Generative Modeling on 2D Synthetic Datasets

Arshdeep Dhillon

6 Dec 2025

1 Introduction

A comparative analysis was conducted between a Variational Autoencoder (VAE) and a Generative Adversarial Network (GAN) on 2-D synthetic datasets commonly used to evaluate performance on structured, multimodal, and geometrically complex distributions. Residual blocks were incorporated into both models to increase representational capacity. The architectures employed in the experiments are shown in Figure 1.

The models were evaluated on four datasets—Pinwheel, Spiral, Checkerboard, and Gaussian Mixtures—using both qualitative and quantitative metrics. Dataset-specific training parameters are presented in the following sections, and results include loss curves, scatter plots of generated vs. real samples, and KDE-based density visualizations for assessing distributional fidelity.

All code, configuration files, and generated outputs (plots, density maps, logs) are available in the project repository: `https://github.com/arshdeep-usc/ee660-project`

Qualitatively, VAE samples tend to form smooth, continuous approximations of the underlying manifolds, while GAN samples produce sharper, more detailed structures but occasionally exhibit mode imbalance (notably in the Gaussian Mixtures dataset). Quantitatively, KDE log-likelihood scores (Table 1) indicate that VAEs perform better on smoother datasets (Pinwheel and Gaussian Mixtures), whereas GANs perform better on datasets characterized by sharper transitions (Spiral and Checkerboard).

2 Training

Both models were trained using PyTorch with dataset-specific hyperparameters described later. Training used the Adam optimizer with a learning rate of 10^{-3}. For each dataset, 500 samples were held out for loss evaluation only.

For the GAN, the generator sampled latent vectors from a standard normal distribution and was updated using Binary Cross-Entropy (BCE) loss, with target labels of 1 to encourage generated outputs to be classified as real. The discriminator was trained on real and generated samples using BCE losses with targets 1 and 0, respectively. Generator, discriminator, and hold-out losses were logged separately.

For the VAE, each input was encoded into posterior parameters μ and $\log \sigma^2$, and latent vectors were sampled via the reparameterization trick. The loss consisted of a mean-squared-error reconstruction term and a KL divergence term weighted by $\beta = 0.2$. Reconstruction, KL, total loss, and hold-out loss were recorded for all epochs.

Training and hold-out losses were computed at every epoch. Convergence was monitored using loss curves, sample visualizations, and KDE heatmaps. All experiments used fixed random seeds, and 2000 samples were generated for evaluation and visualization after training.

2.1 Overtraining Considerations

No significant overtraining was observed. Hold-out losses closely matched training losses for both models, likely due to the simplicity of the 2-D distributions, which lack fine-grained irregularities or noise patterns that typically encourage overfitting. The relatively compact network architectures also limited the risk of memorization. Although GANs can experience mode collapse or discriminator overpowering, these issues manifested primarily in sample distributions rather than loss divergence. Overall, training remained stable with minimal indications of overfitting.

3 Qualitative Comparison

3.1 Pinwheel Dataset

GAN samples capture spiral arms with high fidelity and preserve local structures, whereas VAE samples are smoother and exhibit some blurring of boundaries. The GAN samples fail to capture the shape of one of the arms of the spiral but offer a tighter grouping. The VAE samples in comparison are more accurate in capturing the overall shape but add noisy samples between each arm (Figure 4).

3.2 Spiral Dataset

VAE samples closely follow the continuous spiral curves of the true data manifold, whereas GAN samples occasionally fail to cover portions of the spiral, resulting in uneven representation. The GAN samples fail to capture the tail end of the spiral but offer a tighter grouping. The VAE samples in comparison are more accurate in capturing the overall shape but add noisy samples all throughout (Figure 7).

3.3 Checkerboard Dataset

GAN samples clearly delineate high- and low-density regions, effectively reproducing sharp edges between squares, whereas VAE samples exhibit smoother transitions and slightly blurred boundaries. The GAN samples are still not able to completely recreate the sharp edges but they are far less noisy than the VAE samples. (Figure 10).

3.4 Gaussian Mixtures Dataset

VAE samples form smooth clusters that cover all modes, whereas GAN samples tend to over represent high-density peaks. The VAE samples have more noise between nodes where as the GAN samples are not as noisy (Figure 13).

4 Quantitative Comparison

KDE log-likelihood results are presented in Table 1. VAEs demonstrate superior performance on Pinwheel & Gaussian Mixtures datasets, whereas GANs more effectively capture the Spiral & Checkerboard datasets.

