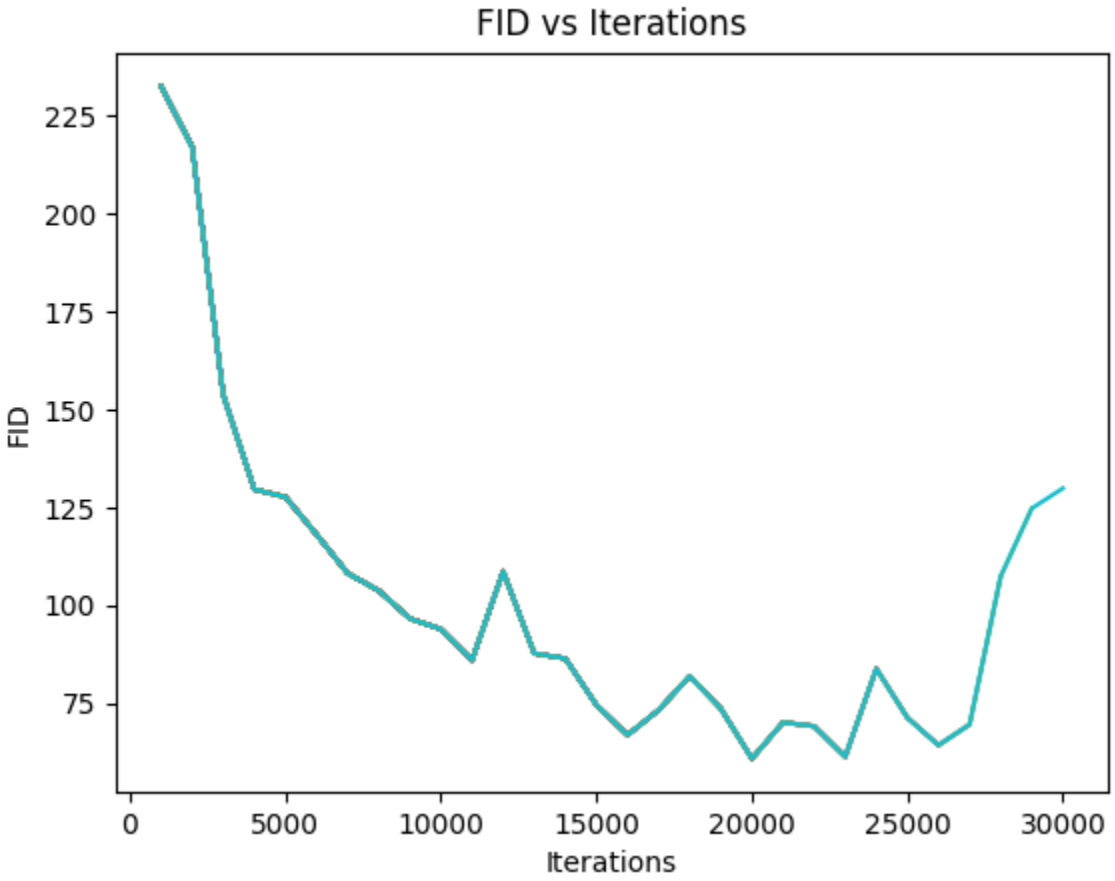
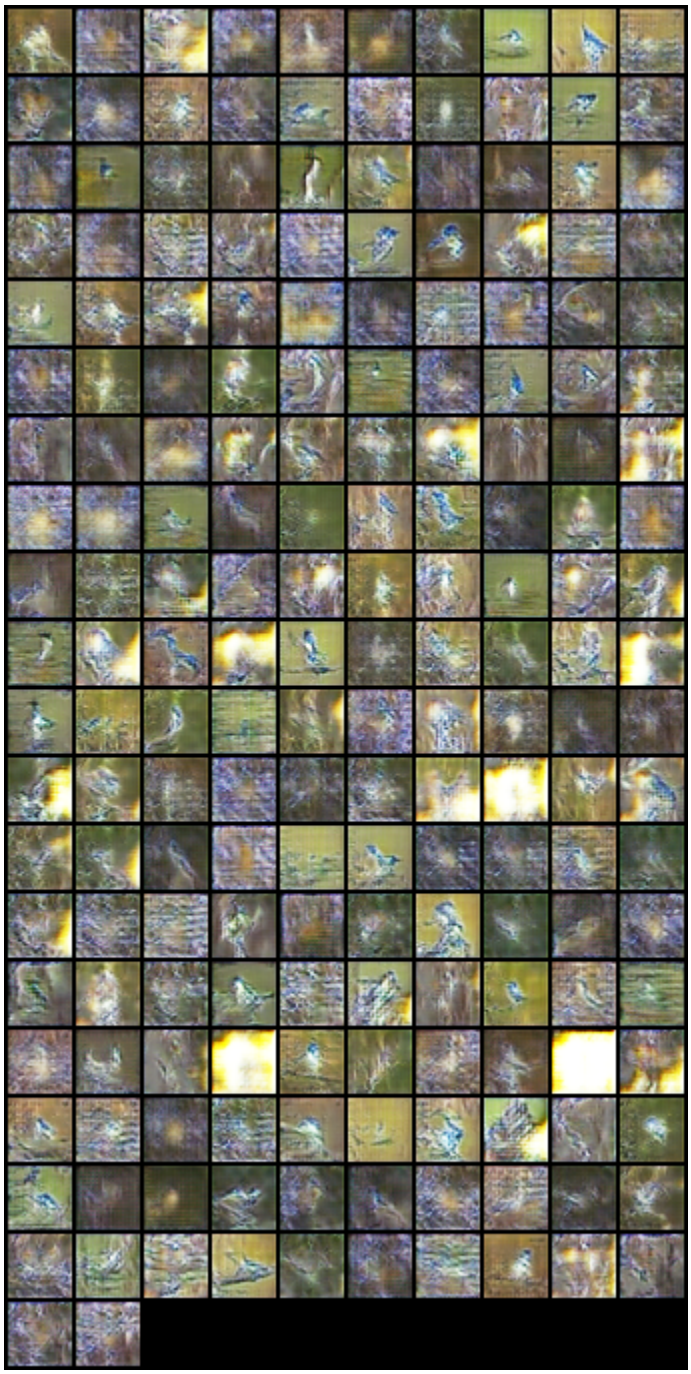
Question 1: **Used a batch_size of 192, for all GAN questions, have inquired TA about it and he was fine.**

1.3.3

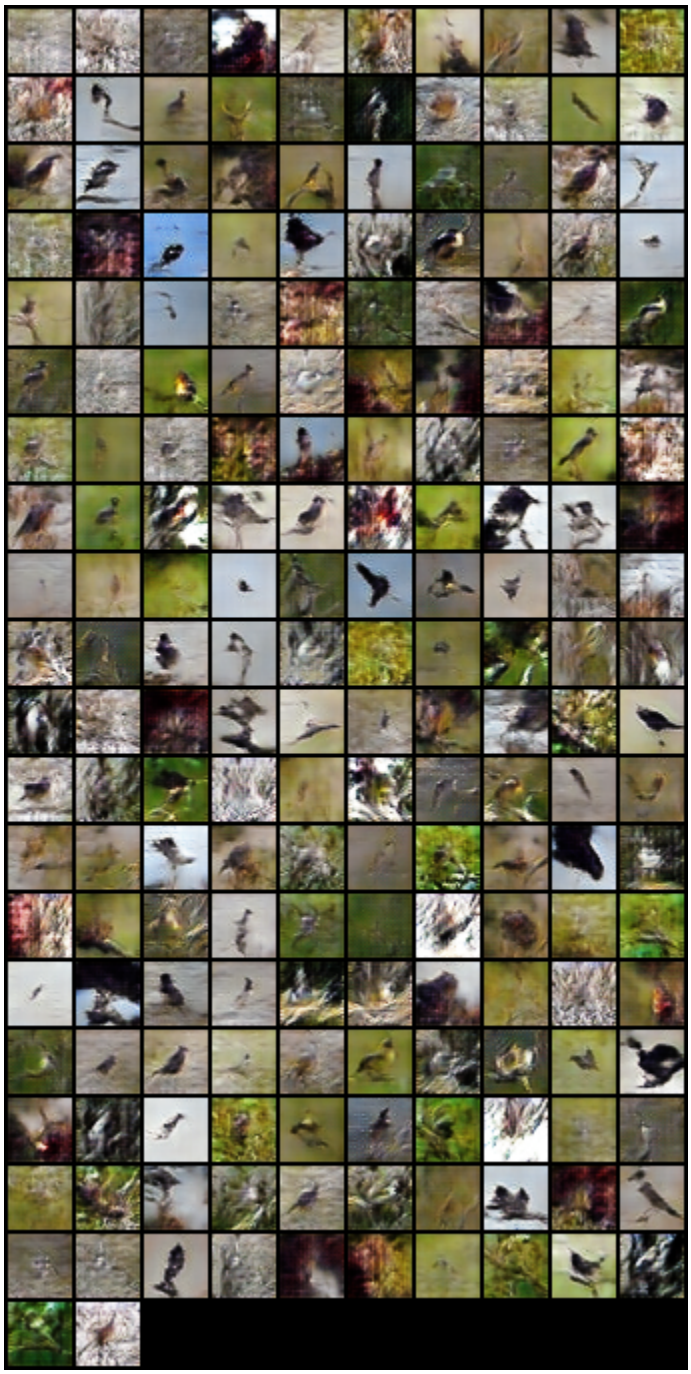
1. The final FID obtained is 130.666



2. Samples look overall fine, but individual pictures lack quality. The best images are generated for an iteration number of 26000, after that GAN becomes worse. I think it collapses.

