Learning Latent Space Representations

Data Science Lab – Master IASD
Team name: Nano Kiwi
Université Paris Dauphine – PSL

Paul Martinetti, Mathéo Quatreboeufs, Hannah Gloaguen

November 17, 2025

1 Introduction

The objective of this project is to improve the training procedure of Generative Adversarial Networks (GANs) by relying on evaluation metrics that capture both the *quality* and the *diversity* of generated images. Classical metrics such as the the Fréchet Inception Distance (FID) provide a global assessment of generative models, but they do not explicitly disentangle these two dimensions. In our work, we focus on three complementary metrics: FID, Precision, and Recall.

The **Fréchet Inception Distance (FID)** [4] measures the similarity between real and generated data by comparing the first two moments of their feature representations extracted by an Inception network. A lower FID indicates that the synthetic distribution is close to the real distribution in terms of global statistics.

The notions of **Precision** and **Recall** for generative models, introduced by Kynkäänniemi et al. [5], allow us to separately assess the two main failure modes of GANs. *Precision* measures the proportion of generated samples that lie on the support of the real data distribution, and therefore reflects the *visual quality* of generated images. Conversely, *Recall* measures the proportion of the real data support that is captured by the generator, thereby quantifying *diversity*. A model can thus achieve high precision but low recall (*mode dropping*), or high recall but low precision (*blurry or unrealistic samples*). Optimizing these metrics independently is therefore crucial for a fine-grained evaluation of generative models. We use the implementation of the recall and precision from [6], [2] to test locally. We used that because we thought it was the one from the testing platform.

All experiments are conducted on the **MNIST dataset**, a benchmark of 28×28 grayscale images of handwritten digits from 0 to 9. Despite its simplicity, MNIST remains a useful testbed to study different aspects of GAN training, as both mode coverage and sample fidelity can be visually and quantitatively assessed.

The purpose of this project is to investigate how modifying the *latent distribution* and the *GAN loss function* can improve these three metrics. We first experiment with replacing the standard Gaussian latent prior by more expressive distributions such as mixtures of Gaussians. We then explore alternative objectives, in particular the Wasserstein GAN with Gradient Penalty (WGAN-GP) loss, in order to stabilize training and reduce mode collapse. By comparing the FID, Precision, and Recall of each method, we aim to identify which design choices lead to the best trade-off between image quality and diversity.

2 Static GM-GAN: Improving recall with a fixed Gaussian mixture

Model	FID	Precision	Recall
(platform) Vanilla GAN	108.1	0.56	0.09

Table 1: FID, precision, and recall for the baseline GAN (vanilla)

In our exploration of Generative Adversarial Networks (GANs), we found it difficult to achieve satisfactory recall. To address this challenge, we decided to implement Gaussian Mixture GAN (GM-GAN), based on the work of Ben-Yosef and Weinshall, "*Gaussian Mixture Generative Adversarial Networks for diverse Dataset, and the Unsupervised Clustering of Imaged*" [1].

The paper describes three main variants of the problem: unsupervised static GM-GAN, unsupervised dynamic GM-GAN, and supervised GM-GAN. Of these three architectures, our experimental focus is exclusively on the unsupervised static GM-GAN model.

Standard GANs typically use either a multivariate uniform distribution (e.g. $\mathcal{U}[-1,1]^d$) or a multivariate normal distribution (e.g. $\mathcal{N}(0, I_{d\times d})$ as the prior distribution p_Z to sample the latent space. GM-GANs models maintain the same objective function as the classic GAN model, but replaces this unimodal prior distribution p_Z with a multi-modal distribution. The primary goal is to provide a better fit to the target data distribution, allowing the model to generate more diverse samples.

The paper argues that a multi-modal distribution is better suited to match the complexity of the real distribution. This is especially relevant because each class in the dataset MNIST may not share the exact same probability distribution, and some classes may event exhibit significant intra-class diversity (subclasses) resulting in distinct probability spaces.