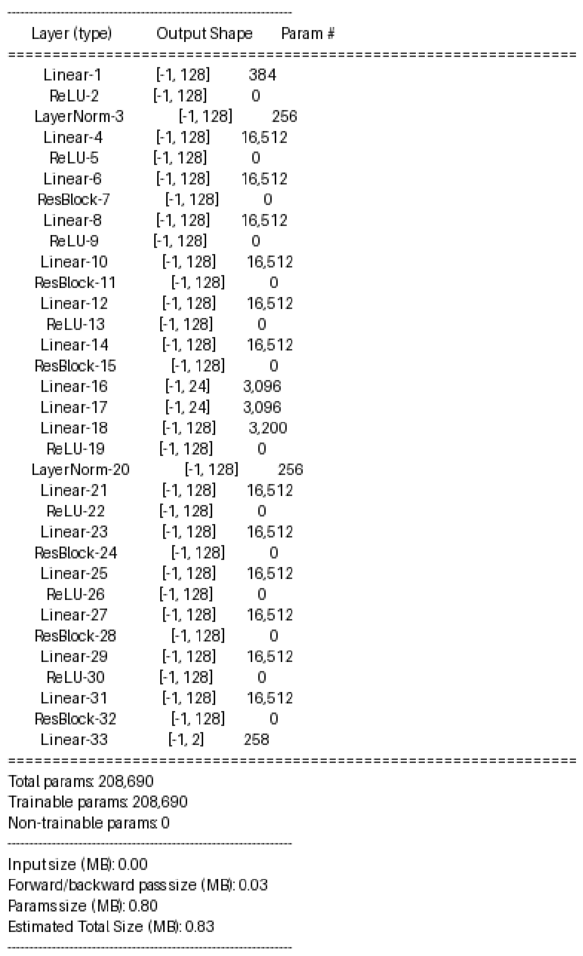
Table 1: KDE log-likelihood (higher is better) for VAE and GAN samples on each dataset.

Dataset	VAE	GAN
Pinwheel	-0.6934	-0.7832
Spiral	-0.7580	-0.7381
Checkerboard	-4.1033	-4.0468
Gaussian Mixtures	-1.4580	-1.4987

5 Conclusion

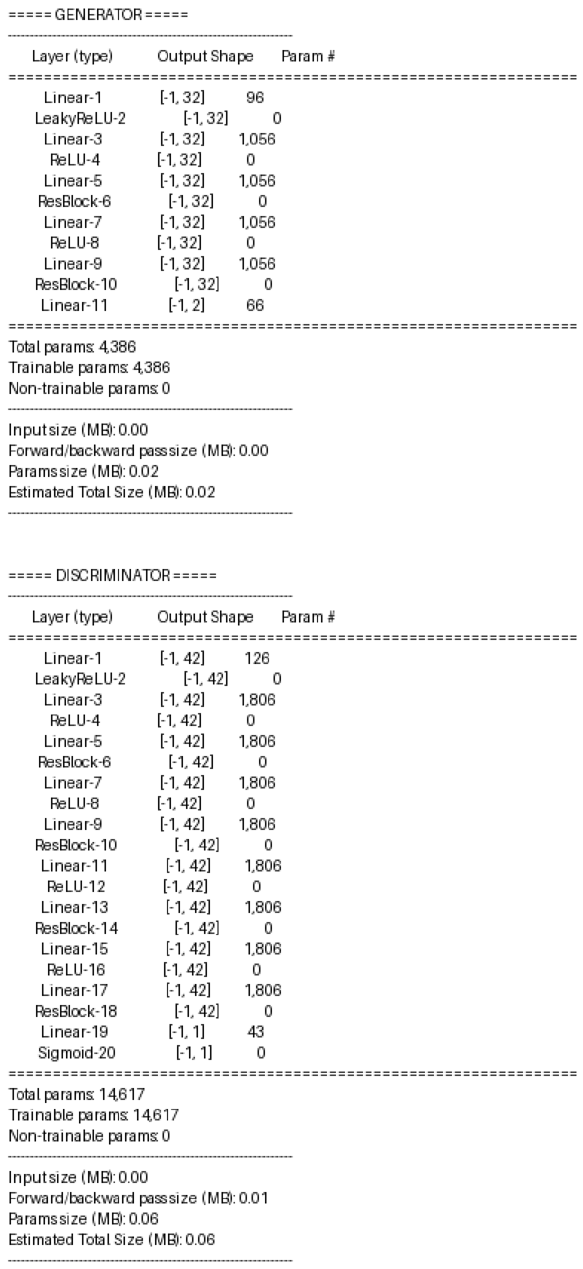
VAEs and GANs were evaluated across four 2-D synthetic datasets and showed complementary strengths. VAEs provided smoother, more complete coverage of the data manifolds, while GANs produced sharper samples and captured datasets with discrete structure more effectively. Both models trained stably, with hold-out losses closely matching training losses.