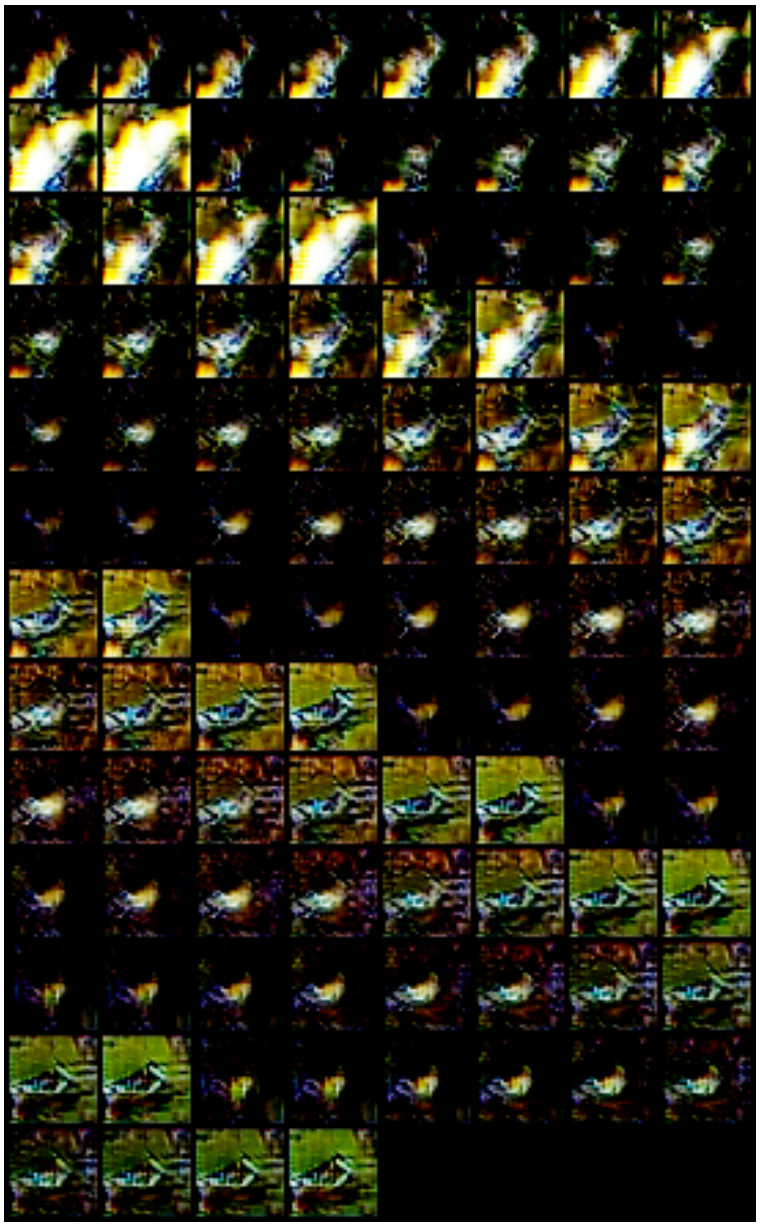


Images at 30000 epoch



Best images at 26,000 epochs

3. The latent space does not look disentangled. Even for a slight change of 1 dimension, the difference in images generated is so high for some cases, mainly row numbers 6, 7, 11, etc. Changes are non-evident for some images. I believe in the best case of disentanglement; the images should differ gradually.

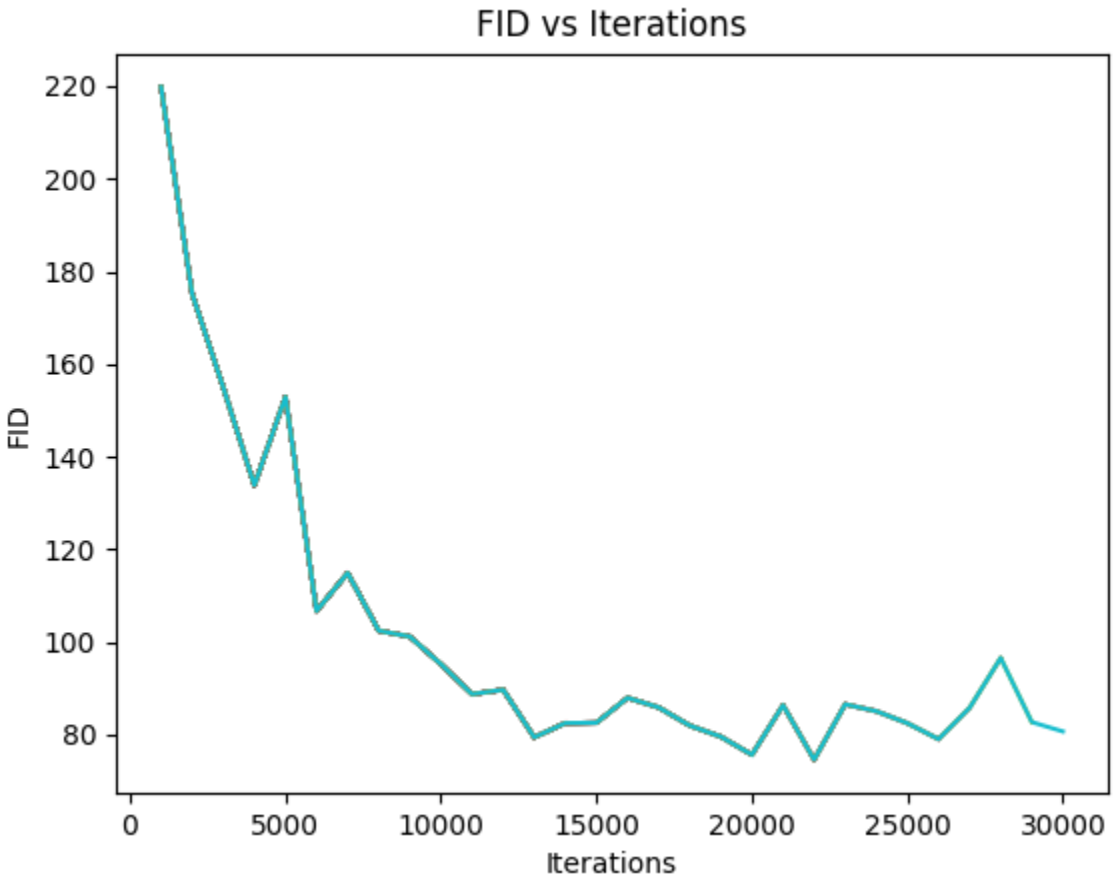


After epoch 26,000 fid increases, as well as the generated images, becomes very bad.

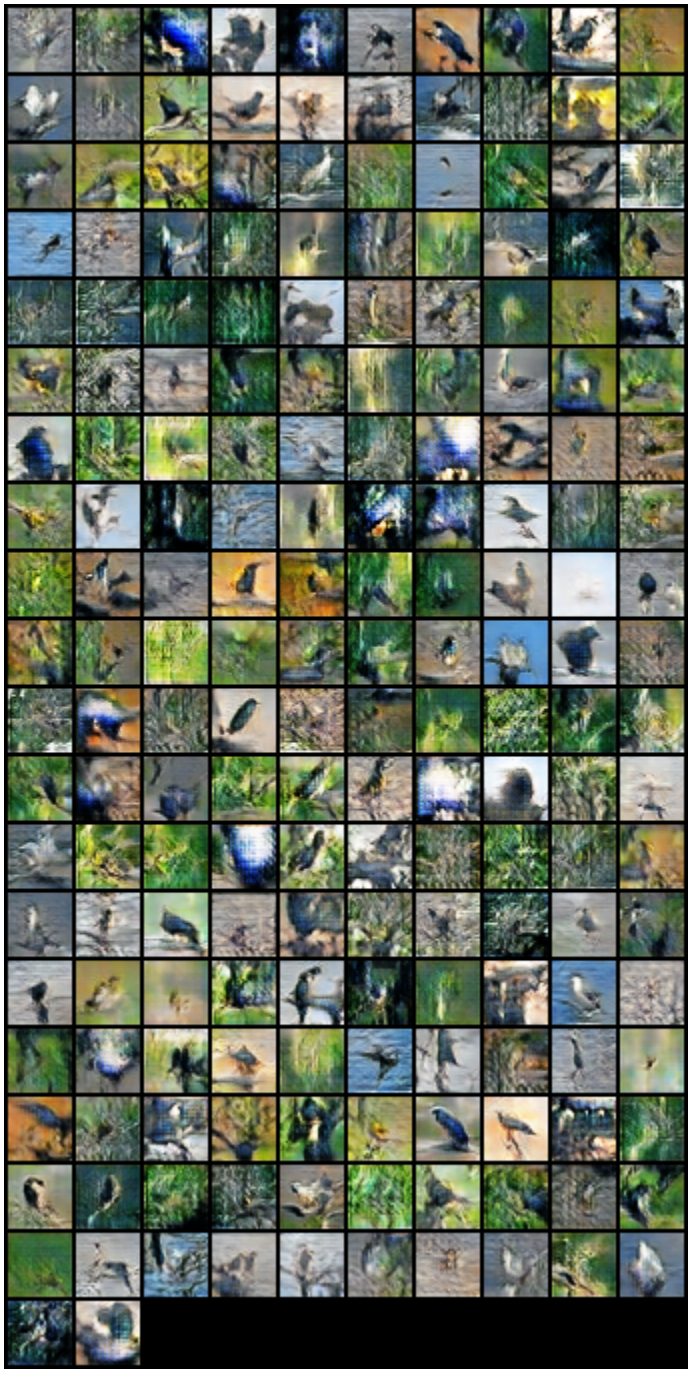
4. The training is unstable because the BCE loss is calculated after passing the discriminator outputs through the sigmoid function. Sigmoid causes vanishing gradients for a high range of values. This mainly affects the generator training because generator loss only has 1 input; i.e. the generated fake image. So this uncertainty in gradients(high chances of gradients vanishing), leads to unstable training. Regarding the bad quality, sigmoid vanishes gradients for samples on either side of 0 (values of large magnitude). Suppose a generated image after the sigmoid produces a high value, the sigmoid outputs will be close to 1, and gradients will be close to 0. No learning will be made for this sample because gradients are 0. All they are classified correctly to be similar to real images should be close to the decision boundary. So such samples cannot improve after a point for vanilla GAN.

