Supplementary Material

1 ARCHITECTURE DESCRIPTION AND OPTIMIZATION

A convolutional layer is first applied to the input. A series of dense blocks and encoding blocks followed by a flatten and fully connected layers then encode the input to the parameterized latent distribution. A series of decoding blocks and dense blocks then decode samples from the latent space to the output. Figure S1 illustrates the general architecture we have selected for the latent mean and log-variance along with the output mean (with the output log-variance being constant but trainable).

A convolutional architecture was initially implemented without the use of dense blocks, but reconstruction of data is not very accurate with this architecture, even with some amount of hyperparameter tuning. Ref. Zhu and Zabaras (2018) illustrated that the architecture implemented there can accurately predict the data of our problem. The architecture contains many hyperparameters such as number of dense blocks, number of layers in each dense block, dense block growth rate, stride of convolutions, fully connected layer width, and others. There were three main goals for us in tuning the hyperparameters: accurate reconstruction, ability to produce disentangled representations, and high computational efficiency. As an example of hyperparameter tuning, we consider changes in the dense block growth rate keeping all other hyperparameters constant. Ten VAEs were trained with the ELBO loss for each growth rate value on the KLE2 dataset with $p(\theta)$ being standard normal. Figure S2 illustrates some statistics on this study. The overall ELBO loss, and in particular the reconstruction loss continues to decrease with and increase in growth rate, which is desirable. Good conclusions cannot be easily drawn from the disentanglement statistics, although at each growth rate a disentangled representation was observed. However, as the growth rate increases, the probability of convergence decreases. This may be improved by introducing lower learning rates, but in our case increase training time was highly undesirable. Thus, a growth rate of 4 was selected.

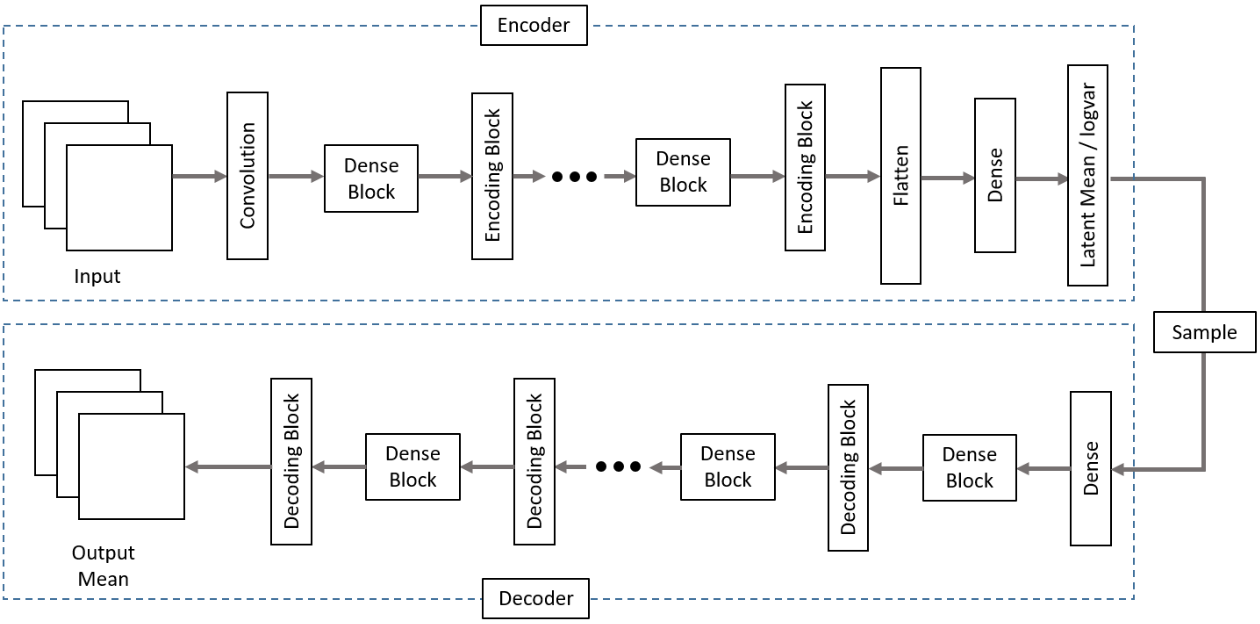

Figure S1. Dense VAE architecture.

