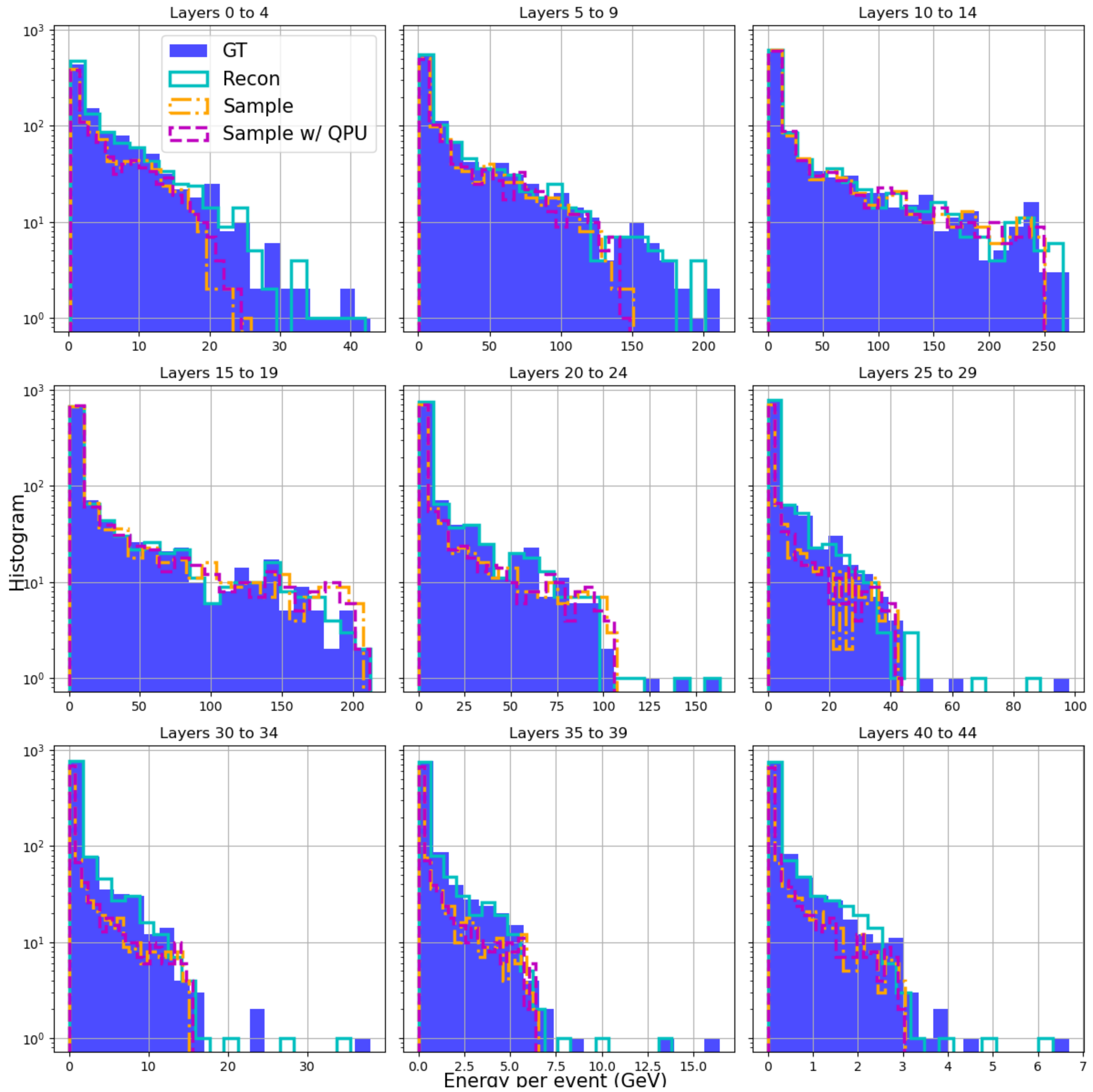
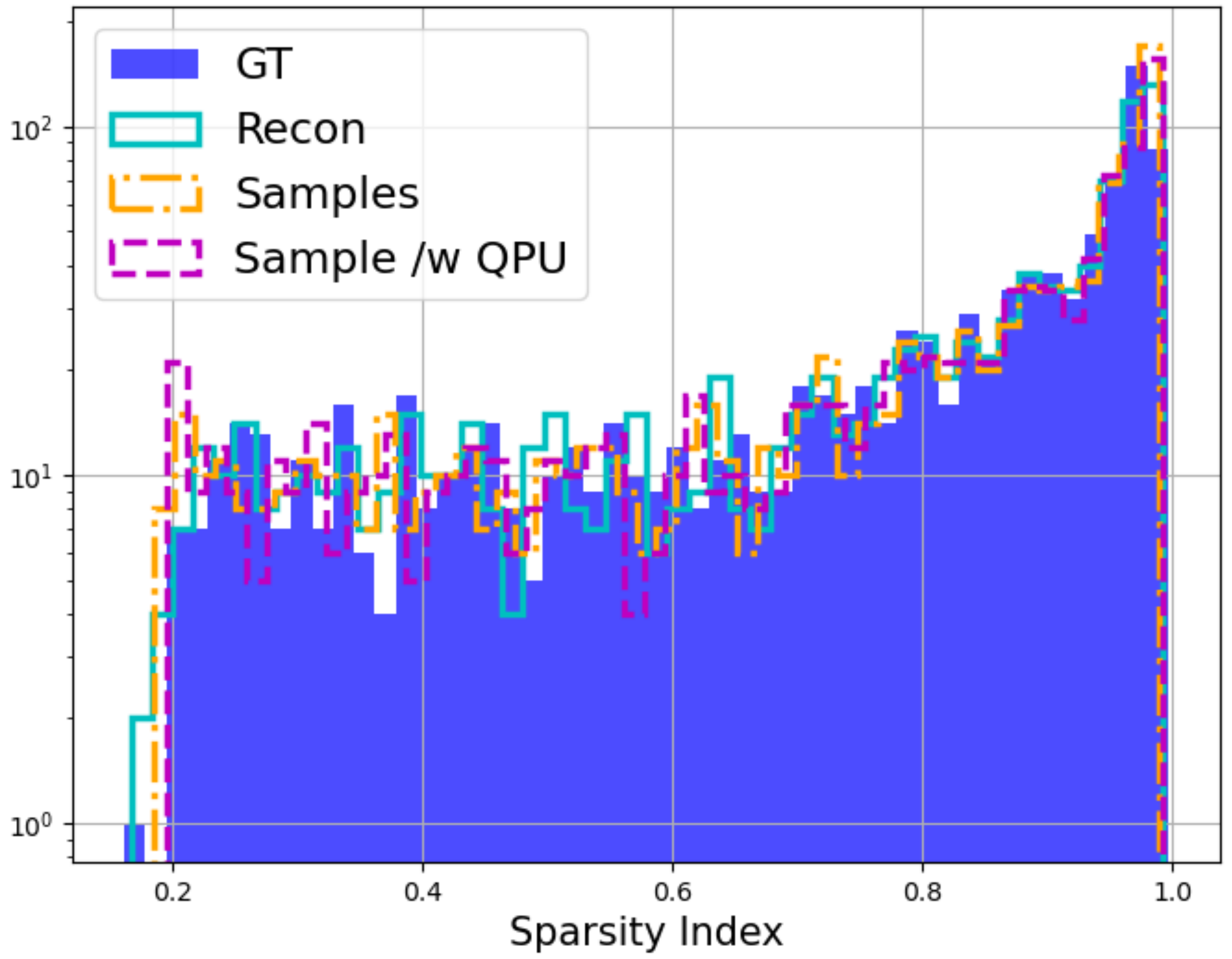
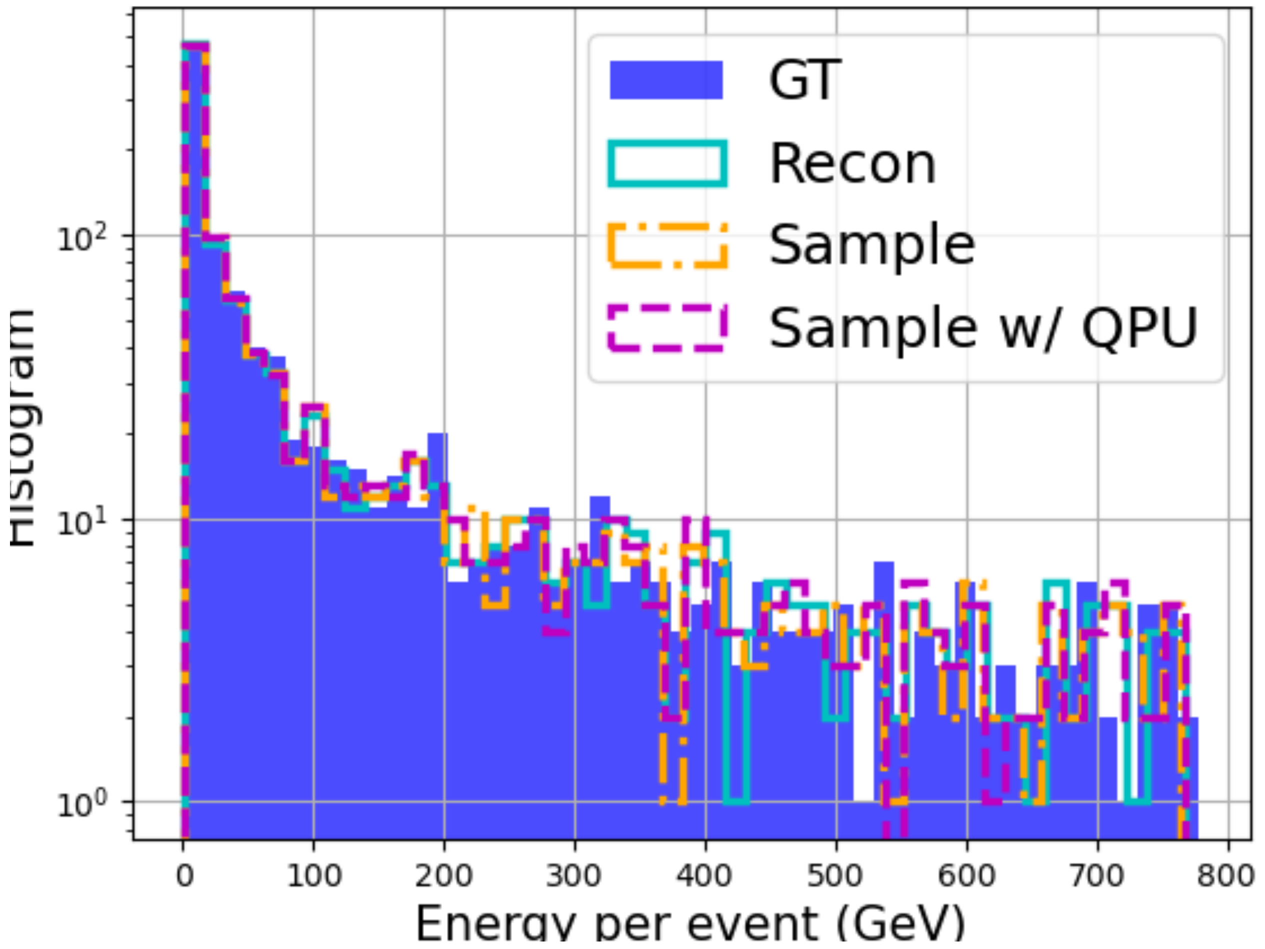
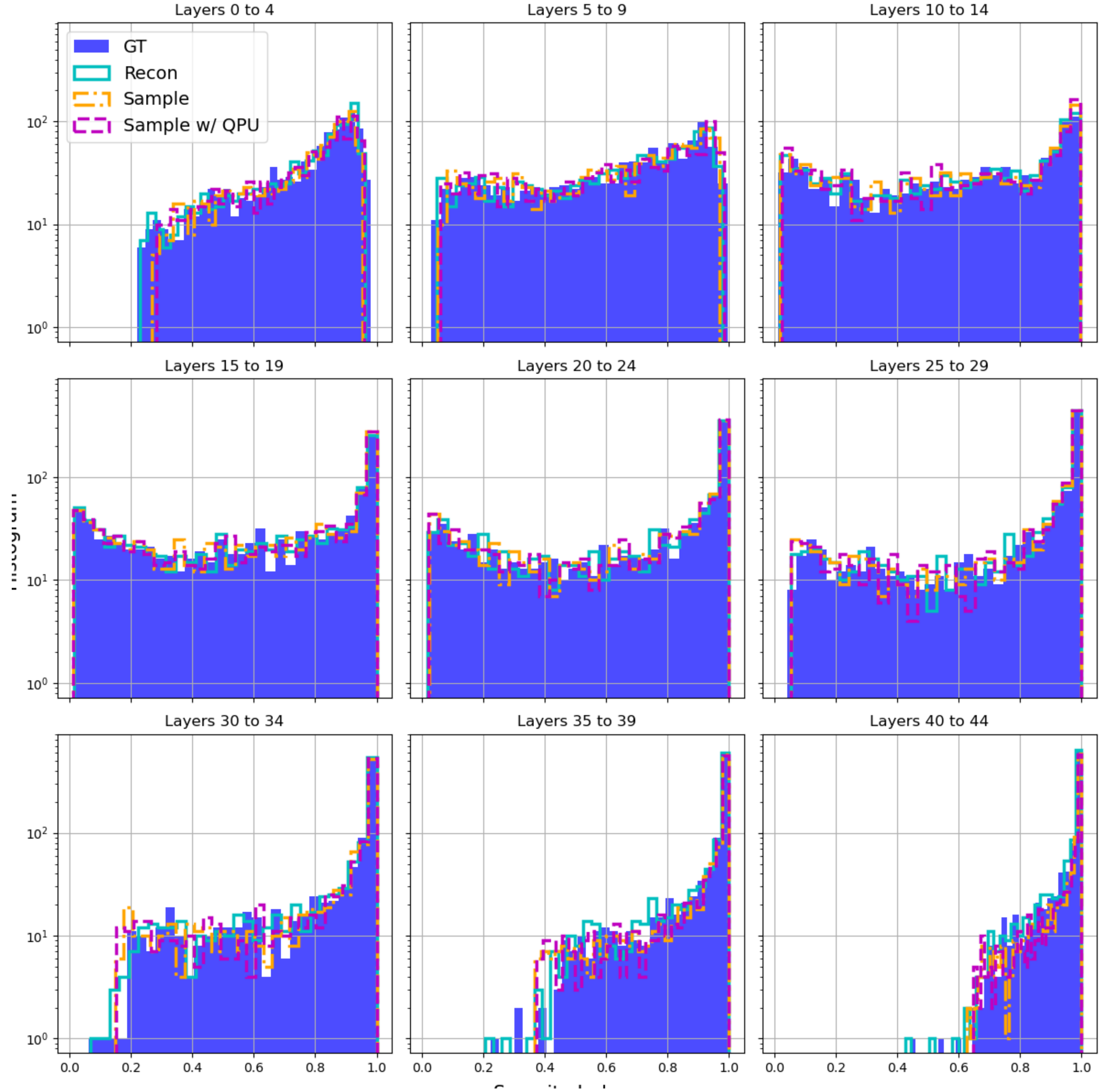


