RCP211 – Modèles génératifs

Réseaux génératifs antagonistes avancés

Arnaud Breloy arnaud.breloy@lecnam.net

16 octobre 2024

Conservatoire national des arts & métiers

Table of contents

1. Rappels

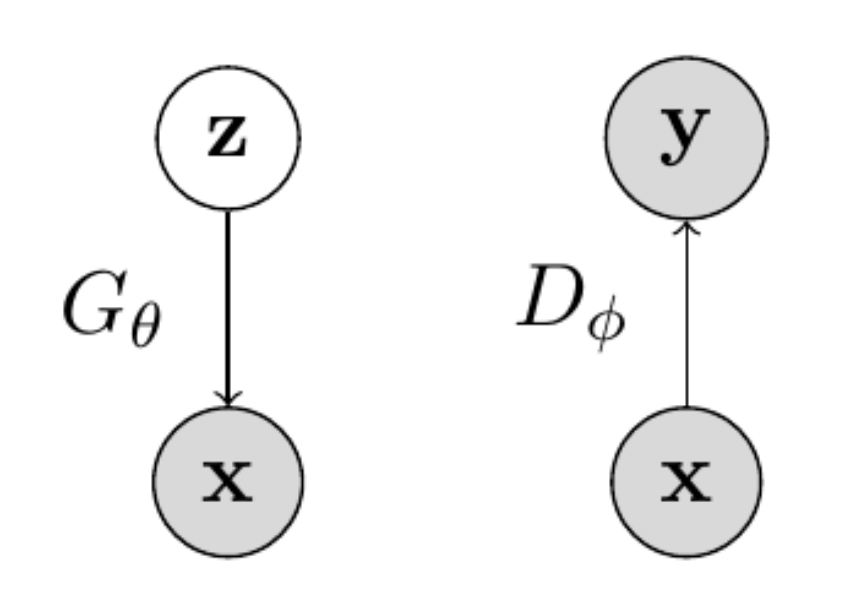
2. Extensions des GAN conditionnels pour les images

3. GAN pour la synthèse d’images à haute résolution

4. Espace latent et contrôle des GAN

Rappels

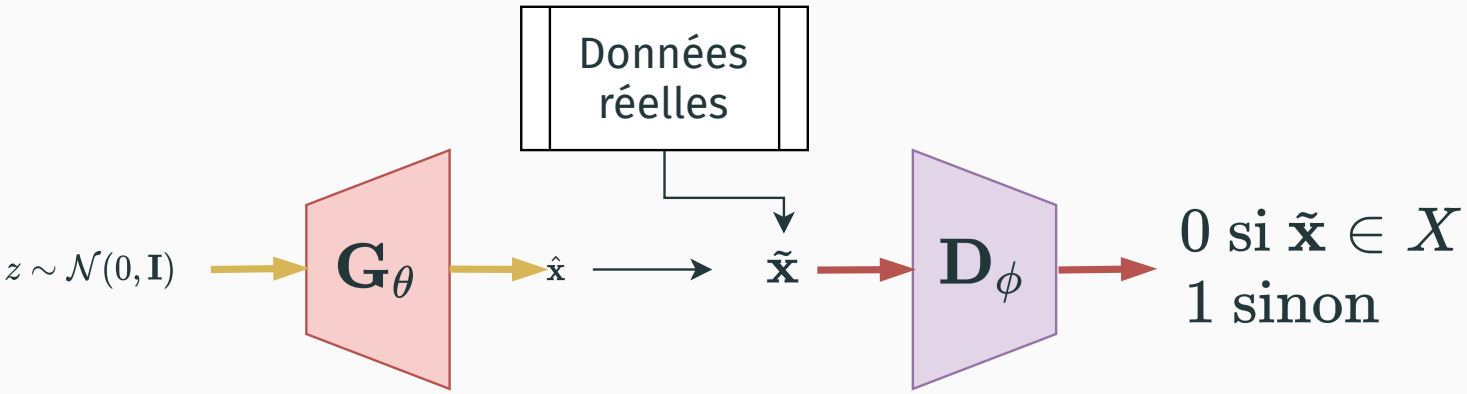
Principe des GAN



Jeu minimax en opposition entre G et D :

- $\max_\phi \mathbb{E}_{\mathbf{x}\sim \mathbf{p}_{\text{data}}}[\log D_\phi(\mathbf{x})] + \mathbb{E}_{\hat{\mathbf{x}}\sim p(\hat{\mathbf{x}}}[\log(1-\hat{\mathbf{x}})]$: trouver les paramètres de D qui séparent les données réelles des données synthétiques.
- $\min_\theta \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - D_\phi(G_\theta(\mathbf{z}))\right)\right]$: trouver les paramètres de G qui produisent des données qui trompent D.

