Auxiliary tasks for the conditioning of generative adversarial networks

Cyprien Ruffino

Normandie Université

THÈSE

Pour obtenir le grade de Docteur de Normandie Université

Spécialité Informatique

École Doctorale Mathématiques, Information, Ingéniérie des Systèmes

Auxiliary Tasks for the Conditioning of Generative Adversarial Networks

Conditionnement par tâches auxiliaires des réseaux antagonistes génératifs

Présentée et soutenue par

Cyprien RUFFINO

Dirigée par Gilles GASSO et encadrée par Romain HÉRAULT

Thèse soutenue publiquement le 20 avril 2021 devant le jury composé de		
M. Patrick GALLINARI	Professeur, Sorbonne Université - LIP6 & Criteo	Rapporteur
M. Jakob VERBEEK	Chercheur HdR, Inria & Facebook AI Research	Rapporteur
M. Stéphane CANU	Professeur, Normandie Université - LITIS	Examinateur
Mme Cécile CAPPONI	Professeure, Université Aix-Marseille - LIS	Examinatrice
M. John Aldo LEE	Professeur, FNRS & UCLouvain (Belgique) - MIRO	Examinateur
M. Gilles GASSO	Professeur, Normandie Université - LITIS	Directeur
M. Romain HÉRAULT	Maître de conférences HdR, Normandie Université - LITIS	Encadrant

INSA ROUEN NORMANDIE

MIIS

litis

Abstract

During the last decade, Generative Adversarial Networks (GANs) have caused a tremendous leap forward in image generation as a whole. Their ability to learn very complex, high-dimension distributions not only had a huge impact on the field of generative modeling, their influence extended to the general public at large. By being the first models able generate high-dimension photo-realistic images, GANs very quickly gained popularity as an image generation and photo manipulation technique. For example, their use as "filters" became common practice on social media, but they also allowed for the rise of *Deepfakes*, images that have been manipulated in order to fake the identity of a person.

In this thesis, we explore the conditioning of Generative Adversarial Networks, that is influencing the generation process in order to control the content of a generated image. We focus on conditioning through auxiliary tasks, that is we explicitly implement additional objective to the generative model to complement the initial goal of learning the data distribution.

First, we introduce generative modeling through several examples, and present the Generative Adversarial Networks framework. We discuss theoretical interpretations of GANs as well as its most prominent issues, notably the lack of stability during training of the model and the difficulty to generate diverse samples. We review classical techniques for conditioning GANs and propose an overview of recent approaches aiming to both solve the aforementioned issues and enhance the visual quality of the generated images.

Afterwards, we focus on a specific generation task that requires conditioning: image reconstruction. In a nutshell, the problem consists in recovering an image from which we only have a handful of pixels available, usually around 0.5%. It stems from an application in geostatistics, namely the reconstruction of underground terrain from a reduced amount of expensive and difficult to obtain measurements. To do so, we propose to introduce an explicit auxiliary reconstruction task to the GAN framework which, in addition to a diversity-restoring technique, allows for the generation of high-quality images that respect the given measurements.

Finally, we investigate a task of domain-transfer with generative models, specifically transferring images from the RGB color domain to the polarimetric domain. Polarimetric images bear hard constraints that directly stem from the physics of polarimetry. Leveraging on the cyclic-consistency paradigm, we extend the training of generative models with auxiliary tasks that push the generator towards enforcing the polarimetric constraints. We highlight that the approach manages to generate physically realistic polarimetric images. Empirical evidence illustrates that using the generated images as data augmentation improves the performance on object detection models for road scene analysis.

Résumé

Au cours de la dernière décennie, les réseaux génératifs antagonistes (Generative Adversatial Networks, ou GANs) ont révolutionné la génération d'images dans son ensemble. Leur capacité à apprendre des distributions très complexes en grande dimension ils ont eu eu un impact important sur le domaine des modèles génératifs et leur influence s'est largement étendue au grand public. En effet, en étant les premiers modèles capables de générer des images photo-réalistes en haute dimension, ils ont très vite gagné en popularité en tant que technique de génération d'images et de manipulation de photos. Par exemple, leur utilisation en tant que "filtres" est devenue une pratique courante sur les médias sociaux : ils ont également permis l'essor des *Deepfakes*, des images manipulées afin de falsifier l'identité d'une personne.

Dans cette thèse, nous étudions le conditionnement des réseaux génératifs antagonistes, c'est-à-dire influencer le processus de génération afin de contrôler le contenu d'une image générée. Nous nous concentrons sur le conditionnement par le biais de tâches auxiliaires, c'est-à-dire l'utilisation d'un ou plusieurs objectifs supplémentaires au modèle génératif en plus de l'objectif initial d'apprentissage de la distribution des données.

Nous introduisons les principes de la modélisation générative à travers plusieurs exemples, et nous présentons le cadre des réseaux génératifs antagonistes. Nous analysons les interprétations théoriques de ce modèle ainsi que ses problèmes les plus importants, notamment l'instabilité de l'apprentissage du modèle et la difficulté de générer des échantillons diversifiés. Nous passons en revue les techniques classiques de conditionnement des GAN et proposons un aperçu des approches récentes visant à résoudre ses problèmes et à améliorer la qualité visuelle des images générées.

Dans la suite de la thèse, nous nous concentrons sur une tâche de génération spécifique qui nécessite un conditionnement : la reconstruction d'images. Ce problème consiste à générer une image dont nous ne connaissons qu'un nombre très réduit de pixels à priori, généralement autour de 0,5 %. Ceci est motivé par une application directe en géostatistique : la reconstruction de données géologiques de sous-sols à partir d'une très petite quantité de mesures coûteuses et difficiles à obtenir. Pour ce faire, nous proposons d'introduire une tâche de reconstruction auxiliaire explicite dans le cadre du GAN qui, combinée à une technique de restauration de la diversité, a permis de générer des images de haute qualité qui respectent les mesures données.

Dans la deuxième contribution nous étudions une tâche de transfert de domaine avec des modèles génératifs, en particulier le transfert d'images du domaine couleur au domaine polarimétrique. Les images polarimétriques sont soumises à des contraintes strictes qui découlent directement des proprétés physiques de la polarimétrie. En s'appuyant sur l'approche de cohérence cyclique, nous étendons la formulation des modèles génératifs

avec des tâches auxiliaires qui poussent le générateur à faire respecter les contraintes polarimétriques. Nous montrons que cette approche permet non seulement de générer des images polarimétriques physiquement réalistes, mais que l'utilisation des images générées comme données augmentées augmente la performance des modèles de détection d'objets sur des applications d'analyse de scène routière.

Remerciements

Il est difficile d'estimer l'impact que certaines personnes peuvent avoir sur notre vie. Cependant, si il est un témoignage de cet impact sur la mienne, c'est l'existence de ce manuscrit. A ce titre, j'ai bien du monde à remercier.

J'aimerais donc, par ordre chronologique, remercier mes professeurs d'informatique théorique, qui m'ont en premier donné envie de m'orienter vers la recherche : Nicolas Ollinger, Matthieu Liedloff, Jean-Michel Couvreur et Ioan Todinca; même si finalement mon choix s'est finalement orienté vers l'intelligence artificielle. J'aimerais également remercier Thierry Paquet, Yann Soullard et Christopher Kermorvant qui m'ont fait suffisamment confiance pour m'embaucher en tant que stagiaire au LITIS, et ce malgré mon bagage limité dans le domaine du *machine learning*.

Je ne remercierai jamais assez Gilles Gasso et Romain Hérault pour leur supervision, pour toute la science que l'on a pu partager, pour toute la patience qu'ils ont sû m'accorder, et surtout pour leur bienveillance. Leur présence tout au long des 4 ans de cette thèse aura été indispensable à l'existence de cette thèse.

J'aimerai également remercier tous mes camarades du LITIS. Que ce soit côté fac': Guillaume, Achille, Rosana, Yann, Andres, Wassim, Gaëtan, Sen ou Fabrice; comme côté INSA: Ismaïla, Rachel, Franco, Matthieu, Ben, JB, Flavie, Linlin, Nikolas ou Soufiane. Que ce soit pour toutes les discussions dans les couloirs ou pour les parties de tarot à la pause café, l'ambiance au labo est excellente et les discussions toujours enrichissantes. Je remercie également toutes mes collaborateurs, notamment Samia Ainouz, Stéphane Canu, Rachel Blin et Eric Laloy. Je remercie également mon jury de thèse de l'intérêt qu'ils ont porté à mes travaux et pour avoir accepté de relire ma thèse et participer à ma soutenance.

Ensuite, j'aimerai énormément remercier les copains Rouennais. Achille, Diego, Victor, Lauréline, Val', Cranky, et de manière générale toute la "team cafard". Votre accueil sans égal au pays de la crème fraiche et du cidre m'aura fait passer de sacrés bons moments et aura bien contribué à abaisser la pression du doctorat. Un énorme merci également aux amis de longue date, Nico, Cassandre, Najib, JF, Romu et Pierre, c'est toujours une joie incroyable de se retrouver dès que l'occasion s'y prête.

