THÈSE

Pour obtenir le grade de Docteur de Normandie Université

Spécialité Informatique

l'École Doctorale Mathématiques, Information, Ingéniérie des Systèmes

Auxiliary Tasks for the Conditioning of Generative Adversarial Networks

Tâches auxilliaires pour le conditionnement des réseaux antagonistes génératifs

Présentée et soutenue par

Cyprien RUFFINO

Dirigée par Gilles GASSO et Romain HÉRAULT

Thèse soutenue publiquement le Thursday 3rd September, 2020
devant le jury composé de

Civilité / prénom NOM,	Grade / fonction / statut / lieu d'exercice	Rapporteur ou examinateur ou directeur de thèse ou codirecteur de thèse
Civilité / prénom NOM,	Grade / fonction / statut / lieu d'exercice	Rapporteur ou examinateur ou directeur de thèse ou codirecteur de thèse
Civilité / prénom NOM,	Grade / fonction / statut / lieu d'exercice	Rapporteur ou examinateur ou directeur de thèse ou codirecteur de thèse
Civilité / prénom NOM,	Grade / fonction / statut / lieu d'exercice	Rapporteur ou examinateur ou directeur de thèse ou codirecteur de thèse
Civilité / prénom NOM,	Grade / fonction / statut / lieu d'exercice	Rapporteur ou examinateur ou directeur de thèse ou codirecteur de thèse

Abstract

Résumé

Remerciements

Contents

List of Figures

List of Tables

Acronyms

CGAN	Conditional Generative Adversarial Networks
CycleGAN	Cycle-Consistent Generative Adversarial Networks
DCGAN	Deep Convolutional Generative Adversarial Networks
ELBO	Evidence Lower Bound
FID	Fréchet Inception Distance
GAN	Generative Adversarial Networks
GMM	Gaussian Mixture Model
IPM	Integral Probability Metric
IS	Inception Score
JS	Jensen-Shannon (Divergence)
KL	Kullback-Leibler (Divergence)
LSGAN	Least-Squares Generative Adversarial Networks
MSE	Mean-Squared Error
ReLU	Rectified Linear Unit
RKHS	Reproducing Kernel Hilbert Space
SGD	Stochastic Gradient Descent
VAE	Variational Auto-Encoder
WGAN	Wasserstein Generative Adversarial Networks
WGAN-GP	Wasserstein Generative Adversarial Networks with Gradient Penalty

Introduction

Context

Generic deep learning introduction, generic introduction to generative modeling (image generation, whichfaceisreal.com, etc...)

Generative Adversarial Networks (GAN) Goodfellow et al., 2014 have been recently highlighted for their ability to generate photo-realistic images. By providing a simple framework for high-quality, high-dimensional generative modeling, they quickly found real-world applications such as the notorious "deepfakes" (Vaccari & Chadwick, 2020), face-aging (Antipov, Baccouche, & Dugelay, 2017), image super-resolution (Wang, Chen, & Hoi, 2020), map style tranfer (Kang, Gao, & Roth, 2019), video prediction (Vondrick, Pirsiavash, & Torralba, 2016) or 3D objects generation (Wu et al., 2017).

Introduction to applied conditional generative modeling : examples others than geology

Motivations

Geostatistical application : introduction and needs

- Tuneable (quality vs enforcement of the constraints)
- Pixel-precise
- Keeping diversity

Polarimetry application : introduction and needs

- Designing custom-made constraints for the problem
- Non-euclidian
- Compatible with domain transfer

Contributions

Outline

Chapter 1

Introduction to Generative Adversarial Networks

Chapter abstract

In this chapter, we first propose an introduction to the problem of generative modeling and some solutions to tackle this problem. We then propose an overview of the Generative Adversarial Networks (Goodfellow et al., 2014) framework, which is a recent method to train deep neural networks as generative models that is particularly adapted to the task of image generation. We will introduce some of its theoretical interpretations, as well as some of its variations and applications. We discuss the different limitations of this approach and expose three main goals among the different works: enhancing the visual quality of the generated samples; maintaining their diversity; and conditioning the model. We then discuss the recent advances that have been made to overcome some of these limitations and propose a taxonomy of these advances using the aforementioned directions. We discuss the evaluation of generative models and the difficulties of evaluating the intrinsic quality of a generated sample through an overview of the different classical metrics and discuss their limitations.

Contents

1.1 Generative modeling with Adversarial models

Generative modeling with deep neural networks has been a challenging task due to the stochastic nature of sampling, which prevents the computation of gradient, thus preventing the classical training of a deep model with gradient descent. However recent approaches such as variational autoencoders (VAEs) (Kingma & Welling, 2014), flow methods (Dinh, Sohl-Dickstein, & Bengio, 2017; Kingma & Dhariwal, 2018) and adversarial models (Goodfellow et al., 2014) managed to overcome this restriction. In this section, we first propose an introduction to generative modeling with a focus on latent variable models.

We will then focus on the Generative Adversarial Networks (Goodfellow et al., 2014) framework, their training process and some of their variants, especially their conditional and domain-transfer variants. We outline some limitations of this framework and propose a formulation of these limitations in the form of a trilemma between the intrinsic quality of the generated samples, their diversity and the quality of the conditioning of the model. With this tool, we propose a taxonomy of the recent GAN approaches and identify trends in these approaches.

1.1.1 Generative modeling with maximum likelihood estimation

Generative modeling is the task of learning the underlying distribution of a dataset in order to generate more samples from that distribution. In other words, it describes how data is generated in terms of a probabilistic model, a distribution from which the whole dataset could have been sampled with a high likelihood.

Indeed, whereas a discriminative model tries to find decision boundaries by fitting a parametric model $p_\theta(y|x)$ to a conditional probability distribution $p(y|x)$ of data $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$ relatively to the data $x \sim p(x)$, a generative model aims to fit $p_\theta(x)$ to $p(x)$ the intrinsic distribution of the data and to provide a sampling mechanism on this distribution (see Figure. 1.1).

These two learning tasks, the discriminative (Equation. (1.1)) modeling and the generative modeling (Equation. (1.2)) can be formulated as a maximum log-likelihood estimation

$$\theta^* = \arg\max_\theta \mathop{\mathbb{E}}_{x,y\sim p(y|x)} \log p_\theta(y|x) \qquad (1.1) \qquad\qquad \theta^* = \arg\max_\theta \mathop{\mathbb{E}}_{x\sim p(x)} \log p_\theta(x) \qquad (1.2)$$

An simple example of generative model are Gaussian Mixture Models (GMM) . They consist in a sum of K Gaussian distributions $\mathcal{N}(\mu_k, \sigma_k^2), k \in 1..K$ which are all attributed a selection prob-

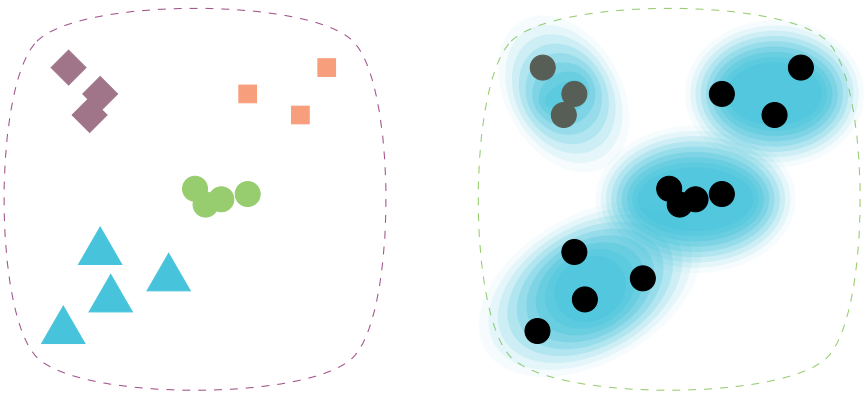


Figure 1.1: Left: Discriminative modeling, the model aims to assign a class to each data point. Right: Generative modeling, the model aims to learn the underlying probability distribution of the data points.

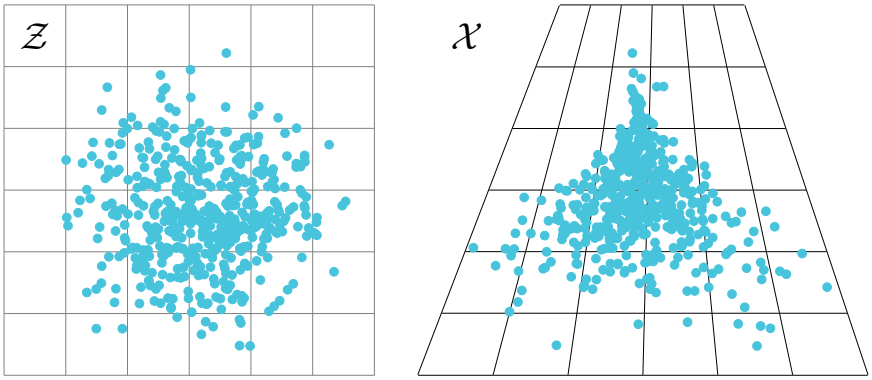


Figure 1.2: A mapping between a latent space $\mathcal{Z}$ and the space of a dataset $\mathcal{X}$.

ability $p(z=k)=\pi_k$, with $z \in \mathcal{Z}$, so that $p(x|z=k)=\mathcal{N}(\mu_k,\sigma_k^2)$. The model is then formulated as

$$p_\theta(x)=\sum_z p(z)\,p_\theta(x|z) \ ,$$

with log-likelihood

$$\log \sum_{x\sim p(x)} p_\theta(x) = \sum_{x\sim p(x)} \log \sum_{k=1}^{k} \pi_k \mathcal{N}(x|\mu_k,\sigma_k^2) \ .$$

In the case of the GMMs, the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977) can be used to find the parameters θ^* which, at convergence, maximizes the log-likelihood of the model. Once the model is trained, sampling a new data is done by picking a component k from the distribution $p(z)$, then drawing a sample from the Gaussian distribution $p(x|z=k)=\mathcal{N}(\mu_k^*,\sigma_k^{*2})$.

1.1.2 Latent variable models

Latent variable models and marginalization

For GMMs, sampling a new point consists in, once the Gaussian component has been selected, sampling a point on a normal distribution. This sampling can be done by using reparametrization: instead directly sampling $x\sim\mathcal{N}(\mu_k^*,\sigma_k^{*2})$, we can instead sample a latent variable $z\sim\mathcal{N}(0,1)$ and compute $x=G(z;\mu,\sigma)=\mu+z\sigma$. Such a model, that consists in a deterministic function $G:\mathcal{Z}\to\mathcal{X}$ with parameters θ applied to a random latent variable drawn from a fixed distribution $p(z)$ is a latent variable model.

Since more complex distributions does not necessarily provide a natural sampling mechanism, using a latent variable model allows to outsource the stochastic part of the sampling process from the learning process and only learn the function $G(z;\theta)$. More formally, instead of directly modeling $p(x)$, a latent variable model learns a deterministic mapping $p_G(x|z)$. From this mapping, the generative model can be obtain through marginalization

$$p_G(x)=\int_{\mathcal{Z}} p(z)\,p_G(x|z)\,dz=\int_{\mathcal{Z}} p(z)\,p(x|G(z;\theta))\,dz \ . \tag{1.3}$$

This marginalization allows for the use of an arbitrary flexible G. However, if G is non-linear, the actual evaluation of $p_G(x)$ is very likely to be intractable due to the integral over $\mathcal{Z}$, which prevents the training of such a model as is.

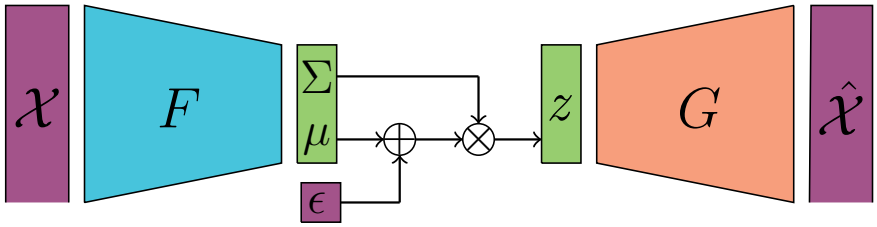


Figure 1.3: Variational auto-encoder framework

While the marginal distribution $p_G(x)$ cannot be explicitly computed for any function G, several solutions exist to overcome this problem and train deep generative models with latent variables anyways.

Variational auto-encoders

Variational Auto-Encoders (VAE) (Kingma & Welling, 2014) are deep latent variable models which tackle the marginalization problem by approximating the integral using a variational approach. To this end, they both learn the distribution of the latent model $p_G(x|z)$ as well as the distribution $q_F(z|x)$. This is done with with two different neural networks, a decoder network $G : \mathcal{Z} \to \mathcal{X}$ and an encoder network $F : \mathcal{X} \to \mathcal{Z}$ and allows to compute the distribution $p(x)$ as

$$\log p_G(x) - D_{KL}\Big(q_F(z|x)\Big|\Big|p(z|x)\Big) = \mathop{\mathbb{E}}_{z\sim q_F(z|x)}\left[\log p_G(x|z)\right] - D_{KL}\Big(q_F(zx)\Big|\Big|p(z)\Big) \ .$$

The KL terms evaluates the distance between the distribution $q_F(z|x)$ learned by the encoder and real distribution $p(z|x)$, and since $p(z)$ is chosen Gaussian, this KL terms can be explicitly computed. The first term, is equivalent to the reconstruction error of an auto-encoder. Hence the model is trained by minimizing

$$L_{VAE}(F,G) = \mathop{\mathbb{E}}_{z\sim q_F(z|x)}\left[||x - G(z)||_2^2\right] - D_{KL}\Big(q_F(z|x)\Big|\Big|p(z)\Big)$$

However, since sampling $z \sim q_F(z|x)$ is not differentiable, the VAE uses the so-called *reparametrization trick*, that is to have $F(x)$ output the mean and the variance (μ_x, σ_x^2) of a normal distribution for a sample x, so that a $\epsilon \sim \mathcal{N}(0,1)$ is sampled outside of the model and given as a parameter, thus allowing to compute $z = \mu_x + \sigma_x\epsilon$, which is differentiable by considering ϵ a parameter.

Finally, generating a sample x with the trained model can be done by sampling a random vector $\epsilon \sim \mathcal{N}(0,1)$ and computing $x = G(z)$.

Normalizing flows

Normalizing flow based techniques is a family of latent variable models that aim to tackle the marginalization problem by using the *change of variable formula*

$$p_G(x) = p(z)\left|\det\left(\frac{\partial G(z)}{\partial z^T}\right)\right|^{-1} = p(G^{-1}(x))\left|\det\left(\frac{\partial G^{-1}(x)}{\partial x^T}\right)\right| \ ,$$

with $z \sim p(z)$ a latent variable. This formulation has notable advantages such as explicitly allowing the computation of the exact inference. However, the model has to enforce some tough constraints: the input and output dimensions must be the same; G must be invertible; and computing the determinant of the Jacobian needs to be efficient and differentiable.

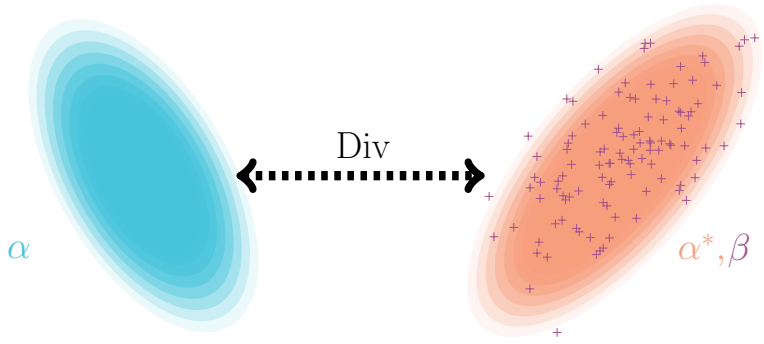


Figure 1.4: A divergence Div(α||β) can capture the distance between a parametric model α_θ and the observations β. Density fitting can then be formulated as $\alpha^* = \arg\min_\theta \mathrm{Div}(\alpha_\theta||\beta)$, where α^* is the best fit model.

These constraints can be enforced through strong restrictions on the architecture of the model. By limiting the transformations to a set of invertible transformations with a tractable Jacobian determinant, the model remains invertible and the determinant of its Jacobian can be computed efficiently.

Real-valued non-volume preserving (RealNVP) normalizing flows (Dinh, Sohl-Dickstein, & Bengio, 2017) uses affine coupling transformations, which transforms a variable by adding and scaling it by a non-linear transformation of itself, usually computed with deep neural networks. These transformations can be inverted by substracting and downscaling by the same transformed variables and their Jacobian is triangular, therefore computing its determinant can be done efficiently by computing the product of its diagonal terms. *Glow* (Kingma & Dhariwal, 2018) extended this set of transformations to 1×1 invertible convolutions as well as a variant of batch normalization that allows for more expressiveness in the model.

1.1.3 Generative Adversarial Networks

In the same fashion as the generative models mentioned in Subsection. 1.1.2, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) aims to learn a parameterized mapping $p_\mathrm{G}(x|z)$ between a simple distribution $p(z)$ (usually normal or uniform) to the real data distribution $p(x)$. However, instead of relying on the likelihood and trying to estimate the distribution through marginalization, it aims to minimize an estimation of a divergence between $p(x)$ and the mapped distribution $p_\mathrm{G}(x)$. Therefore, GANs are often qualified as *likelihood-free* generative models.

Since a divergence $\mathrm{Div}(p(x)||q(x))$ between two distributions $p(x)$ and $q(x)$ is analogous to a distance between these distributions (see Figure. 1.4), minimizing such a divergence allows for a parametric distribution $p_\theta(x)$ to fit a target distribution $p(x)$. When this divergence is both tractable (or estimable) and differentiable w.r.t the parameters θ, it can be directly optimized, allowing for the training of a generative model.

However in practice, such divergences usually intractable in the case of generic distributions. GANs aim to estimate this divergence by relying on a second learned function that will act as a surrogate to the divergence, the discriminator model D. This discriminator is a binary classifier that aims to predict the probability that a sample x was sampled on the real distribution $p(x)$ or was generated from $z \sim p(z)$ and is trained with binary cross-entropy

$$\mathrm{L_D}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{x\sim p(x)}[\log \mathrm{D}(x)] + \mathop{\mathbb{E}}_{x\sim p_\mathrm{G}(x)}[1 - \log \mathrm{D}(x)] \ .$$

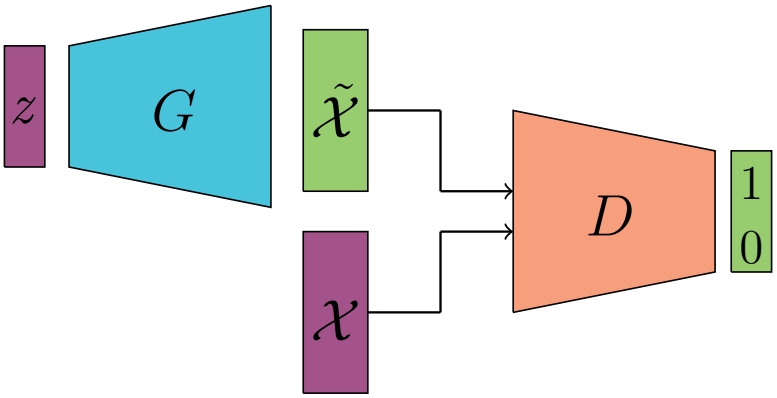


Figure 1.5: Generative Adversarial Networks framework

The intuition behind this model is that once the discriminator is trained, maximizing its error on generated samples $x \sim p_{\mathrm{G}}(x)$ w.r.t the parameters of G should push $p_{\mathrm{G}}(x)$ towards $p(x)$.

