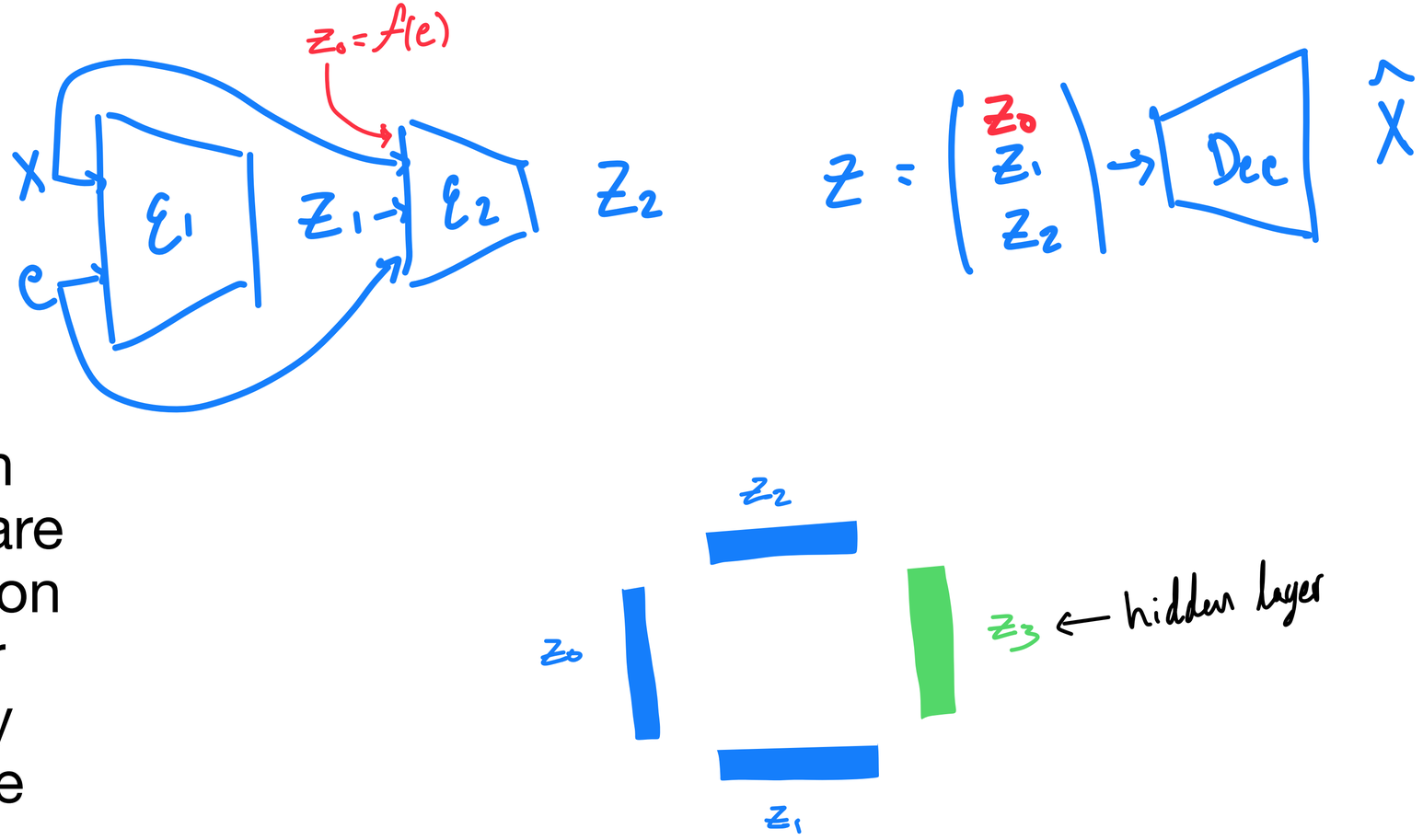
Calo4pQVAE: Progress and updates

Aug 29 2025

CaloQuVAE

1 hidden layer

- More than 1 hidden layer makes the marginalization intractable.
- Currently, the hierarchies in the encoder and decoder are coupled to the RBM partition (e.g., currently the encoder and decoder have as many hierarchies as partitions the RBM). This is intended to change.



Models results not done yet

crimson-galaxy-22 at: http://localhost:8080/caloqvae/caloqvae/6pi2z606

`/raid/javier/Projects/CaloQuVAE/wandb/run-20250826_172842-6pi2z606/files/autoencoderhidden_209_config.yaml`

- VAE up to 258 epochs.