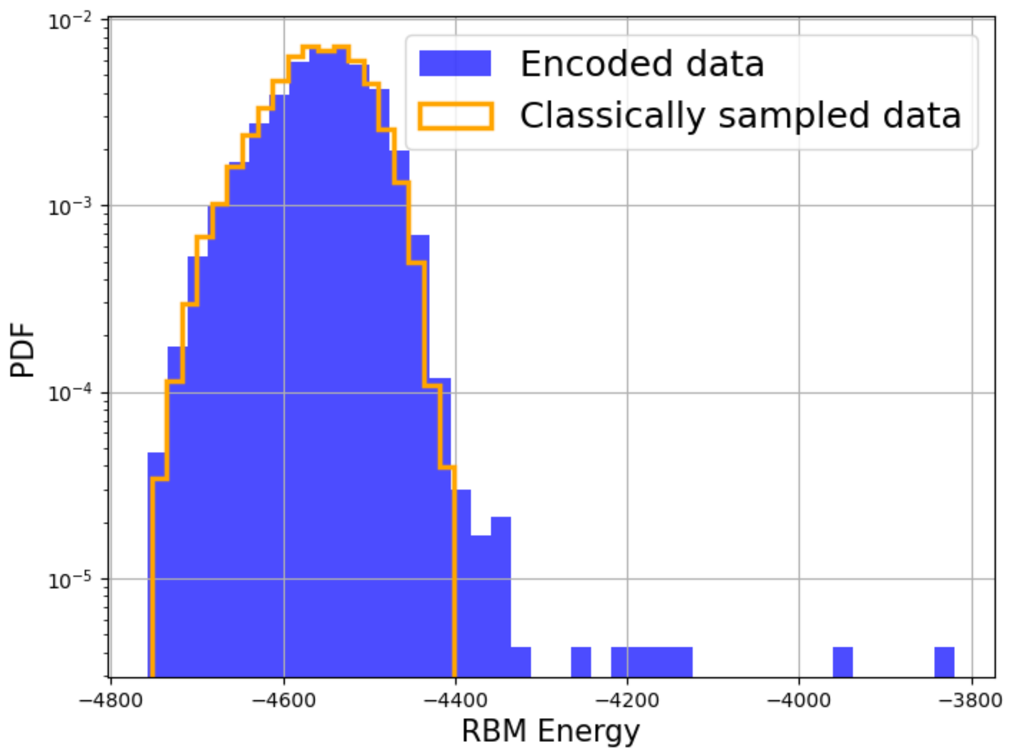
*Fig. 109* Detail of QPU access time. #

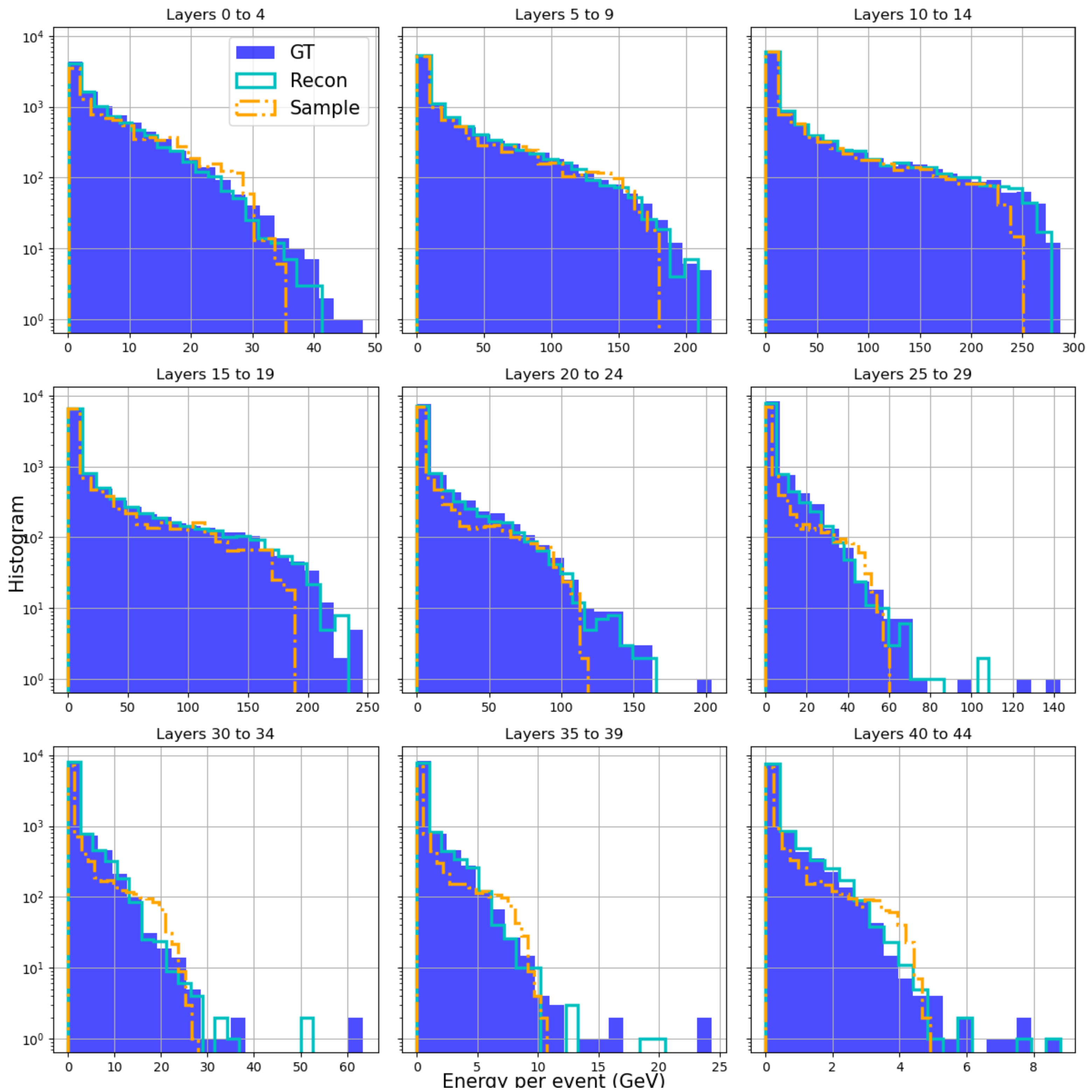

Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
GT
Recon
Sample
Sample w/ QPU
Histogram
Energy per event (GeV)