The minimum of $f(x) = a\log(x) + b\log(1-x)$ is $\frac{a}{a+b}$, so the discriminator that maximizes $\mathrm{L_D}(\mathrm{D},\mathrm{G})$ for a fixed G is given by

$$\mathrm{D}^*_{\mathrm{G}}(x) = \frac{p(x)}{p(x) + p_{\mathrm{G}}(x)} \ .$$

By plugging this optimal into the discriminator cost, we get

$$\min_{\mathrm{D}} \mathrm{L_D}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{x\sim p(x)}\left[\log \frac{p(x)}{p(x)+p_{\mathrm{G}}(x)}\right] + \mathop{\mathbb{E}}_{x\sim p_{\mathrm{G}}(x)}\left[1 - \log \frac{p_{\mathrm{G}}(x)}{p(x)+p_{\mathrm{G}}(x)}\right] \ .$$

As said previously, the objective of the generator model G will be to maximize the error of the discriminator D. Thus, we can formulate a criterion $\mathrm{L_G}(\mathrm{G})$ as $\mathrm{L_G}(\mathrm{G}) = \min_{\mathrm{D}} \mathrm{L_D}(\mathrm{D},\mathrm{G})$. Up to additive and multiplicative constants, the criterion $\mathrm{L_G}(\mathrm{G})$ can be reformulated as

$$\mathrm{L_G}(\mathrm{G}) = \mathrm{D_{KL}}\left(p(x)\middle\|\frac{p(x)+p_{\mathrm{G}}(x)}{2}\right) + \mathrm{D_{KL}}\left(p_{\mathrm{G}}(x)\middle\|\frac{p(x)+p_{\mathrm{G}}(x)}{2}\right) = 2\cdot \mathrm{D_{JS}}\left(p(x)\middle\|p_{\mathrm{G}}(x)\right) \ .$$

When the discriminator is trained to convergence, minimizing the criterion $\mathrm{L_G}(\mathrm{G}) = \mathrm{L_{GAN}}(\mathrm{D}^*,\mathrm{G})$ is equivalent to minimizing the Jensen-Shannon (JS) divergence between $p(x)$ and $p_{\mathrm{G}}(x)$. This training process is summed up as a mini-max game in Equation. (1.4)

$$\arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathrm{L_{GAN}} = \arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathop{\mathbb{E}}_{x\sim p(x)}[\log \mathrm{D}(x)] + \mathop{\mathbb{E}}_{z\sim p(z)}[1-\log \mathrm{D}(\mathrm{G}(z))] \ . \tag{1.4}$$

As shown above, this mini-max game has, assuming infinite capacity for both G and D, a global optimum for $p(x) = p_{\mathrm{G}}(x)$. The GAN training algorithm then consists in alternatively updating the discriminator and the generator via gradient ascent/descent. A summary of this process is presented in Algorithm. 1.

1.1.4 Limitations

GANs have shown strong advantages over the classical generative modeling methods, such as generating sharper samples than VAEs and normalizing flows (Danihelka et al., 2017). They however bear limitations, namely: the instability of the training process; the loss of diversity in the generated samples (*mode-collapse*); and finally the problem of black-box conditioning.

Algorithm 1 The GAN training algorithm

Require: $\mathscr{D}_{\mathrm{X}}$ the real dataset, G the generator model, and D the discriminator model
  repeat
    sample a mini-batch $\{x_i\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{X}}$
    sample a mini-batch $\{z_i\}_{i=1}^m$ from $p(z)$
    update D by stochastic gradient ascent of
      $\sum_{i=1}^m \log(\mathrm{D}(x_i)) + \log(1-\mathrm{D}(\mathrm{G}(z_i)))$
    sample a a mini-batch $\{z_j\}_{j=1}^n$ from distribution $p(z)$;
    update G by stochastic gradient descent of
      $\sum_{j=1}^n \log(1-\mathrm{D}(\mathrm{G}(z_j)))$
  until a stopping condition is met

Instability

As we have seen in the previous section, training GANs consist in solving a minimax problem. While the alternate gradient descent algorithm is a straightforward method for solving such a problem, the alternating updates can cause significant instabilities during the training process. This can result in oscillating values of the loss function which prevents convergence (Mescheder, Geiger, & Nowozin, 2018), which makes it difficult to determine a when to stop training. In the end, this behavior can be harmful in terms of performance.

CR: Figure loss GAN

The instability of the GAN training process has first been conjectured to be caused by the bad quality of the gradients obtained when G generates bad samples, which makes D strongly reject these samples and therefore saturating the loss. The first solution proposed (Goodfellow et al., 2014) was to slightly change the generator's loss function from $\log(1-\mathrm{D}(\mathrm{G}(z)))$ to $-\log(\mathrm{D}(\mathrm{G}(z)))$, which helped considerably in avoiding failures of the training process and was then widely used (Radford, Metz, & Chintala, 2015) CR: Plus de citations.

While this loss term converges to the same minimum as the original loss term, minimizing it no longer correspond to minimizing a JS divergence but the non-symmetric reverse KL divergence (minus a JS term) (Arjovsky & Bottou, 2017). More formally,

$$\mathop{\mathbb{E}}_{z\sim p(z)}\left[\nabla_{\mathrm{G}}\log \mathrm{D}^*(\mathrm{G}(z))\right] = \nabla_{\mathrm{G}}\left[\mathrm{D}_{\mathrm{KL}}\Big(p_{\mathrm{G}}(x)\Big\|p(x)\Big) - 2\mathrm{D}_{\mathrm{JS}}\Big(p_{\mathrm{G}}(x)\Big\|p(x)\Big)\right] .$$

However, albeit an empirical reduction of the instability, this loss substitution has been proved to not solve the instability problem (Arjovsky & Bottou, 2017). This is mainly due to an unstable behavior of these divergences when the real distribution and the learned one does not share the same support.

A lot of similar tricks can be applied to the training process in order to avoid this pitfall (Salimans et al., 2016; Sønderby et al., 2017; Heusel et al., 2017), and while more recent approaches seemed to help alleviate this issue (which will be more detailed in the next section), instability can still be observed in the most recent approaches (Brock, Donahue, & Simonyan, 2018). Even though several theory-backed techniques aimed to solve this issue (Arjovsky, Chintala, & Bottou, 2017; Nowozin, Cseke, & Tomioka, 2016; Li et al., 2017), there are still, at the time of writing this thesis, neither clear consensus on the theoretical causes of this instability nor completely efficient solutions.

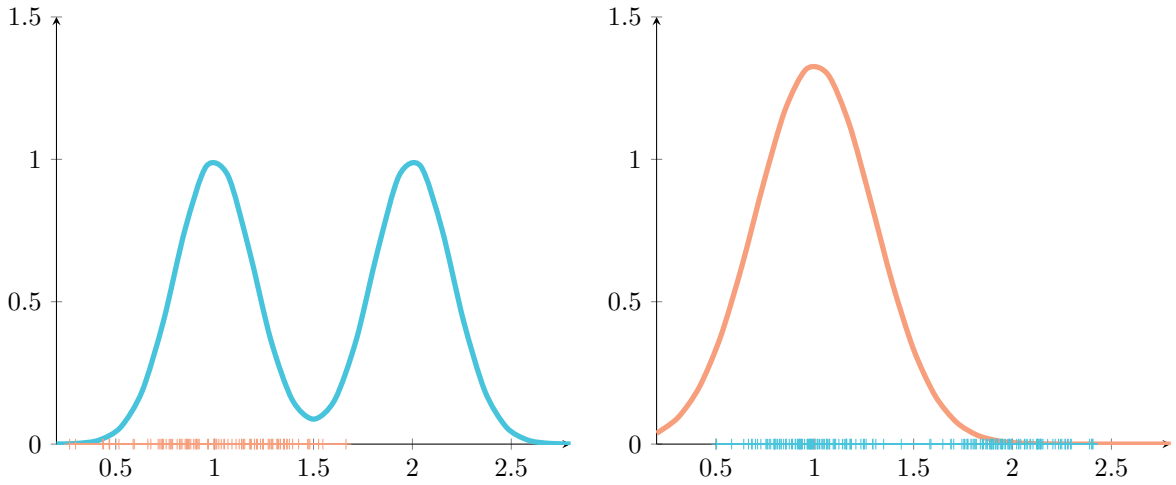


Figure 1.6: Reverse KL (left) and KL (right) divergence between the true blue distribution and the mode-collapsed orange distribution . The distance is lower in the case of the reverse KL, even if a missing mode is clearly visible.

Mode collapse

Although the aforementioned change of loss can help solving the instability issues, using the reverse KL divergence is conjectured to be one of the causes of another issue: the *mode collapse* problem: different z_1, z_2 are mapped to samples $G(z_1)$ and $G(z_2)$ that are very close; and *mode dropping*: only a localized subset of the distribution can actually be mapped to, leading to missing modes in the generated samples.

Indeed, the reverse KL divergence does not penalize "missing" parts of the learned distribution $p_G(x)$, which are some points in the support of $p(x)$ that have zero (or near-zero) probability on $p_G(x)$ (see Figure. 1.6).

Another conjectured cause is raised by the alternate gradient descent, in that it does not clearly prioritize the minimax formulation $G^* = \min_G \max_D L_{GAN}$ over the maximin formulation $G^* = \max_D \min_G L_{GAN}$, which does not behave in the fashion as it pushes the generator towards mapping every z to the single most probable x, evaluated by the generator (Goodfellow, 2016).

In the same fashion as the instability problem, there is at the time of writing this thesis no fundamental explanation to this issue. However, it still raise a first trade-off: since using the original GAN creates instability which lead to a drop of visual quality, and using the non-saturating variant creates a lack of diversity. This extends to more recent approaches in which higher visual quality induces a loss of diversity (Brock, Donahue, & Simonyan, 2018).

In the most extreme cases, this loss of diversity can result in a complete collapsing of the sampling mechanism, making it completely impossible to draw diverse samples. However this is not as much of an issue for conditional tasks that consists in mapping an input to one of many acceptable outputs, with one example of such a task being domain-transfer (see Section. 1.2.1).

1.2 The GAN Zoo

Recently, Generative Adversarial Networks have made considerable progress towards generating realistic high definition images (Brock, Donahue, & Simonyan, 2018; Karras et al., 2020; Wang et al., 2018a). These notable successes leverage on an overwhelming amount of incremental enhancements and variations of the original GAN (Hindupur, 2017) that has been made in the recent

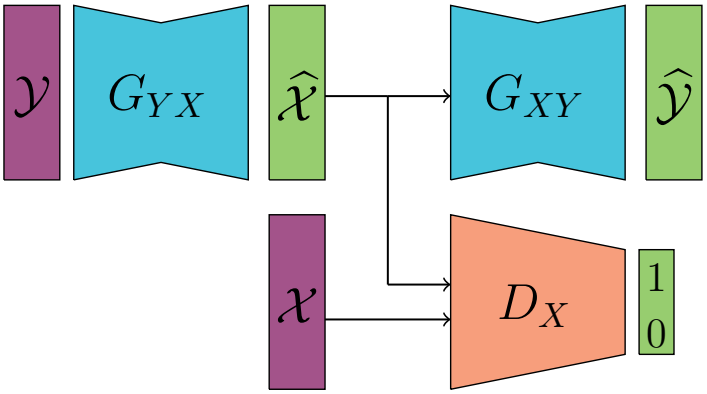


Figure 1.7: The CycleGAN approach. Half of the training setup is illustrated, the other half consisting in the same setup but with inverted X and Y

years. In this section, a summary of these GAN variants is proposed by examining different three objectives: conditioning the generation, enhancing the visual quality of the generated samples, ensuring some diversity among the generated samples. We propose a classification of these approaches into three categories: alternative and additional losses for conditioning; changes to the original objective functions; and improvements to the architecture, regularization, normalization and training process. A summary of this overview is presented in Figure. **??**.

1.2.1 Conditional modeling and domain-transfer

CR: TODO

Conditional modeling

While classical generative models such as GANs try to unconditionally approximate the real-data distribution $p(x)$, a conditional generative model aim to learn a model of the conditional distribution $p(x|y)$, where $y \in \mathcal{Y}$ is a label of any kind.

Several extensions of the GAN framework allow for conditional modeling. First introduced, Conditional GANs (CGANs)(Goodfellow et al., 2014; Mirza & Osindero, 2014) simply adds the label y as an input for both the discriminator and the generator. The new optimization problem that results from this change is summed-up in Equation. (1.5) as follows

$$\arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathrm{L}_{\mathrm{CGAN}} = \arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathop{\mathbb{E}}_{x,y\sim p(x,y)} [\log \mathrm{D}(x,y)] + \mathop{\mathbb{E}}_{\substack{y\sim p(y)\\ z\sim p(z)}} [1-\log \mathrm{D}(\mathrm{G}(y,z),y)] \tag{1.5}$$

While this approach is trivially simple to implement, it relies entirely on the discriminator to use the label. Other approaches try to learn the conditional distribution by adding an explicit loss term to the optimization problem, such as Auxillary Classifier GAN (ACGAN) (Odena, Olah, & Shlens, 2016). This approach aims to learn a conditional generative model with discrete labels by adding another output to the discriminator that acts as a classifier. The model is then trained by having both the generator and the discriminator minimize the categorical cross-entropy between the real and predicted labels.

Domain-transfer

Domain-transfer is the task of learning a mapping $p(x|y)$ between two high-dimensional distributions $p(x)$ and $p(y)$ that maintains semantic information, for example changing the color palette of an image while keeping the same objects at the same position. CGANs already learn to model the conditional distribution $p(x|y)$, and adding a way to enforce the consistency of the semantic information enables domain-transfer.

Pix2Pix (Isola et al., 2016) implemented this approach explicitly by using paired samples $(x, y) \sim p(x|y)$ forcing the generator to minimize the ℓ_1 reconstruction term between x and $\mathrm{G}(y, z)$

$$\arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathrm{L}_{p2p} = \arg\min_{\mathrm{G}}\max_{\mathrm{D}} \mathrm{L}_{\mathrm{CGAN}}(\mathrm{D},\mathrm{G}) + \lambda \mathop{\mathbb{E}}_{\substack{x\sim p(x)\\ y\sim p(y)\\ z\sim p(z)}} ||x - \mathrm{G}(y,z)||_1 \ .$$

However, this kind of approaches rely on paired data which can be very hard to obtain, especially in the case of natural images. When trying for example to transfer images of zebras to images of horses, you need a dataset of very similar zebras and horses in the exact same position for the ℓ_1 term to work.

This problem of paired data was solved by CycleGAN (Zhu et al., 2017b) using cycle-consistency. Instead of training a single model G with reconstruction between x and $\mathrm{G}(y, z)$, the CycleGAN approach train two domain-transfer models simultaneously: $\mathrm{G_{YX}}$ and $\mathrm{G_{XY}}$ that map samples from $p(y)$ onto $p(x)$ and $p(x)$ onto $p(y)$, respectively (see Figure. 1.7). This allows to compute the ℓ_1 reconstruction errors $||x - \mathrm{G_{YX}}(\mathrm{G_{XY}}(x))||_1$ and $||y - \mathrm{G_{XY}}(\mathrm{G_{YX}}(y))||_1$, thus completely removing the need for paired data (x, y). The training of the two models in done an adversarial setup, with two discriminators $\mathrm{D_X}$ and $\mathrm{D_Y}$, and is summed up as an optimization problem in Equation. (1.6)

$$\begin{aligned}\arg\min_{\mathrm{G_{XY}},\mathrm{G_{YX}}}\max_{\mathrm{D_X},\mathrm{D_Y}} \mathrm{L}_{CycGAN} = \arg\min_{\mathrm{G_{XY}},\mathrm{G_{YX}}}\max_{\mathrm{D_X},\mathrm{D_Y}} &\mathrm{L}_{\mathrm{GAN}}(\mathrm{D_X},\mathrm{G_{YX}}) + \mathrm{L}_{\mathrm{GAN}}(\mathrm{D_Y},\mathrm{G_{XY}}) \\ &+ \lambda \mathop{\mathbb{E}}_{x\sim p(x)} ||x - \mathrm{G_{YX}}(\mathrm{G_{XY}}(x)||_1 + \lambda \mathop{\mathbb{E}}_{y\sim p(y)} ||y - \mathrm{G_{XY}}(\mathrm{G_{YX}}(y)||_1 \ . \end{aligned} \tag{1.6}$$

The CycleGAN training process then consists in alternatively updating the two discriminator and the two generators via gradient ascent/descent. A summary of this process is presented in Algorithm. 5.

Task-specific losses

CR: TODO

1.2.2 Objective variants

As mentioned in Subsection. 1.1.4, the original GAN losses as well as the non-saturating losses show strong limitations, the former causes instability and the latter causes a loss in diversity. As a possible solution to these issues, several new loss terms were envisioned

Changing the divergence

As an alternative to the original loss and in order to replace the Jensen-Shannon and the reverse Kullback-Leibler divergences as objectives, the Least-Squares GAN (LSGAN) (Mao et al., 2017) were proposed. In this approach, the loss function are replaced with a least-square formulation of the discriminator error, as

Algorithm 2 CycleGAN training algorithm

Require: $\mathscr{X}$ and $\mathscr{Y}$ two unpaired datasets, G_{XY} and G_{YX} the mapping networks, D_X and D_Y the discrimination models, m the mini-batch size

repeat
- sample a mini-batch $\{x_i\}_{i=1}^m$ from $\mathscr{X}$
- sample a mini-batch $\{y_i\}_{i=1}^m$ from $\mathscr{Y}$
- update D_X by stochastic gradient descent of $\sum_{i=1}^m (D_X(x_i)-1)^2 + (D_X(G_{YX}(y_i)))^2$
- update D_Y by stochastic gradient descent of $\sum_{i=1}^m (D_Y(y_i)-1)^2 + (D_Y(G_{XY}(x_i)))^2$
- sample a mini-batch $\{x_i\}_{i=1}^m$ from X
- sample a mini-batch $\{y_i\}_{i=1}^m$ from Y
- update G_{XY} by stochastic gradient descent of $\sum_{i=1}^n (D_Y(G_{XY}(x_i))-1)^2 + \lambda(||x_i - G_{YX}(G_{XY}(x_i))||_1 + ||y_i - G_{XY}(G_{YX}(y_i))||_1)$
- update G_{YX} by stochastic gradient descent of $\sum_{i=1}^n (D_X(G_{YX}(y_i))-1)^2 + \lambda(||x_i - G_{YX}(G_{XY}(x_i))||_1 + ||y_i - G_{XY}(G_{YX}(y_i))||_1)$

until a stopping condition is met

$$L_{LSGAN}(D,G) = \mathop{\mathbb{E}}_{x\sim p(x)}\left[(1-D(x))^2\right] + \mathop{\mathbb{E}}_{z\sim p(z)}\left[(D(G(z)))^2\right] .$$

While this loss function follow the same idea as the original GAN method, LSGAN actually optimizes the Pearson's χ^2 divergence. Empirically, LSGANs show more stability as well as a higher visual quality of the generated samples than the original GAN approach, which have been conjectured to be caused by a better quality of the gradients.

Although showing notable differences in their behavior when optimized, both the Jensen-Shannon, reverse Kullback-Leibler and Pearson χ^2 divergences are part of the f-divergence family (Liese & Vajda, 2006) defined as

$$D_f(p||q) = \mathop{\mathbb{E}}_{x\sim q(x)} f\left(\frac{p(x)}{q(x)}\right) ,$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex, lower-semicontinuous function satisfying $f(1) = 0$. By carefully choosing f, we can recover the KL ($f(u) = u\log u$), reverse KL ($f(u) = -\log u$), JS ($f(u) = -(u+1)\log(\frac{u+1}{2}) + u\log u$) and Pearson's χ^2 ($f(u) = (u-1)^1$) divergences. Nowozin, Cseke, and Tomioka (2016) proposed a generalized approach for these divergences as well as several new GAN formulation based on divergences such as the Squared Hellinger distance or the Total Variation, which has been shown (Arjovsky, Chintala, & Bottou, 2017) to be the divergence used in the Energy-Based GAN (Zhao, Mathieu, & LeCun, 2017) approach.

While the f-divergences have been the seminal approach to GANs, they can exhibit strong issues. Arjovsky, Chintala, and Bottou (2017) have shown that these divergences can have degenerate behavior, most notably in the case where the two distributions have no shared support, which is reflected by points where the divergence is non-continuous and non-differentiable.

