Normandie Université

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 Wednesday 4th November, 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

INSA ROUEN NORMANDIE

MI∫IΣ

litis

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
DOP	Degree Of Polarization
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)
LiDAR	Light Detection And Ranging
LSGAN	Least-Squares Generative Adversarial Networks
MSE	Mean-Squared Error
RGB	Red-Green-Blue (color model)
ReLU	Rectified Linear Unit
RIP	Restricted Isometry Property
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 et al., 2017), image super-resolution (Wang et al., 2020), map style tranfer (Kang et al., 2019), video prediction (Vondrick et al., 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

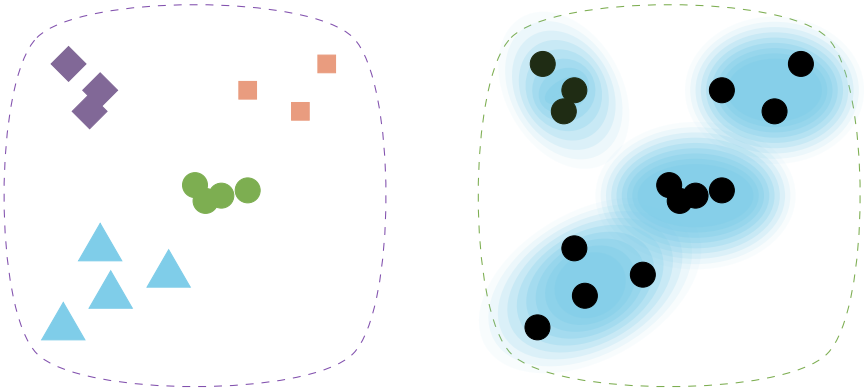


Figure 1.1: Left: Discriminative modeling, the model aims to assign a class to each data point. Right: Generative modeling, the model aims to learn the underlying probability distribution of the data points.

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 et al., 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 $p_{\theta_{\mathbf{y}|\mathbf{x}}}$ to a conditional probability distribution $p_{\mathbf{y}|\mathbf{x}}$ of data $\mathbf{x} \in \mathscr{X}$ and labels $\mathbf{y} \in \mathscr{Y}$ relatively to the data $\mathbf{x} \sim p_x$, a generative model aims to fit $p_{\theta_{\mathbf{x}}}$ to $p_{\mathbf{x}}$ 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 estima-

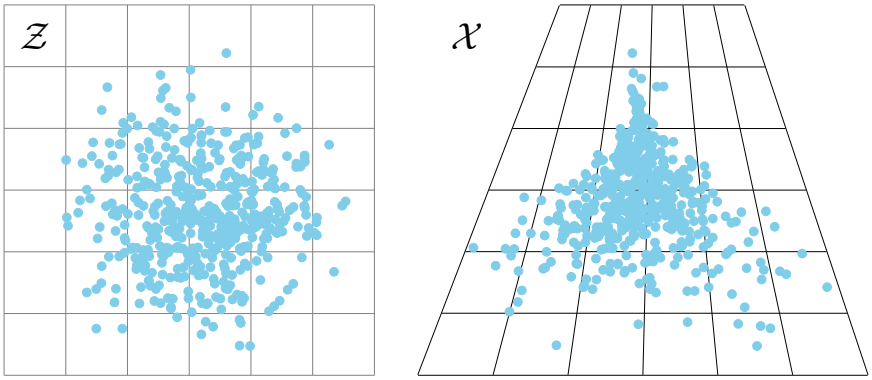


Figure 1.2: CR: A refaire :/ nA mapping between a latent space $\mathcal{Z}$ and the space of a dataset $\mathcal{X}$.

tion

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

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

$$p_{\theta_{\mathbf{x}}} = \sum_z p_z p_{\theta_{\mathbf{x}|z}} \ ,$$

with log-likelihood

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

In the case of the GMMs, the Expectation-Maximization (EM) algorithm (Dempster et al., 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 $p_{\mathbf{z}}$, then drawing a sample from the Gaussian distribution $p_{\mathbf{x}|\mathbf{z}=k} = \mathcal{N}(\mu_k^*, \sigma_k^{*2})$.

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 $\mathbf{x} \sim \mathcal{N}(\mu_k^*, \sigma_k^{*2})$, we can instead sample a latent variable $\mathbf{z} \sim \mathcal{N}(0,1)$ and compute $\mathbf{x} = G(\mathbf{z}; \mu, \sigma) = \mu + \mathbf{z}\sigma$. 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 $p_{\mathbf{z}}$ 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 $G(\mathbf{z};\theta)$. More formally, instead of directly modeling $p_{\mathbf{x}}$, a latent variable model learns a deterministic mapping $p_{G_{\mathbf{x}|\mathbf{z}}}$. From this mapping, the generative model can be obtain through marginalization

$$p_{G_{\mathbf{x}}} = \int_{\mathcal{Z}} p_{\mathbf{z}} p_{G_{\mathbf{x}|\mathbf{z}}} d\mathbf{z} = \int_{\mathcal{Z}} p_{\mathbf{z}} p_{\mathbf{x}|G(\mathbf{z};\theta)} d\mathbf{z} \ . \tag{1.3}$$

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

While the marginal distribution $p_{G_{\mathbf{x}}}$ 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 $p_{G_{\mathbf{x}|\mathbf{z}}}$ as well as the distribution $q_F(\mathbf{z}|\mathbf{x})$. 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 $p_{\mathbf{x}}$ as

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

The KL terms evaluates the distance between the distribution $q_F(\mathbf{z}|\mathbf{x})$ learned by the encoder and real distribution $p_{\mathbf{z}|\mathbf{x}}$, and since $p_{\mathbf{z}}$ 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

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

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

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

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*

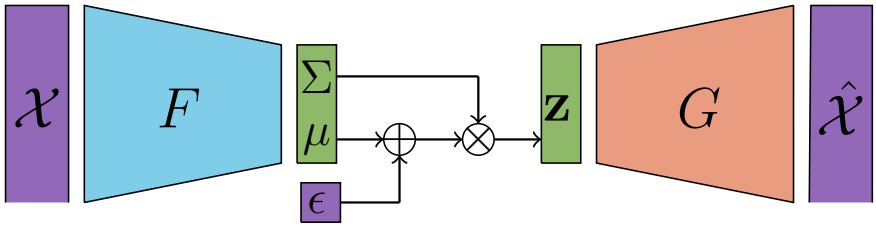


Figure 1.3: Variational auto-encoder framework

$$p_{\mathrm{G}_\mathbf{x}} = p_\mathbf{z} \left| \det \left(\frac{\partial \mathrm{G}(\mathbf{z})}{\partial \mathbf{z}^\mathrm{T}} \right) \right|^{-1} = p_{\mathrm{G}^{-1}(\mathbf{x})} \left| \det \left(\frac{\partial \mathrm{G}^{-1}(\mathbf{x})}{\partial \mathbf{x}^\mathrm{T}} \right) \right| ,$$

with $\mathbf{z} \sim p_\mathbf{z}$ 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.

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 et al., 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 $p_{\mathrm{G}_{\mathbf{x}|\mathbf{z}}}$ between a simple distribution $p_\mathbf{z}$ (usually normal or uniform) to the real data distribution $p_\mathbf{x}$. 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 $p_\mathbf{x}$ and the mapped distribution $p_{\mathrm{G}_\mathbf{x}}$. Therefore, GANs are often qualified as *likelihood-free* generative models.

Since a divergence $\mathrm{Div}(p_\mathbf{x} || q(\mathbf{x}))$ between two distributions $p_\mathbf{x}$ and $q(\mathbf{x})$ is analogous to a distance between these distributions (see Figure 1.4), minimizing such a divergence allows for a parametric distribution $p_{\theta_\mathbf{x}}$ to fit a target distribution $p_\mathbf{x}$. 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.

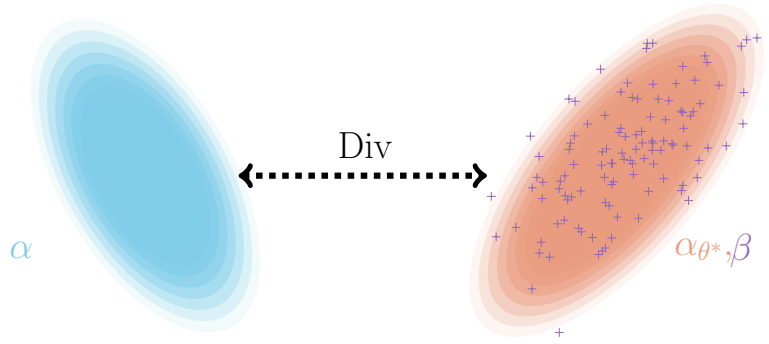


Figure 1.4: A divergence Div(α||β) can capture the distance between a parametric model α_θ and the observations β. Density fitting can then be formulated as $\alpha_{\theta^*} = \arg\min_\theta \mathrm{Div}(\alpha_\theta||\beta)$, where α_{θ^*} is the best fit model.

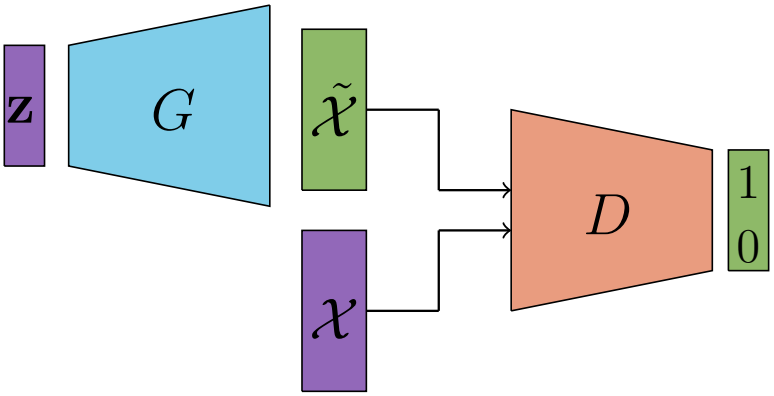


Figure 1.5: Generative Adversarial Networks framework

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 $\mathbf{x}$ was sampled on the real distribution $p_\mathbf{x}$ or was generated from $z \sim p_\mathbf{z}$ and is trained with binary cross-entropy

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

The intuition behind this model is that once the discriminator is trained, maximizing its error on generated samples $\mathbf{x} \sim p_{\mathrm{G}_\mathbf{x}}$ w.r.t the parameters of G should push p_{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 $\mathrm{L_D(D,G)}$ for a fixed G is given by

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

By plugging this optimal into the discriminator cost, we get

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

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 $\mathrm{L}_{\mathrm{G}}(\mathrm{G})$ as $\mathrm{L}_{\mathrm{G}}(\mathrm{G}) = \min_{\mathrm{D}} \mathrm{L}_{\mathrm{D}}(\mathrm{D},\mathrm{G})$. Up to additive and multiplicative constants, the criterion $\mathrm{L}_{\mathrm{G}}(\mathrm{G})$ can be reformulated as

$$\mathrm{L}_{\mathrm{G}}(\mathrm{G}) = \mathrm{D}_{\mathrm{KL}}\left(p_{\mathbf{x}}\middle\|\frac{p_{\mathbf{x}}+p_{\mathrm{G}_{\mathbf{x}}}}{2}\right) + \mathrm{D}_{\mathrm{KL}}\left(p_{\mathrm{G}_{\mathbf{x}}}\middle\|\frac{p_{\mathbf{x}}+p_{\mathrm{G}_{\mathbf{x}}}}{2}\right) = 2\cdot \mathrm{D}_{\mathrm{JS}}\left(p_{\mathbf{x}}\middle\|p_{\mathrm{G}_{\mathbf{x}}}\right) .$$

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

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

As shown above, this mini-max game has, assuming infinite capacity for both G and D, a global optimum for $p_{\mathbf{x}} = p_{\mathrm{G}_{\mathbf{x}}}$. 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.

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 $p_{\mathbf{z}}$
    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 $p_{\mathbf{z}}$;
    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

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.

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

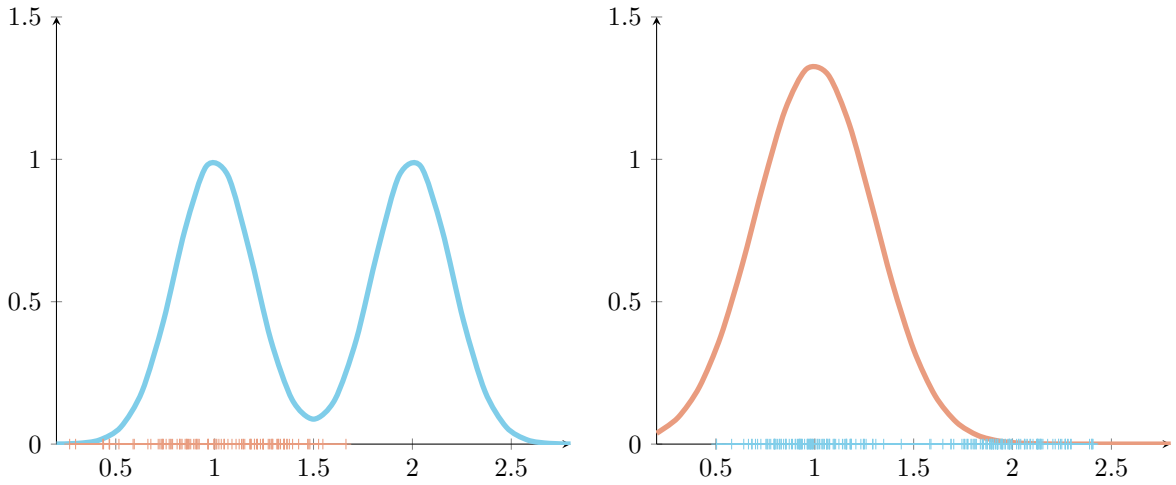


Figure 1.6: Reverse KL (left) and KL (right) divergence between the true blue distribution and the mode-collapsed orange distribution . The distance is lower in the case of the reverse KL, even if a missing mode is clearly visible.

training process. This can result in oscillating values of the loss function which prevents convergence (Mescheder et al., 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 et al., 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,

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

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 et al., 2018). Even though several theory-backed techniques aimed to solve this issue (Arjovsky et al., 2017; Nowozin et al., 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.

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 $\mathbf{z}_1, \mathbf{z}_2$ are mapped to samples $G(\mathbf{z}_1)$ and $G(\mathbf{z}_2)$ 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 $p_{G_\mathbf{x}}$, which are some points in the support of $p_\mathbf{x}$ that have zero (or near-zero) probability on $p_{G_\mathbf{x}}$ (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 $\mathbf{z}$ to the single most probable $\mathbf{x}$, 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 et al., 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 et al., 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 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

CR: TODO

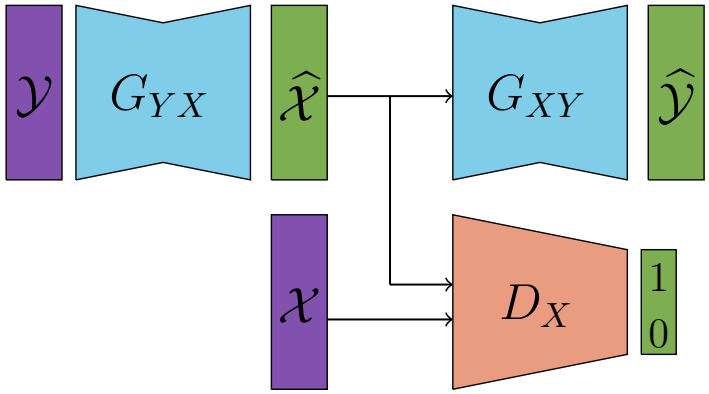


Figure 1.7: The CycleGAN approach. Half of the training setup is illustrated, the other half consisting in the same setup but with inverted X and Y

Conditional modeling

While classical generative models such as GANs try to unconditionally approximate the real-data distribution $p_{\mathbf{x}}$, a conditional generative model aim to learn a model of the conditional distribution $p_{\mathbf{x}|\mathbf{y}}$, 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

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

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 et al., 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 $p_{\mathbf{x}|\mathbf{y}}$ between two high-dimensional distributions $p_{\mathbf{x}}$ and $p_{\mathbf{y}}$ 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 $p_{\mathbf{x}|\mathbf{y}}$, 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 $(x,y) \sim p_{\mathbf{x}|\mathbf{y}}$ forcing the generator to minimize the ℓ_1 reconstruction term between x and $\mathrm{G}(y,z)$

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

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 $\mathrm{G}(\mathbf{y},\mathbf{z})$, the CycleGAN approach train two domain-transfer models simultaneously: $\mathrm{G_{YX}}$ and $\mathrm{G_{XY}}$ that map samples from $p_{\mathbf{y}}$ onto $p_{\mathbf{x}}$ and $p_{\mathbf{x}}$ onto $p_{\mathbf{y}}$, respectively (see Figure 1.7). This allows to compute the ℓ_1 reconstruction errors $||x-\mathrm{G_{YX}}(\mathrm{G_{XY}}(\mathbf{x}))||_1$ and $||y-\mathrm{G_{XY}}(\mathrm{G_{YX}}(\mathbf{y}))||_1$, 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)

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

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 2.

Algorithm 2 CycleGAN training algorithm

Require: $\mathscr{X}$ and $\mathscr{Y}$ two unpaired datasets, $\mathrm{G_{XY}}$ and $\mathrm{G_{YX}}$ 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 $\mathscr{X}$

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

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

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

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

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

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

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

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

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

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

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

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

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

until a stopping condition is met

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

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

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

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

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 et al. (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 et al., 2017) to be the divergence used in the Energy-Based GAN (Zhao et al., 2017) approach.