Vient enfin le tour de ma famille. Je pense que je ne serai jamais capable d'exprimer comme je le souhaiterai la reconnaissance que j'ai envers eux. De la famille plus éloignée à la plus proche, leur bienveillance et leur attention ont été le carburant de mon travail. Mais c'est évidemment à mes parents et mes frères-et-sœurs que je dois le plus. M'man, P'pa, Gab, Flo, Zach, je sais que vous lirez éventuellement ce message. Vous êtes le plus grand soutien de ma vie, et pas une ligne de cette thèse n'existerai sans vous. Dans les moments agréables comme dans les plus difficiles, vous avez toujours été là. Je ne vous en remercierai jamais assez.

Contents

List of Figures

List of Tables

List of Acronyms

AP	Average Precision
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
HOG	Histograms of Oriented Gradients
IPM	Integral Probability Metric
IS	Inception Score
JS	Jensen-Shannon (Divergence)
KL	Kullback-Leibler (Divergence)
LiDAR	Light Detection And Ranging
LBP	Local Binary Patterns
LSGAN	Least-Squares Generative Adversarial Networks
mAP	Mean Average Precision
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 en français

Contexte

Au cours de la dernière décennie, l'apprentissage profond (ou *deep learning*) est apparu comme l'un des domaines les plus prometteur de l'intelligence artificielle, égalant ou surpassant progressivement toutes les approches traditionnelles dans nombre de domaines d'application. Grâce à la capacité de généralisation des réseaux de neurones profonds, l'apprentissage profond est capable d'utiliser des masses de données afin d'en extraire des motifs et des comportements pertinents. L'apprentissage profond s'est donc naturellement appliqué à des domaines aussi différents que la traduction automatique, le jeu de Go et le *trading* à haute-fréquence. Parmi eux, la vision par ordinateur est très certainement le domaine sur lequel l'apprentissage profond aura eu le plus grand impact. Consistant à analyser et traiter des images automatiquement, la vision par ordinateur est un domaine complexe comprenant de nombreux problèmes tels que la détection d'objets ou la reconnaissance de formes. L'apprentissage profond est désormais l'approche de référence pour toutes les approches de vision par ordinateur et s'applique à des domaines tels que l'imagerie médicale, la reconnaissance faciale ou la conduite autonome.

Un des sous-domaines de la vision par ordinateur ayant a connu un essor fulgurant grâce à l'apprentissage profond est la génération automatique d'images. Les réseaux génératifs antagonistes (Generative Adversarial Networks, ou GANs) (Goodfellow et al., 2014), mis en avant lors de ces dernières années pour leur capacité à générer des images photo-réalistes[1], sont désormais le fer de lance de l'apprentissage profond pour la génération d'images. En permettant la génération d'images de haute qualité et de grande dimension, ils ont rapidement trouvé des applications dans de nombreux domaines techniques tels que l'augmentation de résolution d'image (Wang et al., 2020), la cartographie automatique (Kang et al., 2019), la génération de vidéos (Vondrick et al., 2016) ou la génération automatique d'objets 3D (Wu et al., 2017). L'usage des GANs s'est également étendu à des applications destinées au grand public : certaines inoffensives telles que les très nombreux "filtres" disponibles sur des réseaux sociaux permettant de par exemple de générer une photo d'une personne vieillie (Antipov et al., 2017); d'autres plus néfastes, comme les fameux "deepfakes" (Vaccari and Chadwick, 2020) des images et vidéos automatiquement générées dont le but est de tromper en falsifiant l'identité d'une personne, le plus souvent une célébrité ou une personnalité politique.

Dans cette thèse, nous proposons d'étudier les tâches auxiliaires pour le conditionnement des réseaux génératifs antagonistes. Si les GANs excellent dans la génération d'images

[1] Un exemple particulièrement frappant : `https://www.whichfaceisreal.com/`

et permettent d'obtenir des images de très haute qualité, ils ne présentent leur plein potentiel que lorsqu'ils sont conditionnés, c'est à dire qu'il est possible d'exercer un contrôle sur la sortie du modèle. En effet, c'est ce conditionnement qui permet de s'assurer que l'image obtenue est bien celle attendue et est donc indispensable pour, par exemple, l'ensemble des applications d'édition dynamique d'images telles que les "filtres" ou l'augmentation de résolution.En particulier nous nous concentrons sur une famille de techniques pour ce conditionnement : les tâches auxiliaires. En entraînant un GAN à résoudre une tâche secondaire en parallèle de son apprentissage de la distribution des données réelles, il est possible de le pousser à respecter certaines propriétés désirées. Ces tâches auxiliaiaires nécessitent d'être conçues spécialement pour chaque type de conditionnement. Au cours de cette thèse, nous examinons donc des problèmes pouvant être résolus par des modèles génératifs conditionnées et proposons des tâches annexes appropriées pour résoudre ces problèmes.

Motivations

Nos travaux sont motivés par deux applications directes nécessitant des modèles génératifs conditionnés : la reconstruction d'images, et plus précisément de cartographies de formations de canaux d'eau souterrains ; et la conversion de bases de données d'images couleur de scènes routières dans le domaine polarimétrique.

Reconstruction d'images hydro-géologiques

Nous étudions le problème de la reconstruction d'images, consistant à (ré-)générer une image à partir d'un ensemble très réduit de pixels connus a priori, qui est ici un cadre générique s'appliquant au problème de reconstruction de cartographies de formations de canaux d'eau souterrains. Dans le cadre de cette applicationn menée en collaboration avec le SCK·CEN (Belgique), plusieurs critères sont recherchés :

Précision au pixel près. La tâche de reconstruction d'images consistant à générer une image dont des pixels précis sont pré-tirés, leurs positions et valeurs doivent être préservées dans l'image obtenue. Dans le cadre de l'application en géologie, cela implique de préserver précisément la position et les valeurs des mesures réelles effectuées sur le terrain.

Préservation de la diversité. L'une des limitations des GAN est leur tendance à perdre la capacité à générer des échantillons diversifiés et ainsi ne produire que des images très proches les unes des autres. Dans le cadre de l'application en géologie, il est important de pouvoir produire un grand nombre d'images candidates diversifiées qui respectent les pixels pré-tirés.

Génération rapide. Afin de pouvoir générer un grand nombre d'images candidates, il est également important que le processus de génération soit rapide. Ainsi, les approches existantes nécessitant de résoudre un problème d'optimisation pour chaque image générée seront le plus souvent bien trop lentes pour être applicables ici.

Conversion d'images couleur en images polarimétriques

Dans cette application, nous nous penchons sur le problème de génération d'images polarimétriques comme moyen d'augmentation de données. En effet, le manque de données polarimétriques étiquetées est un frein important pour la recherche dans le domaine de la vision par ordinateur sur les images polarimétriques. Ces images, captant des propriétés de la lumière qui ne sont pas présentes dans des images couleur, permettent par exemple d'obtenir de meilleurs résultats dans des tâches de détection d'objets dans des conditions météorologiques adverses, telles qu'une pluie importante ou de la brume. En transférant des bases de données étiquetées du domaine de l'image couleur au domaine polarimétrique, cette pénurie de données polarimétriques étiquetées pourrait être contournée. Cependant, plusieurs exigences sont à respecter :

Respect des contraintes polarimétriques. L'imagerie polarimétrique est soumise à des contraintes fortes émanant de la physique ondulatoire de la lumière. Ces contraintes doivent être prises en compte afin de générer des images non seulement réalistes, mais surtout ayant les propriétés physiques permettant d'obtenir de bons résultats dans des conditions météorologiques adverses.

Respect de la calibration de la caméra polarimétrique. Pour pouvoir capturer des images polarimétriques, une caméra spécialisée utilise un certain nombres de filtres laissant passer la lumière polarisée à des angles prédéfinis. La configuration de ces filtres peut différer selon la caméra utilisée. Cette calibration affecte directement la nature des images acquises. Ainsi, lors de la génération de données polarimétriques, il est nécessaire de pouvoir assurer que les images produites correspondent à la calibration de la caméra.

Préservation du contenu de l'image. L'objectif de cette application est de produire des bases de données étiquetées artificielles en transférant dans le domaine polarimétrique des bases de données existantes d'images couleur. Afin de pouvoir conserver les étiquettes entre ces deux domaines, il est donc nécessaire que le contenu des images reste similaire en nature et en position.

Processus de génération rapide. Il est également nécessaire que le temps de génération d'une image ne soit pas trop élevé, puisque les bases de données visées peuvent contenir plusieurs centaines de milliers d'images de résolution élevée. Un temps de génération trop important rendrait ainsi cette approche prohibitive.

Structure de la thèse

Puisque le conditionnement des modèles génératifs est une étape cruciale pour leur application à des problèmes du monde réel, nous proposons dans cette thèse d'étudier le conditionnement des réseaux adversaires générateurs, notamment en utilisant des tâches auxiliaires. La thèse est structurée en trois chapites, dont les contenus sont résumés ci-dessous, et une conclusion.