1.4.3

1. The final FID obtained is 78.69

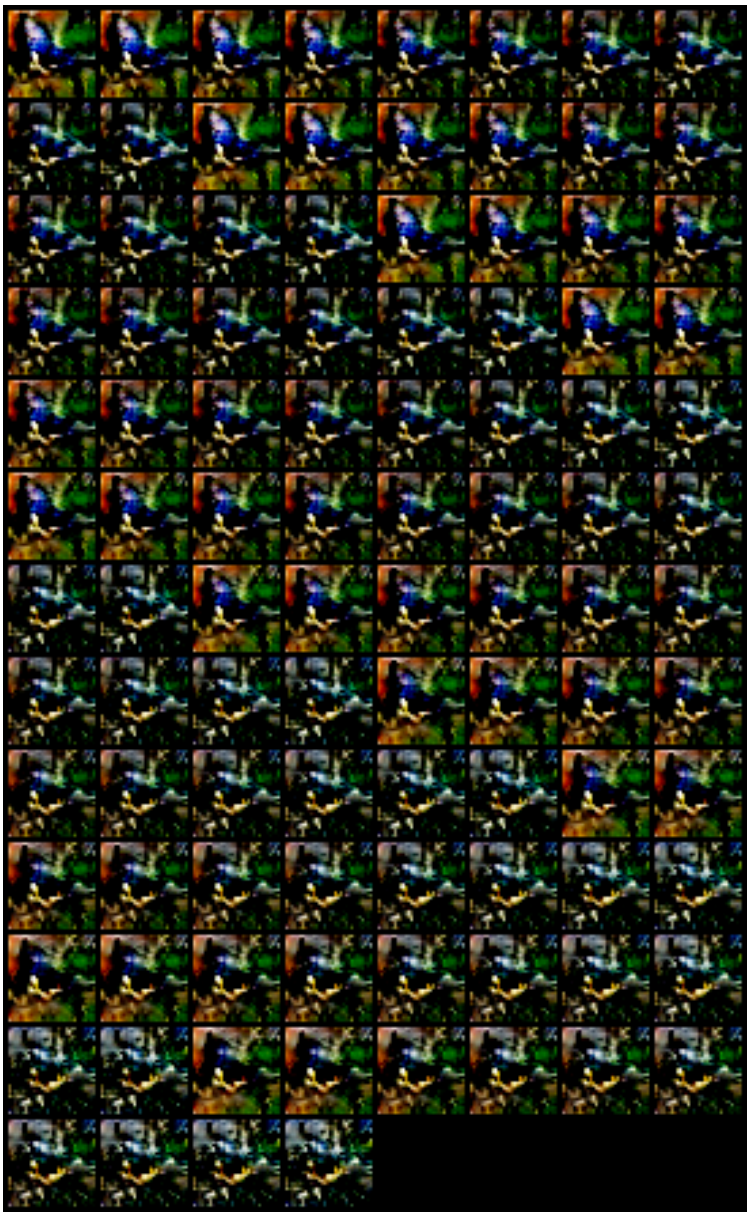


2. Samples look better than previous GAN; the images overall have good quality, and some individual images are very close to the real sample.



Images at epoch 30,000

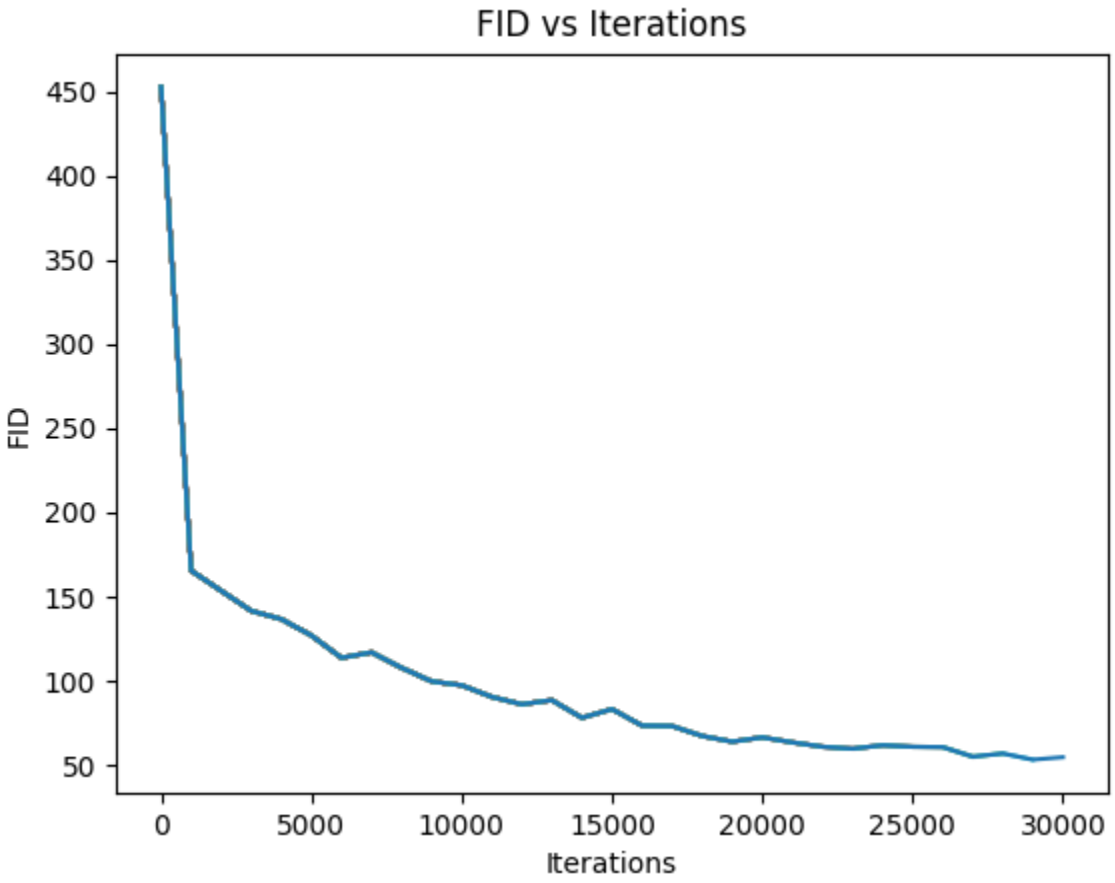
3. Latent space looks disentangled to a small extent at some points, meaning that in some rows and columns, I can see a clear gradual change, but it is nonexistent in other areas. To add to that it seems the GAN gas suffered from mode collapse as most latent spaces look similar.



