QVAE w/ Pegasus

Dec 15th

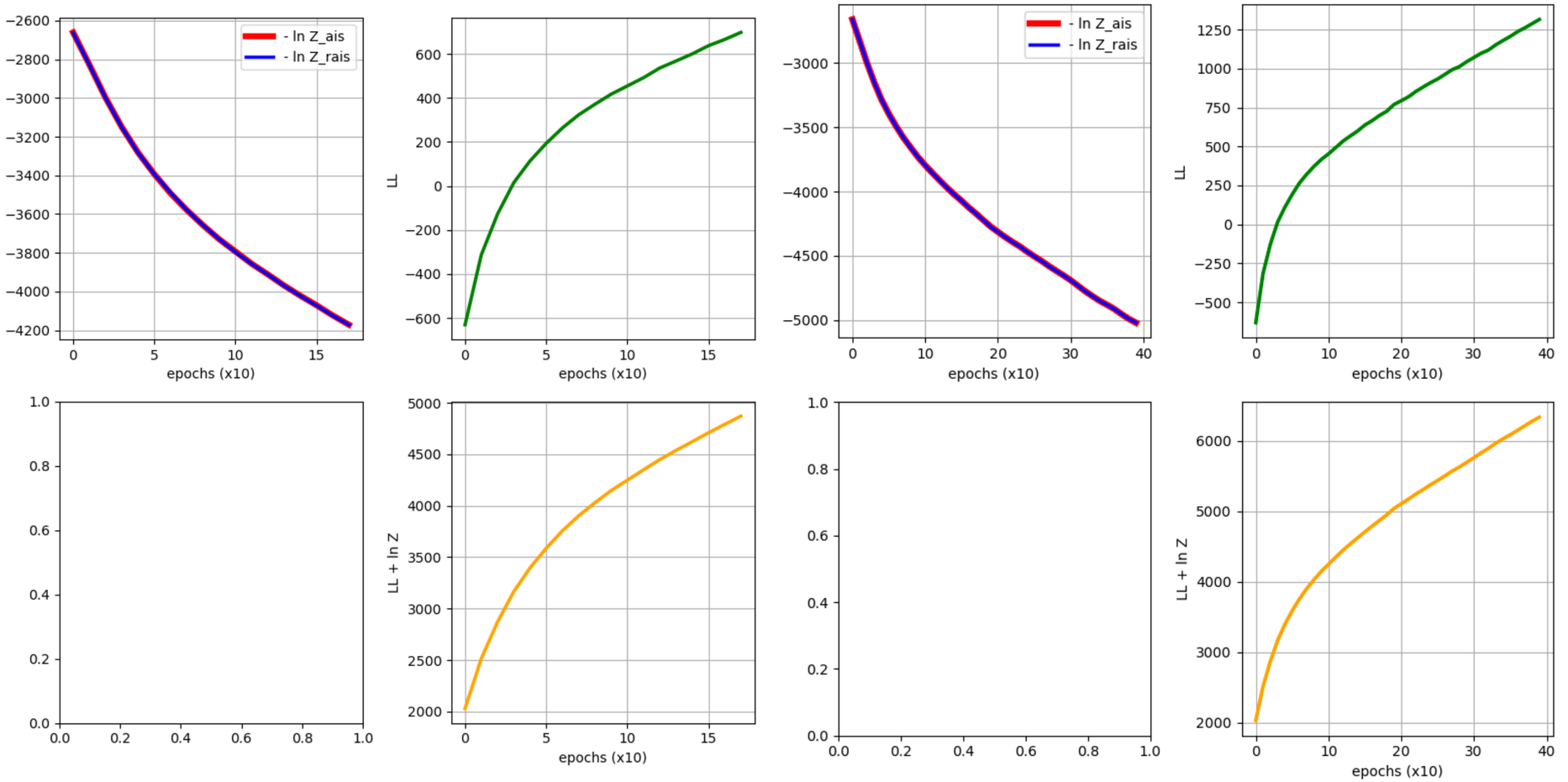
- QVAE
    - Architectures
        - CNN
        - FCN
    - Energy incidence
        - Condition on encoder and decoder
        - Condition on encoder
        - Unconditionalized
        - Modulated energy => Can lead to learning how to modulate more features, position of voxels, angles, etc.
    - 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