Chapitre 1 : Introduction aux réseaux génératifs antagonistes

Nous commençons cette thèse par un chapitre introductif sur les réseaux génératifs antagonistes (GANs), une méthode pour entraîner des réseaux de neurones profonds comme

modèles génératifs particulièrement appropriée pour la génération d'images. Nous mettons également en exergue leurs limitations, notamment l'instabilité du processus d'entraînement et le manque de diversité statistique dans les données générées. Nous nous penchons également sur les variantes conditionnelles des GAN, permettant d'exercer un certain contrôle sur le processus de génération en appliquant des contraintes sur la donnée générée, ainsi que les approches de transfert de domaines, tâche consistant à projeter une donnée d'un domaine vers un autre (par exemple, convertir un tableau de maître en photo). Nous présentons un aperçu des différentes techniques employées pour contrecarrer les limitations intrinsèques des GANs par l'amélioration de l'architecture des réseaux de neurones ainsi que le changement de la fonction de coût du GAN. Enfin, nous terminons ce chapitre par une réflexion sur les difficultés que représentent l'évaluation des modèles génératifs et examinons les métriques les plus couramment utilisées pour évaluer les GANs.

Chapitre 2 : La reconstruction d'images comme tâche auxiliaire à la modélisation générative

Dans ce chapitre, nous proposons un aperçu de la tâche de reconstruction d'images à l'aide de modèles génératifs. Comme contribution, nous proposons une approche de conditionnement des GANs utilisant une tâche de reconstruction auxiliaire explicite. En optimisant cette tâche auxiliaire pendant le processus de génération, combinée à une technique permettant de limiter les problèmes de perte de diversité, les modèles obtenus sont capables de reconstruire rapidement les images, en comparaison avec des méthodes similaires, telles que les approches basées sur l'acquisition comprimée, devant résoudre un problème d'optimisation pour chaque image reconstruite. Un sous-produit de notre approche est un hyper-paramètre qui contrôle l'impact de la fonction de coût liée à la tâche de reconstruction sur le modèle génératif. Nous montrons que cet hyper-paramètre influence directement un compromis entre la fidélité de la reconstruction et la qualité visuelle des images générées.

Nous évaluons notre approche sur plusieurs tâches de reconstruction d'images en utilisant des ensembles de données d'images classiques comme MNIST ou CIFAR10, ainsi qu'un ensemble de données d'images de texture. Enfin, nous appliquons cette méthode à un problème de géologie, à savoir la reconstruction des formations de canaux d'eau souterrains en utilisant très peu de points. Les résultats expérimentaux montrent que notre approche obtient des résultats égalux ou supérieurs aux approches existantes tout en offrant la possibilité de contrôler le compromis entre la qualité visuelle et le respect des contraintes.

Chapitre 3 : Transfert de domaines avec tâches auxiliaires pour la modélisation générative

Dans ce chapitre, nous étudions le conditionnement des modèles de transfert de domaines qui utilisent les réseaux génératifs antagonistes. Ces modèles, généralement basés sur l'idée de cohérence cyclique (ou *cyclic-consistency*), permettent de transférer des images d'un "domaine" à l'autre sans utiliser de données appariées, qui sont générale-

ment très difficiles à obtenir. Nous nous concentrons sur la tâche de transfert d'une image couleur vers le domaine polarimétrique. De telles images sont soumises à des contraintes strictes qui découlent directement de la physique ondulatoire de la lumière, de sorte que les approches de transfert de domaine sans contraintes ne peuvent pas résoudre ce problème à elles seules.

Nous introduisons de nouvelles tâches auxiliaires basées sur une reformulation de ces contraintes et proposons un algorithme pour les intégrer lors de l'entraînement d'un modèle de transfert de domaine. Nous montrons que cette méthode est performante dans une tâche de génération d'images polarimétriques, à la fois en termes de qualité visuelle et de respect des contraintes.

Enfin, nous appliquons cette approche à une tâche d'augmentation de données. En effet, aucune base de données d'images polarimétriques étiquetées n'est publiquement disponible au moment de la rédaction de cette thèse, ce qui rend difficile l'apprentissage de modèles profonds pour résoudre des problèmes sur des images polarimétriques. En transférant des bases de données d'images en couleur étiquetées dans le domaine des images polarimétriques, nous pouvons produire de grandes quantités d'images polarimétriques étiquetées. Nous montrons que de telles données augmentent la performance d'un réseau de détection d'objets dans les images polarimétriques pour l'analyse de scène routière.

Publications

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: Conférence Sur l'Apprentissage

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (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

- Eric Laloy, 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?" In: *Computers and Geosciences* 133

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (Nov. 27, 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing* 416, pp. 218–230

- Rachel Blin, Cyprien Ruffino, Samia Ainouz, Romain Hérault, Gilles Gasso, Fabrice Mériaudeau, and Stéphane Canu (2021). "Generating Polarimetric-Encoded Images Using Constrained Cycle-Consistent Generative Adversarial Networks". In: *Currently in Preparation*

Introduction

Context

Over the last decade, deep learning has emerged as one of the most promising areas of artificial intelligence, progressively equaling or surpassing all traditional approaches in several fields of application. Thanks to the generalization capacity of deep neural networks, it is able to leverage large amounts of data to learn complex patterns and behaviors. Deep learning has been applied successfully to diverse domains such as machine translation, the game of go and high-frequency trading. Among all these application domains, computer vision is surely the one in which deep learning has had the greatest impact. Consisting in analyzing and processing images automatically, computer vision is a complex field that contains many problems such as object detection or pattern recognition. Nowadays, deep learning is the reference approach for all computer vision tasks and is used in areas such as medical imaging, facial recognition and autonomous driving.

One of the sub-domains of computer vision that emerged thanks to deep learning is automatic image generation. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are now the spearhead of deep learning for image generation. They were made famous during recent years for their ability to generate photo-realistic images[1]. Indeed, by allowing for the generation of high quality and high dimensional images, they have quickly found applications in many domains such as increasing image resolution (Wang et al., 2020), cartography (Kang et al., 2019), video completion (Vondrick et al., 2016) or automatic 3D objects generation (Wu et al., 2017). The use of GANs has also been extended to application targeting the public at large, such as the numerous "filters" available on social networks, allowing for example to edit pictures of a person to make them look older (Antipov et al., 2017). GANs also lead to some more harmful applications, such as the famous "deepfakes" (Vaccari and Chadwick, 2020) that automatically generates images and videos whose purpose is to deceive by falsifying the identity of a person, most often a celebrity or a politician.

In this thesis, we propose to study auxiliary tasks for the conditioning of generative adversarial networks. While GANs excel in image generation and allow for generating very high quality images, they only reach their full potential when they are conditioned, i.e. when it is possible to control the model output. Indeed, the conditioning makes it possible to ensure that the obtained images have desired properties, which is essential for, for example, all dynamic image editing applications such as "filters" or the increase in resolution. Indeed, the content of the image, for example the person on which the filter

[1] A striking example : `https://www.whichfaceisreal.com/`

is applied, must remain the same. We therefore propose to focus on a family of techniques conditioning GANs: auxiliary tasks. By training a GAN to solve a secondary task, simultaneously to learning the distribution of real data, it is possible to push the model towards respecting some targeted properties. These auxiliary tasks need to be designed specifically for each type of conditioning. Over the course of this thesis, we will examine examples of problems that can be solved by conditioned generative models and will propose appropriate auxiliary tasks to solve these problems.

Motivations

Our work is motivated by two applications requiring conditioned generative models: the reconstruction of images, and more precisely maps of underground water channel formations; and the conversion of road scene RGB image databases into the polarimetric domain.

Reconstruction of hydro-geological images

First, we study the problem of image reconstruction, consisting in (re-)generating an image from a very reduced set of a priori known pixels. This is a generic task which includes the problem of reconstructing maps of underground water channel formations. Within this application, we seek several properties:

Pixel precise. Since the task of image reconstruction consists in generating an image from which precisely-positioned pixels are pre-drawn, their positions and values must be preserved in the resulting image. In the context of the geology application, this implies preserving precisely the position and value of the real measurements made on the field.

Preserves diversity. One of the limitations of GANs is their tendency to loose the ability to generate diverse samples, and thus they produce images that are very close to each other. In geological applications, it is important to be able to produce a large number of diverse candidate images that fulfill the pre-drawn pixels.

Fast generation process. In order to be able to generate a large number of candidate images, it is also important that the generation process is fast. Thus, existing approaches that require solving an optimization problem for each generated image will most often be far too slow to be applicable here.

Polarimetric Image Conversion

In a second step, we address the problem of polarimetric image generation as a data augmentation technique. Indeed, the lack of labeled polarimetric data is a major impediment to research in the field of computer vision in polarimetric images. These images, capturing properties of light that are not present in color images, allow, for example, for better results in detection tasks in adverse weather conditions such as heavy rain or fog. By transferring labeled databases from the color image domain to the polarimetric domain, this shortage of labeled polarimetric data could be circumvented. However, there are several requirements that must be met:

Respect of polarimetric constraints. Polarimetric imaging is subject to strong constraints emanating from the wave physics of light. These constraints must be taken into account in order to generate images that are not only realistic, but above also the physical properties that may allow improved detection results in adverse weather conditions.

Respect of the polarimetric camera calibration. In order to capture polarimetric images, a dedicated camera uses a number of filters that let polarized light pass through at predefined angles. The configuration of these filters may differ depending on the used camera. The calibration directly affects the nature of the acquired images. Thus, when generating polarimetric data, it is necessary to ensure that the produced images correspond to the calibration of the used camera.

Preserves the image content. The objective of this application is to produce synthetic labeled datasets by transferring existing labeled datasets into the polarimetric domain. In order to be able to preserve the labels between these two domains, it is therefore necessary that the image content remains similar in nature and position.