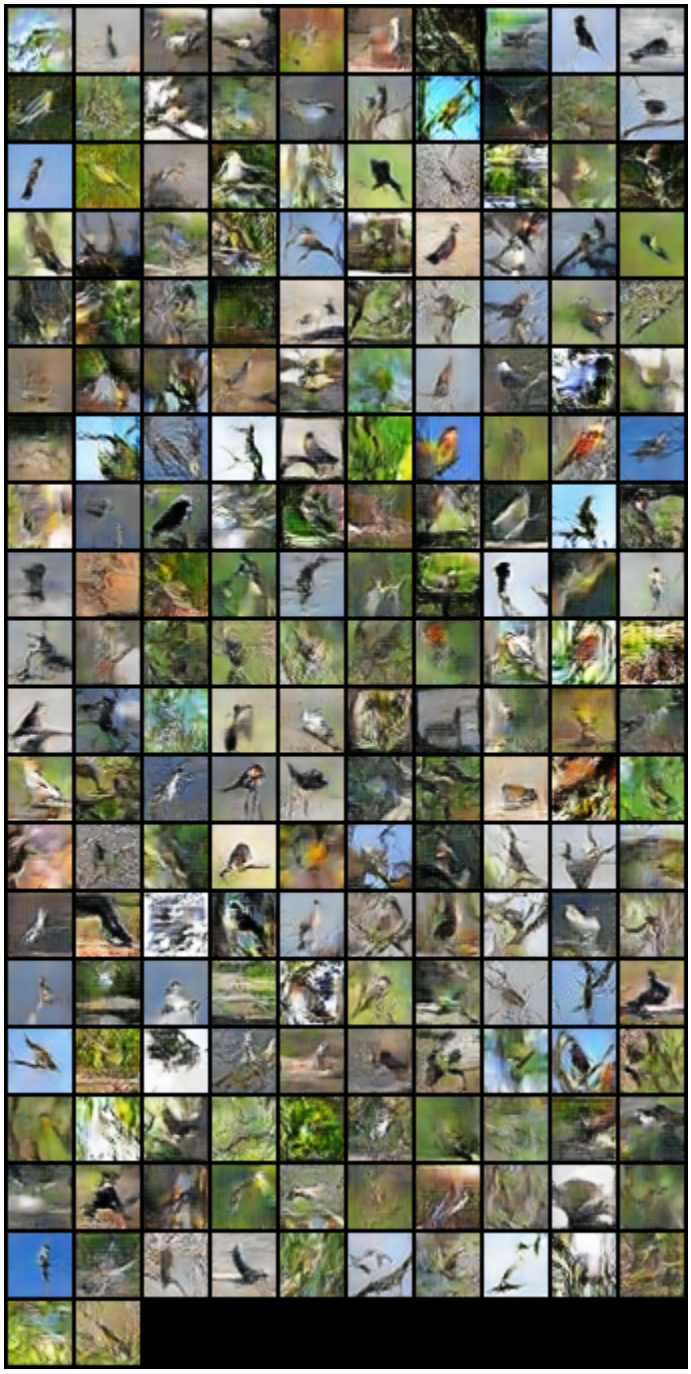
4. The main reason for the stability of this GAN is that it uses least-squares loss, which is more uniform compared to the previously used sigmoid cross-entropy loss which does not change much for values of input. LS loss provides gradients for all data even the ones that are far away from the decision boundary unlike sigmoid cross-entropy. For the adversarial sample to look real you need to be near the decision boundary. The gradients of LS loss provide gradients pointing to the decision boundary for even the generated fake samples which are not the real side of the boundary. As there are no vanishing gradients, the training is more stable, especially for generators of GANs.

1.5.3

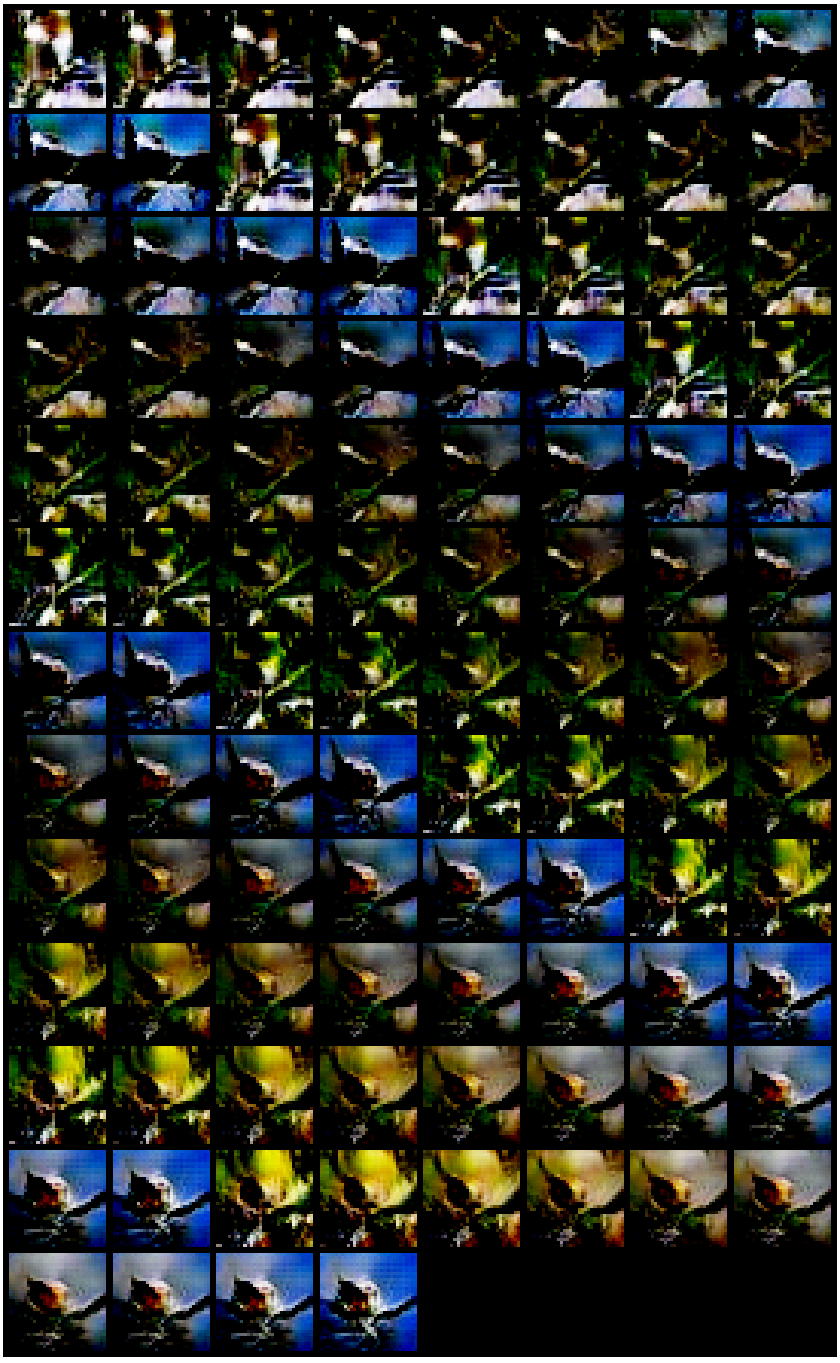
The final FID obtained is **50.8361**



2. The samples look really good, and some birds look really good. Many birds look very realistic and is evident from lower FID.