Schéma récapitulatif



Extensions des GAN conditionnels pour les images

GAN conditionnel convolutif

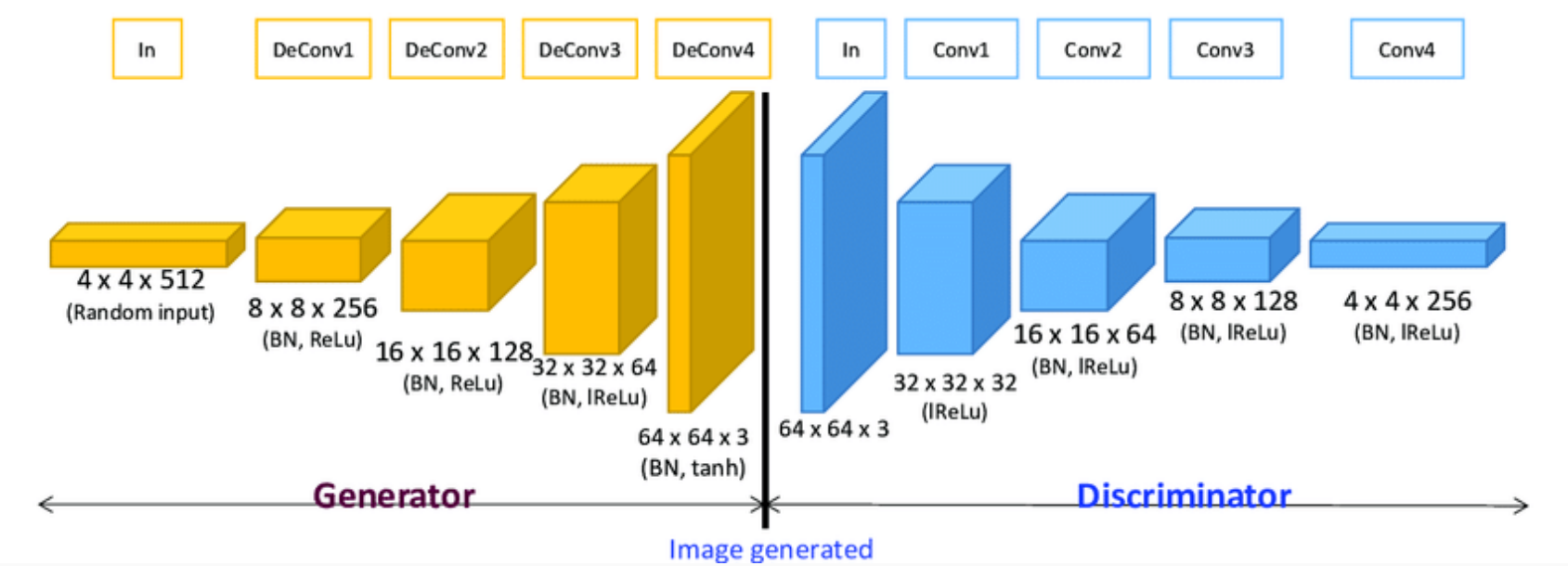


Schéma extrait de Suh et al., 2019

Formule du GAN conditionnel

Jeu minimax à deux joueurs, conditionné sur une variable aléatoire y :

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{y})))]$$

GAN conditionnel convolutif

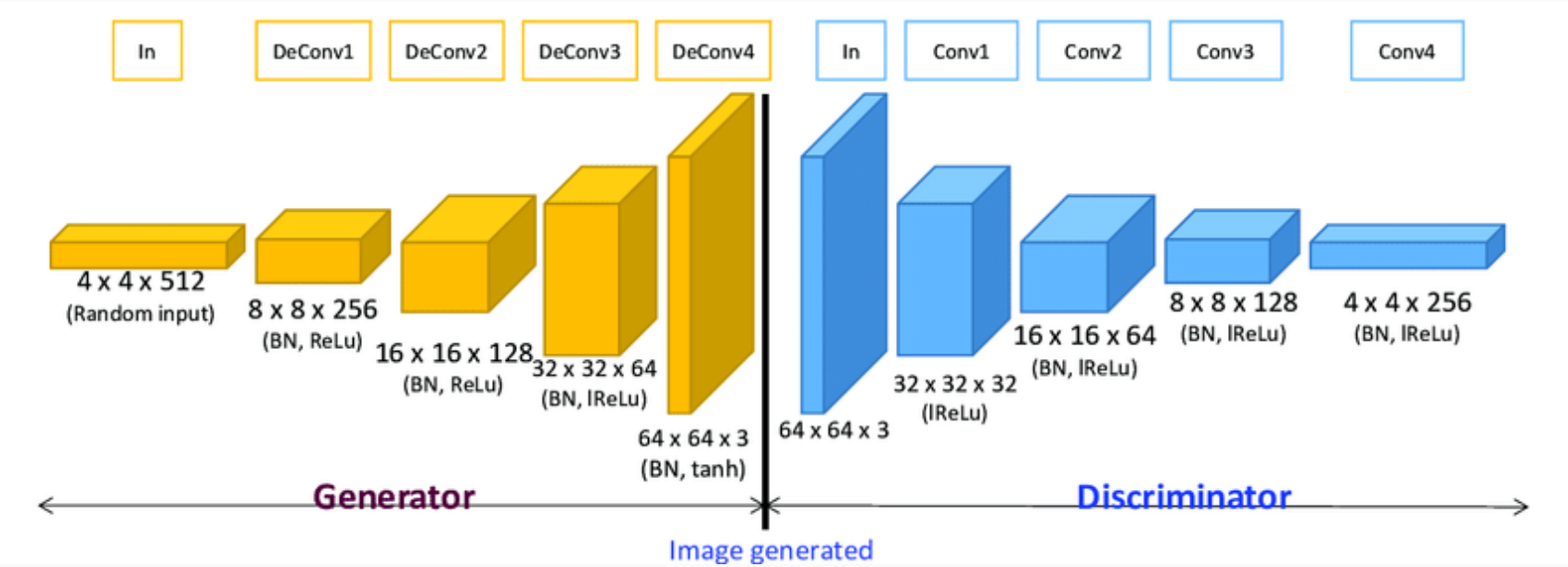


Schéma extrait de Suh et al., 2019

Problème

Le discriminateur renvoie une unique réponse « réel » ou « faux » pour toute l'image.

$\rightarrow$ le GAN ne capture que la structure globale de l'image

PatchGAN

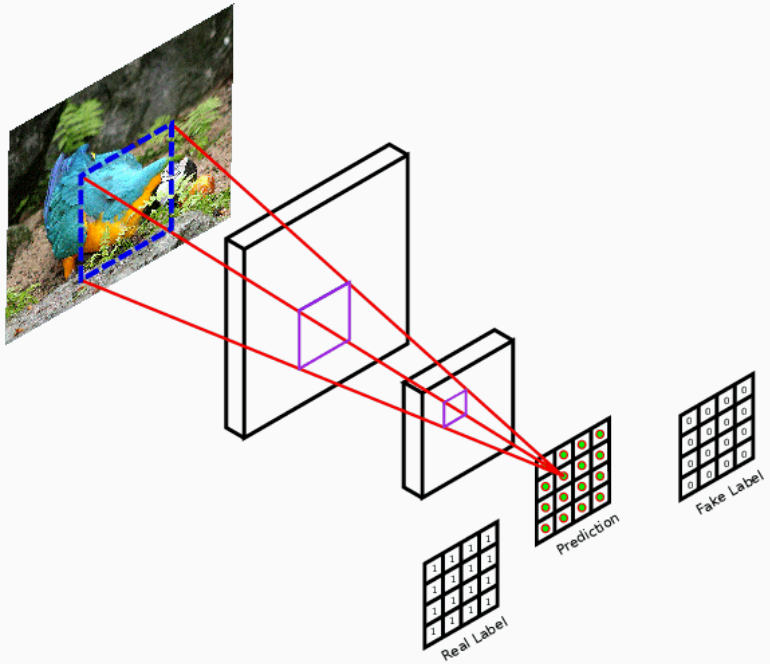


Schéma extrait de Demir et Unal, 2018

PatchGAN produit une **carte** de prédictions « réel/faux ». Chaque pixel de la carte de sortie correspond à une région de l'image d'entrée du discriminateur.