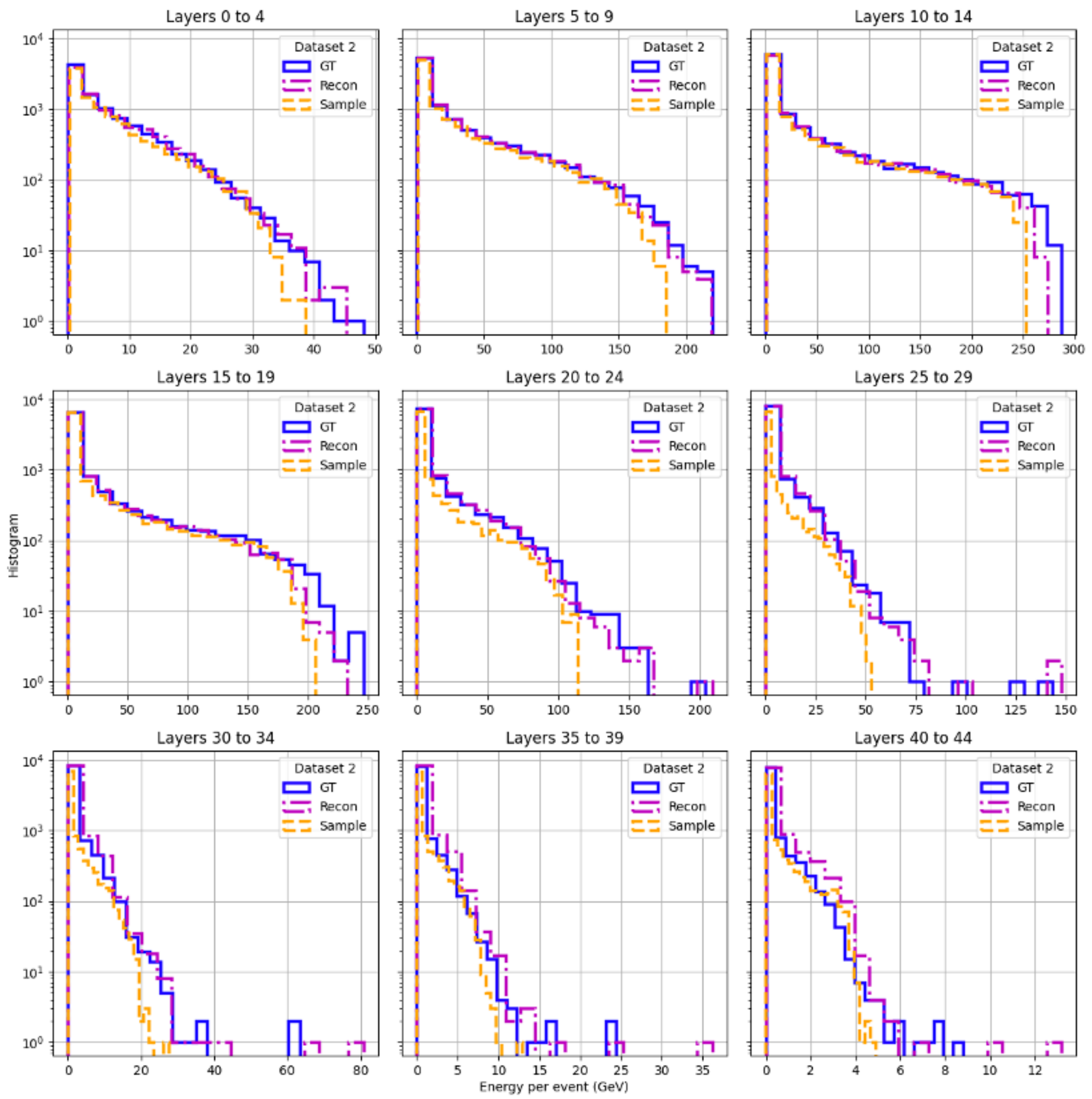

GT
Recon
Samples
Sample /w QPU
Sparsity Index
0.2
0.4
0.6
0.8
1.0

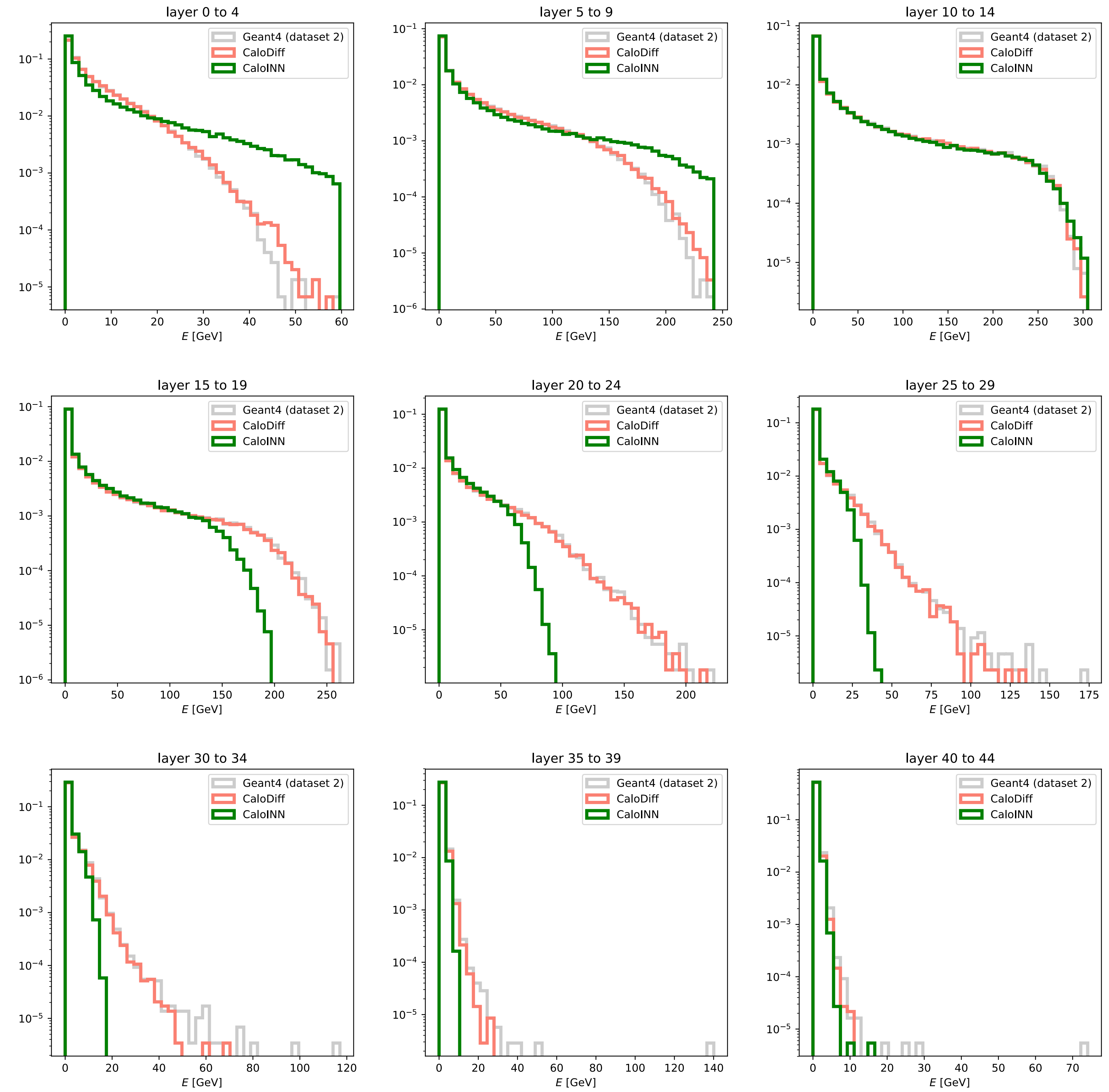

GT
Recon
Sample
Sample w/ QPU
Histogram
Energy per event (GeV)
0
100
200
300
400
500
600
700
800


