Nano Kiwi

Paul Martinetti, Mathéo Quatreboeufs, Hannah Gloaguen

M2 IASD – Data Science Lab

November 17, 2025

Slide 1 – Baseline GAN: Architecture, Loss and Metrics

Architecture

- We start from the **original GAN architecture** provided in the assignment.
- Latent prior: $z \sim \mathcal{N}(0, I)$.

Classical GAN loss

$$\min_G \max_D \; \mathbb{E}_{x \sim P_{\text{data}}}[\log D(x)] \; + \; \mathbb{E}_{z \sim \mathcal{N}(0,I)}[\log(1 - D(G(z)))].$$

Baseline performance

- FID: **108.1**
- Precision: **0.56**
- Recall: **0.09**

Improving Recall with Static GM-GAN

Motivation [1]

- The baseline GAN shows **limited recall**:
- The generator fails to cover all modes of the real data distribution.

Static Gaussian Mixture GAN (Static GM-GAN)

- Replace the standard normal prior by a **mixture of K Gaussians**:

$$z \sim \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\mu_k, \sigma_k^2 I).$$

- Means μ_k are sampled once at initialization and kept fixed.
- The multi-modal latent prior encourages the generator to explore different regions of the data space.

Static GM-GAN results (local)

- FID: **35.82**
- Precision: **0.28**
- Recall: **0.52** (improved recall compared to baseline)

Modifying the Objective: Introducing WGAN

Motivation

- So far, we modified the **latent distribution** (Static and Dynamic GM-GAN).
- The classical GAN loss still suffers from:
  - Vanishing gradients
- We now **modify the training objective itself** using Wasserstein GAN (WGAN).

Wasserstein-1 distance

$$W(P_{\text{data}}, P_G) = \sup_{Lip(f)\leq 1} \mathbb{E}_{x\sim P_{\textbf{data}}}[f(x)] - \mathbb{E}_{z\sim P_Z}[f(G(z))].$$

Motivation

From GAN to Wasserstein GAN (WGAN)

The divergence used in the original GAN objective is replaced by the *(Wasserstein-1) distance*. The objective becomes:

$$\min_{G} \max_{D \in \mathsf{Lip}(1)} \Big(\mathbb{E}_{x \sim P_r} \big[D(x) \big] - \mathbb{E}_{z \sim P_z} \big[D(G(z)) \big] \Big),$$

where:

- P_r is the real data distribution,
- P_z is the prior distribution (e.g., $\mathcal{N}(0, I)$),
- D is constrained to be 1-Lipschitz.

WGAN-GP Objective

Enforcing the 1-Lipschitz constraint

WGAN-GP [2] introduces a **gradient penalty** applied to random interpolations between real and generated samples.

$$\hat{x} = \epsilon x_{\text{real}} + (1 - \epsilon)\, x_{\text{fake}}, \qquad \epsilon \sim \mathcal{U}(0, 1).$$

Gradient Penalty term:

$$\text{GP} \;=\; \mathbb{E}_{\hat{x}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right].$$

Full WGAN-GP critic loss:

$$L_D = \mathbb{E}_{x_{\text{fake}}}[D(x_{\text{fake}})] - \mathbb{E}_{x_{\text{real}}}[D(x_{\text{real}})] + \lambda\, \text{GP}.$$

Generator loss:

$$L_G = -\mathbb{E}_{x_{\text{fake}}}[D(x_{\text{fake}})].$$

WGAN-GP: Experimental Results

Training setup

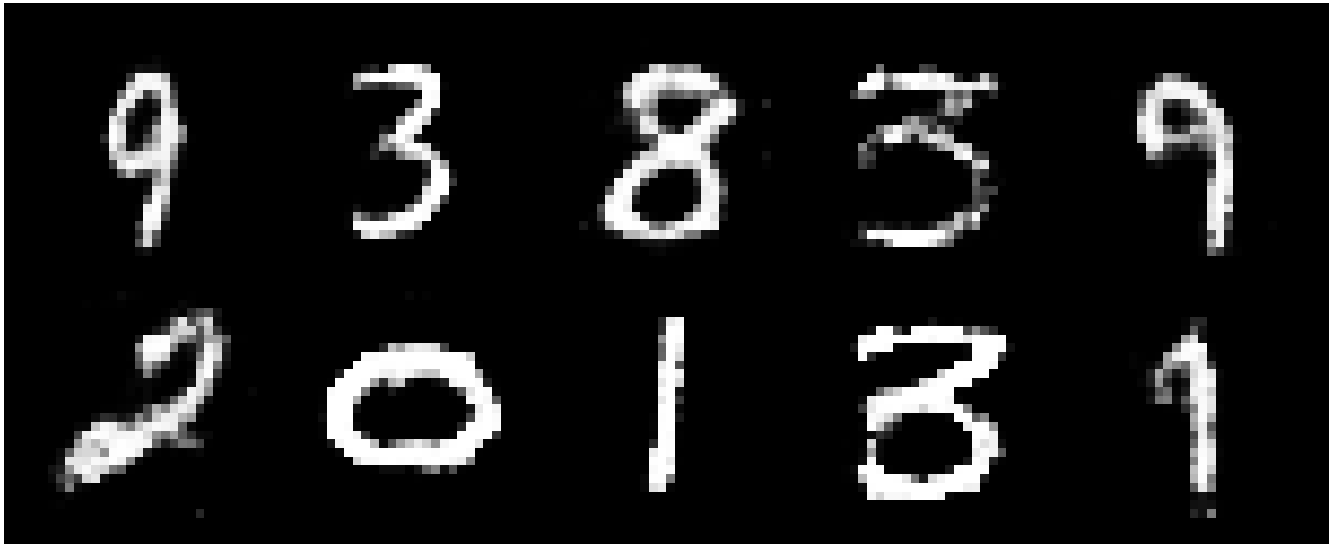
- Same architecture and latent prior as the baseline GAN.
- Loss replaced by the Wasserstein objective with gradient penalty.
- Gradient penalty weight: $\lambda = 10$.
- 5 discriminator updates per generator update.
- 100 epochs, batch size : 64

WGAN-GP: Experimental Results

Quantitative results

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

Observations



Conditional WGAN-GP with Learnable Latent Distributions

Goal: Each digit has its own learnable Gaussian in the latent space.

Conditional Discriminator:

$$D(x, i) \rightarrow \text{score}$$

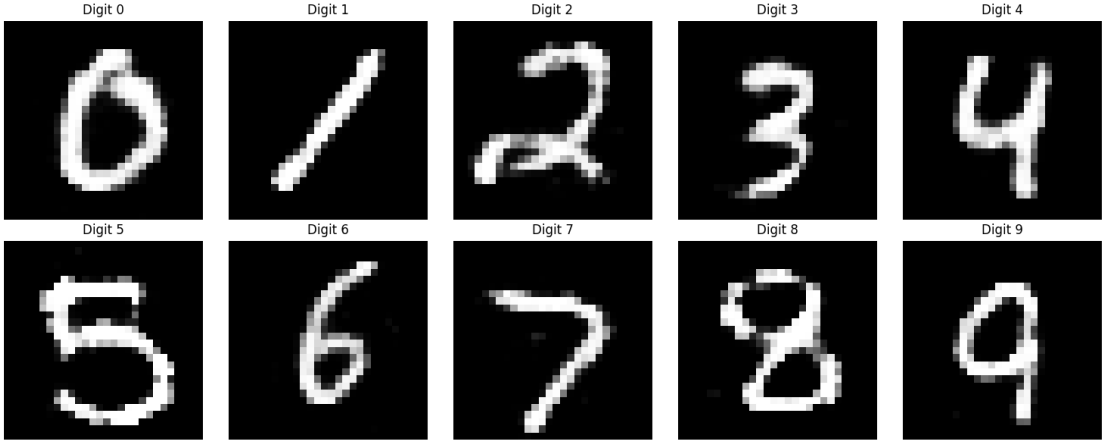
Conditional Generator:

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

Learnable parameters: $a_i, b_i \in \mathbb{R}^{100}$ for each digit $i \in \{0, ..., 9\}$

Results

Generated Samples (Epoch 600)



Training Evolution

Epoch	FID ↓	Precision	Recall
100	35.31	0.43	0.49
200	24.17	0.42	0.59
300	20.23	0.43	0.63
400	17.29	0.51	0.64
500	15.44	0.53	0.66
600	**15.15**	**0.52**	**0.67**

Global Comparison of All Models

Quantitative comparison

Model	FID	Precision	Recall
Baseline GAN	108.1	0.56	0.09
Static GM-GAN	35.82	0.28	0.52
WGAN-GP	26.90	0.42	0.59
WGAN-GP$_cond$	15.15	0.52	0.67

Key observations

- **Recall** improves when introducing a multi-modal latent space (Static & Dynamic GM-GAN).
- Dynamic GM-GAN generally outperforms Static GM-GAN, confirming the benefit of learning mixture parameters.
- WGAN improves training stability and mode coverage, but does not explicitly target recall as a metric.


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.


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.