Data Science Lab - Project 2

MNIST style images generation through noise Variational Auto-Encoders

Ruben Ifrah, Théo Le Pendeven, Filomène Roquefort
M2 IASD - Dauphine-PSL

November 2025

Contents

1 Introduction

This project focuses on generating MNIST-style images using a fixed GAN architecture. Within that constraint, we investigate techniques to improve generated samples quality, measured with FID, precision, and recall. Our work explores conditional generation, strategies to slow discriminator convergence, and the use of Gaussian mixtures in the latent space.

The report is organized as follows. Section 3 presents our Conditional GAN implementation. Section 2 outlines training overview explaining the approaches used to regulate the discriminator. Section 4 introduces the Gaussian mixture extension. Section 5 summarizes the experimental setup and evaluates the impact of each modification. Section 6 concludes with the main insights.

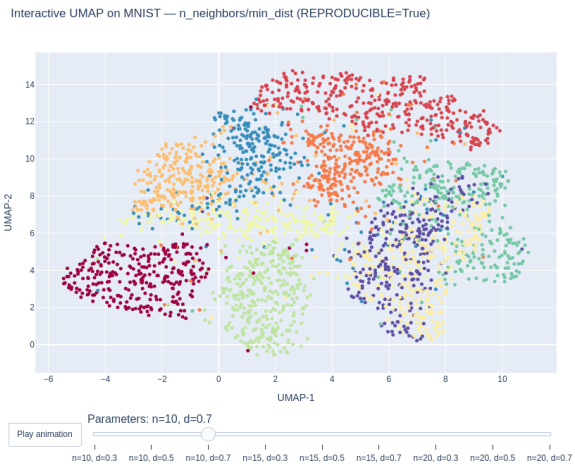
1.1 Visualizing MNIST Structure with UMAP

Before training generative models, it is useful to understand the geometric structure of the MNIST dataset in a low-dimensional space. We employ UMAP (Uniform Manifold Approximation and Projection), a non-linear dimensionality reduction technique that preserves both local neighborhoods and global cluster organization.

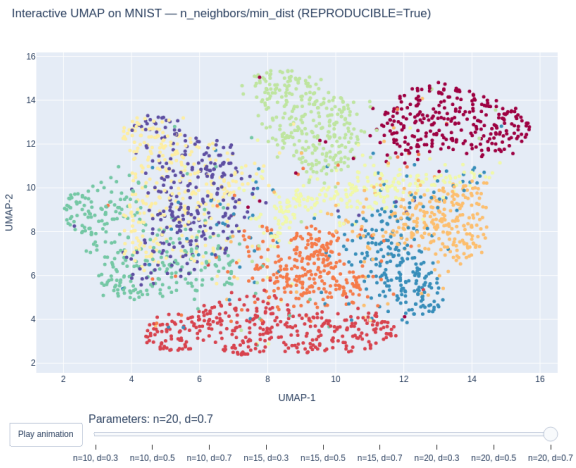
Applied to the 60,000 MNIST training images (flattened into 784-dimensional vectors), UMAP reveals a clear cluster separation between digits. The method highlights:

- well-defined clusters for most classes (e.g., digits 0, 1, 6, 7),
- overlapping regions for visually similar digits (e.g., 3 vs 5, 4 vs 9),
- smooth intra-class manifolds, indicating variability within each label.

These geometric observations motivate the use of more expressive latent distributions in our GAN, such as the Gaussian mixture learned by the LabelNoiseVAE (Section 4).



(a) UMAP projection with $n\_neighbors = 10$ and `min_dist` $= 0.7$.



(b) UMAP projection with $n\_neighbors = 20$ and `min_dist` $= 0.7$.

Figure 1: Comparison of two UMAP embeddings of MNIST. Varying the number of neighbors modifies the balance between local and global structure.

2 Training procedure

Implementation.

For training, we **initialized weights** using a normal distribution $\mathcal{N}(0, 0.02^2)$ and we used **Adam optimizers** with $(\beta_1, \beta_2) = (0.5, 0.999)$.

We used stabilization technique to slow down the Discriminator learning.

2.1 Label smoothing for slowing Discriminator learning

Label smoothing replaces the hard targets used for real samples with slightly lower values. Following the approach proposed by Salimans et al. [2], we replaced the real label, 1, by 0.9. This prevents the discriminator from becoming too confident too fast and reduces the magnitude of its gradients.

In our experiments, this modification stabilizes training in the early epochs and improve performance.

The discriminator is trained on:

$$\mathcal{L}_D = \text{BCE}(D(x_{\text{real}}),\ \mathbf{0.9}) + \text{BCE}(D(x_{\text{fake}}), 0),$$

while the generator minimizes:

$$\mathcal{L}_G = \text{BCE}(D(x_{\text{fake}}), \mathbf{0.9}).$$

2.2 Instance noise

Arjovsky and Bottou [4] suggest adding noise directly to the discriminator input, as another way to regularize its learning. They argue that GAN training is unstable when the supports of the real distribution and the generated distributions do not overlap. Adding noise ensures partial overlap and smooths the discriminator's decision boundary.