PatchGAN

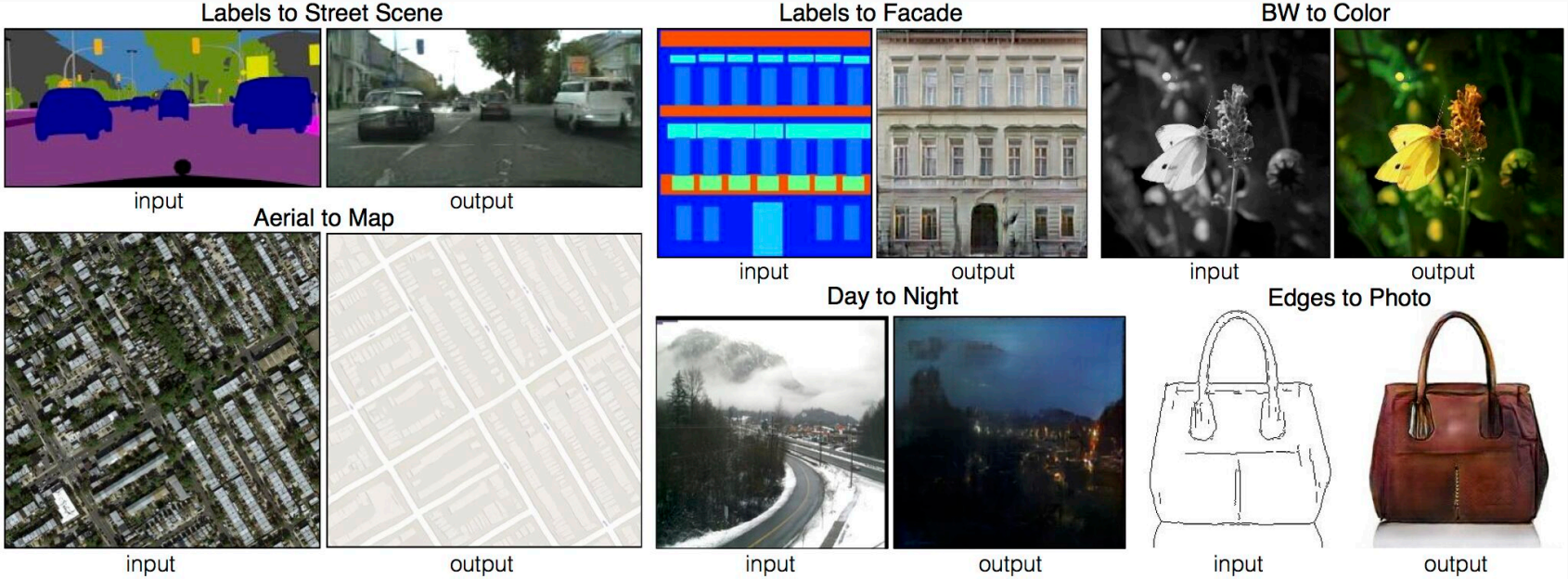
PatchGAN est une architecture de discriminateur proposée par Isola et al. (2017) dans l'article pix2pix.

Principe : on ne conserve que la partie convolutive du discriminateur.

$\rightarrow$ le discriminateur agrège des décisions locales

$\rightarrow$ D optimise la moyenne des entropies croisées sur toutes les prédictions (i, j)

pix2pix : principe



pix2pix

Isola et al., Image-to-Image Translation with Conditional Adversarial Nets, 2017

Principe : convertir des images d'un domaine A vers un domaine B à partir de paires existantes.

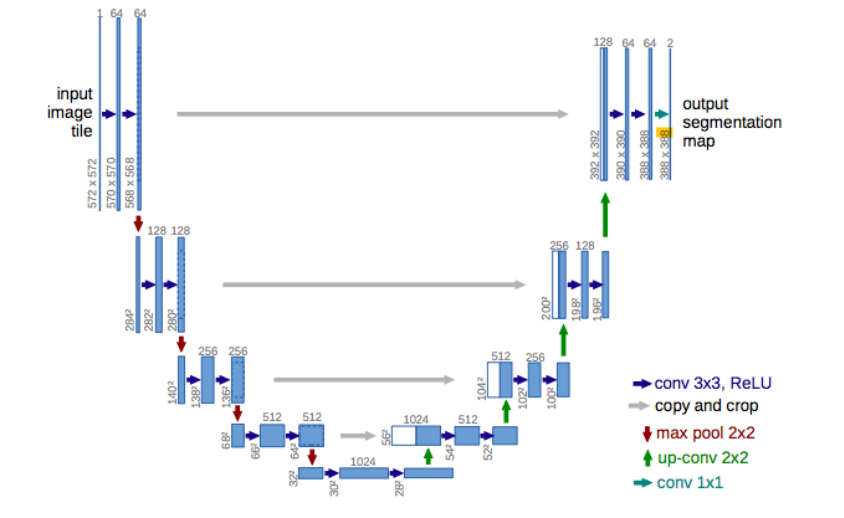
Architecture : DCGAN conditionné à *une image* avec un discriminateur PatchGAN.

pix2pix : architecture

Encodeur

Modèle entièrement convolutif : U-Net

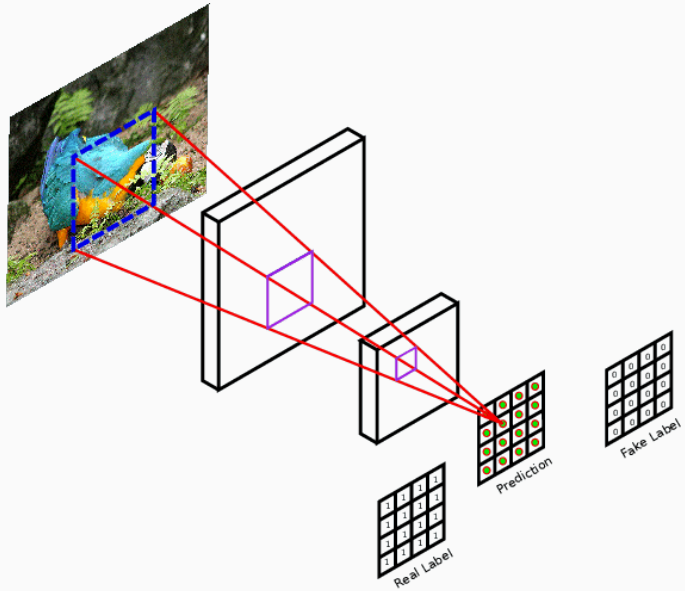
Entrée : image du domaine A
Sortie : image du domaine B



Discriminateur

PatchGAN

Entrée : image (réelle ou fausse)
Sortie : une prédiction par région



pix2pix : optimisation

Optimisation

pix2pix est une méthode **supervisée** : il est nécessaire d'avoir des paires d'images. On optimise :

$$\min_G \max_D \mathcal{L}_{\text{GAN}} + \lambda \mathcal{L}_{\text{data}}(G)$$

avec une double fonction de coût :

- Attache aux données : $\mathcal{L}_{\text{data}}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$
- GAN : $\mathcal{L}_{\text{GAN}}(G,D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,z))]$

avec x les images source, y les images cible et z un bruit.

Interprétation : la perte L_1 capte les basses fréquences (= structure globale) tandis que PatchGAN intègre les hautes fréquences (= détails). Le GAN agit comme une **régularisation perceptuelle**.