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
Sample w/ QPU

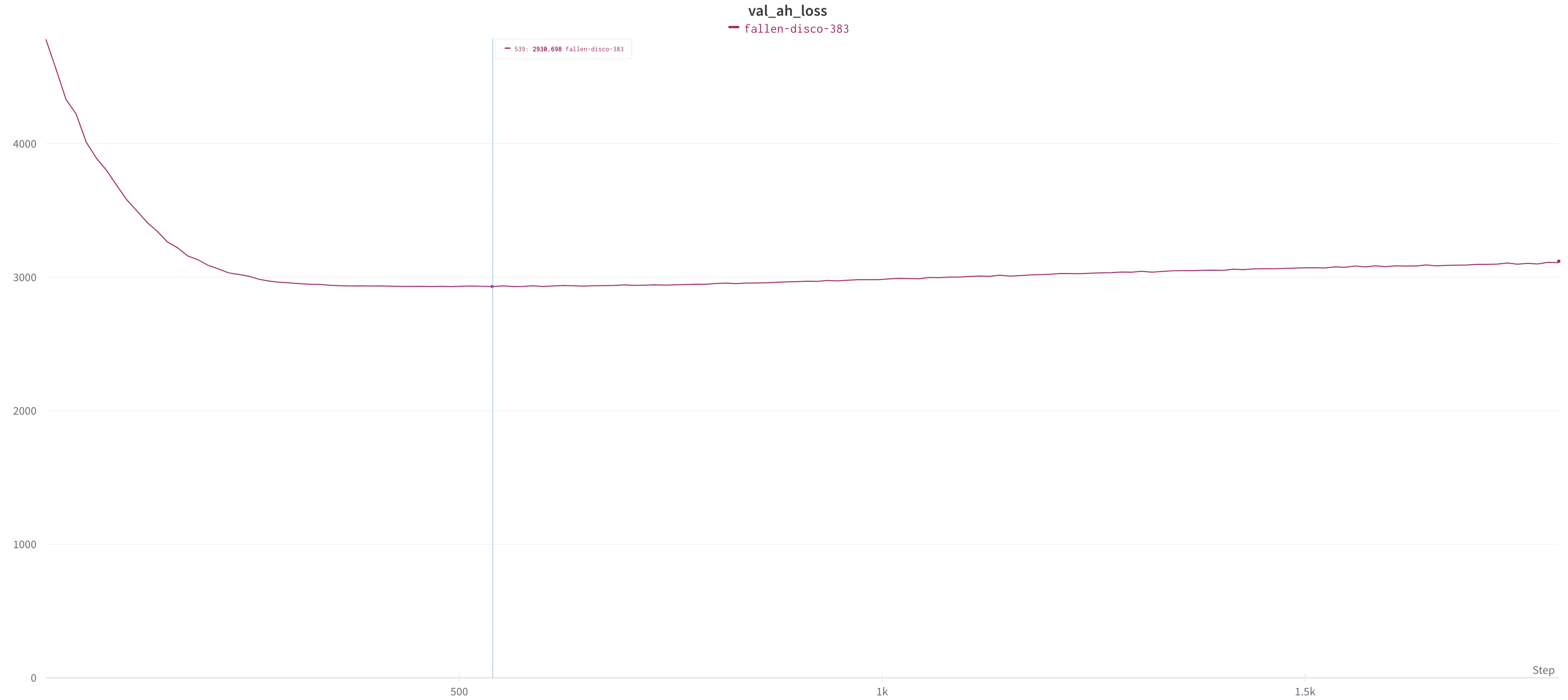

Encoded data
Classically sampled data
PDF
RBM Energy
−4800
−4600
−4400
−4200
−4000
−3800


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)

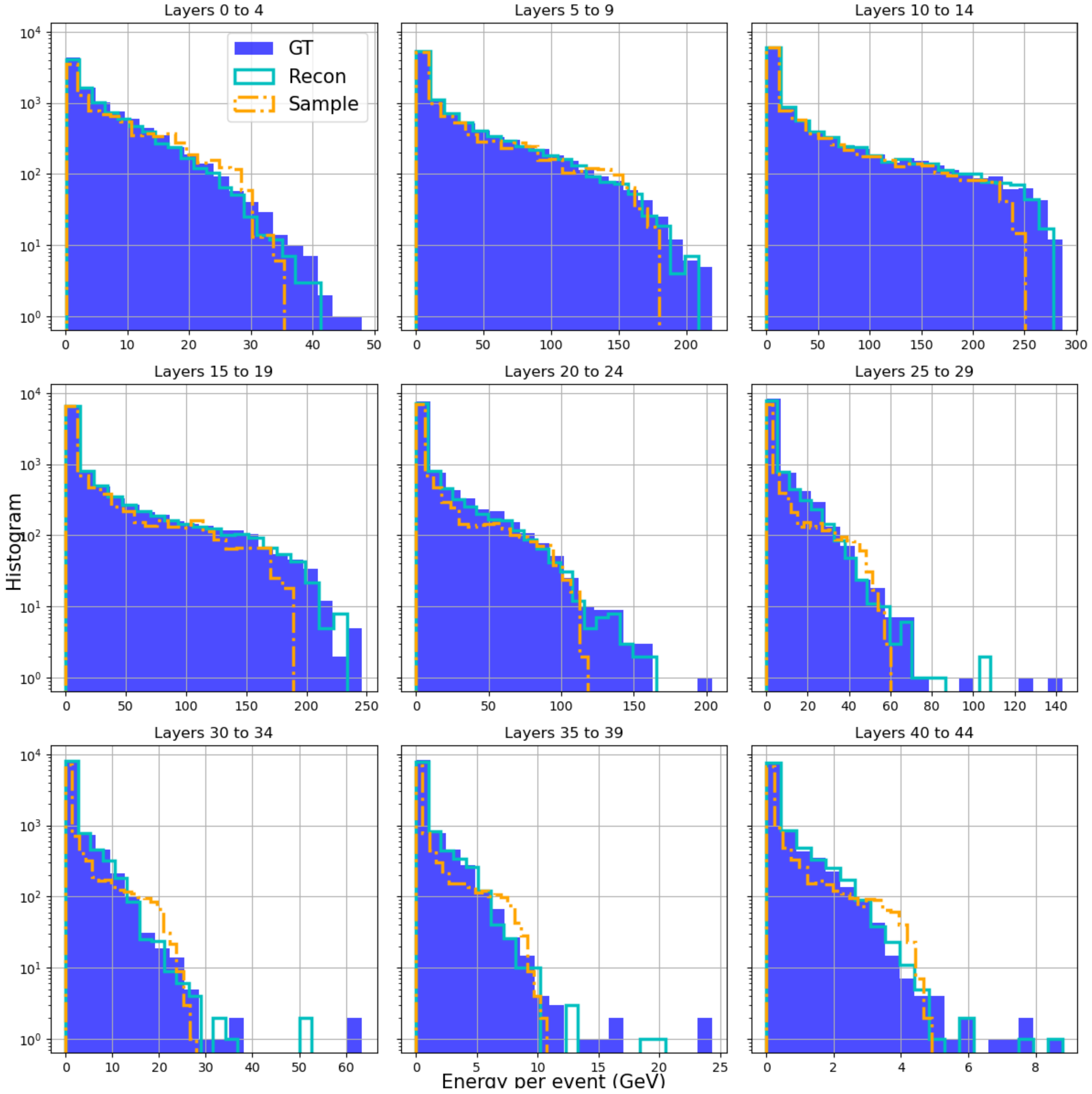

Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
Dataset 2
GT
Recon
Sample
Histogram
Energy per event (GeV)


layer 0 to 4
layer 5 to 9
layer 10 to 14
layer 15 to 19
layer 20 to 24
layer 25 to 29
layer 30 to 34
layer 35 to 39
layer 40 to 44
Geant4 (dataset 2)
CaloDiff
CaloINN
E [GeV]

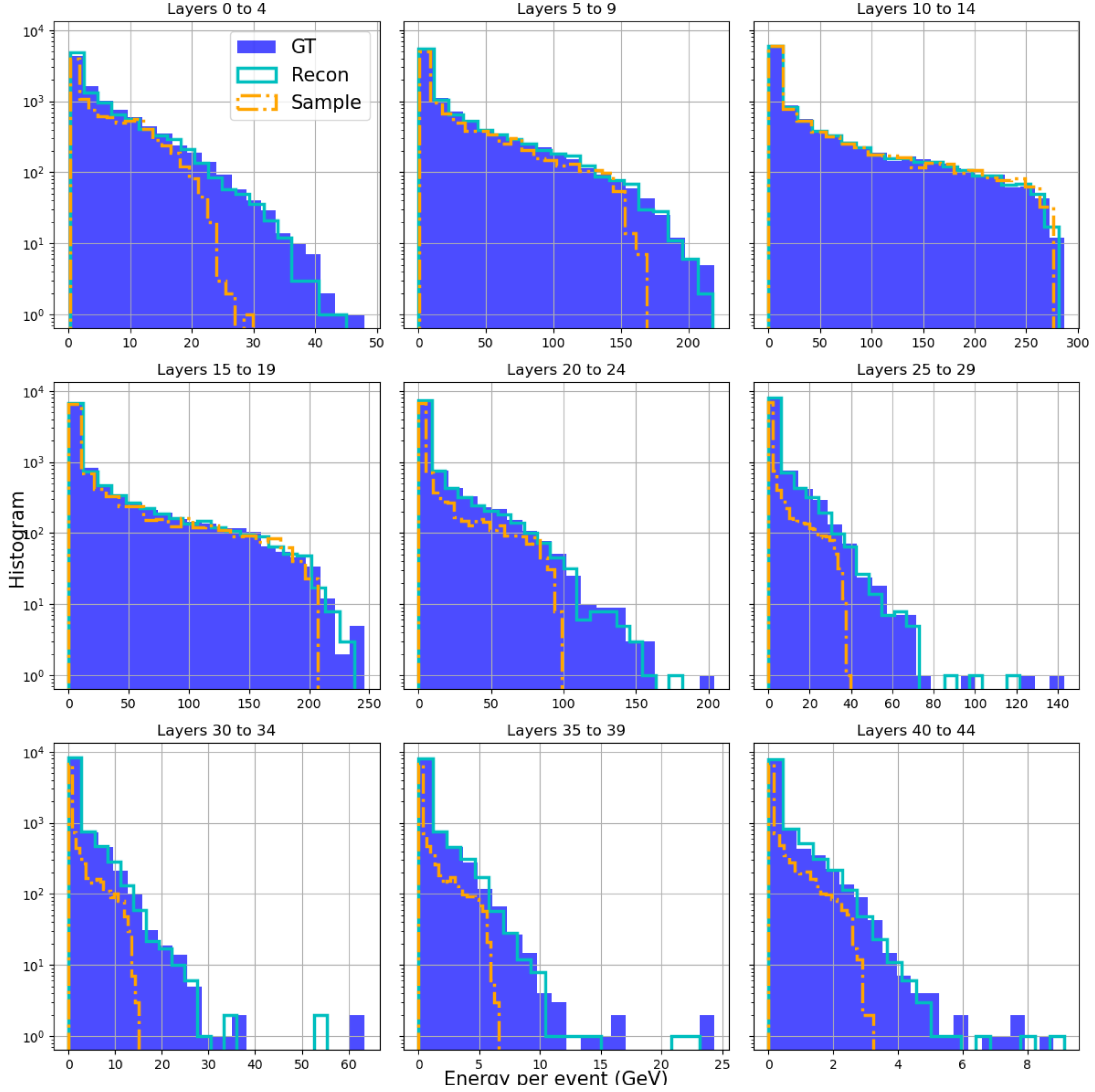
E inc of all 2


val_ah_loss
fallen-disco-383
539: 2930.698 fallen-disco-383
4000
3000
2000
1000
0
500
1k
1.5k
Step