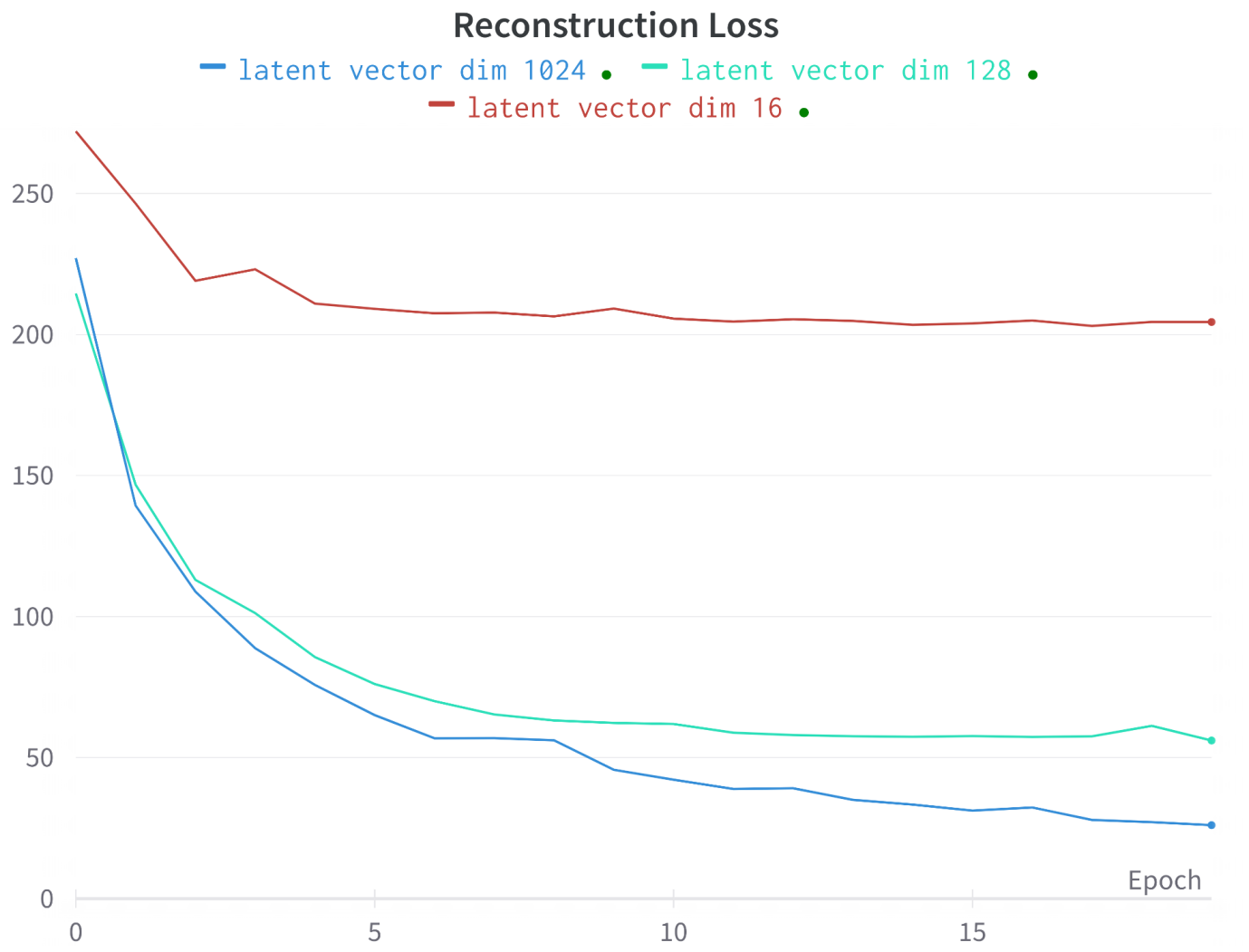


3. From my understanding, it is disentangled slightly, you can see a gradual change in some rows and diagonally also.



4. WGAN-GP training is pretty stable for a couple of reasons. It does not use a sigmoid in the calculation of loss so that the vanishing gradients are not a possibility. They enforce the earth movers' distance loss by using gradient penalty. It provides a continuous loss function with consistent gradients. So the training will be very consistent without any collapse. Although LS-GAN did not have a sigmoid, WGAN-GP is better because the loss is more continuous than LS-GAN, because LS-GAN has a target label to converge (loss had a difference between 1 and 0, similar to classification), but here we maximize the expectation with some gradient penalty. Although LS-GAN had gradients everywhere they were directed to the decision boundary; away from the boundary the magnitude of gradients was lower(but better than BCE Loss), but when we force earth movers distance, that boundary is almost linear(similar to smooth gradients everywhere) so the quality of the samples are also better.

Question 2.1: AutoEncoder



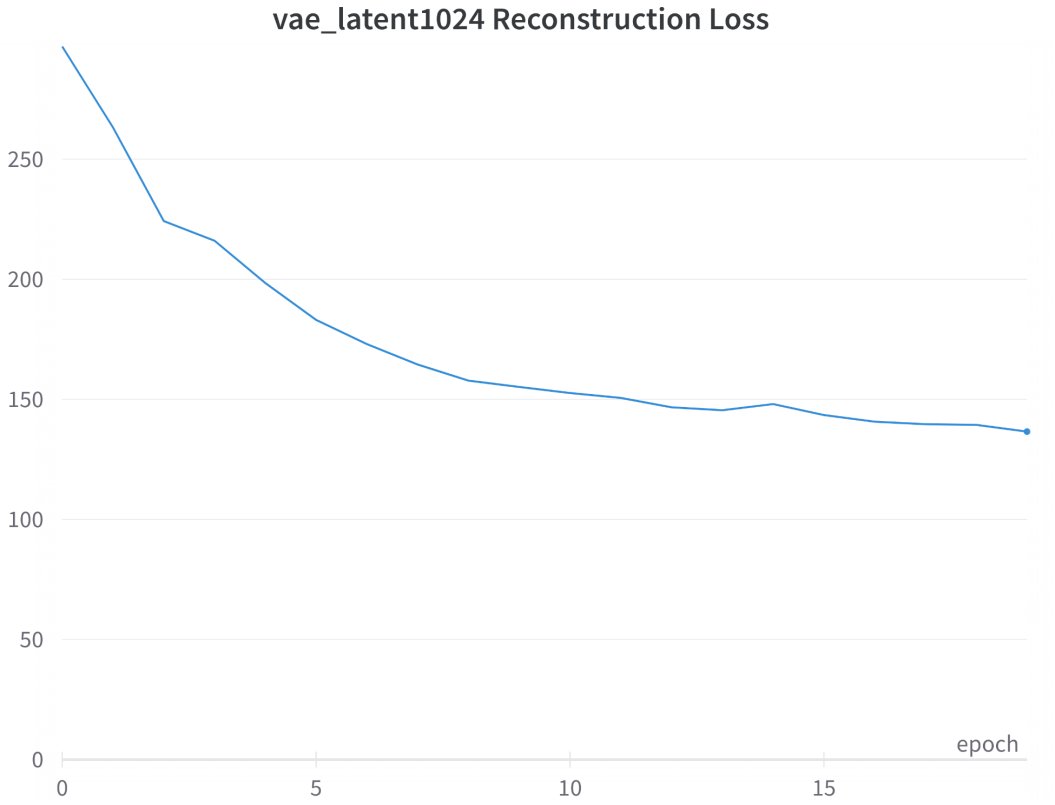
Latent vector 16

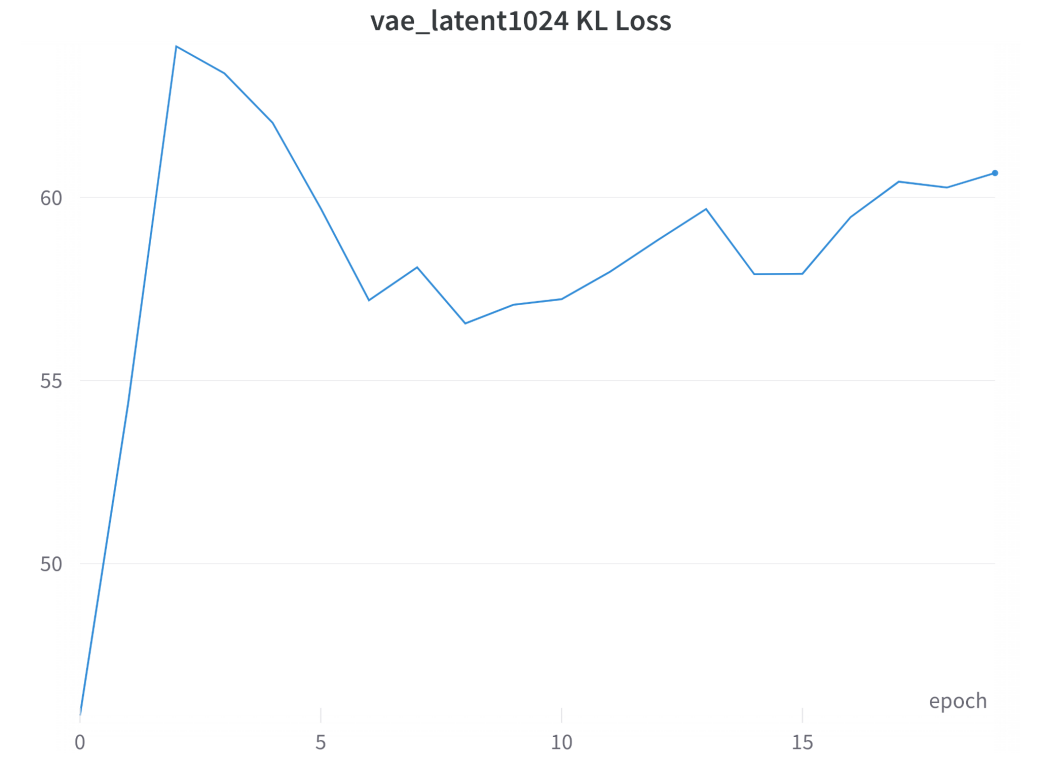
Latent vector dim 128

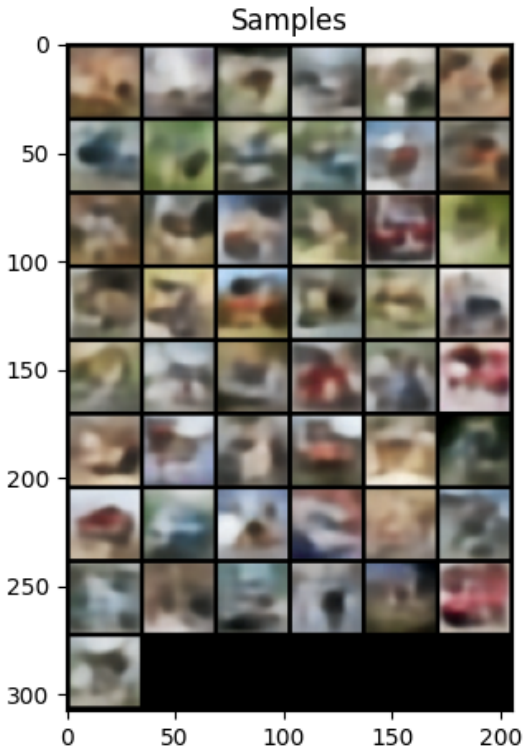
Latent vector dim 1024

The best reconstruction is obtained for latent vector dim=1024, followed by 128 and then 16. The possible reasons should be that a vector of dim=1024 retains more important higher dimensional features compared to 128 and 16. So while reconstruction the decoder has a large number of quality features to generate outputs from. More precisely if latent dim is 128, the network trains ti=o squeeze all the important features to a bottleneck dimension of 128, which might not be sufficient to represent the data completely.