To this end, the paper proposes to use a mixture of Gaussians as the multi-modal prior distribution. Formally, this is defined as:

$$p_Z(z) = \sum_{k=1}^{K} \alpha_k * p_k(z)$$

Here K represents the number of Gaussians in the mixtures, $\{\alpha_k\}_{k=1}^K$ is a categorical random variable, and $p_k(z)$ is the multivariate Normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$. For the majority of our experiments on the MNIST dataset, we set the number of Gaussians at $K = 10$ and assumed a uniform mixture, $\alpha_k = 1/K$, as the data set's ten classes are known to be approximately uniformly distributed.

In the static GM-GAN the parameters of the Gaussian Mixture distribution (the means μ_k and the covariance matrices Σ_k) are fixed prior to training.

Specifically:

- Hyperparameters: $c \in \mathbb{R}_+$ and $\sigma \in \mathbb{R}_+$ are static hyperparameters that the user chooses before training begins.
- Mean vector μ_k: each mean vector is uniformly sampled from the uniform multivariate distribution $\mathcal{U}[-c, c]^d$.
- Covariance Matrices Σ_k: each covariance matrices takes the form $\sigma^2.I_{d\times d}$ as in the article [1]

Note that in our experiment the Generator is fixed, so the dimension on the latent vectors is fixed to $d = 100$.

Impact of σ^2

The quality of generated samples is inversely related to the scaling factor σ. A smaller σ yields higher-quality (or more precise) samples, while a larger σ results in lower-quality samples.

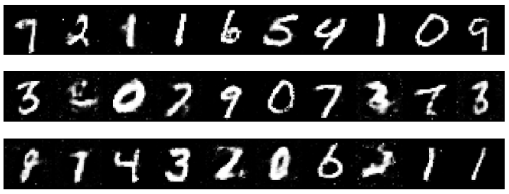

Figure 1: Generated MNIST handwritten digits showing a decrease in quality and an increase in diversity as the latent space scaling factor (σ) increases, top: $\sigma = 0.15$, middle $\sigma = 0.5$, bottom $\sigma = 1$

For our final model, we choose the parameters $c = 0.1$, $\sigma = 0.15$ and 90 epochs.

Model	FID	Precision	Recall
(local) Static GM-GAN	35.82	0.28	0.52

Table 2: FID, precision, and recall for the Static GM-GAN.

3 WGAN(-GP): modifying the objective function

After modifying the latent space, we were disappointed by the performance obtained with Gaussian mixtures. Although they improved recall, the training process remained unstable and the generator still suffered from vanishing gradients. Our new objective is therefore to use a method that **avoids vanishing gradients** and ensures that the training dynamics remain **stable**. For this reason, we introduce a method proposed in [3].

3.1 Motivation

The classical GAN objective relies on the Jensen-Shannon divergence. In high-dimensional settings, the supports of the real and generated distributions seldom overlap, which leads the divergence to saturate. Consequently:

- gradients flowing to the generator **vanish**,
- training becomes **unstable**,
- mode collapse is likely to occur.

WGAN circumvents this issue by replacing the divergence with the **Wasserstein-1 distance**, which is much smoother and provides meaningful gradients even when the distributions are far apart.

3.2 WGAN Objective

WGAN reformulates the discriminator into a *critic* D that estimates the Wasserstein distance under the constraint that D must be 1-Lipschitz.

The ideal objective is:

$$W(P_r, P_g) = \max_{Lip(D)\leq 1} \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim P_z}[D(G(z))].$$

The critic loss used during training is therefore:

$$\mathcal{L}_D = -\mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{z\sim P_z}[D(G(z))].$$

And the generator minimizes the same distance:

$$\mathcal{L}_G = -\mathbb{E}_{z\sim P_z}[D(G(z))].$$

3.3 Gradient Penalty (WGAN-GP)

WGAN-GP, introduces a soft constraint through the **gradient penalty**:

$$\mathcal{L}_{GP} = \lambda\, \mathbb{E}_{\hat{x}} \left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2,$$