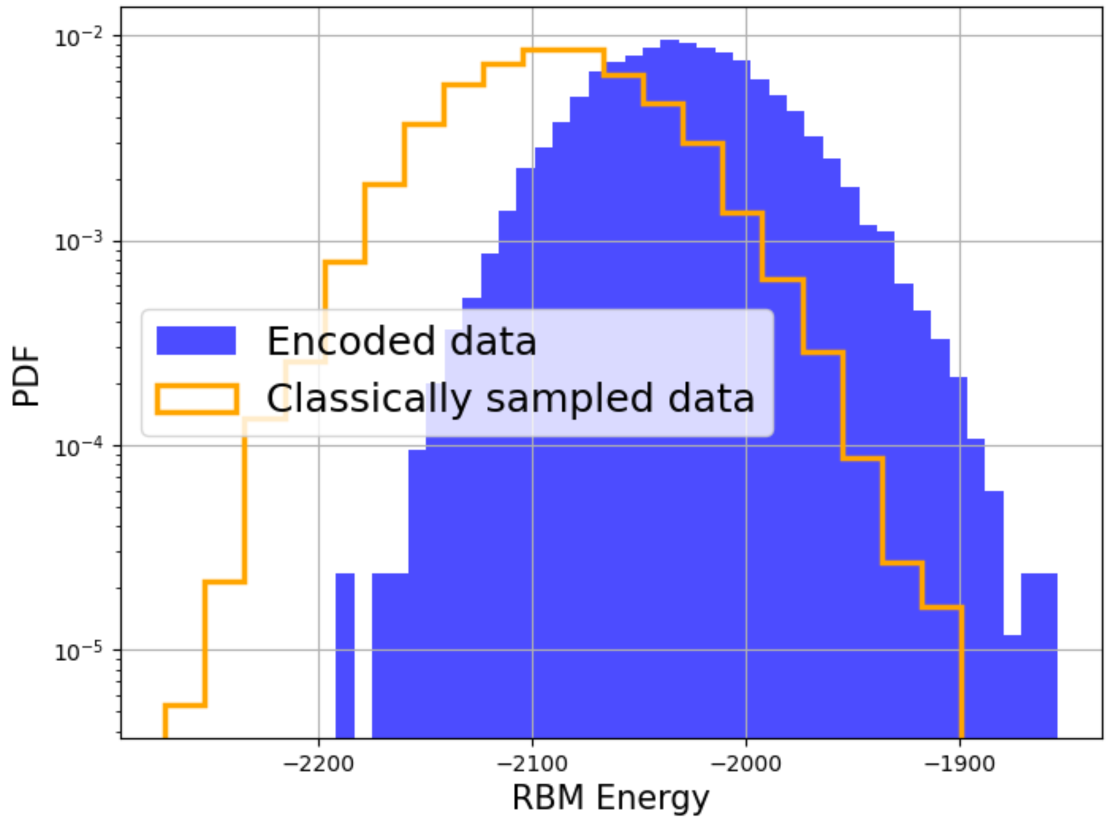
# Best

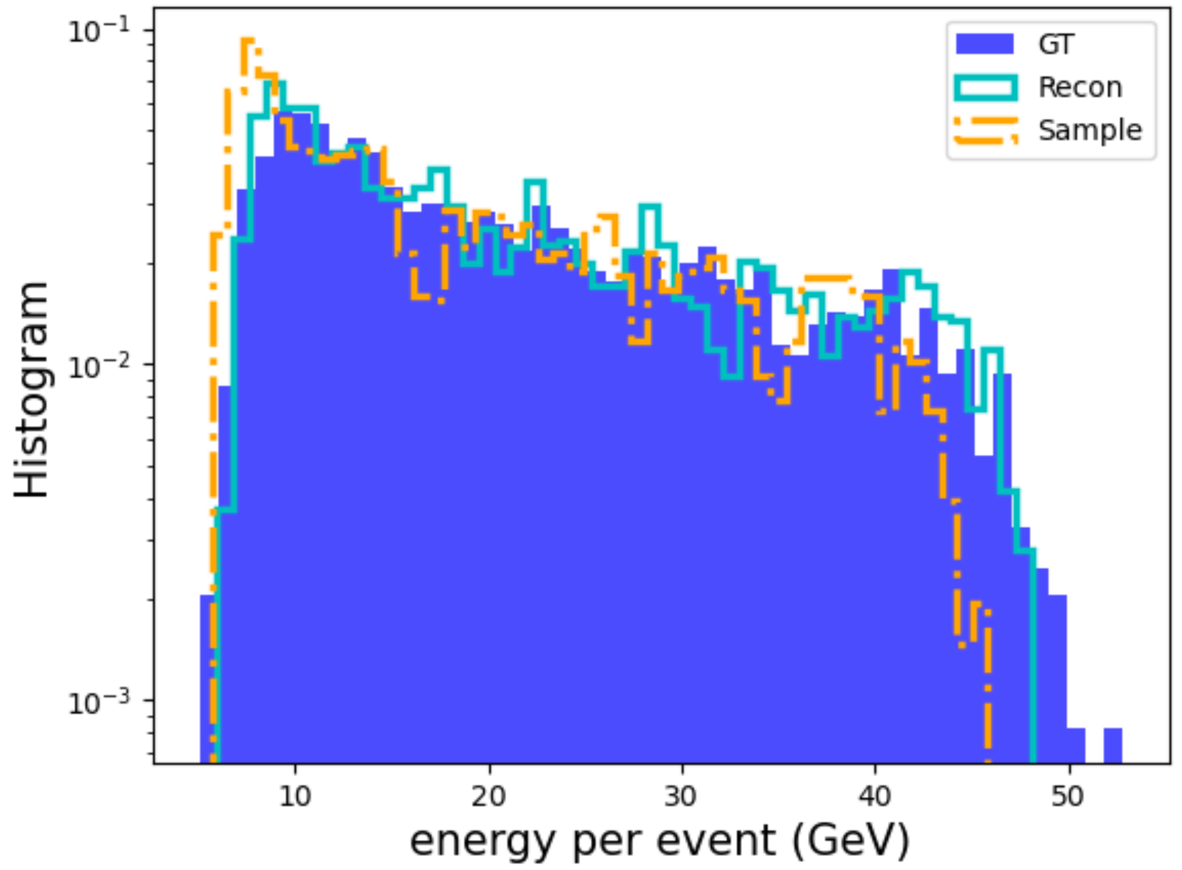

Layers 0 to 4
Layers 5 to 9
Layers 10 to 14
Layers 15 to 19
Layers 20 to 24
Layers 25 to 29
Layers 30 to 34
Layers 35 to 39
Layers 40 to 44
GT
Recon
Sample
Histogram
Energy per event (GeV)
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
0 10 20 30 40 50
0 50 100 150 200
0 50 100 150 200 250 300
0 50 100 150 200 250
0 50 100 150 200
0 20 40 60 80 100 120 140
0 10 20 30 40 50 60
0 5 10 15 20 25
0 2 4 6 8

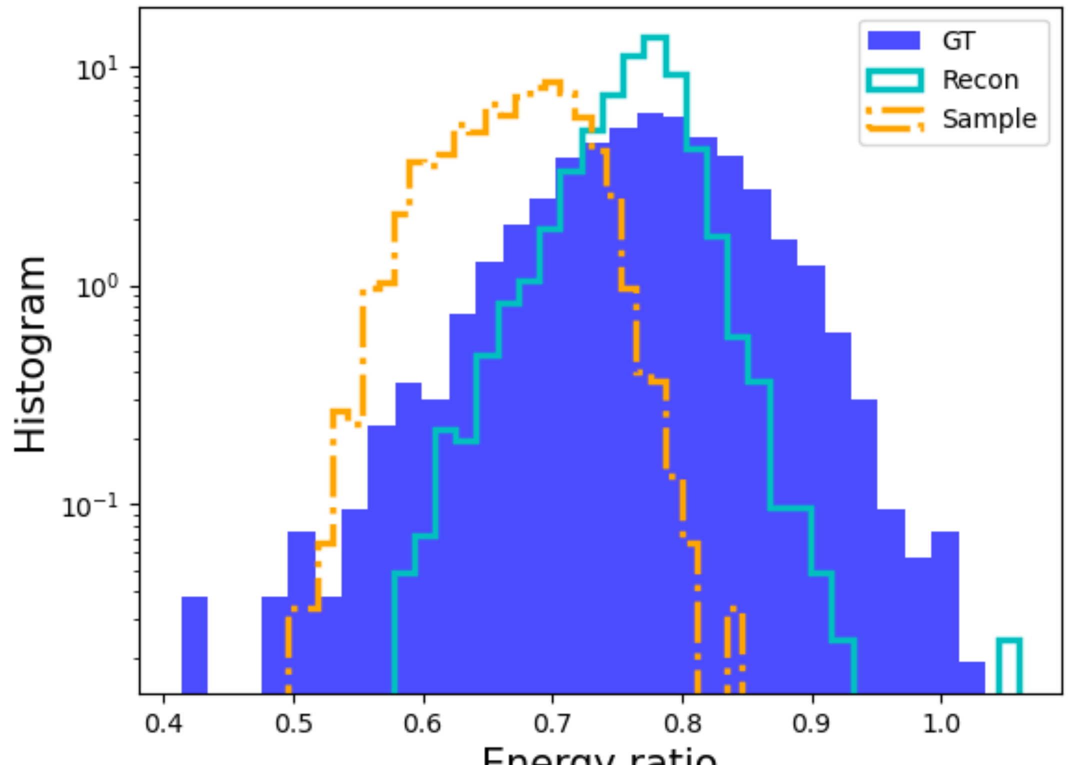
# Latest


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)
$10^0$
$10^1$
$10^2$
$10^3$
$10^4$
0 10 20 30 40 50
0 50 100 150 200
0 50 100 150 200 250 300
0 50 100 150 200 250
0 50 100 150 200
0 20 40 60 80 100 120 140
0 10 20 30 40 50 60
0 5 10 15 20 25
0 2 4 6 8