Question 2.2: Variational Auto-Encoder (10 points)
Final Reconstruction loss at epoch 19: **136.563**
Final KL divergence at epoch 19: **60.671**





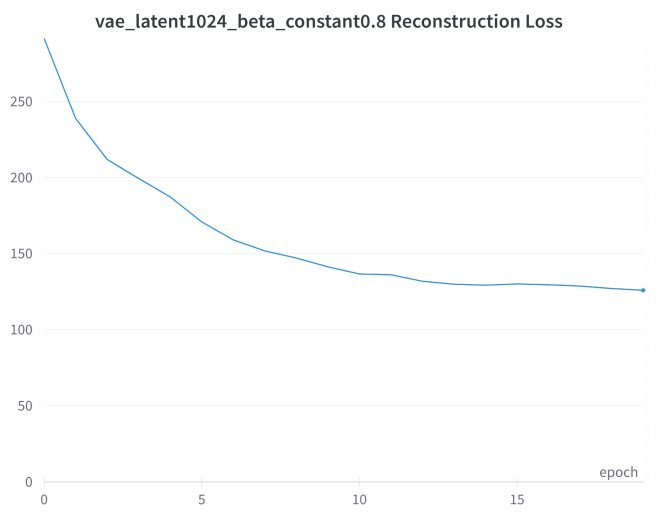


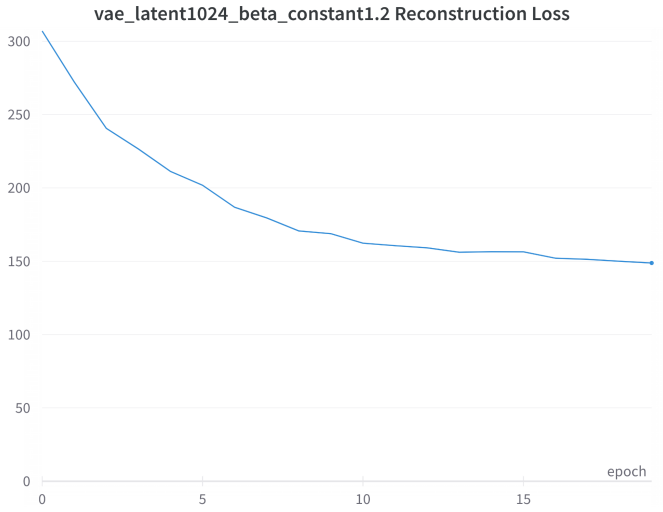
Sampled Images at epoch 19

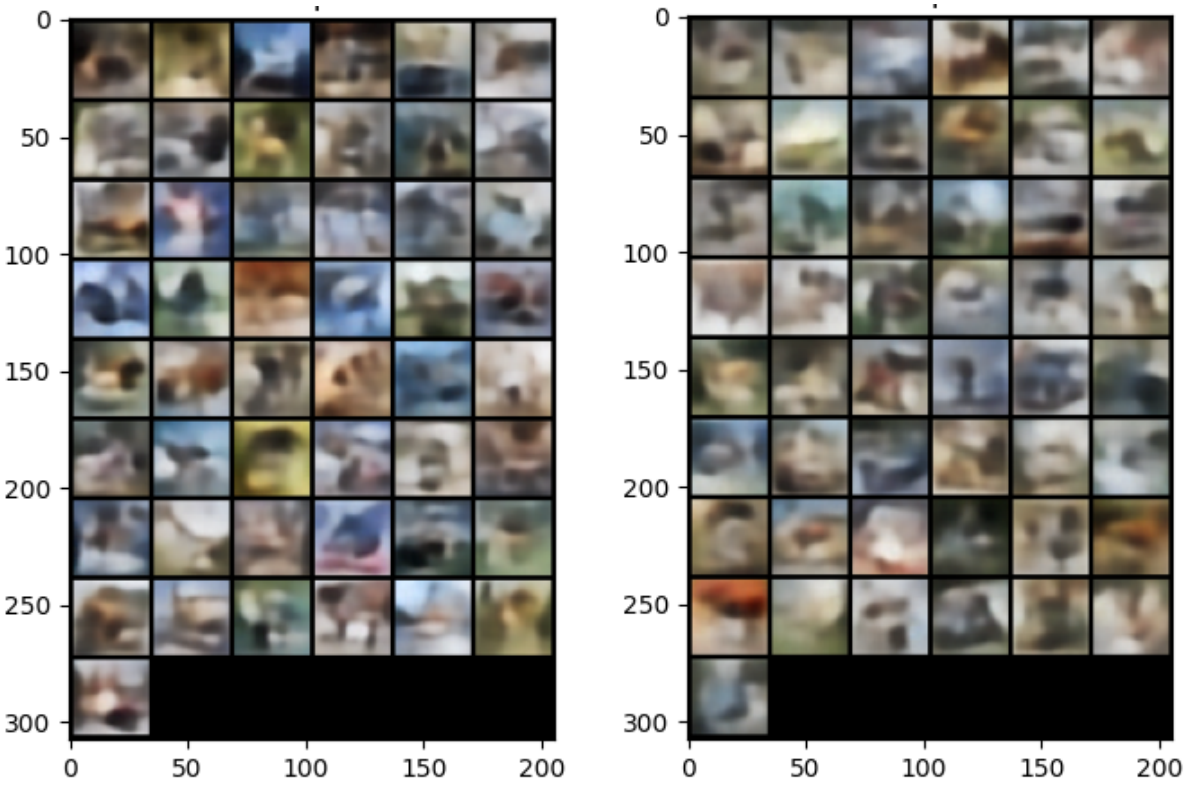
Reconstructed samples at epoch 19

Question 2.3: Beta Variational Auto-Encoder (10 points)

	Beta = 0.8	Beta = 1.0	Beta =1.2
Final Reconstruction loss at epoch 19:	**125.861**	**136.563**	**148.81**
Final KL Loss at epoch 19	**70.373**	**60.671**	**52.774**