CycleGAN

Peut-on réaliser la « traduction » d'image sans paires pour la supervision ?

CycleGAN

Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2018

Extension non supervisée de pix2pix

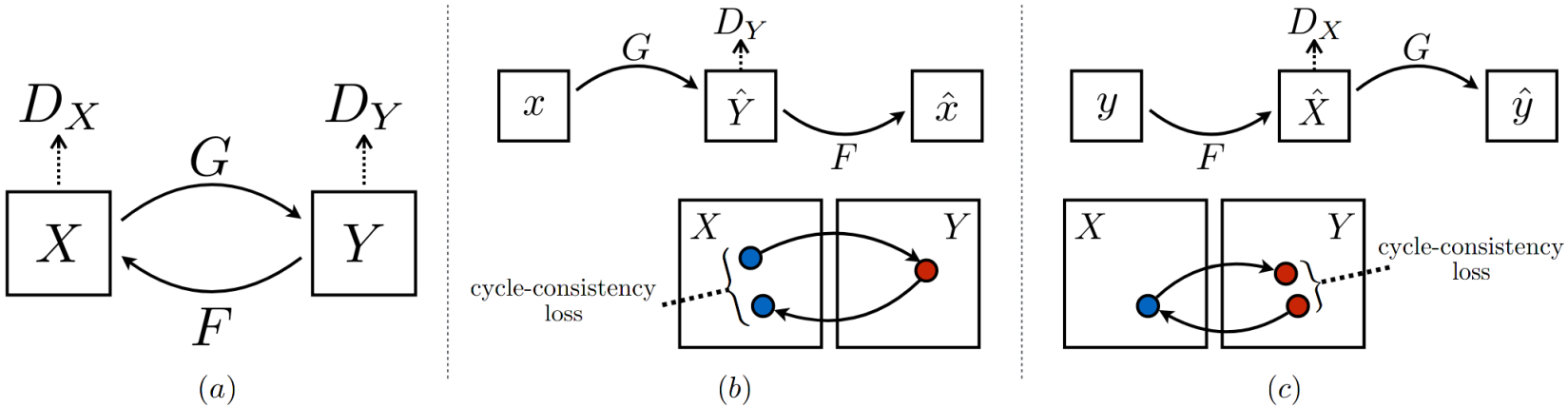
Principe : pour apprendre la transformation $G : X \to Y$ sans paires d'exemples, on apprend aussi la transformation inverse $F : Y \to X$.

On souhaite avoir une idempotence du cycle $F \circ G$ (et $G \circ F$) :

- $x \to G(x) \to F(G(x)) \approx x$ (cohérence *avant*),
- $y \to F(y) \to G(F(y)) \approx y$ (cohérence *arrière*).

CycleGAN : fonctionnement

On dispose d'images $x \in X$ et $y \in Y$. Il n'y a *pas de correspondances* entre les images.



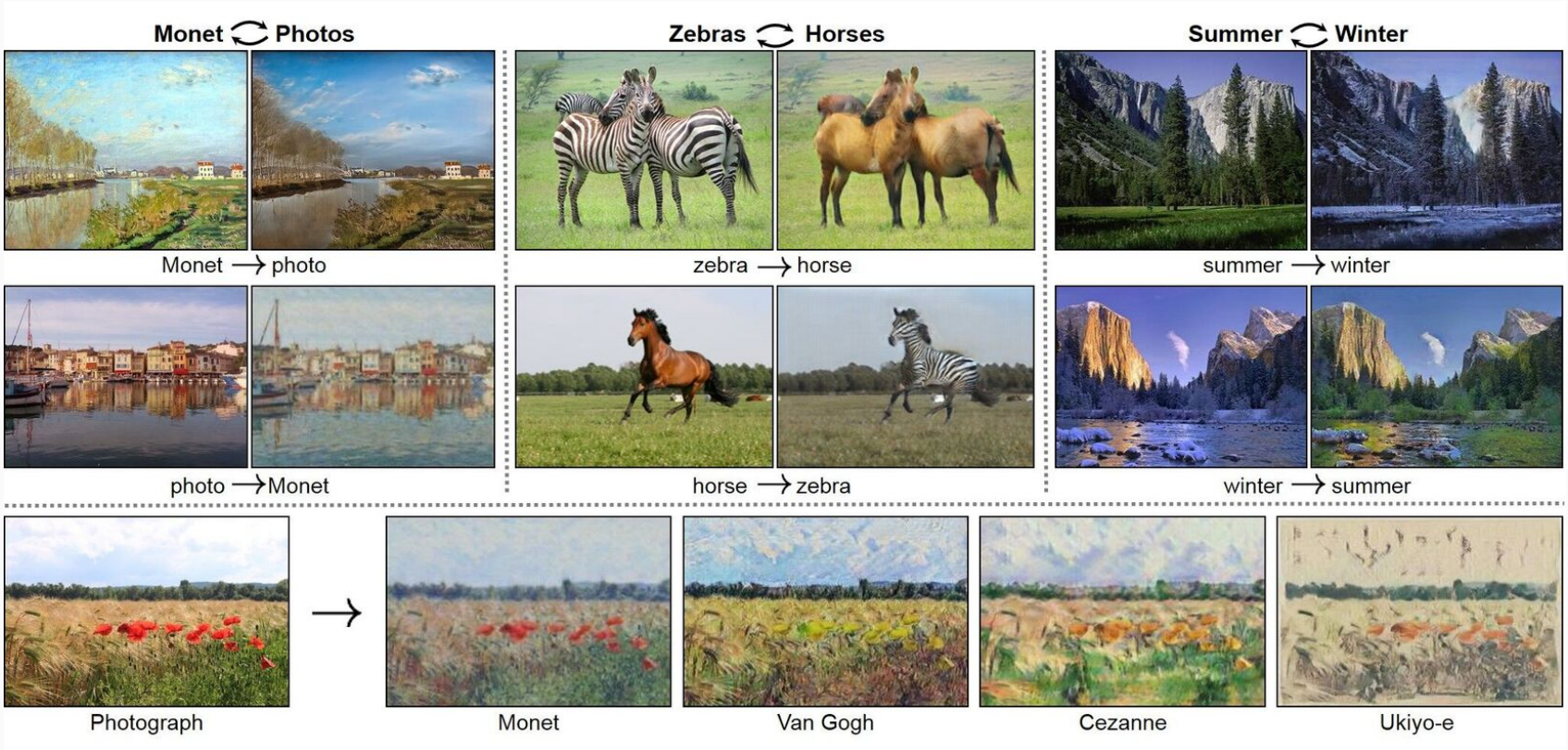
Le modèle apprend $G : X \to Y$ et $F : Y \to X$ par deux GAN :

$$L_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p(x)}[\log(1 - D_Y(G(x)))]$$

et impose une contrainte de **cohérence cyclique** :