As a solution to this issue and orthogonal to the f-divergences, Arjovsky, Chintala, and Bottou (2017) proposed the Wasserstein GAN (WGAN), replacing the Jensen-Shannon divergence by the Wasserstein -1 (or Earth-Mover) distance, which stems from transportation theory (Peyré & Cu-

turi, 2020). The Wasserstein distance, albeit having many different formulations, can be expressed through the Kantorovich-Rubinstein duality (Kantorovich & Akilov, 1982) as

$$\mathrm{W}(p||q) = \sup_{f \in \mathscr{F}} \left[\mathop{\mathbb{E}}_{x \sim p(x)} f(x) - \mathop{\mathbb{E}}_{x \sim q(x)} f(x) \right] ,$$

where $\mathbb{F} = \{f : ||f||_{\mathrm{L}} \leq 1\}$ is the set of all 1-Lipschitz functions. By using a parametrized family of functions D (in our case, a neural network), we can formulate the Wasserstein GAN problem as

$$\mathrm{L}_{\mathrm{WGAN}}(\mathrm{D}, \mathrm{G}) = \min_{\mathrm{G}} \max_{\mathrm{D}} \left[\mathop{\mathbb{E}}_{x \sim p(x)} \mathrm{D}(x) - \mathop{\mathbb{E}}_{z \sim q(z)} \mathrm{D}(\mathrm{G}(z)) \right] .$$

This formulations, however, requires the discriminator to be 1-Lipschitz, which is done by clipping the weights w of the discriminator to a fixed interval $w \in [-c, c]$. This solution proved to be quite harmful in terms of visual quality by Gulrajani et al. (2017), who proposed the WGAN Gradient Penalty (WGAN-GP), which replace this clipping by a gradient penalty. This additional loss term pushes the discriminator towards having a gradient norm equal to 1 and is formulated as

$$\mathrm{W}_{\mathrm{GP}}(p||p_{\mathrm{G}}) = \max_{\mathrm{D}} \left[\mathop{\mathbb{E}}_{x \sim p(x)} \mathrm{D}(x) - \mathop{\mathbb{E}}_{z \sim q(z)} \mathrm{D}(\mathrm{G}(z)) \right] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim p(\hat{x})} \left[(||\nabla_{\hat{x}} \mathrm{D}(\hat{x})||_2 - 1)^2 \right] ,$$

where $p(\hat{x})$ is implicitly defined as an uniform distribution on straight lines between pairs of points sampled on $p(x)$ and $p_{\mathrm{G}}(x)$. This artificial distribution is used to overcome the intractability of enforcing the gradient norm constraint everywhere.

In the same fashion as the f-divergence family, the Wasserstein distance is a special case of the Integral Probability Metrics (IPM) (Müller, 1997), defined as

$$\mathrm{D}_{\mathscr{F}}(p||q) = \sup_{f \in \mathscr{F}} \left[\mathop{\mathbb{E}}_{x \sim p(x)} f(x) - \mathop{\mathbb{E}}_{x \sim q(x)} f(x) \right] ,$$

where $\mathscr{F}$ is a family of real-valued bounded measurable functions. By putting restrictions on $\mathscr{F}$, several classical divergences can be recovered (Sriperumbudur et al., 2009), among them the Wasserstein divergence ($\mathscr{F} = \{f : f \text{ is 1-Lipschitz}\}$), as well as the Total Variation ($\mathscr{F} = \{f : ||f||_\infty \leq 1\}$).

Another category of approaches that are part of the IPMs are moment matching methods, most notably the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), which is defined as the IPM that restricts the set $\mathscr{F}$ to the set of functions in the ball of a Reproducing Kernel Hilbert Space (RKHS) , or more formally: $\mathscr{F} = \{f : ||f||_{\mathscr{H}} \leq 1\}$, with $\mathscr{H}$ a RKHS of kernel $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$).

Although this distance can show nice properties, allowing for two-sample testing, it relies on the choice of the kernel k. Thus by using a fixed kernel, MMD was used to formulate the different MMDGAN (**Li2017a** ; Dziugaite, Roy, & Ghahramani, 2015; Bińkowski et al., 2018) approaches, which train GANs by estimating the MMD with either gaussian or quadratic kernels.

More recent approaches leverage gradient penalty similar as WGAN-GP in order to learn the kernel k, which translates into special cases of MMD such as Energy Distance (Bellemare et al., 2017; Szekely & Rizzo, 2004) or the so-called Fisher IPM Mroueh and Sercu, 2017.

CR: Hinge Loss

Augmenting the objective

Semi-supervised, self supervised ACGAN, ALI/BigGAN, Structured GAN, TripleGAN

Approach	Divergence
f-divergences	
GAN (Goodfellow et al., 2014)	Jensen-Shannon
NS-GAN (Goodfellow et al., 2014)	Reverse Kullback-Leibler
LSGAN (Mao et al., 2017)	Pearson χ^2
EBGAN* (Zhao, Mathieu, & LeCun, 2017)	Total variation
f-GAN (Nowozin, Cseke, & Tomioka, 2016)	Various f-divergences
Integral Probability Metrics (IPMs)	
EBGAN* (Zhao, Mathieu, & LeCun, 2017)	Total variation
WGAN (Arjovsky, Chintala, & Bottou, 2017)	Wasserstein distance
Cramér GAN (Bellemare et al., 2017)	Energy Distance (Unbiased WGAN)
MMDGAN (Li et al., 2017)	Maximum Mean Discrepancy
Fisher GAN(Mroueh & Sercu, 2017)	Fisher IPM

Table 1.1: A summary of common f-divergences and IPM used to train GANs. Note than the Total Variation can be formulated as both.

1.2.3 Architecture, regularization and normalization

The original GAN approach (Goodfellow et al., 2014) used very simple multi-layer perceptrons as discriminator and generator. While this approach showed equal or better performance than most generative models of its time (Kingma & Welling, 2014; Bengio et al., 2014) on small image datasets (LeCun et al., 1998; Krizhevsky, 2009), these simple architectures were quickly enhanced with tools from regular deep learning and computer vision.

The first two notable enhancements were the Laplacian Pyramid GAN (LAPGAN) (Denton et al., 2015) and the Deep Convolutional GAN (DCGAN) (Radford, Metz, & Chintala, 2015). The LAPGAN approach used Laplacian Pyramids (Burt & Adelson, 1983) to iteratively upscale a low-resolution generated sample. The DCGAN approach replaced the discriminator by a simple fully-convolutional network (Springenberg et al., 2015) with strided convolutions and introduced de-convolutional (or transposed convolutional) layers in the generator. It also introduced dropout (Srivastava et al., 2014) and Batch Normalization (Ioffe, Szegedy, & Ioffe, 2015), and used both ReLU (Nair & Hinton, 2010) and Leaky ReLU (Maas, Hannun, & Ng, 2013) as activation functions. This last approach showed much better results than the original GAN and the LAPGAN and became a standard baseline for image generation.

Although this approach remained unstable, it was extended (Salimans et al., 2016) with several tricks such as matching features from real and generated data, smoothing the 0/1 label or adding noise to the discriminator's input (Sønderby et al., 2017) that helped stabilizing the training process. However, the DCGAN approach was stil limited in both the visual quality of the generated samples and in its ability to generate high-dimension images.

Progressive GAN, introduced by Karras et al. (2017), allowed for the first high-dimensional

Proggan, spectral normalization, self-attention, Biggan, stylegan/2

1.2.4 A note on the evaluation of GANs

Unlike discriminative models, evaluating and comparing GAN approaches is a non-trivial task. Two approaches can be envisioned: evaluating the *intrinsic* quality of generated samples with ad-hoc criterions or directly evaluating the likelihood of the generated samples. However, unlike VAEs and flow-based models, GANs offer no explicit way to evaluate or approximate the likelihood of the generated samples. Thus, a significant part of the GAN literature resorted to a subjective visual

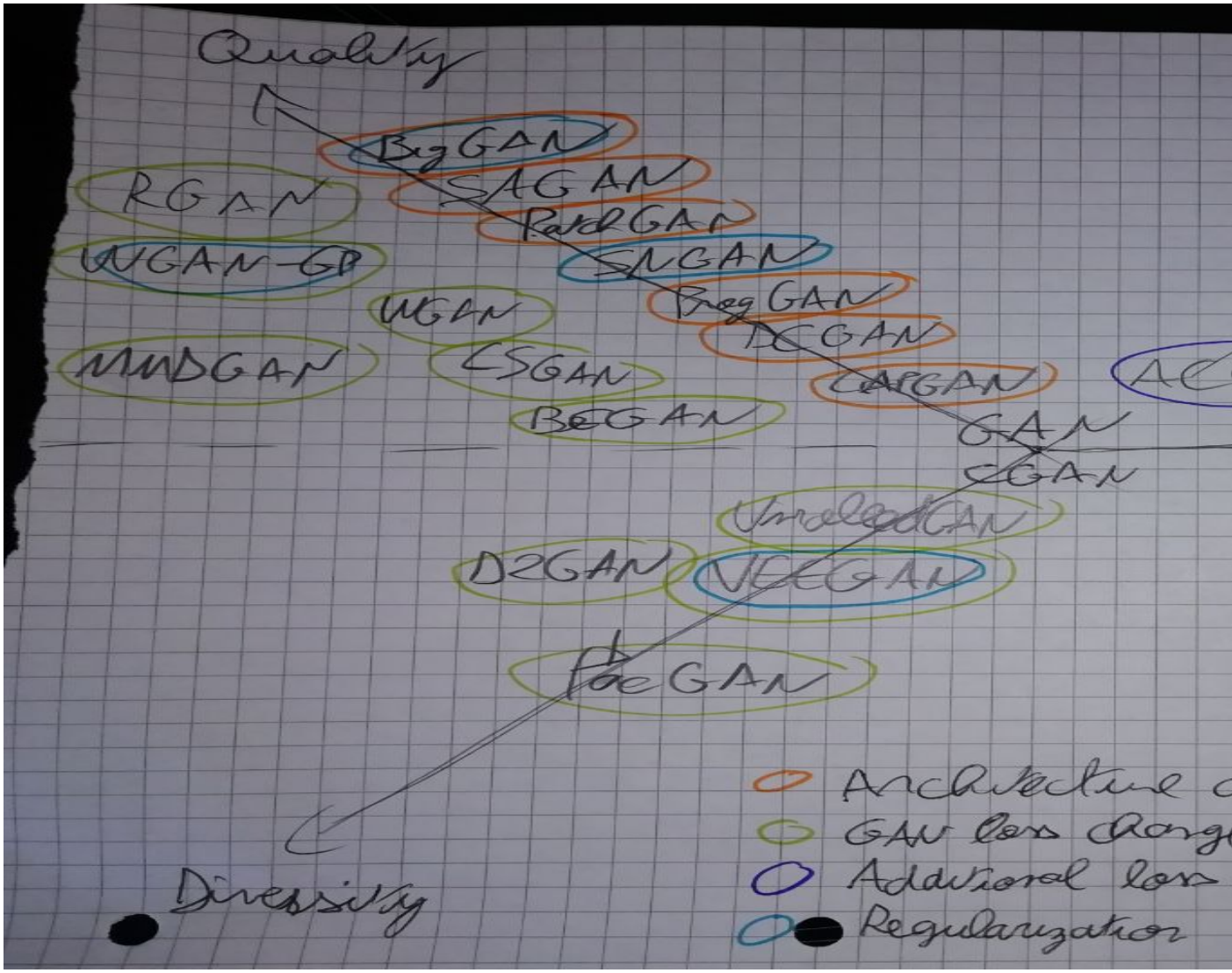


Figure 1.8: Classifications of some advances in GANs on the trilemma

evaluation of the generated samples.

In order to provide a more precise evaluation of the visual quality of generated samples, two ad-hoc methods Inception Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017) were proposed, which both make use of a pre-trained Inception v3 model (Szegedy et al., 2016), a deep classifier trained on the ImageNet dataset (Deng et al., 2009).

Inception Score (IS)) (Salimans et al., 2016) is based on the evaluation of the entropy of the labels y predicted by the Inception classifier of generated data. High-fidelity samples should be easier to classify and therefore have a conditional label distribution $p_{\mathrm{G}}(y|x)$ with low entropy. In addition to the high quality, the samples should be diverse, therefore the marginal distribution $p_{\mathrm{G}}(y) = \int_{\mathcal{Z}} p_{\mathrm{G}}(y|x = \mathrm{G}(z))dz$ should have a high entropy. By combining these two requirements, the IS is formulated as

$$\mathrm{IS}(y) = \exp\left[\mathop{\mathbb{E}}_{x\sim p_{\mathrm{G}}(x)} \mathrm{D}_{\mathrm{KL}}\Big(p_{\mathrm{G}}(y|x)\Big|\Big|p_{\mathrm{G}}(y)\Big)\right] \ .$$

Although it has been widely used, IS has shown major issues (Barratt & Sharma, 2018) that raise from the use of the conditional label distribution. Most notably, examples that are correctly classified are not necessarily of the highest quality and the pre-determined label classes can skew the estimation of the marginal distribution $p_{\mathrm{G}}(y)$.

The **Fréchet Inception Distance (FID)** (Heusel et al., 2017) differs from IS since it evaluates a distance between the distributions of visual features computed on real and generated data, instead of relying on the labels. These features are extracted at the penultimate layer of the Inception classifier. The distributions of these features are assumed Gaussian, so that the Fréchet distance (or Wasserstein-2 distance) can be computed as

$$\mathrm{FID} = ||\mu – \mu_{\mathrm{G}}||^2 + \mathrm{T}r(\Sigma + \Sigma_{\mathrm{G}} – 2\sqrt{\Sigma \times \Sigma_{\mathrm{G}}}) \ ,$$

where $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu_{\mathrm{G}}, \Sigma_{\mathrm{G}})$ are the distributions of the extracted features of the real and generated data, respectively. FID is considered more robust than IS and has been either completing or replacing the use of IS in recent works.

However, while these two metrics are considered to be the standard method for evaluating GANs, their reliance on the pre-trained Inception model can prove to be an issue. Indeed, they behave well when used to compare models learned on natural images datasets such as ImageNet, but they cannot directly be applied to other datasets. A solution to consider can be the training of another classifier network on a more adapted dataset, but this solution cannot be applied when no labeled data is available.

For completeness, we can also refer to notable (albeit less used) among numerous others metrics for evaluating visual quality (Borji, 2018): the Parzen window (or kernel density) estimation (Parzen, 1962) aim to estimate the likelihood of the generated samples; the Sliced Wasserstein Distance (Julien et al., 2011) is an efficient approximation of the Earth-Mover (or Wasserstein) distance; the Kernel Inception Distance (Bińkowski et al., 2018) is a recent metric that evaluates the maximum mean discrepancy between Inception features with a polynomial kernel.

Finally it is to note that for conditioned models, evaluating the aforementioned metrics does not inform about the quality of the conditioning. However, since the conditioning usually requires either labels or prior information, these can often be evaluated by, for example, predicting the labels of generated samples with a pre-trained classifier and computing the error between the predicted label and the original one.

1.3 Conclusion

Chapter 2

Reconstruction as an Auxiliary Task for Generative Modeling

Chapter abstract

While the Conditional GAN approach (Mirza & Osindero, 2014) is theoretically generic enough to model any kind of conditioning, it lacks some form of control or guarantee on the respect of these constraints. In this chapter, we propose to explore another approach for conditioning a GAN model through an image reconstruction task, which consists in (re-)generating images from a very sparse set of randomly-positioned pixels known beforehand. We reformulate this conditional generation task as a Maximum A Posteriori estimation and find a solution in the form of an explicit auxiliary reconstruction task, which adds to the original unconditional GAN objective as an additional loss term. Complemented with the PacGAN (Lin et al., 2018) variant for training GANs, this approach enables the generation of diverse samples from a sparse pixel map. As opposed to the more classical Conditional GAN approach, this auxiliary task is interpretable and a hyperparameter allows to control the importance of the conditioning in the learning process. We evaluate our approach on the classical MNIST, FashionMNIST and CIFAR10 datasets, as well as a custom-made texture dataset. Finally, we apply this approach to a real-world task of geostatistical simulation.

Contents

2.1 Introduction

As we have seen in Section. **??**, Conditional GANs enables a variety of conditioned generation, such as class-conditioned image generation (Mirza & Osindero, 2014), image-to-image translation (Isola et al., 2016; Wang et al., 2018b), or image inpainting (Pathak et al., 2016). CR: Wow, ça fonce rapidement sur AmbientGAN !

On the other side, Ambient GAN (Bora, Price, & Dimakis, 2018) aims at training an unconditional generative model using only noisy or incomplete samples y. Relevant application domain is high-resolution imaging (CT scan, fMRI) where image sensing may be costly. Ambient GAN attempts to produce unaltered images $\tilde{x}$ which distribution matches the true one without accessing to the original images x. For the sake, Ambient GAN considers lossy measurements such as blurred images, images with removed patch or removed pixels at random (up to 95%). Following this setup, Pajot, de Bezenac, and Gallinari (2019) extend the learning strategy to enable the reconstruction instead of the generation of realistic images from similarly altered samples.

In the spirit of Ambient GAN, we consider in this paper an extreme setting of image generation when only a few pixels, less than a percent of the image size, are known and are randomly scattered across the image (see Fig.2.1c). We refer to these conditioning pixels as a constraint map y. To reconstruct the missing information, we design a generative adversarial model able to generate high quality images coherent with given pixel values by leveraging on a training set of similar, but not paired images. The model we propose aims to match the distribution of the real images conditioned on a highly scarce constraint map, drawing connections with Ambient GAN while, in the same manner as CGAN, still allowing the generation of diverse samples following the underlying conditional distribution.

To make the generated images honoring the prescribed pixel values, we use a reconstruction loss measuring how close real constrained pixels are to their generated counterparts. We show that minimizing this loss is equivalent to maximizing the log-likelihood of the constraints given the generated image. Thereon we derive an objective function trading-off the adversarial loss of GAN and the reconstruction loss which acts as a regularization term. We analyze the influence of the related hyper-parameter in terms of quality of generated images and the respect of the constraints. Specifically, empirical evaluation on FashionMNIST (Xiao, Rasul, & Vollgraf, 2017) evidences that the regularization parameter allows for controlling the trade-off between samples quality and constraints fulfillment.

Additionally to show the effectiveness of our approach, we conduct experiments on CIFAR10 (Krizhevsky, 2009), CelebA (Liu et al., 2015) or texture (Jetchev et al., 2017) datasets using various deep architectures including fully convolutional network. We also evaluate our method on a classical geological problem which consists of generating 2D geological images of which the spatial patterns are consistent with those found in a conceptual image of a binary fluvial aquifer(Strebelle, 2002)(Laloy et al., 2018). Empirical findings reveal that the used architectures may lack stochasticity from the generated samples that is the GAN input is often mapped to the same output image irrespective of the variations in latent code (Yang et al., 2019). We address this issue by resorting to the recent PacGAN (Lin et al., 2018) strategy. As a conclusion, our approach performs well both in terms of visual quality and respect of the pixel constraints while keeping diversity among generated samples. Evaluations on CIFAR-10 and CelebA show that the proposed generative model always outperforms the CGAN approach on the respect of the constraints and either come close

or outperforms it on the visual quality of the generated samples.

The remainder of the chapter is organized as follows. In Section **??**, we review the relevant related work focusing first on methods dealing with image generation and reconstruction from highly altered training samples. Section **??** details the overall generative model we propose. In Section 2.4, we present the experimental protocol and evaluation measures while Section **??** gathers quantitative and qualitative effectiveness of our approach. The last section concludes the paper.

The contributions of this chapter are summarized as follows:

- We propose a method for learning to generate images with a few pixel-wise constraints.
- We expose a controllable trade-off between the image quality and the constraints' fulfillment is highlighted,
- We showcase a lack of diversity in generating high-dimensional images which we solve by using PacGAN(Lin et al., 2018) technique. Several experiments allow to conclude that the proposed formulation can effectively generate diverse and high visual quality images while satisfying the pixel-wise constraints.

2.2 Image Reconstruction, Inpainting and Compressed Sensing

2.2.1 Problem formulation

The pursued objective of the chapter is image generation using generative deep network conditioned on randomly scattered and scarce (less than a percent of the image size) pixel values. This kind of pixel constraints occurs in application domains where an image or signal need to be generated from very sparse measurements.

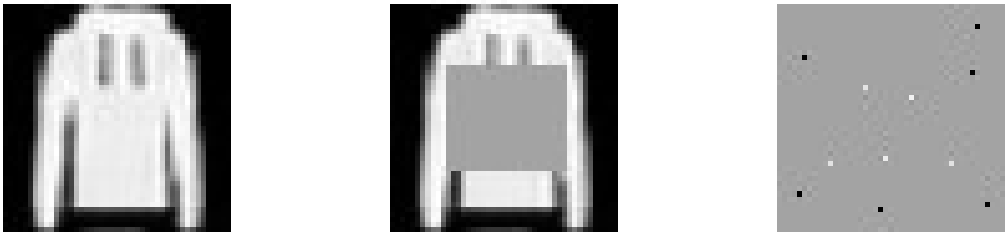
Before delving into the details, let introduce the notations and previous work related to the problem. We denote by $\mathrm{X} \in \mathscr{X}$ a random variable and x its realization. Let p_{X} be the distribution of X over $\mathscr{X}$ and $p_{\mathrm{X}}(x)$ be its evaluation at x. Similarly $p_{\mathrm{X|Y}}$ represents the distribution of X conditioned on the random variable $\mathrm{Y} \in \mathscr{Y}$.