Quick generation process. In order to be able to transfer entire databases into the polarimetric domain, it is also necessary that the image generation time is not too high, since these databases may contain several hundreds of thousands of high-resolution images. High generation times would thus make the approach prohibitively expensive.

Outline and contributions

Since conditioning generative models is a crucial step for applying them to real-world problems, in this thesis we study the conditioning of Generative Adversarial Networks, most notably using auxiliary tasks. The thesis is composed of three chapters, whose contents are detailed below, as well as a conclusion.

Chapter 1: Introduction to Generative Adversarial Networks

We begin the thesis with an introduction chapter on Generative Adversarial Networks (GANs), a framework for training deep neural networks as generative models that is particularly well suited for image generation. We highlight their limitations, most notably the instability of the training process and the lack of diversity in the generated data. We investigate its conditional variants. These allow for exerting some control over the generation process by applying constraints on the generated data, as well as domain transfer approaches, the task of projecting data from one domain to another (e.g., converting a painting into a photo). We present an overview of the different techniques used to counteract the limitations of GANs by improving the neural network architecture and changing the cost function of the GAN. Finally, we conclude this chapter by discussing the difficulties involved in evaluating generative models and examine the most commonly used metrics for evaluating GANs.

Chapter 2: Image reconstruction as an auxiliary task to generative modeling

In this chapter, we propose an overview of the task of reconstructing altered images with generative models. As a contribution to this problem, we propose an approach for conditioning GANs using an explicit auxiliary reconstruction task. Combined with a technique for limiting the diversity-loss issues, optimizing this auxiliary task during the training process, the obtained models are able to quickly reconstruct images, in comparison to similar methods, such as compressed sensing-based approaches, that need to solve an optimization problem for each reconstructed image. A byproduct of our approach is a hyper-parameter that controls the impact of the reconstruction loss on the generative model. We show that this hyper-parameter directly influences a trade-off between the fidelity of the reconstruction and visual quality of the generated images.

We evaluate our approach on several image reconstruction tasks using classical image datasets such as MNIST or CIFAR10, as well as a texture image dataset. Finally, we apply this method to a geology problem, namely reconstructing underground water channels formations using very few points. Empirical results show that our approach equals or outperforms existing approaches while providing the ability to control the trade-off between the visual quality and the fulfillment of the constraints.

Chapter 3: Domain-transfer with auxiliary tasks for generative modeling

In this chapter, we study the conditioning of domain-transfer models that makes use of Generative Adversarial Networks. Such models, usually revolving around the idea of cycle-consistency, allow for transferring images from one "domain" to the other without the used of paired data, which is usually very hard to obtain. We focus on the task of transferring a color image to the polarimetric domain. Such images bear hard constraints that directly stem from the physics of light, thus unconstrained domain-transfer approaches cannot solve this problem by themselves.

We introduce new auxiliary tasks based on a reformulation of these constraints and propose an algorithm to integrate them to the training of a domain-transfer model. We show that this method performs well on a polarimetric image generation task, both in term of visual quality and respect of the constraints.

Finally, we apply this approach to a data-augmentation task. Indeed, no large polarimetric images datasets are publicly available at the time of writing this thesis, so training deep models to solve problems on polarimetric images is difficult. By transferring color-images labeled datasets to the polarimetric images domain, we can produce large datasets of labeled polarimetric images. We show that such a dataset increases the performance of a detection network in polarimetric images.

Related publications

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: Conférence Sur l'Apprentissage
- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (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
- Eric Laloy, 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?" In: *Computers and Geosciences* 133
- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (Nov. 27, 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing* 416, pp. 218–230
- Rachel Blin, Cyprien Ruffino, Samia Ainouz, Romain Hérault, Gilles Gasso, Fabrice Mériaudeau, and Stéphane Canu (2021). "Generating Polarimetric-Encoded Images Using Constrained Cycle-Consistent Generative Adversarial Networks". In: *Currently in Preparation*

Chapter 1

Introduction to Generative Adversarial Networks

Chapter abstract

In this chapter, we propose an introduction to generative modeling and some solutions to tackle this problem. Specifically, we present the Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), a framework for training deep neural networks as generative models that is particularly suited to the task of image generation. We introduce the theoretical insight behind Generative Adversarial Networks and review different GAN variants for learning conditional models. We discuss their limitations, namely: the instability of the GAN training process; the lack of statistical diversity among the generated samples; and the difficulty to generate high-dimension, high-quality images. We discuss the recent advances to overcome some of these limitations, through the neural networks' architecture or variaiations of the objective functions. Finally, we consider the problem of evaluating generative models, notably the intrinsic quality of generated samples, and review the most commonly used metrics and their limitations.

Contents

1.1 Introduction to generative modeling

In this section, we first propose an introduction to generative modeling with a focus on latent variable 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 stochastic gradient descent.

We introduce recent approaches such as variational auto-encoders (VAEs) (Kingma and Welling, 2014), flow methods (Dinh et al., 2017; Kingma and Dhariwal, 2018) and the techniques they used to overcome this restriction and train models through maximum likelihood estimation.

1.1.1 Generative modeling with maximum likelihood estimation

Generative modeling is the task of learning a statistical model of the underlying probability distribution of some observable variable in order to generate samples from that distribution. In other words, it describes how data are generated in terms of a probabilistic model. Indeed, whereas a classification model tries to find decision boundaries by fitting a parametric model $p_{\theta_{Y|X}}$ (with parameter θ) to a conditional probability distribution $p_{Y|X}$ of data $\mathbf{x} \in \mathscr{X}$ and label $\mathbf{y} \in \mathscr{Y}$, a generative model aims to fit p_{θ_X} to p_X the intrinsic marginal distribution of the data and to provide a sampling mechanism based on p_{θ_X} (see Figure 1.1).

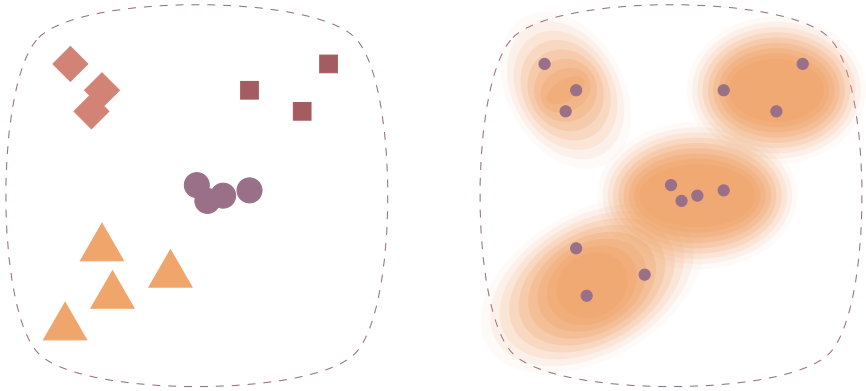


Figure 1.1: Discriminative modeling vs Generative modeling. Left: Discriminative modeling, the model aims to assign a target label to each sample. Right: Generative modeling, the model aims to learn the underlying probability distribution of the data.

Learning a discriminative model (Equation (1.1)) and a generative one (Equation (1.2)) can be formulated as a maximum log-likelihood estimation

$$\theta^* = \arg\max_{\theta} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{\mathrm{Y|X}}} \log p_{\theta_{\mathrm{Y|X}}} \ , \tag{1.1}$$

$$\theta^* = \arg\max_{\theta} \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}} \log p_{\theta_{\mathrm{X}}} \ . \tag{1.2}$$

A simple example of generative model are Gaussian Mixture Models (GMM) . Given $\mathbf{x} \in \mathbb{R}^d$, they consist in a sum of k Gaussian distributions $\mathcal{N}(\mu_i, \Sigma_i), 1 \leq i \leq k, \mu_i \in \mathbb{R}^d, \Sigma_i \in \mathbb{R}^{d\times d}$ which are all attributed a selection probability $p_{\mathrm{Z}}(z = i) = \pi_i$, with $\mathbf{z} \in \mathcal{Z}$, so that $p_{\mathrm{X|Z}=i} = \mathcal{N}(\mu_i, \Sigma_i)$. The GMM is then formulated as

$$p_{\theta_{\mathrm{X}}}(\mathbf{x}) = \sum_{\mathbf{z}} p_{\mathrm{Z}}(\mathbf{z})\, p_{\theta_{\mathrm{X|Z}}}(\mathbf{x}|\mathbf{z}) \ , \tag{1.3}$$

with the log-likelihood

$$\log \sum_{\mathbf{x}\sim p_{\mathrm{X}}} p_{\theta_{\mathrm{X}}}(\mathbf{x}) = \sum_{\mathbf{x}\sim p_{\mathrm{X}}} \log \sum_{i=1}^{k} \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \ . \tag{1.4}$$

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977) can be used to find the parameters θ^* maximizing the log-likelihood for such a model. Once the model is trained, sampling a new data is done by picking a component k from the distribution p_{Z} and then drawing a sample from the Gaussian distribution $p_{\mathrm{X|Z}=i} = \mathcal{N}(\mu_i^*, \Sigma_i^*)$.

