EE660 HW5: Generation via Unsupervised Learning

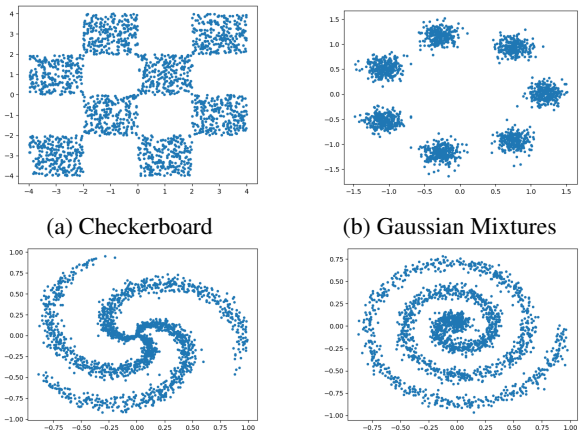
Shauryasikt Jena* **Collaborator: Eric Killian**
University of Southern California
{jenas}@usc.edu

1 Introduction

Unsupervised learning, a branch of Machine Learning that operates without labelled data, is used to generate datasets from noise. The models under study learn the statistical attributes of the input dataset distributions. Then these trained models are able to generate new datapoints, that follow the dataset distribution, from noise. I investigate the generative capabilities of these models, qualitatively and quantitatively in this study.
Extra Credit: Covering 4 models for 4 datasets $\implies$ 10+10+25+25=70 points.

2 Datasets

First, I will discuss the datasets and the quantitative performance of all the four models on each dataset. Then I will elaborate more on model specific implementations. Quantitative comparisons are made by evaluating the Gaussian Kernel Density Estimates (KDE) of the generated samples with respect to the original datasets. Less negative KDEs imply better generations.



(a) Checkerboard (b) Gaussian Mixtures

(c) Pinwheel (d) Spiral

Figure 1: Datasets

The original datasets are visualized in Figure 1. The datasets have 2000 samples each, out of which 500 are randomly selected for the hold-out set for validation. The training and validation loss plots are displayed in Table 2. The similar patterns of training and validation losses in all plots suggest that there is no overfitting and that the training processes were stable.

Dataset	DSM	DDPM	GAN	EBM
Checkerboard	-4.03	-4.04	**-3.94**	-10.58
Gaussian Mix	-1.57	-1.39	**-1.34**	-143
Pinwheel	-1.02	-1.24	**-0.61**	-751
Spiral	-0.92	-1.07	**-0.96**	-888

Table 1: Mean KDEs of generations

Table 1 shows the mean Gaussian KDE values of the generations. The trends show that GANs generate the samples most similar to the original distributions, followed by DSM and DDPM with good performances and EBM performing abysmally.

However, quantitative evaluations are not enough to determine the quality of the generations and the shortcomings of the models.

3 Models

Table 3 shows the generated samples for all models across all the datasets. Qualitatively speaking, EBM does not seem to converge to the original datasets. While GANs do converge to original datasets, they do not seem to be able to capture a higher fraction of the variance. ***DSM and DDPM fare the best compared to other models in generating samples that capture the most variance of the original datasets, as well as their distributions.***

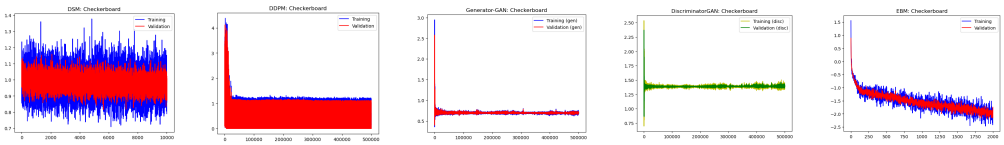
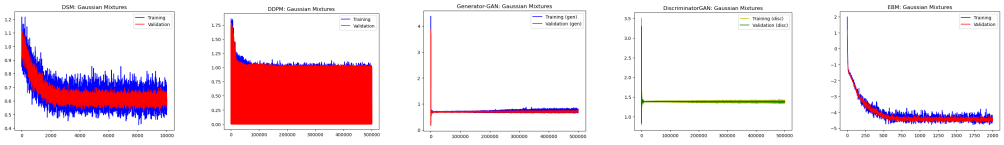
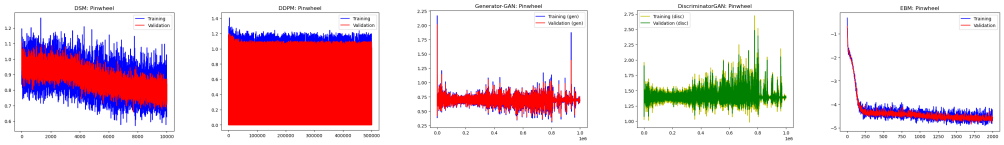
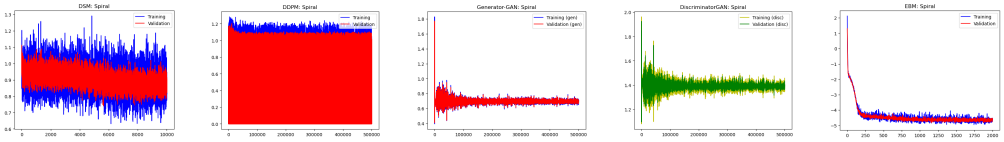
Dataset	DSM	DDPM	GAN(**generator**) GAN(**discriminator**)	EBM
Checkerboard				
Gaussian Mix				
Pinwheel				
Spiral				