Given a set of images $x \in \mathscr{X} = [-1,1]^{n \times p \times c}$ (see Figure 2.1a) drawn from an unknown distribution p_{X} and a sparse matrix $y \in \mathscr{Y} = [-1,1]^{n \times p \times c}$ (Figure 2.1c) as the given constrained pixels, the problem consists in finding a generative model G with inputs z (a random vector sampled from a known distribution p_{Z} over the space $\mathscr{Z}$) and constrained pixel values $y \in [-1,1]^{n \times p \times c}$ able to generate an image satisfying the constraints while likely following the distribution p_{X} (see Figure 2.3).

2.2.2 Related approaches

Among several applications, the GANs was adapted to image inpainting task (Figure 2.1b). For instance Yeh et al. (Yeh et al., 2017) propose an inpainting approach which considers a pre-trained generator, and explores its latent space $\mathscr{Z}$ through an optimization procedure to find a latent vector z, which induces an image with missing regions filled in by conditioning on the surroundings available information. However, the method requires to solve a full optimization problem at inference stage, which is computationally expensive.

Although CGAN was initially designed for class-conditioned image generation by setting y as the class label of the image, several types of conditioning information can apply such as a full image for image-to-image translation (**Isola2017**) or partial image as in inpainting (Yu et al., 2018). CGAN-based inpainting methods rely on generating a patch that will fill up a structured missing part of the image and achieve impressive results. However they are not well suited to reconstruct very sparse and unstructured signal (Demir & Unal, 2018). Additionally, these approaches learn



(a) Original Image (b) Inpainting Input (c) Constraint Map

Figure 2.1: Difference between regular inpainting (2.1b) and the problem undertaken in this work (2.1c) on a real sample (2.1a).

to reconstruct a single sample instead of a full distribution, implying that there is no sampling process for a given constraint map or highly degraded image.

AmbientGAN (Bora, Price, & Dimakis, 2018) (Figure 2.2c) trains a generative model capable to yield full images from only lossy measurements. One of the image degradations considered in this approach is the random removal of pixels leading to sparse pixel map y. It is simulated with a differentiable function f_θ whose parameter θ indicates the pixels to be removed. The underlying optimization problem solved by AmbientGAN is therefore stated as

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{y\sim p_{\mathrm{Y}}} \Big[\log(\mathrm{D}(y)) \Big] + \mathop{\mathbb{E}}_{\substack{z\sim p_{\mathrm{Z}} \\ \theta\sim p_\theta}} \Big[\log(1-\mathrm{D}(f_\theta(\mathrm{G}(z)))) \Big] \ . \tag{2.1}$$

Unsupervised Image Reconstruction (Pajot, de Bezenac, & Gallinari, 2019) combines the AmbientGAN approach with an additional reconstruction task that consists in reconstructing the $f_\theta(\mathrm{G}(y))$ from the twice-altered image $\tilde{y} = f_\theta(\mathrm{G}(y))$ and $\hat{y} = f_\theta(\mathrm{G}(f_\theta(\mathrm{G}(y))))$,

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{y\sim p_{\mathrm{Y}}} \Big[\log(\mathrm{D}(y)) \Big] + \mathop{\mathbb{E}}_{y\sim p_{\mathrm{Y}}} \Big[\log(1-\mathrm{D}(\hat{y})) \Big] + ||\hat{y}-\tilde{y}||_2^2 \ . \tag{2.2}$$

The ℓ_2 norm term ensures that the generator is able to learn to revert f_θ i.e. to revert the alteration process on a given sample. This allows the reconstruction of realistic image only from a given constraint map y. However the reconstruction process is deterministic and does not provide a sampling mechanism.

Compressed Sensing with Meta-Learning (Wu, Rosca, & Lillicrap, 2019) is an approach that combines the exploration of the latent space $\mathcal{Z}$ to recover images from lossy measurements with the enforcing of the Restricted Isometric Property (Candes & Tao, 2005), which states that for two samples $x_1, x_2 \sim p_{\mathrm{X}}$,

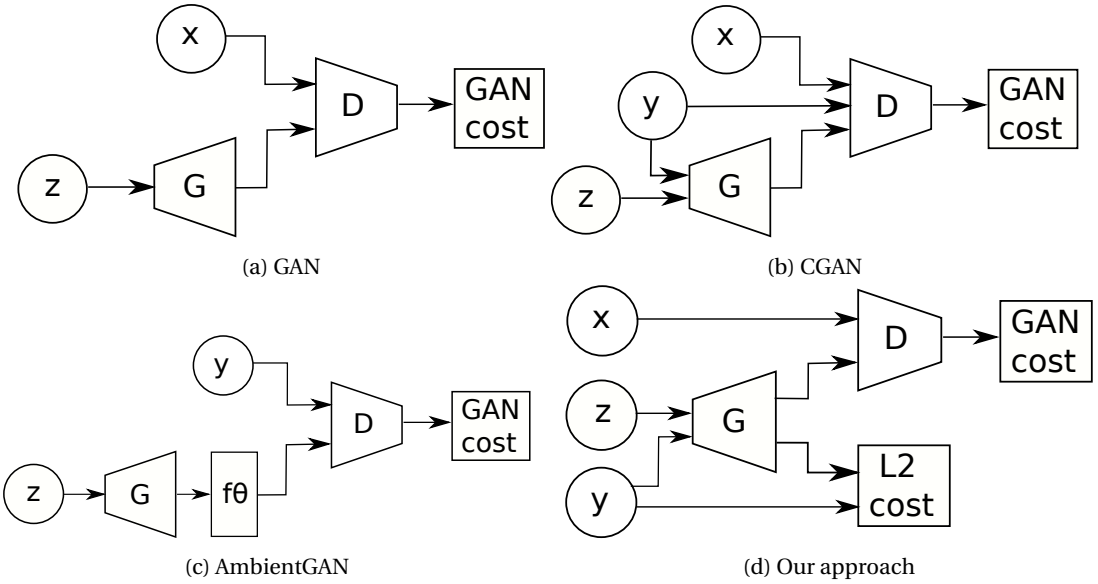
$$(1-\alpha)||x_1-x_2||_2^2 \leq ||f_\theta(x_1-x_2)||_2^2 \leq (1+\alpha)||x_1-x_2||_2^2$$

where α is a small constant. It replaces the adversarial training of the generative model G (Eq. **??**) by searching, for a given degraded image y, a vector $\hat{z}$ such that $\hat{y} = f_\theta(\mathrm{G}(\hat{z}))$ minimizes the ℓ_2 distance between y and $\hat{y}$ while still enforcing the RIP. The overall problem induced by this approach can be formulated as:

$$\min_{\mathrm{G}} \mathrm{L}(\mathrm{G}) = \mathop{\mathbb{E}}_{\substack{x\sim p_{\mathrm{X}} \\ y\sim p_{\mathrm{Y}} \\ z\sim p_{\mathrm{Z}}}} \Big(\sum_{\substack{x_1,x_2\in\mathcal{S} \\ x_1\neq x_2}} (||f_\theta(x_1-x_2)||_2^2 - ||x_1-x_2||_2^2)^2 \Big)/3 + ||y-f_\theta(\mathrm{G}(\hat{z}))||_2^2$$

$$\text{where } \hat{z} = \min_z ||y-f_\theta(\mathrm{G}(z))||^2 \ . \tag{2.3}$$

where $\mathscr{S}$ contains the three samples $x, \mathrm{G}(z), \mathrm{G}(\hat{z})$. In practice, $\hat{z}$ is computed with gradient descent on z by minimizing $||y - f_\theta(\mathrm{G}(z))||^2$, and starting from a random $z \sim p_Z$. As a benefit, this approach may generate an image $\hat{x} = \mathrm{G}(\hat{z})$ from a noisy information y but at a high computation burden since it requires to solve an optimization problem (computing $\hat{z}$) at inference stage for generating an image.



(a) GAN (b) CGAN

(c) AmbientGAN (d) Our approach

Figure 2.2: Different GAN Setups. G and D are the generator and discriminator networks, x and z are samples from the distributions P_x and P_r, y is a label/constraint map sampled from P_y and f_θ is an image degradation function.

2.3 From conditional generation to auxiliary task

2.3.1 Image reconstruction as a Maximum A Posteriori estimation

Let introduce the formal formulation of the addressed problem. Assume y is the given set of constrained pixel values. To ease the presentation, let consider y as a $n \times p \times c$ image with only a few available pixels (less than 1% of $n \times p \times c$). We will also encode the spatial location of these pixels using a corresponding binary mask $\mathrm{M}(y) \in \{0,1\}^{n \times p \times c}$. We intend to learn a GAN whose generation network takes as input the constraint map y and the sampled latent code $z \in \mathscr{Z}$ and outputs a realistic image that fulfills the prescribed pixel values. Within this setup, the generative model can sample from the unknown distribution p_X of the training images $\{x_1, \cdots, x_\mathrm{N}\}$ while satisfying unseen pixel-wise constraints at training stage. Formally our proposed GAN can be formulated as

$$\min_\mathrm{G} \max_\mathrm{D} \mathrm{L}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{x \sim p_x} \Big[\log(\mathrm{D}(x)) \Big] + \mathop{\mathbb{E}}_{\substack{z \sim p_\mathrm{Z} \\ y \sim p_\mathrm{Y}}} \Big[\log(1 - \mathrm{D}(\mathrm{G}(y,z))) \Big] \ , \quad \text{s.t. } y = \mathrm{M}(y) \odot \mathrm{G}(y,z) \tag{2.4}$$

where $\odot$ stands for the Hadamard (or point-wise) product and $\mathrm{M}(y)$ for the mask, a sparse matrix with entries equal to one at constrained pixels location.

As the equality constraint in Problem (2.4) is difficult to enforce during training, we rather investigate a relaxed version of the problems. As in the UNIR approach (Pajot, de Bezenac, & Gal-

linari, 2019) we assume that the constraint map is obtained through a noisy measurement process

$$y = f_{\mathrm{M}}(x) + \varepsilon \ . \tag{2.5}$$

Here f_{M} is the masking operator yielding to $y = \mathrm{M}(y) \odot x$. Also the constrained pixels are randomly and independently selected. ε represents an additive i.i.d noise corrupting the pixels. Therefore we can formulate the Maximum A Posteriori (MAP) estimation problem, which, given the constraint map y, consists in finding the most probable image x^* following the posterior distribution $p_{\mathrm{X|Y}}$,

$$x^* = \arg\max_x \log p_{\mathrm{X|Y}}(x|y) \tag{2.6}$$

$$= \arg\max_x \log p_{\mathrm{Y|X}}(y|x) + \log p_{\mathrm{X}}(x) \ . \tag{2.7}$$

$p_{\mathrm{Y|X}}(y|x)$ is the likelihood that the constrained pixels y are issued from image x while $p_{\mathrm{X}}(x)$ represents the prior probability at x. Assuming that the generation network G may sample the most probable image $\mathrm{G}(y,z)$ complying with the given pixel values y, we get the following problem

$$\mathrm{G}^* = \arg\max_{\mathrm{G}} \mathop{\mathbb{E}}_{\substack{y\sim p_{\mathrm{Y}} \\ z\sim p_{\mathrm{Z}}}} \log p_{\mathrm{Y|X}}(y|\mathrm{G}(y,z)) + \log p_{\mathrm{X}}(\mathrm{G}(y,z)) \ . \tag{2.8}$$

The first term in Problem (2.8) measures the likelihood of the constraints given a generated image. Let rewrite Equation (2.5) as $\mathrm{vect}(y) = \mathrm{vect}(f_{\mathrm{M}}(x)) + \mathrm{vect}(\varepsilon)$ where $\mathrm{vect}(\cdot)$ is the vectorisation operator that consists in stacking the constrained pixels. Therefore, assuming $\mathrm{vect}(\varepsilon)$ is an i.i.d Gaussian noise with distribution $\mathcal{N}(0, \sigma^2 \mathrm{I})$, we achieve the expression of the conditional likelihood

$$log\, p_{\mathrm{Y|X}}(y|\mathrm{G}(y,z)) \propto -\left\|\mathrm{vect}(y) - \mathrm{vect}(\mathrm{M}(y) \odot \mathrm{G}(y,z))\right\|_2^2 \tag{2.9}$$

which evaluates the quadratic distance between the conditioning pixels and their predictions by G. In other words, using a matrix notation of (2.5), the likelihood of the constraints given a generated image equivalently writes

$$\log p_{\mathrm{Y|X}}(y|\mathrm{G}(y,z) \propto -\left\| y - \mathrm{M}(y) \odot \mathrm{G}(y,z)\right\|_{\mathrm{F}}^2 \ . \tag{2.10}$$

$\|\mathrm{A}\|_{\mathrm{F}}^2$ represents the squared Frobenius norm of matrix A that is the sum of its squared entries.

The second term in Problem (2.8) is the likelihood of the generated image under the true but unknown data distribution p_{X}. Maximizing this term can be equivalently achieved by minimizing the distance between p_{X} and the marginal distribution of the generated samples $\mathrm{G}(y,z)$. This amounts to minimizing with respect to G, the GAN-like objective function $\mathbb{E}_{x\sim p_{\mathrm{X}}} \log(\mathrm{D}(x)) + \mathop{\mathbb{E}}_{\substack{z\sim p_{\mathrm{Z}} \\ y\sim p_{\mathrm{Y}}}} \log(1 - \mathrm{D}(\mathrm{G}(y,z)))$ (Goodfellow et al., 2014). Putting altogether these elements, we can propose a relaxation of the hard constraint optimization problem (2.4) (Figure 2.2d) as follows

$$\begin{aligned} \min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) \quad &= \quad \mathop{\mathbb{E}}_{x\sim p_{\mathrm{X}}} \Big[\log(\mathrm{D}(x)) \Big] \\ &+ \quad \mathop{\mathbb{E}}_{\substack{z\sim p_{\mathrm{Z}} \\ y\sim p_{\mathrm{Y}}}} \Big[\log(1 - \mathrm{D}(\mathrm{G}(y,z))) + \lambda \left\| y - \mathrm{M}(y) \odot \mathrm{G}(y,z)\right\|_{\mathrm{F}}^2 \Big] \ . \end{aligned} \tag{2.11}$$

Remarks:

- The assumption of Gaussian noise measurement leads us to explicitly turn the pixel value constraints into the minimization of the ℓ_2 norm between the real enforced pixel values and their generated counterparts (see Figure 2.2d).

(a) Original Image (b) Constraints (c) Generated Image (d) Satisfied Consts.

Figure 2.3: Generation of a sample during training. We first sample an image from a training set (2.3a) and we sample the constraints (2.3b) from it. Then our GAN generates a sample (2.3c). The constraints with squared error smaller than $\epsilon = 0.1$ are deemed satisfied and shown by green pixels in (2.3d) while the red pixels are unsatisfied.

- This additional term acts as a regularization over prescribed pixels by the mask $\mathrm{M}(y)$. The trade-off between the distribution matching loss and the constraint enforcement is assessed by the regularization parameter $\lambda \geq 0$.
- It is worth noting that the noise ϵ can be of any other distribution, according to the prior information, one may associate to the measurement process. We only require this distribution to admit a closed-form solution for the maximum likelihood estimation for optimization purpose. Typical choices are distributions from the exponential family (Brown, 1986).

2.3.2 Conditioning with an auxiliary task

To solve Problem (2.11), we use the stochastic gradient descent method. The overall training procedure is detailed in Algorithm 3 and ends up when a maximal number of training epochs is attained.

When implementing this training procedure we experienced, at inference stage, a lack of diversity in the generated samples (see Figure 2.5) with deeper architectures, most notably the encoder-decoder architectures. This issue manifests itself through the fact that the learned generation network, given a constraint map y, outputs almost deterministic image regardless the variations in the input z. The issue was also pointed out by Yang et al. (Yang et al., 2019) as characteristic of CGANs.

Algorithm 3 Proposed training algorithm

Require: $\mathscr{D}_{\mathrm{X}}$ the set of unaltered images, $\mathscr{D}_{\mathrm{Y}}$ the set of constraint maps, G the generation network, and D the discrimination function

repeat

sample a mini-batch $\{x_i\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{X}}$

sample a mini-batch $\{y_i\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{Y}}$

sample a mini-batch $\{z_i\}_{i=1}^m$ from distribution p_{Z}

update D by stochastic gradient ascent of

$\sum_{i=1}^m \log(\mathrm{D}(x_i)) + \log(1 - \mathrm{D}(\mathrm{G}(y_i, z_i)))$

sample a mini-batch $\{y_j\}_{j=1}^n$ from $\mathscr{D}_{\mathrm{Y}}$

sample a a mini-batch $\{z_j\}_{j=1}^n$ from distribution p_{Z} ;

update G by stochastic gradient descent of

$\sum_{j=1}^n \log(1 - \mathrm{D}(\mathrm{G}(y_j, z_j))) + ||y_j - \mathrm{M}(y_j) \odot \mathrm{G}(y_j, z_j)||_{\mathrm{F}}^2$

until a stopping condition is met

To avoid the problem, we exploit the recent PacGAN (Lin et al., 2018) technique: it consists in passing a set of samples to the discrimination function instead of a single one. PacGAN is intended

to tackle the mode collapse problem in GAN training. The underlying principle being that if a set of images are sampled from the same training set, they are very likely to be completely different, whereas if the generator experiences mode collapse, generated images are likely to be similar. In practice, we only give two samples to the discriminator, which is sufficient to overcome the loss of diversity as suggested in (Lin et al., 2018). The resulting training procedure is summarized in Algorithm 4.

Algorithm 4 Our training algorithm including PacGAN

Require: $\mathscr{D}_{\mathrm{X}}$ the set of unaltered images, $\mathscr{D}_{\mathrm{Y}}$ the set of constraint maps, G the generation network, and D the discrimination function

repeat

sample two mini-batches $\{x_i^a\}_{i=1}^m$, $\{x_i^b\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{X}}$

sample a mini-batch $\{y_i\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{Y}}$

sample two mini-batches $\{z_i^a\}_{i=1}^m$, $\{z_i^b\}_{i=1}^m$ from distribution p_{Z}

update D by stochastic gradient ascent of

$\sum_{i=1}^m \log(\mathrm{D}(x_i^a, x_i^b)) + \log(1 - \mathrm{D}(\mathrm{G}(y_i, z_i^a), \mathrm{G}(y_i, z_i^b)))$

sample a mini-batch $\{y_j\}_{j=1}^n$ from $\mathscr{D}_{\mathrm{Y}}$

sample two mini-batches $\{z_i^a\}_{i=1}^m$, $\{z_i^b\}_{i=1}^m$ from distribution p_{Z}

update G by stochastic gradient descent of

$\sum_{j=1}^n \log(1 - \mathrm{D}(\mathrm{G}(y_j, z_j))) + ||y_j - \mathrm{M}(y_j) \odot \mathrm{G}(y_j, z_j)||_{\mathrm{F}}^2$

until a stopping condition is met

2.4 Experimental setting

We have conducted a series of empirical evaluation to assess the performances of the proposed GAN. Used datasets, evaluation protocol and the tested deep architectures are detailed in this section while Section **??** is devoted to the results presentation.

2.4.1 Datasets

We tested our approach on several datasets listed hereafter. Detailed information on these datasets are provided in the Appendix **??**.

FashionMNIST (Xiao, Rasul, & Vollgraf, 2017) consists of 60,000 28×28 small grayscale images of fashion items, split in 10 classes and is a harder version of the classical MNIST dataset (LeCun et al., 1998). The very small size of the images makes them particularly appropriate for large-scale experiments, such as hyper-parameter tuning.

CIFAR10 (Krizhevsky, 2009) consists of 60,000 32×32 colour images of 10 different and varied classes. It is deemed less easy than MNIST and FashionMnist

CelebA(Liu et al., 2015) is a large dataset of celebrity portraits labeled by identity and a variety of binary features such as eyeglasses, smiling... We use 100,000 images cropped to a size of 128×128, making this dataset appropriate for a high dimension evaluation of our approach in comparison with related work.

Texture is a custom dataset composed of $20,000$ 160×160 patches sampled from a large brick wall texture, as recommended in (Jetchev et al., 2017). It is worth noting that this procedure can

be reproduced on any texture image of sufficient size. Texture is a testbed of our approach on fully-convolutional networks for constrained texture generation task.

Subsurface is a classical dataset in geological simulation (Strebelle, 2002) which consists, similarly to the Texture dataset, of 20,000 160×160 patches sampled from a model of a subsurface binary domain. These models are assumed to have the same properties as a texture, mainly the property of global ergodicity of the data.

