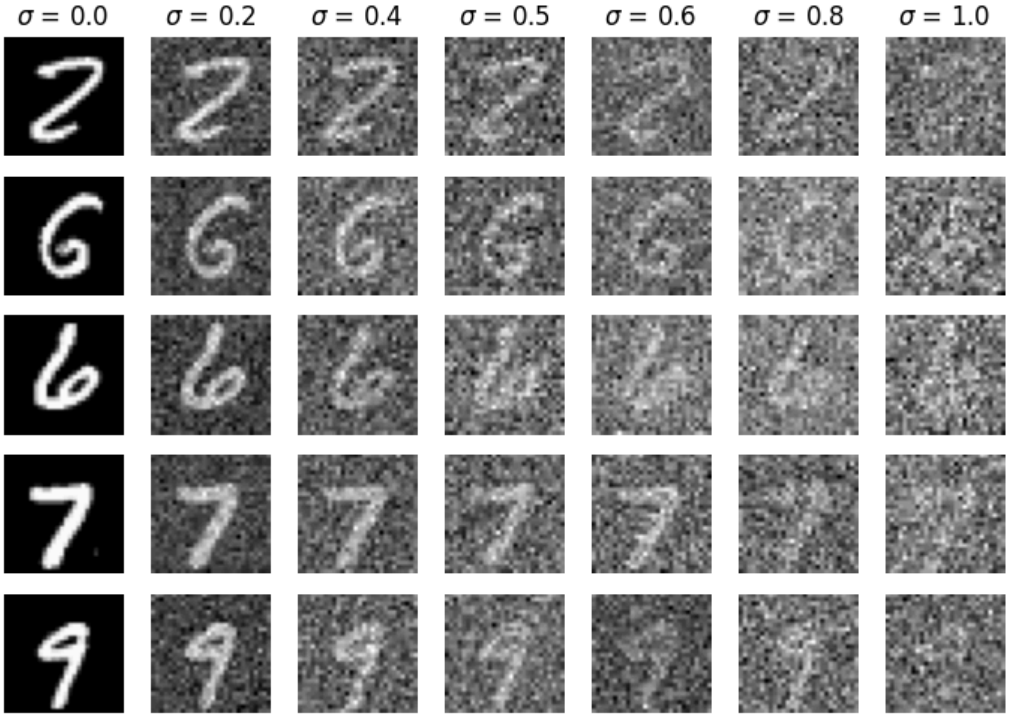
# Assignment 5: Diffusion Models

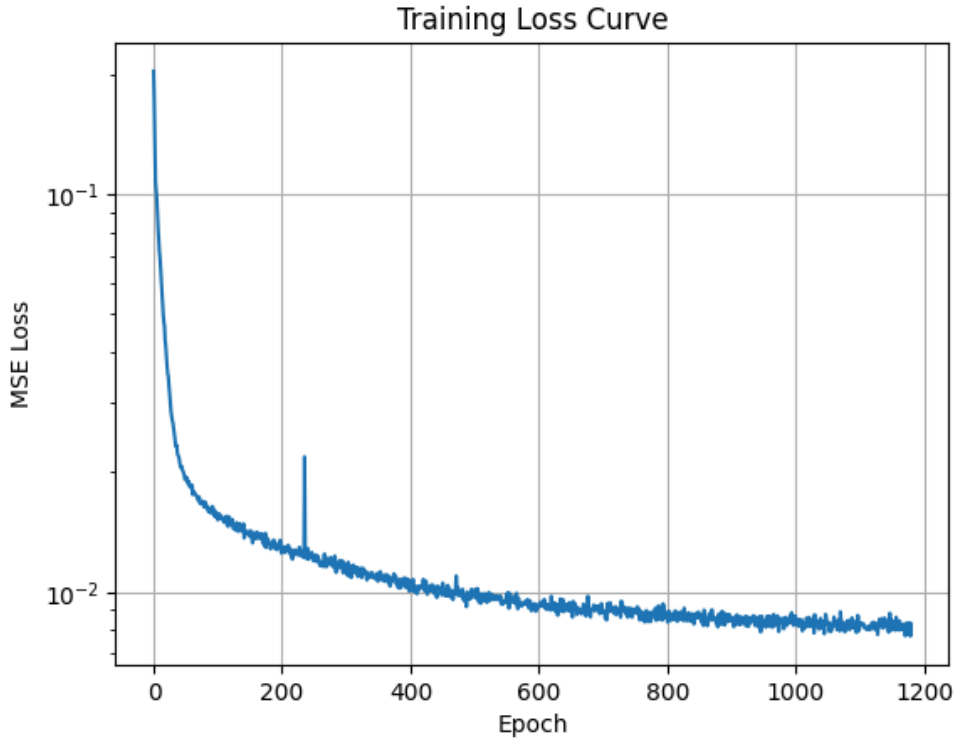
Name(s): Girish Madhavan Venkataramani
NetID(s): gmv4