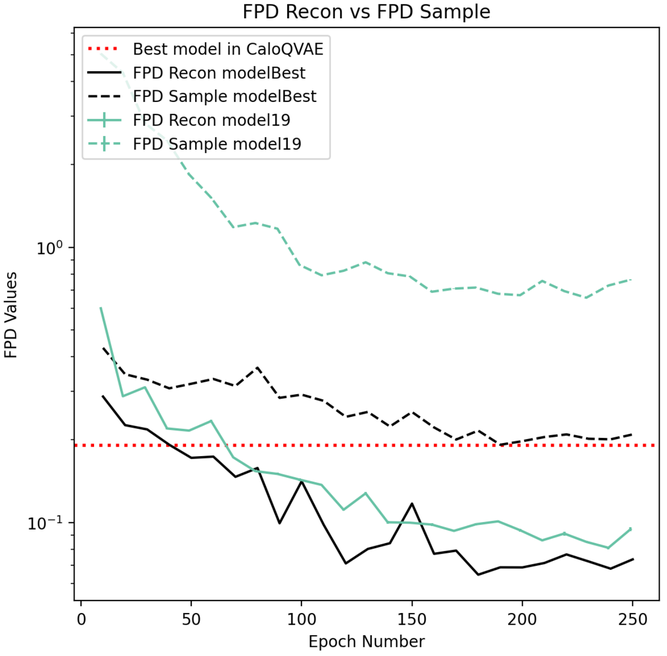
hopeful-elevator-19 at: http://localhost:8080/caloqvae/caloqvae/f33vuki5

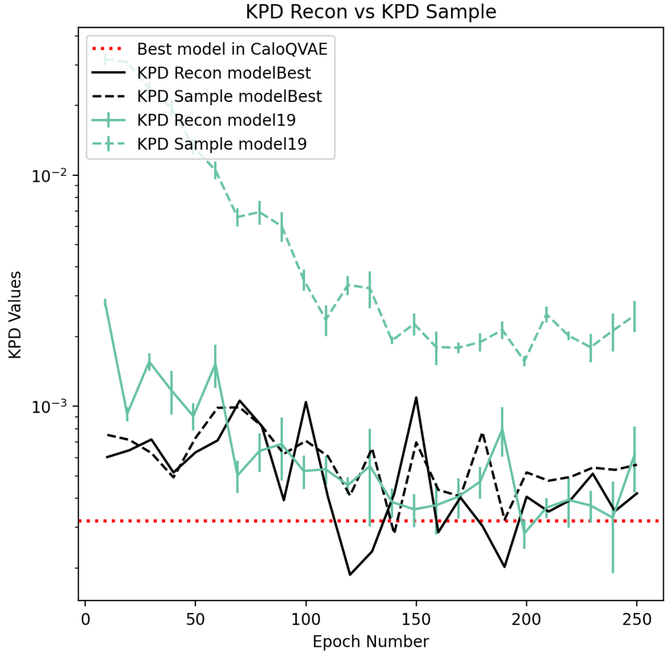
`/raid/javier/Projects/CaloQuVAE/wandb/run-20250826_141555-f33vuki5/files/ae_separate_29_config.yaml`