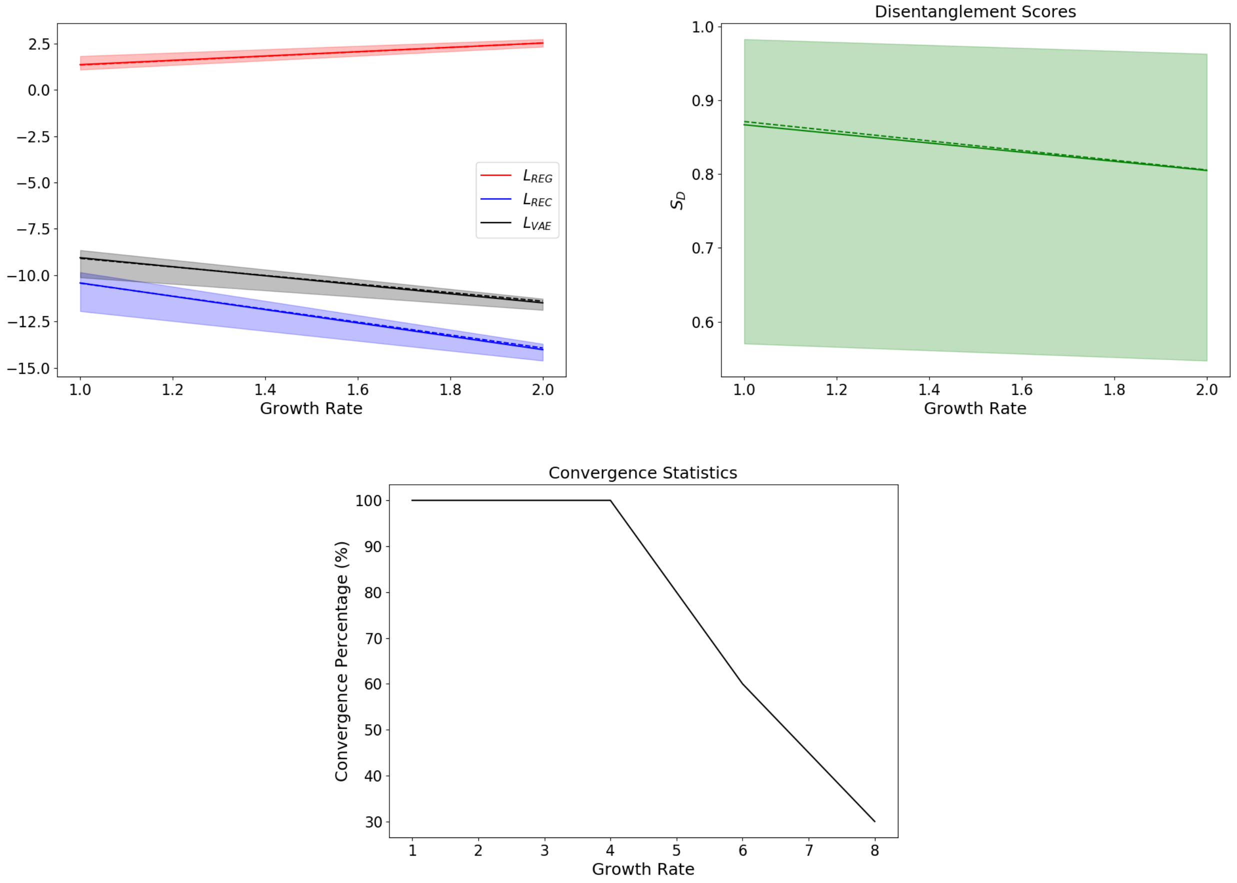

Figure S2. Hyperparameter selection example.

2 OVER-REGULARIZATION

To illustrate the avoidance of over-regularized local minima using our training method, Figure S3 shows the training losses and β as a function of epoch. The VAE loss reaches a local minimum but continues to increase as the true training loss decreases. With a small initial β (10^{-7}), great emphasis is placed on the reconstruction loss. When β begins to increase, the VAE is 'past' the over-regularized region and the training loss rapidly converges to the VAE loss, obtaining a desirable solution.