Instance noise is a more practical implementation of this idea developed by Sønderby et al [5]. They suggest adding Gaussian noise to both real and fake images, gradually reducing its variance during training. This delays the discriminator's ability to exploit fine pixel-level differences from the beginning, giving the generator time to improve before the discriminator becomes too sharp.

At each iteration we add Gaussian noise $\mathcal{N}(0, \sigma_t^2)$ to both real and fake images:

$$\tilde{x} = x + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_t^2 I).$$

The standard deviation σ_t decays linearly from 0.05 at epoch 1 to 0 at epoch 10:

$$\sigma_t = 0.05 \cdot \max\left(0, 1 - \frac{t}{10}\right).$$

3 Conditional GAN

A Conditional GAN extends the standard GAN by providing both the generator and the discriminator with an explicit label or class embedding. This conditioning forces the generator to produce samples consistent with the requested class and trains the discriminator to judge the joint pair (x, y), giving direct control over the output class.

3.1 Architecture and input conditioning

For each batch, instead of feeding a 100-dimensional noise vector to the generator, we concatenate a 90-dimensional noise vector $z \sim \mathcal{N}(0, I)$ with a one-hot encoded label $y \in \mathbb{R}^{10}$:

$$g_{\text{input}} = [z \,\|\, y].$$

Similarly, The discriminator receives the flattened image concatenated with the same label y, whether x is a real sample or a generated one :

$$d_{\text{input}} = [x \,\|\, y].$$

This setup forces both networks to incorporate the class information in their predictions and enables class-controlled generation.

3.2 Adapted training procedure

The discriminator is trained on:

$$\mathcal{L}_D = \text{BCE}(D(x_{\text{real}}, \mathbf{y}), t) + \text{BCE}(D(x_{\text{fake}}, \mathbf{y}), 0),$$

while the generator minimizes:

$$\mathcal{L}_G = \text{BCE}(D(x_{\text{fake}}, \mathbf{y}), t).$$

where t is the target for true images (1, or 0.9 if label smoothing).

Implementation.

4 Gaussian Mixture via LabelNoiseVAE

To enrich the latent space of the generator and reduce mode collapse, we replace the standard Gaussian noise z by a class-conditioned latent distribution learned using a small Variational Autoencoder, which we call **LabelNoiseVAE**. The goal is to learn, for each label, a richer and more structured latent distribution than the isotropic Gaussian used in a standard GAN.

Implementation.

4.1 Motivation

In the vanilla Conditional GAN, the latent vector z is sampled as:

$$z \sim \mathcal{N}(0, I).$$

However, the true latent structure of MNIST digits is highly non-Gaussian and class-dependent. We therefore learn for each class c a Gaussian distribution:

$$z \mid c \sim \mathcal{N}(\mu_c, \Sigma_c),$$

which collectively form a Gaussian mixture across classes.

4.2 LabelNoiseVAE Architecture

The LabelNoiseVAE receives:

$$(\text{one-hot label } y,\ \epsilon \sim \mathcal{N}(0, I)),$$

and outputs:

$$z = \mu(y, \epsilon) + \exp(\tfrac{1}{2} \log \sigma^2(y, \epsilon)) \odot \eta, \quad \eta \sim \mathcal{N}(0, I).$$

During generation: - we either draw z from the learned distribution, - or set $z = \mu(y)$ when using the `--use_mu` deterministic mode.

The generator then receives:

$$g_{\text{input}} = [z \,\|\, y].$$

4.3 Training the Noise VAE

The VAE is trained independently from the GAN. Its objective combines:

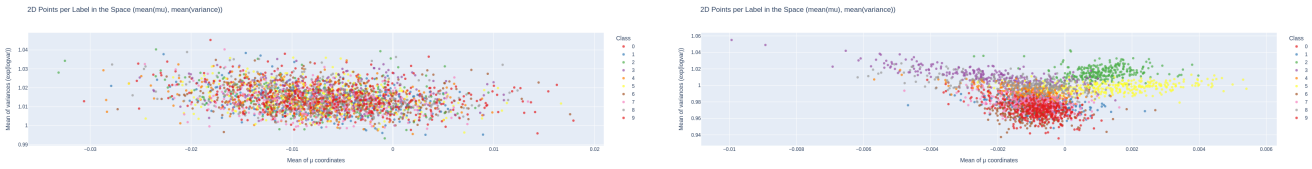
- a reconstruction loss in the latent space of the GAN,
- a KL divergence regularization term,
- class-dependent conditioning through y.

