,

Data Science Lab

Assignment 2 :

Learning latent space representations and application to image generation

GANibal Team

Maxime Bénard
Mehdi Benyamina
Elias Ben Rhouma

IASD — 2025

Université Paris-Dauphine

Our goal is to generate images of handwritten digits using the MNIST dataset, the two relevant metrics are : accuracy (between 0 and 1) which tells us about the quality of our generated samples ; recall (between 0 and 1) which tells us about the diversity of our generated samples. During the project, we implemented several methods, most of them are compatible because they can be merge in the same pipeline, with the aim to improve both accuracy and recall (or improving one without losing too much on the other).

1 Wasserstein GANs

Why switch from Vanilla GAN to WGAN ? The Vanilla GAN suffer three linked pathologies : (i) mode collapse, where the generator maps many latent codes to a few prototypes (some digits repeat, others are rare/missing) ; (ii) non-convergence, because adversarial gradient dynamics are rotational (the phase-portrait spirals) so losses and predictions oscillate rather than settle, even when an equilibrium exists, simultaneous gradient descent/ascent does not converge to it, but cycles around it. The generator and discriminator parameters can keep "chasing" each other, causing oscillating losses and unstable training instead of settling to a good solution. (iii) and vanishing gradients, when the discriminator saturates (real→1, fake→0) the JS-based, saturating generator loss provides almost no signal—the plots show mean D(G(z)) dipping toward 0 and the generator's gradient norm collapsing toward small values. Together, the critic overpowers early, gradients shrink or cycle, and G learns narrow, repetitive modes instead of covering the data distribution.

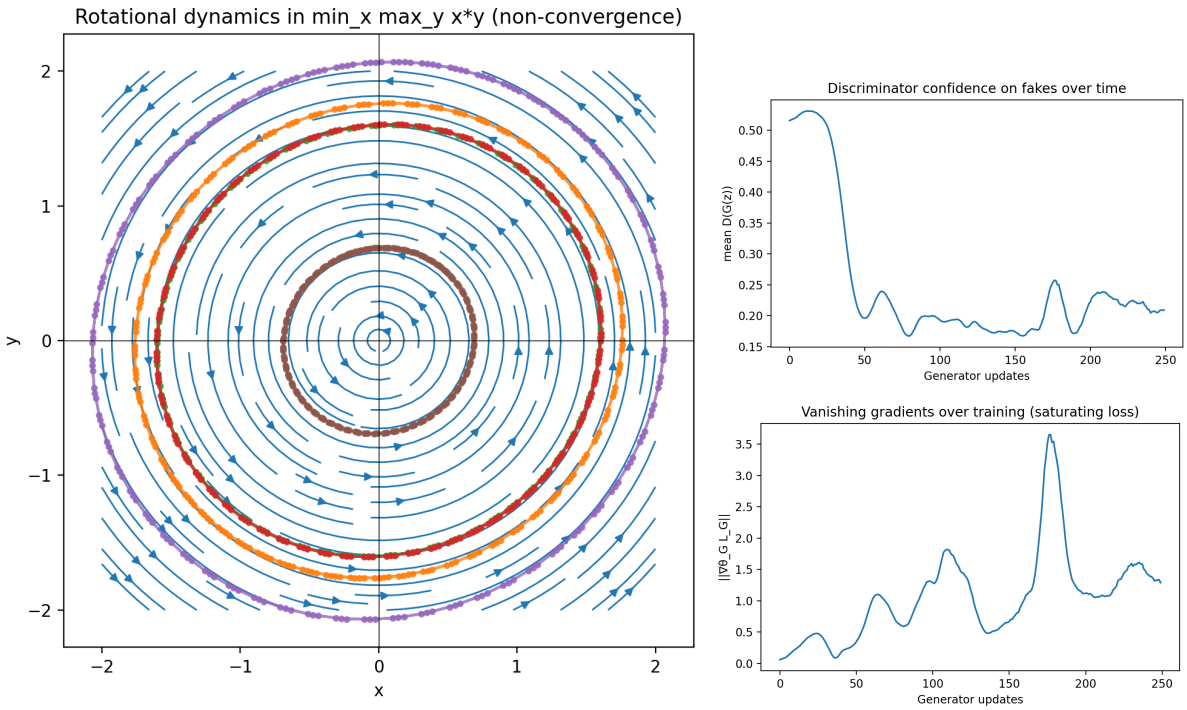

Figure 1 – Vanilla GAN pathologies on MNIST : (left) non-convergence ; (top-right) discriminator confidence ; (bottom-right) generator gradient norms

The switch to WGAN : Vanilla GANs minimize a Jensen–Shannon divergence whose gradient vanishes when p_g and p_{data} have little support overlap, leading to saturation, mode collapse, and cycling. WGAN replaces this with the Earth Mover distance, using the Kantoro-

vich–Rubinstein dual

$$\mathcal{W}_1(p_{\text{data}}, p_g) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)],$$

which yields smooth, informative gradients even when supports are disjoint and induces a weaker topology (distributions converge more easily). Practically, we (i) change the losses to

$$\mathcal{L}_D = \mathbb{E}[f(G(z))] - \mathbb{E}[f(x_{\text{real}})] \quad \text{and} \quad \mathcal{L}_G = -\mathbb{E}[f(G(z))],$$

(ii) enforce 1-Lipschitzness of the critic via weight clipping, a gradient penalty or spectral normalization, and (iii) use a few more critic steps per generator update. These changes mitigate vanishing gradients and improve stability and coverage.