1.1.2 Latent variable models

For GMMs, sampling a new point consists in, once the components have been selected, sampling a point according to a normal distribution. This sampling can be done by using reparametrization: instead of directly sampling $\mathbf{x} \sim \mathcal{N}(\mu_k^*, \Sigma_k^*)$, one can sample a latent variable $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and compute $\mathbf{x} = \mathrm{G}(\mathbf{z}; \mu, \Sigma) = \mu + \Sigma\mathbf{z}$. Such a model, that consists in

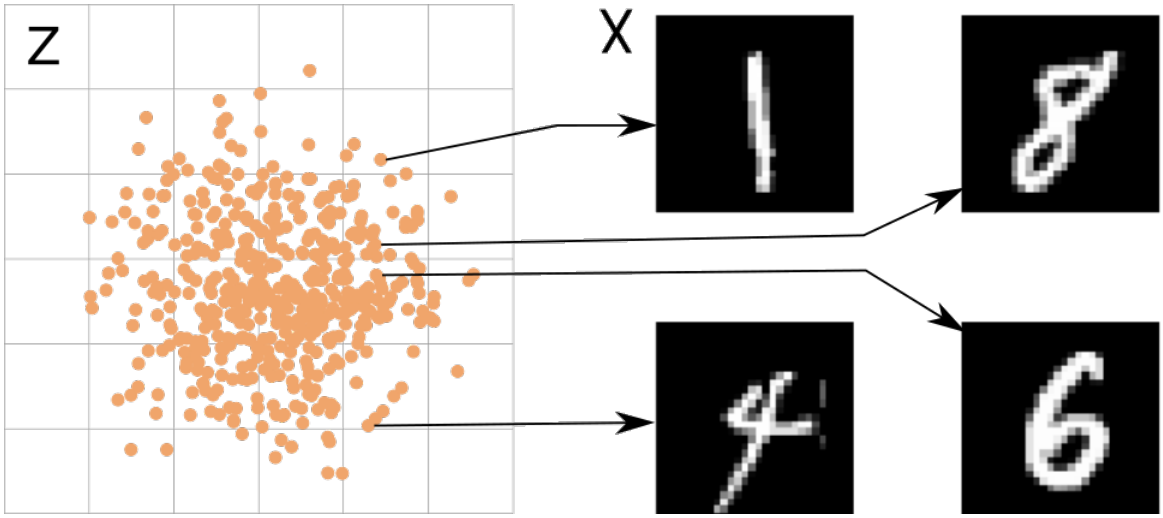


Figure 1.2: A mapping between a latent space $\mathcal{Z}$ and the high-dimensional space of an image set $\mathcal{X}$.

a deterministic function $G : \mathbb{Z} \to \mathbb{X}$ between $\mathbb{Z}$ the latent variable space and $\mathbb{X}$ the space of the data, with parameters θ ($\theta = (\mu, \Sigma)$ in this case) applied to a random latent variable drawn from a fixed distribution p_Z is a latent variable model (see Figure 1.2).

Since more complex distributions do not necessarily provide a simple sampling mechanism, using a latent variable model allows to outsource the stochastic part of the sampling process from the learning process and to only learn the deterministic function $G(\mathbf{z};\theta)$. More formally, instead of directly modeling p_X, a latent variable model learns a deterministic mapping $p_{G_{X|Z}}$. From this mapping, the full generative model can be obtained through marginalization

$$p_{G_X}(\mathbf{x}) = \int_{\mathcal{Z}} p_Z(\mathbf{z})\, p_{G_{X|Z}}(G(\mathbf{z};\theta))\, d\mathbf{z} \ . \tag{1.5}$$

The marginalization allows for the use of an arbitrary flexible G. However, the actual evaluation of p_{G_X} is very likely to be intractable due to the integral over $\mathcal{Z}$, which prevents the training of such a model as is. While the marginal distribution p_{G_X} cannot be explicitly computed for any function G, several solutions exist to overcome this problem. Hereafter, we describe some latent-variable methods to train deep generative models.

Variational Auto-Encoders

Variational Auto-Encoders (VAE) (Kingma and Welling, 2014) are deep latent variable models that learn the distribution of the latent model $p_{G_{X|Z}}$ using an auto-encoder[1] approach to train the generative model. In classical auto-encoders, two functions $F : \mathcal{X} \to \mathcal{Z}$ and $G : \mathcal{Z} \to \mathcal{X}$ are learned jointly by minimizing

$$L_{AE} = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X} ||\mathbf{x} - G(F(\mathbf{x}))|| \ , \tag{1.6}$$

[1] see Glossary, Appendix B

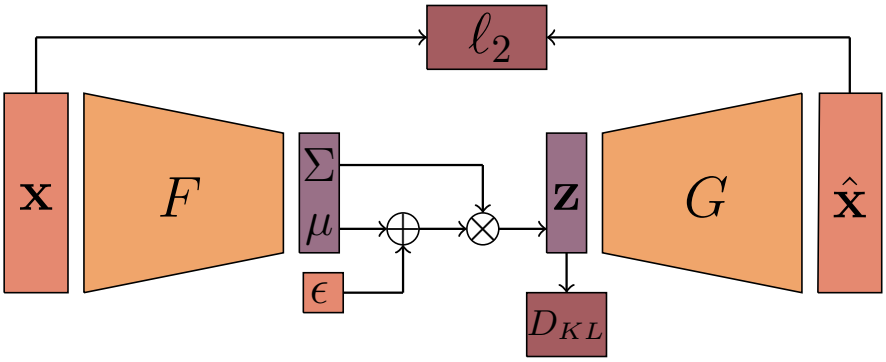


Figure 1.3: Variational auto-encoder framework

where $||.||$ is usually a ℓ_1, ℓ_2 or Frobenius norm, F is an encoding function that maps $\mathbf{x}$ to a latent representation $\hat{\mathbf{z}} = \mathrm{F}(\mathbf{x})$ and G is a decoding function that maps a latent variable $\mathbf{z}$ to a sample $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{z})$. However, in the case of generative modeling, $\mathbf{z}$ needs to be sampled from a random distribution so that generating a new sample $\hat{\mathbf{x}}$ can be done by sampling $\mathbf{z} \sim p_{\mathrm{Z}}$ and computing $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{z})$, with p_{Z} usually chosen to be $\mathcal{N}(0, \mathbf{I})$, with $\mathbf{I}$ the identity matrix. To do so, the VAE approach uses the so-called *reparametrization trick*, that consists in having $\mathrm{F}(\mathbf{x})$ output the mean and the covariance matrix $(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ of a normal distribution for each sample $\mathbf{x}$. By first sampling a random vector $\epsilon \in \mathbb{R}^p$ as $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and using it as a parameter to the model, the random latent code $\mathbf{z} \sim p_{\mathrm{Z|X}}$ can be computed as $\mathbf{z} = \mu_{\mathbf{x}} + \Sigma_{\mathbf{x}}\epsilon$. This is equivalent to sampling $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ and is differentiable by considering ϵ as a parameter. To train the model F, VAEs minimize the Kullback-Leibler (KL) divergence between the distribution $\mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ learned by the encoder and the real distribution $p_{\mathrm{Z|X}}$, and since p_{Z} is chosen Gaussian, this KL terms can be explicitly computed as

$$\mathrm{D_{KL}}\Big(\mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \Big|\Big| \mathcal{N}(0, \mathrm{I})\Big) = \frac{1}{2}\Big(\mathrm{Tr}(\Sigma_{\mathbf{x}}) + \mu_{\mathbf{x}}^{\top}\mu_{\mathbf{x}} - d - \log(\det \Sigma_{\mathbf{x}})\Big) \ , \tag{1.7}$$

with d being the dimension of the distribution $\mathcal{N}(0, \mathrm{I})$. By combining the auto-encoder and KL terms, we get the objective function of the VAE (summed up in Figure 1.3) defined as

$$\mathrm{L_{VAE}}(\mathrm{F}, \mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})}\left[||\mathbf{x} - \mathrm{G}(\mathbf{z})||_2^2\right] - \mathrm{D_{KL}}\Big(\mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \Big|\Big| p_{\mathrm{Z}}\Big) \ . \tag{1.8}$$

Once the model is trained, generating a new sample $\hat{\mathbf{x}}$ then consists in sampling a random vector $\mathbf{z} \sim \mathcal{N}(0, \mathbf{1})$ and computing $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{z})$.