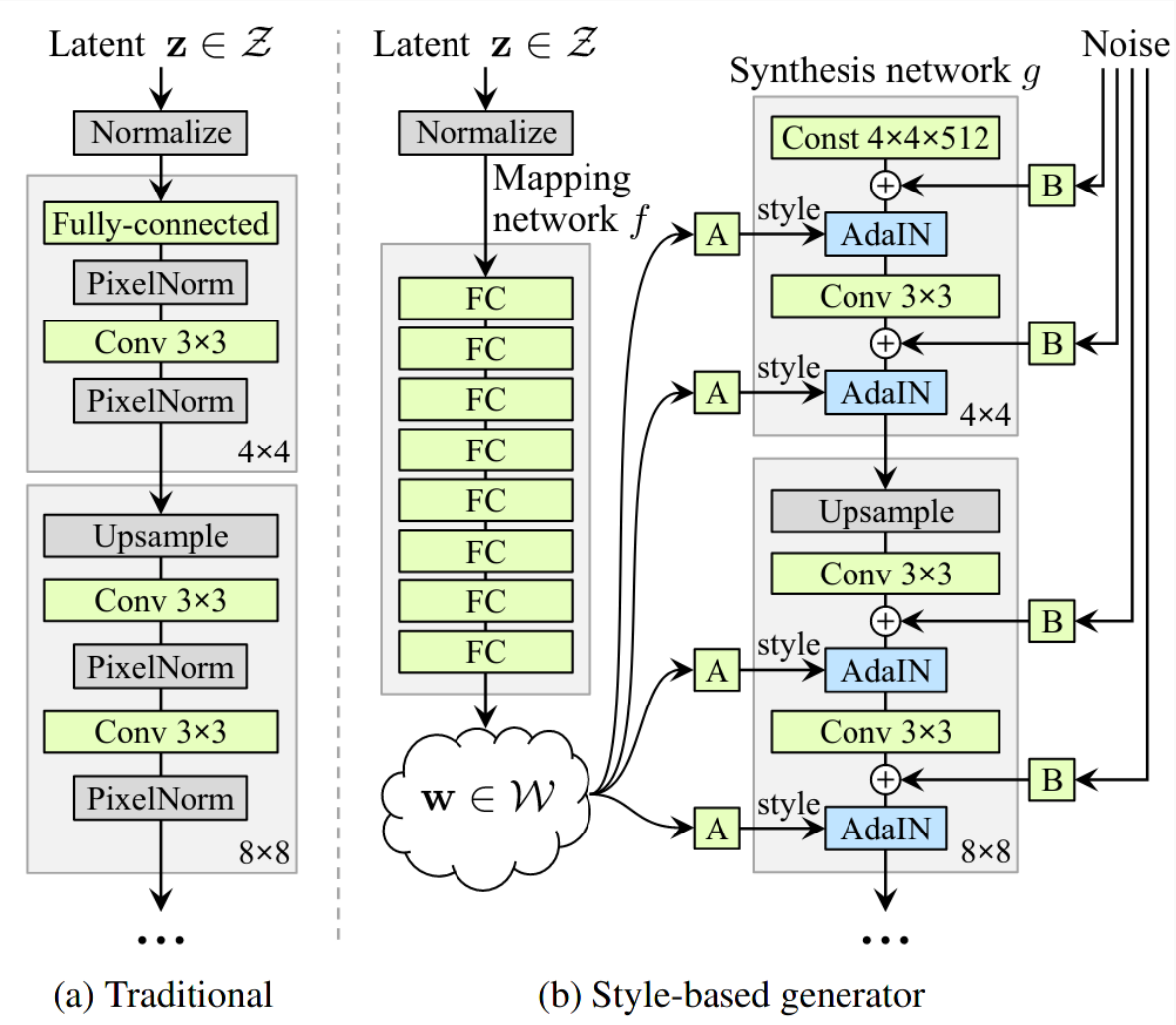
$$L_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p(x)}[\| F(G(x)) - x \|_1] + \mathbb{E}_{y \sim p(y)}[\| G(F(y)) - y \|_1]$$

CycleGAN : résultats



GAN pour la synthèse d’images à haute résolution

StyleGAN

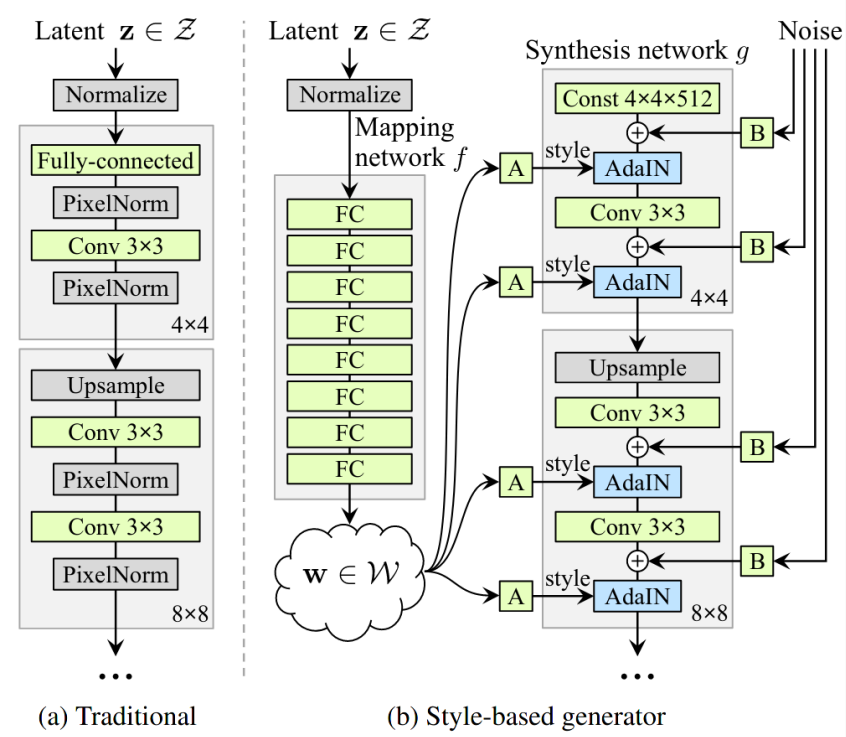


Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, 2019

Principe : version « améliorée » de PGGAN, le code $z \in \mathcal{Z}$ est transformé en style $w \in \mathcal{W}$. Ce style est injecté dans les différentes couches du générateur.

