CS726-Denoising-Diffusion-Probabilistic-Models

Meet Doshi
22m0742

Saswat Meher
22m0804

February 2023

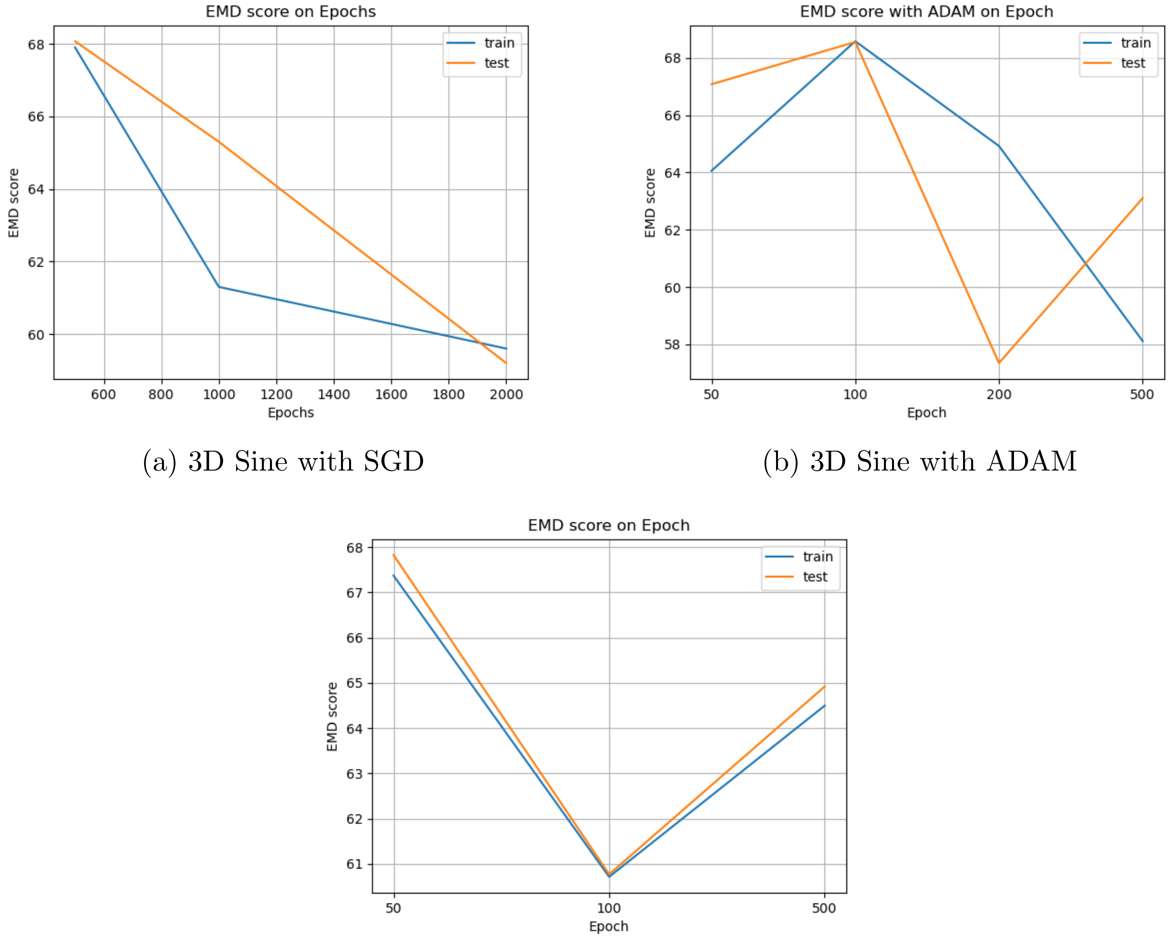
1 Architecture Information

Diffusion models are the current trend in generative models along with some architecture based GANs and normalizing flows [3]. Initial architectures like Original [1] and DDPM[2] showed performance capabilities of denoising models on image generation tasks but they were significantly outperformed by OpenAI's [4] model using cosine noise schedules and various other important improvements. The torch modules for time encoding and sequential neural network were tested with various hyperparameters like modifying number of layers in the model, trying learned vs static time embeddings like sine and cosine. We tried out different learning rate schedules like linear, squared, cubed, sigmoid, logarithmic, cosine, etc.