Contents

- Log-Likelihood for a couple of models after 400 epochs.
- Encoding energy into RBM.
- Training QVAE using QPUs

Happy-sun-270

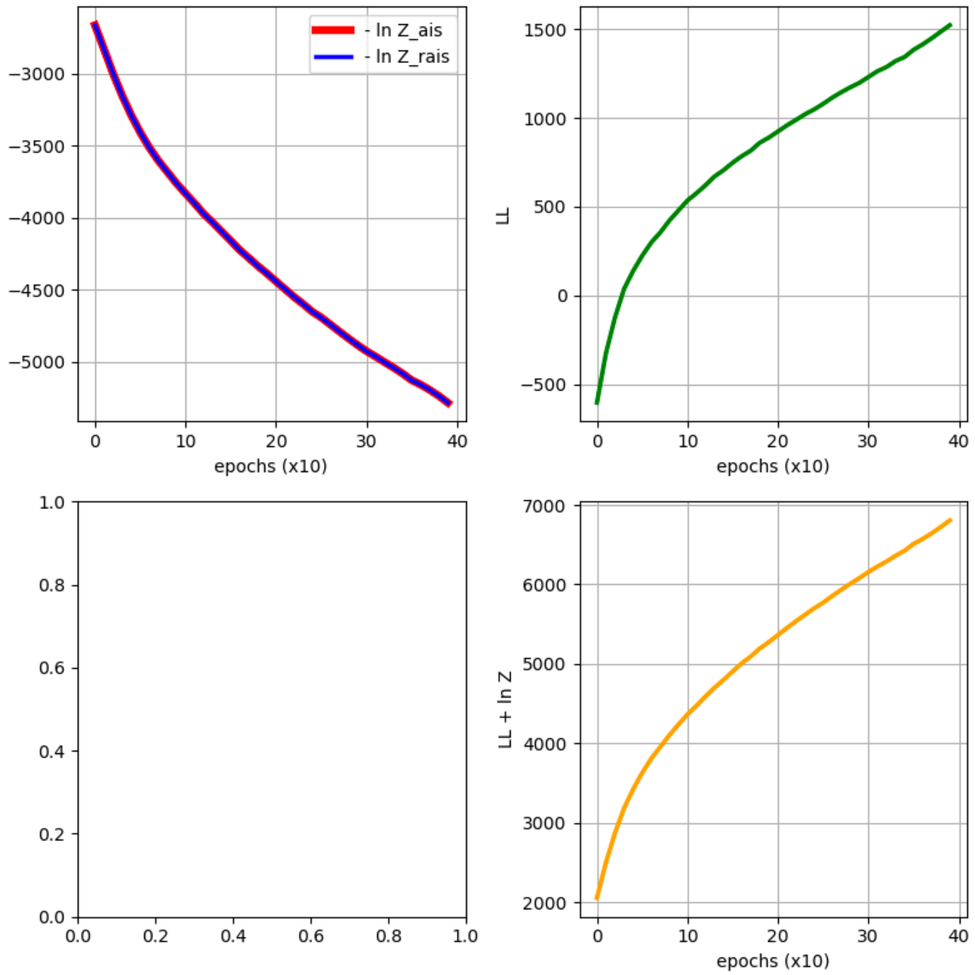
CNN+ cond VAE+posEnc on voxels+scaled data