While the f-divergences have been the seminal approach to GANs, they can exhibit strong issues. Arjovsky et al. (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 et al. (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 & Cuturi, 2020). The Wasserstein distance, albeit having many different formulations, can be expressed through the Kantorovich-Rubinstein duality (Kantorovich & Akilov, 1982) as

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

where $\mathbb{F} = \{f : ||f||_L \leq 1\}$ 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

$$L_{WGAN}(D, G) = \min_G \max_D \left[\mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\mathbf{x}}} D(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{z} \sim q(\mathbf{z})} D(G(\mathbf{z})) \right] .$$

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

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

where $p_{\hat{\mathbf{x}}}$ is implicitly defined as an uniform distribution on straight lines between pairs of points sampled on $p_{\mathbf{x}}$ and $p_{G_{\mathbf{x}}}$. 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

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

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 et al., 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

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 et al., 2017)	Total variation
f-GAN (Nowozin et al., 2016)	Various f-divergences
Integral Probability Metrics (IPMs)	
EBGAN* (Zhao et al., 2017)	Total variation
WGAN (Arjovsky et al., 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.

Augmenting the objective

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

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 et al., 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 deconvolutional (or transposed convolutional) layers in the generator. It also introduced dropout (Srivastava et al., 2014) and Batch Normalization (Ioffe et al., 2015), and used both ReLU (Nair & Hinton, 2010) and Leaky ReLU (Maas et al., 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.

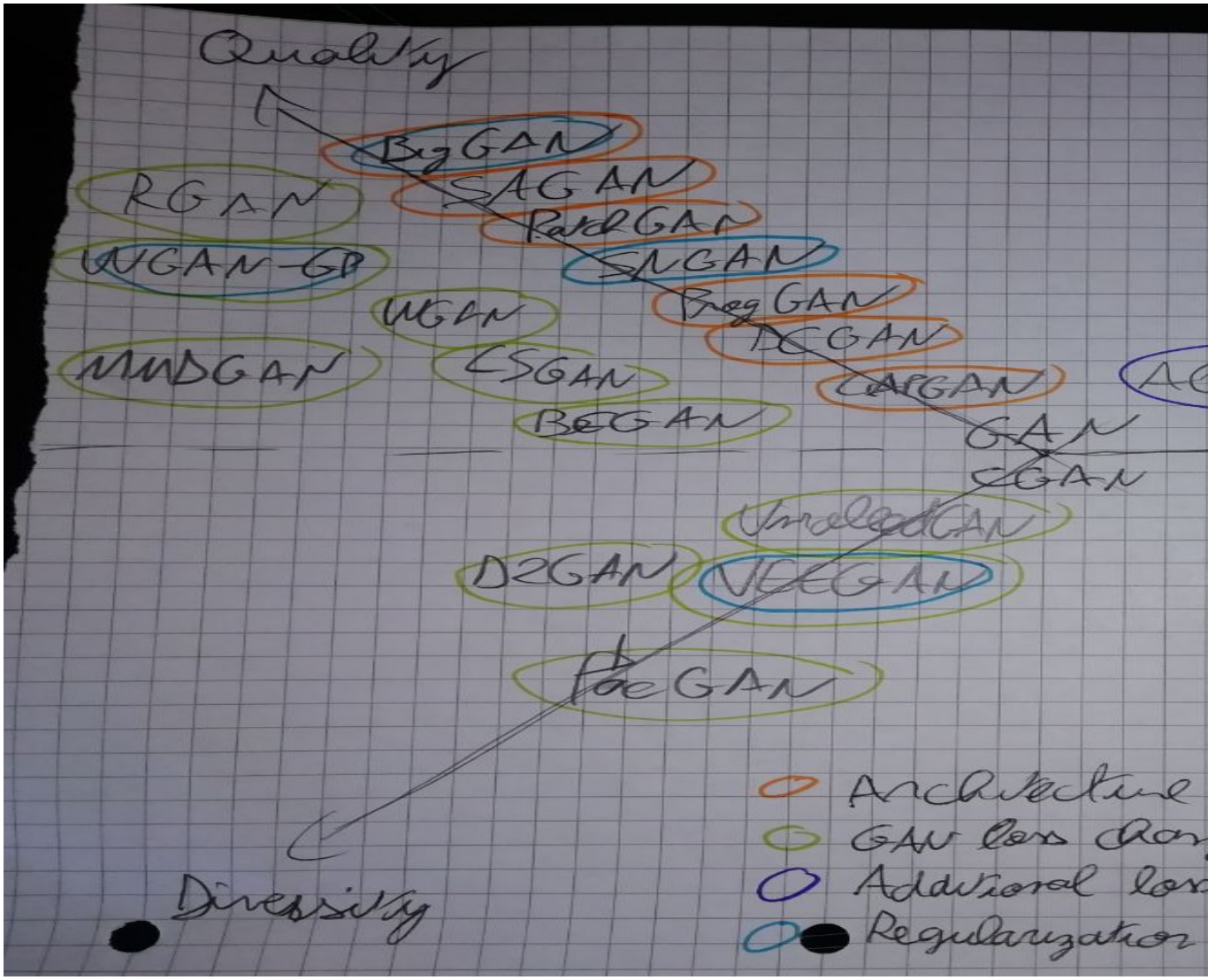


Figure 1.8: Classifications of some advances in GANs on the trilemma

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 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 Incep-

tion 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 $p_{\mathrm{G}_{\mathbf{y}|\mathbf{x}}}$ with low entropy. In addition to the high quality, the samples should be diverse, therefore the marginal distribution $p_{\mathrm{G}_{\mathbf{y}}} = \int_{\mathcal{Z}} p_{\mathrm{G}_{\mathbf{y}|\mathbf{x}=\mathrm{G}(\mathbf{z})}} d\mathbf{z}$ should have a high entropy. By combining these two requirements, the IS is formulated as

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

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 $p_{\mathrm{G}}(\mathbf{y})$.

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

Image reconstruction as an auxiliary task to generative modeling

Chapter abstract

While the Conditional GAN approach (Mirza & Osindero, 2014) is generic enough to model any kind of conditioning, it lacks some form of control or guarantee on the conditioning procedure. 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 small subset of randomly-located pixels known beforehand. This kind of problem is directly motivated by applications in geosciences, most notably the generation of subsurface rock structure (Laloy et al., 2019; Ruffino et al., 2017). 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 scarce 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 procedure. 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 to a common dataset from geosciences of subsurface rock formations.

The work in this chapter has led to the publication of the following papers:

- Cyprien Ruffino, Romain Hérault, Eric Laloy and Gilles Gasso (Nov. 2019). Pixel-Wise Conditioning of Generative Adversarial Networks. In: Proceedings of the 27th European Symposium on Artificial Neural Networks (ESANN).
- Cyprien Ruffino, Romain Hérault, Eric Laloy and Gilles Gasso (Apr. 2020). Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion. In: Neurocomputing.

Contents

2.1 Introduction

Conditional GANs (Mirza & Osindero, 2014) are a powerful method for learning conditional generative models. By simply providing a label to both the generator and discriminator networks, CGAN is able to solve problems such as class-conditioned image generation (Mirza & Osindero, 2014), image-to-image translation (Isola et al., 2016; Wang et al., 2018b), image super-resolution (Wang et al., 2020) or image inpainting (Pathak et al., 2016). Although this approach combined with enough data and the appropriate neural network architectures has led to impressive results (Karras et al., 2020), it lacks some mechanism to strongly enforce conditioning. Indeed, it only relies on the adversarial learning procedure with no explicit method for including the constraints into the generation task.

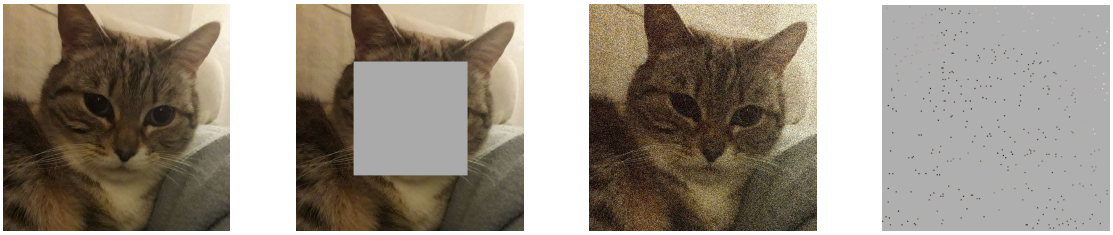
In this chapter, we propose to address the problem of reconstructing images from very few pixels (usually less than a percent). We refer to these conditioning pixels as a constraint map **y**. This kind of task has several applications, in which recovering the entirety of a signal with very sparse measurements is necessary, for example in domains where measuring the signal is expensive. Here, we study the task of generating a subsurface rock formation from very few measurements, which has direct applications in geology, and following previous works on subsurface data generation (Laloy et al., 2018; Laloy et al., 2019).

To reconstruct the missing information, a generative model must be able to generate high quality images coherent with the given pixel values by leveraging on a training set of similar images. The model aims to match the distribution of the real images conditioned on a highly scarce constraint map. To explicitly force the generated images towards honoring the prescribed pixel values, we use a reconstruction loss measuring how close real constrained pixels are to their generated counterparts. By re-framing this problem as a Maximum A Posteriori estimation, 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 comprising a reconstruction loss and the classical adversarial loss of GAN. Both losses are balanced through a regularization parameter.

We analyze the influence of this hyperparameter in terms of quality of generated images and the respect of the constraints. Specifically, empirical evaluation on MNIST (LeCun et al., 1998) and FashionMNIST (Xiao et al., 2017) evidences that the regularization parameter allows for controlling the trade-off between the visual quality of the generated images 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). Our empirical findings reveal that the used architectures may lack stochasticity in the generated samples, that is the conditional 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 PacGAN (Lin et al., 2018) strategy, which consists in providing pairs of images as input to the discriminator during the training process instead of single images, and such for both the generated images and the images from the dataset (see 1.2.2).

Endowed with the PacGAN learning procedure, our resulting GAN 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 matches up or outperforms CGAN on the visual quality of the generated samples.

The remainder of the chapter is organized as follows. In Section 2.3, we review the relevant related work focusing first on two main groups of methods for dealing with image reconstruction from highly altered training samples, namely compressed sensing (Candes & Tao, 2005) approaches and conditional generation methods. Section 2.4 introduces our approach for image reconstruction, and proposes some theoretical insight. In Section



(a) Original image (b) Inpainting input (c) Denoising input (d) Reconstruction input

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

2.4.2, we present the experimental protocol and evaluation measures along with quantitative and qualitative effectiveness of our approach. The last section concludes the chapter.

To sum up, the contributions are summarized as follows:

- We propose a method for learning to generate images with a few pixel-wise constraints, which deals with the trade-off between the image quality and the fulfillment of the constraints.
- 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 The problem of image reconstruction

Image reconstruction is the task of retrieving an image from a very altered source, which can take several forms from additive noise to missing parts of the image. In this chapter, we study a rather extreme case of alteration, which is the removal of over 99% of the original image, leaving only a handful of pixels scattered at random positions.

Image reconstruction belongs to the family of problems consisting in retrieving an image from an altered one. This includes problems such as inpainting (Bertalmio et al., 2000) (see Figure 2.1b) or image denoising (Goyal et al., 2020) (see Figure 2.1) which consists in retrieving missing or altered parts of an image Figure 2.1.

Image inpainting (Figure 2.1b) is the task of recreating missing or damaged regions of an image. This kind of alterations have numerous applications, from the restoration of damaged pictures (Oliveira et al., 2001) to semantic image editing Bau et al., 2019 including for example object removal (Criminisi et al., 2004).

In the same fashion, (Figure 2.1c) image denoising aims to remove alterations induced by some noise, which can be the result of imperfections in the acquisition procedure or natural degradation. The former cause finds applications such as raw image denoising in cameras (Kim, 2014) or medical image denoising (Gondara, 2016).

Image reconstruction however differs from these problems as most of the original image is unavailable. Thus, in comparison to inpainting or denoising in which the altered parts the input can be retrieved from a semantically rich altered image, image reconstruction instead requires to generate a full image from very few and unstructured ob-

servations. This can be done by leveraging on a prior such as generative modeling, while ensuring that the resulting image is coherent with the values given as input.

Before delving into the details, let introduce the notations related to the problem. We denote by $\mathrm{X} \in \mathbb{X}$ a random variable and $\mathbf{x}$ its realization. Let p_{X} be the distribution measure of X over $\mathbb{X}$. Similarly $p_{\mathrm{X}|\mathrm{Y}}$ represents the distribution of X conditioned on the random variable $\mathrm{Y} \in \mathbb{Y}$, while $p_{\mathrm{X},\mathrm{Y}}$ represents the joint distribution.

Whether it is for image inpainting, denoising or reconstruction, we aim to recover a signal from which we only have altered measurements. This problem can be formulated as

$$\mathbf{y} = \mathrm{A}\mathbf{x} + \epsilon \ , \tag{2.1}$$

where $\mathrm{A} \in \mathbb{R}^{a \times b}$ is a wide ($a \ll b$), matrix (so called "measurement matrix") and ϵ is the noise. By varying the nature of the matrix A, we can formulate the three aforementioned problem.

In the case of image reconstruction, assume $\mathbf{y}$ is the given set of constrained pixel values. To ease the presentation, let consider $\mathbf{y}$ 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 $\mathrm{M}_{\mathbf{y}} \in \{0,1\}^{n \times p \times c}$. We also consider that the measurements $\mathbf{y}$ are free of noise, thus we have $\epsilon = 0$.

Having access to a set of ground-truth images $\mathscr{X} = \{\mathbf{x}_1, ...\mathbf{x}_s\}, \mathbf{x}_i \in \mathbb{R}^{n \times p \times c}$ (see Figure 2.1a) drawn from an unknown distribution p_{X} and a set of sparse matrices $\mathscr{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_t\}, \mathbf{y}_i \in \mathbb{R}^{n \times p \times c}$ (Figure 2.1d) as the given constrained pixels, the image reconstruction problem consists in finding an approximated image $\hat{\mathbf{x}}$ that maximizes $p_{\mathrm{X}}(\hat{\mathbf{x}})$ for a given constraint map $\mathbf{y}$.

In other words, the problem consists in retrieving $\mathbf{x}$ such that $\mathbf{y} = \mathrm{M}_{\mathbf{y}} \odot \mathbf{x}$ and $\mathbf{x}$ is close to the data distribution p_{X}. More formally, we can formulate it as finding

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p_{\mathrm{X}}(\mathbf{x}) \ \text{ subject to } \ \mathbf{y} = \mathrm{M}_{\mathbf{y}} \odot \mathbf{x} \tag{2.2}$$

where $\odot$ stands for the Hadamard (or point-wise) product[1] and $\mathrm{M}_{\mathbf{y}}$ for the mask, the sparse matrix with entries equal to one at constrained pixels location.

2.3 Approaches for image reconstruction

We propose here an overview of some of the seminal approaches for solving similar tasks. We present two main types of approaches: compressed sensing-based approaches and conditional modeling. We detail some strengths and weakness of these approaches, summarized in Table 2.1.

2.3.1 Sparsity-based approaches for image reconstruction

A first approach to tackle the image reconstruction problem is to recover the image through per-sample optimization. Although the original problem (Equation (2.1)) is linear, it is

[1] Note that this expression can be formulated with the Hadamard product instead of a matrix product in 2.1, since $\mathrm{vect}(\mathrm{M}_{\mathbf{y}} \odot \mathbf{x}) = \mathrm{Tr}(\mathrm{Diag}(\mathrm{vect}(\mathrm{M}_{\mathbf{y}}))\mathrm{vect}(\mathbf{x}))$.

highly under determined, thus it induces an infinite number of solutions as the problem is ill-posed. However, by including prior knowledge on the signal $\mathbf{x}$ and by ensuring some constraints on the matrix A, solving this system can be done using techniques such as linear programming.

Image reconstruction using compressed sensing

Candes and Tao (2005) introduced **Compressed Sensing**, which consists in solving the system by assuming that the signal $\mathbf{x}$ to be recovered is sparse. In order to guarantee that the obtained image is indeed a reconstruction, they introduced the Restricted Isometry Property (RIP) (Candès, 2008) on the family of matrices A, which states that for two samples $\mathbf{x}_1, \mathbf{x}_2 \sim p_X$,

$$(1-\alpha)\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \leq \|A(\mathbf{x}_1 - \mathbf{x}_2)\|_2^2 \leq (1+\alpha)\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \ , \tag{2.3}$$

where α is a small constant. This states that distance between two samples is preserved when altered by A. Candes and Tao (2005) used this property to show that if the matrix A enforces the RIP, samples $\hat{\mathbf{x}}$ retrieved by compressed sensing will have a high probability of having a high likelihood on the real data distribution p_X. Examples of matrices that enforces the RIP are random Gaussian or Fourier matrices. Under the RIP setting, the sparse signal $\mathbf{x}$ can be retrieved by solving for

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 0 \ \text{ subject to } \ \|A\mathbf{x} - \mathbf{y}\|_2^2 \leq \delta \ , \tag{2.4}$$

where δ is a small constant or, when the measurement process is assumed to be noiseless,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ subject to } \ A\mathbf{x} = \mathbf{y} \ . \tag{2.5}$$

Solving this problem is NP-hard, however in such a case where the system is underdetermined, the minimal ℓ_1-norm solution is also the sparsest solution (Donoho, 2006b), thus we can instead recover the sparse signal $\mathbf{x}$ by solving for

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ subject to } \ A\mathbf{x} = \mathbf{y} \ , \tag{2.6}$$

thus the problem becomes a convex optimization problem.

This method raises three important issues, the first one being that, in practice, the assumption of sparsity on $\mathbf{x}$ is usually not enforced, especially when the signal is high-dimensional, thus, it cannot be expected to work well on images. The second problem of this approach is that it requires to solve an optimization problem for each sample. Even if the compressed sensing approach allows for the problem to be formulated as linear programming, which can be solved in polynomial time, it is still computationally expensive. Finally, the third issue is that the measurement matrix A does not necessarily respect the RIP. This is a problem since the RIP guarantees the coherency of the reconstructed sample. However, verifying that the matrix A respects the RIP is NP-hard in general. While several approaches for image compression uses techniques for generating random matrices that have a high probability of respecting the RIP (Rudelson & Vershynin, 2008; Rauhut, 2010), there are no guarantees in the case when A is fixed, such as image reconstruction.