2 WGAN with Weight Clipping

How to pick the clipping value c. In WGAN with weight clipping (method presented in [**WGAN**]), every critic parameter is constrained to $[-c, c]$ to encourage 1-Lipschitzness. If c is *too small*, the critic underfits (flat scores, weak gradients $\Rightarrow$ mode collapse). If c is *too large*, the critic becomes overly steep (training oscillations/instability). When training the WGAN (up to 20 epochs to keep training time reasonable), we observe that deeper layers saturate quicker. After testing, our choice for the clipping value is $c = 0.09$.

Clipping saturation
We pick the smallest c that yields :

1. a non-trivial Wasserstein estimate $\hat{W} = \mathbb{E}[D(x_{\text{real}})] - \mathbb{E}[D(G(z))]$ (not ≈ 0),
2. stable losses (no exploding spikes),
3. low clipping saturation (20%–30% of weights at $\pm c$ over many steps).

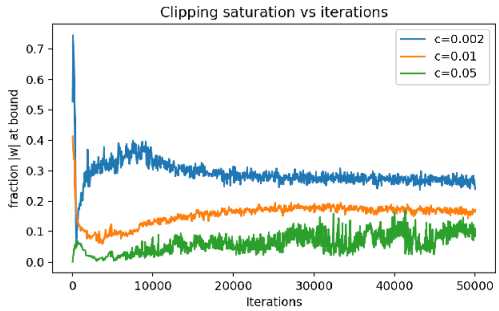

Figure 2 – Clipping saturation versus training iterations.

3 Wasserstein GAN with Gradient Penalty (WGAN-GP)

The original WGAN enforces the 1-Lipschitz constraint on the critic f_ψ via *weight clipping* : every weight is forced into $[-c, c]$. This crude constraint often harms the critic : for small c the network underfits (almost linear, flat scores, weak gradients), while for large c the critic becomes too sharp and training becomes unstable.

WGAN-GP replaces clipping with a *gradient penalty* that directly encourages the critic to have gradient norm 1 with respect to its input :

$$L_D = \mathbb{E}_{\tilde{x}\sim p_g}\big[f_\psi(\tilde{x})\big] - \mathbb{E}_{x\sim p_{\text{data}}}\big[f_\psi(x)\big] \tag{1}$$

$$+ \lambda\, \mathbb{E}_{\hat{x}}\Big(\|\nabla_{\hat{x}} f_\psi(\hat{x})\|_2 - 1\Big)^2, \tag{2}$$

$$L_G = -\mathbb{E}_z\big[f_\psi(G_\theta(z))\big], \tag{3}$$

where $\hat{x} = \epsilon x + (1-\epsilon)\tilde{x}$, $x \sim p_{\text{data}}$, $\tilde{x} \sim p_g$, and $\epsilon \sim \mathcal{U}[0,1]$. In other words, we sample points *on the straight lines* between real and generated samples and penalize the critic whenever $\|\nabla_{\hat{x}} f_\psi(\hat{x})\|_2$ deviates from 1. For the optimal Wasserstein critic, the gradient norm is 1 almost everywhere along these lines, so this penalty is a principled way to approximate the 1-Lipschitz constraint.

Advantages of gradient penalty (vs. weight clipping)

- **Better critics, less underfitting.** The critic is no longer forced into a tiny weight box ; it can use its full capacity while still being approximately 1-Lipschitz, leading to more meaningful Wasserstein estimates.
- **Smoother and more stable gradients.** Enforcing unit-norm gradients along real–fake interpolation paths yields smooth critic landscapes and stable generator updates, instead of the jagged, saturated behaviour we saw with clipping.
- **Less sensitive hyperparameters.** We replace a delicate clip value c (too small/too large breaks training) by a penalty weight λ that is much easier to tune in practice.
- **Improved mode coverage.** Because the critic provides informative gradients even when p_g and p_{data} live on low-dimensional, disjoint manifolds, the generator receives useful signals to move toward missing modes instead of collapsing around a few prototypes.
- **Compatibility with other tricks.** The gradient penalty term is just an extra expectation, so it works well with our other extensions (Gaussian mixture priors, conditioning, DRS) without changing their architecture.