3 LOSS ANALYSIS WITH INCREASING NUMBER OF TRAINING SAMPLES

This study shows the relationship of the loss function and disentanglement with respect to the number of data samples used to train the VAE (Figure S4). All results are obtained with $\beta = 1$ during training and a latent dimension $n = 2$. For every number of training data ([32, 64, 128, 256, 512]), 10 VAEs are trained.

4 LOCAL MINIMA IN REGULARIZATION LOSS FROM ROTATION OF LATENT SPACE

We hypothesize that local minima exist in the regularization loss with respect to rotations in the latent space for the multimodal generative parameter distribution case. This results in the aggregated posterior being rotated 45 degrees relative to the generative parameter distribution (Figure S5).

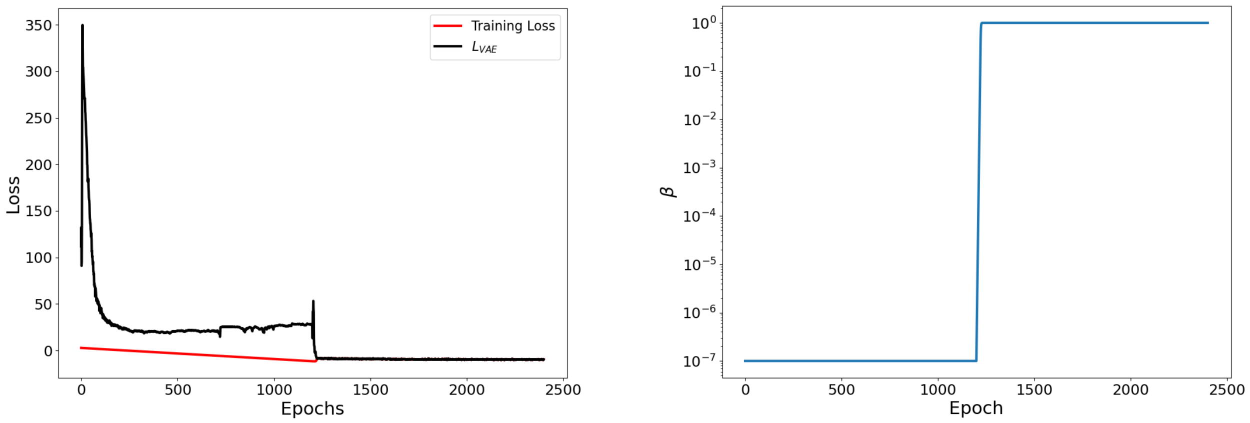

Figure S3. Training with initial increased weight on reconstruction loss helps to avoid over-regularized local minima.

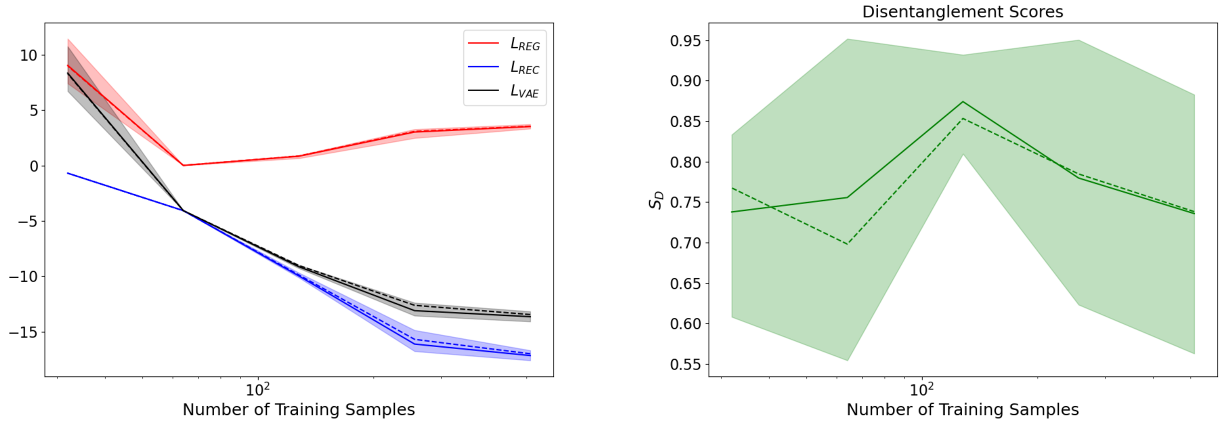

Figure S4. Solid lines indicate averages over training data for 10 VAEs trained at each point. Dashed lines represent averages over testing data. Ranges indicate minimum and maximum values. *left* Converged VAE losses for various numbers of training samples. *right* Converged VAE disentanglement score as a function of number of training samples.