6 Model Architectures



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                 [-1, 128]             384
              ReLU-2                 [-1, 128]               0
         LayerNorm-3                 [-1, 128]             256
            Linear-4                 [-1, 128]          16,512
              ReLU-5                 [-1, 128]               0
            Linear-6                 [-1, 128]          16,512
          ResBlock-7                 [-1, 128]               0
            Linear-8                 [-1, 128]          16,512
              ReLU-9                 [-1, 128]               0
           Linear-10                 [-1, 128]          16,512
         ResBlock-11                 [-1, 128]               0
           Linear-12                 [-1, 128]          16,512
             ReLU-13                 [-1, 128]               0
           Linear-14                 [-1, 128]          16,512
         ResBlock-15                 [-1, 128]               0
           Linear-16                  [-1, 24]           3,096
           Linear-17                  [-1, 24]           3,096
           Linear-18                 [-1, 128]           3,200
             ReLU-19                 [-1, 128]               0
        LayerNorm-20                 [-1, 128]             256
           Linear-21                 [-1, 128]          16,512
             ReLU-22                 [-1, 128]               0
           Linear-23                 [-1, 128]          16,512
         ResBlock-24                 [-1, 128]               0
           Linear-25                 [-1, 128]          16,512
             ReLU-26                 [-1, 128]               0
           Linear-27                 [-1, 128]          16,512
         ResBlock-28                 [-1, 128]               0
           Linear-29                 [-1, 128]          16,512
             ReLU-30                 [-1, 128]               0
           Linear-31                 [-1, 128]          16,512
         ResBlock-32                 [-1, 128]               0
           Linear-33                   [-1, 2]             258
================================================================
Total params: 208,690
Trainable params: 208,690
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.03
Params size (MB): 0.80
Estimated Total Size (MB): 0.83
----------------------------------------------------------------
```

(a) VAE Architecture



```
===== GENERATOR =====
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                  [-1, 32]              96
         LeakyReLU-2                  [-1, 32]               0
            Linear-3                  [-1, 32]           1,056
              ReLU-4                  [-1, 32]               0
            Linear-5                  [-1, 32]           1,056
          ResBlock-6                  [-1, 32]               0
            Linear-7                  [-1, 32]           1,056
              ReLU-8                  [-1, 32]               0
            Linear-9                  [-1, 32]           1,056
         ResBlock-10                  [-1, 32]               0
           Linear-11                   [-1, 2]              66
================================================================
Total params: 4,386
Trainable params: 4,386
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.02
Estimated Total Size (MB): 0.02
----------------------------------------------------------------