(Idea: $Lip(D) = 1$ if $\|\nabla_{\hat{x}} D(\hat{x})\|_2$ is close to 1 for some x)

where $\hat{x}$ are points sampled along straight lines between real and generated samples:

$$\hat{x} = \epsilon x + (1-\epsilon) G(z), \qquad \epsilon \sim \mathcal{U}(0,1).$$

The full critic objective becomes:

$$\mathcal{L}_D^{\text{WGAN-GP}} = -\mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{z\sim P_z}[D(G(z))] + \lambda\, \mathbb{E}_{\hat{x}} \left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2.$$

3.4 Training Procedure

Training WGAN-GP follows a specific schedule:

- the critic is updated $n_{\textbf{critic}}$ **times** (commonly 5) per generator step,
- the critic maximizes the estimated Wasserstein distance,
- the generator minimizes it by producing samples that reduce this distance.

This leads to:

- smoother and more reliable gradients,
- greatly improved stability,
- often better sample quality.

Choice of hyperparameters:

- $\lambda = 10$, learning rate = 0.0001, 100 epochs, batch size = 64.

Model	FID	Precision	Recall
(local) WGAN(-GP)	26.90	0.42	0.59
(plateform) WGAN(-GP)	35.48	0.48	0.32

Table 3: FID, precision, and recall for WGAN(-GP).

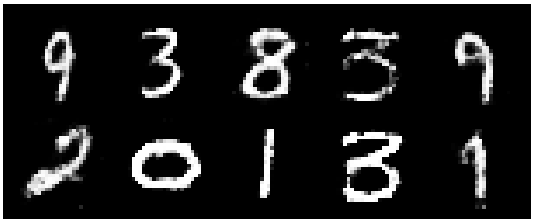

Figure 2: Examples of images generated by the WGAN GP.

4 Conditional WGAN-GP with Learnable Class-Specific Latent Distributions

4.1 Motivation

Having established that WGAN-GP provides stable training dynamics, we now combine this framework with the idea of using multiple Gaussian distributions in the latent space. However, instead of learning an unsupervised mixture like in GM-GAN, we leverage the supervised information available in MNIST: the digit labels.

Our approach explicitly associates each class (digit 0–9) with its own learnable Gaussian distribution $\mathcal{N}(a_i, b_i)$ in the latent space, where the parameters a_i (mean) and b_i (standard deviation) are trainable. This creates a conditional generation mechanism that makes both tasks more challenging compared to unconditional generation:

- The **generator** can no longer simply produce "any digit between 0 and 9" – it must generate the correct digit i by sampling from the corresponding class-specific distribution $\mathcal{N}(a_i, b_i^2 I)$
- The **discriminator** must distinguish real from fake images conditioned on the digit label – effectively learning separate real/fake boundaries for each of the 10 classes

Crucially, WGAN-GP's stable gradients allow us to tackle this more complex objective without the training instabilities that would plague vanilla GANs.

4.2 Architecture

Our model introduces three key components:

1. Conditional Latent Sampler: Instead of sampling $z \sim \mathcal{N}(0, I)$, we maintain 10 learnable Gaussian distributions, one per digit:

$$z \sim \mathcal{N}(a_i, b_i^2 I) \quad \text{for digit } i \in \{0, \ldots, 9\} \tag{1}$$

The parameters $\{a_i, b_i\}_{i=0}^{9}$ are **learned during training** via backpropagation through the generator loss.

2. Conditional Generator: The generator G remains unchanged architecturally, but now receives latent codes sampled from class-specific distributions:

$$G(z), \quad \text{where } z \sim \mathcal{N}(a_i, b_i^2 I) \tag{2}$$

3. Conditional Discriminator (Critic): The discriminator is augmented to take both the image x and its label i as input:

$$D(x, i) \to \text{score} \tag{3}$$

The label is embedded and concatenated with the image features.

4.3 Experimental Results

Training Setup: We trained the model for 600 epochs with learning rates of 0.0001 for both generator and discriminator, batch size 64, $n_{\text{critic}} = 3$, and $\lambda_{GP} = 10$.