LL not saturating :(

Drawn-cosmos—270

CNN+ cond VAE+scaled data



LL not saturating :(

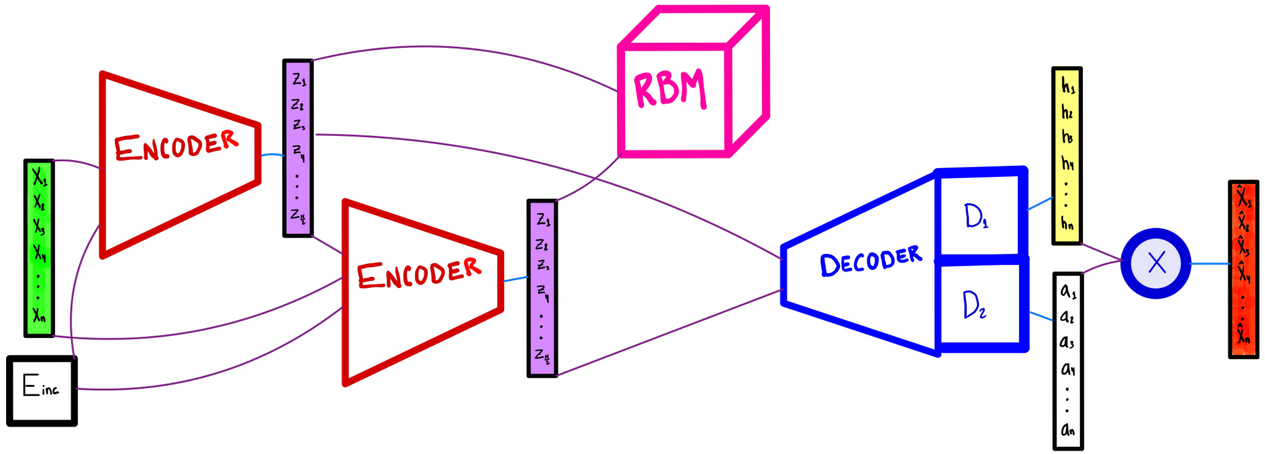
Why?

Possible reasons(?)

- The encoded data a moving target, perhaps leading to some kind of adversarial learning. ———> Try freezing encoder and decoder.
- Bug in code.
- If none of the above, when can we say our RBM is trained enough?

*Encoded val-set is not static. Need to compute <LL>.

Encoding energy into RBM.



$$Loss = MSE(\blacksquare\blacksquare) + BCE(\blacksquare\blacksquare) + Div_{KL}(\blacksquare\blacksquare)$$

How to conditionalize QPU? Via reverse annealing (?)

Initialize state. Specify annealing schedule.

This state should be (I think…) a random [0,1] for all spins but for those representing the energy.

Requirements:

- Familiarity w/ reverse annealing.
- Simple tests to see whether we recover part of the same state we initialized with.

How to conditionalize RBM?