Compressed sensing with sparse representation

When aiming to recover high-dimensional signals such as images, the assumption of sparsity on $\mathbf{x}$ is unrealistic, since it would mean that most of the image actually consists in black pixels. This requirement can however be replaced by the more generic approach of considering sparsity in another basis. Let B be a basis such that for $\mathbf{x} = \mathrm{B}\mathbf{s}$, the majority of the coefficients of $\mathbf{s}$ are zero. Thus, the problem becomes

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathrm{B}\mathbf{s}\|_1 \text{ subject to } \mathrm{AB}\mathbf{s} = \mathbf{y} \ , \\ \hat{\mathbf{x}} = \mathrm{B}\hat{\mathbf{s}} \ . \tag{2.7}$$

By either carefully selecting B, such as a Fourier or wavelet bases (Mallat, 2008), **basis pursuit** (Shaobing & Donoho, 1994; Donoho, 2006a) is much more robust and provides good results in real-world situations, for example in medical imaging (Lustig et al., 2008), image acquisition (Kolev, 2011; Duarte et al., 2008).

Another category of approaches is to learn a basis $\hat{\mathrm{B}}$ as a dictionary using a dataset of samples $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_\mathrm{K}\}, \mathbf{x}_i \in \mathbb{R}^{n \times m \times c}$ such that $\mathbf{x}_i = \hat{\mathrm{B}}\mathbf{s}_i$, where $\mathbf{s}_i$ is sparse.

This can be formulated as solving for

$$\hat{\mathrm{B}} = \arg\min_{\mathrm{B}, \{\mathbf{s}_i\}} \sum_{i=1}^{\mathrm{K}} \|\mathrm{B}\mathbf{s}_i - \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{s}_i\|_0 \ , \tag{2.8}$$

where λ is a parameter that controls the trade-off between the quality of the reconstruction and the sparsity of the representation. Again, solving this problem is NP-hard thus, in practice, we use search for the ℓ_1-norm solution, by solving for

$$\hat{\mathrm{B}} = \arg\min_{\mathrm{B}, \{\mathbf{s}_i\}} \sum_{i=1}^{\mathrm{K}} \|\mathrm{B}\mathbf{s}_i - \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{s}_i\|_1 \ . \tag{2.9}$$

Several algorithms exist for solving this problem, usually by iteratively updating the basis $\hat{\mathrm{B}}$ and the representations $\mathbf{s}_i$ alternatively. Examples of such algorithms are LASSO (Tibshirani, 1996), the method of optimal directions (Engan et al., 1999), K-SVD (Aharon et al., 2006), stochastic gradient descent or the Lagrange dual method.

Generative modeling as a prior to compressed sensing

Compressed sensing-based methods for image reconstruction have the advantage of explicitly modeling the constraints, which ensures that they will be enforced in the reconstructed image,. However, there are no guarantees on the quality of the reconstruction procedure if the measurement matrix A does not satisfy the RIP Equation (2.3). In the case of the image reconstruction process, this means that while the reconstructed image $\hat{\mathbf{x}}$ is guaranteed to enforce the constraints, it is not necessarily close to the real data distribution $p_\mathbf{x}$.

To overcome these problems, **Compressed Sensing with Meta-Learning** (Wu et al., 2019) extend compressed sensing by replacing the sparsity assumptions on the signal $\mathbf{x}$ with a learned prior on the data distribution p_X, which is done using a generative model G. By first generating an image $\mathrm{G}(\mathbf{z})$ and then exploring the latent space $\mathscr{Z}$ of the generative model G by minimizing $\|\mathrm{AG}(\mathbf{z}) - \mathbf{y}\|_2^2$. This is done so that altering the generated

image gives $\hat{\mathbf{y}} = \mathrm{AG}(\mathbf{z})$ where $\hat{\mathbf{y}}$ is as close as possible to $\mathbf{y}$. Then, Compressed sensing with meta-learning trains the generative model G to enforce the RIP (Equation (2.3)) so that it does not try to map all $\mathrm{G}(\mathbf{z})$ into the null space of A. The overall problem induced by this approach is formulated as

$$\begin{aligned}\min_{\mathrm{G}} \mathrm{L}(\mathrm{G}) = \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim p_{\mathrm{X}}\\ \mathbf{y}\sim p_{\mathrm{Y}}\\ \mathbf{z}\sim p_{\mathrm{Z}}}} \Big(&(\|\mathrm{A}(\mathbf{x}-\mathrm{G}(\mathbf{z}))\|_2^2 - \|\mathbf{x}-\mathrm{G}(\mathbf{z})\|_2^2)^2 + (\|\mathrm{A}(\mathbf{x}-\mathrm{G}(\hat{\mathbf{z}}))\|_2^2 - \|\mathbf{x}-\mathrm{G}(\hat{\mathbf{z}})\|_2^2)^2 \\ &+ (\|\mathrm{A}(\mathrm{G}(\mathbf{z})-\mathrm{G}(\hat{\mathbf{z}}))\|_2^2 - \|\mathrm{G}(\mathbf{z})-\mathrm{G}(\hat{\mathbf{z}})\|_2^2)^2\Big)/3 + \|\mathbf{y}-\mathrm{AG}(\hat{\mathbf{z}})\|_2^2 \\ &\text{where } \hat{\mathbf{z}} = \min_{\mathbf{z}} \|\mathbf{y}-\mathrm{AG}(\mathbf{z})\|^2 \quad . \end{aligned} \tag{2.10}$$

This objective try to minimize the difference between the distances among samples (generated or real) and the distances among samples altered by A. Solving this problem induces pushes the generator towards producing samples on which the RIP of A is respected. This implies that the generated samples will have a high likelihood on the real data distribution. Note that, in practice, $\hat{\mathbf{z}}$ is computed with gradient descent on $\mathbf{z}$ by minimizing $\|\mathbf{y}-\mathrm{AG}(\mathbf{z})\|_2^2$, starting from a random $\mathbf{z} \sim p_{\mathrm{Z}}$.

Deep Compressed Sensing (Wu et al., 2019) extend even further compressed sensing by replacing the (usually random) measurement matrix A in the Compressed sensing with Meta-Learning approach by a learned measurement function f_θ, so that the altered sample becomes $\tilde{\mathbf{y}} = f_\theta(\mathbf{x})$. Then, Deep Compressed sensing consists in training, in the same fashion as the GAN algorithm, G and f_θ by alternate gradient descent. This induced optimization problem is therefore

$$\min_{\mathrm{G}} \mathrm{L}_{\mathrm{G}} = \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{Y}}} \|\mathbf{y} - f_\theta(\mathrm{G}(\hat{\mathbf{z}}))\|_2^2 \tag{2.11}$$

$$\min_{\theta} \mathrm{L}_{f_\theta} = \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim p_{\mathrm{X}}\\ \mathbf{z}\sim p_{\mathrm{Z}}}} \Big(\sum_{\substack{\mathbf{x}_1,\mathbf{x}_2\in\mathscr{X}\cup\{\mathrm{G}(\mathbf{z})\}\\ x_1\neq\mathbf{x}_2}} ((\|f_\theta(\mathbf{x}_1-\mathbf{x}_2)\|_2^2 - \|\mathbf{x}_1-\mathbf{x}_2\|_2^2)^2\Big)/3 \quad . \tag{2.12}$$

In the same fashion as the objective of the previous method (Equation (2.10)), solving the problem 2.12 pushes f_θ towards respecting the RIP. Wu et al. (2019) showed that optimizing these two criterions trains both the generator G and the measurement function f_θ, thus replacing the discriminator of a more classical GAN framework. As a benefit, the approach may generate an image $\hat{\mathbf{x}} = \mathrm{G}(\hat{\mathbf{z}})$ from a noisy information $\mathbf{y}$ but at a high computation burden since it requires to solve an optimization problem (computing $\hat{\mathbf{z}}$) at inference stage for generating an image.

2.3.2 Conditional generation for image reconstruction

As opposed to the aforementioned methods, approaches based on conditional generation try to learn the conditional distribution $p_{\mathrm{X}|\mathrm{Y}}$ with a dataset of samples $\mathscr{X} = \{\mathbf{x}_1, ...\mathbf{x}_s\}$ and either aims to generate the most probable solution or provide a sampling mechanism over potential solutions.

In the case of image reconstruction, a generative model G which input is constraint map $\mathbf{y} \in \mathbb{R}^{n\times p\times c}$ learns to generate an image satisfying the constraints while likely following the distribution p_{X} (see Figure 2.2). For a generative model to provide a sampling

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

Figure 2.2: Generation of a sample during training. We first sample an image from a training set (2.2a) and we compute the constraints (2.2b) from it. Our GAN use it to generate a sample (2.2c). The constraints with squared error smaller than $\epsilon = 0.1$ are deemed satisfied and shown by green pixels in (2.2d) while the red pixels are unsatisfied (Best viewed in colors).

mechanism, the common solution consists in relying on a random vector $\mathbf{z}$ sampled from a known distribution p_Z (usually uniform or Gaussian) over a space $\mathbb{Z}$ that will be used as a latent variable for the model.

Conditional generative adversarial networks for image reconstruction

Although CGAN was initially designed for class-conditioned image generation by setting $\mathbf{y}$ as the class label of the image, it can naturally be applied to several types of conditioning information, including constraint maps, thus obtaining an image reconstruction with a high likelihood on the conditional distribution $p_{X|Y}$ is equivalent to taking a sample or image $\hat{\mathbf{x}} = G(\mathbf{y}, \mathbf{z})$, with $\mathbf{z} \sim p_Z$, using the generative model G solution to the problem

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{X,Y}} [\log D(\mathbf{x}, \mathbf{y})] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}} [1 - \log D(G(\mathbf{y}, \mathbf{z}), \mathbf{y})] \ ,$$

where $\mathbf{y}$ is the constraint map and D is the discriminator network.

While using the CGAN approach alone could theoretically be enough to solve the tasks of image reconstruction and inpainting, as it directly learns the conditional distribution of the samples, the most efficient approaches rely on extending the CGAN with a reconstruction loss, such as a ℓ_1 or ℓ_2 norm, between the pixels known beforehand and the corresponding pixels in the generated sample. This has been often carried out for the inpainting task (Pathak et al., 2016; Xiang et al., 2017), and can be formulated (in the case of the ℓ_2 norm) as finding a generator G and related discriminator D that optimize

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{X,Y}} [\log D(\mathbf{x}, \mathbf{y})] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}} [1 - \log D(G(\mathbf{y}, \mathbf{z}), \mathbf{y})] + \|M_\mathbf{y} \odot G(S\mathbf{y}, \mathbf{z}) - \mathbf{y}\|_2^2 \ ,$$

These approaches are often extended with techniques, such as using multiple discriminators (Yu et al., 2018; Armanious et al., 2019), extending the training with extra information and features (Armanious et al., 2019) such as medical imaging modalities, or using style losses (Guo et al., 2019) (See Section 1.2.2). However, several of these CGAN-based inpainting methods (Demir & Unal, 2018) rely on generating a patch that will fill up a structured missing part of the image and achieve impressive results. As such, they are not well suited to reconstruct very sparse and unstructured signal.

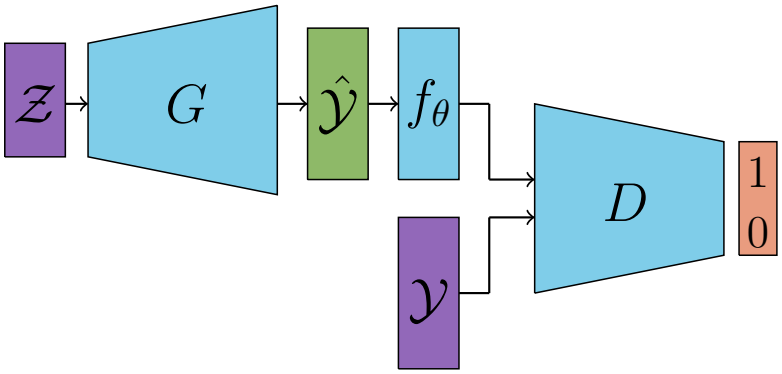


Figure 2.3: AmbientGAN

Unsupervised image reconstruction with generative adversarial networks

Another trend of approaches aims to reconstruct images without any knowledge of the real data distribution p_X, in other words they only hinge on datasets of altered samples $\mathbf{y} \sim p_Y$. This problem is different from the one we tackle, since ours supposes that a dataset of unaltered samples $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_s\}, \mathbf{x}_i \in \mathbb{R}^{n \times m \times c}$ is available. Among these approaches is **Ambient GAN** (Bora et al., 2018) (Figure 2.3), which aims at training an unconditional generative model using a dataset of noisy or incomplete samples $\mathcal{Y} = \{\mathbf{y}_1, ...\mathbf{y}_t\}, \mathbf{y}_i \in \mathbb{R}^{n \times m \times c}$. Ambient GAN attempts to produce unaltered images $\tilde{\mathbf{x}}$ which distribution matches the true one without having access to any of the original images $\mathbf{x}$. For the sake, Ambient GAN considers lossy measurements such as blurred images, images with removed patch or removed pixels at random (up to 95%), leading to sparse pixel map $\mathbf{y}$. This lossy measurement is simulated with a parameterized alteration function f_θ instead of the matrix A

$$\mathbf{y} = f_\theta(\mathbf{x}) \quad . \tag{2.13}$$

The underlying optimization problem solved by Ambient GAN is therefore stated as

$$\min_G \max_D L(D,G) = \mathop{\mathbb{E}}_{\mathbf{y} \sim p_Y} \Big[\log(D(\mathbf{y})) \Big] + \mathop{\mathbb{E}}_{\substack{\mathbf{z} \sim p_Z \\ \theta \sim p_\theta}} \Big[\log(1 - D(f_\theta(G(\mathbf{z})))) \Big] \quad . \tag{2.14}$$

Here, the discriminator has no knowledge of the distribution of the full images p_X, as its input is either real altered samples $\mathbf{y}$ or generated samples $G(\mathbf{z})$ on which the alteration function f_θ is applied. Thus, the Ambient GAN generator network G actually learns to generate samples $\hat{\mathbf{x}} = G(\mathbf{z})$ that, once f_θ is applied on them, are close to the real$\mathbf{y}$. This is equivalent to learning to invert the function f_θ. The Ambient GAN process is described in Figure 2.3.

Unsupervised Image Reconstruction (UNIR) (Pajot et al., 2019) extends the AmbientGAN approach by adding a conditioning to the model, which allows for the reconstruction of an image $\mathbf{x}$ from an altered image $\mathbf{y} \sim p_Y$, without any knowledge of the real data distribution p_X. UNIR is deterministic and does not allow for sampling, as the only input of the model is the altered image $\mathbf{y}$. For this, an additional reconstruction task is considered. It consists in first generating a reconstruction $\tilde{\mathbf{x}} = G(\mathbf{y})$, then applying the alteration function f_θ to the generated image $\tilde{\mathbf{x}}$ to get $\tilde{\mathbf{y}} = f_\theta(G(\mathbf{y}))$; re-generating an image

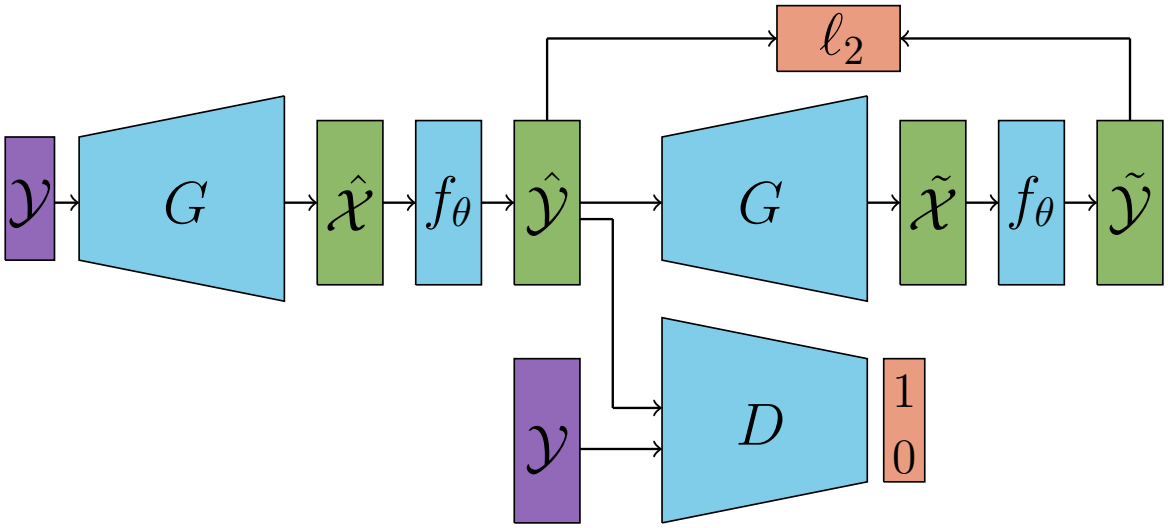


Figure 2.4: Unsupervised Image Reconstruction

as $\hat{\mathbf{x}} = f_\theta(\mathrm{G}(f_\theta(\mathrm{G}(\mathbf{y}))))$ and finally re-applying $f\theta$ to the image $\hat{\mathbf{x}}$ to get $\hat{\mathbf{y}} = f_\theta(\mathrm{G}(f_\theta(\mathrm{G}(\mathbf{y}))))$. This procedure can be deemed as

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{Y}}}\Big[\log(\mathrm{D}(\mathbf{y}))\Big] + \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{Y}}}\Big[\log(1-\mathrm{D}(\hat{\mathbf{y}}))\Big] + \|\hat{\mathbf{y}}-\tilde{\mathbf{y}}\|_2^2 \ . \tag{2.15}$$

Here, the ℓ_2 norm term ensures that the generator is able to learn to revert f_θ i.e. to revert the alteration procedure on a given sample. This allows the reconstruction of realistic image $\hat{\mathbf{x}}$ only from a given constraint map $\mathbf{y}$. The full process is described in Figure 2.4.

