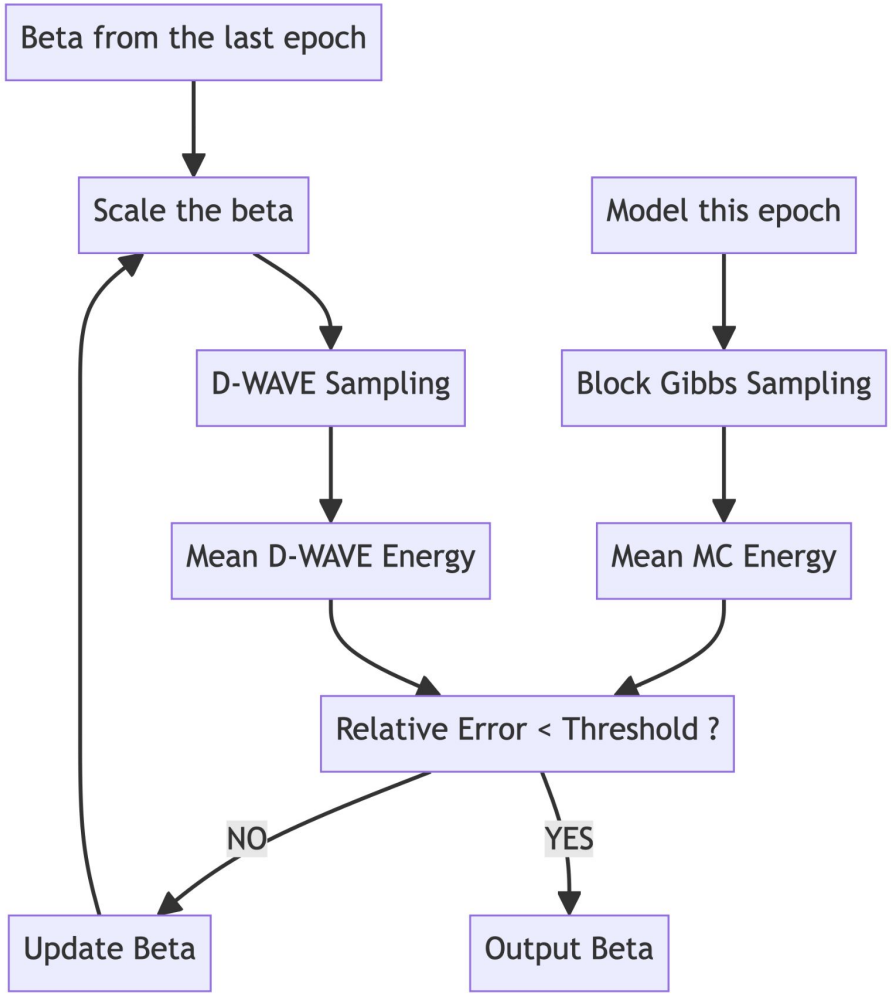
Temperature Estimation During Training (Problems)

- When is the beta well trained?
- How to make the standard independent from?
  - Learning Rate
  - Number of Nodes (Mean Energy, energy scale)
- How to train the beta effectively (safely)?
- How to prevent reaching D-WAVE limit during training/beta training?

The Algorithm :



Loss Limit (threshold)

energy_loss = abs(mean_dwave_energy - mean_rbm_energy)

$$Lim = \frac{2\sigma(E_{DWave})\sigma(E_{RBM})}{\sqrt{N}(\sigma(E_{DWave})+\sigma(E_{RBM}))}$$

If energy_loss < loss_limit: break

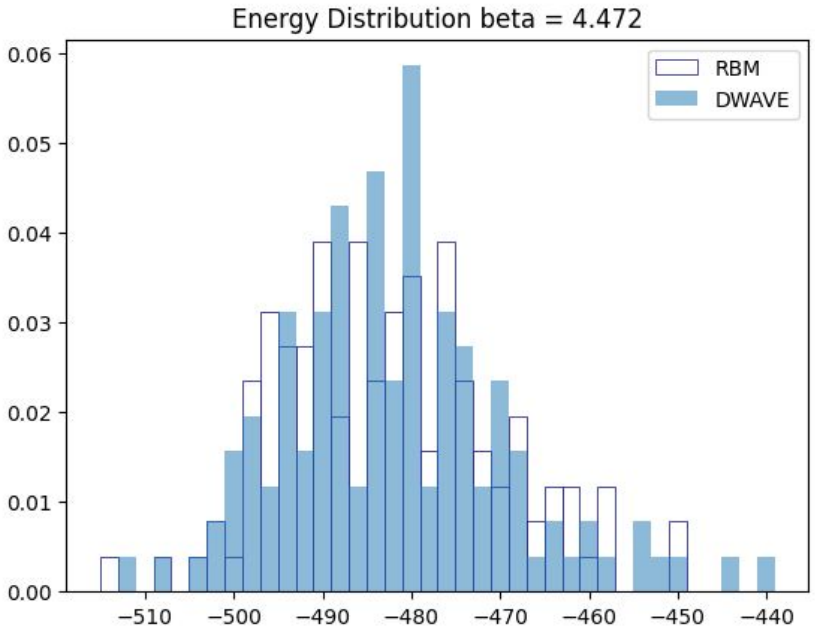
- Independent with number of nodes
- Independent with the size of mean energy.