**Part 1: Single-Step Denoising UNet**

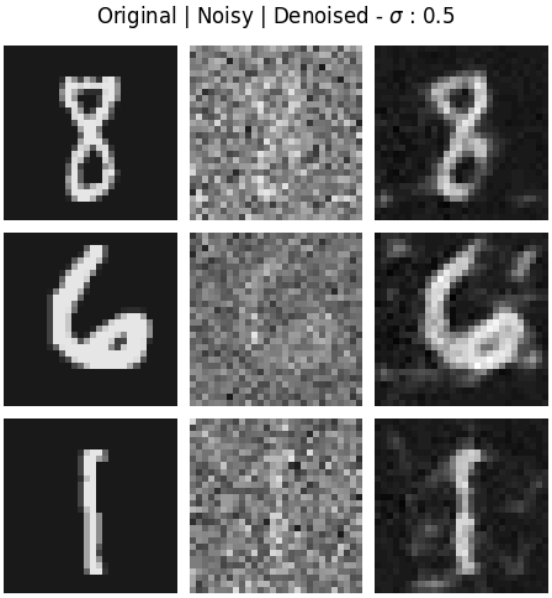
1. A visualization of the noising process using $\sigma$ = [0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0] (Figure 3)



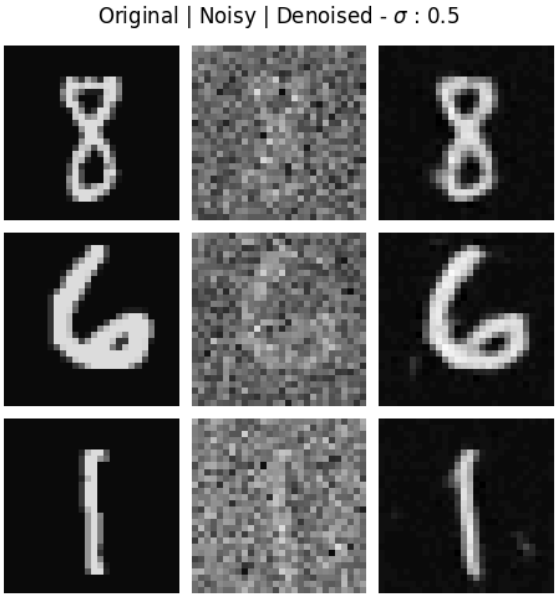
2. A training loss curve plot every few iterations during the whole training process (Figure 4)