To avoid learning explicit pairing of real images seen by the discrimination function with constraint maps provided to the generative network, we split each dataset into training, validation and test sets, to which we add a set composed of constraint maps that should remain unrelated to the three others. In order to do so, a fifth of each set is used to generate the constrained pixel map y by randomly selecting 0.5% of the pixels from a uniform distribution, composing a set of constraints for each of the train, test and validation sets. The images from which these maps are sampled are then removed from the training, testing and validation sets. For each carried experiment the best model is selected based on some performance measures (see Section 2.4.3) computed on the validation set. Finally, reported results are computed on the test set.

2.4.2 Network architectures

We use a variety of GAN architectures in order to adapt to the different scales and image sizes of our datasets. The detailed configuration of these architectures are exposed in Appendix **??**.

For the experiments on the FashionMNIST (Xiao, Rasul, & Vollgraf, 2017), we use a lightweight network for both the discriminator and the generator similarly to DCGAN (Radford, Metz, & Chintala, 2015) due to the small resolution of FashionMnist images.

To experiment on the Texture dataset, we consider a set of fully-convolutional generator architectures based on either dilated convolutions (Yu & Koltun, 2015), which behave well on texture datasets (Ruffino et al., 2019), or encoder-decoder architectures that are commonly used in domain-transfer applications such as CycleGAN (Zhu et al., 2017b). We selected these architectures because they have very large receptive fields without using pooling, which allow the generator to use a large context for each pixel.

We keep the same discriminator across all the experiments with these architectures, the PatchGAN discriminator (Isola et al., 2016), which is a five-layer fully-convolutional network with a sigmoid activation.

The Up-Dil architecture consists in a set of transposed convolutions (the upscaling part), and a set of dilated convolutional layers (Yu & Koltun, 2015), while the Up-EncDec has an upscaling part followed by an encoder-decoder section with skip-connections, where the constraints are downscaled, concatenated to the noise, and re-upscaled to the output size.

The UNet (Ronneberger, Fischer, & Brox, 2015) architecture is an encoder-decoder where skip-connections are added between the encoder and the decoder. The Res architecture is an encoder-decoder where residual blocks (He et al., 2015) are added after the noise is concatenated to the features. The UNet-Res combines the UNet and the Res architectures by including both residual blocks and skip-connections.

Finally, we will evaluate our approach on the Subsurface dataset using the architecture that yields to the best performances on the Texture dataset.

2.4.3 Evaluation

We evaluate our approach based on both the satisfaction of the pixel constraints and the visual quality of sampled images. From the assumption of Gaussian measurement noise (as discussed in

Section **??**), we assess the constraint fulfillment using the following mean square error (MSE)

$$\mathrm{MSE} = \frac{1}{\mathrm{L}} \sum_{i=1}^{\mathrm{L}} \left\| y_i - \mathrm{M}(y_i) \odot \mathrm{G}(y_i, z_i) \right\|_{\mathrm{F}}^2 \tag{2.12}$$

This metric should be understood as the mean squared error of reconstructing the constrained pixel values.

Visual quality evaluation of an image is not a trivial task (Theis, Van Den Oord, & Bethge, 2015). However, Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (Salimans et al., 2016), have been used to evaluate the performance of generative models. We employ FID since the Inception Score has been shown to be less reliable (Barratt & Sharma, 2018). The FID consists in computing a distance between the distributions of relevant features extracted from generated and real samples. To extract these features, a pre-trained Inception v3 (Szegedy et al., 2016) classifier is used to compute the embeddings of the images at a chosen layer. Assuming these embeddings shall follow a normal distribution, the quality of the generated images is assessed in term of a Wasserstein-2 distance between the distribution of real samples and generated ones. Hence the FID writes

$$\mathrm{FID} = ||\mu_r - \mu_g||^2 + \mathrm{T}r(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}), \tag{2.13}$$

where Tr is the trace operator, (μ_r, Σ_r) and (μ_g, Σ_g) are the pairs of mean vector and covariance matrice of embeddings obtained on respectively the real and the generated data. Being a distance between distributions, a small FID corresponds to a good matching of the distributions.

Since the FID requires a pre-trained classifier adapted to the dataset in study, we trained simple convolutional neural networks as classifiers for the FashionMNIST and the CIFAR-10 datasets. For the Texture dataset, since the dataset is not labeled, we resort to a CNN classifier trained on the Describable Textures Dataset (DTD) (**Cimpoi14**), which is a related application domain.

However, since we do not have labels for the Subsurface dataset, we could not train a classifier for this dataset, thus we cannot compute the FID. To evaluate the quality of the generated samples, we use metrics based on a distance between feature descriptors extracted from real samples and generated ones. Similarly to (Ruffino et al., 2019), we rely on a χ^2 distance between the Histograms of Oriented Gradients (HOG) or Local Binary Patterns (LBP) features computed on generated and real images.

Histograms of Oriented Gradients (HOG) (Dalal & Triggs, 2005) and Local Binary Patterns (LBP) (Pietikäinen et al., 2011) are computed by splitting an image into cells of a given radius and computing on each cell the histograms of the oriented gradients for HOGs and of the light level differences for each pixel to the center of the cell for LBPs. Additionally, we consider the domain-specific metric, the connectivity function (Lemmens et al., 2017) which is presented in Appendix **??**.

Finally, we check by visual inspection if the trained model G is able to generate diverse samples, meaning that for a given y and for a set of latent codes $(z_1, ..., z_n) \sim p_{\mathrm{Z}}$, the generated samples $\mathrm{G}(y, z_1), \ldots, \mathrm{G}(y, z_n)$ are visually different.

2.5 Experimental results and application to underground soil generation

2.5.1 Quality-fidelity trade-off

We first study the influence of the λ regularization hyper-parameter on both the quality of the generated samples and the respect of the constraints. We experiment on the FashionMNIST (Xiao,

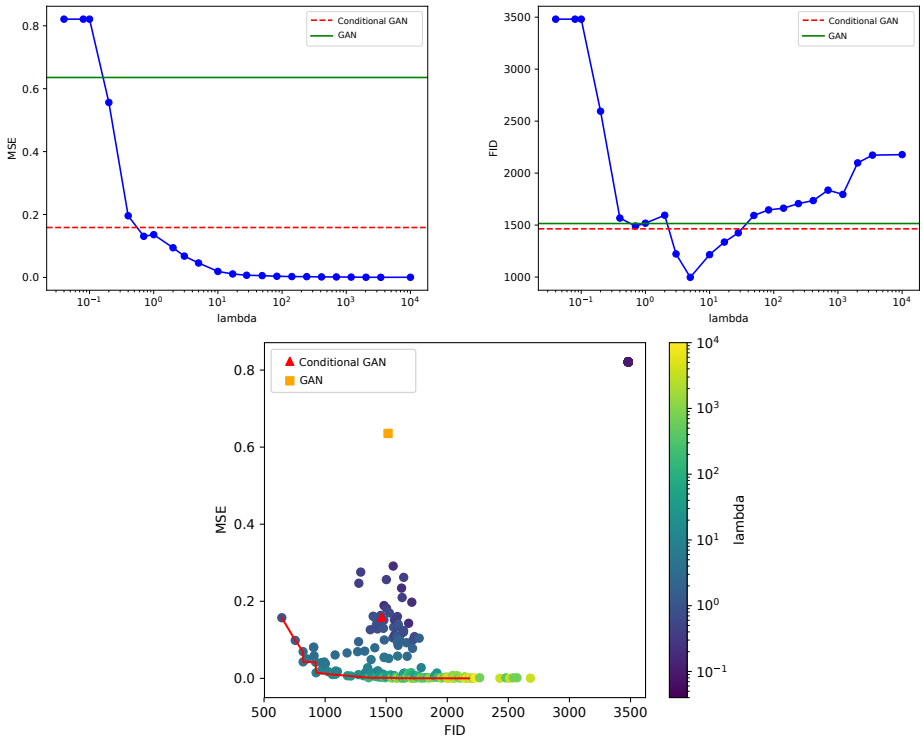


Figure 2.4: Our approach compared to the GAN and CGAN baselines. MSE (left) and FID (right) w.r.t. the regularization parameter λ, MSE w.r.t the FID (bottom).

Rasul, & Vollgraf, 2017) dataset, since such a study requires intensive simulations permitted by the low resolution of FashionMnist images and the used architectures (see Section 2.4.2).

To overcome classical GANs instability, the networks are trained 10 times and the median values of the best scores on the test set at the best epoch are recorded. The epoch that minimizes:

$$\sqrt{\left(\frac{\text{FID}-\text{FID}_{min}}{\text{FID}_{max}-\text{FID}_{min}}\right)^2+\left(\frac{\text{MSE}-\text{MSE}_{min}}{\text{MSE}_{max}-\text{MSE}_{min}}\right)^2}$$

on the validation set is considered as the best epoch, where FID_{min}, MSE_{min}, FID_{max} and MSE_{max} are respectively the lowest and highest FIDs and MSEs obtained on the validation set.

Empirical evidences (highlighted in Figure 2.4) show that with a good choice of λ, the regularization term helps the generator to enforce the constraints, leading to smaller MSEs than when using the CGAN ($\lambda = 0$) without compromising on the quality of generated images. Also, we can note that using the regularization term even leads to a better image quality compared to GAN and CGAN. The bottom panel in Figure 2.4 illustrates that the trade-off between image quality and the satisfaction of the constraints can be controlled by appropriately setting the value of λ. Nevertheless, for small values of λ (less or equal to 10^{-1}), our GAN model fails to learn meaningful distribution of the training images and only generates uniformly black images. This leads to the plateaus on the MSE and FID plots (top panels in Figure 2.4).

2.5.2 Texture generation with fully-convolutional architectures

Fully-convolutional architectures for GANs are widely used, either for domain-transfer applications (Zhu et al., 2017b)(Isola et al., 2016) or for texture generation (Jetchev et al., 2017). In order to evaluate the efficiency of our method on relatively high resolution images, we experiment the

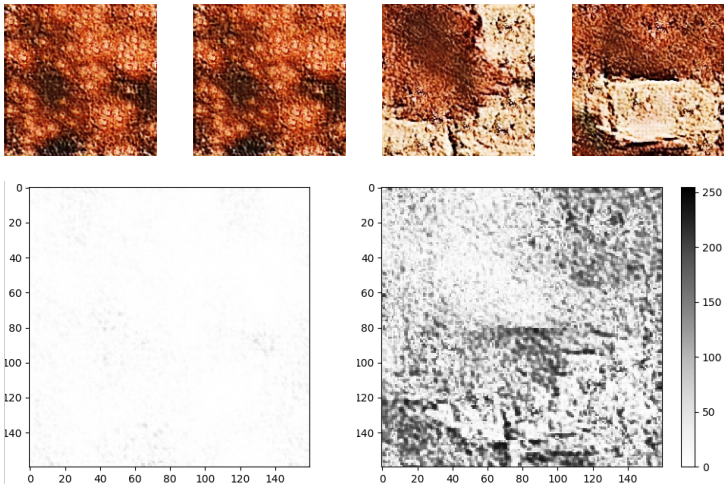


Figure 2.5: An example of a loss of diversity when generating Texture samples with a trained UNetRes network using two different random noises z and a single constraint map y. The two samples on the top left are generated using the classical GAN discriminator whereas the samples on the top right are generated using the PacGAN approach. The loss of diversity is clearly visible on the absolute differences between the greyscaled images (bottom).

fully-convolutional networks described in Section 2.4.2 on a texture generation task using Texture dataset. We investigate the upscaling-dilatation network, the encoder-decoder one and the resnet-like architectures.

Our training algorithm was run for 40 epochs on all reported results. We provide a comparison to CGAN(Mirza & Osindero, 2014) approach by using the selected best architectures. The models are evaluated in terms of best FID (visual quality of sampled images) at each epoch and MSE (conditioning on fixed pixel values). We also compute the FID score of the models at the epochs where the MSE is the lowest. In the other way around, the MSE is reported at epoch when the FID is the lowest. The obtained quantitative results are detailed in Table 2.1.

For the encoder-decoder models, we can notice that the models using ResNet blocks perform better than just using a UNet generator. A trade-off can also be seen between the FID and MSE for the ResNet models and the UNet-ResNet, which could mean that skip-connections help the generator to fulfill the constraints but at the price of lowered visual quality.

Although the encoder-decoder models perform the best, they tend to lose diversity in the generated samples (see Figure 2.5), whereas the upscaling-based models have high FID and MSE but naturally preserve diversity in the generated samples.

Changing the discriminator for a PacGAN discriminator with 2 samples in the encoder-decoder based architectures allows to restore diversity, while keeping the same performances as previously or even increasing the performances for the UNetRes (see Table 2.1).

Table 2.2 compares our proposed approach to CGAN using fully convolutional networks. It shows that our approach is more able to comply with the pixel constraints while producing realistic images. Indeed, our approach outperforms CGAN (see Table 2.2) by a large margin on the respect of conditioning pixels (see the achieved MSE metrics by our UNetPAC or UNetResPAC) and gets close FID performance on the generated samples. This finding is in accordance of the obtained results on FashionMnist experiments.

Model	Best FID	Best MSE	FID at best MSE	MSE at best FID	Diversity
Up-Dil	0.0949	0.4137	1.0360	0.7057	✓
Up-EncDec	0.1509	0.7570	0.2498	0.9809	✓
UNet	0.0442	0.1789	0.0964	0.4559	✗
Res	0.0458	0.0474	0.0590	0.0476	✗
UNetRes	0.0382	0.0307	0.0499	0.0338	✗
ResPAC	**0.0350**	0.0698	0.0466	0.4896	✓
UNetPAC	0.0672	≤ **0.0001**	0.3120	0.2171	✓
UNetResPAC	0.0431	0.0277	**0.0447**	**0.0302**	✓

Table 2.1: Results obtained by the different fully-convolutional architectures on the Texture dataset. We can remark that the encoder-decoder greatly outperforms the upscaling ones and that using the PacGAN technique helps keeping the performance of these models while restoring the diversity in the samples. The bottom part of the table refers to PacGan architectures.

Model	Best FID	Best MSE	FID at best MSE	MSE at best FID
CGAN-ResPAC	**0.0234**	0.1337	**0.0340**	0.2951
CGAN-UNetPAC	0.0518	0.2010	0.0705	0.4828
CGAN-UNetResPAC	0.0428	0.1060	0.0586	0.2250
Ours-ResPAC	0.0350	0.0698	0.0466	0.4896
Ours-UNetPAC	0.0672	≤ **0.0001**	0.3120	0.2171
Ours-UNetResPAC	0.0431	0.0277	0.0447	**0.0302**

Table 2.2: Results obtained by the selected best fully-convolutional architectures on the Texture dataset for both the CGAN approach and our approach.

2.5.3 Extended architectures

We extend the comparison of our approach to CGAN on the CIFAR10 and CelebA datasets (Table 2.3). We investigated the architectures described in Section 2.4.2. All reported results are obtained with the regularization parameter fixed to $\lambda = 1$. We train the networks for 150 epochs using the same dataset split as stated previously in order to keep independence between the images constraint maps. The evaluation procedure remains also unchanged. We use the PacGAN approach to avoid the loss of diversity issues. The experiments on both datasets show that though CGAN provides better results in terms of visual quality, our approach outperforms it according to the respect of the pixel constraints.

	Model	Best FID	Best MSE	FID at best MSE	MSE at best FID
CIFAR-10	CGAN	**2,68**	0.081	**2.68**	0.081
	Ours	3.120	**0.010**	3.530	**0.011**
CelebA	CGAN	**1.34e-4**	0.0209	**1.81e-4**	0.0450
	Ours	2.09e-4	**0.0053**	5.392e-4	**0.0249**

Table 2.3: Results on the CIFAR10 and CelebA datasets. The reported performances compare CGAN to our proposed GAN conditioned on scarce constraint map.

	Model	Best HOG	Best MSE	HOG at best MSE	MSE at best HOG
Subsurface	CGAN	**2.92e-4**	0.2505	**3.06e-4**	1.1550
	Ours	4.31e-4	**0.0325**	5.69e-4	**0.2853**

Table 2.4: Evaluation of the trade-off between the visual quality of the generated samples and the respect of the constraints for the CGAN approach and ours on the Subsurface dataset.

2.5.4 Application to hydro-geology

Finally, we evaluate our approach on the Subsurface dataset. We use the UNetResPAC architecture, since it performed the best on Texture data as exposed in Section 2.5.2. As previously, we simply set the regularization parameter at $\lambda = 1$ and, the network is trained for 40 epochs using the same experimental protocol. To evaluate the trade-off between the visual quality and the respect of the constraints, instead of FID we rather compute distances between visual Histograms of Oriented Gradients (see Section 2.4), extracted from real and generated samples. We also evaluate the visual quality of our approach with a distance between Local Binary Patterns. Indeed, Subsurface application lacks labelled data in order to learn a deep network classifier from which the FID score can be computed.

The obtained results are summarized in Tables 2.4 and 2.5. They are coherent with the previous experiments since the generated samples are diverse and have a low error regarding the constrained pixels. The conditioning have a limited impact on the visual quality of the generated samples and compares well to unconditional approaches (Ruffino et al., 2019). Evaluation of the generated images using the domain-connectivity function highlights this fact on Figures **??** and **??** in the supplementary materials. Also examples of generated images by our approach pictured in Figure **??** (see appendix **??**) show that we preserve the visual quality and honor the constraints.

	Model	Best HOG	Best MSE	Best LBP (radius=1)	Best LBP (radius=2)
Subsurface	CGAN	**2.92e-4**	0.2505	**2.157**	**3.494**
	Ours	4.31e-4	**0.0325**	10.142	16.754

Table 2.5: Evaluation of the visual quality between the CGAN approach and ours on the Subsurface dataset using several metrics.

2.6 Conclusion

In this chapter, we address the task of learning effective generative adversarial networks when only very few pixel values are known beforehand. To solve this pixel-wise conditioned GAN, we model the conditioning information under a probabilistic framework. This leads to the maximization of the likelihood of the constraints given a generated image. Under the assumption of a Gaussian distribution over the given pixels, we formulate an objective function composed of the conditional GAN loss function regularized by a ℓ_2-norm on pixel reconstruction errors. We describe the related optimization algorithm.

Empirical evidences illustrate that the proposed framework helps obtaining good image quality while best fulfilling the constraints compared to classical GAN approaches. We show that, if we include the PacGAN technique, this approach is compatible with fully-convolutional architectures and scales well to large images. We apply this approach to a common geological simulation task

and show that it allows the generation of realistic samples which fulfill the prescribed constraints.

In future work, we plan to investigate other prior distributions for the given pixels as the Laplacian or β-distribtutions. We are also interested in applying the developed approach to other applications or signals such as audio inpainting (Marafioti et al., 2018).

Chapter 3

Conditioning generation with multiple task-specific constraints

Chapter abstract

The problem addressed in this chapter is the generation of polarimetric images using Cyle-Consistent Generative Adversarial Networks (CycleGAN) with constraints derived from the optics of polarimetry. The conducted research is motivated by the increasing popularity of the combination of deep learning frameworks with polarimetric imaging in various domains, including medical imaging and scene analysis. Even if polarimetric imaging has shown improved performances, their robustness may be questioned because of the small size of the training datasets. This issue could be resolved by data augmentation. However, polarization modality is subject to some physical feasibility constraints that could be impeded with classical data augmentation techniques. In this paper we propose a framework based on CycleGAN, integrating the constraints of polarimetric images during training stage. We evaluate the proposed generative model on road scene images. The obtained results achieved an effective generation of physical polarization-encoded images. Further experiments on the task of road object detection show that with the generated images, the detection of cars and pedestrian are improved by up to 9%.

Contents

3.1 Introduction

Generative adversarial networks (GAN) (Goodfellow et al., 2014; Wang, Zheng, & Chuang, 2019) are powerful deep generative models used to implicitly learn complex data distributions and to generate realistic samples from them. In its standard form, a GAN consists of two models: a generator which maps samples drawn from a latent low-dimensional distribution (usually uniform or gaussian distributions) to high-dimensional points expected to follow the sought data distribution, and a discrimination model which discriminate the real samples from the generated ones (Goodfellow et al., 2014). GANs have proven remarkable in various application domains including image generation (Arjovsky, Chintala, & Bottou, 2017), image-to-image translation (Isola et al., 2016; Zhu et al., 2017b; Hoffman et al., 2018) or image attribute manipulation (Antipov, Baccouche, & Dugelay, 2017) to name a few.

Arguably most of the impressive achievements of the GAN were obtained for RGB images. A body of work attempted to extend GAN architectures to other uncommon imaging domains. For instance, some existing methods rely on CycleGAN (Zhu et al., 2017a), an image-to-image translation network, to generate infrared road scenes from RGB counterpart images (Zhang et al., 2018), to produce thermal images for person re-identification (Kniaz et al., 2018) or for infrared image colorization Mehri and Sappa, 2019. In the same vein, Nie et al. (2017) achieved data augmentation in the field of medical imaging by transforming MRI inputs into pseudo-CT images. From another point of view, Sallab et al. (2019) used CycleGANs to produce realistic LiDAR points cloud from simulated ones.

