Notes and Ideas

1. Occasionally, some inverses cannot be computed in GP model and MLL. I suspect this is due to instability issues when training the encoder / decoder as these issues are not present when training GP on MLL loss in isolation.
2. Last run used a dataset generated with M=10 and Q=100 so it could run on CPU. Typically be training with dataset generated with M=100, Q=100. Also passed in true GP parameters on last run to see what difference this made during training. Set these to None when training these parameters.

1. Initialise decoder such that pixels of reconstructed frames are all black. Can do this by initialising weights as zero, then translating logits by -6. As the logits are the input to the sigmoid, this should initialise the probabilities close to zero. Should save a decent amount of time at the start of training.
2. Simplify the covariance of the likelihood such that each element of the state has the same variance i.e., \Sigma(v\_{i}) = \sigma(v\_{i})^{2} \* I . Alternatively, could set covariance to the identity as is common in other VAEs.
3. Start by training VAE excluding GP. I.e., Encoder still learns a mapping to the mean and variance of the likelihood, but sample likelihood instead of posterior; then map samples to images through decoder as usual. Once this works fairly well, then train SEGP end-to-end. Might be able to increase learning rate to 1e-3 when doing this. I expect this will make the most significant difference since occasionally nans occur in GP parameters. This was not the case when training GP in isolation.
4. Play around with encoder and decoder architecture - i.e., add in batch norm etc. Be careful when doing this as batchnorm requires input to be passed in with a specific shape - this is not the default shape with the current setup. Should just need to change VAE code to fix this.