===== DISCRIMINATOR =====
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                  [-1, 42]             126
         LeakyReLU-2                  [-1, 42]               0
            Linear-3                  [-1, 42]           1,806
              ReLU-4                  [-1, 42]               0
            Linear-5                  [-1, 42]           1,806
          ResBlock-6                  [-1, 42]               0
            Linear-7                  [-1, 42]           1,806
              ReLU-8                  [-1, 42]               0
            Linear-9                  [-1, 42]           1,806
         ResBlock-10                  [-1, 42]               0
           Linear-11                  [-1, 42]           1,806
             ReLU-12                  [-1, 42]               0
           Linear-13                  [-1, 42]           1,806
         ResBlock-14                  [-1, 42]               0
           Linear-15                  [-1, 42]           1,806
             ReLU-16                  [-1, 42]               0
           Linear-17                  [-1, 42]           1,806
         ResBlock-18                  [-1, 42]               0
           Linear-19                   [-1, 1]              43
          Sigmoid-20                   [-1, 1]               0
================================================================
Total params: 14,617
Trainable params: 14,617
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.01
Params size (MB): 0.06
Estimated Total Size (MB): 0.06
----------------------------------------------------------------
```

(b) GAN Architecture

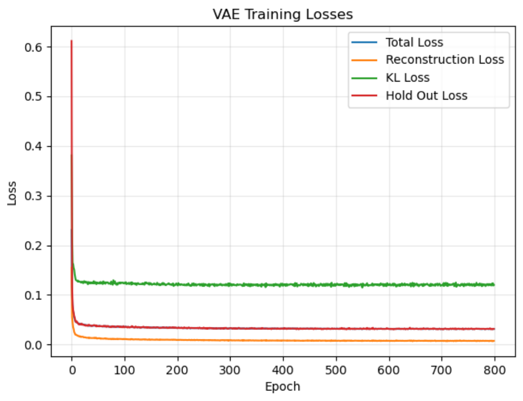
Figure 1: Summary of VAE and GAN architectures.

7 Pinwheel Dataset

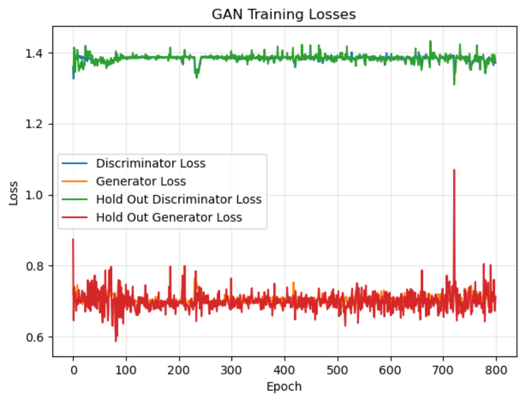
7.1 Training Setup

VAE: hidden dim $h = 128$, latent dim $z = 24$, residual blocks $n_{res} = 3$, $\beta = 0.2$, learning rate 10^{-3}, batch size 128, epochs 800.
GAN: Generator hidden dim $h_G = 32$, Discriminator hidden dim $h_D = 42$, Generator residual blocks $n^G_{res} = 2$, Discriminator residual blocks $n^D_{res} = 4$, learning rate 10^{-3}, betas $(0.5, 0.9)$, batch size 128, epochs 800.

7.2 Results

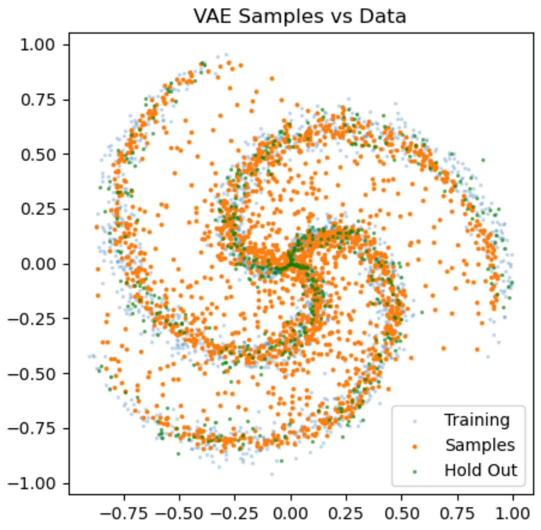


(a) VAE Loss