Following the previous stream of work, this paper contributes generative models for non-conventional imaging techniques. Specifically we propose a generative model framework to produce realistic polarimetric images. The significant interest resides in the fact that polarimetric imaging is a rich modality that enables to characterize an object by its reflective properties. Those properties are object specific, hence, they convey strong features to analyse the content of a scene. In a polarimetric image, each pixel encodes information regarding the object's roughness, its orientation and its reflection (Wolff & Andreou, 1995). Applications of polarimetric imaging range from indoor autonomous navigation (Berger, Voorhies, & Matthies, 2017), depth map estimation (Zhu & Smith, 2019), 3D objects reconstruction (Morel et al., 2006) to differentiation of healthy and unhealthy cervical tissues in order to detect cancer at an early stage (Rehbinder et al., 2016). Also, recently, polarization imaging was exploited in autonomous driving applications either to enhance car detection (Fan et al., 2018), road mapping and perception (Aycock et al., 2017) or to detect road objects in adverse weather conditions (Blin et al., 2019). However, these applications are characterized by the reduced size of the available training databases which restrains them from using deep neural networks, thus the need of polarimetric data generation model.

Contrary to RGB, LiDAR or infrared image generation which mostly responded to visual qualitative constraints, unless some learnable knowledge constraints are enforced (see Hu et al., 2018 for pose conditional person image generation), sampling polarization images is more challenging. Indeed, this imaging technique comes with physical admissibility constraints on the pixels of an image. To be physically feasible, each pixel entry of such an image should satisfy some physical constraints related to light polarization principle and to the calibration setup of the acquisition devices. Therefore, we formulate our problem of polarimetric image generation as a CycleGAN learning problem under physical constraints to ensure that the generated images are valid. CycleGANs (Zhu et al., 2017a) enabled to achieve unpaired image-to-image translation with only a few number of images. They allow to circumvent the expensive labelling step by transferring a source labelled dataset to one or multiple target domain (Almahairi et al., 2018) by keeping unchanged the shapes of the source image.

Starting from unpaired sets of RGB and polarimetric images, we propose a learning framework based on CycleGAN and able to handle the physical polarization constraints during training. We demonstrate the effectiveness of our constrained-output CycleGAN on the KITTI dataset (Geiger, Lenz, & Urtasun, 2012) and the Berkeley Deep Drive dataset (BDD100K) Xu et al., 2017, two common datasets used for object detection in road scenes. Using the generated polarization-encoded images to train a deep object detectors, we witness an improvement of the detection performances of cars and pedestrians which are of great interest for autonomous driving applications.

To summarize, the contributions of this paper are:

- as far as our knowledge can go, we propose the first framework for generating physical polarization-encoded images starting from RGB images,
- we propose an extension of CycleGAN which allows to generate polarimetric-encoded images while handling the physical constraints the pixels of the generated image should satisfy,
- when plugged into the training procedure of an object detector for pre-training, the generated images help improving the detection performances.

The remainder of the paper is organized as follows: the polarization formalism and the physical constraints it involves are first presented. Then, the image-to-image translation using Cycle-Consistent GAN is described and a way to take into account these physical constraints during the training process of the CycleGAN for generating polarimetric images is investigated. Experimental evaluations are conducted ; they aim to translate RGB images of KITTI and BDD100K datasets into polarimetric images. Finally, the generated images are exploited to boost the performances of an object detection network. The code for the experiments is available at: `https://anonymous.4open.science/r/4a83820e-9c65-417c-af3a-ab2979d6e2e8/`

3.2 Framework

This section introduces the polarization formalism, the Generative Adversarial Network approach and the CycleGAN principles. It focuses on the formulation of the proposed modelling framework, namely the learning of a CycleGAN with output constraints. A solution approach is detailed and the related learning principle is presented.

3.2.1 Polarization formalism

Light waves can oscillate in different orientations. Polarization represents the direction of propagation of the electrical field of the light wave. When the direction is linear, elliptical or circular, the polarization state is said to be totally polarized. However, it is partially polarized or non polarized when the light wave partly propagates in a random way (Bass et al., 1995).

Polarimetric imaging consists in representing the polarization state of the light wave reflected from each part of the scene. When an unpolarized light wave is being reflected, it becomes partially linearly polarized. Its polarization depends on the normal surface and the refractive index of the material it impinges on. The linear part of the reflected light can be described by measurable parameters and specifically by the linear Stokes vector $S = \begin{bmatrix} S_0 & S_1 & S_2 \end{bmatrix}^\top$ where $S_0 > 0$ represents the total intensity, S_1 the amount of horizontally and vertically linearly polarized light and S_2 the amount of linearly polarized light at $\pm$ 45°. It is important to note that by design, the Stokes vector is physically admissible if and only if the two following conditions are met:

$$S_0 > 0 \quad \text{and} \quad S_0^2 \geqslant S_1^2 + S_2^2 \ . \tag{3.1}$$

Figure 3.1: Example of a polarimetric image. From left to right, the intensities corresponding to the polarizer rotation angles 0°, 45°, 90° and 135°.

One salient physical property, obtained from the Stokes parameters, is the degree of polarization (DOP) (Ainouz et al., 2013) defined by:

$$\mathrm{DOP} = \frac{\sqrt{S_1^2 + S_2^2}}{S_0} \ .$$

The DOP $\in [0,1]$ refers to the amount of polarized light in a wave. It is equal to 1 for a totally polarized light, 0 for unpolarized light and between 0 and 1 for partially polarized light.

Polarization images are accordingly obtained by the computation of the Stokes vector related to each pixel. The acquisition principle is based on a device composed of a polarizer oriented at an angle α between the object and the sensor (Wang, Zheng, & Chuang, 2019). At least three acquisitions with three different angles are required to get the Stokes parameters. The reflected light from the object, represented by the unknown Stokes vector, passes through the rotated polarizer before reaching the camera.

For this work, a Polarcam 4D Technology polarimetric camera was used, enabling to get simultaneously four images respectively obtained with four different linear polarizers oriented at $(\alpha_i)_{i=1:4} = (0°, 45°, 90°, 135°)$. The polarimetric camera measures an intensity I_{α_i} of the scene for each angle α_i. The relationship between the Stokes vector S and the intensities $I(\alpha_i)_{i=1:4}$ reaching the camera is given by:

$$I_{\alpha_i} = \tfrac{1}{2}\begin{bmatrix}1 & \cos(2\alpha_i) & \sin(2\alpha_i)\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\end{bmatrix} \ , \forall i = 1,4$$

that is:

$$I = AS \ , \tag{3.2}$$

where $I = \begin{bmatrix}I_0 & I_{45} & I_{90} & I_{135}\end{bmatrix}^\top$ refers to the four intensities according to each angle of the polarizer $(\alpha_i)_{i=1:4}$ and $A \in \mathbb{R}^{4\times 3}$, to the calibration matrix of the polarization camera, defined as:

$$A = \frac{1}{2}\begin{bmatrix}1 & \cos(2\alpha_1) & \sin(2\alpha_1)\\1 & \cos(2\alpha_2) & \sin(2\alpha_2)\\1 & \cos(2\alpha_3) & \sin(2\alpha_3)\\1 & \cos(2\alpha_4) & \sin(2\alpha_4)\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & 0\\1 & 0 & 1\\1 & -1 & 0\\1 & 0 & -1\end{bmatrix} \ .$$

An example of the different intensities for the same scene is shown in Figure 3.1.

To get the unknown Stokes parameters from the measured intensities (equation 3.2), we require $\tilde{A} = (A^\top A)^{-1}A^\top \in \mathbb{R}^{3\times 4}$ the pseudoinverse of the matrix A. The relationship between S and I

is then defined by:

$$S = \tilde{A}I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} = \begin{bmatrix} I_0 + I_{90} \\ I_0 - I_{90} \\ I_{45} - I_{135} \end{bmatrix} . \tag{3.3}$$

Combining equations (3.2) and (3.3), we attain the following condition

$$I = A\tilde{A}I ,$$

which is satisfied if and only if:

$$I_0 + I_{90} = I_{45} + I_{135} . \tag{3.4}$$

Stokes images should then satisfy two main conditions: the physical admissibility constraints in equation (3.1) and the calibration constraint given by equation (3.4). The generation of new polarimetric images have to comply with these essential constraints.

3.2.2 Unpaired image-to-image translation with CycleGAN

Given two domains X and Y, unpaired image-to-image translation is the task of learning the mapping functions $M_{XY} : X \rightarrow Y$ and $M_{YX} : Y \rightarrow X$ using unpaired samples $x_i \in X$ with $i \in [1..N]$ and $y_j \in Y$ with $j \in [1..M]$. An effective approach to achieve the task is CycleGAN (Zhu et al., 2017a). It consists in learning the two mapping models M_{XY} and M_{YX} by combining the objective function of the standard Generative Adversarial Network (GAN) (Goodfellow et al., 2014) with a Cycle-Consistency loss function. The adversarial cost related to the GAN serves for training the models to generate samples that will match the target domain distribution, while the Cycle-Consistency cost ensures that the learned models are able to correctly reconstruct an original image (of the source domain) from a generated one.

Formally a GAN is composed of a generative model $G : Z \rightarrow X$ which maps a known distribution p_Z, usually normal or uniform, to the unknown distribution p_X of the samples and a discrimination model $D : X \rightarrow [0,1]$. The generator G attempts to fool the discriminator D, which in turn tries to distinguish a real sample from a sample generated by the model G. Learning a GAN amounts to solve the following problem:

$$G^*, D^* = \arg\min_G \max_D L_{GAN}(D,G) ,$$
$$\text{with } L_{GAN}(D,G) = \mathbb{E}_{x \sim p_X}\Big[\log(D(x))\Big] + \mathbb{E}_{z \sim p_Z}\Big[\log(1 - D(G(z)))\Big] ,$$

where $\mathbb{E}$ refers to the expectation.

For its part, CycleGAN learns the two models M_{XY} and M_{YX} by using unpaired real samples $x \in X$ and $y \in Y$ respectively drawn according to the (unknown) distributions p_X and p_Y as input. It also learns two discrimination networks $D_X : X \rightarrow [0,1]$ and $D_Y : Y \rightarrow [0,1]$ able to detect generated samples from real ones in the domains X and Y respectively. CycleGAN relies on the Least-Squares variant of GAN (Mao et al., 2017) and considers the following adversarial costs:

$$L_{GAN}(D_Y, M_{XY}) = \mathop{\mathbb{E}}_{y \sim p_Y}\Big[(D_Y(y) - 1)^2\Big] + \mathop{\mathbb{E}}_{x \sim p_X}\Big[D_Y(M_{XY}(x))^2\Big] ,$$
$$L_{GAN}(D_X, M_{YX}) = \mathop{\mathbb{E}}_{x \sim p_X}\Big[(D_X(x) - 1)^2\Big] + \mathop{\mathbb{E}}_{y \sim p_Y}\Big[D_X(M_{YX}(y))^2\Big] .$$

In order to ensure the cyclic consistency that is both the compositions $M_{XY} \circ M_{YX}$ and $M_{YX} \circ M_{XY}$ are identity functions, a ℓ_1 reconstruction error term is devised for the mapping models:

$$L_{reco}(M_{XY}, M_{YX}) = \mathop{\mathbb{E}}_{y \sim p_Y} ||y - M_{XY}(M_{YX}(y))||_1 + \mathop{\mathbb{E}}_{x \sim p_X} ||x - M_{YX}(M_{XY}(x))||_1 .$$

Gathering all these elements leads to the objective function

$$\begin{aligned} &\mathrm{L}_{Cycle\mathrm{GAN}}(\mathrm{D_X},\mathrm{D_Y},\mathrm{M_{XY}},\mathrm{M_{YX}}) = \\ &\qquad \mathrm{L}_{\mathrm{GAN}}(\mathrm{D_Y},\mathrm{M_{XY}}) + \mathrm{L}_{\mathrm{GAN}}(\mathrm{D_X},\mathrm{M_{YX}}) + \lambda \mathrm{L}_{reco}(\mathrm{M_{XY}},\mathrm{M_{YX}}) \ , \end{aligned} \tag{3.5}$$

where $\lambda > 0$ is an hyper-parameter that controls the influence of the reconstruction term. Training a CycleGAN consists in solving, via alternate gradient descent, the following minmax problem

$$\mathrm{M}^*_{\mathrm{XY}},\mathrm{M}^*_{\mathrm{YX}},\mathrm{D}^*_{\mathrm{X}},\mathrm{D}^*_{\mathrm{Y}} = \arg\min_{\substack{\mathrm{M_{XY}}\\ \mathrm{M_{YX}}}} \max_{\substack{\mathrm{D_X}\\ \mathrm{D_Y}}} \mathrm{L}_{Cycle\mathrm{GAN}}(\mathrm{D_X},\mathrm{D_Y},\mathrm{M_{XY}},\mathrm{M_{YX}}) \ . \tag{3.6}$$

The full learning procedure of a CycleGAN is sketched in Algorithm 5.

Algorithm 5 CycleGAN training algorithm

Require: X and Y two unpaired datasets, $\mathrm{M_{XY}}$ and $\mathrm{M_{YX}}$ the mapping networks, $\mathrm{D_X}$ and $\mathrm{D_Y}$ the discrimination models, m the mini-batch size

repeat

sample a mini-batch $\{x_i\}_{i=1}^m$ from X

sample a mini-batch $\{y_i\}_{i=1}^m$ from Y

update $\mathrm{D_X}$ by stochastic gradient descent of

$\sum_{i=1}^m (\mathrm{D_X}(x_i)-1)^2 + (\mathrm{D_X}(\mathrm{M_{YX}}(y_i)))^2$

update $\mathrm{D_Y}$ by stochastic gradient descent of

$\sum_{i=1}^m (\mathrm{D_Y}(y_i)-1)^2 + (\mathrm{D_Y}(\mathrm{M_{XY}}(x_i)))^2$

sample a mini-batch $\{x_i\}_{i=1}^m$ from X

sample a mini-batch $\{y_i\}_{i=1}^m$ from Y

update $\mathrm{M_{XY}}$ by stochastic gradient descent of

$\sum_{i=1}^n (\mathrm{D_Y}(\mathrm{M_{XY}}(x_i))-1)^2 + \lambda(||x_i - \mathrm{M_{YX}}(\mathrm{M_{XY}}(x_i))||_1$

$+||y_i - \mathrm{M_{XY}}(\mathrm{M_{YX}}(y_i))||_1)$

update $\mathrm{M_{YX}}$ by stochastic gradient descent of

$\sum_{i=1}^n (\mathrm{D_X}(\mathrm{M_{YX}}(y_i))-1)^2 + \lambda(||x_i - \mathrm{M_{YX}}(\mathrm{M_{XY}}(x_i))||_1$

$+||y_i - \mathrm{M_{XY}}(\mathrm{M_{YX}}(y_i))||_1)$

until a stopping condition is met

3.3 Conditioning domain-transfer approaches

As discussed above, our main goal is to learn a generative model able to produce realistic polarization-based images starting from RGB images. For the sake, we adopt the image-to-image translation framework and extend it to account for the constraints a polarimetric image must fulfill.

To generate a polarimetric image from an RGB image, we propose to use the CycleGAN approach to learn the translation models $\mathrm{M_{XY}}$ and $\mathrm{M_{YX}}$ between X the domain of the polarimetric images and Y the RGB image domain. Let $\hat{\mathrm{I}} \in \mathbb{R}^4$ be the intensity vector associated to a pixel of a generated polarimetric image. To be physically admissible, each pixel has to satisfy the admissibility constraints (3.1) and the calibration constraint (3.4). We refer in the sequel these polarimetric constraints by $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_3$ as follows:

$$\begin{aligned} \mathscr{C}_1 &: \mathrm{I} = \mathrm{AS} \ , \\ \mathscr{C}_2 &: \mathrm{S}_0^2 \geqslant \mathrm{S}_1^2 + \mathrm{S}_2^2 \ , \\ \mathscr{C}_3 &: \mathrm{S}_0 > 0 \ . \end{aligned}$$

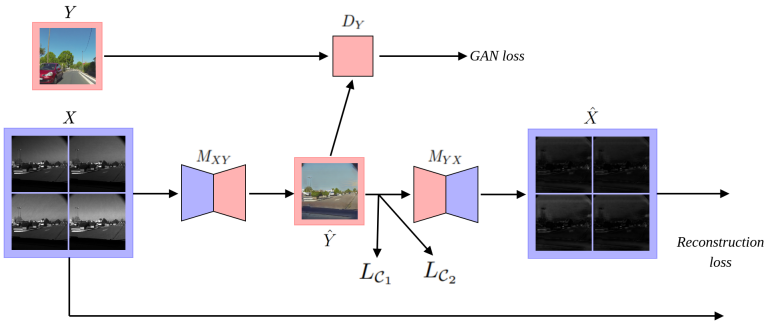


Figure 3.2: Overview of the CycleGAN training process extended with $L_{\mathscr{C}_1}$ and $L_{\mathscr{C}_2}$.

By design, the first component of the Stokes vector is always positive as it represents the total intensity reflected from an object. As the last layer of the generation models customary uses the tangent hyperbolic as activation function, each output intensity $\hat{I}$ is within the range $]-1,1[$ which we scale to $]0,255[$. Hence $\hat{S}_0 = \hat{I}_0 + \hat{I}_{90}$ (see equation (3.3)) is ensured to be strictly positive. Therefore, constraint $\mathscr{C}_3$ can be deemed satisfied for the real and the generated polarimetric images. To handle the remaining constraints $\mathscr{C}_1$ and $\mathscr{C}_2$, one could resort to the Lagrangian dual of CycleGAN optimization problem (3.6) subject to these constraints. However, this may be computationally expensive, as it requires to entirely optimize four neural networks (respectively the discrimination and the mapping network models) in an inner loop of a dual ascent algorithm. Moreover the overall optimization procedure may not be stable because of the minmax game involved in the CycleGAN learning.

In order to derive an efficient algorithm to learn CycleGAN under output constraints, we introduce a relaxation of the problem. Instead of strictly enforcing the constraints, we measure how far the generated image pixels are from the feasibility domain through additional cost functions we attempt to minimize. For the constraint $\mathscr{C}_1$, a ℓ_2 distance between the generated image M_{YX} and $A\hat{S}$ is proposed. It reads

$$L_{\mathscr{C}_1} = \mathop{\mathbb{E}}_{y\sim p_Y} ||M_{YX}(y) - A\hat{S}||_2 \ ,$$

with $\hat{S} = \begin{bmatrix}\hat{S_0} & \hat{S_1} & \hat{S_2}\end{bmatrix}^\top$ the Stokes vector calculated from the generated image by M_{YX} using equation (3.3). Similarly, to enforce the constraint $\mathscr{C}_2$, a rectified linear penalty $L_{\mathscr{C}_2}$ is considered. It is defined by:

$$L_{\mathscr{C}_2} = \mathop{\mathbb{E}}_{y\sim p_Y} \max\left(\hat{S_1}^2 + \hat{S_2}^2 - \hat{S_0}^2, 0\right) \ .$$

The loss $L_{\mathscr{C}_1}$ translates the respect of the acquisition conditions according to the calibration matrix A while $L_{\mathscr{C}_2}$ is related to the physical admissibility constraint on the deduced Stokes vectors from the generated image.

Gathering all these elements, we train our CycleGAN under physical constraints, by optimizing the following objective function:

$$L_{final} = L_{CycleGAN} + \mu L_{\mathscr{C}_1} + \nu L_{\mathscr{C}_2} \ . \tag{3.7}$$

The non-negative hyper-parameters μ and $\nu \in \mathbb{R}^+$ control respectively the balance of admissibility and calibration constraints according to the CycleGAN loss $L_{CycleGAN}$ (see equation (3.5)). As the values of $L_{\mathscr{C}_1}$ and $L_{\mathscr{C}_2}$ are computed pixel-wisely, we consider their averages over the whole image in the objective function. The training principle of the proposed generative model is illustrated in Figure 3.2.

Class	Train	Val	Test
Images	3861	1248	509
car	19587	3793	2793
person	2049	294	161
bike	16	35	3
motorbike	52	4	5

Table 3.1: Polarimetric dataset features. The bottom rows indicate the total number of instances within each class.

Figure 3.3: Examples of images in the polarimetric dataset (Blin et al., 2020). Only the intensities I_0 are shown here.

3.4 Experimental evaluation

Hereafter, the experimental setup, including the image generation procedure and its evaluation, is presented.