Normalizing flows

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

$$p_{\mathrm{G_X}} = p_{\mathrm{Z}}\left|\det\left(\frac{\partial \mathrm{G}(\mathbf{z})}{\partial \mathbf{z}^{\mathrm{T}}}\right)\right|^{-1} = p_{\mathrm{G_X^{-1}}}\left|\det\left(\frac{\partial \mathrm{G}^{-1}(\mathbf{x})}{\partial \mathbf{x}^{\mathrm{T}}}\right)\right| \ , \tag{1.9}$$

with $\mathbf{z} \sim p_{\mathrm{Z}}$ a latent variable. This formulation has notable advantages such as explicitly allowing the computation of the exact inference, that is to compute $\mathbf{z}$ such that $\mathbf{x} = \mathrm{G}(\mathbf{z})$

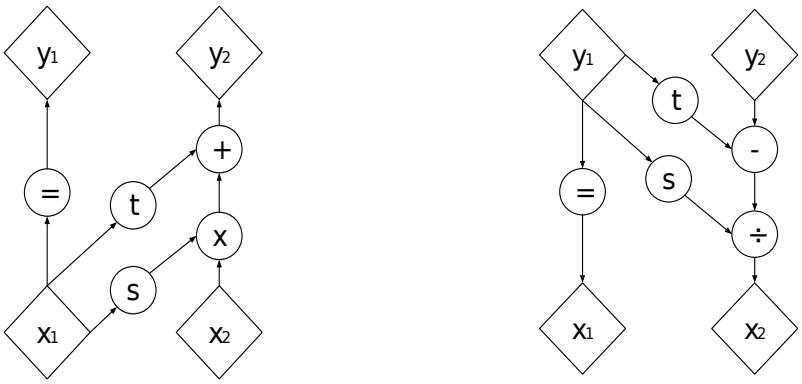


Figure 1.4: The affine transformation used in RealNVP. Left: Forward pass, Right: backward pass. A variable $\mathbf{x}$ is split in two variables $\mathbf{x}_1$ and $\mathbf{x}_2$. The non-linear functions $s(\mathbf{x}_1)$ and $t(\mathbf{x}_1)$ are then used to compute the output variables $\mathbf{y}_1 = \mathbf{x}_1$ and $\mathbf{y}_2 = s(\mathbf{x}_1)\mathbf{x}_2 + t(\mathbf{x}_1)$. Inverting this transformation can then be done by computing $\mathbf{x}_1 = \mathbf{y}_1$ and $\mathbf{x}_2 = (\mathbf{y}_2 - t(\mathbf{y}_1))/s(\mathbf{y}_1)$. Figure by Dinh et al. (2017)

for any given sample $\mathbf{x}$. However, the model has to enforce some tough constraints: the input and output dimensions must be the same; G must be invertible; and the computation of 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 transforms, which converts a set of variable by adding and scaling it by a non-linear transformation, usually computed with deep neural networks (see Figure 1.4). These transformations can be inverted by substracting and downscaling by the same transformed variables. Their Jacobians are triangular, therefore computing the determinants can be done efficiently by computing the product of their diagonal terms. ***Glow*** (Kingma and Dhariwal, 2018) extended this set of transformations to 1×1 invertible convolutions as well as a variant of Batch Normalization[1] (Ioffe et al., 2015) that allows for more expressiveness in the model.

1.2 Generative Adversarial Networks

As for the latent-variable generative models, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) aim to learn a mapping between samples from the latent distribution (usually normal or uniform) and samples from the data distribution. However, instead of directly relying on the likelihood and trying to estimate the distribution through marginalization, it aims to minimize the distance between the modeled distribution and the real data distribution. Therefore, GANs are sometimes qualified as *likelihood-free* generative models.

[1] see Glossary, Appendix B

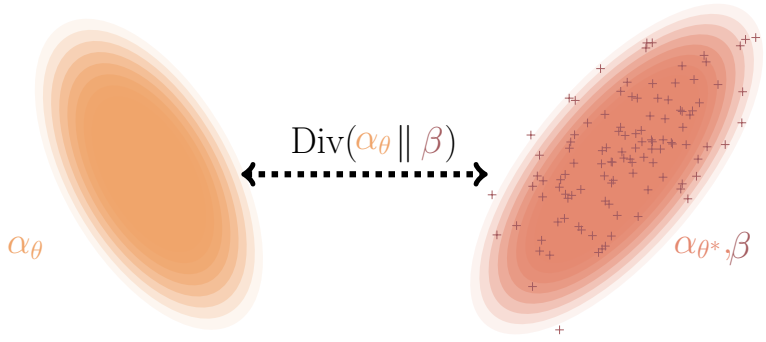


Figure 1.5: A divergence $\mathrm{Div}(\alpha_\theta||\beta)$ can capture the distance between a parametric density model α_θ and the distribution β from which a set of samples are observed. Density fitting can then be formulated as finding $\theta^* = \arg\min_\theta \mathrm{Div}(\alpha_\theta||\beta)$, such that α_{θ^*} is the best fit model.

We denote the generative model as a deterministic function $G : \mathbb{Z} \to \mathbb{X}$, that maps samples $\mathbf{z} \sim p_Z$ from the latent distribution p_Z and samples $\mathbf{x} \sim p_X$ from the the real data distribution. We seek the modeled distribution as $p_{G_X} = G_\sharp p_Z$[1].

In this section, we introduce the formulation of adversarial learning as an approximation of a divergence. Leveraging this formulation, we present the Generative Adversarial Networks framework as a pair of models, the generator model and a discriminator model, which is a binary classifier that aim to distinguish real and generated samples. Training these two models as a min-max problem, using alternate gradient descent, approximates the minimization of the Kullback-Leibler divergence between the real data distribution and the distribution of generated samples. We discuss some limitations of this model, most notably stability issues implied by alternate gradient descent, and the lack of statistical diversity.

1.2.1 Generative modeling through divergence approximation

A divergence $\mathrm{Div}(p_X||q_X)$ between two distributions p_X and q_X is a weak form of distance between these distributions (see Figure 1.5), thus minimizing such a divergence allows for a parametric distribution p_{θ_X} that fits a target distribution p_X. Whenever the divergence is both tractable and differentiable w.r.t the parameters θ, stochastic gradient descent can be used to estimate θ, thus allowing for the training of a generative model.

However in practice, such divergences are usually intractable for most distributions. Hence GANs aim to estimate the divergence by relying on another learnable function that will act as a surrogate to the divergence, the discriminator model D. This discriminator is trained as a binary classifier that predicts the probability of a sample $\mathbf{x}$ to be issued from the real distribution p_X or generated from $p_{G_\mathbf{x}}$ using binary cross-entropy (see Figure 1.6) as

$$L_D(D,G) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}[\log D(\mathbf{x})] + \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim p_{G_\mathbf{x}}}[\log D(1-\hat{\mathbf{x}})] \ . \tag{1.10}$$

[1] $\sharp$ is the push-forward operator (Bogachev, 2007) that transfers a probability distribution from one space to another using a function. Here, $G_\sharp p_Z$ represents the distribution obtained by "pushing" p_Z through the function G.

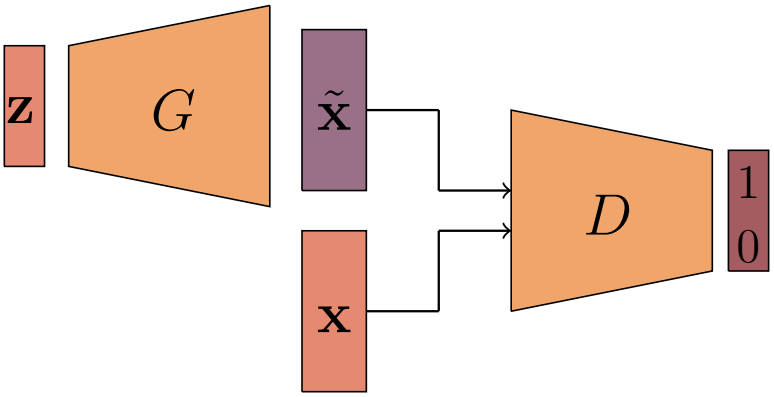


Figure 1.6: Generative Adversarial Network framework

The intuition behind this model is that once the discriminator is trained, maximizing its error on generated samples $\hat{\mathbf{x}} \sim p_{\mathrm{G_X}}$ w.r.t G will push $p_{\mathrm{G_X}}$ towards p_{X}.

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

$$\mathrm{D^*_G}(\mathbf{x}) = \frac{p_{\mathrm{X}}(\mathbf{x})}{p_{\mathrm{X}}(\mathbf{x}) + p_{\mathrm{G_X}}(\mathbf{x})} \ . \tag{1.11}$$

By plugging back Equation 1.11 into $\mathrm{L_D(D,G)}$ (Equation 1.10), we get

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

As said previously, the objective of the generator model G will be to maximize the error of the discriminator D. Thus, we can formulate the objective function $\mathrm{L_G(G)}$ related to G as $\mathrm{L_G(G)} = \max_{\mathrm{D}} \mathrm{L_D(D,G)}$. Up to additive and multiplicative constants, $\mathrm{L_G(G)}$ can be reformulated (Goodfellow et al., 2014) as

$$\mathrm{L_G(G)} = \mathrm{D_{KL}}\left(p_{\mathrm{X}} \middle|\middle| \frac{p_{\mathrm{X}} + p_{\mathrm{G_X}}}{2}\right) + \mathrm{D_{KL}}\left(p_{\mathrm{G_X}} \middle|\middle| \frac{p_{\mathrm{X}} + p_{\mathrm{G_X}}}{2}\right) = 2 \cdot \mathrm{D_{JS}}\left(p_{\mathrm{X}} \middle|\middle| p_{\mathrm{G_X}}\right) \ . \tag{1.12}$$

Algorithm 1 The GAN training algorithm

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

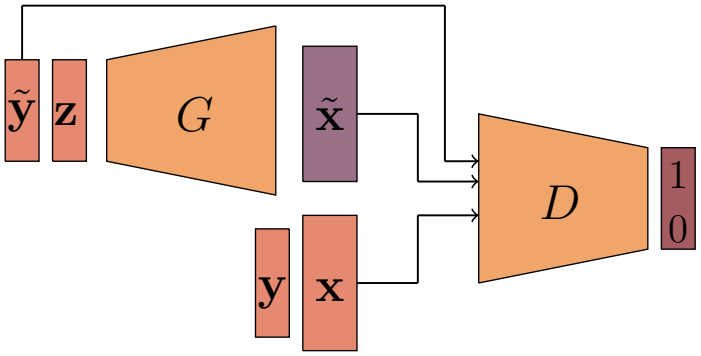


Figure 1.7: Conditional Generative Adversarial Networks

Assume the discriminator is trained to convergence. Minimizing $L_G(G)$ is therefore equivalent to minimizing the Jensen-Shannon (JS) divergence between p_X and p_{G_X}. This training process is summarized as the mini-max problem

$$\arg\min_G \min_G L_{GAN} = \arg\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X} [\log D(\mathbf{x})] + \mathop{\mathbb{E}}_{\mathbf{z}\sim p_Z} [\log 1 - D(G(\mathbf{z}))] \ . \tag{1.13}$$

From Equation 1.12, we find that the mini-max game has, assuming infinite capacity for both G and D, a global optimum for $p_X = p_{G_X}$. The GAN training algorithm then consists in alternatively updating the discriminator and the generator via gradient ascent/descent respectively. A summary of this process is presented in Algorithm 1.