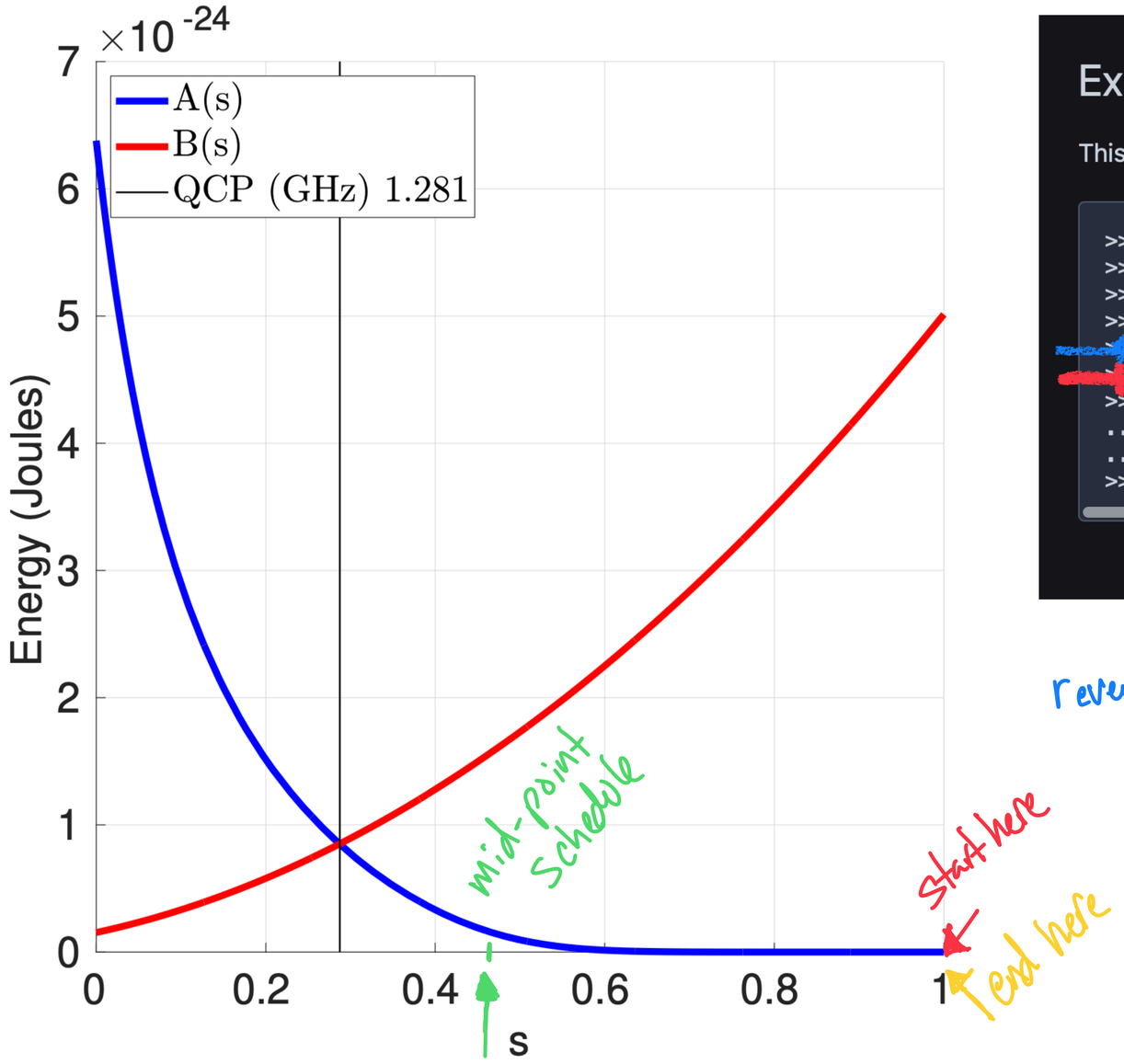
1. Convert Einc to binary number (31 + 1 bits. 1 for sign)
2. Append to encoded data. Hence, the encoded data has its label.
3. Either train RBM same way or fix 31+1 nodes of the RBM during Gibbs sampling

Example

This illustrative example configures a reverse-anneal schedule on a random native problem.

```
>>> from dwave.system import DWaveSampler
>>> import random
>>> qpu = DWaveSampler()
>>> J = {coupler: random.choice([-1, 1]) for coupler in qpu.edgelist}
>>> initial = {qubit: random.randint(0, 1) for qubit in qpu.nodelist}
>>> reverse_schedule = [[0.0, 1.0], [5, 0.45], [99, 0.45], [100, 1.0]]
>>> reverse_anneal_params = dict(anneal_schedule=reverse_schedule,
...                              initial_state=initial,
...                              reinitialize_state=True)
>>> sampleset = qpu.sample_ising({}, J, num_reads=1000, **reverse_anneal_params)
```

Encoding energy into RBM.



Example

This illustrative example configures a reverse-anneal schedule on a random native problem.

```
>>> from dwave.system import DWaveSampler
>>> import random
>>> qpu = DWaveSampler()
>>> J = {coupler: random.choice([-1, 1]) for coupler in qpu.edgelist}
>>> initial = {qubit: random.randint(0, 1) for qubit in qpu.nodelist}
>>> reverse_schedule = [[0.0, 1.0], [5, 0.45], [99, 0.45], [100, 1.0]]
>>> reverse_anneal_params = dict(anneal_schedule=reverse_schedule,
...                              initial_state=initial,
...                              reinitialize_state=True)
>>> sampleset = qpu.sample_ising({}, J, num_reads=1000, **reverse_anneal_params)
```

reverse_schedule = $[[t_0, s_0], \ldots, [t_s, s_s], \ldots, [t_n, s_n]]$