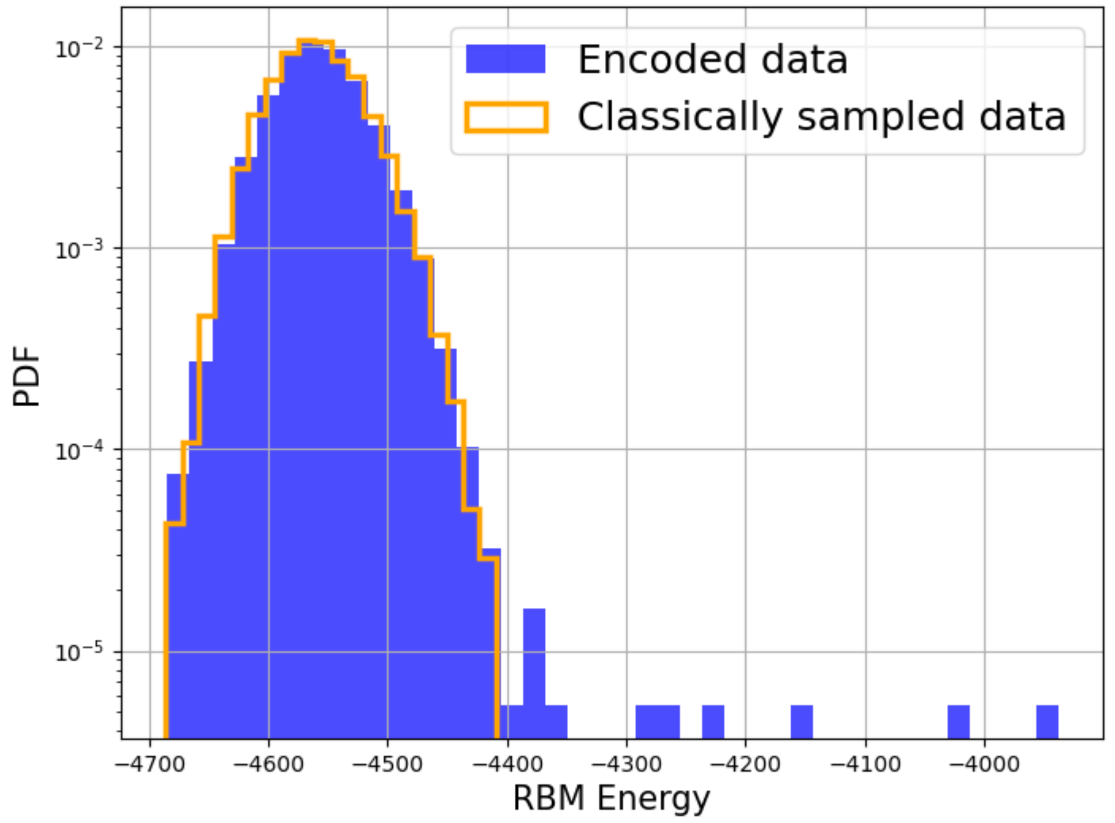
# Best

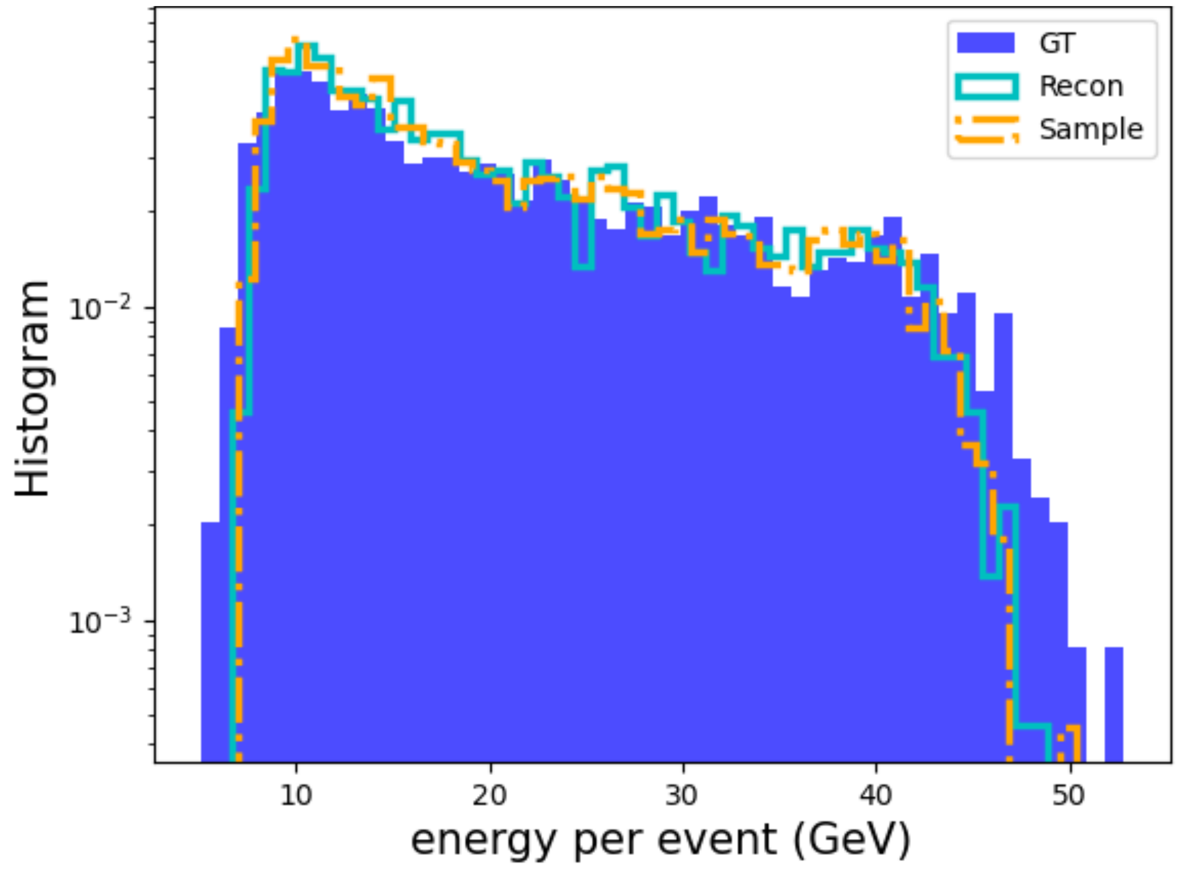

$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
PDF
Encoded data
Classically sampled data
−2200
−2100
−2000
−1900
RBM Energy

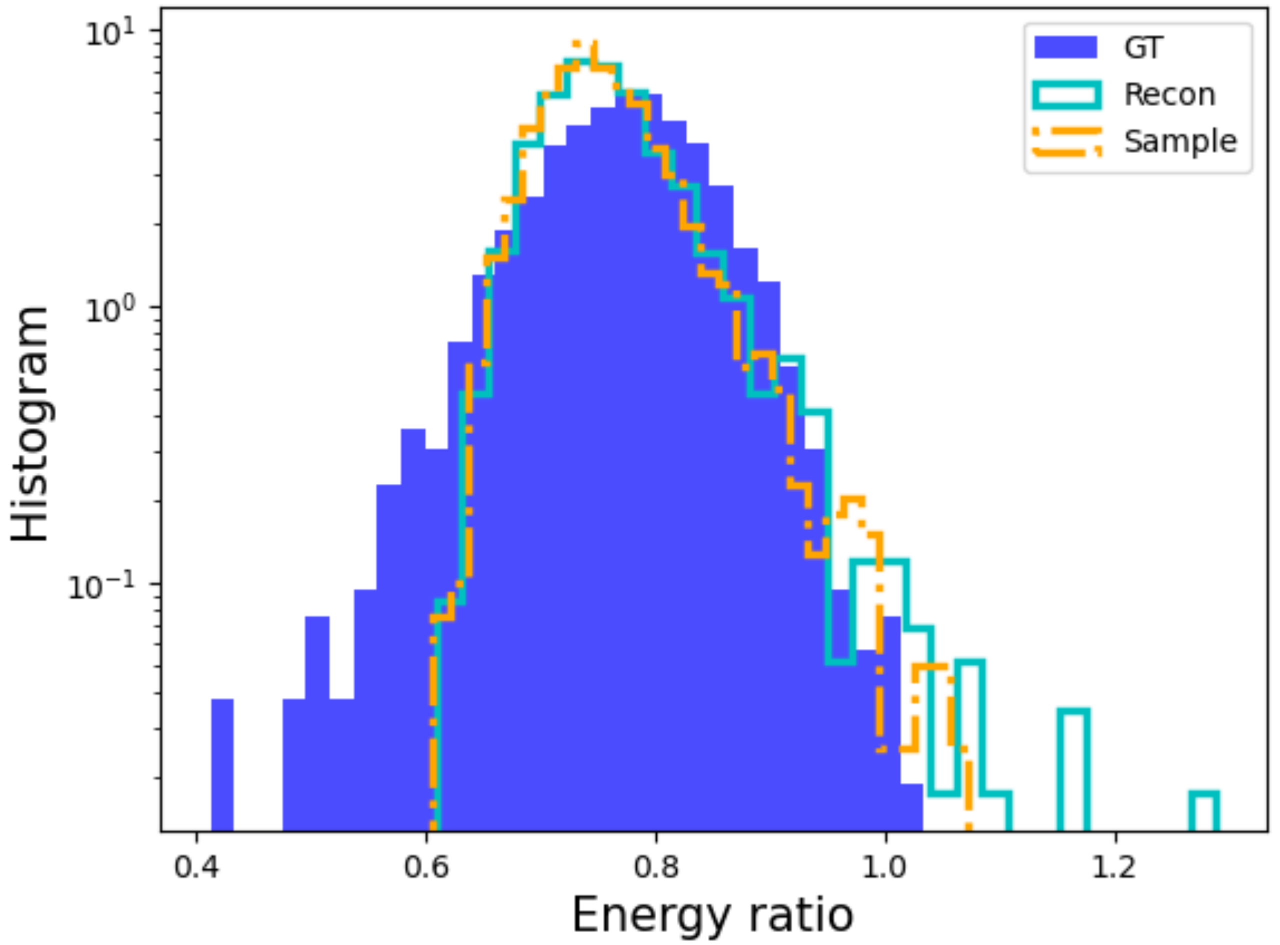

$10^{-1}$
$10^{-2}$
$10^{-3}$
Histogram
GT
Recon
Sample
10
20
30
40
50
energy per event (GeV)


$10^{1}$
$10^{0}$
$10^{-1}$
Histogram
GT
Recon
Sample
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Energy ratio

# Latest


$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
PDF
Encoded data
Classically sampled data
−4700
−4600
−4500
−4400
−4300
−4200
−4100
−4000
RBM Energy