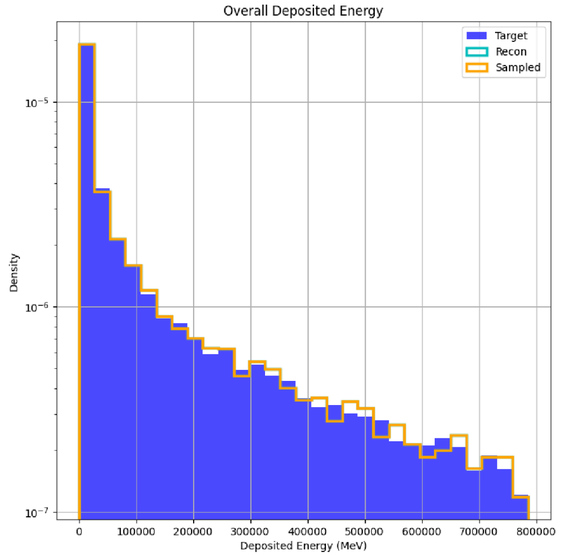
- AE up to 258 epochs. No reg

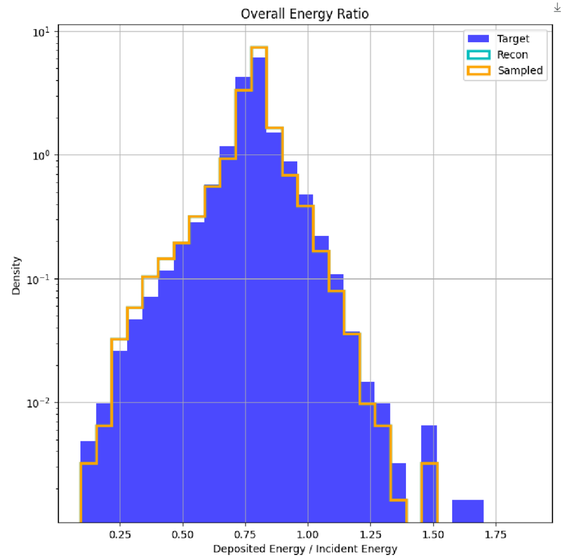
Model_19

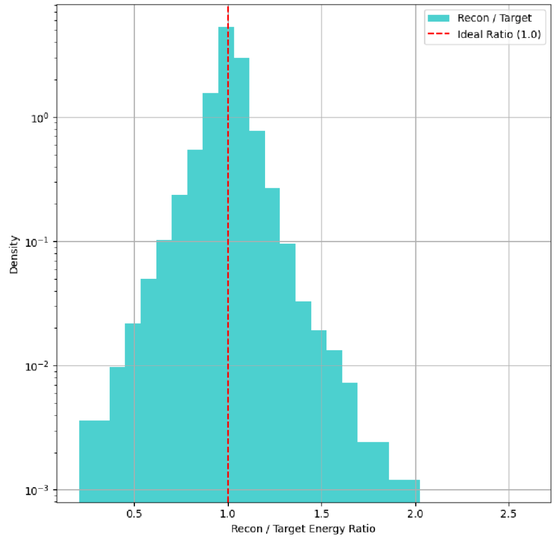
- no regularizer
- No hidden layer

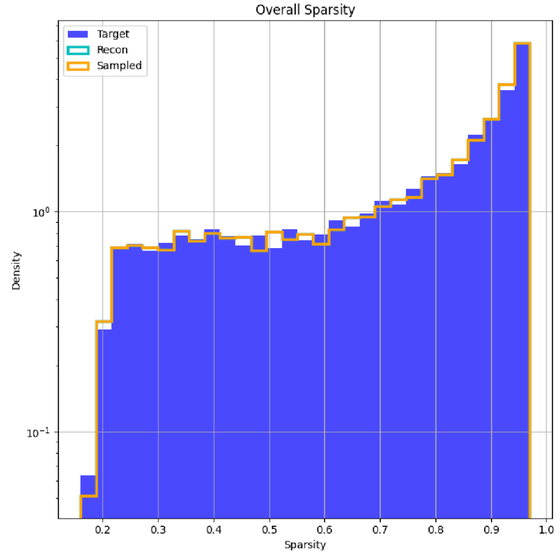






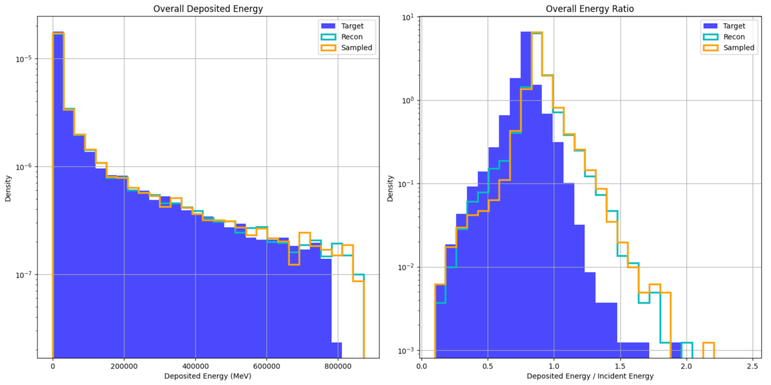




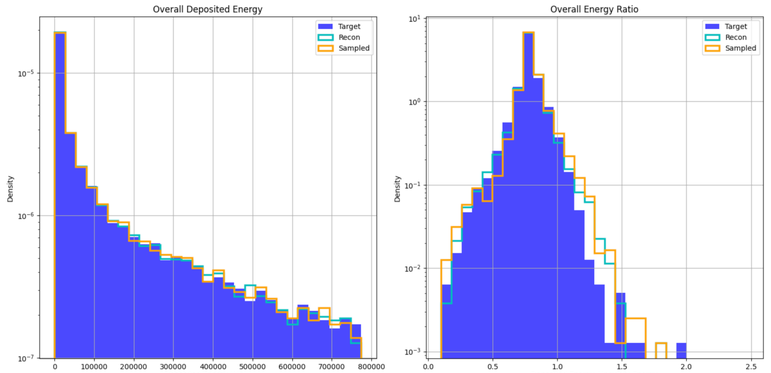


Model_22

- VAE
- Hidden layer
- Still generating the metrics
- Previous RBM opt
- No freezing at this point



Epoch ~220



Epoch ~221

Models results not done yet

fiery-morning-513 at: http://localhost:8080/caloqvae/caloqvae/z4hoj71e

`/fast_scratch_1/caloqvae/jtoledo/wandb/run-20250813_233227-z4hoj71e/files/autoencoderbase_249_config.yaml`

- VAE up to 199 epochs. Loads best model. RBM training for 50 epochs more.
- Using TFv2
- Adam for RBM
- Getting metrics -> performed poorly

pretty-resonance-550 at: http://localhost:8080/caloqvae/caloqvae/qtzevzdq

`/fast_scratch_1/caloqvae/jtoledo/wandb/run-20250825_231749-qtzezvdq/files/autoencoderhidden_279_config.yaml`

- VAE up to 250 epochs. Loads best model. RBM training for 50 epochs more.
- Using decoderhierachy0hidden :: 1 hidden layer
- Adam for RBM
- Getting metrics

logical-wood-547 at: http://localhost:8080/caloqvae/caloqvae/cysb33l7

`/fast_scratch_1/caloqvae/jtoledo/wandb/run-20250825_185123-cysb33l7/files/autoencoderhidden_349_config.yaml`