3. Sample results on the test set after the first and the 5th epoch. (Figures 5 and 6)

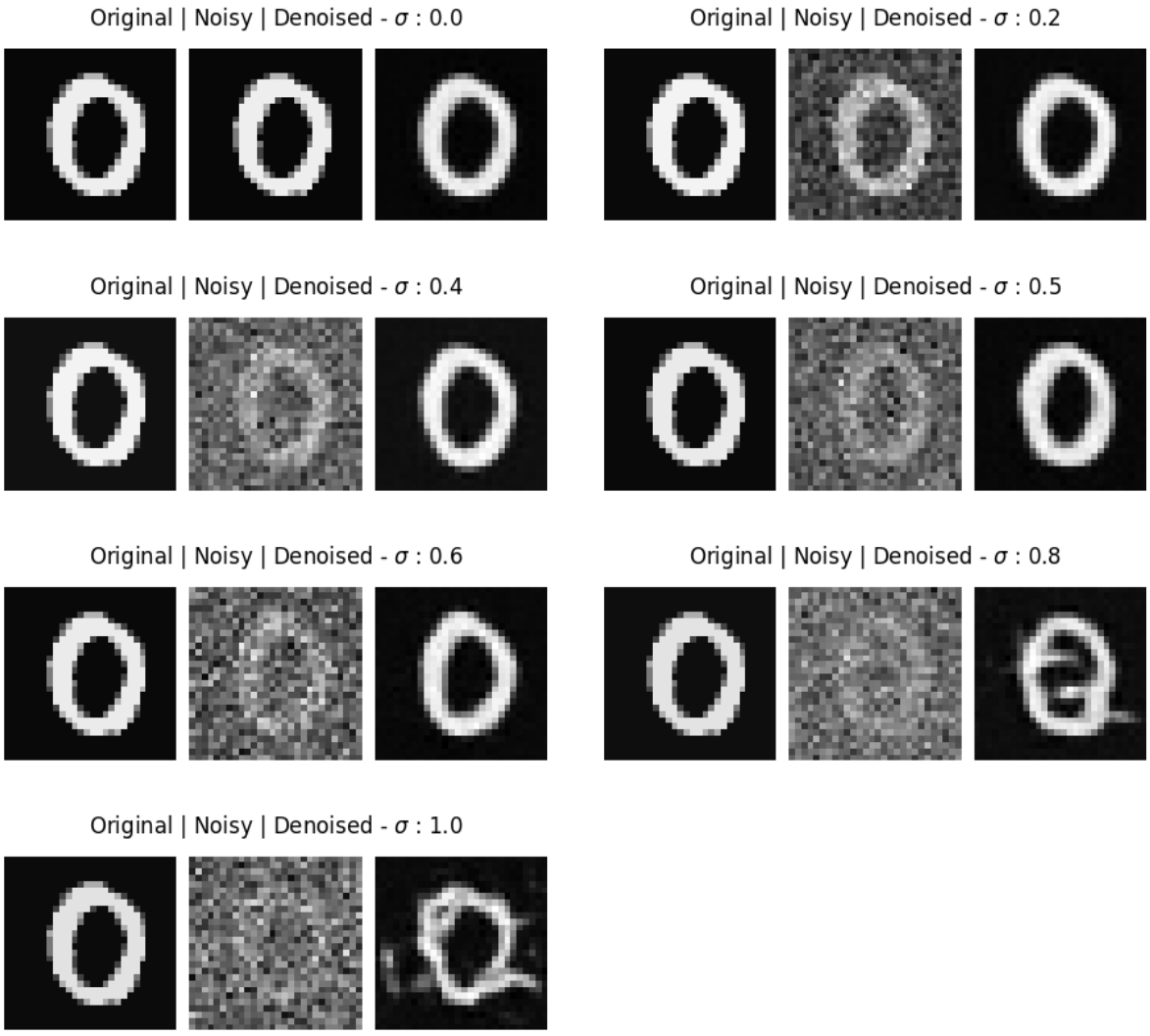



After 1 Epoch



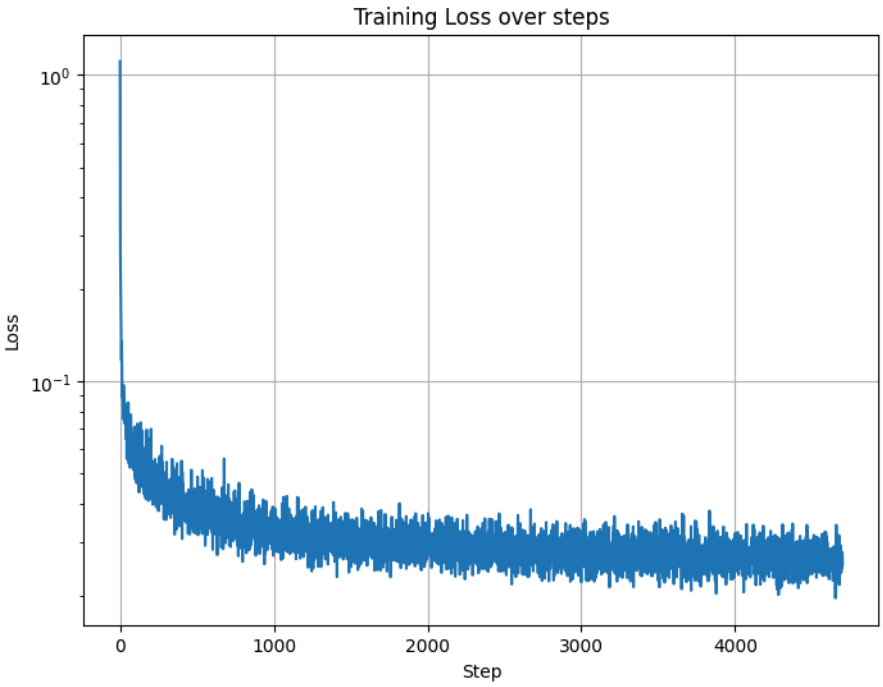

After 5 Epochs

4. Sample results on the test set with out-of-distribution noise levels after the model is trained. Keep the same image and vary $\sigma$ = [0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0] (Figure 7)