5 REGULARIZATION LOSS AS A FUNCTION OF ROTATION OF LATENT ANGLE

Using rotationally-invariant priors does not enforce any particular rotation of the learned aggregated posterior distribution. In contrast, a non-rotationally-invariant prior can be used to enforce a particular rotation of the latent space (Figure S6).

6 DISENTANGLEMENT OF CORRELATED GENERATIVE PARAMETERS

Disentanglement has not been observed using our architecture when generative parameter distributions exhibit correlations between dimensions. Figure S7 shows that the aggregated posteriors are rotated relative to the generative parameter distributions, which does not facilitate learning a disentangled latent representation.

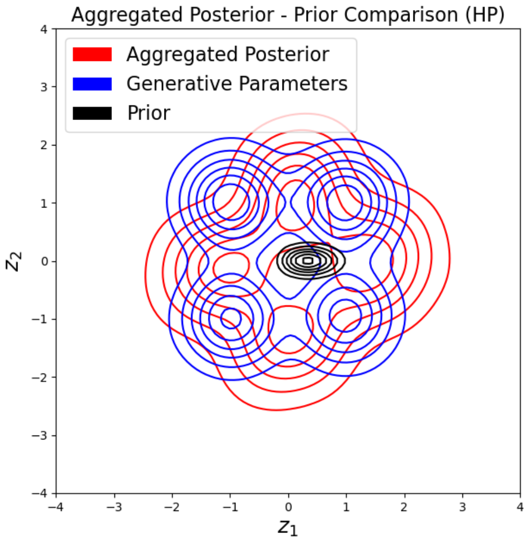

Figure S5. A 45 degree rotation of the latent space may be the result of local minima in the regularization loss during training.

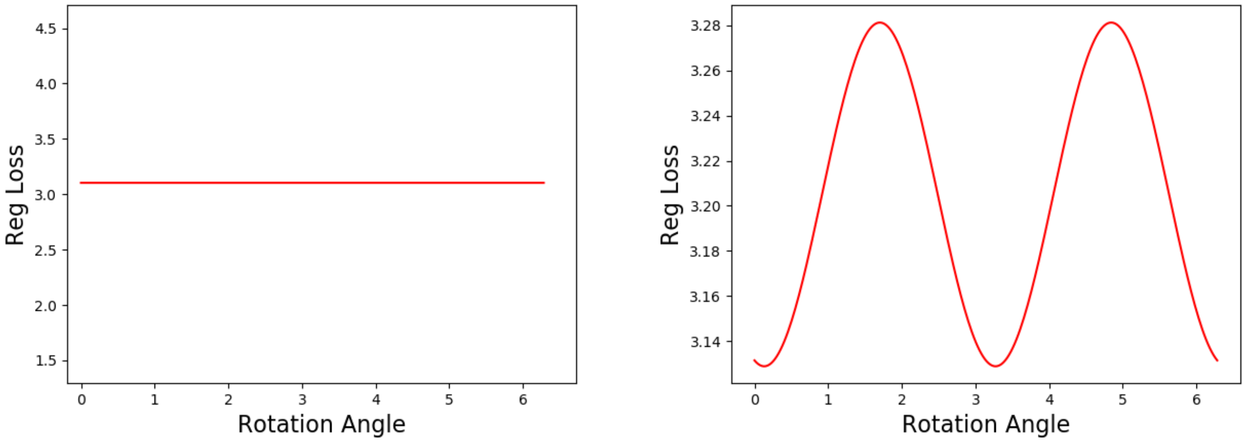

Figure S6. (*left*) Regularization loss unaffected by latent rotation when training with rotationally-invariant priors, (*right*) regularization loss is affected by latent rotation when training with non-rotationally-invariant priors.