StyleGAN : astuces

Adaptive Instance Normalization (AdaIN)



AdaIN

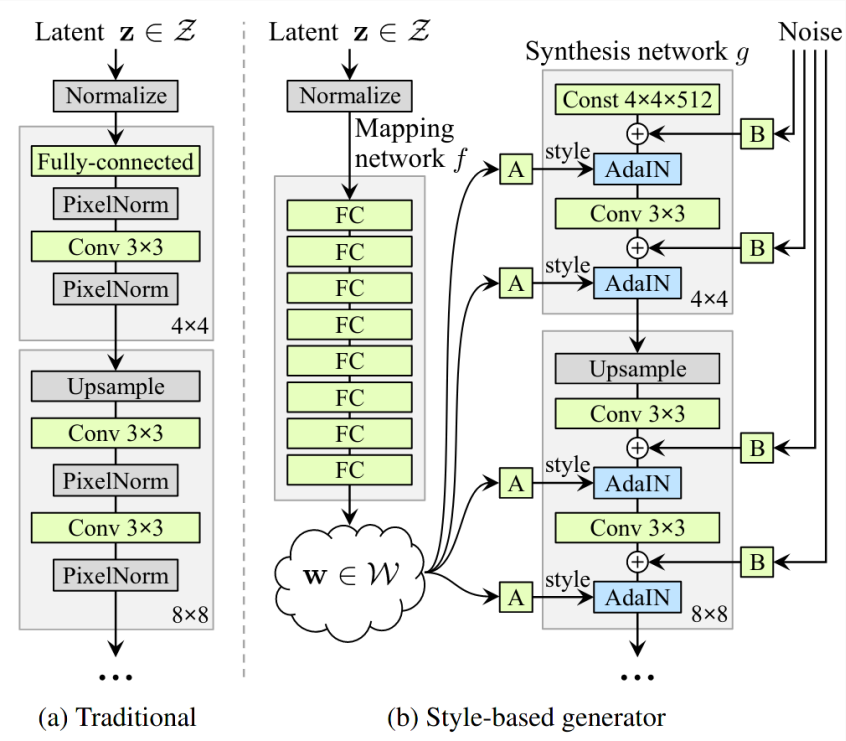
w est décomposé par une transformation affine en $\mathbf{y} = (\mathbf{y_s}, \mathbf{y_b})$.

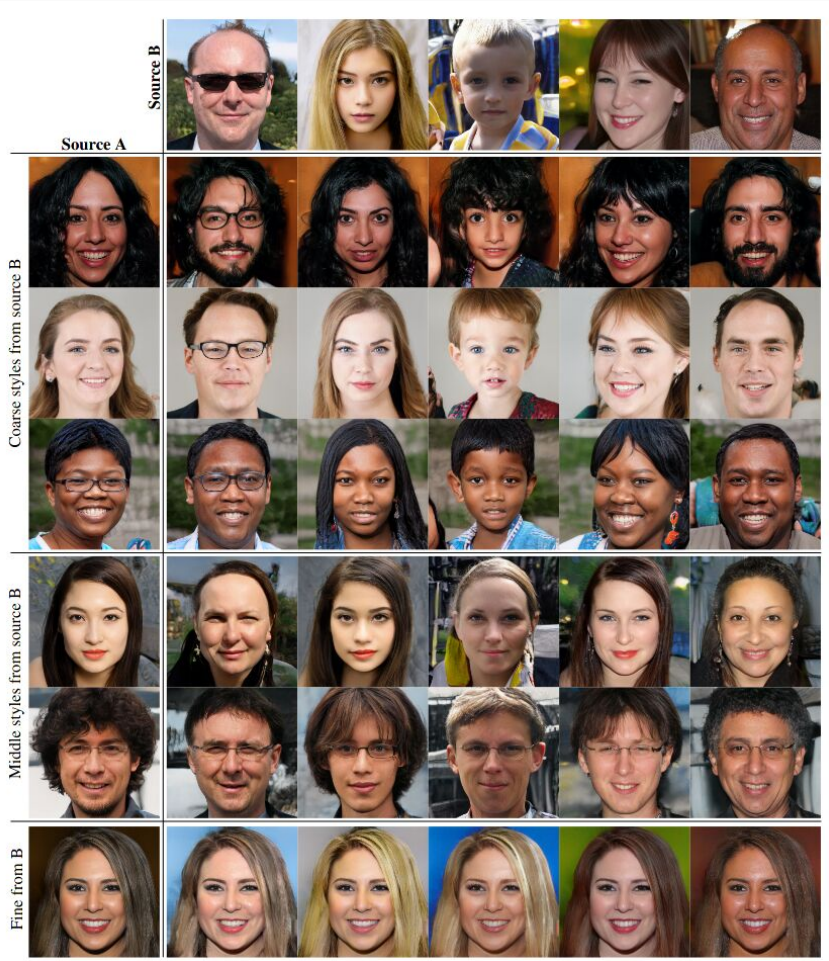
Le style permet ensuite de contrôler chaque carte de caractéristiques après une convolution dans l'AdaIN :

$$\mathrm{AIN}(\mathbf{x_i}, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x_i} - \mu((x_i)}{\sigma(\mathbf{x_i})} + \mathbf{y}_{b,i}$$

StyleGAN : astuces

Style mixing : on remplace aléatoirement certains styles w par le style w' d'une autre image.





