1) Autoencoders use an encoder to transform an input tensor into an intermediate latent space, which serves as an abstract representation of the input parameter space, and then use a decoder to attempt to reconstruct the original input from its latent space representation. This type of network can be used to generate novel samples from the input parameter space by sampling from the latent space representation. We often use smaller-sized latent spaces when compared to our input spaces as it forces the network to distill down its representation to the most important information and helps the network perform better in generalization.

2) The bottleneck in an Autoencoder forces the network to use only the most important information to construct the intermedient latent representation. This has many benefits for the network's performance, including better generalization/reduction in overfitting, improved denoising, and a better and more meaningful identity function. If the latent space is the same size or larger than the inputs, you risk the network learning a trivial representation, such as directly copying each input over, which would lead to poor performance when generating new samples and a greater presence of noise.

3) Standard Autoencoders learn a deterministic and unstructured latent representation, which is directly optimized solely for reconstruction of the input features. In contrast, a Variational Autoencoder learns the distributions of the parameters in a probabilistic process, which, along with regularization from KL divergence, produces a smooth and standard latent space, which improves overall performance and decreases noise.

4) The reperamatirization trick is essential for VAEs as it allows the network to sample from the latent space in a way that is differentiable, which is critical for efficient backpropagation training. Without this trick, sampling directly during backpropogation would result in the model struggling to update network parameters and lead to poor convergence and performance.

5) The shape of a latent space can have immense effects on model performance, as a latent space with gaps and jagged edges will produce outputs that are unpredictable and unrealistic. On the other hand, a smooth and standard latent space allows for higher generalization and sampling performance while also assisting in interpolating between samples.

   To determine whether a latent space is “well-structured”, one can test the performance of the model when interpolating between samples or could even visualize the latent space through an algorithm such as PCA or t-SNE. We would want to try and quantify its similarity in behavior to an ideal smoother and standard-shaped latent space.

   When choosing the dimensionality of the latent space in an Autoencoder model, you must consider the previously discussed properties. Lower-dimension latent spaces risk not providing the model with enough parameters to properly encapsulate the input feature space, losing performance while potentially providing better generalization and denoising. In contrast, higher-dimension latent spaces can result in poor generalization performance and high noise but do not have the same risk of information loss. The factors must all be weighed, and many values for dimensionality must be tested before an informed decision can be made.

6) VAEs are very applicable to situations present in my own research on Fusion Energy. Since Fusion Reactors are very expensive to both construct and run, experimental data from such reactors is quite sparse, with each reactor also occupying an entirely separate area of the parameter space due to differing constructions and purposes. Thus, VAEs could be applied in this instance to assist in both data augmentation for a single machine and interpolation between the samples of differing machines to allow for the construction of a more generalized dataset for use in modeling the underlying physics present in all the reactors. You would likely need to use a different prior when compared to the standard one, as the underlying physics has effects on the distribution of what parameters are non-physical in a non-standard way.

   Additionally, you would have to be wary of interpolating too far from your samples, as you would want to check if the area of the parameter space you are examining corresponds to a physically possible scenario. In addition, you would likely want to make sure your data is most reflective of the current types of reactors being built and their underlying structures and assumptions.

7) The issue with the VAE was regarding the loss function's return statement, in which KLD had to be replaced with KLD.mean().

8) Autoencoders tend to perform very well in reconstruction and sample generation when the samples are pulled from very close to the training samples. Variational Autoencoders sacrifice a little on precision of reconstruction but generally perform better at producing generalized samples due to their more smooth latent space.

   I tested a few values for latent_dim and documented the performance (all with default parameters otherwise).

   latent_dim=1, loss=0.0697
   latent_dim=2, loss=0.0675
   latent_dim=3, loss=0.0672
   latent_dim=4, loss=0.0673
   latent_dim=5, loss=0.0674
   latent_dim=6, loss=0.0749
   latent_dim=7, loss=0.0739
   latent_dim=8, loss=0.0686

latent_dim=9, loss=0.0672

latent_dim=10, loss=0.0580
latent_dim=20, loss=0.0681
latent_dim=30, loss=0.0670
latent_dim=40, loss=0.0662
latent_dim=50, loss=0.0671
latent_dim=60, loss=0.0669
latent_dim=70, loss=0.0674
latent_dim=80, loss=0.0715
latent_dim=90, loss=0.0720

The loss for the VAE is converging well, with the loss consistently decreasing and the changes in loss between epochs decreasing as well. Thus, more epochs are not necessary for the model to converge.

The performance metrics listed above illustrate that the model does initially perform better when given a larger latent space, however, it begins to degrade in performance past a certain point, which aligns well with what we expect.