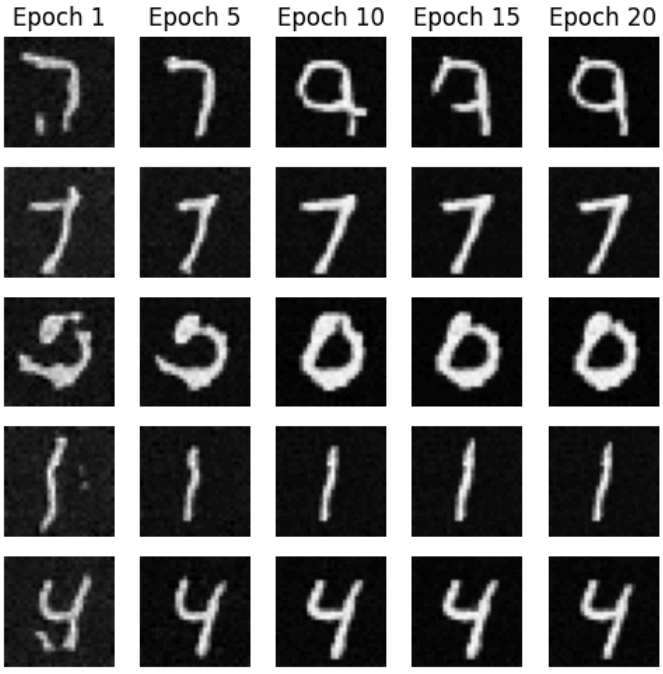


**Part 2.1-2.3: Time-conditioned UNet**

1. A training loss curve plot for the time-conditioned UNet over the whole training process (Figure 10).



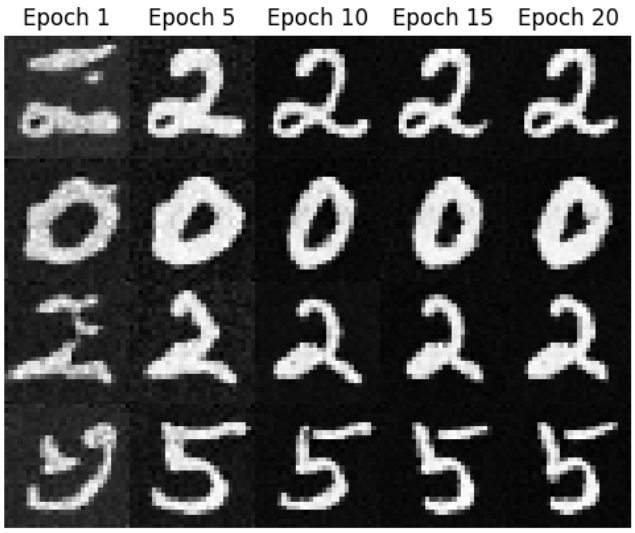

2. Sampling results for the class-conditioned UNet after the 5th and the 20th epoch.



**Part 2.4-2.5: Class-conditioned UNet**

1. A training loss curve plot for the class-conditioned UNet over the whole training process.