In another fashion, **Semantic Inpainting by Constrained Image Generation** (Yeh et al., 2017) is an approach for inpainting which considers the generator G of a pre-trained GAN as a prior on the data distribution p_{X}, and explores its latent space $\mathbb{Z}$ through an optimization procedure to find a latent vector $\mathbf{z}$, which induces an image with missing regions filled in by conditioning on the surroundings available information. To ensure that the reconstruction is accurate, this approach uses the discriminator D as a prior instead of ensuring the RIP. This is done by adding the discriminator loss to the reconstruction loss, so that it prevents the optimization procedure from providing images that are too far away from the real data distribution. As such, the problem becomes $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{z}^*)$ with $\mathbf{z}^*$ minimizing

$$\min_{\mathbf{z}} \|A\mathrm{G}(\mathbf{z}) - \mathbf{y}\|_2^2 + \lambda \log(1-\mathrm{D}(\mathrm{G}(\mathbf{z}))) \ , \tag{2.16}$$

where λ is a hyperparameter. To yield on an image satisfying some given constraints $\mathbf{y}$, the method requires to solve a full optimization problem for each sample to reconstruct, which is computationally expensive.

Approach	**Dataset free**	**One-step reconstruction**	**Sampling mechanism**	**Constraints enforcement**
Compressed sensing-based				
Compressed sensing (Candes & Tao, 2005)	✓	✗	✗	Exact
Compressed sensing with dictionary learning (Donoho, 2006a)	✗	✗	✗	Exact
Compressed sensing with Meta Learning (Wu et al., 2019)	✗	✗	✗	Control parameter
Deep compressed sensing (Wu et al., 2019)	✗	✗	✓	Control parameter
Generative modeling-based				
Conditional GAN (Mirza & Osindero, 2014)	✗	✓	✓	Implicit
Ambient GAN (Bora et al., 2018)	✗ (altered*)	✗	✗	Explicit
UNIR (Pajot et al., 2019)	✗ (altered*)	✓	✗	Explicit
Constrained image generation (Yeh et al., 2017)	✗	✗	✓	Control parameter
Our approach (see Section 2.4)	✗	✓	✓	Control parameter

* Only altered samples are required during training

Table 2.1: Summary of the different advantages and limitations of all the aforementioned methods for image reconstruction. We consider the need for a dataset, the necessity of solving an optimization problem for each generated sample, the ability to sample different solutions and the mechanism for enforcing the good reconstruction of the constraints.

2.4 Image reconstruction as an auxiliary task to generative modeling

As we have seen, two main categories of solutions aim to tackle the problem of image reconstruction. First, the approaches that try to directly solve the problem by finding a solution through optimization, among them is compressed sensing. However, the main drawback of these categories of approaches are that they require to solve an optimization problem per reconstructed image, which is computationally expensive in the cases where a lot of samples need to be reconstructed. Then, the approaches that aims to learn the conditional data distribution $p_{X|Y}$ and try to reconstruct the image by sampling on this distribution. Among these approaches are CGAN-based methods, norm-based methods, AmbientGAN or UNIR. While they have the advantage of only requiring to solve a unique optimization problem at training instead of one for each reconstructed sample, they do require a dataset to train on.

As main contribution to this chapter, we introduce a GAN model whose generation network takes as input the constraint map $\mathbf{y}$ and the sampled latent code $\mathbf{z} \in \mathbb{Z}$ and outputs a realistic image that fulfills the prescribed pixel values. Within this setup, such a generative model can sample in a single step from the unknown distribution p_X of the training images $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ while satisfying unseen pixel-wise constraints at training stage.

2.4.1 Image reconstruction as a maximum a posteriori estimation

Starting from the image reconstruction problem ((Equation (2.1)), estimating the maximum a posteriori consists in finding an image $\hat{\mathbf{x}}$ that has the maximum likelihood on the posterior distribution $p_{X|Y}$, in other words the image that is the most likely to be the original image, from which the constraints $\mathbf{y}$ has been measured. We find that this allows for replacing the implicit conditioning of the CGAN by a norm-based reconstruction loss is an approach that naturally emerges from the denoising formulation Equation (2.13). This provides a rationale for the use of these losses in the similar aforementioned approaches (Section 2.3.2).

Such a generative model offers does not rely on per-sample optimization and provides a simple and efficient sampling mechanism through the latent variable $\mathbf{z}$. This allows for the efficient sampling of several potential solutions, which can be crucial in some applications in which a large amount of solutions need to be sampled, such as solving inverse problems (Laloy et al., 2019) . It also naturally provides a mechanism for controlling the trade-off between the respect of the constraints and the likelihood of the reconstructed image.

Conditional generative models for maximum a posteriori estimation

Recall that the image reconstruction problem consist in recovering $\mathbf{x}$ the solution, assuming the constraint map $\mathbf{y}$ is resulting from applying a $M_{\mathbf{y}}$ on the image $\mathbf{x}$ (Equation (2.1)), to

$$\mathbf{y} = M_{\mathbf{y}} \odot \mathbf{x} \ . \tag{2.17}$$

Here $M_{\mathbf{y}}$ is the masking matrix in which the constrained pixels are assumed to be randomly and independently selected. We can formulate the Maximum A Posteriori (MAP) estimation of this problem, which, given the constraint map $\mathbf{y}$, consists in finding the most probable image $\mathbf{x}^*$ following the posterior distribution $p_{X|Y}$, as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \log p_{X|Y}(\mathbf{x}|\mathbf{y}) + \log p_Y(\mathbf{y}) \tag{2.18}$$

$$= \arg\max_{\mathbf{x}} \log p_{Y|X}(\mathbf{y}|\mathbf{x}) + \log p_X(\mathbf{x}) \ . \tag{2.19}$$

$p_{Y|X}(\mathbf{y}|\mathbf{x})$ is the likelihood that the constrained pixels $\mathbf{y}$ are issued from image $\mathbf{x}$ while $p_{X|Y}(\mathbf{x}|\mathbf{y})$ is the likelihood of an image knowing the constrained pixels $\mathbf{y}$. $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$ represent the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$. As such, we can introduce a conditional generative model $G : (\mathbb{Y}, \mathbb{Z}) \to \mathbb{X}$ in order to that replaces the conditional distribution $p_{X|Y}$, which allows for reformulating the original problem (Equation (2.17)) as

$$\mathbf{y} = M_{\mathbf{y}} \odot G(\mathbf{y}, \mathbf{z}) + \epsilon_G \ , \tag{2.20}$$

where ϵ_G represents the error of the model, which we consider to by an i.i.d noise corrupting the constrained pixels. Assuming that the generation network G may sample an image $G(\mathbf{y}, \mathbf{z})$ complying with the given pixel values $\mathbf{y}$, we get the following problem

$$\max_G \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}} \log p_{Y|X}(\mathbf{y}|G(\mathbf{y}, \mathbf{z})) + \log p_X(G(\mathbf{y}, \mathbf{z})) \ . \tag{2.21}$$

The first term in Problem (2.21) measures the likelihood of the constraints given a generated image. The second term measures the likelihood of the generated images according to the data distribution p_X. Let rewrite Equation (2.21) as $\text{vect}(\mathbf{y}) = \text{vect}(M_{\mathbf{y}} \odot \mathbf{x}) + \text{vect}(\varepsilon)$ where $\text{vect}(\cdot)$ is the vectorisation operator that consists in stacking the pixels, with $\text{vect}(\mathbf{y}) \in \mathbb{R}^{n.m.c}$ for $\mathbf{y} \in \mathbb{R}^{n\times m\times c}$. Therefore, assuming $\text{vect}(\varepsilon)$ is i.i.d and follows a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$, the conditional likelihood of $\mathbf{y}$ knowing $\mathbf{x}$ reads as

$$\log p_{Y|X}(\mathbf{y}, G(\mathbf{y}|\mathbf{z})) \propto -\left\|\text{vect}(\mathbf{y}) - \text{vect}(M_{\mathbf{y}}) \odot G(\mathbf{y}, \mathbf{z}))\right\|_2^2 \ . \tag{2.22}$$

It evaluates the euclidean distance between the conditioning pixels and their predictions by G. In other words, using a matrix notation of Equation (2.20), the latter conditional likelihood given a generated image equivalently writes

$$\log p_{Y|X}(\mathbf{y}, G(\mathbf{y}, \mathbf{z}) \propto -\left\|\mathbf{y} - M_{\mathbf{y}} \odot G(\mathbf{y}, \mathbf{z})\right\|_F^2 \ . \tag{2.23}$$

$\|N\|_F^2$ represents the squared Frobenius norm of matrix N that is the sum of its squared entries.

The second term in Problem (2.21) 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 $G(\mathbf{y}, \mathbf{z})$. This amounts to minimizing with respect to G, the GAN-like objective function $\mathbb{E}_{\mathbf{x}\sim p_X} \log(D(\mathbf{x})) + \mathbb{E}_{\substack{\mathbf{z}\sim p_Z \\ \mathbf{y}\sim p_Y}} \log(1 - D(G(\mathbf{y}, \mathbf{z})))$ (Goodfellow et al., 2014).

Putting altogether these elements, we can propose a relaxation of the hard constraint optimization problem (2.2) (Figure 2.5) that consists in learning on a generative network G and the related discriminator model D by solving

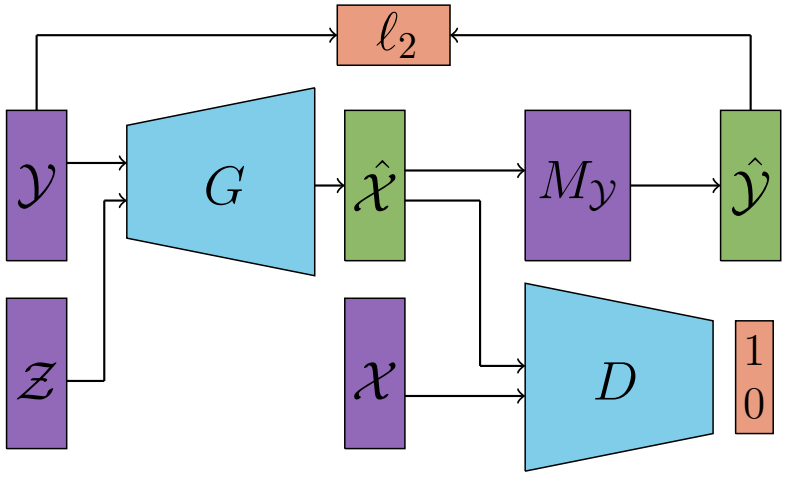


Figure 2.5: Our approach

$$\begin{aligned} \min_{G}\max_{D} L(D,G) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\Big[\log(D(\mathbf{x}))\Big] \\ &+ \mathop{\mathbb{E}}_{\substack{\mathbf{z}\sim p_Z \\ \mathbf{y}\sim p_Y}}\Big[\log(1-D(G(\mathbf{y},\mathbf{z}))) + \lambda \left\|\mathbf{y} - M_{\mathbf{y}} \odot G(\mathbf{y},\mathbf{z})\right\|_F^2\Big] \ . \end{aligned} \tag{2.24}$$

It is worth to note that the assumption of Gaussian noise measurement leads us to explicitly turn the pixel value constraints into the minimization of the quadratic error between the real enforced pixel values and their generated counterparts as it corresponds to maximizing the conditional likelihood of the pixels in the generated image. The additional term acts as a regularization over prescribed pixels by the mask $M_{\mathbf{y}}$. The trade-off between the distribution matching loss and the constraint enforcement is assessed by the regularization parameter $\lambda \geq 0$. Figure 2.2d illustrates the overall principle of the model. It is also worth noting that the noise ϵ can be of any other distribution, according to the prior information, one may associate to the measurement process. To formulate the maximum a posteriori, we however 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).

Conditional image generation with an image reconstruction auxiliary task

To solve Problem (2.24), we use stochastic gradient descent. 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 2.6). This issue manifests itself through the fact that the learned generation network, given a constraint map $\mathbf{y}$, outputs almost deterministic image regardless the variations in the input $\mathbf{z}$. The issue was also pointed out by Yang et al. (Yang et al., 2019) as characteristic of CGANs. To avoid the problem, we exploit the PacGAN (Lin et al., 2018) technique , detailed in Section 1.2.2, which consists in passing a small set of samples to the discrimination function instead of a single

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 $\{\mathbf{x}_i\}_{i=1}^{m}$ from $\mathscr{D}_{\mathrm{X}}$

sample a mini-batch of masks $\{\mathrm{M}_{\mathbf{y}_i}\}_{i=1}^{m}$ and compute the labels $\mathbf{y}_i = \mathrm{M}_{\mathbf{y}_i} \odot \mathbf{x}_i$

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

update D by stochastic gradient ascent of

$\sum_{i=1}^{m} \log(\mathrm{D}(\mathbf{x}_i)) + \log(1 - \mathrm{D}(\mathrm{G}(\mathbf{y}_i, \mathbf{z}_i)))$

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

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

update G by stochastic gradient descent of

$\sum_{j=1}^{n} \log(1 - \mathrm{D}(\mathrm{G}(\mathbf{y}_j, \mathbf{z}_j))) + \lambda \|\mathbf{y}_j - \mathrm{M}_{\mathbf{y}_j} \odot \mathrm{G}(\mathbf{y}_j, \mathbf{z}_j)\|_{\mathrm{F}}^2$

until a stopping condition is met

one. PacGAN is intended to tackle the mode collapse problem in GAN training (see Section 1.1.4) . 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.

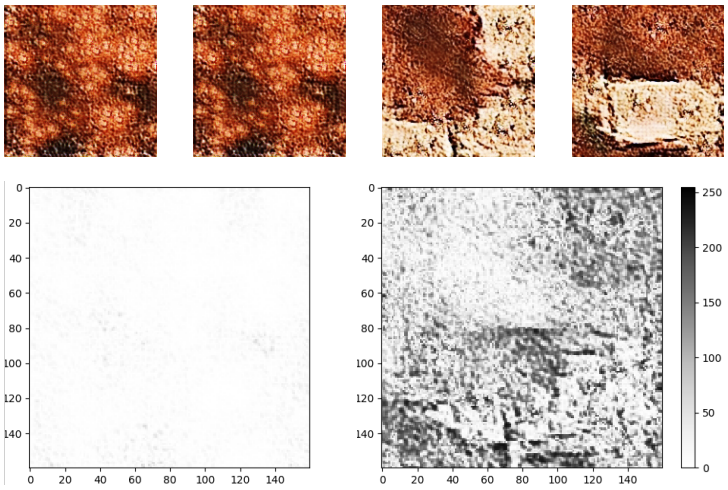


Figure 2.6: An example of a loss of diversity when generating brick texture samples (see 2.4.2) using two different random noises **z** and a single constraint map **y**. The two samples on the top left are generated using the classical GAN discriminator whereas the samples on the top right are generated using the PacGAN approach. The loss of diversity is clearly visible on the absolute differences between the greyscaled images (bottom).

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 $\{\mathbf{x}_i^a\}_{i=1}^m$, $\{\mathbf{x}_i^b\}_{i=1}^m$ from $\mathscr{D}_{\mathrm{X}}$

sample a mini-batch of masks $\{\mathrm{M}_{\mathbf{y}_i}\}_{i=1}^m$ and compute the labels $\mathbf{y}_i = \mathrm{M}_{\mathbf{y}_i} \odot \mathbf{x}_i^a$

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

update D by stochastic gradient ascent of

$\sum_{i=1}^m \log(\mathrm{D}(\mathbf{x}_i^a, \mathbf{x}_i^b)) + \log(1 - \mathrm{D}(\mathrm{G}(\mathbf{y}_i, \mathbf{z}_i^a), \mathrm{G}(\mathbf{y}_i, \mathbf{z}_i^b)))$

sample a mini-batch of masks $\{\mathrm{M}_{\mathbf{y}_i}\}_{i=1}^m$ and compute the labels $\mathbf{y}_i = \mathrm{M}_{\mathbf{y}_i} \odot \mathbf{x}_i$

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

update G by stochastic gradient descent of

$\sum_{j=1}^m \log(1-\mathrm{D}(\mathrm{G}(\mathbf{y}_j, \mathbf{z}_j^a), \mathrm{G}(\mathbf{y}_j, \mathbf{z}_j^b))) + \lambda \|\mathbf{y}_j - \mathrm{M}_{\mathbf{y}_j} \odot \mathrm{G}(\mathbf{y}_j, \mathbf{z}_j^a)\|_{\mathrm{F}}^2 + \lambda \|\mathbf{y}_j - \mathrm{M}_{\mathbf{y}_j} \odot \mathrm{G}(\mathbf{y}_j, \mathbf{z}_j^b)\|_{\mathrm{F}}^2$

until a stopping condition is met

2.4.2 Experimental results and application

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 2.4.2 is devoted to the results presentation.

Datasets

We tested our approach on several datasets listed hereafter. Detailed information on these datasets are provided in the Appendix B.1.

FashionMNIST (Xiao et al., 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.

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. To do so, a fifth of each set is used to generate the constrained pixel map $\mathbf{y}$ 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 2.4.2) computed on the validation set. Finally, reported results are computed on the test set.

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 B.2.

For the experiments on the FashionMNIST (Xiao et al., 2017), we use a lightweight convolutional network for both the discriminator and the generator, similar to DCGAN (Radford et al., 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., 2017), or encoder-decoder architectures that are commonly used in domain-transfer applications such as CycleGAN (Zhu et al., 2017b).

We keep the PatchGAN discriminator (Isola et al., 2016) across all the experiments with these architectures, 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 et al., 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.

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 2.4.1), we assess the constraint fulfillment using the following mean square error (MSE)

$$\text{MSE} = \frac{1}{\text{L}} \sum_{i=1}^{\text{L}} \left\| \mathbf{y}_i - \text{M}_{\mathbf{y}_i} \odot \text{G}(\mathbf{y}_i, \mathbf{z}_i) \right\|_{\text{F}}^2 \quad . \tag{2.25}$$

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 et al., 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. These approaches both consist in comparing, for both real and generated images, features extracted with a pre-trained classifier. We explain these approaches in section 1.2.4 of the chapter 1. We employ FID since the Inception Score has been shown to be less reliable (Barratt & Sharma, 2018). 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, the dataset is not labeled, hence we resort to a CNN classifier trained on the Describable Textures Dataset (DTD) (Cimpoi et al., 2014), which is a related application domain.

For the Subsurface dataset, there is not labels nor similar labeled dataset. Thus, we could not train a classifier for this dataset, so 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 from generated ones. Similarly to (Ruffino et al., 2017), 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 B.3.