2 Hyperparameter Tuning

2.1 Training Steps

We started with stochastic gradient descent but after some training we realised ADAM optimiser would be better so we switched our optimizer and immediately saw better results. For Sine and Helix we found 150 and 100 epochs giving best results on test data. Other hyperparameters include n_steps=50, l_beta=2e-5 u_beta=1.28e-2 and a simple 3 hidden layer architecture corresponding to model 3.



(a) 3D Sine with SGD

(b) 3D Sine with ADAM

(c) 3D Helix with Adam

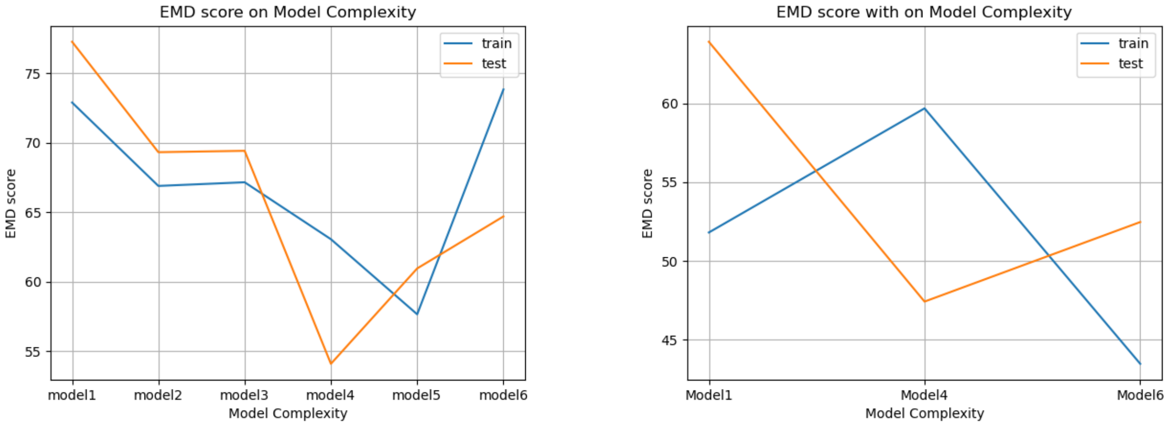
Figure 1: Comparison of number of training steps.

2.2 Model Complexity

We tried incresing complexity of models each with ReLU activations after each layer except the last layer. We found out model 4 and 6 with 3 and 5 hidden layers each respectively performed best overall in both datasets. We used 100 training steps from the previous iteration as it gave best approximate results. Keeping other hyperparameters same as previous section we just modify number of epochs to 100 for 3D helix and 200 for 3D sine dataset.

Model Complexity	
Model Name	Hidden Layer Size
Model 1	X_t,32,3
Model 2	X_t,32,32,3
Model 3	X_t,64,64,3
Model 4	X_t,64,128,64,3
Model 5	X_t,64,128,128,64,3
Model 6	X_t,64,128,256,128,64,3

Table 1: Comparison of different hidden layer combinations over 3D sine and helix data