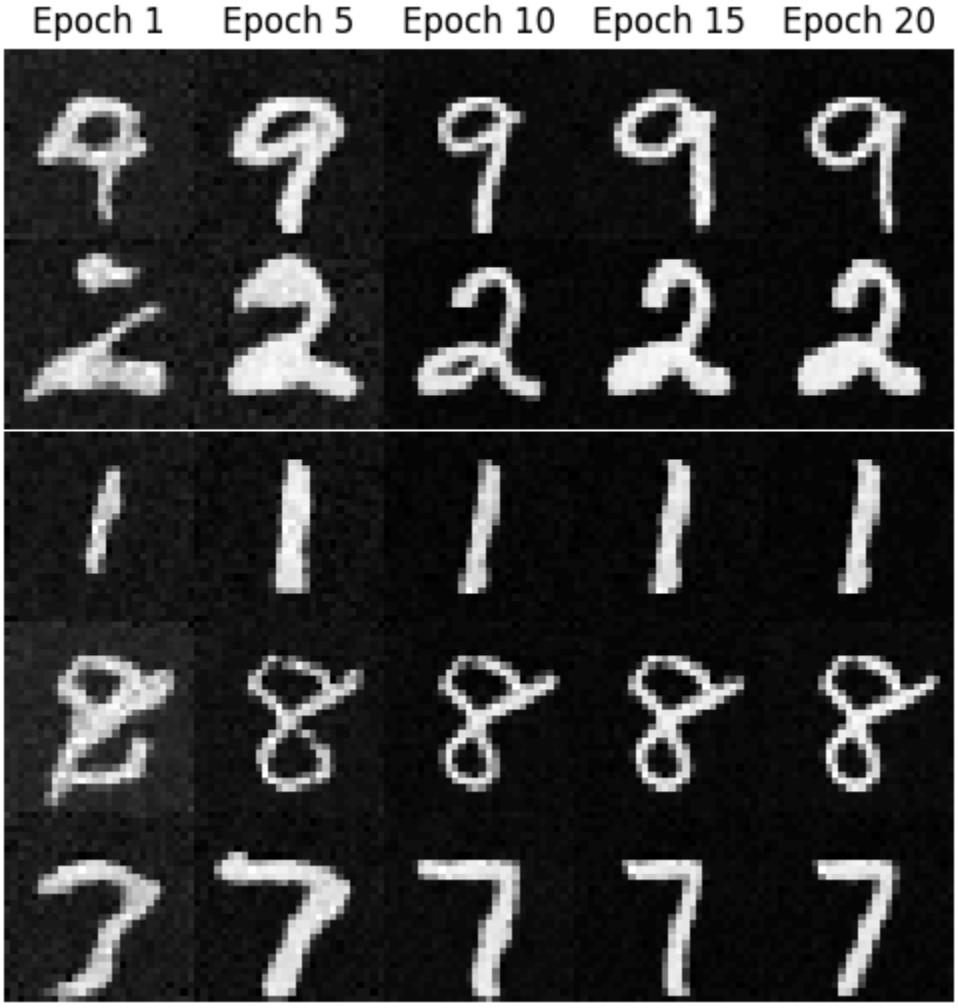

2. Sampling results for the class-conditioned UNet after the 5th and the 20th epoch. Generate 4 instances of each digit as shown on the assignment page. Providing a GIF is optional and can be done as extra credit.



**Extra credit:**

- Gifs generated and submitted separately, as pdf files can't display gifs
  The Gifs are placed in a folder in extra_credits.zip and these are for Class conditioned Unet