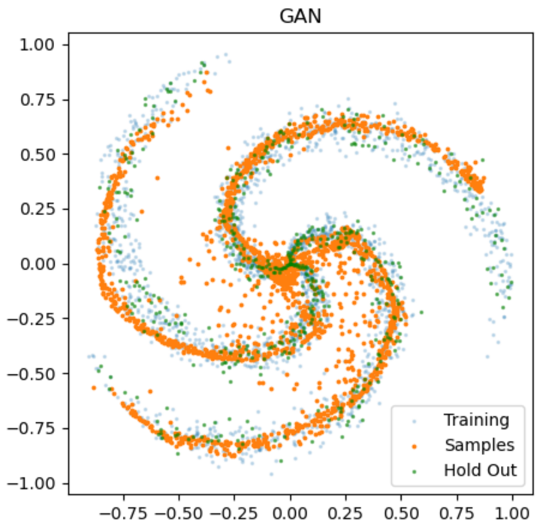


(b) GAN Loss

Figure 2: Training loss curves for VAE and GAN on the Pinwheel dataset.

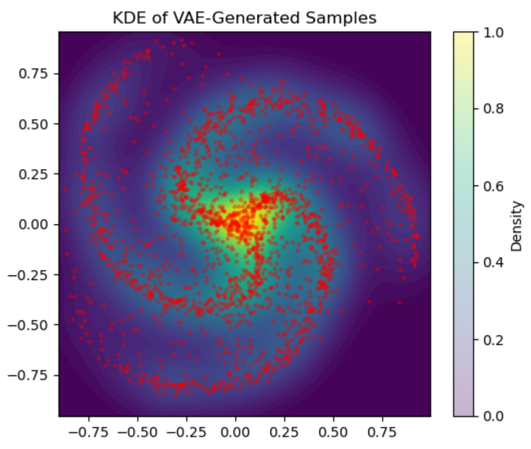


(a) VAE Samples

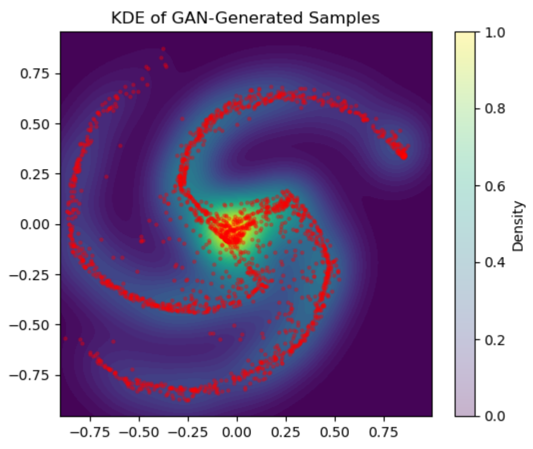


(b) GAN Samples

Figure 3: Generated samples on the Pinwheel dataset.



(a) VAE KDE



(b) GAN KDE

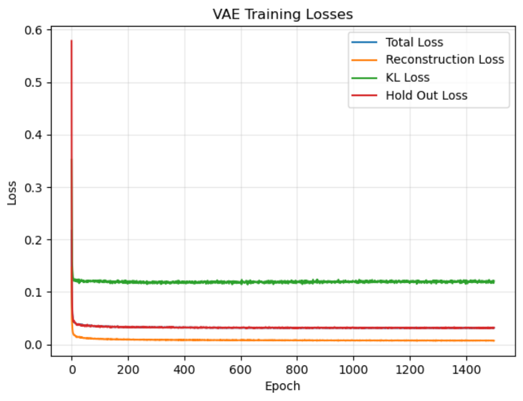
Figure 4: KDE visualizations of generated samples on the Pinwheel dataset.

8 Spiral Dataset

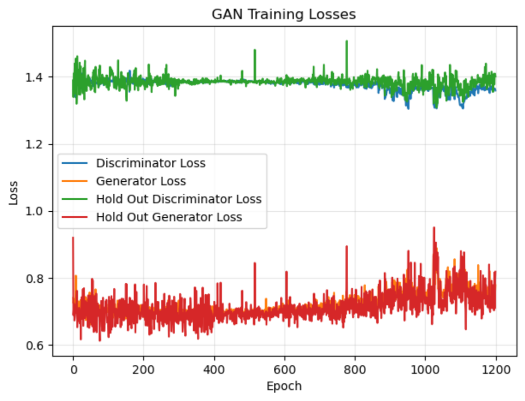
8.1 Training Setup

VAE: hidden dim $h = 128$, latent dim $z = 24$, residual blocks $n_{res} = 3$, $\beta = 0.2$, learning rate 10^{-3}, batch size 128, epochs 1500.
GAN: Generator hidden dim $h_G = 32$, Discriminator hidden dim $h_D = 42$, Generator residual blocks $n^G_{res} = 2$, Discriminator residual blocks $n^D_{res} = 4$, learning rate 10^{-3}, betas $(0.5, 0.9)$, batch size 128, epochs 1200.

8.2 Results

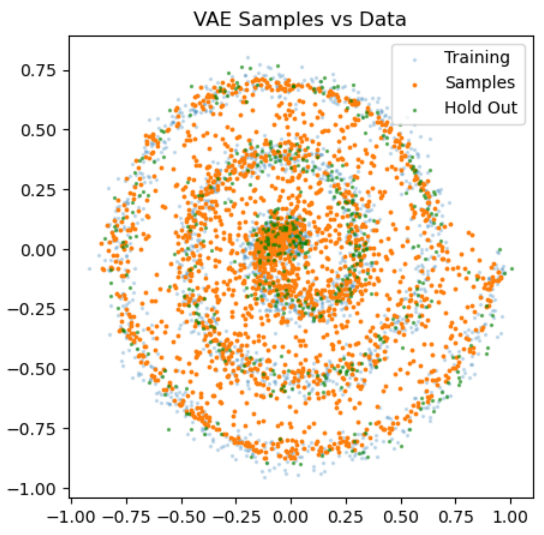


(a) VAE Loss

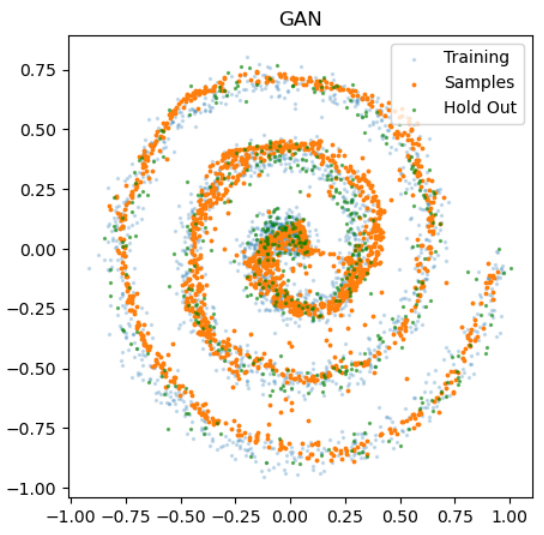


(b) GAN Loss

Figure 5: Training loss curves for VAE and GAN on the Spiral dataset.

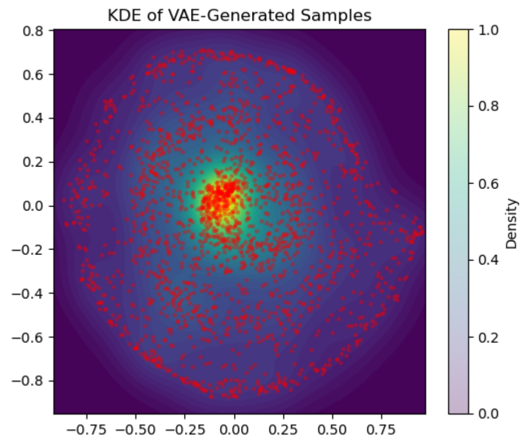