Once trained, the VAE serves as a learned Gaussian mixture sampler that replaces the simplistic $z \sim \mathcal{N}(0, I)$.

4.4 Empirical Analysis of Latent Distributions

We visualize for each label the mean of the latent coordinates and the mean variance:

$$\bar{\mu}_c = \mathbb{E}[\mu(z \mid c)], \qquad \bar{\sigma}_c^2 = \mathbb{E}[\exp(\log \sigma^2(z \mid c))].$$

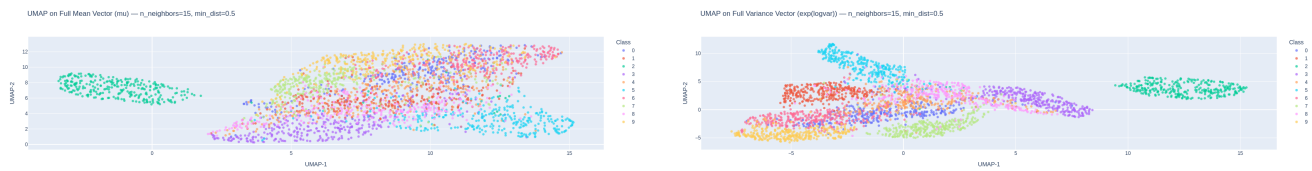


(a) without the Gaussian mixture.

(b) using the VAE-based Gaussian mixture.

Figure 2: Mean and variance per class without the Gaussian mixture (standard $\mathcal{N}(0, I)$) and using the VAE-based Gaussian mixture.

With the noise VAE, classes organize into well-separated clusters, each with consistent variance (Figure 2b). Without the VAE, all classes collapse around nearly identical means and variances (Figure 2a) — confirming that the standard latent space cannot encode class-specific structure. This richer latent geometry leads to better sample diversity and improved recall scores.



(a) UMAP on the full mean vector μ.

(b) UMAP on the full variance vector σ^2.

Figure 3: UMAP embeddings of the latent mean and variance vectors produced by the Label-VAE.

To further investigate the structure of the latent representations, we apply UMAP on the full mean vector $\mu \in \mathbb{R}^d$ and on the full variance vector $\sigma^2 = \exp(\log \sigma^2) \in \mathbb{R}^d$ produced by the Label-VAE as shown Figure 3.

These projections show that the VAE learns class-dependent latent geometries not only in mean space but also in variance space. The organization differs strongly between μ and σ^2, confirming that the model encodes label information in both the central location and the uncertainty structure of the latent distribution.

5 Experiments and Results

5.1 Experimental Protocol

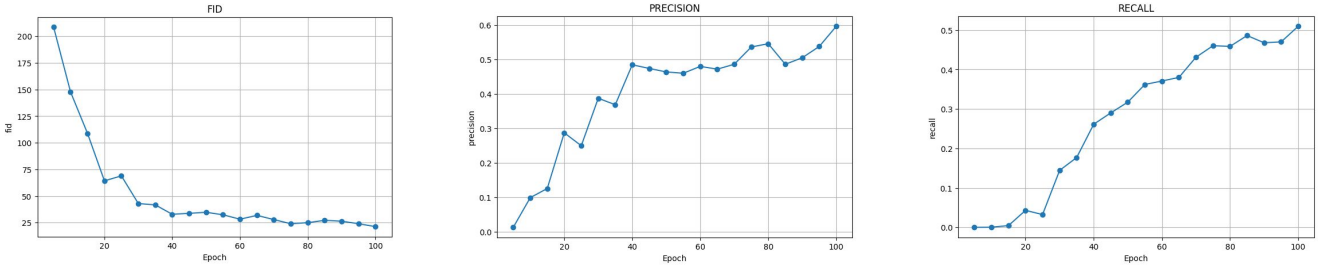
All models were trained for 100 epochs using the same optimizer, architecture, and training schedule. We evaluate four optional components:

- **instance noise**: Gaussian noise added to real and fake images,
- **label smoothing**: real labels set to 0.9,
- **conditional training**: one-hot labels concatenated to both G and D inputs,
- **LabelNoiseVAE**: a class-dependent Gaussian mixture replacing the standard latent prior.

Every 5 epochs, we generated 10,000 samples and computed the **FID** and the **PRDC** metrics (precision, recall, density, coverage) [6].

5.2 Training Dynamics

Figure 4 displays the evolution of FID, precision and recall during training for the configuration combining all four components (instance noise, label smoothing, conditional GAN, and VAE). FID decreases rapidly during the first epochs and then stabilizes, while precision and recall steadily increases as the generator learns sharper digit structures. This suggests longer runs would provide better results.



FID over training epochs. Precision over training epochs. Precision over training epochs.

Figure 4: Training curves for the full configuration (noise + smoothing + conditional + VAE).