Beta Limit (in case scaling up the w&b to exceed limit)

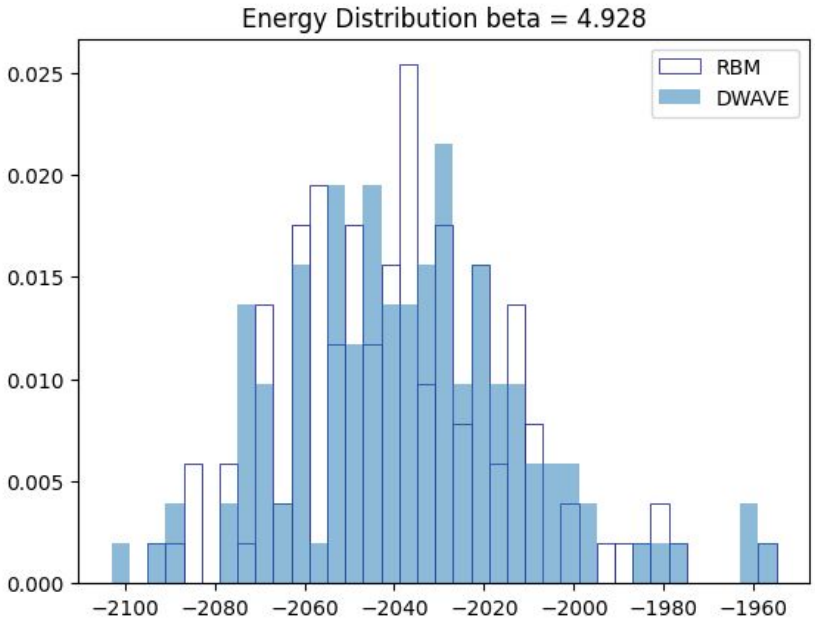
- Min beta = 2 during the beta training
- Won’t need to care about higher beta

Result of beta_training with pre-set threshold

`n_of_nodes_per_layer = 128, AdvSys 4.1`



`n_of_nodes_per_layer = 512 AdvSys 4.1`



Initial Gaussian Model: n_of_nodes_per_layer = 512

Method 1

beta = max(beta - lr * (mean_dwave_energy - mean_rbm_energy), 2.0) , lr=0.1

```
beta, error_list, rbm_energies, dwave_energies = beta_training(model, beta_init = 100.0)
```

43.0s

```
The error_limit is: 4.999545288085938
The number of epoch is: 1        The beta is: 2.0       The energy loss is: 1856.0181958675385
The number of epoch is: 2        The beta is: 30.483887505531314       The energy loss is: 284.8388750553131
The number of epoch is: 3        The beta is: 2.0       The energy loss is: 1417.5135419368744
The number of epoch is: 4        The beta is: 30.023780083656312       The energy loss is: 280.2378008365631
The number of epoch is: 5        The beta is: 2.0       The energy loss is: 1418.334464788437
The number of epoch is: 6        The beta is: 30.452881646156314       The energy loss is: 284.5288164615631
The number of epoch is: 7        The beta is: 2.0       The energy loss is: 1433.2275922298431
The number of epoch is: 8        The beta is: 30.230835747718814       The energy loss is: 282.3083574771881
```

Method 2

beta = max(beta * (mean_dwave_energy / mean_rbm_energy)**power, 2.0) power = 4

```
The error_limit is: 4.883040771484375
The number of epoch is: 1        The beta is: 2.0       The energy loss is: 1842.6183339357376
The number of epoch is: 2        The beta is: 3.3803340567305784       The energy loss is: 285.8223398923874
The number of epoch is: 3        The beta is: 4.732109431334821       The energy loss is: 178.8621348142624
The number of epoch is: 4        The beta is: 5.07920413696327       The energy loss is: 36.39631450176239
The number of epoch is: 5        The beta is: 5.149997790701203       The energy loss is: 7.06672465801239
The number of epoch is: 6        The beta is: 5.065853906822259       The energy loss is: 8.37858784198761
The number of epoch is: 7        The beta is: 5.117269493574827       The energy loss is: 5.15315043926239
The number of epoch is: 8        The beta is: 5.211023693651247       The energy loss is: 9.27400004863739
The number of epoch is: 9        The beta is: 5.070120653940952       The energy loss is: 13.92277729511261
The number of epoch is: 10        The beta is: 5.132352068205577       The energy loss is: 6.22700297832489
The number of epoch is: 11        The beta is: 5.135008568138234       The energy loss is: 0.26374614238739014
The trained beta is: 5.135008568138234
The number of epoch of the beta training is: 11
```