(a) VAE Samples

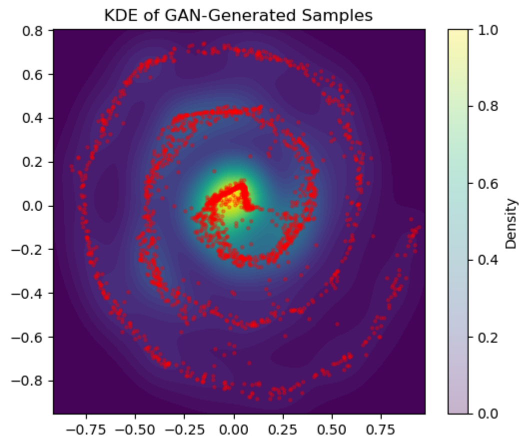


(b) GAN Samples

Figure 6: Generated samples on the Spiral dataset.



(a) VAE KDE



(b) GAN KDE

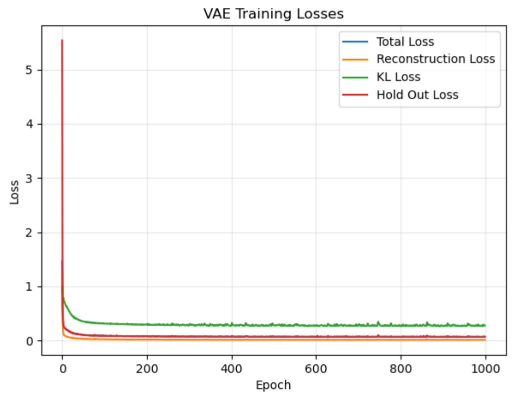
Figure 7: KDE visualizations of generated samples on the Spiral dataset.

9 Checkerboard Dataset

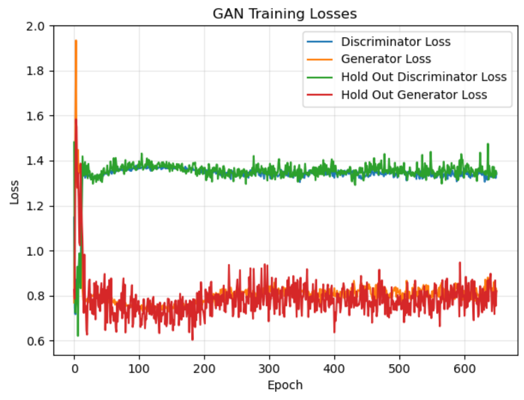
9.1 Training Setup

VAE: hidden dim $h = 128$, latent dim $z = 24$, residual blocks $n_{res} = 3$, $\beta = 0.2$, learning rate 10^{-3}, batch size 128, epochs 1000.
GAN: Generator hidden dim $h_G = 32$, Discriminator hidden dim $h_D = 42$, Generator residual blocks $n_{res}^G = 2$, Discriminator residual blocks $n_{res}^D = 4$, learning rate 10^{-3}, betas $(0.5, 0.9)$, batch size 128, epochs 650.

9.2 Results

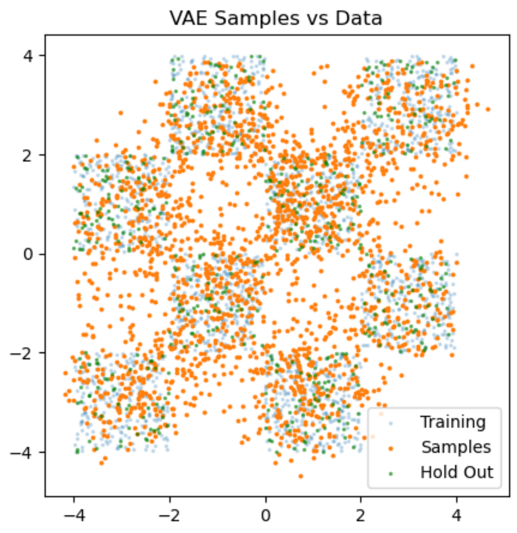


(a) VAE Loss

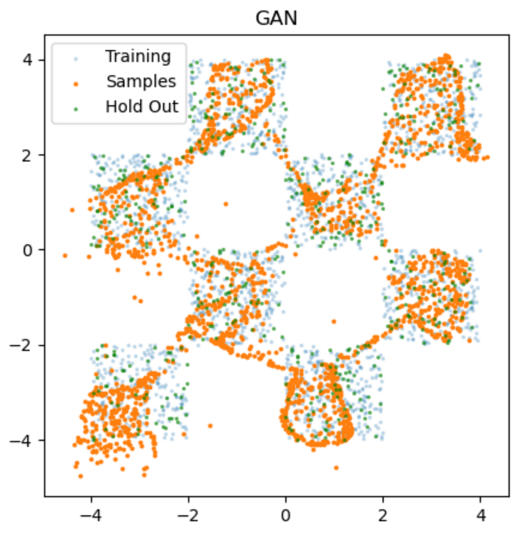


(b) GAN Loss

Figure 8: Training loss curves for VAE and GAN on the Checkerboard dataset.

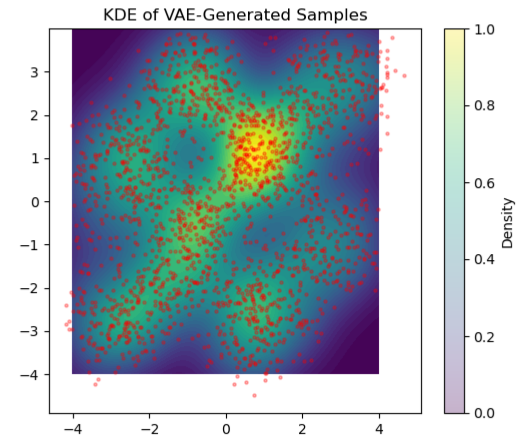


(a) VAE Samples

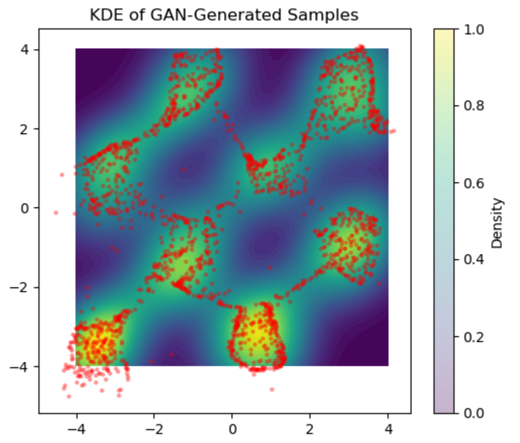


(b) GAN Samples

Figure 9: Generated samples on the Checkerboard dataset.