3.4.1 Polarimetric images generation using CycleGAN

To conduct the experiments, we rely on the polarimetric dataset presented in (Blin et al., 2020) whose details are summarized in Table 3.1. From this dataset we select 2485 unpaired images from each domain (RGB and polarimetry). Example instances are shown in Figures 3.3 and 3.4 for polarimetric and RGB images respectively. The polarimetric images are of dimension $500 \times 500 \times 4$. The latter dimension is due to the four intensities acquired by the camera, namely I_0, I_{45}, I_{90} and I_{135}. The RGB images are of dimension $906 \times 945 \times 3$.

Our CycleGAN was trained for 400 epochs on randomly cropped patches of size 200×200. As for the constraints, we found experimentally that setting the hyper-parameters $\mu = 1$ and $\nu = 1$ in equation (3.7) provides the best performances. As for the original CycleGAN, the hyper-parameter λ, controlling the reconstruction cost, was set to $\lambda = 10$. The learning rate is decreased linearly

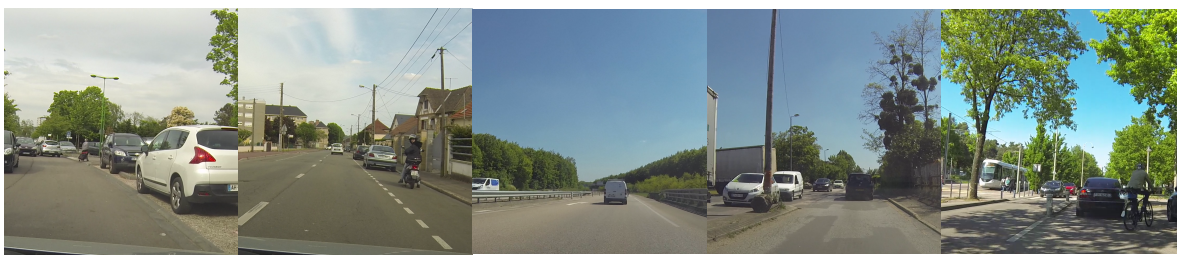


Figure 3.4: Examples of images in the RGB dataset.

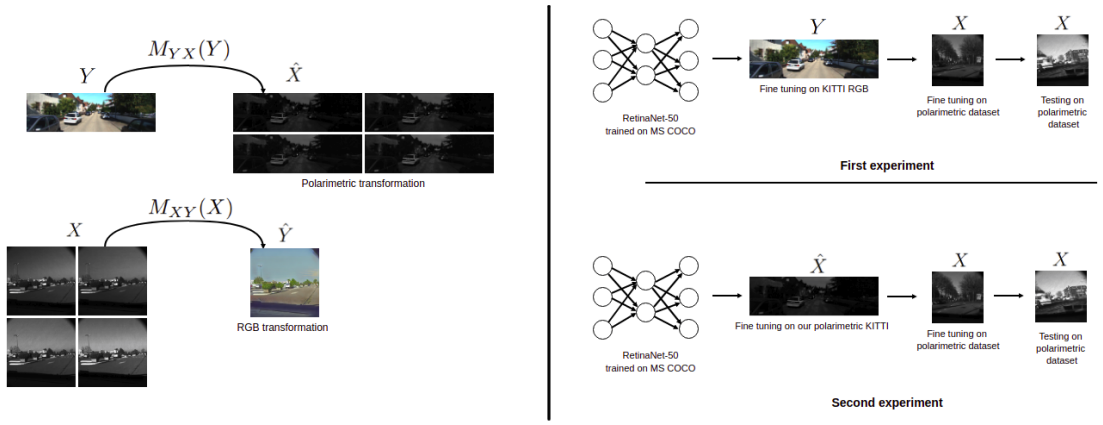


Figure 3.5: Setup of the detection evaluation experiment. The procedure is illustrated with the KITTI dataset and straightforwardly extends to the BDD100K dataset.

from 2×10^{-4} to 2×10^{-6} during the 400 training epochs.

To evaluate the effectiveness of our trained generative model, we consider KITTI and BDD100K (only using daytime images since polarimetry fails to characterize objects during nighttime) which often serve as testbed in applications related to road scene object detection. The constrained-output CycleGAN we train is used to transfer RGB images from KITTI and BDD100K to the polarimetric domain. The resulting datasets are denoted respectively as Polar-KITTI and Polar-BDD100K. Since the CycleGAN architecture is fully convolutional, it has no requirement on the size of the input image. Therefore, even if the model was trained on 200×200 patches, it scales straightforwardly to the images of size 1250×375 from KITTI and of size 1280×720 from BDD100K datasets.

To assess whether or not fulfilling the physical constraints is paramount, we investigate a variant of Polar-KITTI and Polar-BDD100K: we learn a standard unconstrained CycleGAN based on the same unpaired RGB/polarimetric images. It is worth mentioning that the so generated polarization-encoded images do not mandatory satisfy the feasibility constraints.

3.4.2 Evaluation of the generated images

In order to assert the ability of the generated Polar-KITTI and Polar-BDD100K datasets to preserve the relevant features for road scene applications, we train a detection network following the setup in Figure 3.5. For this experiment, a RetinaNet-50 (Lin et al., 2017) pre-trained on the MS COCO dataset (Lin et al., 2014) is fine-tuned in two different settings. In the first setup the detection model is fine-tuned based on the original RGB KITTI (or BDD100K) while the second experimental setting considers the fine-tuning on the generated polarimetric images from KITTI (Polar-KITTI) or BDD100K (Polar-BDD100K) datasets. Afterwards the final detection models are obtained in both settings by a final fine-tuning on the real polarimetric dataset (see Table 3.1). The same experiments were carried out for the unconstrained variant of the generated images.

Overall, the trained CycleGANs and detection networks under these settings are evaluated in qualitative and quantitative ways. The end goal is to check: (i) the ability of the generated images to help learning polarimetry-based features for object detection, and (ii) the influence of respecting the polarimetric feasibility constraints on detection performances.

We measure the visual quality of the generated images by computing the classical Fréchet Inception Distance (Heusel et al., 2017). Computing this distance requires to extract visual features

Figure 3.6: Examples of polarimetric image reconstruction. From left to right: I_0, I_{45}, I_{90} and I_{135} ground truth, RGB image and I_0, I_{45}, I_{90} and I_{135} generated from RGB image.

from each set of images (real and generated) using a pre-trained deep neural network (usually an Inception v3 (Szegedy et al., 2016) network pre-trained on ImageNet (Deng et al., 2009)) and to evaluate the Fréchet (or Wasserstein) distance between the distributions of these features, which are assumed be Gaussian distributions. We calculate this distance using 509 images from each generated polarimetric dataset and from the test set as described in Table 3.1.

As feature extractor, since the classical Inception v3 network is not adapted to polarimetric images, we use the convolutional part of a polarimetry-adapted RetinaNet detection network (Blin et al., 2019), which has been trained on the MS-COCO dataset and fine-tuned on a real polarimetric dataset. In order to evaluate the improvements in the detection, we compute the error rate evolution ER_o. The improvement ER_o on the detection of the object o is given by:

$$\mathrm{ER}_o = \frac{1 - \mathrm{AP}_o^p - (1 - \mathrm{AP}_o^{\mathrm{RGB}})}{1 - \mathrm{AP}_o^{\mathrm{RGB}}} \quad ,$$

where $\mathrm{AP}_o^{\mathrm{RGB}}$ and AP_o^p respectively denote the average precision for object o detection in RGB and in polarimetric images.

3.4.3 Results and discussion

First we evaluate whether the generated images are qualitatively coherent. For the sake, we reconstruct the polarimetric images from their RGB generation, wich refers to $\mathrm{M}_{\mathrm{XY}} \circ \mathrm{M}_{\mathrm{YX}}$ in subsection 2.2. The reconstruction of these RGB images is shown in Figure 3.6.

As for the constraints, Table 3.2 shows how including them to the CycleGAN's loss helps generating images which better fulfill the physical polarimetric properties at the pixel scale. The errors related to the constraints $\mathscr{C}_1$ and $\mathscr{C}_2$ on generated images using our approach are consistent with the observed errors on the real images, whereas the unconstrained approach yields poor results. Obviously, constraint $\mathscr{C}_3$ is met for all generated images thanks to the tanh activation at the last layer of the generative models. Additionally, the obtained Fréchet Inception Distances are of **6022.7** for the unconstrained CycleGAN and **4485.1** for our approach[1], which indicates that taking

[1]Note that the scale of the FID scores computed with the pre-trained RetinaNet is larger than when using a pre-trained Inception v3 network.

Datasets	$\mathscr{C}$	Mean	Median
Real polar	$\mathscr{C}_1$	0.06 ± 0.04	0.04
	$\mathscr{C}_2$	2.47 ± 7.11%	0.48%
	$\mathscr{C}_3$	0%	0%
Generated polar no $\mathscr{C}$	$\mathscr{C}_1$	0.26 ± 0.19	0.23
	$\mathscr{C}_2$	27.31 ± 43.5%	2.15%
	$\mathscr{C}_3$	0%	0%
Generated polar with $\mathscr{C}$	$\mathscr{C}_1$	0.12 ± 0.04	0.12
	$\mathscr{C}_2$	1.55 ± 3.36%	0.14%
	$\mathscr{C}_3$	0%	0%

Table 3.2: Evaluation of the constraint fulfillment using the designed losses $L_{\mathscr{C}_1}$ and $L_{\mathscr{C}_2}$ at the pixel scale. Here, the column $\mathscr{C}$ indicates the evaluated constraint. $\mathscr{C}_1$ refers to the constraints I = AS, $\mathscr{C}_2$ to $S_0^2 \geqslant S_1^2 + S_2^2$ and $\mathscr{C}_3$ to $S_0 > 0$. The mean and the median of the percentage of pixels in an image that do not fulfill the constraints $\mathscr{C}_2$ and $\mathscr{C}_3$ are computed. Regarding the constraint $\mathscr{C}_1$, we compute the mean and the median of $||I - AS||/(||I|| + ||AS||)$.

the constraints into account improves visual and physical quality of the generated samples.

Next, we show the benefit of the generated images in object detection task, enabling to verify that objects in them are globally physically coherent. The RetinaNet-based detection model were trained according to the setups described in Section 3.4.2 and the obtained detection performances in term of mean average precision (mAP) are summarized in Table 3.3. We choose not to evaluate the bike and motorbike detection performances as the polarimetric dataset does not contain enough objects of those two classes.

As we can see in Table 3.3, using the generated polarimetric images improves the detection performance in real polarimetric images. The improvement is substantial for car and pedestrian detection. We achieve an improvement of 4% for car detection and of 12% for pedestrian detection which leads to a global improvement of 9% in the detection, using Polar-KITTI with constraints. Similarly for Polar-BDD100K dataset, we notice an improvement of 10% for pedestrian detection which leads to an increased mAP of 5% (pedestrians and cars). However, we shall notice that for BDD100K similar detection performances are obtained either for RGB or polarimetric images and this is due to the fact that generated images using CycleGANs don't perform well on small objects. To verify that, we compared the evolution of the detections scores while setting a minimal area to the bounding boxes to be detected. The results of this experiment are shown for the training ingluding the Polar-BDD100K and the RGB BDD100K in Figure 3.7.

The results of this experiment showed that when the minimal area of bounding boxes increases the AP of car regarding the training including Polar-BDD100K overcomes the one including RGB BDD100K. We can thus conclude that the limit of this work is the low quality of the small objects in the generated images.

Databases used	Class	Test	ER_o	Databases used	Class	Test	ER_o
KITTI RGB	person	0.663	N/A	BDD100K RGB	person	0.736	N/A
+ real polar	car	0.785	N/A	+ real polar	car	**0.821**	N/A
*m*AP		0.724	N/A	*m*AP		0.778	N/A
Polar-KITTI no $\mathscr{C}$	person	0.673	-0.03	Polar-BDD100K no $\mathscr{C}$	person	0.720	0.06
+ real polar	car	0.786	-0.01	+ real polar	car	0.816	0.03
*m*AP		0.730	-0.02	*m*AP		0.768	0.05
Polar-KITTI with $\mathscr{C}$	person	**0.704**	-0.12	Polar-BDD100K with $\mathscr{C}$	person	**0.762**	-0.10
+ real polar	car	**0.794**	-0.04	+ real polar	car	0.815	0.03
*m*AP		**0.749**	-0.09	*m*AP		**0.789**	-0.05

Table 3.3: Comparison of the detection performance after the two successive fine-tunings. RetinaNet-50 pre-trained on MS COCO is the baseline of all the experiments. The first row refers to the RetinaNet-50 fine-tuned on KITTI or BDD100K RGB. The second row refers to the fine-tuning on Polar-KITTI or Polar-BDD100K without constraints while the bottom row represents the detection models fine-tuned on Polar-KITTI or Polar-BDD100K with the constraints. All these models are finally fine-tuned on the real polarimetric dataset.

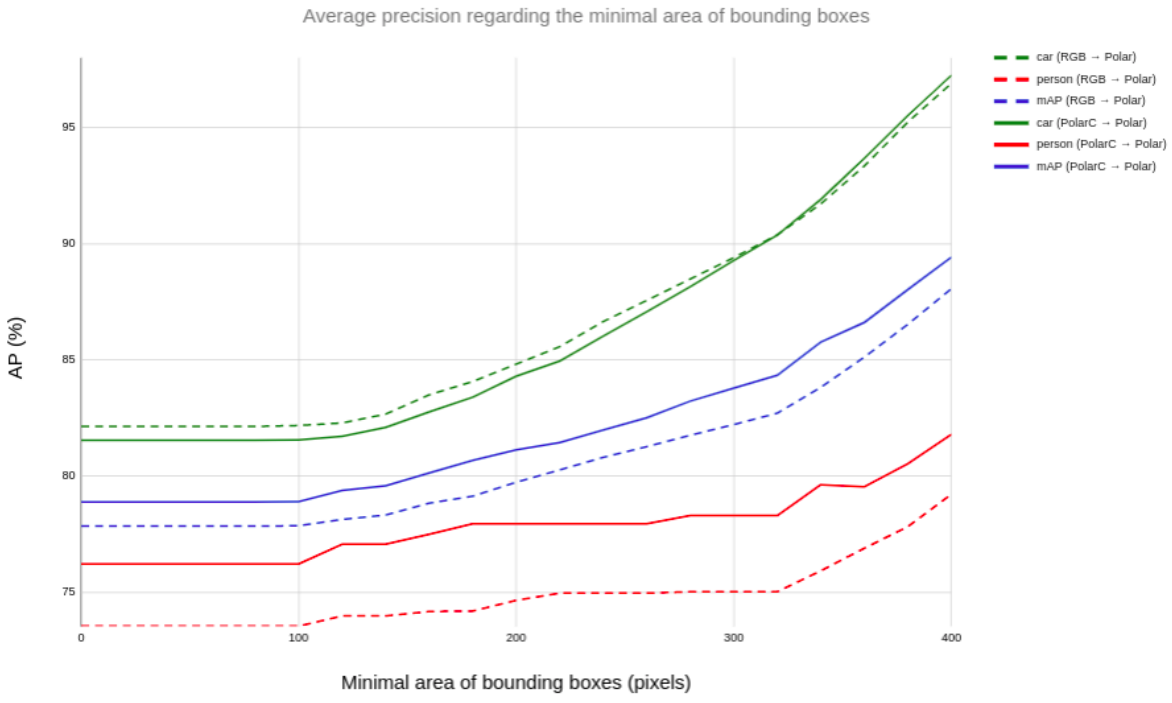


Figure 3.7: Evolution of the average precision when setting a minimal area of the bounding boxes to be detected. Here green lines refer to the evolution of cars' detection, blue lines to the evolution of the *m*AP and red lines to the evolution of person's detection. The dashed lines refer to the training including the BDD100K RGB and the solid lines to the training including Polar-BDD100K.

3.5 Conclusion and future work

In this work, we proposed an efficient way to generate realistic polarimetric images subject to physical admissibility constraints. An adapted CycleGan is used to achieve the generation of pixel-wise physical images. To train the proposed output-constrained CycleGAN, we combined the standard CycleGAN's objective function with two designed cost functions in order to handle the feasibility constraints related to each polarization-encoded pixel in the image. With the proposed generative model, we successfully translated RGB images from road scenes to polarimetric images showing an enhancement of the detection performances. Future work would consist in improving the quality of the small objects in generated images. It would also be interesting to extend the generation of polarimetric images to other domains such as medical and Synthetic-Aperture Radar (SAR) imaging. Extension of the generation procedure to road scene images under adverse weather conditions may help improving object detection in these situations. From the optimization side, we plan to directly address the genuine constrained CycleGAN problem instead of its proposed relaxation.

END OF COPY PASTED CONTENT

3.6 Introduction

Formulation as a constrained optimization problem
- Reformulation of CycleGAN as a constrained optimization problem
- Relaxation of the constraints
- Ici, expérimenter sur des datasets artificiels ?

3.7 Conditioning domain-transfer approaches

3.8 Proximal method for non-Euclidean output space

Travail sur le proximal ?
- Envelope theorem application

3.9 Application to RGB to Polarimetric domain transfer

Introduction to polarimetry-specific physical constraints (briefly, no need to write a physics essay)
- Reformulation as constraints on the output space
- Relaxations : L_2 term + rectified term
- Dataset, Evaluation
- Experiments and results

3.10 Conclusion

Relaxation of the constraints works even when a lot of constraints are applied
- The application to the polarimetric dataset works
- Future works : using adapted metrics for the non-euclidean outspace X

Chapter 4

Conclusion and Perspectives

Bibliography

Ainouz, Samia et al. (2013). "Adaptive Processing of Catadioptric Images Using Polarization Imaging: Towards a Pola-Catadioptric Model". In: *Optical engineering* 52.3, p. 037001 (cit. on p. 33).

Almahairi, Amjad et al. (2018). "Augmented Cyclegan: Learning Many-to-Many Mappings from Unpaired Data". In: *arXiv preprint arXiv:1802.10151* (cit. on p. 32).

Antipov, Grigory, Moez Baccouche, and Jean-Luc Dugelay (May 30, 2017). "Face Aging With Conditional Generative Adversarial Networks". In: arXiv: 1702.01983 [cs]. URL: http://arxiv.org/abs/1702.01983 (visited on 05/19/2020) (cit. on pp. 1, 32).

Arjovsky, Martin and Léon Bottou (2017). "Towards Principled Methods for Training Generative Adversarial Networks". In: URL: https://arxiv.org/pdf/1701.04862.pdf (cit. on p. 9).

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein GAN". In: URL: https://arxiv.org/pdf/1701.07875.pdf (cit. on pp. 9, 12–14, 32).

Aycock, Todd M et al. (Mar. 7, 2017). "Polarization-Based Mapping and Perception Method and System". In: (cit. on p. 32).

Barratt, Shane and Rishi Sharma (2018). *A Note on the Inception Score*. URL: https://github.com/ (cit. on pp. 16, 25).

Bass, Michael et al. (1995). *Handbook of Optics*. Vol. 2. McGraw-Hill New York (cit. on p. 33).

Bellemare, Marc G. et al. (May 30, 2017). "The Cramer Distance as a Solution to Biased Wasserstein Gradients". In: arXiv: 1705.10743 [cs, stat]. URL: http://arxiv.org/abs/1705.10743 (visited on 05/21/2020) (cit. on pp. 13, 14).

Bengio, Yoshua et al. (May 23, 2014). "Deep Generative Stochastic Networks Trainable by Backprop". In: arXiv: 1306.1091 [cs]. URL: http://arxiv.org/abs/1306.1091 (visited on 05/22/2020) (cit. on p. 14).

Berger, Kai, Randolph Voorhies, and Larry H Matthies (2017). "Depth from Stereo Polarization in Specular Scenes for Urban Robotics". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1966–1973 (cit. on p. 32).

Bińkowski, Miko\ laj et al. (Jan. 2018). "Demystifying MMD GANs". In: URL: http://arxiv.org/abs/1801.01401 (cit. on pp. 13, 16).

Blin, Rachel et al. (2019). "Road Scenes Analysis in Adverse Weather Conditions by Polarization-Encoded Images and Adapted Deep Learning". In: *22nd International Conference on Intelligent Transportation Systems*. arXiv: 1910.04870 [cs.CV] (cit. on pp. 32, 40).

Blin, Rachel et al. (2020). "A New Multimodal RGB and Polarimetric Image Dataset for Road Scenes Analysis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 216–217 (cit. on pp. 37, 38).