Finally, we check by visual inspection if the trained model G is able to generate diverse samples, meaning that for a given $\mathbf{y}$ and for a set of latent codes $(\mathbf{z}_1, ..., \mathbf{z}_n) \sim p_Z$, the generated samples $\text{G}(\mathbf{y}, \mathbf{z}_1), \ldots, \text{G}(\mathbf{y}, \mathbf{z}_n)$ are visually different.

Study of the quality-fidelity trade-off

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

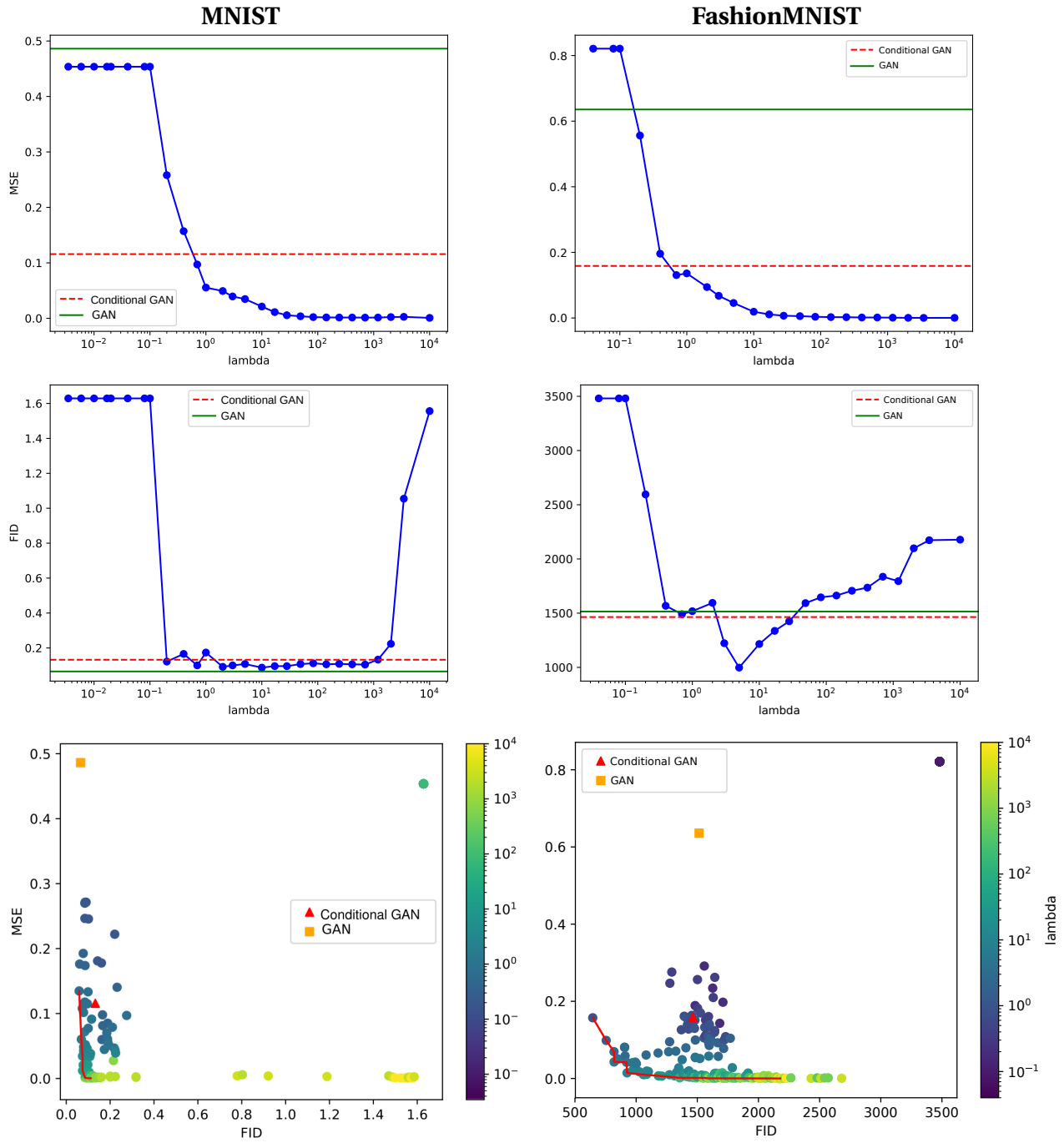


Figure 2.7: Our approach compared to the GAN and CGAN baselines. MSE (Top) and FID (center) w.r.t. the regularization parameter λ; MSE w.r.t the FID (bottom), on the MNIST (left) and Fashion MNIST (right) datasets. Note that the different orders of magnitude for the FID is due to the different classifiers used to compute this distances.

that minimizes the cost

$$\mathrm{C}(\mathrm{FID},\mathrm{MSE}) = \sqrt{\left(\frac{\mathrm{FID}-\mathrm{FID}_{min}}{\mathrm{FID}_{max}-\mathrm{FID}_{min}}\right)^2 + \left(\frac{\mathrm{MSE}-\mathrm{MSE}_{min}}{\mathrm{MSE}_{max}-\mathrm{MSE}_{min}}\right)^2}$$

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 2.7) 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 2.7 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 2.7).

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 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 performances are detailed in Table 2.2.

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 2.6), 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 2.2).

Table 2.3 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 2.3) by a large margin on the respect of conditioning pixels (see the achieved MSE metrics by

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	✓
Res	0.0458	0.0474	0.0590	0.0476	✗
UNet	0.0442	0.1789	0.0964	0.4559	✗
UNetRes	0.0382	0.0307	0.0499	0.0338	✗
ResPAC	**0.0350**	0.0698	0.0466	0.4896	✓
UNetPAC	0.0672	$\leq$ **0.0001**	0.3120	0.2171	✓
UNetResPAC	0.0431	0.0277	**0.0447**	**0.0302**	✓

Table 2.2: 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	$\leq$ **0.0001**	0.3120	0.2171
Ours-UNetResPAC	0.0431	0.0277	0.0447	**0.0302**

Table 2.3: Results obtained by the selected best fully-convolutional architectures on the Texture dataset for both the CGAN approach and our approach.

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.

High-dimension image reconstruction

We extend the comparison of our approach to CGAN on the CIFAR10 and CelebA datasets (Table 2.4). We investigate all the architectures described in Section 2.4.2. According to the results obtained in Section 2.4.2, we fixed the regularization parameter 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 and the 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. However 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 2.4: 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 2.5: 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.

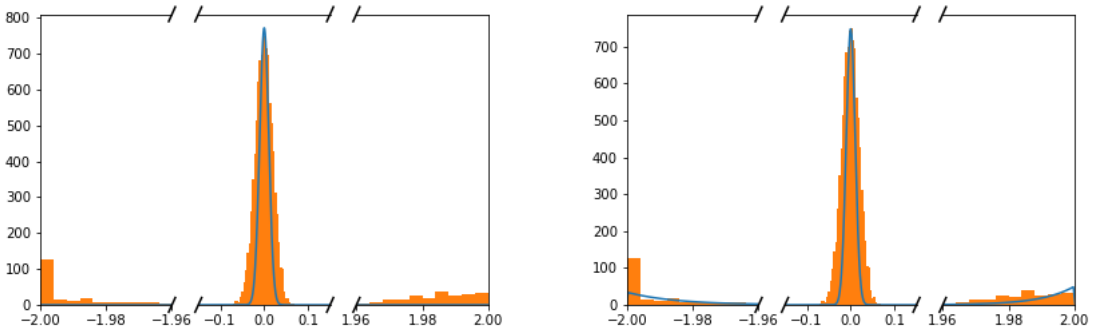
Application to hydro-geology

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 2.4.2. 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 2.4.2), 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 2.5 and 2.6. 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., 2017). Evaluation of the generated images using the domain-connectivity function highlights this fact on Figure B.2 in the supplementary materials. Also examples of generated images by our approach pictured in Figure B.4 (see appendix B.4) 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 2.6: Evaluation of the visual quality between the CGAN approach and ours on the Subsurface dataset using several metrics.



(a) Gaussian best fit

(b) Mixture distribution fit

Figure 2.8: In orange: Histogram of the reconstruction error of the UNetResPAC model on 100 generated subsurface images, with $\lambda = 1$. As we can see, the error is either close to 0, -2 or 2. In blue: Probability density functions of a best fit Gaussian and a mixture model of a Gaussian distribution with a weight of 0.95 and two exponential distribution, each with weights of 0.025 (0.05 being roughly the error rate of the UNetResPAC model).

2.5 Conclusion and perspective

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, even when including the PacGAN technique, this approach allows for the use of fully-convolutional architectures and scales well to larger 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, an interesting direction would be to investigate other prior distributions for the given pixels. As mentioned in the section Section 2.4.1, we assume that the reconstruction error of the model is Gaussian. This however is not necessarily true in practice, as we observed that in the case of the Subsurface dataset, since the pixels are always either -1 or 1, the reconstruction error tend to be close to either 0, -2, or 2 (see Fig-

ure 2.8a). For this example, a mixture distribution could be more appropriate as it could model both the cases where the error is close to 0 (which can be assumed to be normal) and the cases where it is close to -2 or 2 (Figure 2.8b). This however raises the questions of finding a closed form solution for the maximum likelihood estimation for such a distribution.

On the other hand, applying the developed approach to other applications or signals such as audio inpainting (Marafioti et al., 2018) could also be an interesting perspective. Domains in which measuring points in any signal is costly or very noisy could benefit from an approach that allows fast sampling of potential solutions.

Chapter 3

Domain-transfer modeling with auxiliary tasks

Chapter abstract

In this chapter, we propose to examine the problem of constrained image domain-transfer with generative models. We focus on the generation of images using Cycle-Consistent Generative Adversarial Networks (CycleGAN) with added constraints. This work is driven by an application for generating images with constraints derived from the optics of polarimetry, motivated by the increasing popularity of the combination of deep learning frameworks with polarimetric imaging in various domains, including medical imaging and scene analysis. However, even if polarimetric imaging has shown improved performances on diverse tasks, such as detection in road scenes images, 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. To this purpose, we propose a framework based on the CycleGAN approach. We derive constraints from the optics of polarimetry that characterize the physical admissibility of a polarimetric image. By integrating these constraints as an auxiliary task during training stage, the model learns to generate high-quality polarimetric images that follow the physics constraints of polarimetry. This allows for transferring existing labeled RGB datasets to the polarimetric domain without the need for re-labeling the data. We evaluate the proposed generative model on road scene images. The obtained results achieved an effective generation of physical polarization-encoded images of high visual quality. The generated imaged are indeed coherent from a physics perspective. Further experiments on the task of road object detection show that by training on a polarimetric images dataset that includes generated images, the detection of cars and pedestrian are improved by up to 9%.

The work in this chapter has led to the submission of the following paper:

- Rachel Blin, Cyprien Ruffino, Samia Ainouz, Gilles Gasso, Romain Hérault, Stéphane Canu and Fabrice Meriaudeau (June. 2020). Generating Polarimetric-encoded Images using Constrained Cycle-Consistent Generative Adversarial Networks. CR: In: Asian Conference on Computer Vision 2020 (ACCV2020)

Contents

3.1 Introduction

Generative adversarial networks (Goodfellow et al., 2014) are powerful deep generative models that can learn complex data distributions and generate realistic samples from them. Arguably most of the impressive achievements of the GAN were obtained for RGB images but some works attempted to extend GAN approaches to other uncommon imaging domains. Among these works, we find the task of generating images from the RGB domain to these uncommon imaging domains, using domain-translation approaches such as CycleGAN (Zhu et al., 2017a). For instance, we find methods 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 and Sallab et al. (2019) used it to produce realistic LiDAR points cloud from simulated ones.

Following the previous stream of work, this chapter explores domain-transfer generative models through an application on non-conventional imaging techniques. Specifically we investigate 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 analyze 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 et al., 2017), depth map estimation (Zhu & Smith, 2019), 3D objects reconstruction (Morel et al., 2006), or early-stage cancer detection (Rehbinder et al., 2016). Also, polarization imaging was recently 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, thermal or infrared image generation which mostly responded to visual qualitative constraints, 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 labeling step by transferring a source labeled 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, our framework based on CycleGANis able to handle the physical polarization constraints during training.

We demonstrate the effectiveness of our constrained-output CycleGAN on the of Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset (Geiger et al., 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 chapter 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 pretraining, the generated images help improving the detection performances.

The remainder of the chapter is organized as follows: CR: NON ! 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.

3.2 Context and application

This section introduces the polarization formalism. CR: TODO

3.2.1 Introduction to polarimetric imaging

Polarization is a property of light that 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 un-polarized 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. As such, it is a different modality than classical color images since they do not represent the wavelength of light. CR: TODO

3.2.2 Mathematical formulation

The linear part of the reflected light can be described by measurable parameters, specifically by the linear Stokes vector $\mathbf{s} = \begin{bmatrix}\mathbf{s}_0 & \mathbf{s}_1 & \mathbf{s}_2\end{bmatrix}^\top \in \mathbb{R}^{n\times m\times 3}$. Here, $\mathbf{s}_0 > 0$ represents the total intensity, $\mathbf{s}_1$ the amount of horizontally and vertically linearly polarized light and $\mathbf{s}_2$ the amount of linearly polarized light at $\pm$ 45°. It is important to note that by design, any Stokes vector is physically admissible if and only if the two following conditions are met:

$$\mathbf{s}_0 > 0 \quad \text{and} \quad \mathbf{s}_0^2 \geqslant \mathbf{s}_1^2 + \mathbf{s}_2^2 \ . \tag{3.1}$$

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

$$\text{DOP} = \frac{\sqrt{\mathbf{s}_1^2 + \mathbf{s}_2^2}}{\mathbf{s}_0} \ .$$

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 un-polarized 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 et al., 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.

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°.

For this work, we rely on a polarimetric image encoding format that consists in four channel images respectively obtained with four different linear polarizers oriented at $\alpha_i, i \in \{1,...,4\} = (0°, 45°, 90°, 135°)$. The polarimetric camera captures an image $\mathbf{x} \in \mathbb{R}^{n \times m \times 4}$ consisting in the intensities $\mathbf{x}_{\alpha_i}$ of the scene for each angle α_i for each pixel. The relationship between the Stokes vector $\mathbf{s}$ and the intensities at angles $\mathbf{x}_{\alpha_i}, i \in \{1,...,4\}$ reaching the camera is given by:

$$\mathbf{x}_{\alpha_i} = \frac{1}{2}\begin{bmatrix}1 & \cos(2\alpha_i) & \sin(2\alpha_i)\end{bmatrix}\begin{bmatrix}\mathbf{s}_0 \\ \mathbf{s}_1 \\ \mathbf{s}_2\end{bmatrix} \quad , \forall i = 1,4$$

that is:

$$\mathbf{x} = \mathrm{A}\mathbf{s} \ , \tag{3.2}$$

where $\mathbf{x} = \begin{bmatrix}\mathbf{x}_0 & \mathbf{x}_{45} & \mathbf{x}_{90} & \mathbf{x}_{135}\end{bmatrix}^\top$ refers to the four intensities according to each angle of the polarizer $(\alpha_i)_{i=1:4}$ and $\mathrm{A} \in \mathbb{R}^{4\times 3}$, to the calibration matrix of the polarization camera, defined as:

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

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 $\mathrm{A}^\dagger = (\mathrm{A}^\top\mathrm{A})^{-1}\mathrm{A}^\top \in \mathbb{R}^{3\times 4}$ the pseudoinverse of the matrix A. The relationship between $\mathbf{s}$ and $\mathbf{x}$ is then defined by:

$$\mathbf{s} \quad = \quad \mathrm{A}^\dagger\mathbf{x} = \begin{bmatrix}1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1\end{bmatrix}\begin{bmatrix}\mathbf{x}_0 \\ \mathbf{x}_{45} \\ \mathbf{x}_{90} \\ \mathbf{x}_{135}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_0 + \mathbf{x}_{90} \\ \mathbf{x}_0 - \mathbf{x}_{90} \\ \mathbf{x}_{45} - \mathbf{x}_{135}\end{bmatrix} \ . \tag{3.3}$$

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

$$\mathbf{x} = \mathrm{A}\mathrm{A}^\dagger\mathbf{x} \ ,$$

which is satisfied if and only if:

$$\mathbf{x}_0 + \mathbf{x}_{90} = \mathbf{x}_{45} + \mathbf{x}_{135} \ . \tag{3.4}$$

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.

3.3 Conditional domain-transfer approaches

3.3.1 Constrained problem formulation

CR: TODO

3.3.2 Limits of domain transfer approaches

CR: TODO

3.3.3 Related works

CR: TODO: CyCADA, travaux d'Ahmed Rida Sekkat, ...

3.4 Approaches for solving the constrained domain-transfer problem

3.4.1 Relaxing the constraints

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 G_{XY} and G_{YX} between $\mathbb{X}$ the space of the polarimetric images and $\mathbb{Y}$ the RGB image domain. Let $\hat{\mathbf{x}} \in \mathbb{R}^{n\times m\times 4}$ be a generated polarimetric image. To be physically admissible, it 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:

$$\begin{aligned} \mathscr{C}_1 &: \mathbf{x} = \mathrm{A}\mathbf{s} \ , \\ \mathscr{C}_2 &: \mathbf{s}_0^2 \geqslant \mathbf{s}_1^2 + \mathbf{s}_2^2 \ , \\ \mathscr{C}_3 &: \mathbf{s}_0 > 0 \ . \end{aligned}$$

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 hyperbolic tangent as activation function, each output intensity $\hat{\mathbf{x}}$ is within the range $]-1,1[$ which we scale to $]0,255[$. Hence $\hat{\mathbf{s}}_0 = \hat{\mathbf{x}}_0 + \hat{\mathbf{x}}_{90}$ (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 (**??**) 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 min-max game involved in the CycleGAN learning.

