# Latent Reweighting for Generative Adversarial Networks

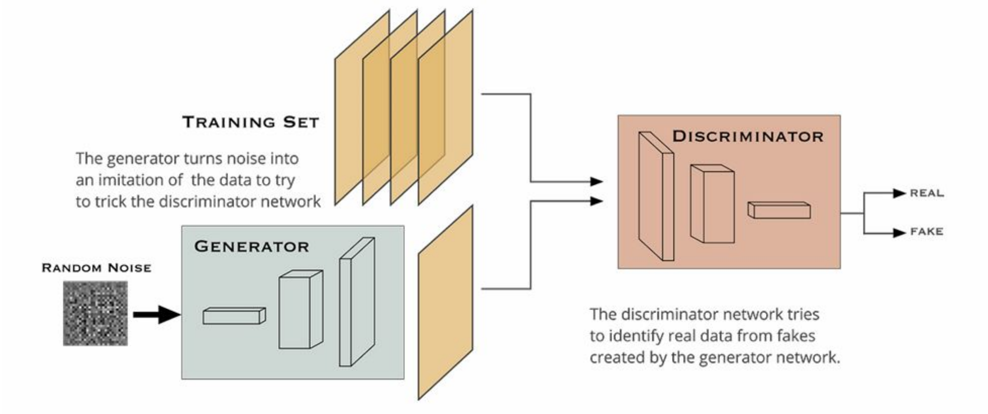
## DisorGANized

**November 13, 2024**

Nabil Arbel, Barthélémy Brégeon, Kenta Vert

PSL UNIVERSITÉ PARIS

# Vanilla GAN


Training Set
The generator turns noise into an imitation of the data to try to trick the discriminator network
Random Noise
Generator
Discriminator
Real
Fake
The discriminator network tries to identify real data from fakes created by the generator network.

# Vanilla GAN

- Running different multiple vanilla models tuning hyperparameters
- Final Vanilla GAN Model :
  - Epochs = 150
  - Learning rate = 2e-4
  - Batch size =  64
- Yielded good results (close to last year's 2nd best result)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| BestOf2023-2 | profs | coktailjet | Success | 92.61 | 36.33 | 0.52 | 0.27 |
| disorganized | Master-IASD | coktailjet | Success | 110.82 | 38.94 | 0.54 | 0.22 |

# Latent reweighting

# Limits of GANS

- Problem when the latent distribution is not continuous
- No GAN's land on disconnected manifold

(a) Synthetic WGAN: real samples in green and fake ones in blue.

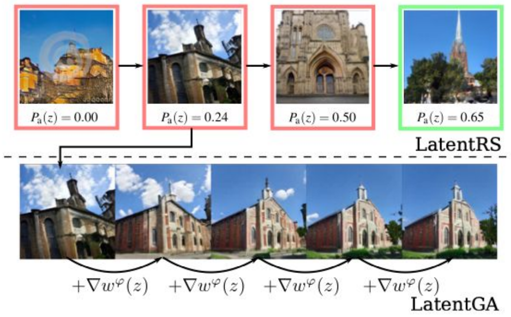
# Choosing a better sample

## Latent reweighting, an almost free improvement for GANs

Thibaut Issenhuth[1,2], Ugo Tanielian[1], David Picard [2], Jérémie Mary[1]
[1] Criteo AI Lab, Paris, France.
[2] LIGM, Ecole des Ponts, Marne-la-Vallée, France.
{thibaut.issenhuth,ugo.tanielian,jeremie.mary}@criteo.com,david.picard@enpc.fr

**Abstract**

*Standard formulations of GANs, where a continuous function deforms a connected latent space, have been shown to be misspecified when fitting different classes of images. In particular, the generator will necessarily sample some low-quality images in between the classes. Rather than modifying the architecture, a line of works aims at improving the sampling quality from pre-trained generators at the expense of increased computational cost. Building on this, we introduce an additional network to predict latent importance weights and two associated sampling methods to avoid the*



# Choosing a better sample

- Using pretrained-GAN (vanilla)
- Train additional network to give **importance weights** to latent vectors (higher value for good images and lower for bad images)
- Train it adversarially with the discriminator using a Wasserstein critic

# Latent Rejection Sampling

# LatentRS

**Algorithm 2:** LatentRS

**Requires:** Prior Z, Gen. $G_\theta$, Importance weight network $w^\varphi$, maximum importance weight $m$;

**while** *True* **do**

  Sample $z \sim Z$ ;

  Sample $\alpha \sim \text{Uniform}[0, 1]$ ;

  **if** $\frac{w^\varphi(z)}{m} \geq \alpha$ **then**

    break;