Hyperparameter Analysis: Reducing n_{critic} from the standard value of 5 to 3 provided substantial computational savings (approximately 40% faster training) with minimal impact on final performance. However, further reduction below 3 led to noticeable quality degradation, suggesting that at least 3 discriminator updates per

generator step are necessary for stable convergence. The gradient penalty coefficient $\lambda = 10$ proved essential for preventing loss divergence throughout training.

Training Progression: FID was evaluated every 100 epochs using 10,000 generated samples. The metric showed continuous improvement throughout training, with the rate of progress beginning to plateau around epoch 600 without fully converging, suggesting potential for further improvement with extended training.

Distribution Matching: We compared two generation strategies: uniform distribution (equal samples per class) and MNIST-matched distribution (reflecting the actual class frequencies in the training set, ranging from 9.04% for digit 5 to 11.24% for digit 1). Both approaches yielded nearly identical results, indicating that the model successfully learned the underlying data distribution regardless of the sampling strategy used during evaluation.

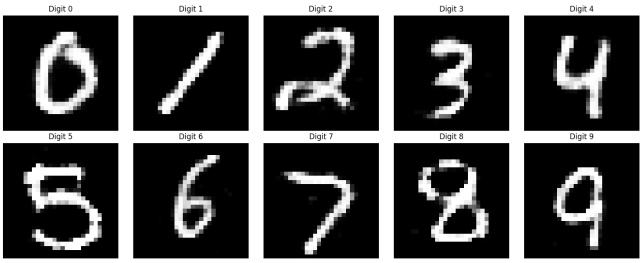

Figure 3: Examples of images generated by Conditional WGAN-GP with learnable latent distributions at epoch 590.

Epoch	FID	Precision	Recall
100	35.31	0.432	0.492
200	24.17	0.423	0.587
300	20.23	0.426	0.633
400	17.29	0.515	0.637
500	15.44	0.529	0.657
600	**15.15**	**0.516**	**0.670**

Table 4: Evolution of evaluation metrics during training (MNIST distribution, 10,000 samples). Metrics computed using pytorch-fid and the IPR implementation with $k = 5$ nearest neighbors.

Platform Evaluation:

Model	FID	Precision	Recall
Conditional WGAN-GP	24.24	0.50	0.20

Table 5: Performance on the course platform's evaluation metrics.

Discrepancy Analysis: We observe significant differences between our local evaluation and the platform results, particularly in FID (15.15 vs 24.24) and Recall (0.670 vs 0.20). Despite extensive investigation, the exact source of this discrepancy remains unclear.

5 Conclusion

This work addressed two fundamental challenges in GAN training: stability and mode coverage.

Addressing Training Instability: Standard GANs suffer from convergence issues where one model can dominate the other, leading to gradient explosion or vanishing. We resolved this by implementing WGAN-GP, which replaces the Jensen-Shannon divergence with the Wasserstein distance and enforces the Lipschitz constraint via gradient penalty. This provides stable, reliable gradients throughout training.

Ensuring Mode Coverage: To achieve high recall and prevent mode dropping, we introduced conditional generation with learnable class-specific latent distributions $\mathcal{N}(a_i, b_i^2 I)$. By learning distinct Gaussian distributions for each digit class during training, we force the generator to cover all modes uniformly. This explicit conditioning, combined with label information in the discriminator, ensures comprehensive mode coverage.

Results: Our approach achieves FID of 15.15, Precision of 0.516, and Recall of 0.670, with our metrics, demonstrating both high sample quality and excellent mode coverage.

References

[1] Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. *arXiv preprint arXiv:1808.10356*, 2018.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[5] Tuomas Kynkäänniemi, Kimmo Kangas, Victor Lempitsky, and Tero Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

[6] Alexandre Verine, Benjamin Negrevergne, Muni Sreenivas Pydi, and Yann Chevaleyre. Precision-recall divergence optimization for generative modeling with gans and normalizing flows, 2023.