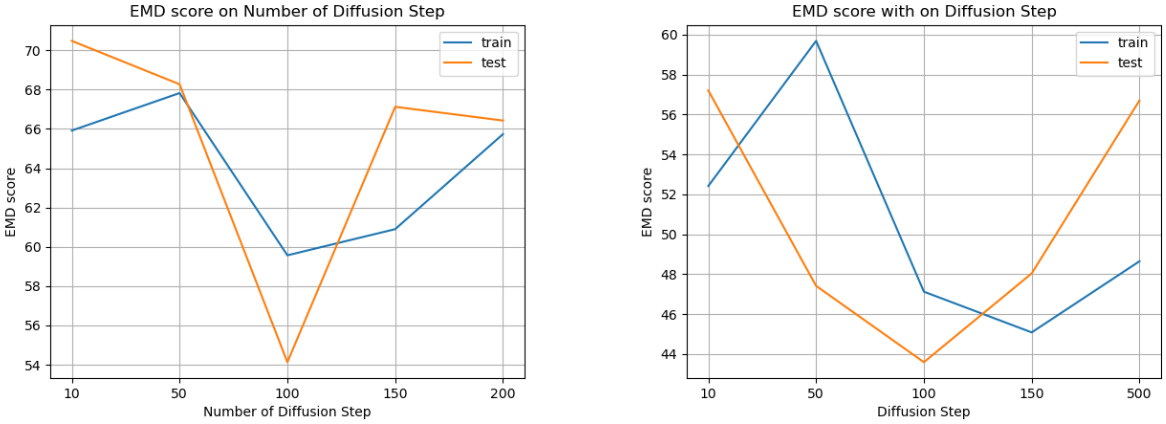
(a) 3D Sine

(b) 3D Helix

Figure 2: Comparison of number of training steps.

2.3 Diffusion Steps

We notice that number of diffusion steps are very important as more steps lead to corrupting the data even after it has reached unit variance gaussian curve. So a large number of diffusion steps for data with such low dimension is harmful so we compare different values and find values are 100 provide best generalised version of the test data. For the hyperparameters we use n_steps=varying, l_beta=2e-5 u_beta=1.28e-2 and n_epochs=100 for helix and 200 for sine dataset. We find that data is fully corrupted with the linear variance schedule after about 80 steps. We use model 4 and 6 respectively for our datasets.



(a) 3D Sine EMD Scores

(b) 3D Helix EMD Scores

Figure 3: Comparison of the number of diffusion steps.

2.4 Noise Schedule

For noise scheduling, we try linear, cosine, and sigmoid functions each with varied l_beta and u_beta. To our surprise cosine function poorly performed compared to other functions on helix dataset. For the hyperparameters, we use n_steps=100, l_beta=2e-5 u_beta=1.28e-2 and n_epochs=100 for helix and 200 for sine dataset. We use model 4 and 6 respectively for our datasets. We have added a new argument for scheduler type in the train.py and model function to accomodate handling different variance schedules. We know from diffusion models that we want to slow and gradually increase variance in forward diffusion process but linear schedules incorporate noise very quickly so we tried to find different functions which have slower convergence and provide better generalisation. We also found out that decreasing l_beta had an adverse effect which caused slower convergence and harder to predict noise for the model, increasing u_beta to 1.28e-1 had a good convergence but failed to perform well compared to other hyperparams. The below diagram shows some of the functions with slower convergence compared to other schedules.