1.2.2 Conditional modeling with generative adversarial networks

While classical generative models such as GANs try to unconditionally approximate the real-data distribution p_X, a conditional generative model aims to match the conditional distribution $p_{X|Y}$ related to $p_{X,Y}$ the joint distribution that constitutes the data, where $y \in \mathcal{Y}$ is a label of any kind.

Several extensions to the GAN framework allow for conditional modeling: **Conditional GANs (CGANs)** (Goodfellow et al., 2014; Mirza and Osindero, 2014), simply adds the label y as an input for both the discriminator and the generator (see Figure 1.7). It results the optimization problem

$$\arg\min_G \max_D L_{CGAN} = \arg\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{X,Y}} [\log D(\mathbf{x},\mathbf{y})] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}} [1 - \log D(G(\mathbf{y},\mathbf{z}),y)] \ . \tag{1.14}$$

Other approaches such as **Auxillary Classifier GAN (ACGAN)** (Odena et al., 2016) try to learn the conditional distribution by adding an explicit loss term to the optimization problem. ACGAN aims to learn a conditional generative model with discrete labels by adding another output, with dimension n equal to the number of labels, to the discriminator that acts as a classifier C sharing its weights with the discriminator (see Figure 1.8). The model is then trained by having both the generator and the discriminator minimize the categorical cross-entropy between the real and predicted labels.

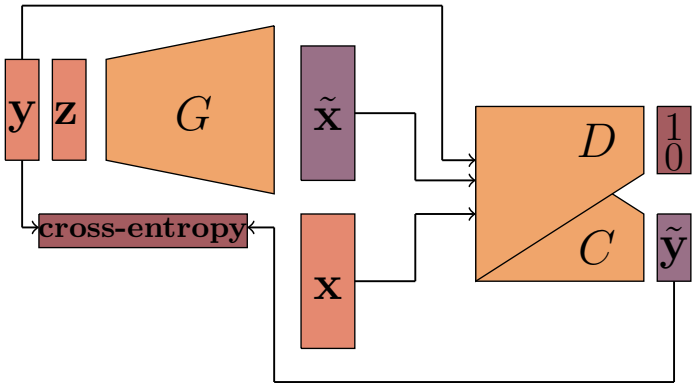


Figure 1.8: Auxiliary Classifier Generative Adversarial Networks approach.

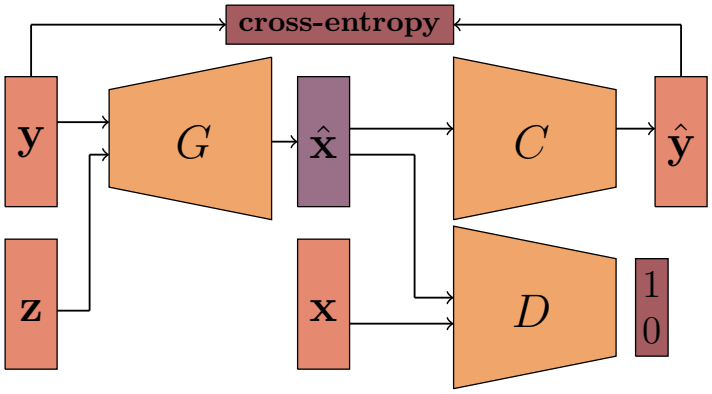


Figure 1.9: Triple Generative Adversarial Network approach.

$$\mathrm{L}_{\mathrm{ACGAN_D}}(\mathrm{D},\mathrm{G}) = \mathrm{L}_{\mathrm{GAN}}(\mathrm{D},\mathrm{G}) + \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim p_{\mathrm{X,Y}}}\left[-\sum_{i=1}^{n}\mathbf{y}_i \mathrm{C}(\mathbf{x})_i\right] , \tag{1.15}$$

$$\mathrm{L}_{\mathrm{ACGAN_G}}(\mathrm{D},\mathrm{G}) = \mathrm{L}_{\mathrm{GAN}}(\mathrm{D},\mathrm{G}) - \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}}\\ \mathbf{z}\sim p_{\mathrm{Z}}}}\left[-\sum_{i=1}^{n}\mathbf{y}_i \mathrm{C}(\mathrm{G}(\mathbf{z}))_i\right] . \tag{1.16}$$

TripleGAN (Li et al., 2017a) considers a classifier C, disjoint from D, which can be pre-trained or learned jointly to the GAN models. This classifier is then used to train the generator to generate images whose label $\hat{\mathbf{y}} = \mathrm{C}(\mathrm{G}(\mathbf{y},\mathbf{z})), \mathbf{y} \sim p_{\mathrm{Y}}, \mathbf{z} \sim p_{\mathrm{Z}}$ actually corresponds to the original label $\mathbf{y}$ (see Figure 1.9). This is done by adding a classification loss to the GAN objective function, as

$$\mathrm{L}_{\mathrm{T}riple\mathrm{GAN}}(\mathrm{D},\mathrm{G},\mathrm{C}) = \mathrm{L}_{\mathrm{GAN}}(\mathrm{D},\mathrm{G}) + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}}\\ \mathbf{z}\sim p_{\mathrm{Z}}}}\left[-\sum_{i=1}^{n}\mathbf{y}_i \log \mathrm{C}(\mathrm{G}(\mathbf{y},\mathbf{z}))\right] . \tag{1.17}$$

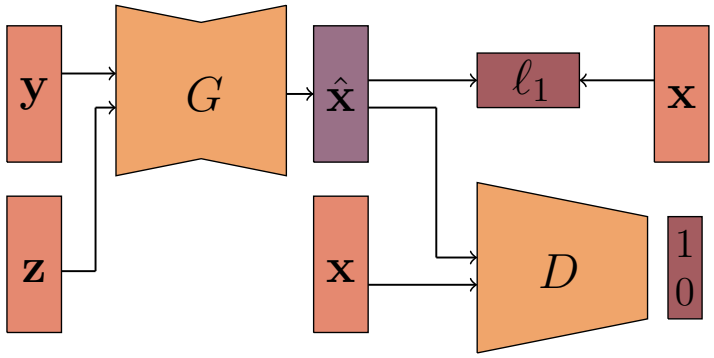


Figure 1.10: Pix2Pix domain-transfer approach

1.2.3 Domain-transfer approaches using generative adversarial networks

Domain-transfer is the task of learning a mapping $G_{YX} : \mathbb{X} \rightarrow \mathbb{Y}$ such that the generated samples $\hat{\mathbf{x}}$ are issued from the distribution p_X while maintaining some semantic information. This can be, for example, changing the color palette of an image, or transforming a photo of an object into a painting of the same object (see Figure 1.11).

Several approaches exist for domain-transfer (Isola et al., 2016; Taigman et al., 2017) that require paired samples $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_s, \mathbf{y}_s)\}, \mathbf{x}_i \in \mathbb{X}, \mathbf{y}_i \in \mathbb{Y}$ from both domains. CGANs already learn to model the conditional distribution $p_{X|Y}$, and adding a way to enforce the consistency of the semantic information enables domain-transfer.