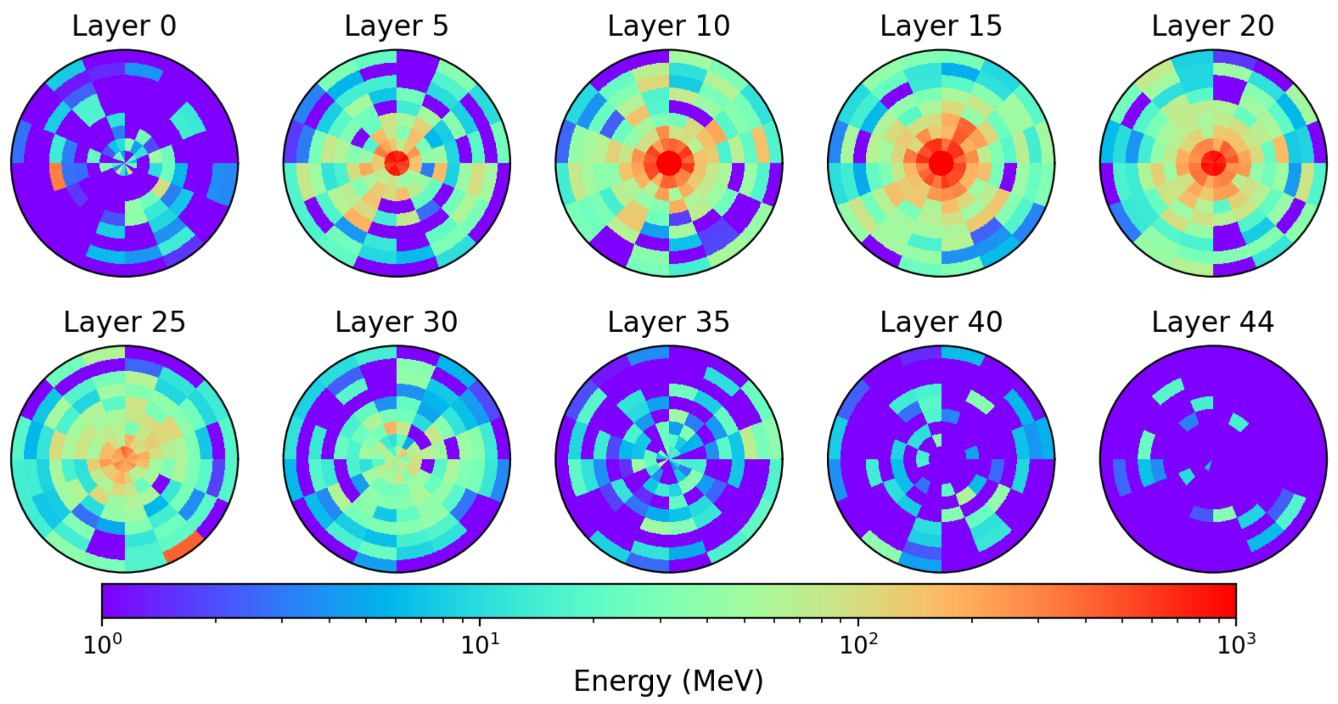

$10^{-2}$
$10^{-3}$
Histogram
GT
Recon
Sample
10
20
30
40
50
energy per event (GeV)

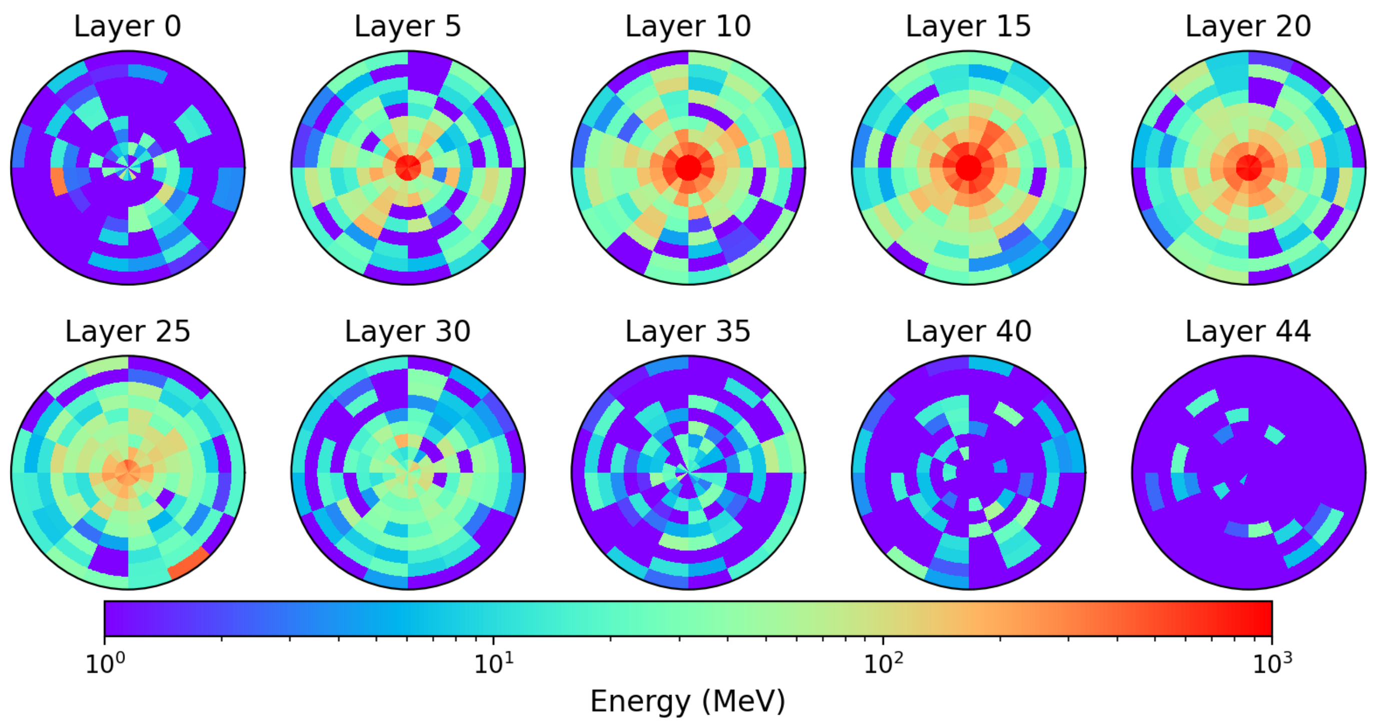

$10^{1}$
$10^{0}$
$10^{-1}$
Histogram
GT
Recon
Sample
0.4
0.6
0.8
1.0
1.2
Energy ratio