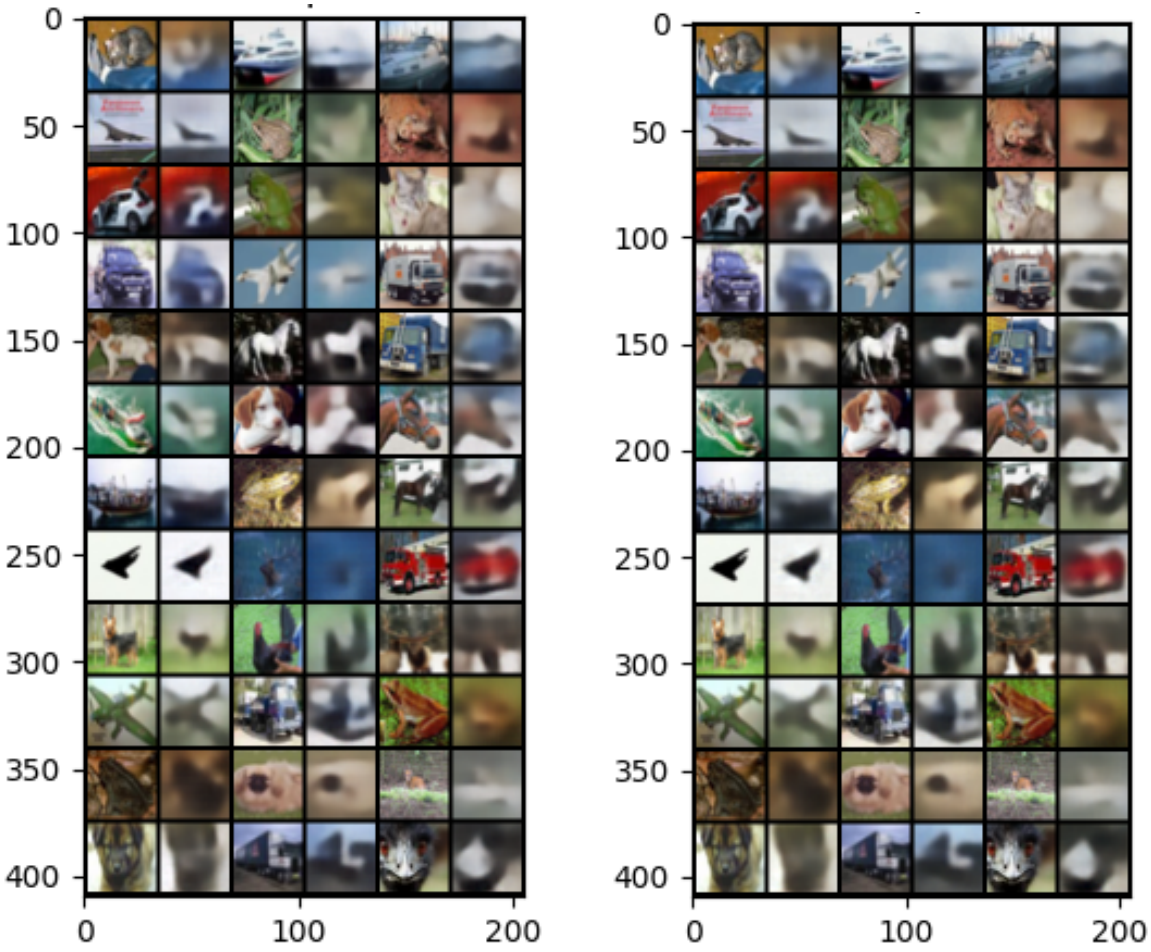






Beta = 0.8 Beta = 1.2

Sampled outputs at epoch 19



Beta = 0.8 Beta = 1.2

Reconstructed outputs at epoch 19

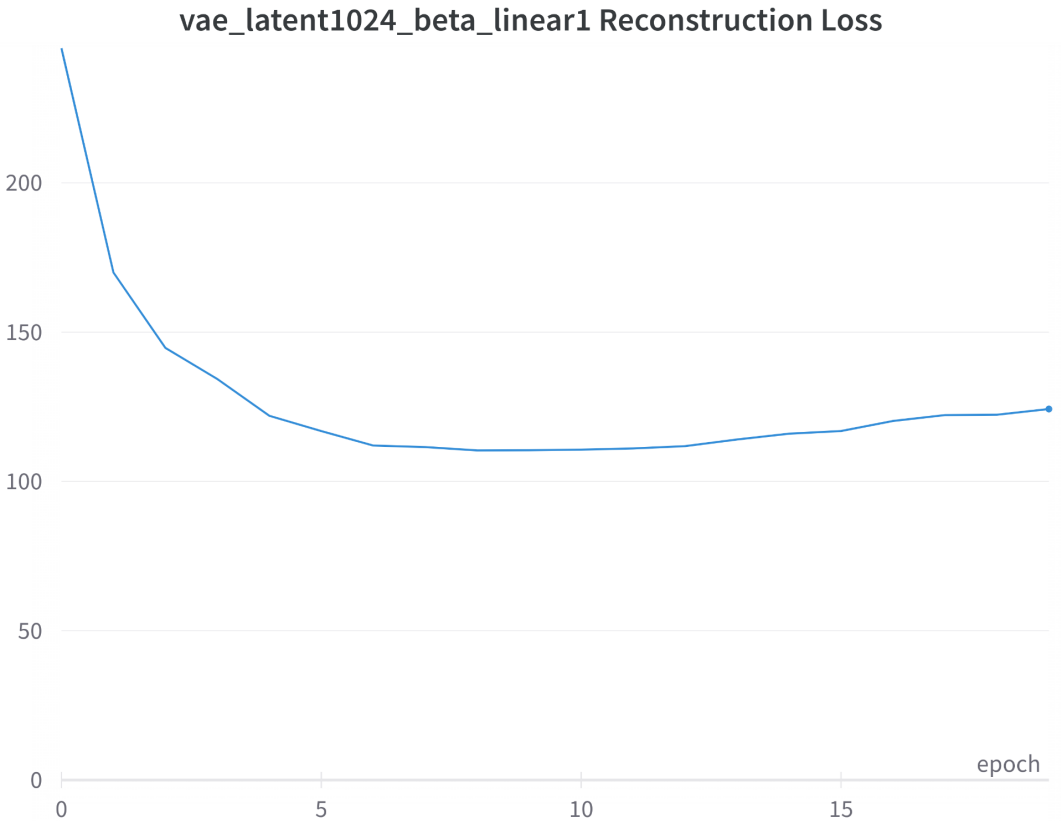
Compare the performance of the models with beta values 0.8, 1, 1.2. (Recon loss at epoch 19 of reference solutions are < 130 for beta0.8, and <155 for beta1.2)Comment on the recon loss, kl loss and quality of samples. (Even with tuning, the samples will still be blurry)

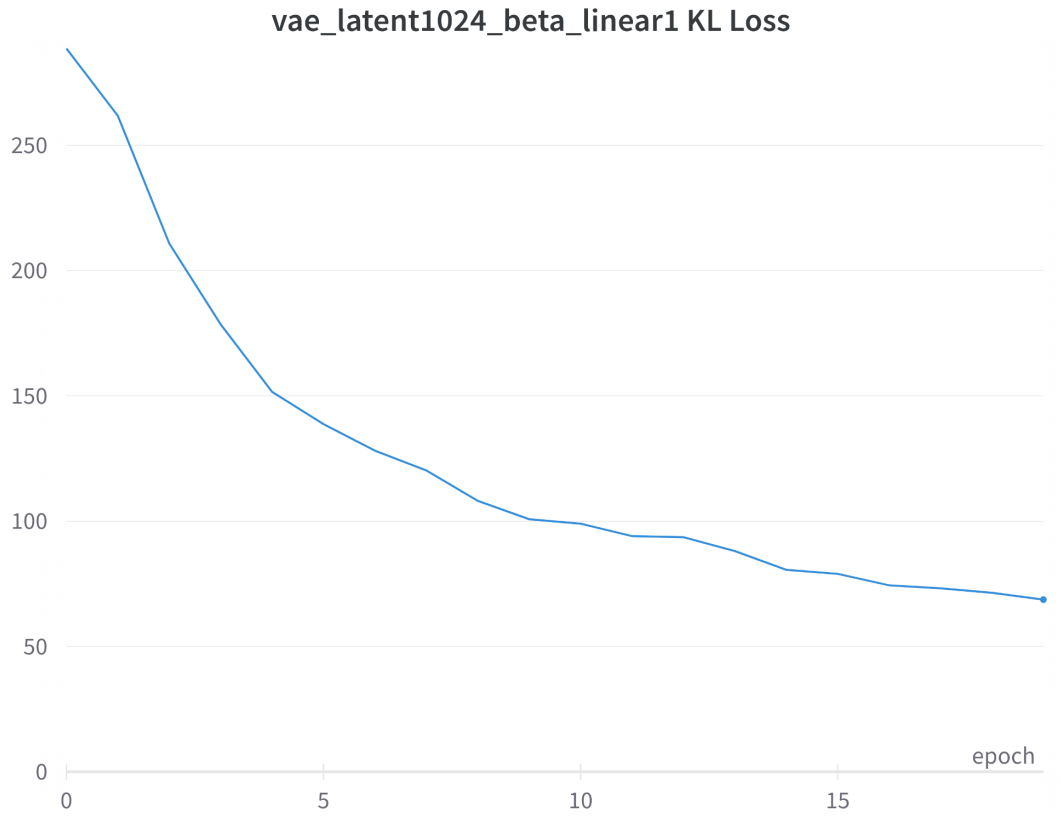
Comparing the reconstruction losses of different betas, the best performance is shown for beta =.8, the reconstruction loss obtained is 125.58, which is lower than 136 for beta = 1 and 148 for beta =1.2. But kl divergence is lower for beta =1.2 followed by beta = 1 and highest for beta = 0.8. So increasing the weightage of a loss in the final loss term increases the extent to which it is optimized. In beta=1.2, KL loss is given higher priority so the model is biased towards kl divergence reduction more. The best quality reconstruction is found for beta = 0.8, but all beta value reconstructions seem almost comparable. The sampling for beta=0.8 is less blurry compared to beta = 1.2

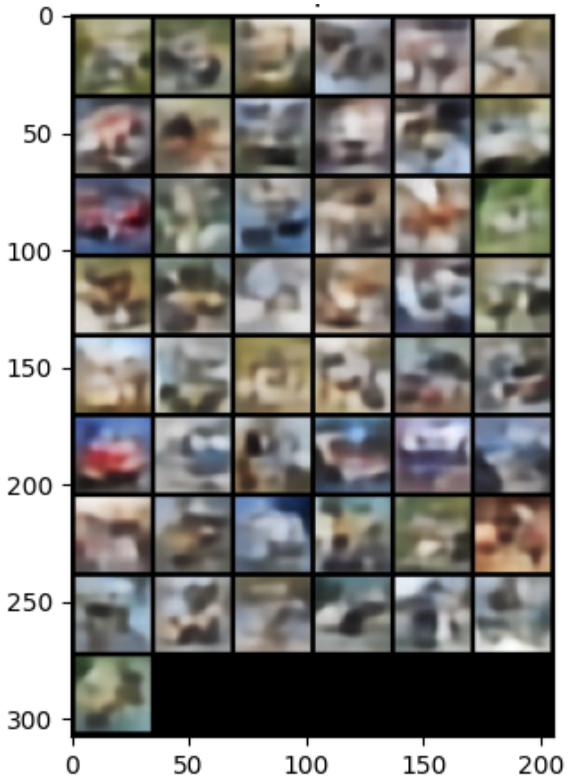
For what value of beta does the VAE reduce to an auto-encoder?

For beta = 0, the kl loss will be zero, then VAE is similar to AE in terms of that network only optimizes to reduce the reconstruction loss. But still, it is not still completely AE because the reparameterization trick is involved. If we avoid the reparameterization trick and make beta = 0, VAE reduces to an Autoencoder.