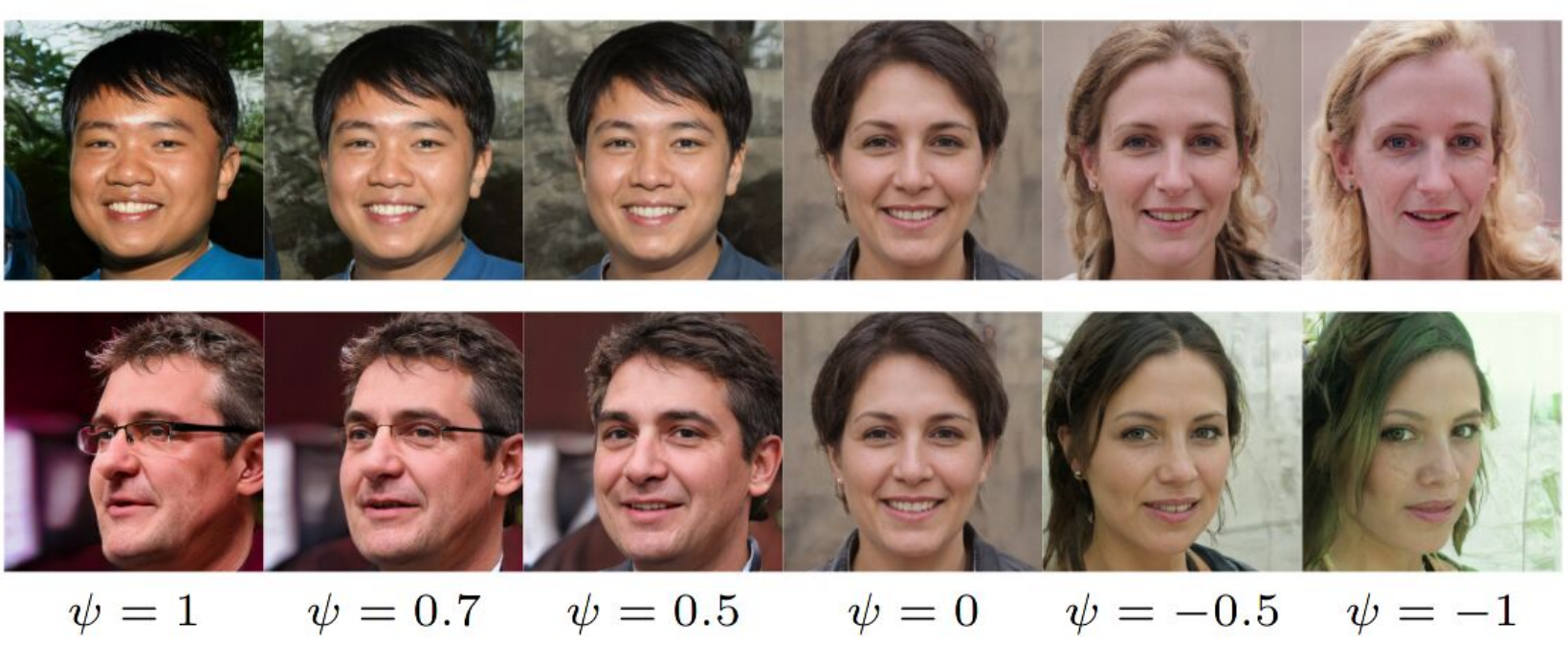
StyleGAN : troncature

Pour simplifier l'échantillonnage, on considère une distribution tronquée :

$$w' = \bar{w} + \psi\,(w - \bar{w})$$

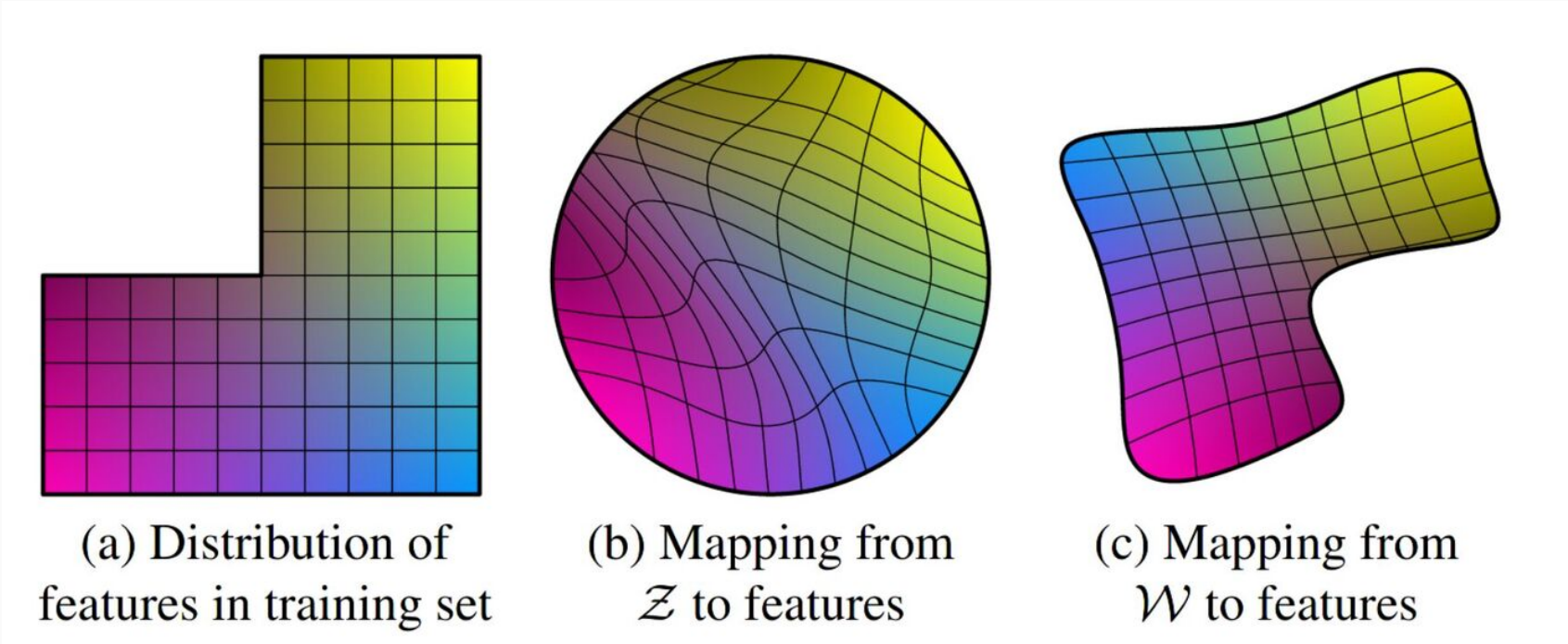
où $\bar{w} = \mathbb{E}_{z \sim p(z)}[f((z)]$ est le centre de masse de $\mathcal{W}$.

$G(\bar{w})$ est l'image « moyenne » et s'en éloigner dans une direction ou son opposée « inverse » une image :



StyleGAN : espace des styles

L'espace de style $\mathcal{W}$ permet d'obtenir un espace latent structuré différemment de l'a priori gaussien de $\mathcal{Z}$. Le style correspond à un *embedding* que l'on espère plus interprétable.

