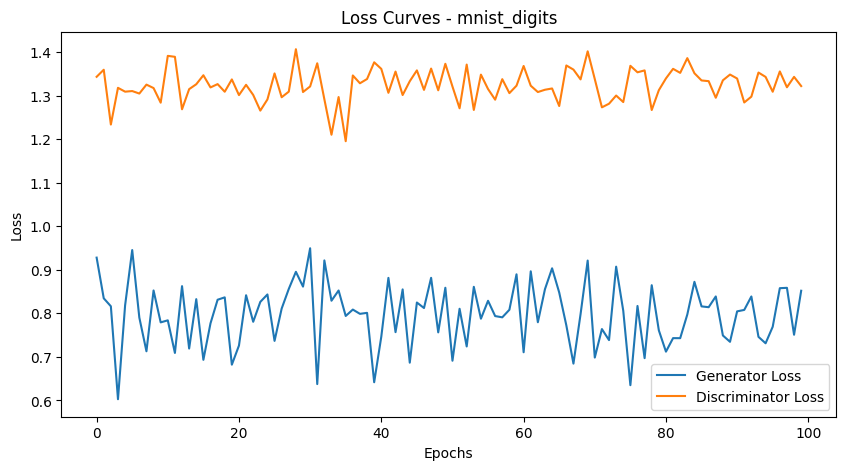
Generative Ai Asm #1:

**GANS Mnist-digits:**



**10 random created images after training**



**Specific Digit 3:**

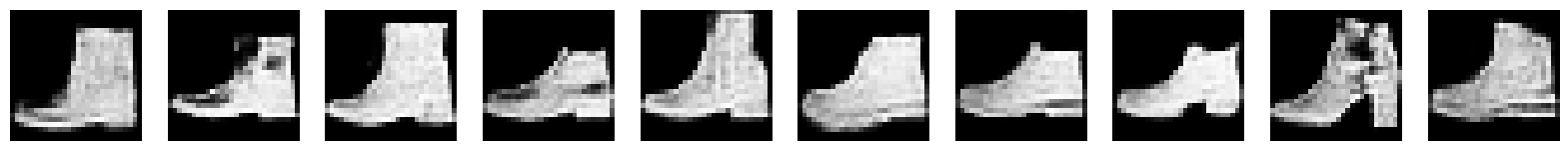
A white letter on a black background

AI-generated content may be incorrect.

**GANS Mnist\_fashion**

A graph of a loss

AI-generated content may be incorrect.  
  
**Generated images of shoes:**



**VAE’s on Mnist -Digit:**

**Latent space:**

A colorful dots with numbers and a white background

AI-generated content may be incorrect.

**Generated new images:**

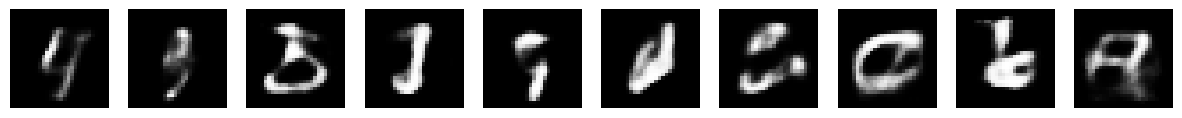


**Specific image generation:**

A white cane on a black background

AI-generated content may be incorrect.

**10 new image generation:**

  
  
**Loss curve of Vae’s:**

A graph with orange and blue lines

AI-generated content may be incorrect.  
  
**Specific image generation i.e 2:**

A white logo on a black background

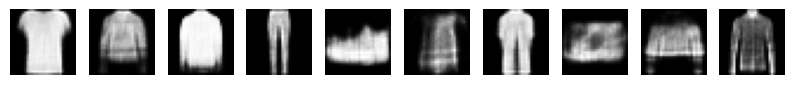
AI-generated content may be incorrect.

**VAE’s for Mnist fashion:**

A colorful dots on a white background

AI-generated content may be incorrect.

**Generated Images after training**

  
  
**Specific image generation Like shirt**

A white t-shirt with a black background

AI-generated content may be incorrect.

**Comparison: GAN vs. VAE:**

|  |  |  |
| --- | --- | --- |
| Criteria | GAN (Generative Adversarial Network) | VAE (Variational Autoencoder) |
| a. Image Quality | ✅ **GANs generate sharper, more realistic images** by learning a direct mapping from noise to data through adversarial training. However, they might produce artifacts or mode collapse (lack of diversity). | ❌ **VAEs generate blurrier images** because they maximize likelihood using a probabilistic approach, leading to smoother outputs but sometimes losing finer details. |
| b. Training Stability | ❌ **GANs are harder to train** because of the adversarial nature between the generator and discriminator. Training can become unstable, leading to mode collapse (where the generator produces limited types of images). | ✅ **VAEs are easier to train** since they optimize a well-defined loss function (reconstruction loss + KL divergence). However, balancing these losses can be challenging. |
| c. Latent Space Representation | 🚫 **GANs do not explicitly learn a structured latent space**—the mapping between noise and generated data is learned indirectly, making interpolation less meaningful. | ✅ **VAEs learn a continuous and meaningful latent space**, enabling smooth interpolations and better representation learning. |

**Potential Improvements via Hyperparameter Tuning**

**1. Improving GANs**

* **Use Wasserstein GAN (WGAN)**: Helps improve stability by minimizing a Wasserstein distance instead of traditional adversarial loss.
* **Feature Matching Loss**: Encourages the generator to produce images similar to the real ones based on feature statistics.
* **Spectral Normalization**: Stabilizes discriminator updates, preventing gradient explosions.
* **Better Architecture (DCGAN, StyleGAN)**: Use CNNs with batch normalization and leaky ReLU to improve generated image quality.
* **Hyperparameters to Tune:**
  + **Learning rate:** 1e-4 to 1e-3 (often different for generator and discriminator)
  + **Batch size:** 64–256
  + **Latent vector size:** 100–512
  + **Optimizer:** Adam with β1 = 0.5, β2 = 0.999

**2. Improving VAEs:**

* **β-VAE (Beta-VAE)**: Introduces a weighting factor on KL-divergence to improve disentanglement of features in the latent space.
* **Deeper Networks**: Use CNNs instead of fully connected layers for better image generation.
* **Better Likelihood Models**: Instead of simple Gaussian priors, use more expressive priors like normalizing flows.
* **Hyperparameters to Tune:**
  + **Learning rate:** 1e-4 to 1e-3
  + **Latent space size:** 16–128
  + **KL divergence weight (β-VAE):** β = 1 (default), increase to improve disentanglement