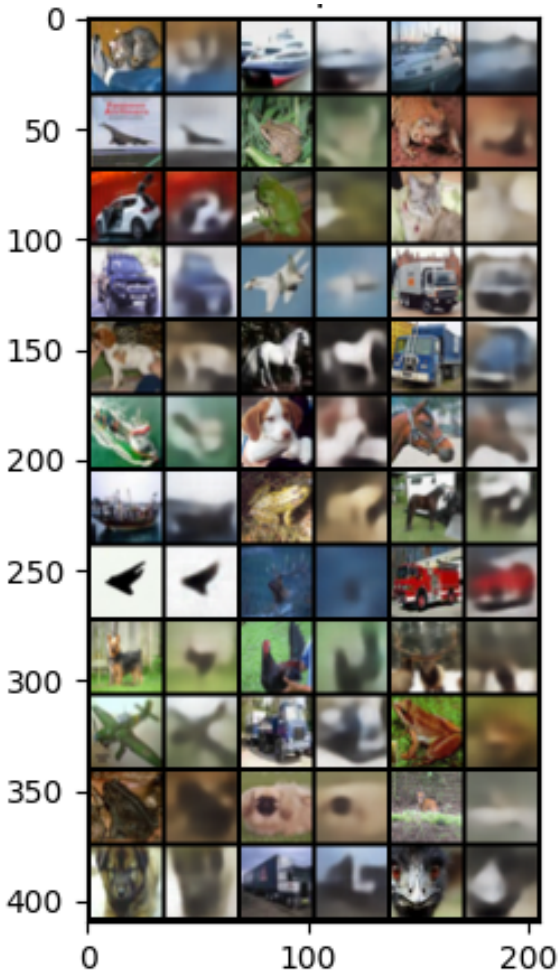
Question 2.3.2: Linear schedule for beta
Final Reconstruction loss at epoch 19: **124.263**
Final KL divergence loss at epoch 19: **68.69**





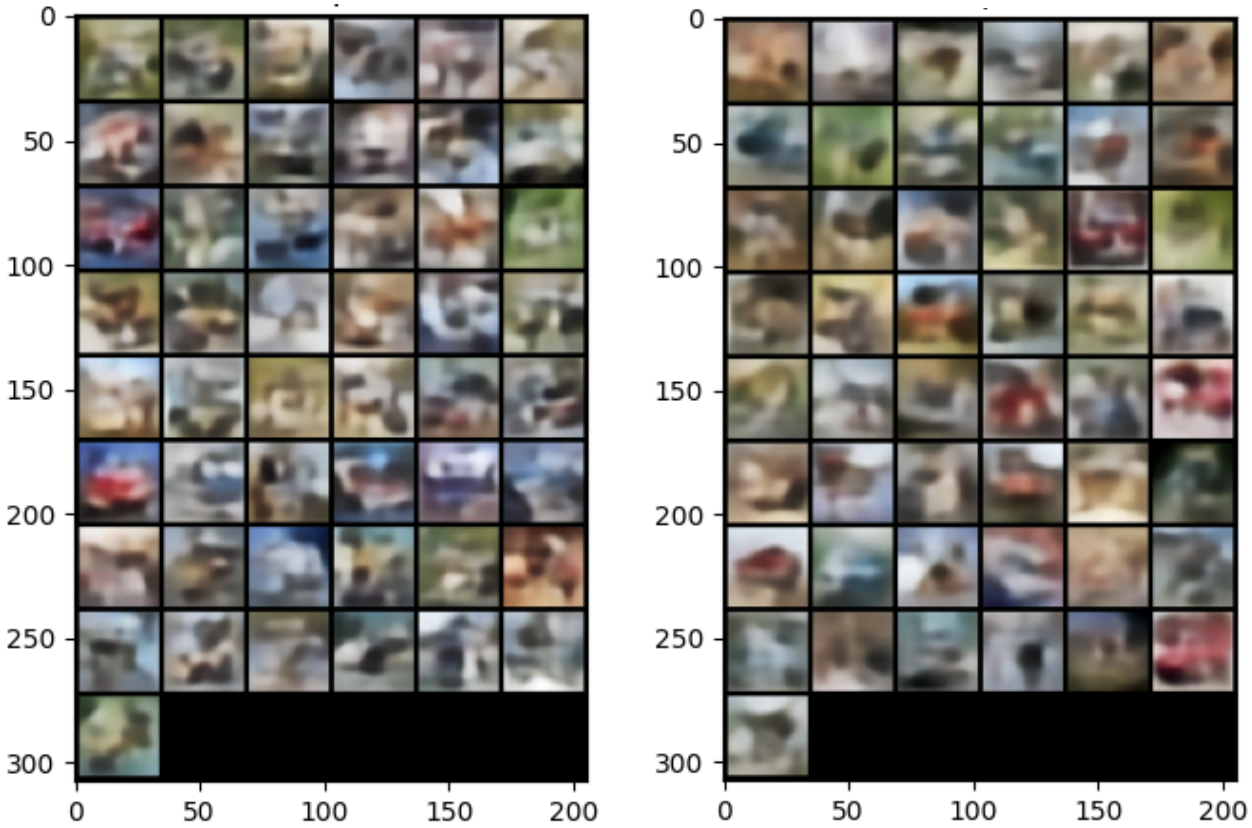


Sampled outputs



Reconstructed outputs

Comparing to Vanilla VAE

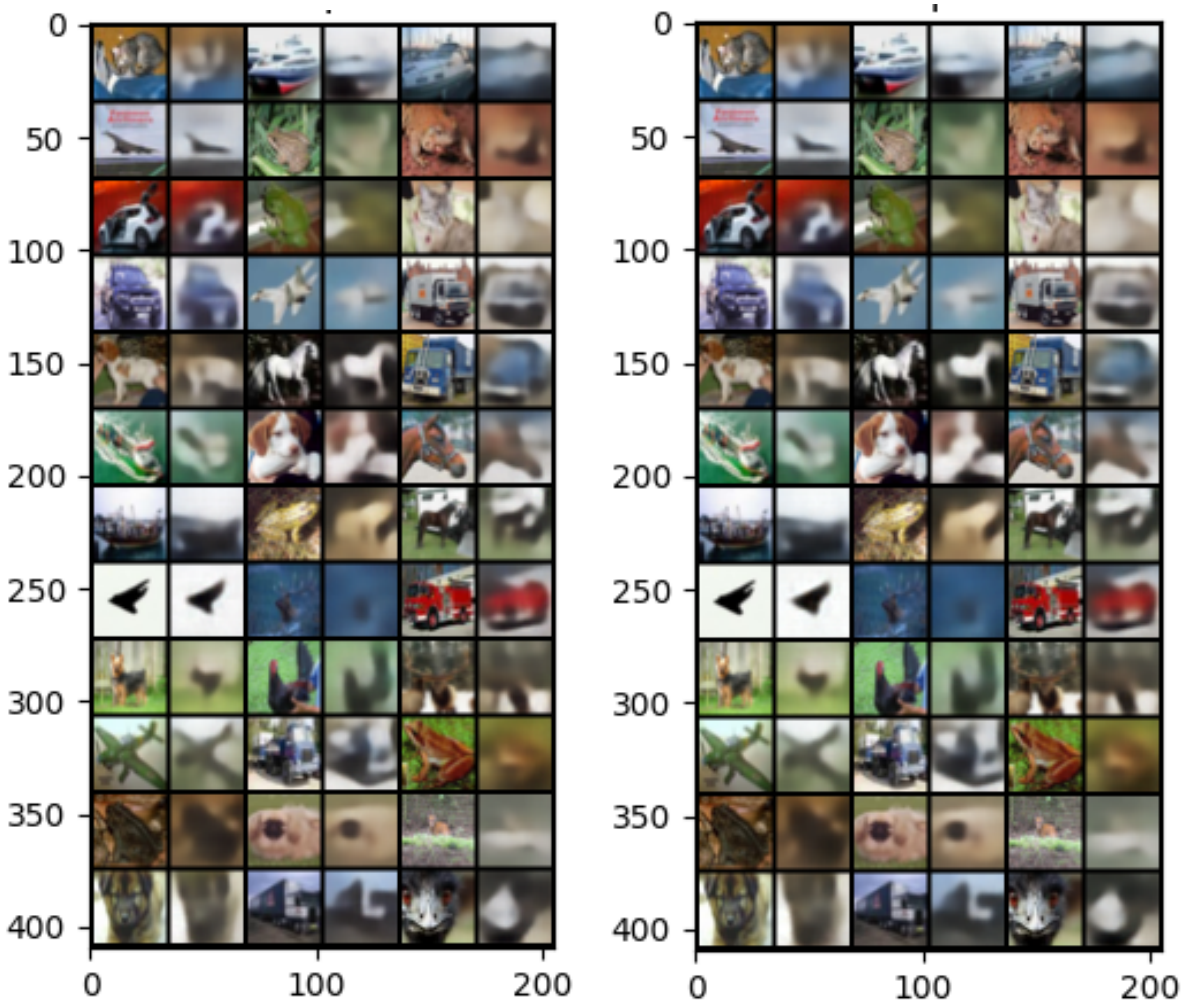


Linear scheduled beta Vanilla VAE

Sampled Images at epoch 19

Comparing this linear scheduled beta produces better results. Although both are blurry we can see various different shapes in Linear scheduled beta.

	Linear scheduled beta	Vanilla VAE
Reconstruction Loss	**124.263**	**136.563**
KL loss	**68.69**	**60.671**



Linear scheduled beta Vanilla VAE (beta =1)

Reconstructed Images at epoch 19

The reconstruction loss is lower for linear scheduled beta, but the reconstructions are both blurry but linear scheduled beta seems to be slightly better at least quantitatively.