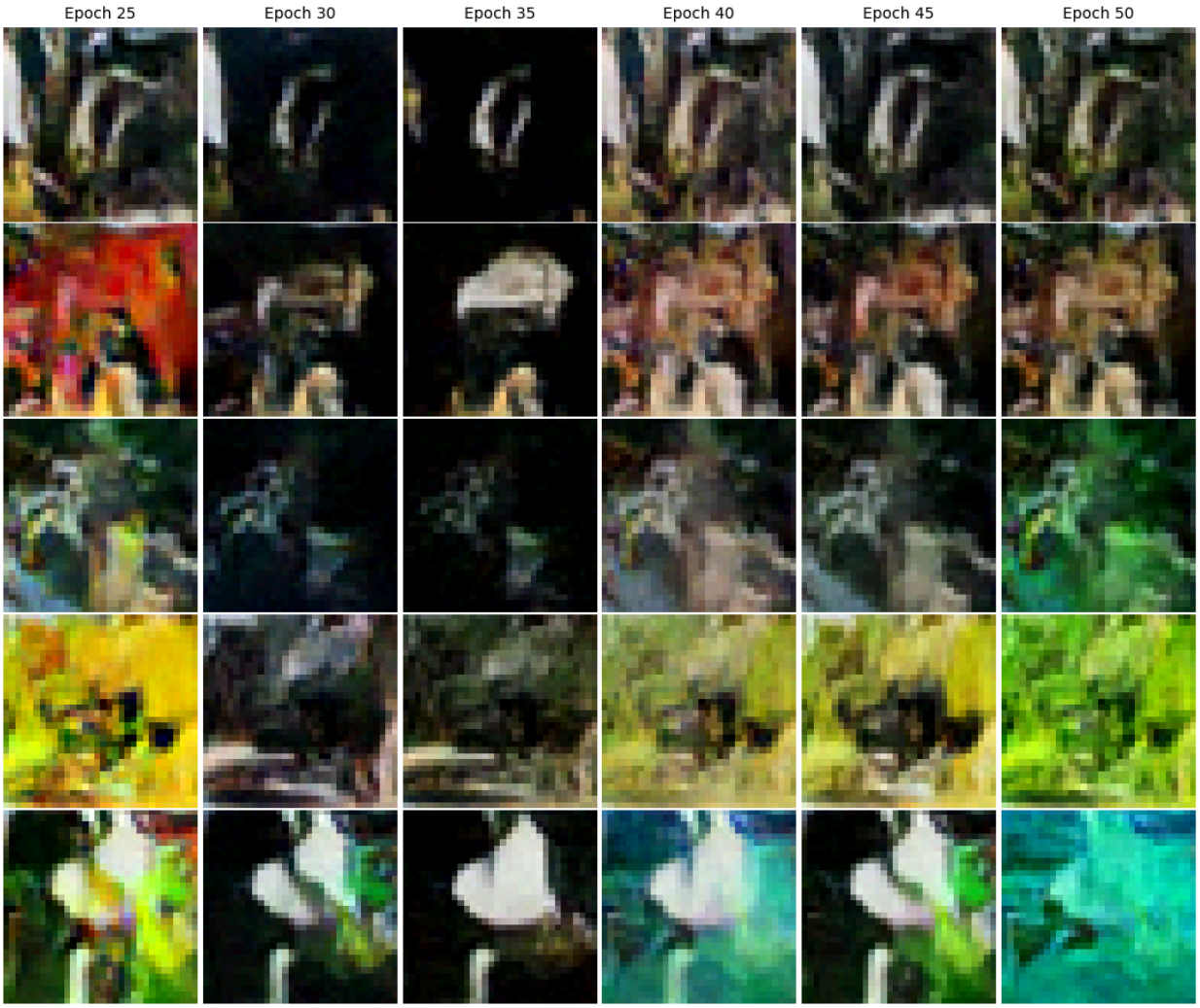
- Digit Classifier :
  - Digit classifier accuracy after 20 Epochs : 99.07%