4 Spectral Normalization for WGAN Critics

Spectral normalization (SN) is another way to approximately enforce the 1-Lipschitz constraint on the critic. Instead of constraining the *outputs* of f_ψ through a penalty term, SN directly rescales each weight matrix so that its largest singular value (its *spectral norm*) is equal to 1.

For a weight matrix W, spectral normalization replaces it by

$$\bar{W} = \frac{W}{\sigma(W)}, \tag{4}$$

where $\sigma(W)$ is the spectral norm of W. In practice, $\sigma(W)$ is estimated efficiently with a few steps of the power iteration method during each update. If all layers have spectral norm at most 1 and we use 1-Lipschitz activations (LeakyReLU without batch normalization in the critic), then the whole critic becomes approximately 1-Lipschitz. This gives us a simple, parameter-free way to stabilize GAN training.
It leads to faster training but doesn't capture as much information as Gradient Penalty method. After tuning our parameters we didn't manage to out perform WGAN-GP. That's why, for the rest of our project, we stuck with our WGAN-GP model.

5 Gaussian Mixtures and Conditional GAN

Why extend WGAN-GP ? Even with the smoother gradients of WGAN-GP, the model still relies on a *unimodal* latent prior $z \sim \mathcal{N}(0, I)$. When the data distribution is *multi-modal* (digits, object classes, styles, poses), a single isotropic Gaussian is mismatched : its samples concentrate around a single latent region, forcing the generator to learn a multi-modal output distribution from a *single-mode* input distribution. This mismatch contributes to (i) residual mode under-coverage, (ii) unnecessary burden on the critic to separate distant modes, and (iii) slow convergence when modes are well-separated.

To address this, we enrich the latent space with a **Gaussian Mixture prior**, and we further guide generation using a **Conditional WGAN-GP** architecture. These two extensions are complementary : the GMM fixes the *input-side* multi-modality, while conditioning disentangles *semantic structure* during training.

5.1 Gaussian Mixture Latent Priors

Instead of sampling latent codes from a single Gaussian, we draw

$$z \sim p(z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z \mid \mu_k, \Sigma_k), \qquad \sum_k \pi_k = 1,\ \pi_k > 0.$$

Each mixture component captures a different region of latent space ; in practice, $\Sigma_k = \sigma^2 I$ usually suffices.

Intuition. The GMM prior introduces multiple *entry points* into the generator :

— each component can specialize in a mode of the data distribution (e.g. a digit class),

— gradients received by the generator are distributed across several latent basins rather than one, reducing collapse,

— the critic receives generated samples that cover the data geometry more uniformly, which stabilizes the Wasserstein estimate.

In short, a GMM aligns the latent geometry with the multi-modality of p_{data}, making the transport problem simpler for both G and the critic.

Effect on the WGAN objective. Since only the latent prior changes, the WGAN-GP critic still optimizes

$$\min_G \max_{f_\psi \in \text{Lip}(1)} \mathbb{E}_{x \sim p_{\text{data}}}\big[f_\psi(x)\big] - \mathbb{E}_{z \sim p(z)}\big[f_\psi(G(z))\big].$$

However, the pushforward distribution $p_g = G_\# p(z)$ becomes multi-modal by construction, bringing it closer (in Wasserstein distance) to the real data structure, which facilitates faster critic convergence.

5.2 Conditional WGAN-GP

The second extension is to make both generator and critic *conditional* :

$$G(z, y), \qquad f_\psi(x, y),$$

where y denotes class labels or structured attributes. Conditioning turns the WGAN objective into :

$$\min_G \max_{f_\psi \in \text{Lip}(1)} \mathbb{E}_{(x,y) \sim p_{\text{data}}}\big[f_\psi(x, y)\big] - \mathbb{E}_{z \sim p(z),\, y \sim p(y)}\big[f_\psi(G(z, y), y)\big] + \lambda\, \mathbb{E}_{\hat{x}}(\|\nabla_{\hat{x}} f_\psi(\hat{x}, y)\|_2 - 1)^2 .$$

Intuition. Conditioning simplifies the distribution-matching problem :

— the critic compares real and generated samples *within the same conditional slice* $p(\cdot \mid y)$, reducing the difficulty of estimating global Wasserstein distances,

— the generator no longer must disentangle semantic variation by itself, which accelerates training and mitigates mode collapse,

— conditioning often leads to sharper, more coherent samples because the critic enforces structure-aware constraints.

5.3 Combining GMMs and Conditioning

We use a GMM prior while also feeding the component index k (or an external label y) into the generator :

$$z \sim \sum_k \pi_k \mathcal{N}(z \mid \mu_k, \Sigma_k), \qquad y \in \{1, \ldots, K\}.$$

Two types of structure emerge :

— **Implicit multi-modality** from the Gaussian components, improving coverage of latent modes ;
— **Explicit semantic structure** from conditioning, reducing the complexity of the Wasserstein critic's decision boundary.