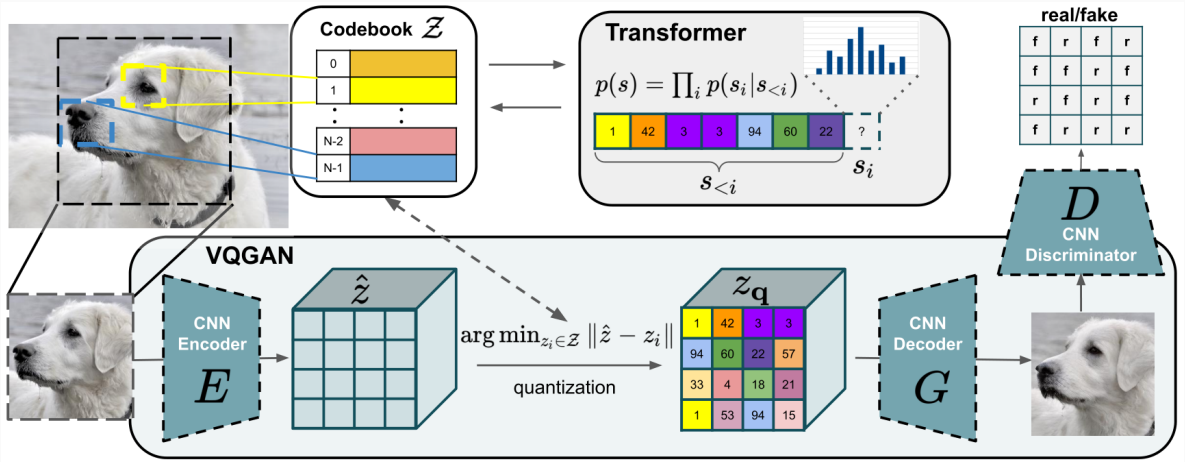


(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

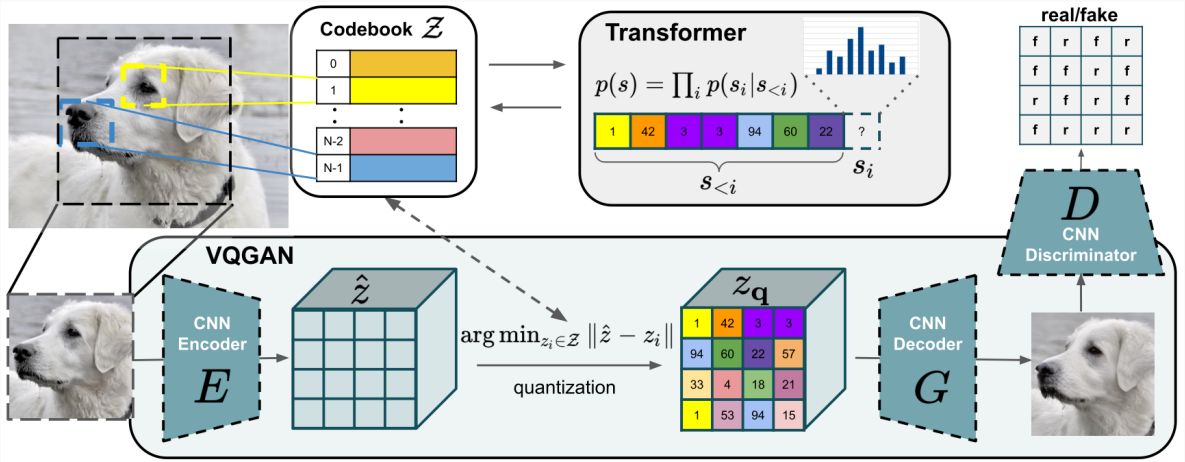
VQ-GAN



Motivation

- La qualité des images générées par les GAN est supérieure à celle des images générées par un VAE.
- Mais les VAE permettent d'encoder une image et de mieux structurer l'espace latent.
- Peut-on cumuler les deux avantages ?

VQ-GAN



VQ-GAN

- VQ-VAE : encodeur convolutif $\mathcal{E}$ + décodeur convolutif $\mathcal{D}$
- GAN : générateur G + discriminateur D
- Et si on optimisait le décodeur $\mathcal{D}$ comme le générateur d'un GAN ?

$$\mathcal{L}_{\text{VQ-GAN}}(E, G, D, \mathcal{Z}) = \mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) + \mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D)$$

$p(z_q)$ est modélisé par un modèle autorégressif : Transformer (cf. RCP217).

Espace latent et contrôle des GAN

Espace latent

L'espace latent des modèles génératifs permet de parcourir la distribution des données synthétiques.

Mais les directions de l'espace latent n'ont en général pas de signification particulière. Les attributs sémantiques d'une image sont mélangés : on parle d'*entanglement*.

Comment explorer cet espace latent pour :

- comprendre la structure des données ?
- interpréter sémantiquement les variations statistiques latentes ?
- modifier une donnée réelle de façon plausible ?

Manipulation d'image

Pour une image générée $G(z) = \hat{x}$, comment déplacer z de sorte à produire un effet donné sur $\hat{x}$ (*e.g.* pose, expression, couleur, etc.) ?

Problématique
Comment trouver des **directions sémantiques** dans l'espace latent ?

Idée naïve : arithmétique des vecteurs dans l'espace latent

« homme avec lunettes » - « homme sans lunettes » + « femme sans lunettes » = « femme avec lunettes »

InterFaceGAN

Prérequis : un classifieur C capable d'identifier un attribut binaire $y \in \{0, 1\}$.

Objectif : trouver une direction $d \in \mathcal{Z}$ qui permet de contrôler la présence de l'attribut y dans les images générées $G(z)$.

Shen et al., Interpreting the latent space of GANs for semantic face editing, 2020

Algorithme

1. Échantillonner N codes latents
2. Générer les N images $\hat{x}_i$ correspondantes
3. Obtenir les pseudo-étiquettes $\hat{y}_i = C(G(z_i))$ grâce au classifieur
4. Conserver les n codes latents z_i^+ (resp. z_i^-) qui ont produit les images qui obtiennent les scores les plus élevés (resp. les plus faibles)
5. Trouver la direction d comme la normale à l'hyperplan séparateur d'une SVM linéaire qui séparent $\mathcal{Z}^+$ et $\mathcal{Z}^-$

GANSpace

Härkönen et al., Discovering Interpretable GAN Controls, 2020

Principe

GANSpace exploite la structure de $\mathcal{W}$ et applique une ACP dans l'espace latent de StyleGAN.

La découverte des directions est non-supervisée mais leur sémantique est interprétée *a posteriori*.

Édition d'image

Modifier une image existante

Pour modifier une image réelle, il est nécessaire de connaître son équivalent z dans l'espace latent : c'est le problème de l'**inversion d'image**.

Comparaison GAN/VAE

Un auto-encodeur (variationnel) possède un encodeur qui réalise une projection $\mathcal{X} \to \mathcal{Z}$. Ce n'est pas le cas des GAN.

Recherche par descente de gradient

Une solution coûteuse pour retrouver z^* qui génère une image cible $\mathbf{x}$ consiste à minimiser :

$$\min_{z \in \mathcal{Z}} \| G(z) - \mathbf{x} \|_p$$

par descente de gradient.

Apprentissage d'un encodeur pour l'inversion

Une solution moins coûteuse que la descente de gradient consiste à apprendre un réseau de neurones P paramétré par θ qui réalise la projection $\mathcal{X} \to \mathcal{Z}$.

Algorithme

1. Échantillonner n codes z_i et les images $G(z_i) = x_i$ correspondantes
2. Apprendre P par descente de gradient :
   $\theta_P^* = \arg\min_\theta \sum_{i=1}^n \mathcal{L}(P(G(z_i;\theta)), z_i)$
3. Pour une image x', le code associé est $z' = P(x')$.

Une approche rapide et performante consiste à utiliser $z = P(x)$ comme initialisation pour une recherche par descente de gradient. (Zhu et al., 2016)