(a) VAE KDE



(b) GAN KDE

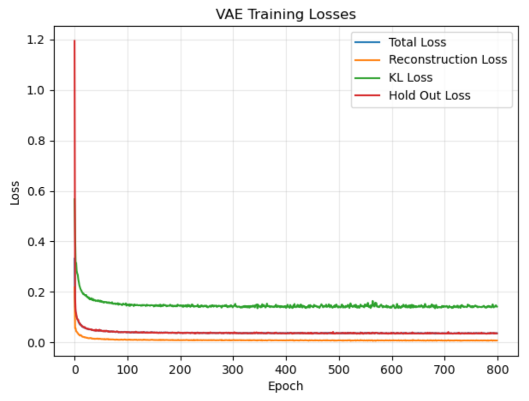
Figure 10: KDE visualizations of generated samples on the Checkerboard dataset.

10 Gaussian Mixtures Dataset

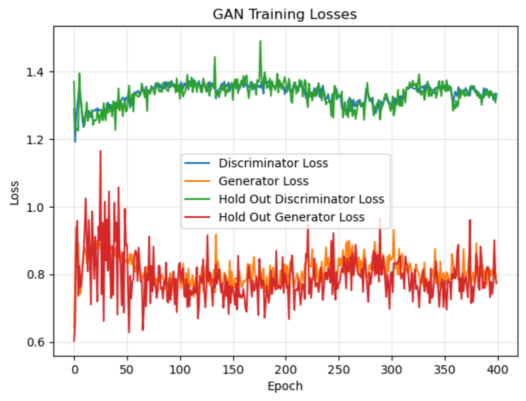
10.1 Training Setup

VAE: hidden dim $h = 128$, latent dim $z = 24$, residual blocks $n_{res} = 3$, $\beta = 0.2$, learning rate 10^{-3}, batch size 128, epochs 800.
GAN: Generator hidden dim $h_G = 32$, Discriminator hidden dim $h_D = 42$, Generator residual blocks $n_{res}^G = 2$, Discriminator residual blocks $n_{res}^D = 4$, learning rate 10^{-3}, betas $(0.5, 0.9)$, batch size 128, epochs 400.

10.2 Results

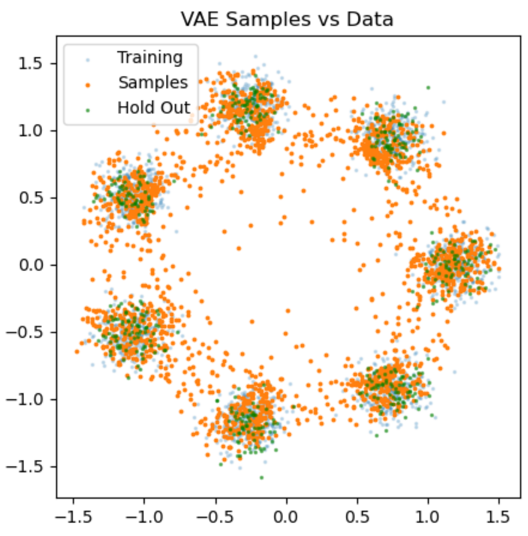


(a) VAE Loss

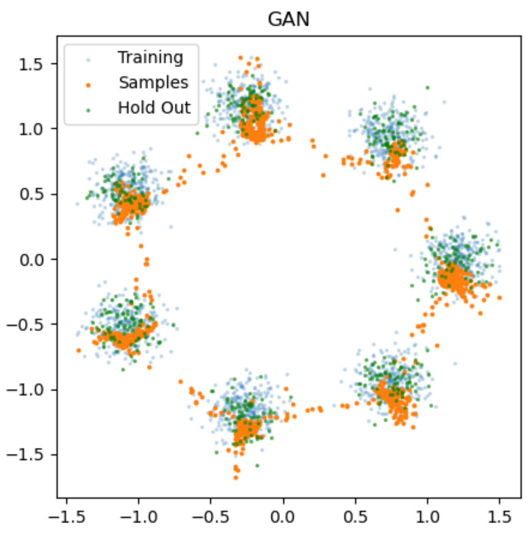


(b) GAN Loss

Figure 11: Training loss curves for VAE and GAN on the Gaussian Mixtures dataset.

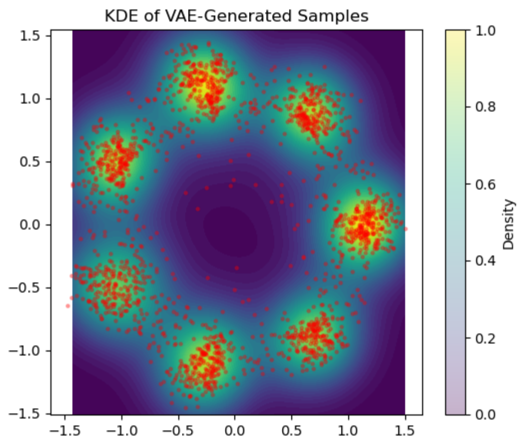


(a) VAE Samples

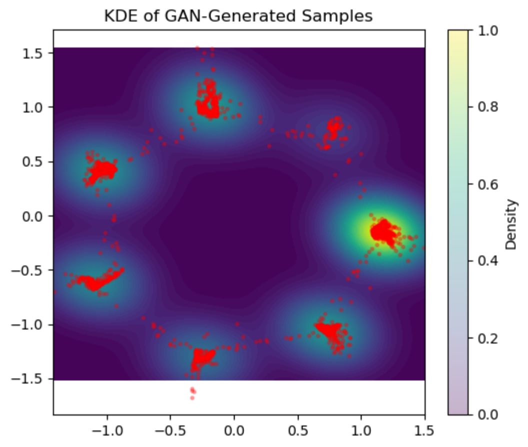


(b) GAN Samples

Figure 12: Generated samples on the Gaussian Mixtures dataset.



(a) VAE KDE



(b) GAN KDE

Figure 13: KDE visualizations of generated samples on the Gaussian Mixtures dataset.