Best

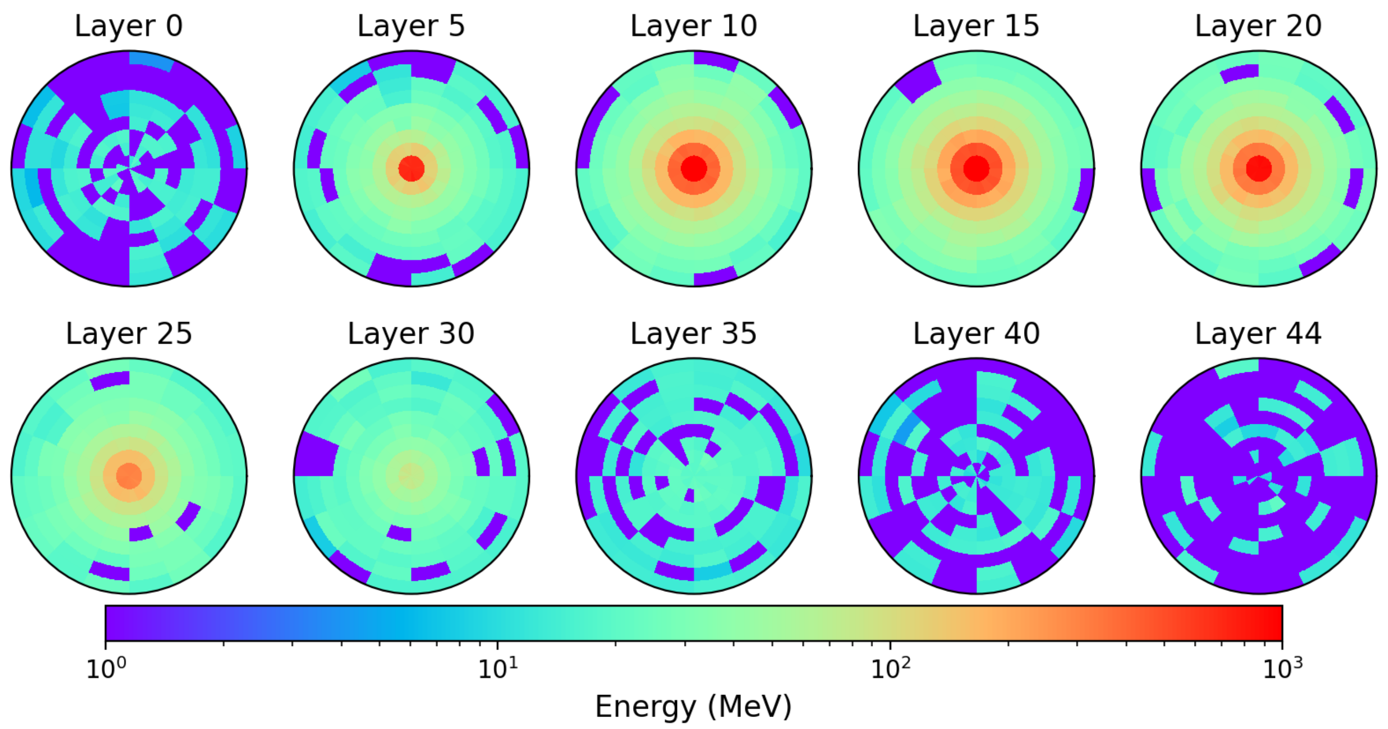
Latest

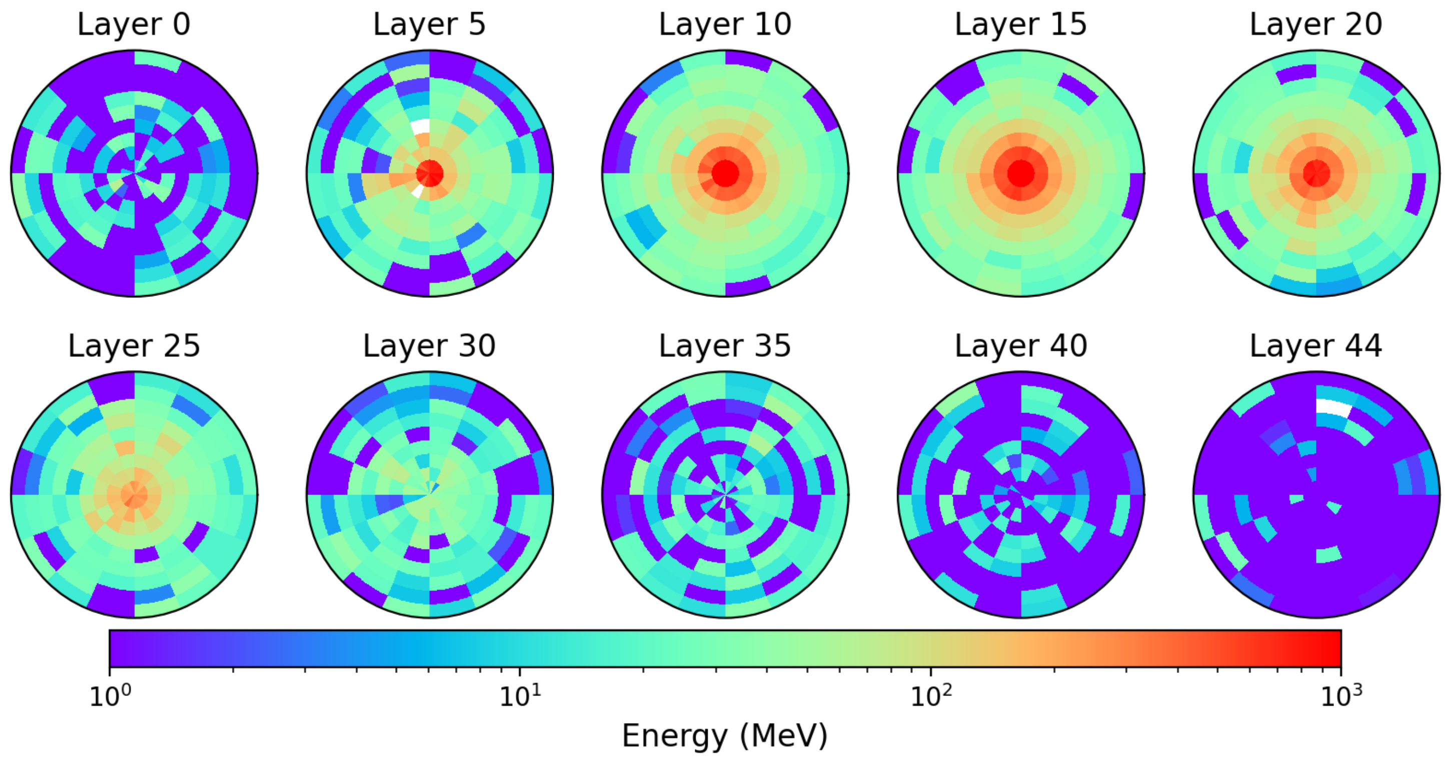
GT



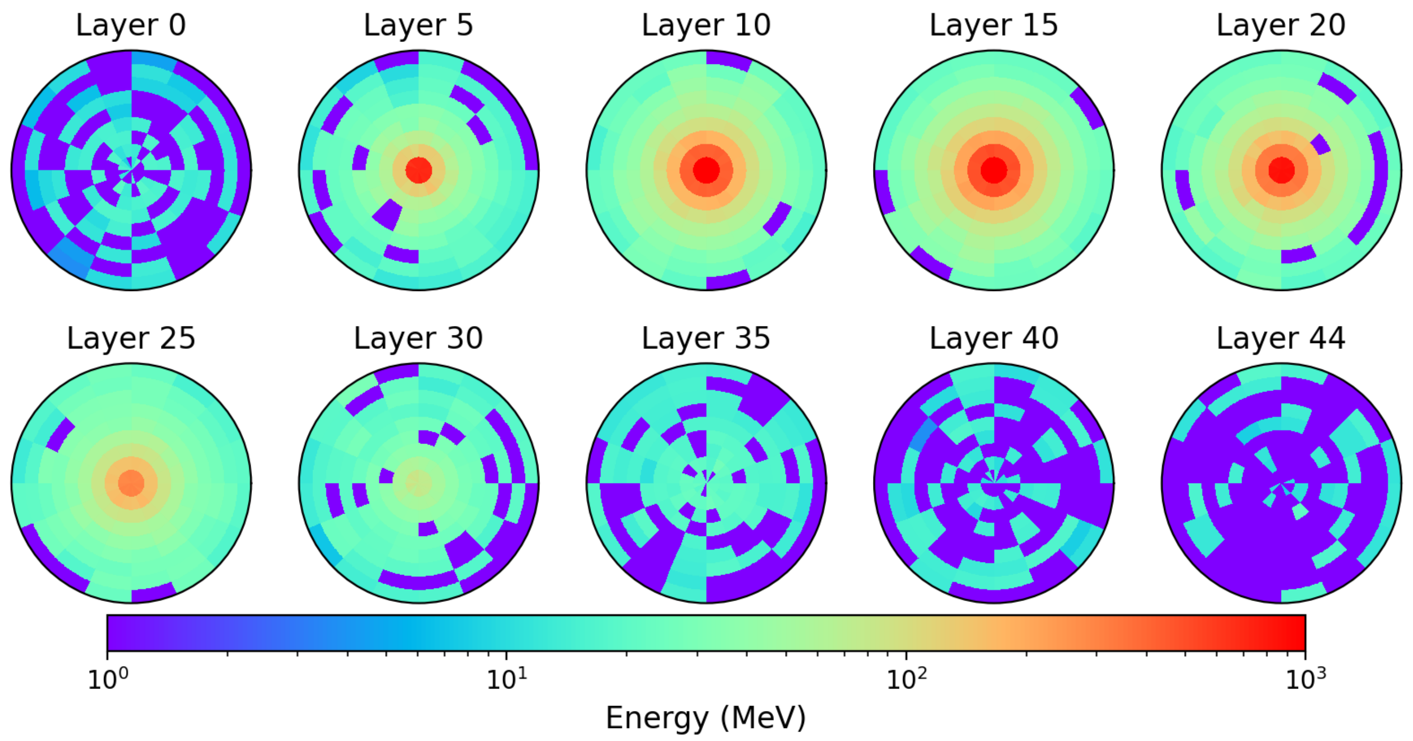


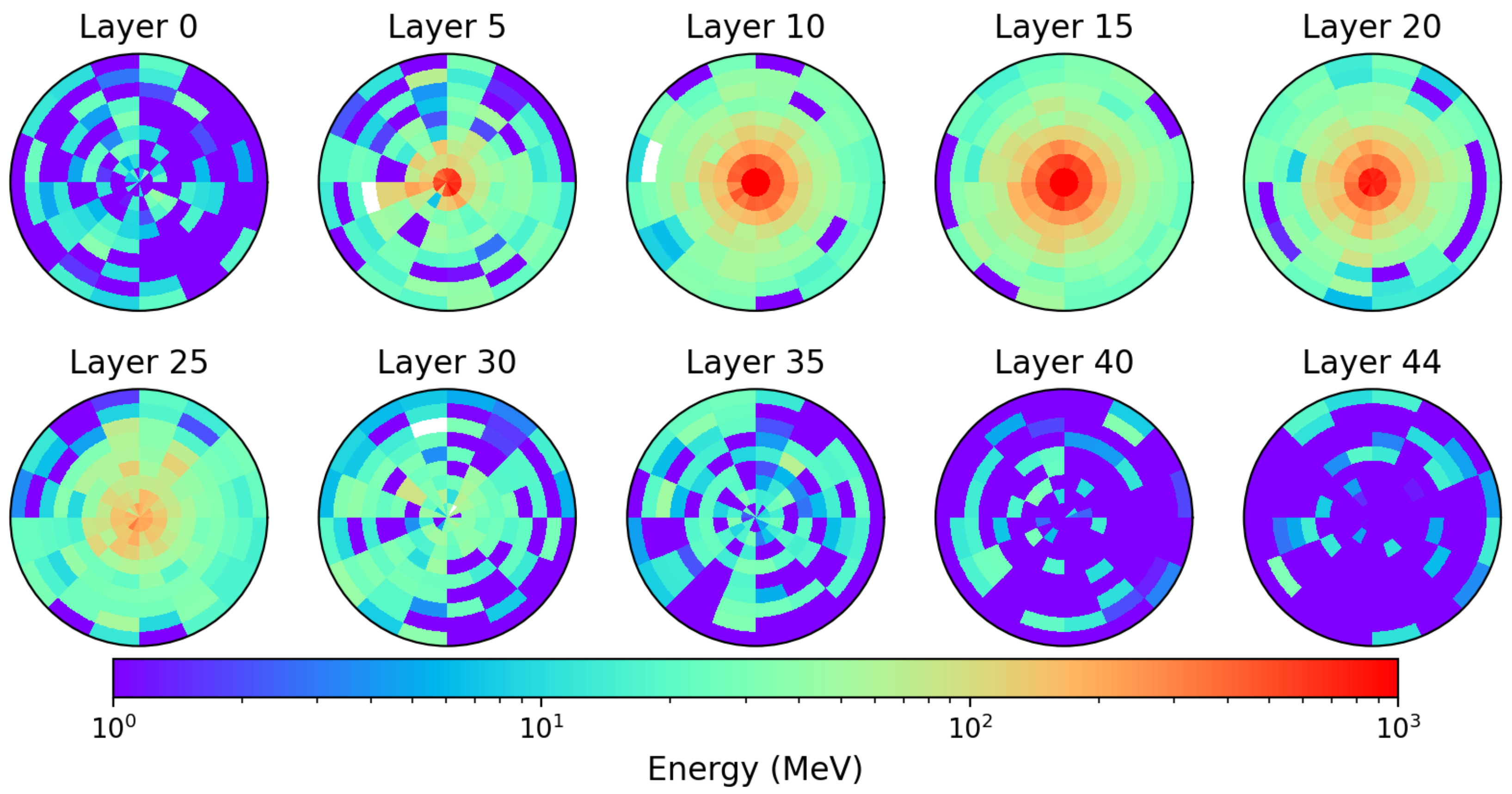
Recon





Sample





# Summary

- Zephyr topology is available (we already knew this from)
- Flux biases now available for sampling. Need to figure out the partial reprogramming issue.
- CRBM scheme is yielding much better results than RBM.
- Issue with making the model with **lowest MSE and hits loss** align with the model that yields good granularity and good overlap b/w encoded data and Gibbs sampled data.