  **end**

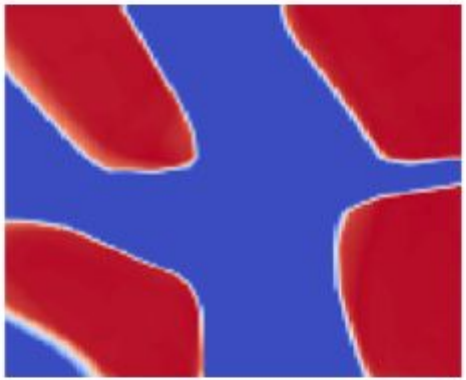
**end**

$x \leftarrow G_\theta(z)$;

**Result:** Selected point x

# LatentRS

**(c)** Optimizing for Wasserstein criterion with latentRS (ours $\star$ ).



**(d)** Heatmap of the $w^{\varphi}$ in the latent space (in the blue areas, $w^{\varphi}$=0).

# Latent Gradient Ascent

# LatentGA

- Gradient Ascent to maximize the importance weights in the latent space
- Number of steps and learning rates are hyperparamameters (we used the same as the paper)
- More expensive than RS

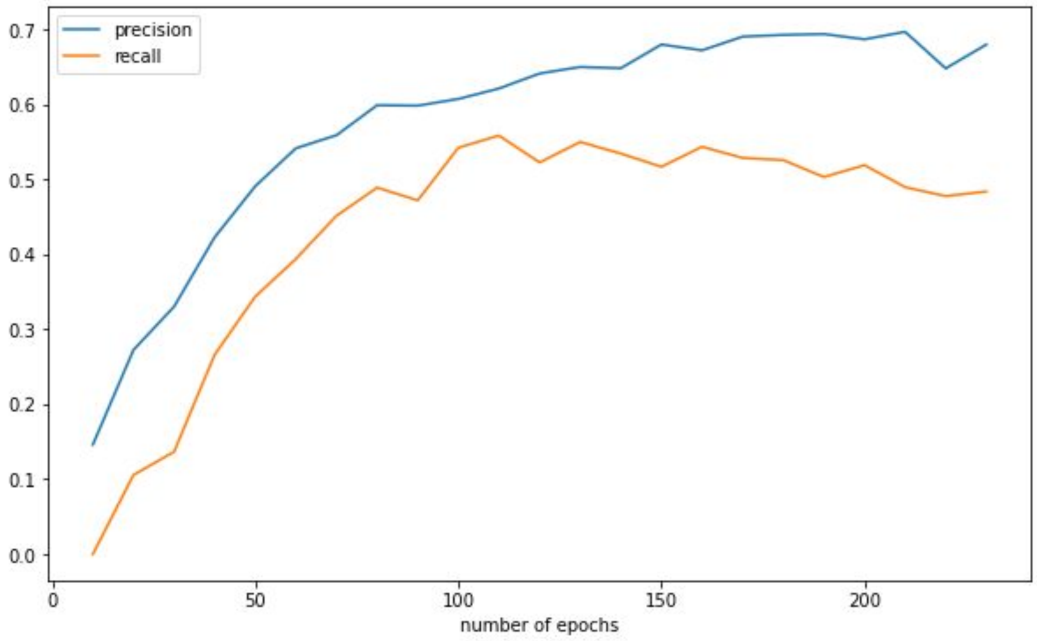
# Latent RS+GA

# LatentRS+GA

- Rejection sampling, then Gradient ascent
- Less GA step
- Theoretically better results

# Results

- Lots of training needed
- Limited diversity
- Sampling focusing on specific latent space region



# Results

- Noticing some improvement compared Vanilla GAN
- Three latent reweighting methods providing similar results
- Bigger impact on WGAN

| Method | FID | Precision | Recall |
| --- | --- | --- | --- |
| Rejection Sampling (RS) | 33.61 | 0.50 | 0.21 |
| Gradient Ascent (GA) | 32.52 | 0.52 | 0.20 |
| Rejection Sampling + Gradient Ascent (RS+GA) | 32.81 | 0.50 | 0.21 |

*Final results obtained in the platform*

# Formulas

- New distribution defined by w

$$for \quad all \quad z \in \mathbb{R}^d, d\gamma^{\phi}(z) = w^{\phi}(z)d\gamma(z)$$

- Optimizing scheme

$$\inf_{\phi\in\Phi} \sup_{\alpha\in\Lambda} \mathbb{E}\, D_\alpha(x) - \mathbb{E}\, w_\phi(z) D_\alpha(G(z))$$

- Optimizing scheme for w with regularization

$$\sup_{\phi\in\Phi} \mathbb{E}\, w_\phi(z)(D_\alpha(G(z)) - \Delta) - \lambda_1(\mathbb{E}\, w_\phi(z) - 1)^2 - \lambda_2 \mathbb{E}\, \max(0, (w_\phi(z) - m))^2$$

$$\Delta = \min_{z\sim Z} D(G(z))$$