Table 2: Training and Validation Losses

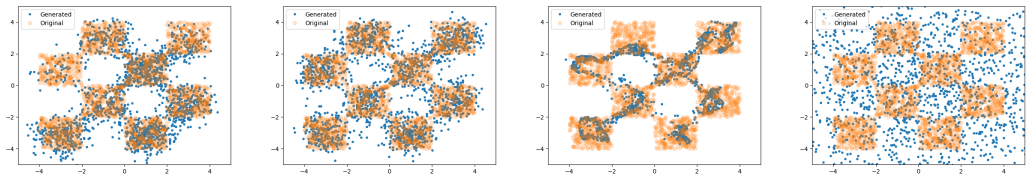
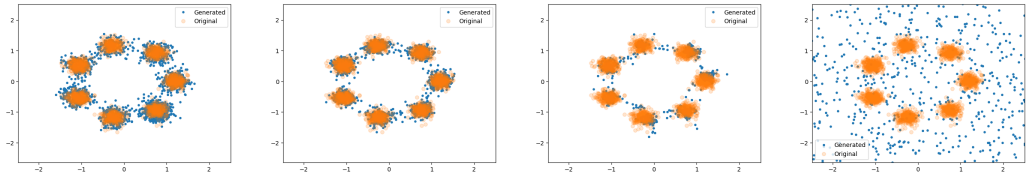
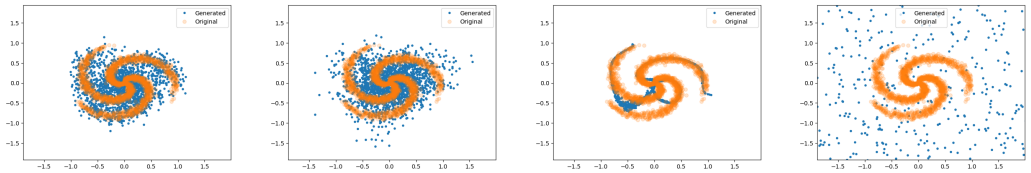
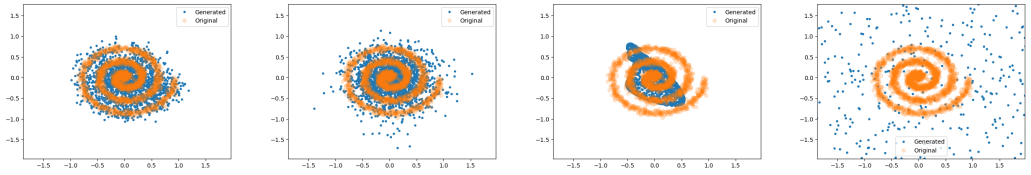
Dataset	DSM	DDPM	GAN	EBM
Checkerboard				
Gaussian Mix				
Pinwheel				
Spiral				

Table 3: Generated samples

3.1 Energy Based Model (EBM)

EBM requires to implement Langevin sampling in its loss function, which is essentially an L1 norm of the difference between the energies of the original and generated datasets. The minimization of loss causes the energy of the original distribution to be lower and other noisy points to be higher, while Langevin sampling is supposed to move noisy points to the original distribution via several small gradient steps to the minima of energy. Even though the energy distribution captured is good, the sampling fails to generate good points. Even after a thorough hyperparameter tuning process, I was not able to stabilize the learning of EBM, leading to the inference that EBM is not the most convenient model for generation.

Dataset	LR	L2	Epochs	Batch
Checkerboard	10^{-3}	0.1	2000	128
Gaussian Mix	10^{-3}	0.1	2000	128
Pinwheel	10^{-3}	0.1	2000	128
Spiral	10^{-3}	0.1	2000	128

Table 4: EBM Hyperparameters

Here, LR is learning rate, L2 is the coefficient for l2-regularization term, Batch is the batch size. The architecture for EBM for all models is a 2-layer MLP with [64,64] nodes that outputs 1-dimensional energy. The step-size for Langevin Sampling is 5×10^{-3} and the number of steps is 1000.