5.3 Results on the platform

Table 1: Performance comparison of our Conditional and GMM-Conditional models, with metrics computed on the course evaluation platform.

Model	Time	FID	Precision	Recall
Conditional	94.24	28.82	0.81	0.43
GMM + Conditional	101.45	25.88	0.85	0.48

5.4 Ablation Study

Table 2 summarizes the impact of enabling or disabling each optional component. Each row corresponds to a separate full training run, evaluated with 10,000 generated samples.

Implementation.

Table 2: Ablation of instance noise, label smoothing, conditional generation, and the VAE latent prior, with metrics computed on our own.

noise	smooth	cond	VAE	FID	precision	recall	density	coverage
x				49.22	0.46	0.30	0.28	0.32
	x			38.38	0.56	0.44	0.41	0.38
x	x			39.68	0.52	0.44	0.41	0.39
		x		23.71	0.54	0.52	0.35	0.51
x		x		25.83	0.52	0.51	0.34	0.53
	x	x		27.17	0.48	0.52	0.30	0.49
x	x	x		23.02	0.54	0.54	0.36	0.53
	x	x	x	26.92	0.55	0.55	0.35	0.52
x		x	x	23.92	0.54	0.54	0.36	0.53
	x	x	x	24.30	0.54	0.54	0.35	0.52
x	x	x	x	23.02	0.54	0.54	0.36	0.53

5.5 Analysis

Three clear trends emerge from our ablation study:

- **Conditional GAN is the most impactful single component.** Enabling conditioning reduces FID from approximately 40 to the 23–27 range and significantly improves recall.
- **Label smoothing consistently reduces FID and increases precision.** It stabilizes early training and prevents discriminator overconfidence.
- **The VAE-induced Gaussian mixture improves diversity.** Recall and coverage increase when replacing the standard $\mathcal{N}(0, I)$ prior with the learned mixture.

The combination of all four techniques yields the best overall metrics (FID = 23.0), confirming that a smoother discriminator and a richer latent distribution complement each other.

5.6 Qualitative Samples

Figure 5 shows examples generated by our best-performing model.

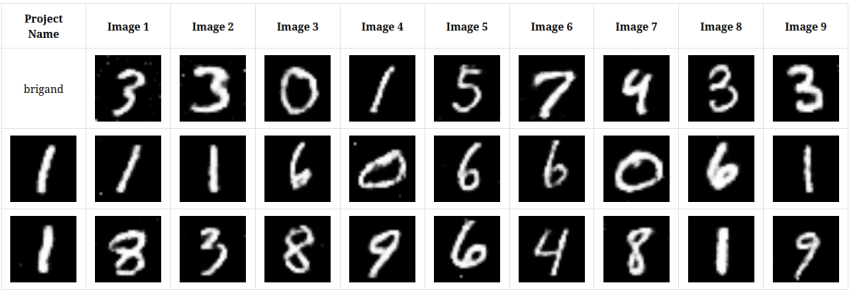


Figure 5: Samples generated by the best model (instance noise + smoothing + conditional + VAE).

6 Conclusion and Lessons Learned

Our experiments show that competitive GAN performance can be achieved even under limited computational resources and without extensive hyperparameter sweeps. The Conditional GAN provides a strong baseline, and the Gaussian mixture extension consistently improves diversity and class coverage, as reflected in higher recall scores.

A natural next step would be to introduce an acceptance–rejection mechanism during generation to filter low-quality samples. Such a filtering strategy could further increase recall and reduce outliers, potentially leading to better FID scores.

However, implementing and tuning this mechanism requires a more reliable experimental setup, including stable metric computation and functional large-scale training infrastructure (e.g., Mesonet). With better tooling, we expect that our models could be improved significantly and evaluated more rigorously.

Overall, this project highlights the robustness of conditional generative modelling and the practical benefits of enriching the latent distribution through a Gaussian mixture.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Networks*. arXiv:1406.2661, 2014. https://arxiv.org/abs/1406.2661.

[2] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. *Improved Techniques for Training GANs*. CoRR, 2016. http://arxiv.org/abs/1606.03498.

[3] A. Radford, L. Metz, and S. Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. CoRR, abs/1511.06434, 2015. https://api.semanticscholar.org/CorpusID:11758569.

[4] M. Arjovsky and L. Bottou. *Towards Principled Methods for Training Generative Adversarial Networks*. arXiv:1701.04862, 2017. https://arxiv.org/abs/1701.04862.

[5] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. *Amortised MAP Inference for Image Super-resolution*. arXiv:1610.04490, 2017. https://arxiv.org/abs/1610.04490.

[6] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. *Improved Precision and Recall Metric for Assessing Generative Models*. arXiv:1904.06991, 2019. https://arxiv.org/abs/1904.06991.