Adaptive Training

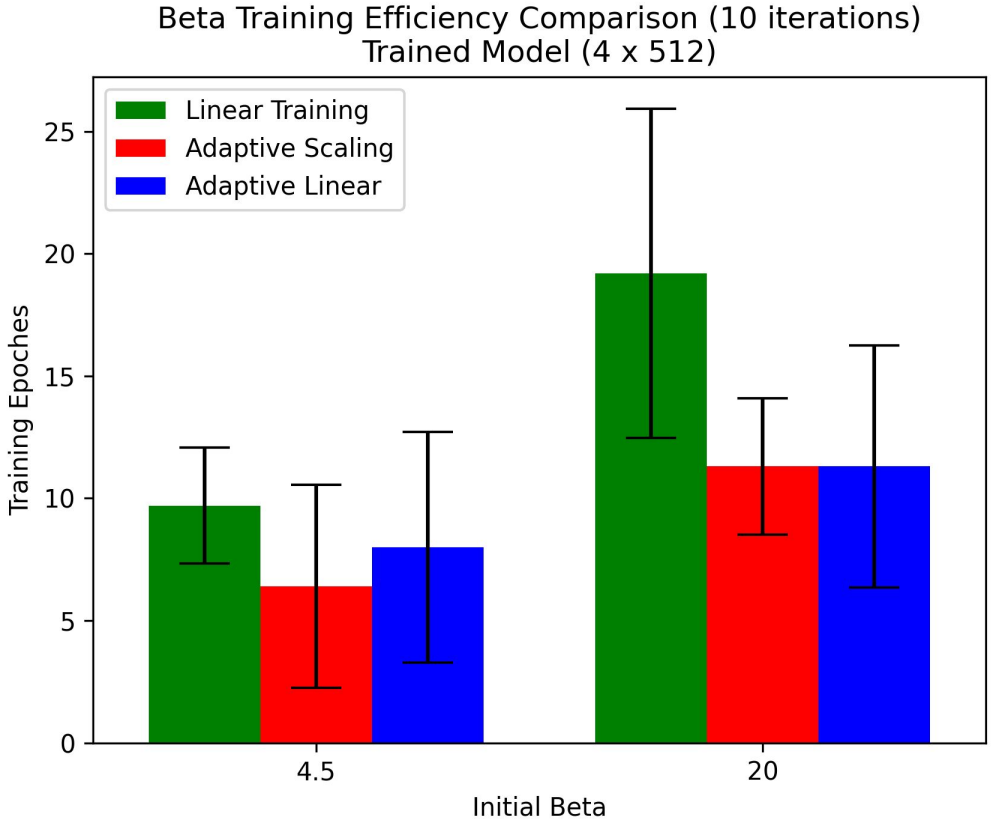
- Adaptive scale training: Power of training is dynamic.

```
training_order = max(1, np.floor( - 2 * mean_dwave_energy/np.var(dwave_energies)))
```

- Adaptive linear training: lr is dynamic

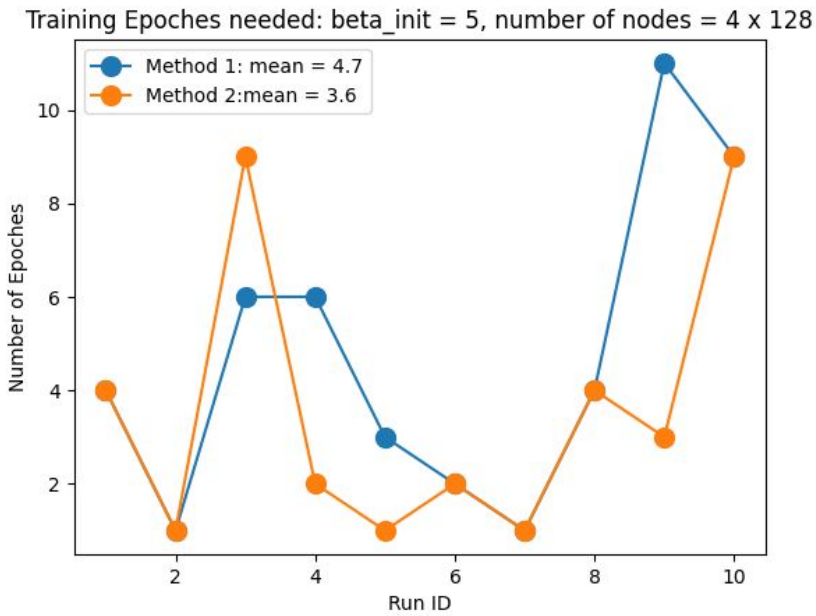
```
lr = 2 * beta/np.var(dwave_energies)
```