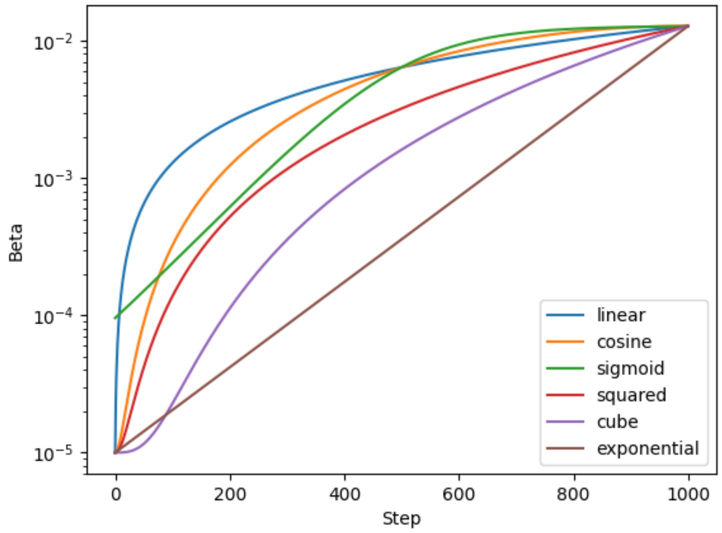
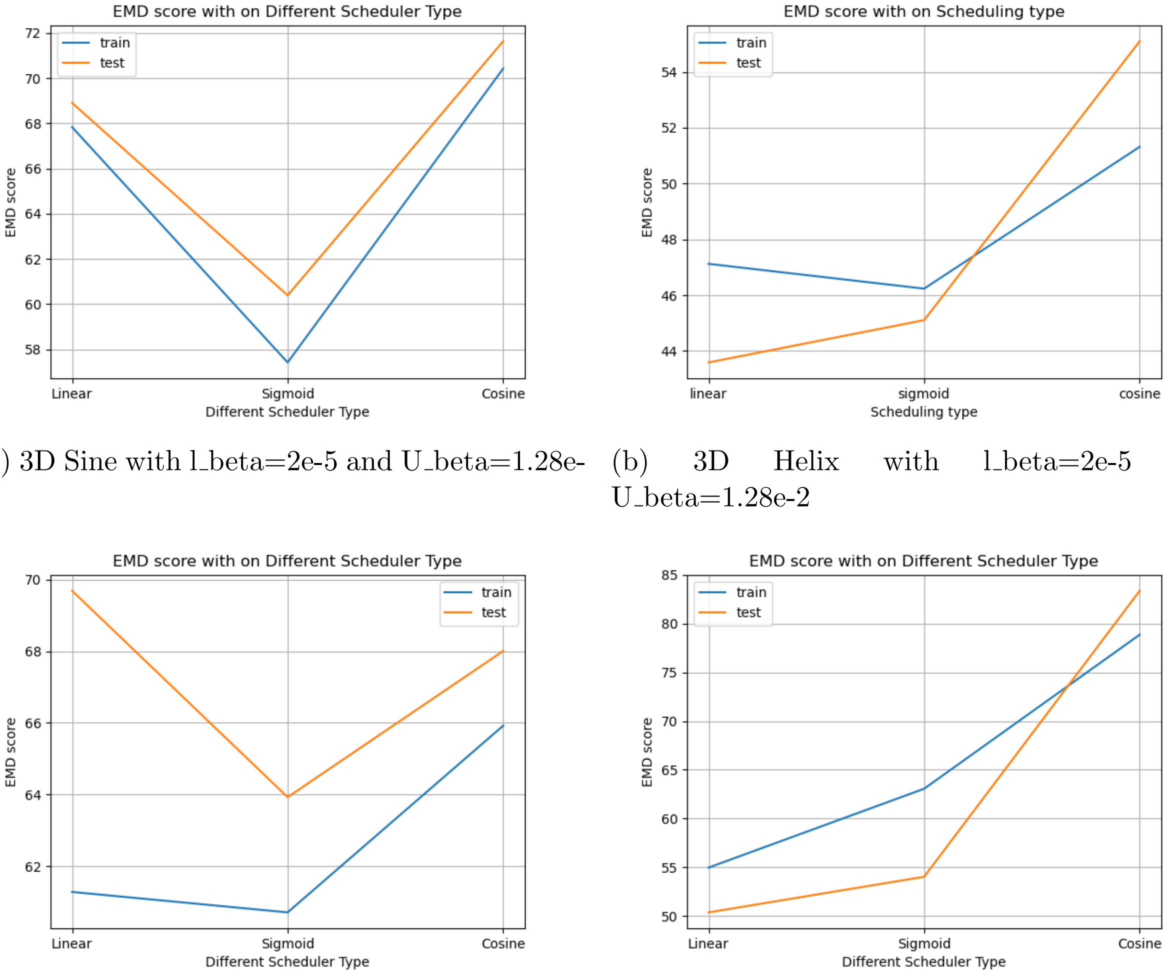