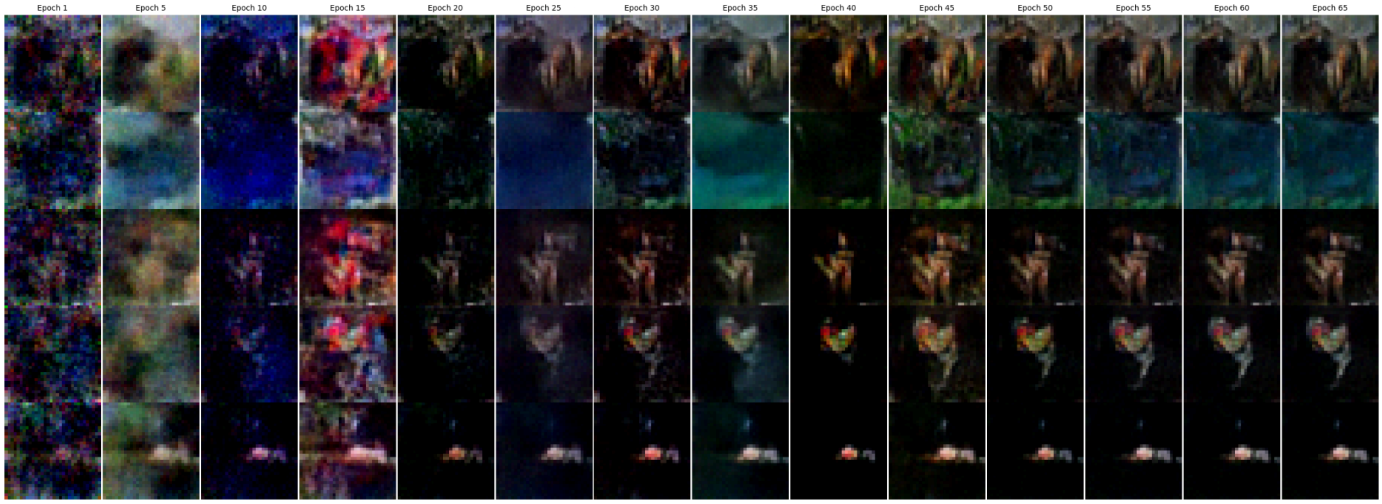

predicted : tensor([9, 2, 1, 8, 7], device='cuda:0')
correct : tensor([9, 2, 1, 8, 7], device='cuda:0')

The results are similar to classifier free guidance

- Time conditioned Unet trained on CIFAR 100 dataset :

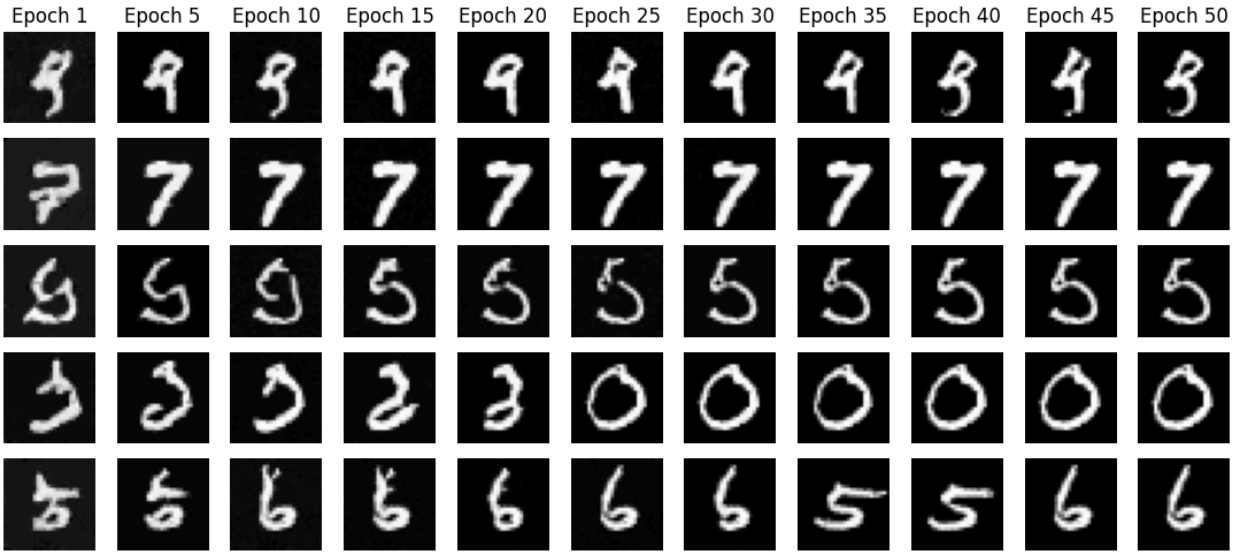



- Improved Time Conditional Unet Architecture :
    - Added skip connections inside ConvBlock and one skip connection from after the first block to before conv2d after upscaling
    - Added sinusoidal positional time encoding
    - Tried self-attention but didn’t improve performance, maybe due to the network architecture not being complicated enough for expresiveness




Gives slightly better rendering, the first image has some resemblance to a horse, the second one looks like a dog in some frames, etc. Also this used lesser number of hidden channels (D) as compared to previous case, Entire image in extra_credits.zip

- Rectified flow :




The lack of complexity in the Unet makes it difficult to further reduce the errors of the rectified flow model, as it saturates around 0.08 and doesn't go down. Similar situation can be observed with the contents of mp5 when trained for longer epochs