3.2 Denoising Score Matching (DSM)

DSM is modelled to minimize the L2 loss between the samples of the original distribution and those samples perturbed by noise. The perturbation caused the training process to be noisy, however it was stable for all datasets and DSM converged to parameters for an excellent Langevin Sampling.

Dataset	LR	σ	Epochs	Batch
Checkerboard	10^{-3}	0.1	10000	128
Gaussian Mix	10^{-3}	0.1	10000	128
Pinwheel	10^{-3}	0.1	10000	128
Spiral	10^{-3}	0.1	10000	128

Table 5: DSM Hyperparameters

Here, LR is learning rate, σ is the coefficient for noise perturbation, Batch is the batch size. The architecture for DSM for all models is a 2-layer MLP with [128,128] nodes that outputs 2-dimensional points in consistency with the datasets. The step-size for Langevin Sampling is 5×10^{-3} and the number of steps is 2000.

3.3 Denoising Diffusion Probabilistic Model (DDPM)

DDPM is modelled to minimize the L2 loss between the noise and the original samples that undergo a diffusion process over consecutive timesteps to resemble noise. The model, having learned how to convert datasets into noise over timesteps, reverses the process while sampling from noise by retracing the timesteps to generate points consistent with the original distribution.

For sampling timesteps, I followed the suggestions in the lecture notes to set $\alpha_1 = 1 - 10^{-4}$, $\alpha_T = 0.98$.

Dataset	LR	T	Epochs	Batch
Checkerboard	10^{-4}	1000	500000	128
Gaussian Mix	10^{-4}	1000	500000	128
Pinwheel	10^{-4}	1000	500000	128
Spiral	10^{-4}	1000	500000	128

Table 6: DDPM Hyperparameters

I set $T = 1000$ as suggested, and ran for 500×1000 epochs so that roughly each timestep gets sampled ≈ 500 times, enough for the base MLP to learn over all the timesteps. The architecture for DDPM for all models is a 2-layer MLP with [128,128] nodes that outputs 2-dimensional points in consistency with the datasets.

3.4 Generative Adversarial Networks (GANs)

GAN has the two standard components — generator and discriminator. The generator loss function is the log-loss between the probability distributions of the original dataset and the samples that it generates. The discriminator acts as a check for the generator so that when the original and generated probability distributions are fit, the probability of a selected sample from being the generations is 0.5. A small $\epsilon = 10^{-8}$ is added to all probabilities in order to avoid high-magnitude log-losses that might cause exploding gradients.

Dataset	LR	Epochs	Batch
Checkerboard	10^{-4}	500000	64
Gaussian Mix	10^{-4}	500000	64
Pinwheel	10^{-4}	1000000	64
Spiral	10^{-4}	500000	64

Table 7: GAN Hyperparameters

After an extensive hyperparameter tuning, I found that GANs require a lot of epochs to gain stability. However, I could not realize the potential of GANs to act as the best generators. To that end, adding batch-normalization layers to the generator and discriminator MLPs would have helped to capture more variance but my system compatibility with *jax* prevented me from such. The architecture for GAN generators and discriminators for all models is a 2-layer MLP with [256,256] nodes that outputs 2-dimensional points in consistency with the datasets for the generator and 1-dimensional output (probability) for the discriminator.

Appendix A

The KDE distributions for the generations are shown in Table 8. With the exception of EBM as already discussed, most of the models generate up to 75% of their samples very close to the mean KDE implying good quality and few outliers.

Appendix B

The energy landscapes of the generative models are displayed in Table 9. GAN energies are hard to interpret, however all other models show their clear adaptation to the energies of the original dataset distributions.