- VAE up to 250 epochs. Loads best model. RBM training for 50 epochs more.
- Using decoderhierachy0hidden :: 1 hidden layer
- RBMTorch
- Getting metrics

silvery-planet-542 at: http://localhost:8080/caloqvae/caloqvae/pp65hpwu

`/fast_scratch_1/caloqvae/jtoledo/wandb/run-20250821_230537-pp65hpwu/files/autoencoderhidden_249_config.yaml`

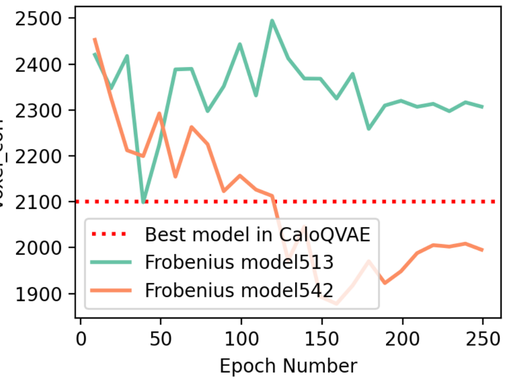
- VAE up to 199 epochs. Loads best model. RBM training for 50 epochs more.
- Using decoderhierachy0hidden :: 1 hidden layer
- This model uses 2 modules per encoder and decoder
- Adam for RBM
- Getting metrics -> done

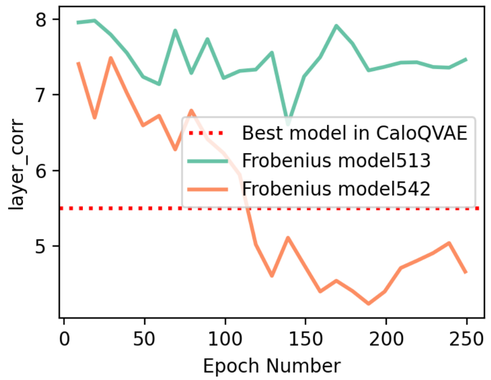
colorful-salad-535 at: http://localhost:8080/caloqvae/caloqvae/u88m8d6w

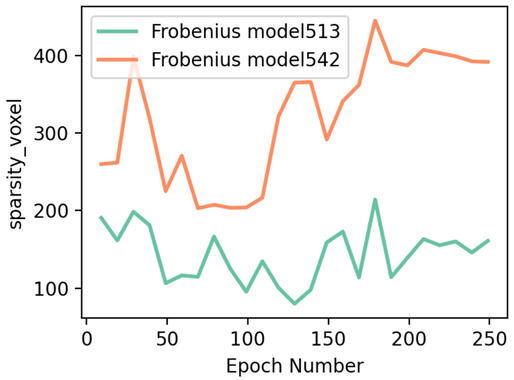
`/fast_scratch_1/caloqvae/jtoledo/wandb/run-20250819_203111-u88m8d6w/files/autoencoderbase_249_config.yaml`

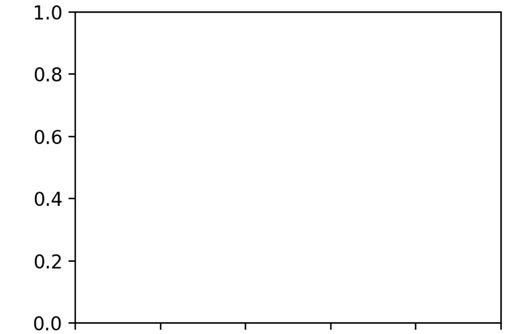
- VAE up to 199 epochs. Loads best model. RBM training for 50 epochs more.
- Using decoderhierachy0ca which has cross attention
- Adam for RBM
- Getting metrics