Figure 4: Different Variance Schedules with log scaled y axes



(a) 3D Sine with l_beta=2e-5 and U_beta=1.28e-2

(b) 3D Helix with l_beta=2e-5 and U_beta=1.28e-2

(c) 3D Sine with l_beta=2e-6 and U_beta=1.28e-2 n_steps=150 epochs=500

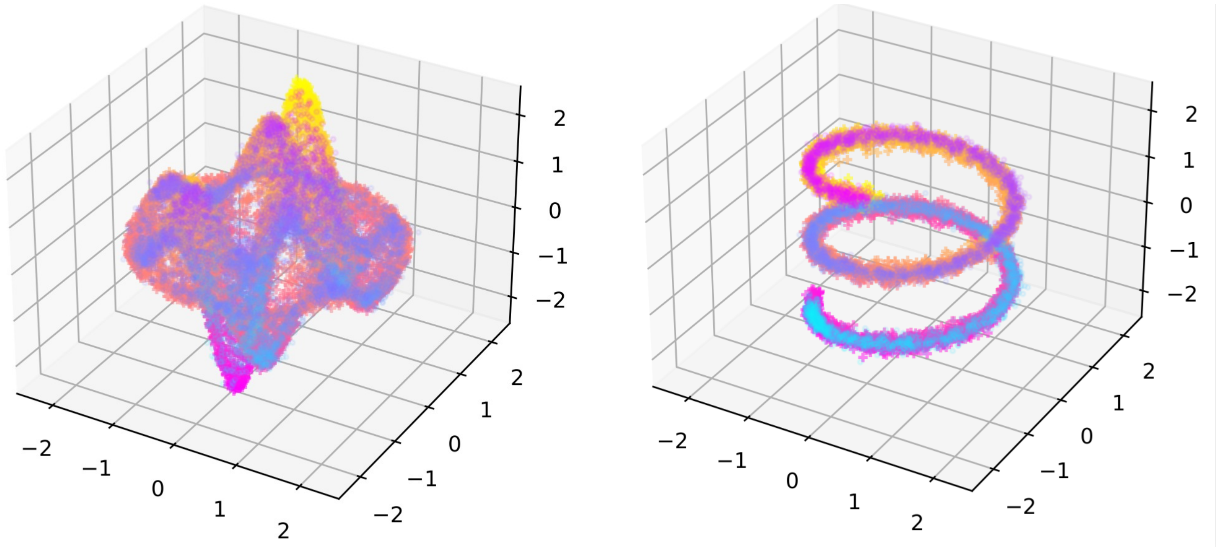
(d) 3D Helix with l_beta=2e-5 and U_beta=1.28e-1 n_steps=100 epochs=250

Figure 5: Comparison of different noising schedules for variance.

2.5 Final Hyperparameters

Model Complexity		
Hyperparam	3D Sine	3D Helix
Model Complexity	X_t,64,128,64,3	X_t,64,128,256,128,64,3
Epochs	500	100
Diffusion Steps 3	100	100
Optimizer	Adam	Adam
L_beta	2e-5	2e-5
U_beta	1.28e-2	1.28e-2
Noising Function	Sigmoid	Linear

Table 2: List of hyperparameters and their performance



(a) 3D Sine DDPM

(b) 3D Helix DDPM

Figure 6: Plots for comparing learned and original distributions.

Test Data Results			
Data	EMD	NLL	Chamfer
3D Sine	57.90	2.60	20.56
3D Helix	49.85	2.25	13.77

Table 3: Metrics Comparison

References

[1] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[3] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. "Normalizing flows: An introduction and review of current methods". In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 3964–3979.

[4] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.