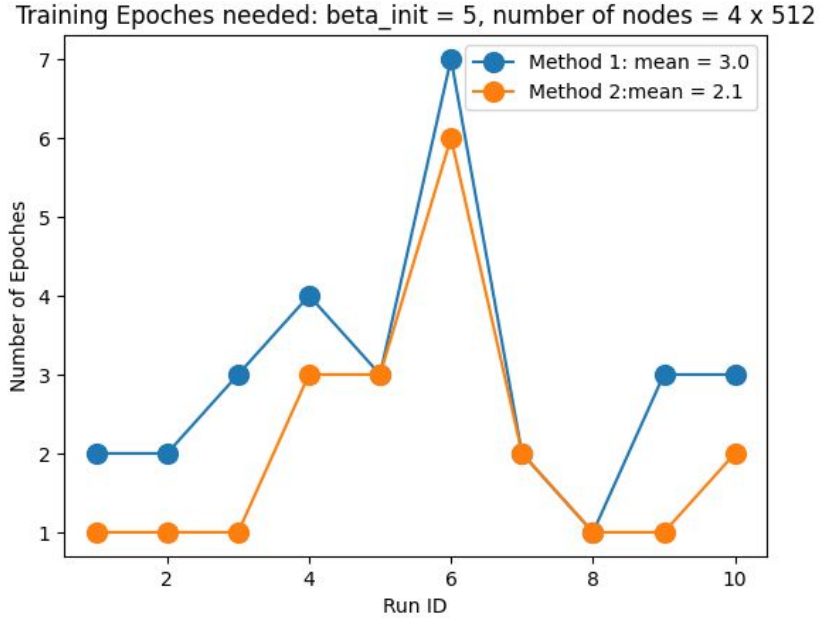
AdvSys 6.3:


Beta Training Efficiency Comparison (10 iterations)
Trained Model (4 x 512)
Linear Training
Adaptive Scaling
Adaptive Linear
Training Epoches
0
5
10
15
20
25
4.5
20
Initial Beta

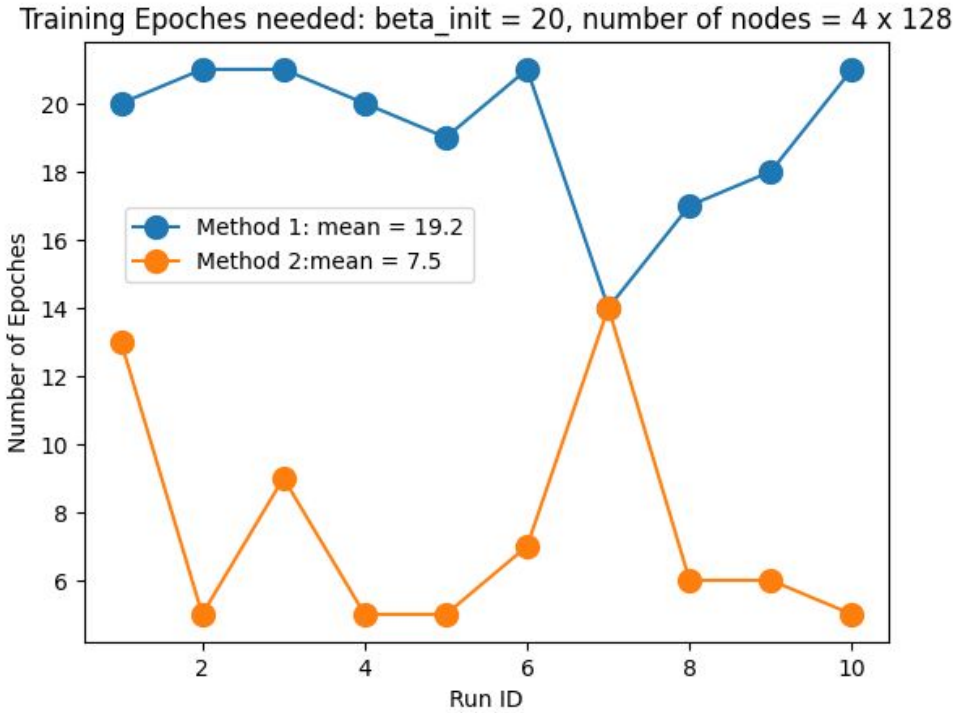
Archive

Lr = 0.01 vs Training power = 4, Error Limit Ratio = 0.01





Lr = 0.01 vs Training power = 4, Error Limit Ratio = 0.01



Trained Model: 4 x 512 nodes, lr = 0.01, power = adaptive