Overall, the combination reduces mode collapse, stabilizes critic training, and improves visual quality. In practice, we found that both extensions produce more diverse samples and more stable critic gradients, while remaining fully compatible with our WGAN-GP implementation.

6 Discriminator Rejection Sampling

We integrated Discriminator Rejection Sampling (DRS) into the Wasserstein GAN (W-GAN) in order to improve sample quality by leveraging the discriminator's density-ratio information. DRS, introduced by Azadi et al., interprets the generator as a proposal distribution $p_g(x)$ and uses the discriminator to approximate the likelihood ratio $p_d(x)/p_g(x)$. Under the idealized GAN setting with a sigmoid discriminator trained via cross-entropy, the optimal logit $\tilde{D}^{(x)}$ satisfies

$$\frac{p_d(x)}{p_g(x)} = e^{\tilde{D}^*(x)}.$$

This allows rejection sampling with acceptance probability proportional to $\exp(\tilde{D}^*(x) - \tilde{D}^*_M)$, where $\tilde{D}^*_M$ denotes the maximum logit over the support.

Since W-GAN uses a critic optimized under the Wasserstein objective rather than a probabilistic discriminator, we first calibrated the critic. After training the W-GAN normally, we froze the critic and added a small sigmoid output layer trained with a binary cross-entropy loss to distinguish real and generated data. This calibrated discriminator provides logits $\tilde{D}(x)$ that approximate the theoretical density-ratio logit required for DRS.

We then implemented the practical DRS acceptance procedure. We first estimated $\tilde{D}_M = \max_x \tilde{D}(x)$. For any generated sample x, we computed

$$\hat{F}(x) = \tilde{D}(x) - \tilde{D}_M - \log\Big(1 - e^{\tilde{D}(x) - \tilde{D}_M - \varepsilon}\Big),$$

and defined the acceptance probability as $\sigma(\hat{F}(x))$. The constant ε ensures numerical stability. Since we were not able to reliably tune the parameters in practice, we chose to accept approximately 20% of the generated images.

This integration leaves the W-GAN training unchanged while providing a principled post-processing mechanism : generated samples are filtered according to how well the calibrated discriminator estimates they match the data distribution. The resulting sample set is therefore closer to the true target distribution and displays higher visual and statistical fidelity.

7 Conclusion

We started from the limitations of Vanilla GANs (mode collapse, vanishing gradients, and unstable training) and progressively transitioned to more robust frameworks based on the Wasserstein distance. WGAN with weight clipping offered initial stability improvements, but its sensitivity to the clipping parameter motivated the adoption of WGAN-GP. The gradient penalty yielded smoother critic landscapes, more reliable gradients, and overall better mode coverage. We further investigated spectral normalization, which provided faster training but ultimately underperformed compared to the gradient penalty approach. To better align the latent space with the multimodal nature of the data, we introduced Gaussian Mixture priors and incorporated conditional generation. These extensions helped structure the latent space, reduce mode collapse, and enhance sample coherence. Finally, we implemented Discriminator Rejection Sampling as a post-processing step to refine the generator's outputs. By leveraging density-ratio estimates from a

calibrated discriminator, DRS allowed us to filter low-quality samples and significantly improve both accuracy and recall. Overall, the experimental results confirm the benefits of these methodological advances : WGAN-GP, combined with conditioning and DRS, achieved the strongest performance across all metrics, demonstrating superior generation quality, broader mode coverage, and enhanced fidelity to the data distribution. This progression highlights how principled modifications to the training objective, latent structure, and sampling procedure can substantially improve generative modelling performance.

model	time(s)	FID	accuracy	recall
VanGAN	-	-	0.52	0.23
WGAN-WC	-	-	0.5	0.27
WGAN-GP	77	45	0.53	0.29
WGAN-SN (DRS)	105	52	0.5	0.44
WGAN-GP (DRS)	240	62	0.67	0.62

Table 1 – Results

References

1. Arjovsky M., Chintala S., & Bottou L. *Wasserstein GAN*. Courant Institute of Mathematical Sciences & Facebook AI Research.
2. Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., & Courville A. *Improved Training of Wasserstein GANs*. Montreal Institute for Learning Algorithms (MILA), Courant Institute of Mathematical Sciences, CIFAR Fellow.
3. Mirza M., & Osindero S. *Conditional Generative Adversarial Nets*. Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, QC H3C 3J7, 2014.
4. Ben-Yosef M., & Weinshall D. *Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images*. School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel, 2018.
5. Azadi S., Olsson C., Darrell T., Goodfellow I., & Odena A. *Discriminator Rejection Sampling*. UC Berkeley & Google Brain.