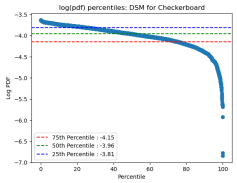
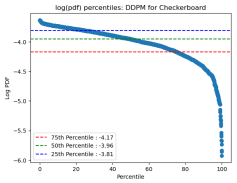
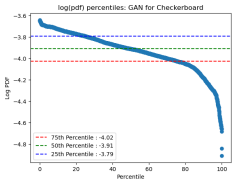
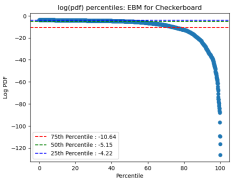
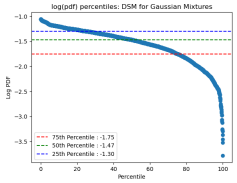
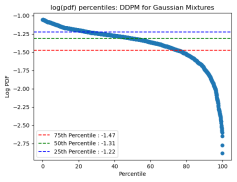
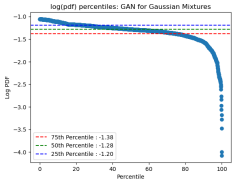
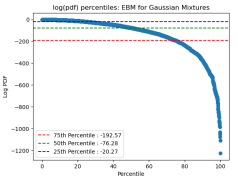
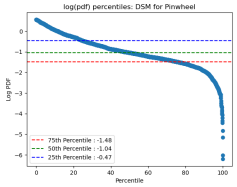
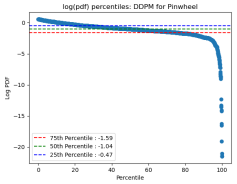
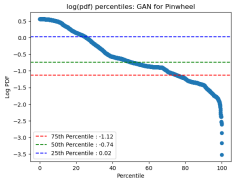
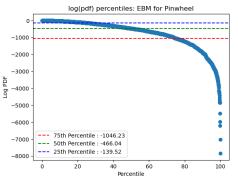
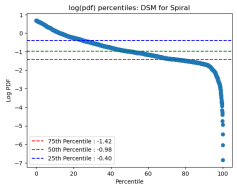
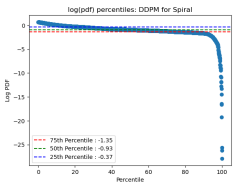
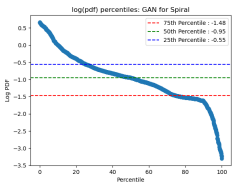
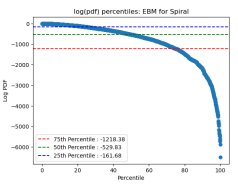
Dataset	DSM	DDPM	GAN	EBM
Checkerboard				
Gaussian Mix				
Pinwheel				
Spiral				

Table 8: KDE distributions of generations

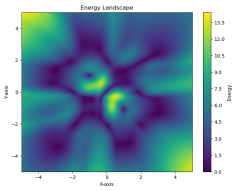
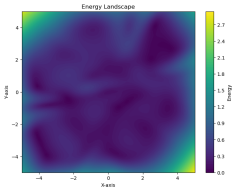
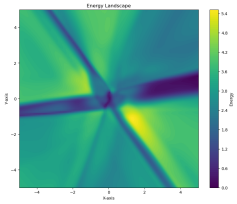
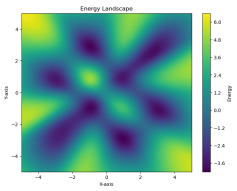
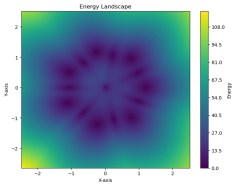
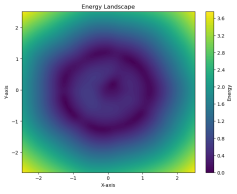
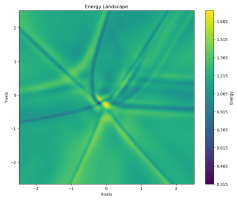
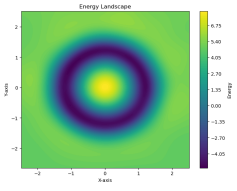
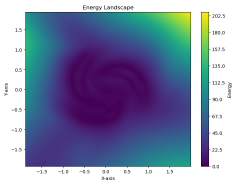
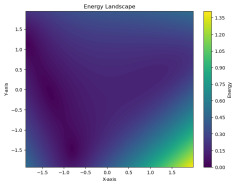
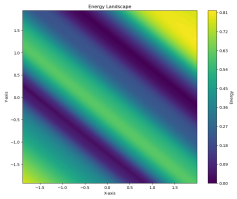
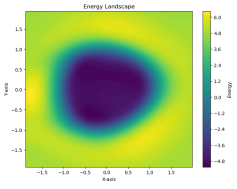
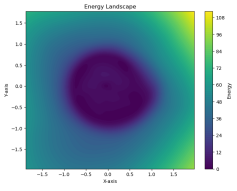
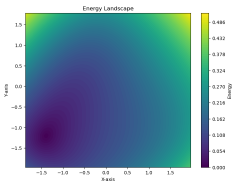
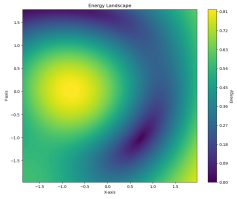
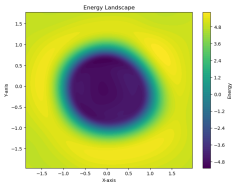
Dataset	DSM	DDPM	GAN	EBM
Checkerboard				
Gaussian Mix				
Pinwheel				
Spiral				

Table 9: Energy landscapes of models