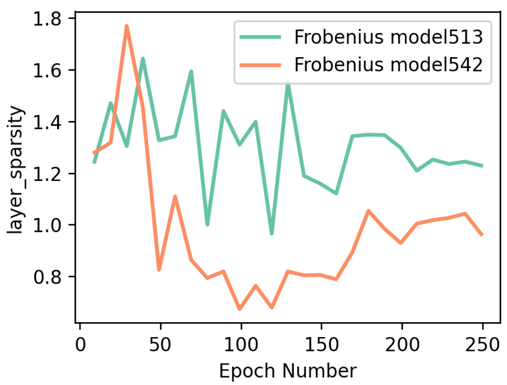
Model w/ and w/o hidden layer

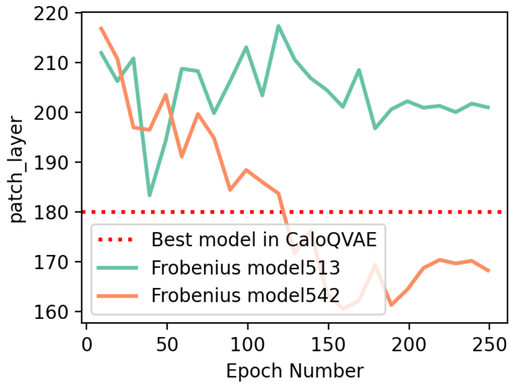


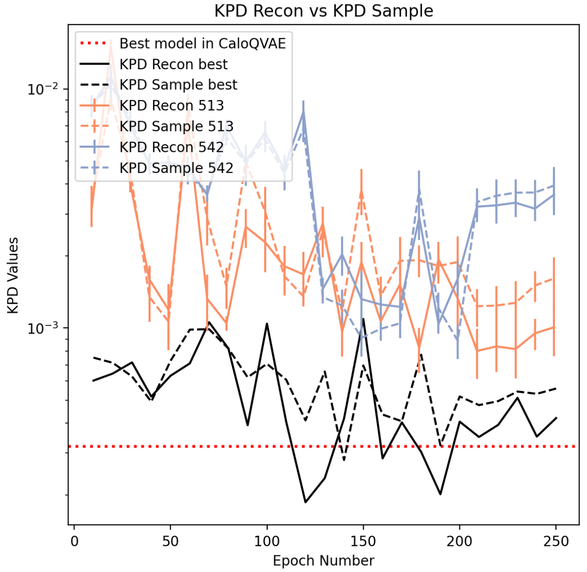


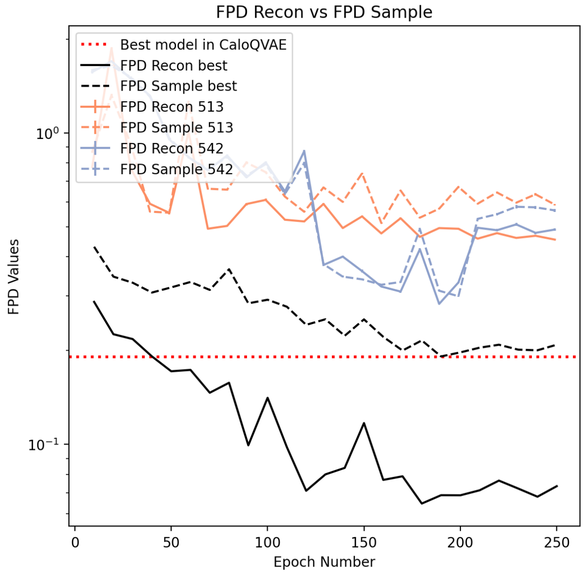






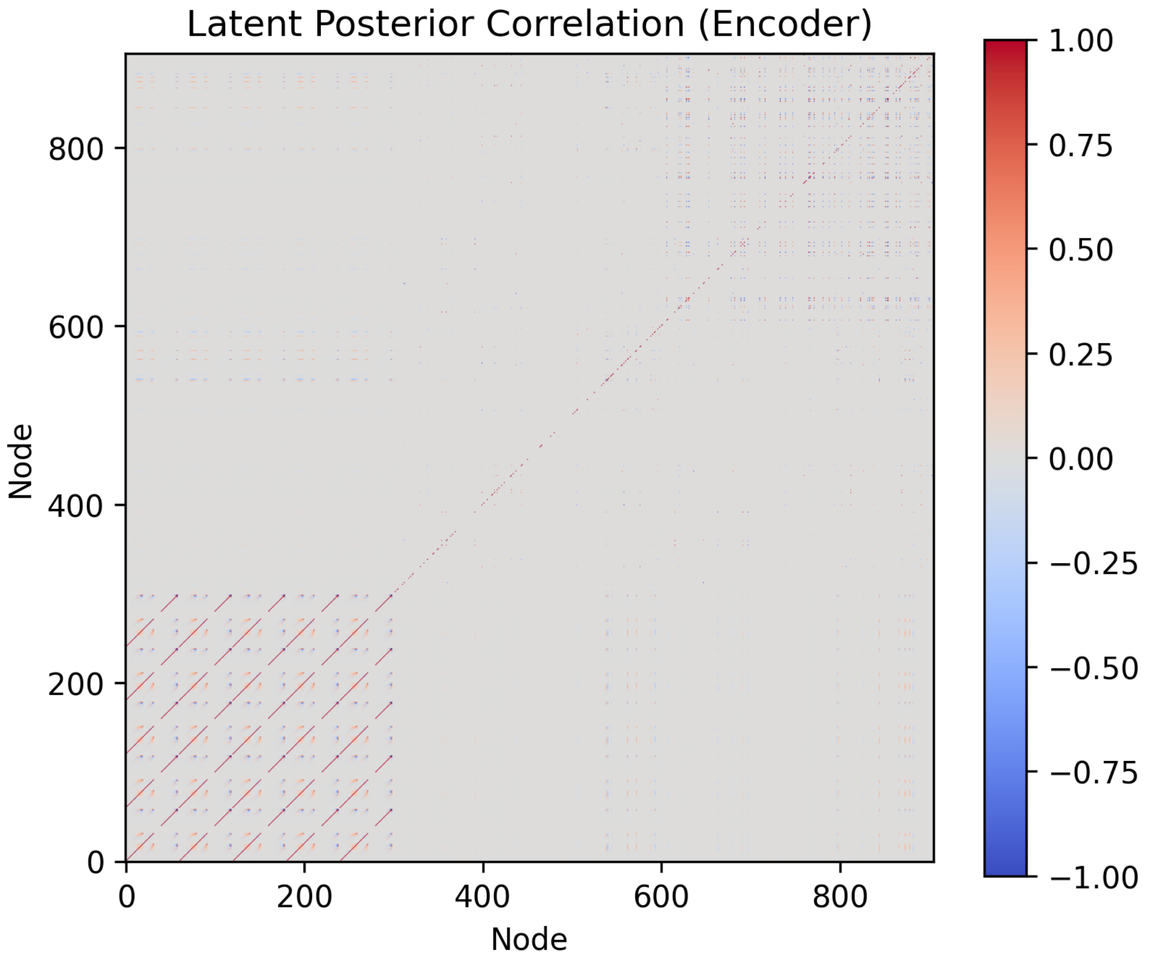


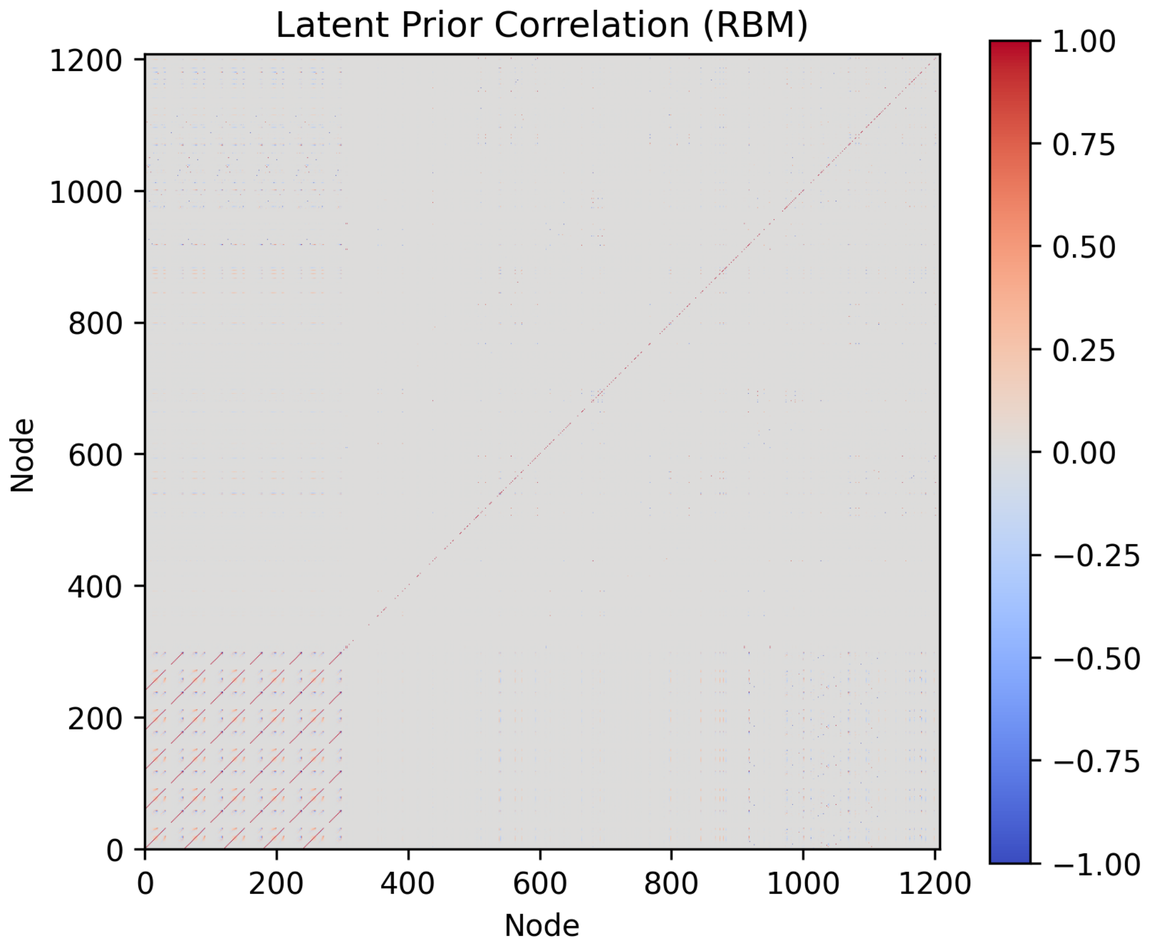




Correlation between latent space nodes

At epoch 249





6 KL divergence in a DVAE

In this section we derive the KL divergence equations in the case of the DVAE.

We denote as $\boldsymbol{z}$ the latent vector representation of $\boldsymbol{x}$, such that $\boldsymbol{z} \sim P(\boldsymbol{z}|\boldsymbol{x})$. Here $P(\boldsymbol{z}|\boldsymbol{x})$ is the approximate posterior, and we assume a multivariate Bernoulli-like distribution, *viz.*

$$P(\boldsymbol{z}|\boldsymbol{x}) = \prod_{i=1}^{N} p_i(\boldsymbol{x})^{z_i}(1-p_i(\boldsymbol{x}))^{1-z_i} \tag{9}$$

Notice that parameters $p_i(\boldsymbol{x})$ depend on the data point $\boldsymbol{x}$.