Bora, Ashish, Eric Price, and Alexandros G Dimakis (2018). "AmbientGAN: Generative Models from Lossy Measurements". In: *International Conference on Learning Representations (ICLR)* (cit. on pp. 18, 19).

Borji, Ali (2018). "Pros and Cons of GAN Evaluation Measures". In: URL: https://arxiv.org/pdf/1802.03446.pdf (cit. on p. 16).

Brock, Andrew, Jeff Donahue, and Karen Simonyan (Sept. 2018). "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: URL: http://arxiv.org/abs/1809.11096 (cit. on pp. 9, 10).

Brown, Lawrence D (1986). "Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory". In: Ims (cit. on p. 22).

Burt, Peter J and Edward H Adelson (1983). "The Laplacian Pyramid as a Compact Image Code". In: p. 9 (cit. on p. 14).

Candes, Emmanuel J. and Terrence Tao (Dec. 2005). "Decoding by Linear Programming". In: *IEEE Transactions on Information Theory* 51.12, pp. 4203–4215. DOI: 10.1109/TIT.2005.858979 (cit. on p. 20).

Dalal, N. and B. Triggs (2005). "Histograms of Oriented Gradients for Human Detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE, pp. 886–893. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.177. URL: http://ieeexplore.ieee.org/document/1467360/ (cit. on p. 26).

Danihelka, Ivo et al. (May 15, 2017). "Comparison of Maximum Likelihood and GAN-Based Training of Real NVPs". In: arXiv: 1705.05263 [cs]. URL: http://arxiv.org/abs/1705.05263 (visited on 05/23/2020) (cit. on p. 8).

Demir, Ugur and Gozde Unal (Mar. 2018). "Patch-Based Image Inpainting with Generative Adversarial Networks". In: URL: http://arxiv.org/abs/1803.07422 (cit. on p. 19).

Dempster, A. P., N. M. Laird, and D. B. Rubin (Sept. 1977). "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x. URL: http://doi.wiley.com/10.1111/j.2517-6161.1977.tb01600.x (cit. on p. 5).

Deng, Jia et al. (2009). *ImageNet: A Large-Scale Hierarchical Image Database*. URL: http://www.image-net.org. (cit. on pp. 16, 39).

Denton, Emily et al. (June 18, 2015). "Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks". In: arXiv: 1506.05751 [cs]. URL: http://arxiv.org/abs/1506.05751 (visited on 05/22/2020) (cit. on p. 14).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (Feb. 27, 2017). "Density Estimation Using Real NVP". In: arXiv: 1605.08803 [cs, stat]. URL: http://arxiv.org/abs/1605.08803 (visited on 05/11/2020) (cit. on pp. 4, 6).

Dziugaite, Gintare Karolina, Daniel M. Roy, and Zoubin Ghahramani (May 14, 2015). "Training Generative Neural Networks via Maximum Mean Discrepancy Optimization". In: arXiv: 1505.03906 [cs, stat]. URL: http://arxiv.org/abs/1505.03906 (visited on 05/25/2020) (cit. on p. 13).

Fan, Wang et al. (2018). "Polarization-Based Car Detection". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3069–3073 (cit. on p. 32).

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 3354–3361 (cit. on p. 32).

Goodfellow, Ian (2016). "NIPS 2016 Tutorial: Generative Adversarial Networks". In: URL: https://arxiv.org/pdf/1701.00160.pdf (cit. on p. 10).

Goodfellow, Ian J et al. (2014). "Generative Adversarial Nets". In: URL: https://arxiv.org/pdf/1406.2661.pdf (cit. on pp. 1, 3, 4, 7, 9, 10, 14, 22, 32, 35).

Gretton, Arthur et al. (2012). "A Kernel Two-Sample Test". In: *Journal of Machine Learning Research* 13.25, pp. 723–773. URL: http://jmlr.org/papers/v13/gretton12a.html (visited on 05/23/2020) (cit. on p. 13).

Gulrajani, Ishaan et al. (2017). "Improved Training of Wasserstein GANs". In: URL: https://arxiv.org/pdf/1704.00028.pdf (cit. on p. 13).

He, Kaiming et al. (2015). "Deep Residual Learning for Image Recognition". In: URL: https://arxiv.org/pdf/1512.03385.pdf%20http://image-net.org/challenges/LSVRC/2015/ (cit. on p. 25).

Heusel, Martin et al. (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: URL: https://arxiv.org/pdf/1706.08500.pdf (cit. on pp. 9, 14, 16, 25, 39).

Hindupur, Avinash (2017). *The GAN Zoo*. URL: https://github.com/hindupuravinash/the-gan-zoo (visited on 05/21/2020) (cit. on p. 10).

Hoffman, Judy et al. (July 10–15, 2018). "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 1989–1998. URL: http://proceedings.mlr.press/v80/hoffman18a.html (cit. on p. 32).

Hu, Zhiting et al. (2018). *Deep Generative Models with Learnable Knowledge Constraints*. 10522–10533. URL: http://papers.nips.cc/paper/8250-deep-generative-models-with-learnable-knowledge-constraints (cit. on p. 32).

Ioffe, Sergey, Christian Szegedy, and Sergey Ioffe (Feb. 2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: URL: https://arxiv.org/pdf/1502.03167.pdf%20http://arxiv.org/abs/1502.03167 (cit. on p. 14).

Isola, Phillip et al. (2016). "Image-to-Image Translation with Conditional Adversarial Networks". In: URL: https://arxiv.org/pdf/1611.07004v1.pdf (cit. on pp. 11, 18, 25, 27, 32).

Jetchev, Nikolay et al. (2017). "Texture Synthesis with Spatial Generative Adversarial Networks". In: URL: https://arxiv.org/pdf/1611.08207.pdf (cit. on pp. 18, 24, 27).

Julien, Rabin et al. (2011). *Wasserstein Barycenter and Its Application to Texture Mixing*, pp. 435–446. URL: https://hal.archives-ouvertes.fr/hal-00476064 (cit. on p. 16).

Kang, Yuhao, Song Gao, and Robert E. Roth (May 4, 2019). "Transferring Multiscale Map Styles Using Generative Adversarial Networks". In: *International Journal of Cartography* 5.2-3, pp. 115–141. ISSN: 2372-9333, 2372-9341. DOI: 10.1080/23729333.2019.1615729. URL: https://www.tandfonline.com/doi/full/10.1080/23729333.2019.1615729 (visited on 05/19/2020) (cit. on p. 1).

Kantorovich, L. V. and G. P. Akilov (1982). *Functional Analysis*. Elsevier. 605 pp. ISBN: 978-1-4831-3825-1 (cit. on p. 13).

Karras, Tero et al. (2017). *Progressive Growing of GANs for Improved Quality, Stability and Variation*. URL: https://youtu.be/G06dEcZ-QTg. (cit. on p. 14).

Karras, Tero et al. (Mar. 23, 2020). "Analyzing and Improving the Image Quality of StyleGAN". In: arXiv: 1912.04958 [cs, eess, stat]. URL: http://arxiv.org/abs/1912.04958 (visited on 05/21/2020) (cit. on p. 10).

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". In: URL: https://arxiv.org/pdf/1312.6114.pdf (cit. on pp. 4, 5, 14).

Kingma, Durk P. and Prafulla Dhariwal (2018). *Glow: Generative Flow with Invertible 1x1 Convolutions*. 10236–10245. URL: http://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-convolutions (cit. on pp. 4, 6).

Kniaz, Vladimir V. et al. (Sept. 2018). "ThermalGAN: Multimodal Color-to-Thermal Image Translation for Person Re-Identification in Multispectral Dataset". In: *The European Conference on Computer Vision (ECCV) Workshops* (cit. on p. 32).

Krizhevsky, Alex (2009). "Learning Multiple Layers of Features from Tiny Images". In: p. 60 (cit. on pp. 14, 18, 24).

Laloy, Eric et al. (2018). "Training-Image Based Geostatistical Inversion Using a Spatial Generative Adversarial Neural Network". In: *Water Resources Research* 54.1, pp. 381–406 (cit. on p. 18).

LeCun, Yann et al. (1998). "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 00189219. DOI: 10.1109/5.726791. URL: http://ieeexplore.ieee.org/document/726791/ (cit. on pp. 14, 24).

Lemmens, L. et al. (2017). *Effective Structural Descriptors for Natural and Engineered Radioactive Waste Confinement Barrier*. Vienna (cit. on p. 26).

Li, Chun-Liang et al. (Nov. 27, 2017). "MMD GAN: Towards Deeper Understanding of Moment Matching Network". In: arXiv: 1705.08584 [cs, stat]. URL: http://arxiv.org/abs/1705.08584 (visited on 05/19/2020) (cit. on pp. 9, 14).

Liese, F. and I. Vajda (Oct. 2006). "On Divergences and Informations in Statistics and Information Theory". In: *IEEE Transactions on Information Theory* 52.10, pp. 4394–4412. ISSN: 0018-9448. DOI: 10.1109/TIT.2006.881731. URL: http://ieeexplore.ieee.org/document/1705001/ (visited on 05/22/2020) (cit. on p. 12).

Lin, Tsung-Yi et al. (2014). "Microsoft Coco: Common Objects in Context". In: *European Conference on Computer Vision*. Springer, pp. 740–755 (cit. on p. 39).

Lin, Tsung-Yi et al. (2017). "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988 (cit. on p. 39).

Lin, Zinan et al. (2018). *PacGAN: The Power of Two Samples in Generative Adversarial Networks*, pp. 1505–1514. URL: https://arxiv.org/pdf/1712.04086.pdf%20http://papers.nips.cc/paper/7423-pacgan-the-power-of-two-samples-in-generative-adversarial-networks (cit. on pp. 17–19, 23).

Liu, Ziwei et al. (2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)* (cit. on pp. 18, 24).

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: p. 6 (cit. on p. 14).

Mao, Xudong et al. (Apr. 5, 2017). "Least Squares Generative Adversarial Networks". In: arXiv: 1611.04076 [cs]. URL: http://arxiv.org/abs/1611.04076 (visited on 05/21/2020) (cit. on pp. 12, 14, 35).

Marafioti, Andrés et al. (2018). "A Context Encoder for Audio Inpainting". In: *arXiv preprint arXiv:1810.12138* (cit. on p. 30).

Mehri, Armin and Angel D Sappa (2019). "Colorizing near Infrared Images through a Cyclic Adversarial Approach of Unpaired Samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (cit. on p. 32).

Mescheder, Lars, Andreas Geiger, and Sebastian Nowozin (2018). "Which Training Methods for GANs Do Actually Converge?" In: URL: http://proceedings.mlr.press/v80/mescheder18a/mescheder18a.pdf (cit. on p. 8).

Mirza, Mehdi and Simon Osindero (2014). "Conditional Generative Adversarial Nets". In: URL: https://arxiv.org/pdf/1411.1784.pdf (cit. on pp. 10, 17, 18, 27).

Morel, Olivier et al. (2006). "Active Lighting Applied to Three-Dimensional Reconstruction of Specular Metallic Surfaces by Polarization Imaging". In: *Applied optics* 45.17, pp. 4062–4068 (cit. on p. 32).

Mroueh, Youssef and Tom Sercu (Nov. 3, 2017). "Fisher GAN". In: arXiv: 1705.09675 [cs, stat]. URL: http://arxiv.org/abs/1705.09675 (visited on 05/21/2020) (cit. on pp. 13, 14).

Müller, Alfred (June 1997). "Integral Probability Metrics and Their Generating Classes of Functions". In: *Advances in Applied Probability* 29.2, pp. 429–443. ISSN: 0001-8678, 1475-6064. DOI:

10.2307/1428011. URL: https://www.cambridge.org/core/product/identifier/S000186780002807X/type/journal_article (visited on 05/22/2020) (cit. on p. 13).

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: p. 8 (cit. on p. 14).

Nie, Dong et al. (2017). "Medical Image Synthesis with Context-Aware Generative Adversarial Networks". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 417–425 (cit. on p. 32).

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). *F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization*. URL: https://arxiv.org/pdf/1606.00709.pdf (cit. on pp. 9, 12, 14).

Odena, Augustus, Christopher Olah, and Jonathon Shlens (2016). "Conditional Image Synthesis with Auxiliary Classifier GANs". In: URL: https://arxiv.org/pdf/1610.09585.pdf (cit. on p. 10).

Pajot, Arthur, Emmanuel de Bezenac, and Patrick Gallinari (2019). "Unsupervised Adversarial Image Reconstruction". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=BJg4Z3RqF7 (cit. on pp. 18, 19, 21).

Parzen, Emanuel (1962). "On Estimation of a Probability Density Function and Mode". In: *Annals of Mathematical Statistics* 33.3, pp. 1065–1076. ISSN: 0003-4851. DOI: 10.1214/AOMS/1177704472 (cit. on p. 16).

Pathak, Deepak et al. (2016). "Context Encoders: Feature Learning by Inpainting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544 (cit. on p. 18).

Peyré, Gabriel and Marco Cuturi (Mar. 18, 2020). "Computational Optimal Transport". In: arXiv: 1803.00567 [stat]. URL: http://arxiv.org/abs/1803.00567 (visited on 05/23/2020) (cit. on p. 13).

Pietikäinen, Matti et al. (2011). *Computer Vision Using Local Binary Patterns*. Vol. 40. Computational Imaging and Vision. London: Springer London. ISBN: 978-0-85729-747-1. DOI: 10.1007/978-0-85729-748-8. URL: http://link.springer.com/10.1007/978-0-85729-748-8 (cit. on p. 26).

Radford, Alec, Luke Metz, and Soumith Chintala (Nov. 2015). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: arXiv: 1511.06434. URL: http://arxiv.org/abs/1511.06434 (cit. on pp. 9, 14, 25).

Rehbinder, Jean et al. (2016). "Ex Vivo Mueller Polarimetric Imaging of the Uterine Cervix: A First Statistical Evaluation". In: *Journal of biomedical optics* 21.7, p. 071113 (cit. on p. 32).

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241 (cit. on p. 25).

Ruffino, Cyprien et al. (May 2019). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: URL: http://arxiv.org/abs/1905.08613 (cit. on pp. 25, 26, 29).

Salimans, Tim et al. (June 2016). "Improved Techniques for Training GANs". In: URL: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf%20http://arxiv.org/abs/1606.03498 (cit. on pp. 9, 14, 16, 25).

Sallab, Ahmad El et al. (2019). "LiDAR Sensor Modeling and Data Augmentation with GANs for Autonomous Driving". In: *arXiv preprint arXiv:1905.07290* (cit. on p. 32).

Sønderby, Casper Kaae et al. (Feb. 21, 2017). "Amortised MAP Inference for Image Super-Resolution". In: arXiv: 1610.04490 [cs, stat]. URL: http://arxiv.org/abs/1610.04490 (visited on 05/19/2020) (cit. on pp. 9, 14).

Springenberg, Jost Tobias et al. (Apr. 13, 2015). "Striving for Simplicity: The All Convolutional Net". In: arXiv: 1412.6806 [cs]. URL: http://arxiv.org/abs/1412.6806 (visited on 05/22/2020) (cit. on p. 14).

Sriperumbudur, Bharath K. et al. (Oct. 12, 2009). "On Integral Probability Metrics, \phi-Divergences and Binary Classification". In: arXiv: 0901.2698 [cs, math]. URL: http://arxiv.org/abs/0901.2698 (visited on 05/22/2020) (cit. on p. 13).

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: https://www.cs.toronto.edu/%20hinton/absps/JMLRdropout.pdf (cit. on p. 14).

Strebelle, Sebastien (Jan. 1, 2002). "Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics". In: *Mathematical Geology* 34.1, pp. 1–21. ISSN: 1573-8868. DOI: 10.1023/A:1014009426274. URL: https://doi.org/10.1023/A:1014009426274 (cit. on pp. 18, 24).

Szegedy, Christian et al. (Dec. 2016). "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. IEEE Computer Society, pp. 2818–2826. ISBN: 978-1-4673-8850-4. DOI: 10.1109/CVPR.2016.308. URL: http://arxiv.org/abs/1512.00567 (cit. on pp. 16, 25, 39).

Szekely, Gabor J and Maria L Rizzo (2004). "Testing for Equal Distributions in High Dimension". In: p. 15 (cit. on p. 13).

Theis, Lucas, Aäron Van Den Oord, and Matthias Bethge (2015). "A Note on the Evaluation of Generative Models". In: URL: https://arxiv.org/pdf/1511.01844.pdf (cit. on p. 25).

Vaccari, Cristian and Andrew Chadwick (2020). "Deepfakes and Disinformation: Exploring the Impact of Synthetic Political Video on Deception, Uncertainty, and Trust in News". In: *Social Media and Society* 6.1. ISSN: 20563051. DOI: 10.1177/2056305120903408 (cit. on p. 1).

Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba (Oct. 26, 2016). "Generating Videos with Scene Dynamics". In: arXiv: 1609.02612 [cs]. URL: http://arxiv.org/abs/1609.02612 (visited on 05/19/2020) (cit. on p. 1).

Wang, Ting-Chun et al. (Dec. 3, 2018a). "Video-to-Video Synthesis". In: arXiv: 1808.06601 [cs]. URL: http://arxiv.org/abs/1808.06601 (visited on 05/21/2020) (cit. on p. 10).

Wang, Yi et al. (2018b). *Image Inpainting via Generative Multi-Column Convolutional Neural Networks*. 329–338. URL: http://papers.nips.cc/paper/7316-image-inpainting-via-generative-multi-column-convolutional-neural-networks (cit. on p. 18).

Wang, Zhihao, Jian Chen, and Steven C.H. Hoi (2020). "Deep Learning for Image Super-Resolution: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2020.2982166 (cit. on p. 1).

Wang, Zhixiang, Yinqiang Zheng, and Yung-Yu Chuang (June 2019). "Polarimetric Camera Calibration Using an LCD Monitor". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 32, 34).

Wolff, Lawrence B and Andreas G Andreou (1995). "Polarization Camera Sensors". In: *Image and Vision Computing* 13.6, pp. 497–510 (cit. on p. 32).

Wu, Jiajun et al. (Jan. 4, 2017). "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". In: arXiv: 1610.07584 [cs]. URL: http://arxiv.org/abs/1610.07584 (visited on 05/19/2020) (cit. on p. 1).

Wu, Yan, Mihaela Rosca, and Timothy Lillicrap (2019). "Deep Compressed Sensing". In: *Proceedings of the 36th International Conference on Machine Learning* (cit. on p. 20).

Xiao, Han, Kashif Rasul, and Roland Vollgraf (Aug. 2017). "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: URL: http://arxiv.org/abs/1708.07747 (cit. on pp. 18, 24–26).

Xu, Huazhe et al. (2017). "End-to-End Learning of Driving Models from Large-Scale Video Datasets". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2174–2182 (cit. on p. 32).

Yang, Dingdong et al. (2019). "Diversity-Sensitive Conditional Generative Adversarial Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJliMh09F7 (cit. on pp. 18, 23).

Yeh, Raymond A et al. (2017). "Semantic Image Inpainting with Deep Generative Models". In: URL: https://arxiv.org/pdf/1607.07539.pdf (cit. on p. 19).

Yu, Fisher and Vladlen Koltun (2015). "Multi-Scale Context Aggregation by Dilated Convolutions". In: URL: https://arxiv.org/pdf/1511.07122.pdf (cit. on p. 25).

Yu, Jiahui et al. (Jan. 2018). "Generative Image Inpainting with Contextual Attention". In: URL: http://arxiv.org/abs/1801.07892 (cit. on p. 19).

Zhang, Lichao et al. (2018). "Synthetic Data Generation for End-to-End Thermal Infrared Tracking". In: *IEEE Transactions on Image Processing* 28.4, pp. 1837–1850 (cit. on p. 32).

Zhao, Junbo, Michael Mathieu, and Yann LeCun (Mar. 6, 2017). "Energy-Based Generative Adversarial Network". In: arXiv: 1609.03126 [cs, stat]. URL: http://arxiv.org/abs/1609.03126 (visited on 05/21/2020) (cit. on pp. 12, 14).

Zhu, Dizhong and William A. P. Smith (June 2019). "Depth from a Polarisation + RGB Stereo Pair". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 32).

Zhu, Jun-Yan et al. (2017a). "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks Monet Photos". In: URL: https://arxiv.org/pdf/1703.10593.pdf (cit. on pp. 32, 35).

Zhu, Xinyue et al. (2017b). *Emotion Classification with Data Augmentation Using Generative Adversarial Networks*. URL: https://arxiv.org/pdf/1711.00648.pdf (cit. on pp. 11, 25, 27, 32).

Appendix A

Publications

Appendix B

Experiment details for the Pixel-Wise Conditionned GAN

Appendix C

Experiment details for the Polarimetric CycleGAN