CR: TODOWe aim to learn a generative model G_{XY} such that the generated images $\hat{\mathbf{y}}$ have a high likelihood on p_Y the distribution of the real polarimetric images and such that $\hat{\mathbf{s}} = A^\dagger \hat{\mathbf{y}}$ respects the constraints. Thus, we can formulate the problem as

$$\begin{aligned} \max_{G} \quad & L(G) = \mathbb{E}_{\mathbf{x}\sim p_X}\left[\log(p_Y(G_{XY}(\mathbf{x}))\right] \\ \text{s.c.} \quad & \mathbf{x}_i = A\mathbf{s}_i \;\; ; \;\; \mathbf{s}_{0_i}^2 \geq \mathbf{s}_{2_i}^2 + \mathbf{s}_{1_i}^2 \;\text{ and }\; \mathbf{s}_{0_i}^2 > 0 \\ \text{for} \quad & \mathbf{s}_i = A^\dagger(G_{XY}(\mathbf{x})_i) \quad \forall i \end{aligned} \tag{3.5}$$

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 G_{YX} and $A\hat{\mathbf{s}}$ is proposed. It reads

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

with $\hat{\mathbf{s}} = \begin{bmatrix}\hat{\mathbf{s}}_0 & \hat{\mathbf{s}}_1 & \hat{\mathbf{s}}_2\end{bmatrix}^\top$ the Stokes vector calculated from the generated image by G_{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:

$$L_{\mathscr{C}_2} = \mathop{\mathbb{E}}_{\mathbf{y}\sim p_Y} \max\left(\hat{\mathbf{s}}_1^2 + \hat{\mathbf{s}}_2^2 - \hat{\mathbf{s}}_0^2, 0\right) \;\; .$$

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:

$$L_{final} = L_{CycleGAN} + \mu L_{\mathscr{C}_1} + \nu L_{\mathscr{C}_2} \;\; . \tag{3.6}$$

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 (**??**)). 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.

3.4.2 Gradient projection

We aim to generate images $\hat{\mathbf{y}} = G_{XY}(\mathbf{x})$, where $\mathbf{y} \in \mathbb{Y}$ is a sample from the RGB domain, such that $\hat{\mathbf{s}} = A^\dagger \hat{\mathbf{y}} \in \mathbb{S}$ the space of the Stokes vectors. Each of the vectors must respect $\mathscr{C}_1$,

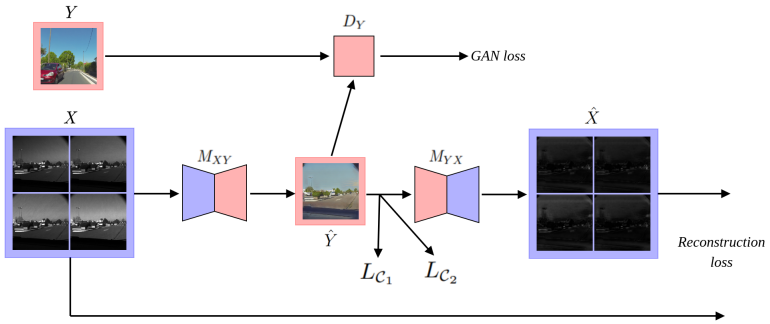


Figure 3.2: Overview of the CycleGAN training process extended with $L_{\mathscr{C}_1}$ and $L_{\mathscr{C}_2}$.

$\mathscr{C}_2$ and $\mathscr{C}_3$, which correspond to a second-order cone, or Lorentz cone (Boyd et al., 2004). Thus, let

$$\mathscr{C} = \left\{ (\mathbf{v}, \mathbf{t}) \in \mathbb{S} \mid \|\mathbf{v}\|_2 \leq \mathbf{t},\ \mathbf{t} = \mathbf{s}_0, \mathbf{v} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \right\} \ , \tag{3.7}$$

a convex set whose vectors satisfy the aforementioned constraints.

CR: On souhaite générer des images $\{A^\dagger(G(z))\}$ où $z \in \mathbb{R}^d$ est le bruit suivant une loi de distribution connue (loi normale ou loi uniforme), M l'opérateur (linéaire ??) permettant de passer de l'espace des images $\mathscr{I}$ à l'espace des vecteurs de Stokes $\in \mathscr{S}$. Chaque vecteur doit respecter la contrainte (**??**) qui est un cône de second ordre ou cône de Lorentz Boyd et al., 2004, Section 2.2.3. Soit l'ensemble convexe

$$\mathscr{C} = \left\{ (v, t) \in \mathscr{S} \mid \|v\|_2 \leq t,\ t = S_0, v = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \right\}$$

.

On cherche à déterminer un modèle génératif G couplé à un discriminateur D permettant de distinguer entre les images réelles et les images synthétiques et tel que chaque super-pixel $s_{ij} \in \mathbb{R}^4$ vérifie la contrainte (**??**). Ceci conduit au problème d'optimisation usuel d'un GAN avec contraintes sur la sortie:

Etant donné un super-pixel $s_{i,j}$, que nous écrivons sous la forme $s_{i,j} = (v_{i,j}, t_{i,j})$ pour nous conformer à la notation du cône de second-ordre, l'opérateur de projection sur $\mathscr{C}$ peut être déterminé comme le problème proximal suivant :

$$\min_{(u,r)\in\mathscr{C}} \frac{1}{2}\|(v,t)-(u,r)\|_2^2$$

dont la solution est donnée par (Parikh & Boyd, 2014)

$$\Pi_{\mathscr{C}}(v,t) = \begin{cases} 0 & \text{si} \quad \|v\|_2 \leq -t \\ (v,t) & \text{si} \quad \|v\|_2 \leq t \\ \frac{1+t/\|v\|_2}{2}(v, \|v\|_2) & \text{si} \quad \|v\|_2 \geq t \end{cases} \tag{3.8}$$

Partant de cela, on peut proposer une formulation alternative au problème (3.5) comme suit :

$$\begin{aligned} \min_{G}\max_{D} \quad & \mathbb{E}_{x\sim P_r}\left[\log(D(x))\right] + \mathbb{E}_{z\sim P_z}\left[\log(1-D(G(z)))\right] + \lambda \sum_i \sum_j \Omega_{\mathscr{C}}(s_{i,j}) \\ \text{s.c.} \quad & s_{i,j} = M(G(z)_{i,j}) \quad \forall i,j \end{aligned} \tag{3.9}$$

avec

$$\Omega_{\mathscr{C}}(s_{i,j}) = \| s_{i,j} - \Pi_{\mathscr{C}}(s_{i,j}) \|_2^2$$

et $\lambda > 0$ un paramètre de régularisation. Une telle approche de prise en compte des contraintes a été utilisée par exemple dans (Kervadec et al., 2019) comme une alternative à une version lagrangienne d'un problème de segmentation d'image sous contraintes de volume sur certaines régions de l'image, lesdites contraintes étant exprimées à partir d'un réseau CNN Pathak et al., 2015.

Notons θ_G les paramètres du générateur G. Avec la notation $s = (v, t)$, le gradient de $\Omega_{\mathscr{C}}$ par rapport à θ_G s'écrit

$$\frac{\partial \Omega_{\mathscr{C}}}{\partial \theta_G}(s) = (s - \Pi_{\mathscr{C}}(s)) \times \begin{cases} \frac{\partial s}{\partial \theta_G} & \text{si} \quad \|v\|_2 \leq -t \\ 0 & \text{si} \quad \|v\|_2 \leq t \\ \frac{\partial s}{\partial \theta_G} - \frac{\partial \frac{1+t/\|v\|_2}{2}(v, \|v\|_2)}{\partial \theta_G} & \text{si} \quad \|v\|_2 \geq t \end{cases} \tag{3.10}$$

3.5 Experimental evaluation

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

3.5.1 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 (3.6) provides the best performances. As for the original CycleGAN, the hyper-parameter λ, controlling the reconstruction cost, was set to $\lambda = 10$.

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.

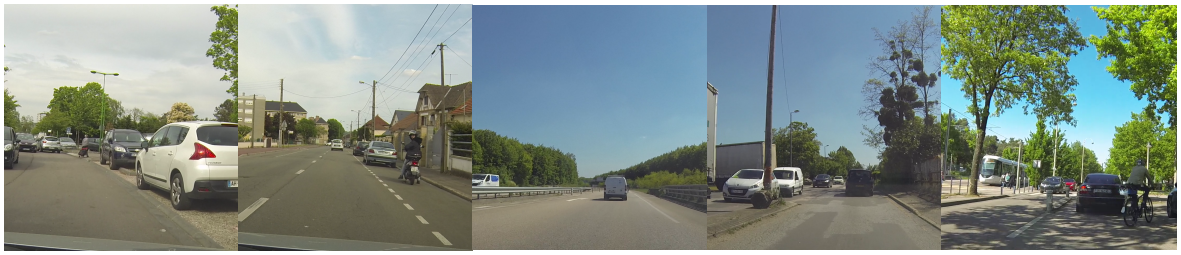


Figure 3.4: Examples of images in the RGB dataset.

The learning rate is decreased linearly 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 test-bed 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.

3.5.2 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)

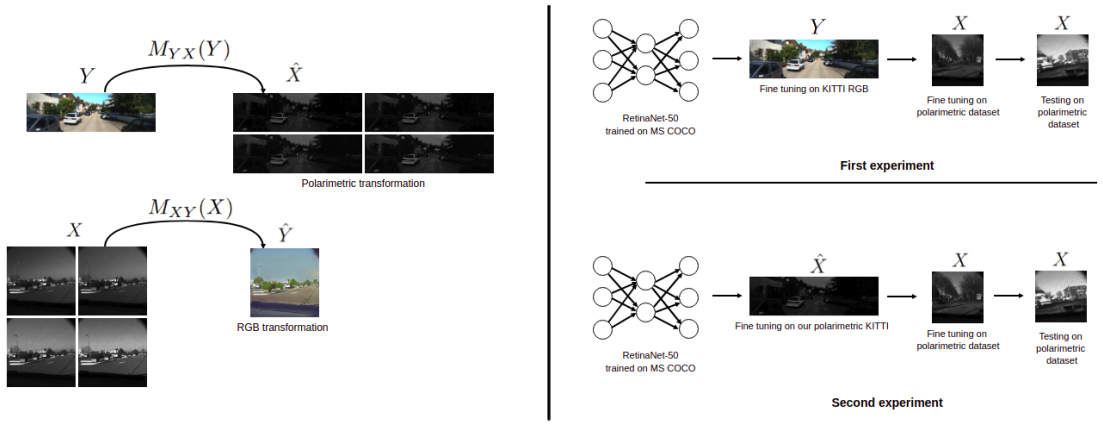


Figure 3.5: Setup of the detection evaluation experiment. The procedure is illustrated with the KITTI dataset and straightforwardly extends to the BDD100K dataset.

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 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:

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

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.

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.

3.5.3 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, which refers to $M_{XY} \circ M_{YX}$ 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 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 3.5.2 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%

[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 $||I - AS|| / (||I|| + ||AS||)$.

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.

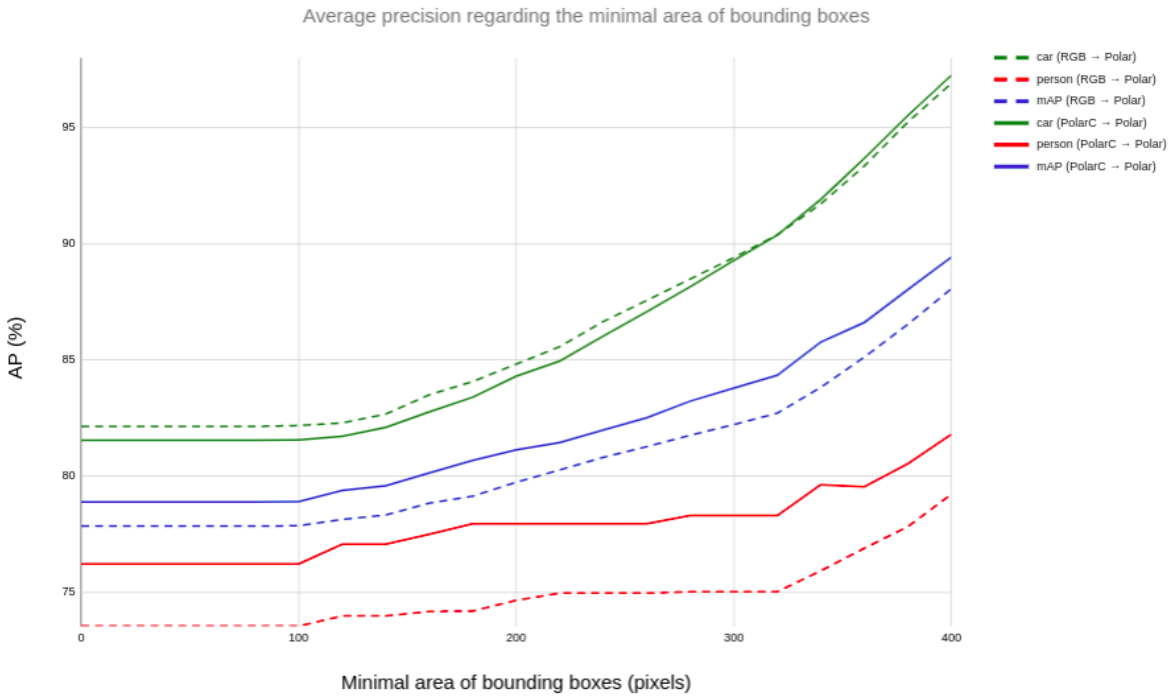


Figure 3.7: Evolution of the average precision when setting a minimal area of the bounding boxes to be detected. Here green lines refer to the evolution of cars' detection, blue lines to the evolution of the *m*AP and red lines to the evolution of person's detection. The dashed lines refer to the training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

(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 including 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.

3.6 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.

CR: Future works : using adapted metrics for the non-euclidean outspace

Chapter 4

Conclusion and Perspectives

Bibliography

Aharon, M., M. Elad, and A. Bruckstein (Nov. 2006). "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". In: *IEEE Transactions on Signal Processing* 54.11, pp. 4311–4322. ISSN: 1053-587X. DOI: 10.1109/TSP.2006.881199. URL: http://ieeexplore.ieee.org/document/1710377/ (visited on 10/26/2020) (cit. on p. 27).

Ainouz, Samia, Olivier Morel, David Fofi, Saleh Mosaddegh, and Abdelaziz Bensrhair (2013). "Adaptive Processing of Catadioptric Images Using Polarization Imaging: Towards a Pola-Catadioptric Model". In: *Optical engineering* 52.3, p. 037001 (cit. on p. 50).

Almahairi, Amjad, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville (2018). *Augmented Cyclegan: Learning Many-to-Many Mappings from Unpaired Data*. arXiv: 1802.10151 (cit. on p. 49).

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

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. 10).

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein GAN". In: URL: https://arxiv.org/pdf/1701.07875.pdf (cit. on pp. 10, 14, 16).

Armanious, Karim, Youssef Mecky, Sergios Gatidis, and Bin Yang (May 2019). "Adversarial Inpainting of Medical Image Modalities". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3267–3271. DOI: 10.1109/ICASSP.2019.8682677. arXiv: 1810.06621. URL: http://arxiv.org/abs/1810.06621 (visited on 09/30/2020) (cit. on p. 29).

Aycock, Todd M, David B Chenault, Jonathan B Hanks, and John S Harchanko (Mar. 7, 2017). "Polarization-Based Mapping and Perception Method and System". In: (cit. on p. 49).

Barratt, Shane and Rishi Sharma (2018). *A Note on the Inception Score*. URL: https://github.com/ (cit. on pp. 18, 39).

Bass, Michael, Eric W Van Stryland, David R Williams, and William L Wolfe (1995). *Handbook of Optics*. Vol. 2. McGraw-Hill New York (cit. on p. 50).

Bau, David, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba (July 12, 2019). "Semantic Photo Manipulation with a Generative Image Prior". In: *ACM Transactions on Graphics* 38.4, pp. 1–11. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3306346.3323023. URL: https://dl.acm.org/doi/10.1145/3306346.3323023 (visited on 10/21/2020) (cit. on p. 24).

Bellemare, Marc G., Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos (May 30, 2017). *The Cramer Distance as a Solution to Biased Wasserstein Gradients*. arXiv: 1705.10743 [cs, stat]. URL: http://arxiv.org/abs/1705.10743 (visited on 05/21/2020) (cit. on pp. 15, 16).

Bengio, Yoshua, Éric Thibodeau-Laufer, Guillaume Alain, and Jason Yosinski (May 23, 2014). *Deep Generative Stochastic Networks Trainable by Backprop*. arXiv: 1306.1091 [cs]. URL: http://arxiv.org/abs/1306.1091 (visited on 05/22/2020) (cit. on p. 16).

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. 49).

Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester (July 1, 2000). "Image Inpainting". In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., pp. 417–424. ISBN: 978-1-58113-208-3. DOI: 10.1145/344779.344972. URL: https://doi.org/10.1145/344779.344972 (visited on 09/15/2020) (cit. on p. 24).

Bińkowski, Miko\ laj, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton (Jan. 2018). "Demystifying MMD GANs". In: URL: http://arxiv.org/abs/1801.01401 (cit. on pp. 15, 18).

Blin, Rachel, Samia Ainouz, Stephane Canu, and Fabrice Meriaudeau (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. 55, 56).

Blin, Rachel, Samia Ainouz, Stéphane Canu, and Fabrice Meriaudeau (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. 49, 57).

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. 30, 32).

Borji, Ali (2018). "Pros and Cons of GAN Evaluation Measures". In: URL: https://arxiv.org/pdf/1802.03446.pdf (cit. on p. 18).

Boyd, Stephen, Stephen P. Boyd, and Lieven Vandenberghe (Mar. 8, 2004). *Convex Optimization*. Cambridge University Press. 744 pp. ISBN: 978-0-521-83378-3. Google Books: mYm0bLd3fcoC (cit. on p. 54).

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. 10, 11).

Brown, Lawrence D (1986). "Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory". In: Ims (cit. on p. 35).

Burt, Peter J and Edward H Adelson (1983). "The Laplacian Pyramid as a Compact Image Code". In: p. 9 (cit. on p. 16).

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 pp. 23, 26, 32).

Candès, Emmanuel J. (May 1, 2008). "The Restricted Isometry Property and Its Implications for Compressed Sensing". In: *Comptes Rendus Mathematique* 346.9, pp. 589–592. ISSN: 1631-073X. DOI: 10.1016/j.crma.2008.03.014. URL: http://www.sciencedirect.com/science/article/pii/S1631073X08000964 (visited on 09/22/2020) (cit. on p. 26).