The Kullback-Liebler divergence is given by

$$D(P|Q) = \prod_{i=1}^{N}\sum_{z_i=0}^{1} P(\boldsymbol{z})\ln P(\boldsymbol{z}) - \prod_{i=1}^{N}\sum_{z_i=0}^{1} P(\boldsymbol{z})\ln Q(\boldsymbol{z}) \tag{10a}$$

$$= \langle \ln P(\boldsymbol{z})\rangle_{P(\boldsymbol{z})} - \langle \ln Q(\boldsymbol{z})\rangle_{P(\boldsymbol{z})} \tag{10b}$$

where $Q(\boldsymbol{z})$ denotes the prior distribution and $P(\boldsymbol{z}) = \int d\boldsymbol{x} P(\boldsymbol{z}|\boldsymbol{x})\mathcal{P}(\boldsymbol{x})$ with $\mathcal{P}(\boldsymbol{x})$ being the distribution of the data of interest. We can rewrite Eq. (10b) as

$$D(P|Q) = \int d\boldsymbol{x} D(P(\boldsymbol{z}|\boldsymbol{x})||Q(\boldsymbol{z}))\mathcal{P}(\boldsymbol{x}) \tag{11}$$

with

$$D(P(\boldsymbol{z}|\boldsymbol{x})||Q(\boldsymbol{z})) = \langle \ln P(\boldsymbol{z}|\boldsymbol{x})\rangle_{P(\boldsymbol{z}|\boldsymbol{x})} - \langle \ln Q(\boldsymbol{z})\rangle_{P(\boldsymbol{z}|\boldsymbol{x})} \tag{12}$$

Computing the gradient w.r.t. the RBM parameters yields:

$$\frac{\partial \langle \ln Q(\boldsymbol{z})\rangle_{P(\boldsymbol{z}|\boldsymbol{x})}}{\partial h_k} = \beta\left(p_k(\boldsymbol{x}) - \langle z_k\rangle_{RBM}\right) \tag{20a}$$

$$\frac{\partial \langle \ln Q(\boldsymbol{z})\rangle_{P(\boldsymbol{z}|\boldsymbol{x})}}{\partial \mathcal{V}_{kl}} = \beta\left(p_k(\boldsymbol{x})p_l(\boldsymbol{x}) - \langle z_k z_l\rangle_{RBM}\right) \tag{20b}$$

Hence the gradient of the KL divergence density w.r.t. the RBM parameters is simply Eqs. (20), where w.r.t. the approximate posterior parameters yields

$$\frac{\partial D(P(\boldsymbol{z}|\boldsymbol{x})||Q(\boldsymbol{z}))}{\partial \theta_k} = \frac{\partial \langle \ln P(\boldsymbol{z}|\boldsymbol{x})_{P(\boldsymbol{z}|\boldsymbol{x})}\rangle}{\partial \theta_k} - \frac{\partial \langle \ln Q(\boldsymbol{z})\rangle_{P(\boldsymbol{z}|\boldsymbol{x})}}{\partial \theta_k} \tag{22a}$$

$$= \sum_{\mu=0}^{3}\sum_{j=1}^{N}\frac{\partial p_j(\boldsymbol{x})}{\partial \theta_k}\left[\ln\left(\frac{p_j(\boldsymbol{x})}{1-p_j(\boldsymbol{x})}\right) - \beta\left(h_j + \sum_{\nu\neq\mu}\sum_{k=1}^{M} W_{jk}^{(\mu\nu)} p_k(\boldsymbol{x})\right)\right] \tag{22b}$$

The solution we are looking for is given by the term inside the brackets, namely,

$$\ln\left(\frac{p_j(\boldsymbol{x})}{1-p_j(\boldsymbol{x})}\right) = \beta\left(h_j + \sum_{\nu\neq\mu}\sum_{k=1}^{M} W_{jk}^{(\mu\nu)} p_k(\boldsymbol{x})\right) \tag{23}$$

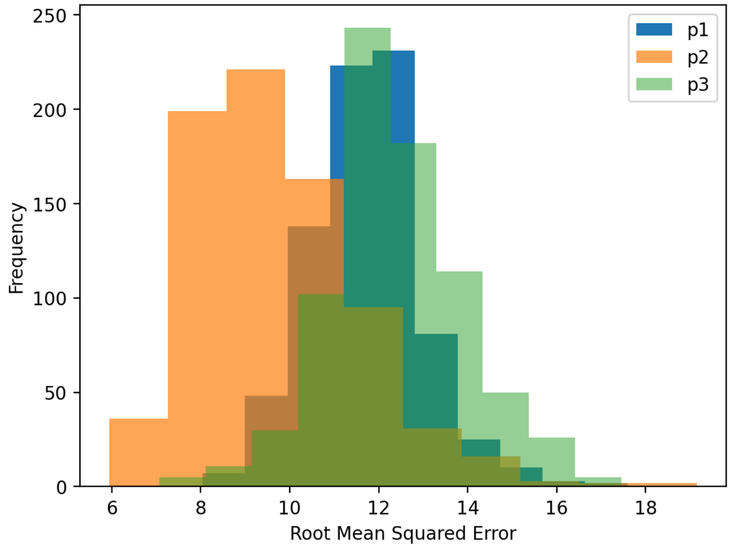
Notice that solving for $p_j(\boldsymbol{x})$ yields:

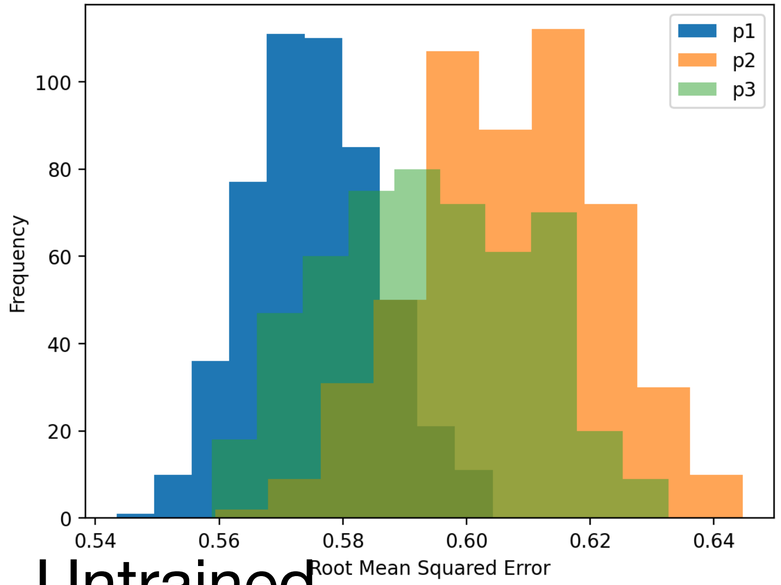
$$p_j(\boldsymbol{x}) = \sigma\left[\beta\left(h_j + \sum_{\nu\neq\mu}\sum_{k=1}^{M} W_{jk}^{(\mu\nu)} p_k(\boldsymbol{x})\right)\right] \tag{24}$$

From the previous we reach the following conclusions:

1. The KL divergence density gradient with respect to the RBM weights affects only the RBM. Via this term, the RBM can learn the correlation of the encoded data if such data is correlated (see Eq. (20b)).
2. The KL divergence density gradient with respect to the approximate posterior will push the logits towards matching the block Gibbs sampling Boltzmann factor (see Eq. (24)). If the cross entropy is neglected, the logits will be pushed towards zero instead. This may explain why when training the DVAE as a regularized autoencoder, one still needs to include the *positive energy term* in the loss function.
3. The gradients discussed here most likely will have many roots, yet we are only interested in one root. We should work on testing whether our models are getting close to the root of interest.

Some notes…

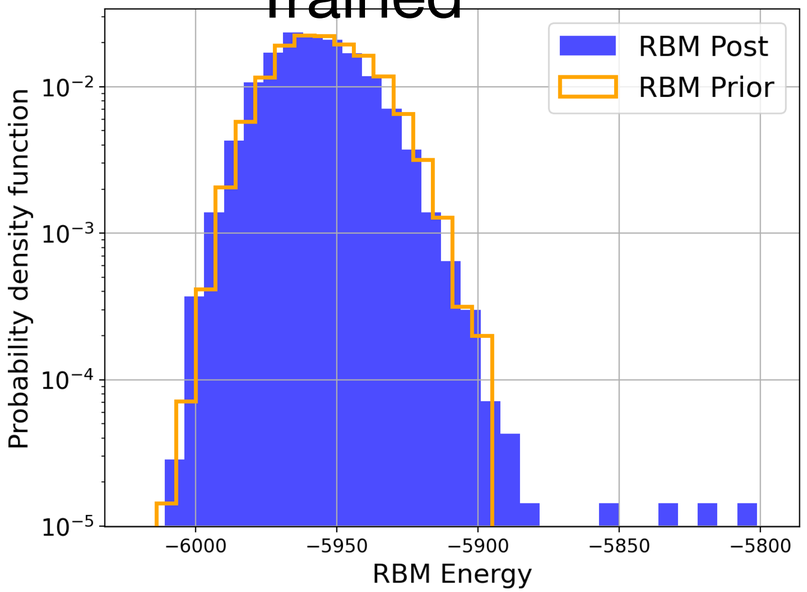




$$p_j(\boldsymbol{x}) = \sigma\left[\beta\left(h_j + \sum_{\nu\neq\mu}\sum_{k=1}^{M} W_{jk}^{(\mu\nu)} p_k(\boldsymbol{x})\right)\right]$$

Take the left hand side minus the right hand side, square it, take the mean and take the square root.

Trained



Untrained