REFERENCES

Zhu Y, Zabaras N. Bayesian Deep Convolutional Encoder–Decoder Networks for Surrogate Modeling and Uncertainty Quantification. *Journal of Computational Physics* **366** (2018) 415–447.

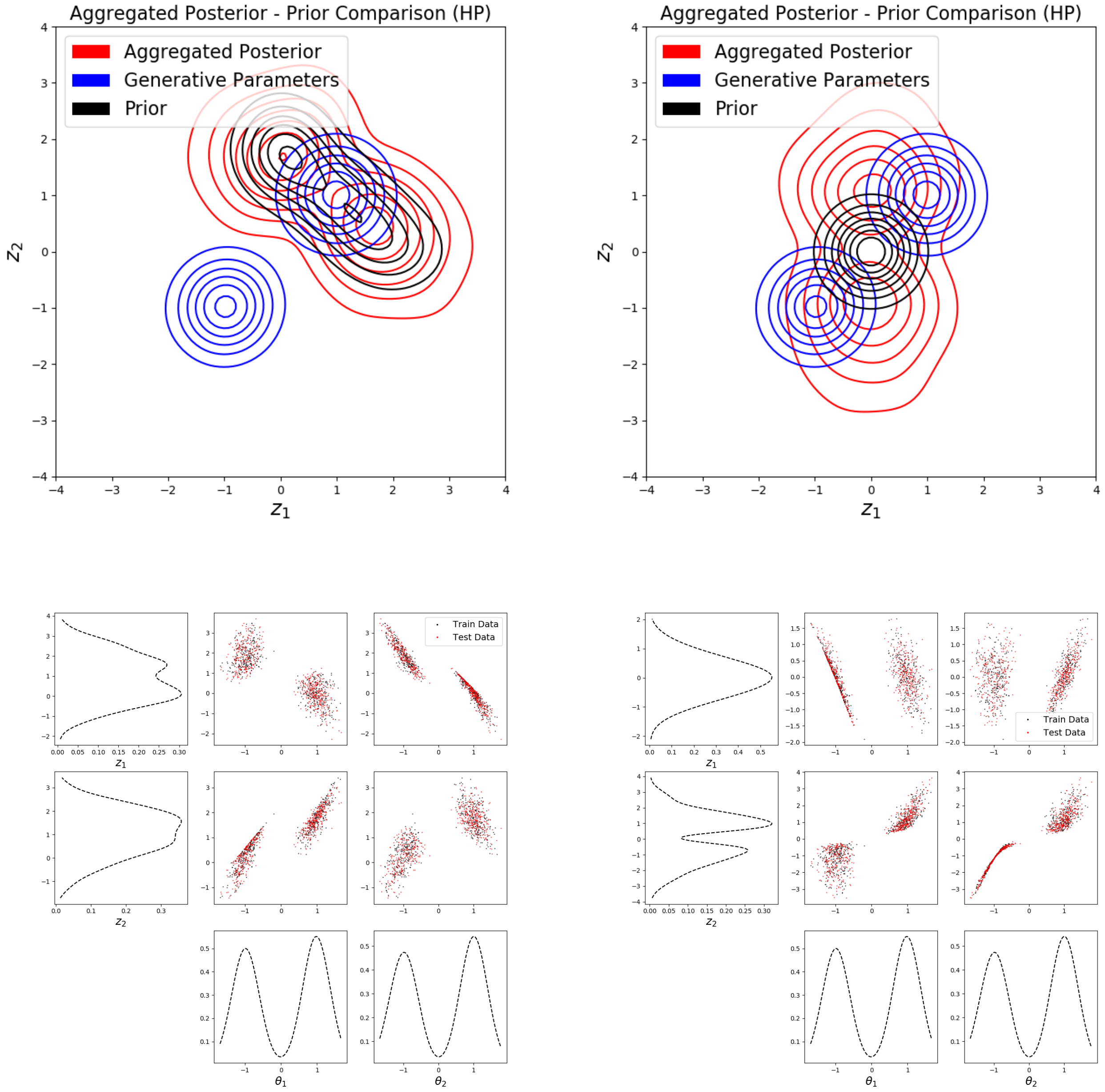

Figure S7. (*top*) aggregated posterior comparison correlations / rotations relative to the generative parameter distribution, (*bottom*) worse disentanglement when correlations not expressed in latent space.