Cimpoi, Mircea, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and and Andrea Vedaldi (2014). "Describing Textures in the Wild". In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 39).

Criminisi, A., P. Perez, and K. Toyama (Sept. 2004). "Region Filling and Object Removal by Exemplar-Based Image Inpainting". In: *IEEE Transactions on Image Processing* 13.9, pp. 1200–1212. ISSN: 1941-0042. DOI: 10.1109/TIP.2004.833105 (cit. on p. 24).

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. 39).

Danihelka, Ivo, Balaji Lakshminarayanan, Benigno Uria, Daan Wierstra, and Peter Dayan (May 15, 2017). *Comparison of Maximum Likelihood and GAN-Based Training of Real NVPs*. arXiv: 1705.05263 [cs]. URL: http://arxiv.org/abs/1705.05263 (visited on 05/23/2020) (cit. on p. 9).

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. 29).

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, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: Ieee. URL: http://www.image-net.org. (cit. on pp. 18, 57).

Denton, Emily, Soumith Chintala, Arthur Szlam, and Rob Fergus (June 18, 2015). *Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks*. arXiv: 1506.05751 [cs]. URL: http://arxiv.org/abs/1506.05751 (visited on 05/22/2020) (cit. on p. 16).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (Feb. 27, 2017). *Density Estimation Using Real NVP*. arXiv: 1605.08803 [cs, stat]. URL: http://arxiv.org/abs/1605.08803 (visited on 05/11/2020) (cit. on pp. 4, 7).

Donoho, D.L. (Apr. 2006a). "Compressed Sensing". In: *IEEE Transactions on Information Theory* 52.4, pp. 1289–1306. ISSN: 0018-9448. DOI: 10.1109/TIT.2006.871582. URL: http://ieeexplore.ieee.org/document/1614066/ (visited on 09/21/2020) (cit. on pp. 27, 32).

Donoho, David L. (2006b). "For Most Large Underdetermined Systems of Linear Equations the Minimal \ell_1-Norm Solution Is Also the Sparsest Solution". In: *Commu-*

nications on Pure and Applied Mathematics 59.6, pp. 797–829. ISSN: 1097-0312. DOI: 10.1002/cpa.20132. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20132 (visited on 10/26/2020) (cit. on p. 26).

Duarte, Marco F., Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk (Mar. 2008). "Single-Pixel Imaging via Compressive Sampling". In: *IEEE Signal Processing Magazine* 25.2, pp. 83–91. ISSN: 1053-5888. DOI: 10.1109/MSP.2007.914730. URL: http://ieeexplore.ieee.org/document/4472247/ (visited on 09/21/2020) (cit. on p. 27).

Dziugaite, Gintare Karolina, Daniel M. Roy, and Zoubin Ghahramani (May 14, 2015). *Training Generative Neural Networks via Maximum Mean Discrepancy Optimization*. arXiv: 1505.03906 [cs, stat]. URL: http://arxiv.org/abs/1505.03906 (visited on 05/25/2020) (cit. on p. 15).

Engan, K., S. O. Aase, and J. Hakon Husoy (Mar. 1999). "Method of Optimal Directions for Frame Design". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258). Vol. 5, 2443–2446 vol.5. DOI: 10.1109/ICASSP.1999.760624 (cit. on p. 27).

Fan, Wang, Samia Ainouz, Fabrice Meriaudeau, and Abdelaziz Bensrhair (2018). "Polarization-Based Car Detection". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3069–3073 (cit. on p. 49).

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. 49).

Gondara, L. (Dec. 2016). "Medical Image Denoising Using Convolutional Denoising Autoencoders". In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pp. 241–246. DOI: 10.1109/ICDMW.2016.0041 (cit. on p. 24).

Goodfellow, Ian (2016). "NIPS 2016 Tutorial: Generative Adversarial Networks". In: URL: https://arxiv.org/pdf/1701.00160.pdf (cit. on p. 11).

Goodfellow, Ian J, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets". In: URL: https://arxiv.org/pdf/1406.2661.pdf (cit. on pp. 1, 3, 4, 7, 10, 12, 16, 34, 48).

Goyal, Bhawna, Ayush Dogra, Sunil Agrawal, B. S. Sohi, and Apoorav Sharma (Mar. 1, 2020). "Image Denoising Review: From Classical to State-of-the-Art Approaches". In: *Information Fusion* 55, pp. 220–244. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.09.003. URL: http://www.sciencedirect.com/science/article/pii/S1566253519301861 (visited on 09/21/2020) (cit. on p. 24).

Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola (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. 15).

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville (2017). "Improved Training of Wasserstein GANs". In: URL: https://arxiv.org/pdf/1704.00028.pdf (cit. on p. 15).

Guo, Zongyu, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu (Sept. 18, 2019). *Progressive Image Inpainting with Full-Resolution Residual Network*. arXiv: 1907.10478 [eess]. URL: http://arxiv.org/abs/1907.10478 (visited on 09/30/2020) (cit. on p. 29).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (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. 38).

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (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. 10, 18, 39, 57).

Hindupur, Avinash (2017). *The GAN Zoo*. URL: https://github.com/hindupuravinash/the-gan-zoo (visited on 05/21/2020) (cit. on p. 11).

Hu, Zhiting, Zichao Yang, Ruslan R. Salakhutdinov, LIANHUI Qin, Xiaodan Liang, Haoye Dong, and Eric P. Xing (2018). *Deep Generative Models with Learnable Knowledge Constraints*. 10522–10533. URL: http://papers.nips.cc/paper/8250-deep-generative-models-with-learnable-knowledge-constraints.

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 pp. 16, 80).

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2016). "Image-to-Image Translation with Conditional Adversarial Networks". In: URL: https://arxiv.org/pdf/1611.07004v1.pdf (cit. on pp. 12, 22, 38, 41).

Jetchev, Nikolay, Urs Bergmann, Roland Vollgraf, and Zalando Research (2017). "Texture Synthesis with Spatial Generative Adversarial Networks". In: URL: https://arxiv.org/pdf/1611.08207.pdf (cit. on pp. 23, 37, 41).

Julien, Rabin, Gabriel Peyré, Julie Delon, Bernot Marc, Marc Wasserstein Barycenter, Julien Rabin, and Marc Bernot (2011). *Wasserstein Barycenter and Its Application to Texture Mixing*, pp. 435–446. URL: https://hal.archives-ouvertes.fr/hal-00476064 (cit. on p. 18).

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. 14).

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen (2017). *Progressive Growing of GANs for Improved Quality, Stability and Variation*. URL: https://youtu.be/G06dEcZ-QTg. (cit. on p. 17).

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila (Mar. 23, 2020). *Analyzing and Improving the Image Quality of StyleGAN*. arXiv: 1912.

04958 [cs, eess, stat]. URL: http://arxiv.org/abs/1912.04958 (visited on 05/21/2020) (cit. on pp. 11, 22).

Kervadec, Hoel, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed (May 1, 2019). "Constrained-CNN Losses for Weakly Supervised Segmentation". In: *Medical Image Analysis* 54, pp. 88–99. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.02.009. URL: http://www.sciencedirect.com/science/article/pii/S1361841518306145 (visited on 10/30/2020) (cit. on p. 55).

Kim, Sung-Un (2014). "An Image Denoising Algorithm for the Mobile Phone Cameras". In: *The Journal of the Korea institute of electronic communication sciences* 9.5, pp. 601–608. ISSN: 1975-8170. DOI: 10.13067/JKIECS.201.9.5.601. URL: https://www.koreascience.or.kr/article/JAKO201415642602071.page (visited on 10/21/2020) (cit. on p. 24).

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". In: URL: https://arxiv.org/pdf/1312.6114.pdf (cit. on pp. 4, 6, 16).

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, 7).

Kniaz, Vladimir V., Vladimir A. Knyaz, Jiri Hladuvka, Walter G. Kropatsch, and Vladimir Mizginov (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. 48).

Kolev, Vasil (2011). "Compressed Sensing of Astronomical Images: Orthogonal Wavelets Domains". In: p. 8 (cit. on p. 27).

Krizhevsky, Alex (2009). "Learning Multiple Layers of Features from Tiny Images". In: p. 60 (cit. on pp. 16, 23, 37).

Laloy, Eric, Romain Hérault, Diederik Jacques, and Niklas Linde (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. 23).

Laloy, Eric, Niklas Linde, Cyprien Ruffino, Romain Hérault, Gilles Gasso, and Diederik Jacques (Dec. 2019). *Gradient-Based Deterministic Inversion of Geophysical Data with Generative Adversarial Networks: Is It Feasible?* Vol. 133. Elsevier Ltd. 104333 pp. DOI: 10.1016/j.cageo.2019.104333 (cit. on pp. 21, 23, 33).

LeCun, Yann, Leon Bottou, Yoshua Bengio, and Patrick Haffner (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. 16, 23, 37).

Lemmens, L., B. Rogiers, M. Craen, E. Laloy, D. Jacques, and et al Huysmans D (2017). *Effective Structural Descriptors for Natural and Engineered Radioactive Waste Confinement Barrier*. Vienna (cit. on pp. 39, 84).

Li, Chun-Liang, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos (Nov. 27, 2017). *MMD GAN: Towards Deeper Understanding of Moment Matching Network*. arXiv: 1705.08584 [cs, stat]. URL: http://arxiv.org/abs/1705.08584 (visited on 05/19/2020) (cit. on pp. 10, 16).

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. 14).

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017). "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988 (cit. on p. 56).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft Coco: Common Objects in Context". In: *European Conference on Computer Vision*. Springer, pp. 740–755 (cit. on p. 56).

Lin, Zinan, Ashish Khetan, Giulia Fanti, and Sewoong Oh (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. 21, 23, 24, 35, 36).

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)* (cit. on pp. 23, 37).

Lustig, M., D. L. Donoho, J. M. Santos, and J. M. Pauly (Mar. 2008). "Compressed Sensing MRI". In: *IEEE Signal Processing Magazine* 25.2, pp. 72–82. ISSN: 1558-0792. DOI: 10.1109/MSP.2007.914728 (cit. on p. 27).

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: p. 6 (cit. on p. 16).

Mallat, Stéphane (2008). *A Wavelet Tour of Signal Processing - 3rd Edition*. URL: https://www.elsevier.com/books/a-wavelet-tour-of-signal-processing/mallat/978-0-12-374370-1 (visited on 10/26/2020) (cit. on p. 27).

Mao, Xudong, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley (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. 14, 16).

Marafioti, Andrés, Nathanaël Perraudin, Nicki Holighaus, and Piotr Majdak (2018). *A Context Encoder for Audio Inpainting*. arXiv: 1810.12138 (cit. on p. 45).

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. 48).

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. 10).

Mirza, Mehdi and Simon Osindero (2014). "Conditional Generative Adversarial Nets". In: URL: https://arxiv.org/pdf/1411.1784.pdf (cit. on pp. 12, 21, 22, 32, 41).

Morel, Olivier, Christophe Stolz, Fabrice Meriaudeau, and Patrick Gorria (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. 49).

Mroueh, Youssef and Tom Sercu (Nov. 3, 2017). *Fisher GAN*. arXiv: 1705.09675 [cs, stat]. URL: http://arxiv.org/abs/1705.09675 (visited on 05/21/2020) (cit. on pp. 15, 16).

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. 15).

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: p. 8 (cit. on p. 16).

Nie, Dong, Roger Trullo, Jun Lian, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen (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. 48).

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. 10, 14, 16).

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. 12).

Oliveira, Manuel M, Brian Bowen, Richard McKenna, and Yu-Sung Chang (2001). "Fast Digital Image Inpainting". In: p. 7 (cit. on p. 24).

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. 30, 32).

Parikh, Neal and Stephen Boyd (Jan. 13, 2014). "Proximal Algorithms". In: *Foundations and Trends in Optimization* 1.3, pp. 127–239. ISSN: 2167-3888. DOI: 10.1561/2400000003. URL: https://doi.org/10.1561/2400000003 (visited on 10/30/2020) (cit. on p. 54).

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. 18).

Pathak, Deepak, Philipp Krahenbuhl, and Trevor Darrell (Dec. 2015). "Constrained Convolutional Neural Networks for Weakly Supervised Segmentation". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile: IEEE, pp. 1796–1804. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.209. URL: http://ieeexplore.ieee.org/document/7410566/ (visited on 10/30/2020) (cit. on p. 55).

Pathak, Deepak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros (2016). "Context Encoders: Feature Learning by Inpainting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544 (cit. on pp. 22, 29).

Peyré, Gabriel and Marco Cuturi (Mar. 18, 2020). *Computational Optimal Transport*. arXiv: 1803.00567 [stat]. URL: http://arxiv.org/abs/1803.00567 (visited on 05/23/2020) (cit. on p. 14).

Pietikäinen, Matti, Abdenour Hadid, Guoying Zhao, and Timo Ahonen (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. 39).

Radford, Alec, Luke Metz, and Soumith Chintala (Nov. 2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv: 1511.06434. URL: http://arxiv.org/abs/1511.06434 (cit. on pp. 10, 16, 38).

Rauhut, Holger (2010). "Compressive Sensing and Structured Random Matrices". In: p. 94 (cit. on p. 26).

Rehbinder, Jean, Huda Haddad, Stanislas Deby, Benjamin Teig, André Nazac, Tatiana Novikova, Angelo Pierangelo, and François Moreau (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. 49).

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. 38).

Rudelson, Mark and Roman Vershynin (2008). "On Sparse Reconstruction from Fourier and Gaussian Measurements". In: *Communications on Pure and Applied Mathematics* 61.8, pp. 1025–1045. ISSN: 1097-0312. DOI: 10.1002/cpa.20227. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20227 (visited on 09/22/2020) (cit. on p. 26).

Ruffino, Cyprien, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: URL: http://arxiv.org/abs/1905.08613 (cit. on pp. 21, 38, 39, 43).

Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen (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. 10, 16–18, 39).

Sallab, Ahmad El, Ibrahim Sobh, Mohamed Zahran, and Nader Essam (2019). *LiDAR Sensor Modeling and Data Augmentation with GANs for Autonomous Driving*. arXiv: 1905.07290 (cit. on p. 48).

Shaobing, Chen and D. Donoho (Oct. 1994). "Basis Pursuit". In: *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*. Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers. Vol. 1, 41–44 vol.1. DOI: 10.1109/ACSSC.1994.471413 (cit. on p. 27).

Sønderby, Casper Kaae, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár (Feb. 21, 2017). *Amortised MAP Inference for Image Super-Resolution*. arXiv: 1610.04490 [cs, stat]. URL: http://arxiv.org/abs/1610.04490 (visited on 05/19/2020) (cit. on pp. 10, 16).

Springenberg, Jost Tobias, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller (Apr. 13, 2015). *Striving for Simplicity: The All Convolutional Net*. arXiv: 1412.6806 [cs]. URL: http://arxiv.org/abs/1412.6806 (visited on 05/22/2020) (cit. on p. 16).

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet (Oct. 12, 2009). *On Integral Probability Metrics, \phi-Divergences and Binary Classification*. arXiv: 0901.2698 [cs, math]. URL: http://arxiv.org/abs/0901.2698 (visited on 05/22/2020) (cit. on p. 15).

Srivastava, Nitish, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (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. 16).

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. 23, 38).

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna (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. 18, 57).

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

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. 39).

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. ISSN: 0035-9246. JSTOR: 2346178 (cit. on p. 27).

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). *Instance Normalization: The Missing Ingredient for Fast Stylization*. arXiv: 1607.08022 (cit. on p. 81).

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*. arXiv: 1609.02612 [cs]. URL: http://arxiv.org/abs/1609.02612 (visited on 05/19/2020) (cit. on p. 1).

Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro (Dec. 3, 2018a). *Video-to-Video Synthesis*. arXiv: 1808.06601 [cs]. URL: http://arxiv.org/abs/1808.06601 (visited on 05/21/2020) (cit. on p. 11).

Wang, Yi, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia (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. 22).

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 pp. 1, 22).

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 p. 50).

Wolff, Lawrence B and Andreas G Andreou (1995). "Polarization Camera Sensors". In: *Image and Vision Computing* 13.6, pp. 497–510 (cit. on p. 49).

Wu, Jiajun, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum (Jan. 4, 2017). *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*. 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 pp. 27, 28, 32).

Xiang, Peng, Lei Wang, Jun Cheng, Bin Zhang, and Jiaji Wu (Dec. 2017). "A Deep Network Architecture for Image Inpainting". In: *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1851–1856. DOI: 10.1109/CompComm.2017.8322859 (cit. on p. 29).

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. 23, 37–39).

Xu, Huazhe, Yang Gao, Fisher Yu, and Trevor Darrell (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. 49).

Yang, Dingdong, Seunghoon Hong, Yunseok Jang, Tiangchen Zhao, and Honglak Lee (2019). "Diversity-Sensitive Conditional Generative Adversarial Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJliMh09F7 (cit. on pp. 23, 35).

Yeh, Raymond A, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do (2017). "Semantic Image Inpainting with Deep Generative Models". In: URL: https://arxiv.org/pdf/1607.07539.pdf (cit. on pp. 31, 32).

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. 38).

Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang (2018). "Generative Image Inpainting With Contextual Attention". In: p. 10 (cit. on p. 29).

Zhang, Lichao, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan (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. 48).

Zhao, Junbo, Michael Mathieu, and Yann LeCun (Mar. 6, 2017). *Energy-Based Generative Adversarial Network*. arXiv: 1609.03126 [cs, stat]. URL: http://arxiv.org/abs/1609.03126 (visited on 05/21/2020) (cit. on pp. 14, 16).

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. 49).

Zhu, Jun-Yan, Taesung Park, Phillip Isola, Alexei A Efros, and Berkeley Ai Research (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. 48, 49).

Zhu, Xinyue, Yifan Liu, Zengchang Qin, and Jiahong Li (2017b). *Emotion Classification with Data Augmentation Using Generative Adversarial Networks*. URL: https://arxiv.org/pdf/1711.00648.pdf (cit. on pp. 13, 38, 41).

Appendix A

Publications

References

Laloy, Eric, Niklas Linde, Cyprien Ruffino, Romain Hérault, Gilles Gasso, and Diederik Jacques (Dec. 2019). *Gradient-Based Deterministic Inversion of Geophysical Data with Generative Adversarial Networks: Is It Feasible?* Vol. 133. Elsevier Ltd. 104333 pp. DOI: 10.1016/j.cageo.2019.104333 (cit. on p. 77).

Ruffino, Cyprien, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: URL: http://arxiv.org/abs/1905.08613 (cit. on p. 77).

— (Nov. 2019). "Pixel-Wise Conditioning of Generative Adversarial Networks". In: *ESANN 2019 - Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 25–30. URL: http://arxiv.org/abs/1911.00689 (cit. on p. 77).

— (Apr. 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing*. ISSN: 09252312. DOI: 10.1016/j.neucom.2019.11.116. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231220305154 (cit. on p. 77).

Appendix B

Experiment details for the Pixel-Wise Conditionned GAN

B.1 Details of the datasets

Dataset	Size (in pixels)	Training set	Validation set	Test set
FashionMNIST	28x28	55,000	5,000	10,000
Cifar-10	32x32	55,000	5,000	10,000
CelebA	128x128	80,000	5,000	15,000
Texture	160x160	20,000	2,000	4,000
Subsurface	160x160	20,000	2,000	4,000

Additional information:

- For FashionMNIST and Cifar-10, we keep the original train/test split and then sample 5000 images from the training set that act as validation samples.

- For the Texture dataset, we sample patches randomly from a 3840x2400 image of a brick wall.

B.2 Detailed deep architectures

B.2.1 DCGAN for FashionMNIST

Layer type	Units	Scaling	Activation	Output shape
Input z	-	-	-	7x7
Input y	-	-	-	28x28
Dense	343	-	ReLU	7x7
Conv2DTranspose	128 3x3	x2	ReLU	14x14
Conv2DTranspose	64 3x3	x2	ReLU	28x28
Conv2DTranspose	1 3x3	x1	tanh	28x28
Input x	-	-	-	28x28
Input y	-	-	-	28x28
Conv2D	64 3x3	x1/2	LeakyReLU	14x14
Conv2D	128 3x3	x1/2	LeakyReLU	7x7
Conv2D	1 3x3	x1	tanh	28x28
Dense	1	-	Sigmoid	1

Additional information:

- Batch normalization (Ioffe et al., 2015) is applied across all the layers
- A Gaussian noise is applied to the input of the discriminator

B.2.2 UNet-Res for CIFAR10

Layer type	Units	Scaling	Activation	Output shape
Input y	-	-	-	32x32
Conv2D*	64 5x5	x1	ReLU	32x32
Conv2D*	128 3x3	x1/2	ReLU	16x16
Conv2D*	256 3x3	x1/2	ReLU	8x8
Input z	-	-	-	8x8
Dense	256	-	ReLU	8x8
Residual block	3x256 3x3	x1	ReLU	8x8
Residual block	3x256 3x3	x1	ReLU	8x8
Residual block	3x256 3x3	x1	ReLU	8x8
Residual block	3x256 3x3	x1	ReLU	8x8
Conv2DTranspose*	256 3x3	x2	ReLU	16x16
Conv2DTranspose*	128 3x3	x2	ReLU	32x32
Conv2DTranspose*	64 3x3	x1	ReLU	32x32
Conv2D	3 3x3	x1	tanh	32x32
Input x	-	-	-	32x32
Input y	-	-	-	32x32
Conv2D	64 3x3	x1/2	LeakyReLU	16x16
Conv2D	128 3x3	x1/2	LeakyReLU	8x8
Conv2D	256 3x3	x1/2	LeakyReLU	4x4
Dense	1	-	Sigmoid	1

Additional information:

- Instance normalization (Ulyanov et al., 2016) is applied across all the layers instead of Batch normalization. This is involved by the use of the PacGAN technique.

- A Gaussian noise is applied to the input of the discriminator

- The layers noted with an asterisk are linked with a skip-connection

B.2.3 UNet-Res for CelebA

Layer type	Units	Scaling	Activation	Output shape
Input y	-	-	-	128x128
Conv2D	64 5x5	x1	ReLU	128x128
Conv2D*	128 3x3	x1/2	ReLU	64x64
Conv2D*	256 3x3	x1/2	ReLU	32x32
Conv2D*	512 3x3	x1/2	ReLU	16x16
Input z	-	-	-	16x16
Dense	256	-	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Residual block	3x256 3x3	x1	ReLU	16x16
Conv2DTranspose*	256 3x3	x2	ReLU	32x32
Conv2DTranspose*	128 3x3	x2	ReLU	64x64
Conv2DTranspose*	64 5x5	x2	ReLU	128x128
Conv2D	3 3x3	x1	tanh	128x128
Input x	-	-	-	128x128
Input y	-	-	-	128x128
Conv2D	64 3x3	x1/2	LeakyReLU	64x64
Conv2D	128 3x3	x1/2	LeakyReLU	32x32
Conv2D	256 3x3	x1/2	LeakyReLU	16x16
Conv2D	512 3x3	x1/2	LeakyReLU	32x32
Dense	1	-	Sigmoid	1

This network follows the same additional setup as described in Appendix (B.2.2).

B.2.4 Architectures for Texture

PatchGAN discriminator

Layer type	Units	Scaling	Activation	Output shape
Input x	-	-	-	160x160
Input y	-	-	-	160x160
Conv2D	64 3x3	x1/2	LeakyReLU	80x80
Conv2D	128 3x3	x1/2	LeakyReLU	40x40
Conv2D	256 3x3	x1/2	LeakyReLU	20x20
Conv2D	512 3x3	x1/2	LeakyReLU	10x10

UpDil Texture

Layer type	Units	Scaling	Activation	Output shape
Input z	-	-	-	20x20
Conv2DTranspose	256 3x3	x2	ReLU	40x40
Conv2DTranspose	128 3x3	x2	ReLU	80x80
Conv2DTranspose	64 3x3	x2	ReLU	160x160
Input y	-	-	-	160x160
Conv2D	64 3x3 dil. 1	x1	ReLU	160x160
Conv2D	128 3x3 dil. 2	x1	ReLU	160x160
Conv2D	256 3x3 dil. 3	x1	ReLU	160x160
Conv2D	512 3x3 dil. 4	x1	ReLU	160x160
Conv2D	3 3x3	x1	tanh	160x160

UpEncDec Texture

Layer type	Units	Scaling	Activation	Output shape
Input z	-	-	-	20x20
Conv2DTranspose	256 3x3	x2	ReLU	40x40
Conv2DTranspose	128 3x3	x2	ReLU	80x80
Conv2DTranspose	64 5x5	x2	ReLU	160x160
Input* y	-	-	-	160x160
Conv2D*	64 3x3	x1/2	ReLU	80x80
Conv2D*	128 3x3	x1/2	ReLU	40x40
Conv2D	256 3x3	x1/2	ReLU	20x20
Conv2DTranspose*	256 3x3	x2	ReLU	40x40
Conv2DTranspose*	128 3x3	x2	ReLU	80x80
Conv2DTranspose*	64 3x3	x2	ReLU	160x160
Conv2D	3 3x3	x1	tanh	160x160

UNet Texture

Layer type	Units	Scaling	Activation	Output shape
Input y	-	-	-	160x160
Conv2D	64 5x5	x1	ReLU	160x160
Conv2D*	128 3x3	x1/2	ReLU	80x80
Conv2D*	256 3x3	x1/2	ReLU	40x40
Conv2D*	512 3x3	x1/2	ReLU	20x20
Input z	-	-	-	20x20
Conv2DTranspose*	256 3x3	x2	ReLU	40x40
Conv2DTranspose*	128 3x3	x2	ReLU	80x80
Conv2DTranspose*	64 5x5	x2	ReLU	160x160
Conv2D	3 3x3	x1	tanh	160x160

Res Texture

Layer type	Units	Scaling	Activation	Output shape
Input y	-	-	-	160x160
Conv2D	64 5x5	x1	ReLU	160x160
Conv2D	128 3x3	x1/2	ReLU	80x80
Conv2D	256 3x3	x1/2	ReLU	40x40
Conv2D	512 3x3	x1/2	ReLU	20x20
Input z	-	-	-	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Conv2DTranspose	256 3x3	x2	ReLU	40x40
Conv2DTranspose	128 3x3	x2	ReLU	80x80
Conv2DTranspose	64 5x5	x2	ReLU	160x160
Conv2D	3 3x3	x1	tanh	160x160

UNet-Res Texture

Layer type	Units	Scaling	Activation	Output shape
Input y	-	-	-	160x160
Conv2D	64 5x5	x1	ReLU	160x160
Conv2D*	128 3x3	x1/2	ReLU	80x80
Conv2D*	256 3x3	x1/2	ReLU	40x40
Conv2D*	512 3x3	x1/2	ReLU	20x20
Input z	-	-	-	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Residual block	3x256 3x3	x1	ReLU	20x20
Conv2DTranspose*	256 3x3	x2	ReLU	40x40
Conv2DTranspose*	128 3x3	x2	ReLU	80x80
Conv2DTranspose*	64 5x5	x2	ReLU	160x160
Conv2D	3 3x3	x1	tanh	160x160

As for Cifar10, this network follows the same additional setup described in Appendix (B.2.2).

B.3 Domain-specific metrics for underground soil generation

In this section, we compute the connectivity function Lemmens et al., 2017 of generated soil image, a domain-specific metric, which is the probability that a continuous pixel path exists between two pixels of the same value (called Facies) in a given direction and a given distance (called Lag). This connectivity function should be similar to the one obtained on real-world samples. In this application, the connectivity function models the probability that two given pixels are from the same sand brick or clay matrix zone.

We sampled 100 real and 100 generated images using the UNetResPAC architecture (see Section 2.4.2) on which the connectivity function was evaluated for both the CGAN and our approach. The obtained graphs are shown respectively in Figures B.1 and B.2.

The blue curves are the mean value for the real samples, and the blue dashed curves are the minimum and maximum values on these samples. The green curves are the connectivity functions for each of the 100 synthetic samples and the red curves are their mean connectivity functions. From these curves we observe that that our approach has similar connectivity functions as the CGAN approach while being significantly better at respecting the given constraints (see Section Table 2.5).

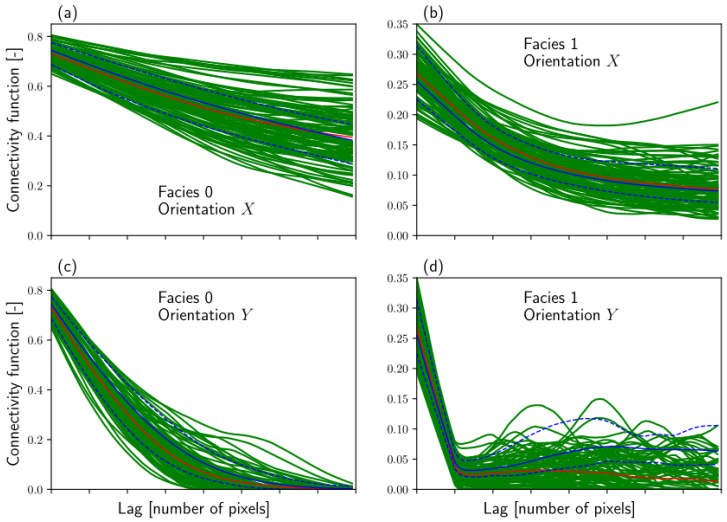


Figure B.1: Connectivity curves obtained on 100 samples generated with the CGAN approach.

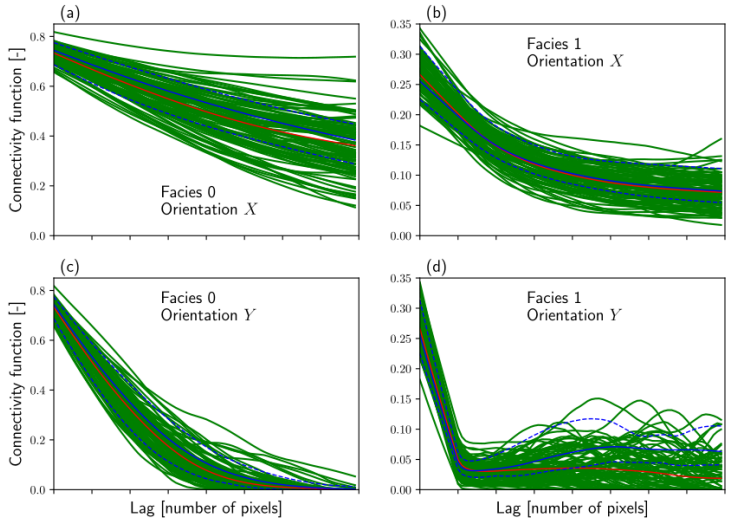


Figure B.2: Connectivity curves obtained on 100 samples generated with our approach.

B.4 Additional samples from the Texture and Subsurface datasets

In this section, we show some samples generated with the UNetResPAC architecture, which performs the best in our experiments (see Sections 2.4.2 and 2.4.2) compared to real images sampled from the Texture (Figure B.3) and Subsurface (Figure B.4) datasets. For the generated samples, the enforced pixel constraints are colored in the images, green corresponding to a squared error less than 0.1 and red otherwise.

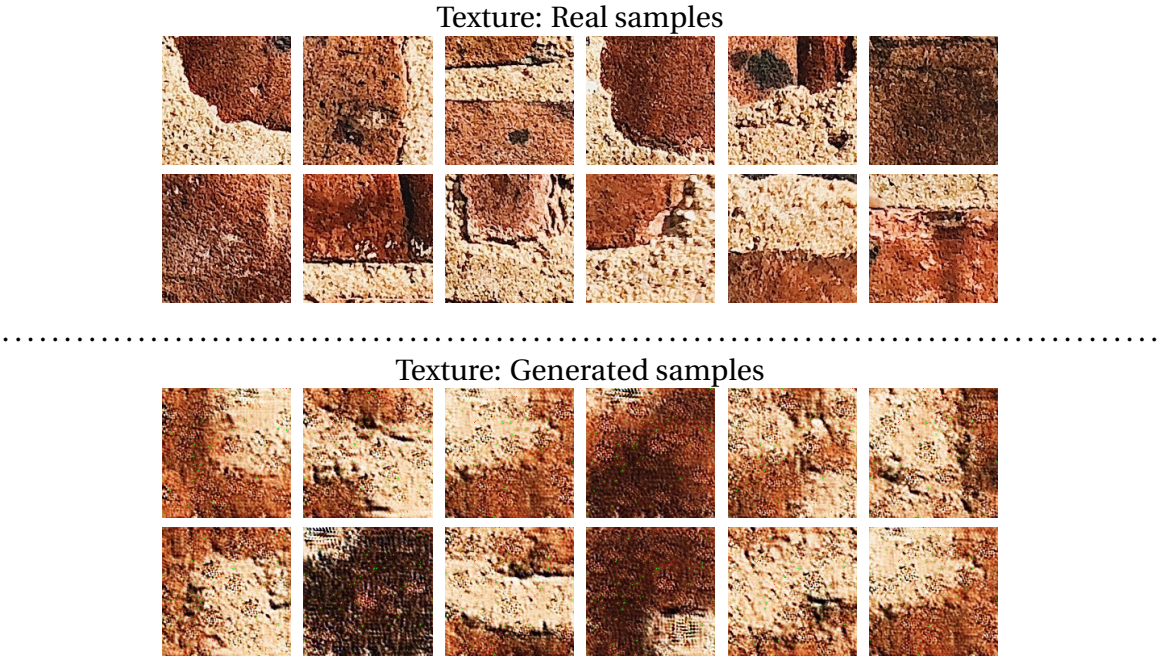


Figure B.3: Real and generated samples from the Texture dataset.

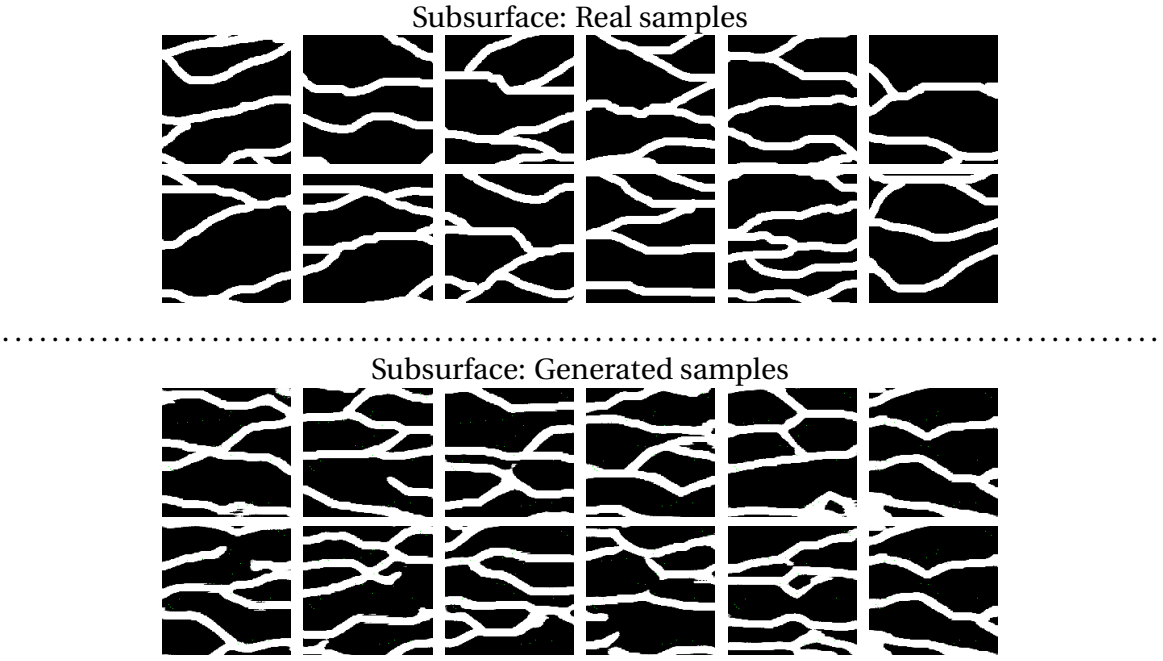


Figure B.4: Real and generated samples from the Subsurface dataset.

Appendix C

Experiment details for the Polarimetric CycleGAN