Pix2Pix (Isola et al., 2016) implemented this approach explicitly by using paired samples $(\mathbf{x}, \mathbf{y}) \sim p_{X,Y}$ and by forcing the generator to minimize a reconstruction term (in this case, the authors choose the ℓ_1 norm) between $\mathbf{x}$ and $G(\mathbf{y}, \mathbf{z})$ (see Figure 1.10) as

$$\arg\min_{G_{YX}} \max_{D} L_{p2p} = \arg\min_{G_{YX}} \max_{D} L_{CGAN}(D, G) + \lambda \mathop{\mathbb{E}}_{\substack{(\mathbf{x},\mathbf{y})\sim p_{X,Y} \\ \mathbf{z}\sim p_Z}} ||\mathbf{x} - G_{YX}(\mathbf{y}, \mathbf{z})||_1 \ . \tag{1.18}$$

However, these approaches rely on paired data which can be very hard to obtain, especially in the case of natural images. Zhu et al. (2017a) present an example of this issue with a domain-adaptation task, in which a model turns images of horses into zebras. In such a case, it is very hard to get paired images of identical zebras and horses (see Figure 1.11). A solution to this problem of paired data was proposed in the form of **cyclic consistency** (Kim et al., 2017; Liu et al., 2018; Yi et al., 2017; Zhu et al., 2017c). Instead of training a single model G_{XY} with a reconstruction loss between $\mathbf{x}$ and $G(\mathbf{y})$, the cycle-consistent approaches train two domain-transfer models simultaneously: G_{YX} and G_{XY}

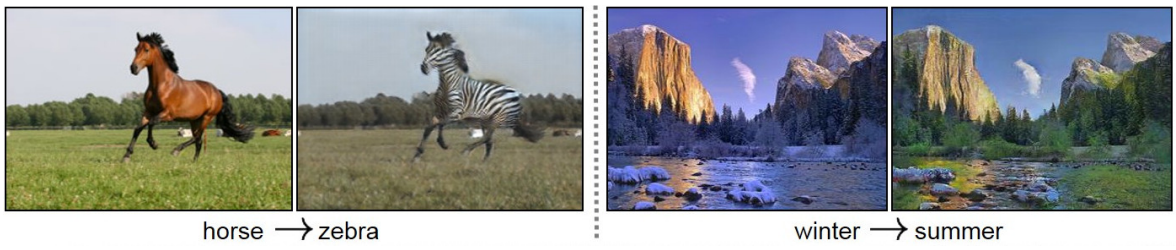


Figure 1.11: Examples of domain-transfer with CycleGAN (Zhu et al., 2017a). For this kind of images, paired data can be hard to acquire.

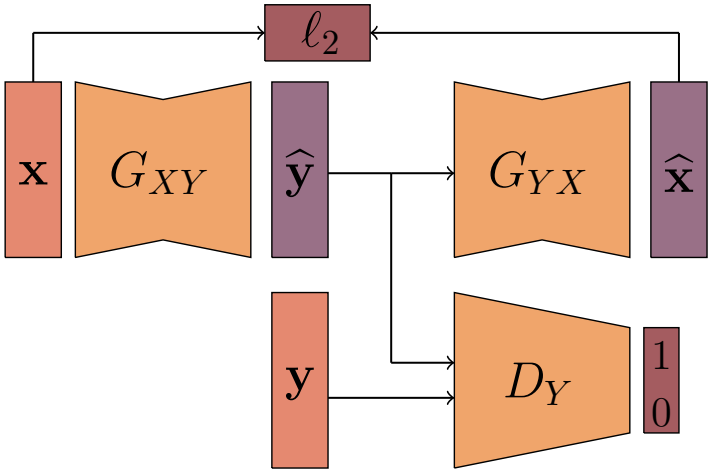


Figure 1.12: The CycleGAN approach. Half of the training setup is illustrated, the other half consisting in the same setup but with inverted $\mathbf{x}$ and $\mathbf{y}$

that push-forward p_Y onto p_X and p_Y onto p_X, respectively (see Figure 1.12). This allows to compute the reconstruction errors $||\mathbf{x} - G_{YX}(G_{XY}(\mathbf{x}))||_1$ and $||\mathbf{y} - G_{XY}(G_{YX}(\mathbf{y}))||_1$, which ensure that the content of the image is preserved through the mappings. Note that these reconstruction errors do not require paired data $(\mathbf{x}, \mathbf{y})$.

The training of the two models in done an adversarial setup, with two discriminators D_X and D_Y, and is wrapped up in the **CycleGAN** approach (Zhu et al., 2017a) as

$$\begin{aligned}
&\min_{G_{XY},G_{YX}} \max_{D_X,D_Y} L_{CycGAN}(G_{XY}, G_{YX}, D_X, D_Y) = \\
&\min_{G_{XY},G_{YX}} \max_{D_X,D_Y} L_{GAN}(G_{YX}, D_X) + L_{GAN}(G_{XY}, D_Y) \\
&\qquad + \lambda \Big[\mathop{\mathbb{E}}_{\mathbf{x}\sim p_X} ||\mathbf{x} - G_{YX}(G_{XY}(\mathbf{x}))||_1 + \mathop{\mathbb{E}}_{\mathbf{y}\sim p_Y} ||\mathbf{y} - G_{XY}(G_{YX}(\mathbf{y}))||_1 \Big] \ . \qquad (1.19)
\end{aligned}$$

Algorithm 2 CycleGAN training algorithm

Require: $\mathscr{D}_X$ and $\mathscr{D}_Y$ two unpaired datasets, G_{XY} and G_{YX} the mapping networks, D_X and D_Y the discrimination models, m the mini-batch size, λ a hyperparameter

repeat

sample a mini-batch $\{\mathbf{x}_i\}_{i=1}^m$ from $\mathscr{D}_X$

sample a mini-batch $\{\mathbf{y}_i\}_{i=1}^m$ from $\mathscr{D}_Y$

update D_X by stochastic gradient ascent of $\sum_{i=1}^m \big(L_{GAN}(G_{YX}, D_X)\big)$

update D_Y by stochastic gradient ascent of $\sum_{i=1}^m \big(L_{GAN}(G_{XY}, D_Y)\big)$

update G_{XY} by stochastic gradient descent of

$\sum_{i=1}^m \big(L_{GAN}(G_{YX}, D_X)\big) + \lambda\big(||\mathbf{x}_i - G_{YX}(G_{XY}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - G_{XY}(G_{YX}(\mathbf{y}_i))||_1\big)$

update G_{YX} by stochastic gradient descent of

$\sum_{i=1}^m \big(L_{GAN}(G_{XY}, D_Y)\big) + \lambda\big(||\mathbf{x}_i - G_{YX}(G_{XY}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - G_{XY}(G_{YX}(\mathbf{y}_i))||_1\big)$

until a stopping condition is met

The CycleGAN training process then consists in alternatively updating the two discriminators and the two generators via gradient ascent/descent (Algorithm 2). Note that,

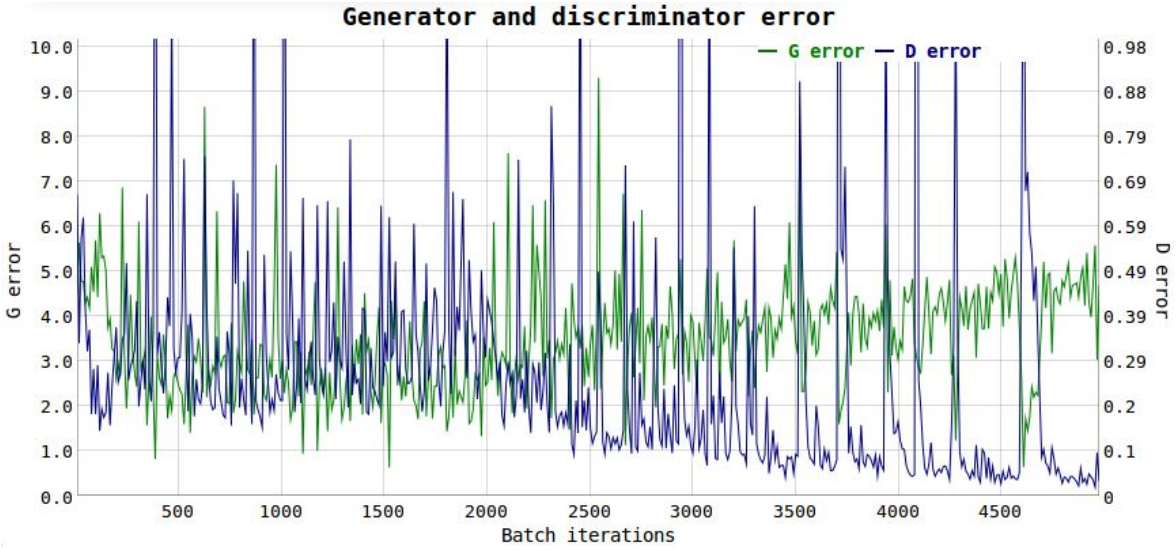


Figure 1.13: An example of real losses obtained during the training of a GAN.

as opposed to the previous approaches, CycleGAN does not use random vectors $\mathbf{z} \in p_Z$ to generate images. The stochastic part of the generation process is instead contained in the initial image $\mathbf{y} \sim p_Y$.

CycleGAN, as well as similar methods relying on cycle-consistency such as **DualGAN** (Yi et al., 2017) or **DiscoGAN** (Kim et al., 2017), have been used in several domains. Among them are medical imaging (Chen et al., 2019), training models with synthetic images obtained with a simulator, for example for robot grasping (Bousmalis et al., 2018)), for image segmentation (Perone et al., 2019) or for converting near-infrared images to color images Sun et al. (2019). These approaches shows that even conversion between different image modalities can be done with cycle-consistent generative models.

1.2.4 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 procedure and the lack of diversity in the generated samples (*mode-collapse*).

Instability

Training GANs consist in solving a minimax problem. While the alternate gradient descent algorithm is a common method for solving such a problem, the alternating updates can cause significant instabilities during the training process. This can result in oscillating values of the GAN objective function which prevents the optimization from converging (Mescheder et al., 2018) and makes it difficult to set a stopping criterion (see Figure 1.13 for an illustration).

The instability of the GAN training has first been conjectured to be caused by the bad quality of the gradients obtained when G generates bad samples, which makes D strongly

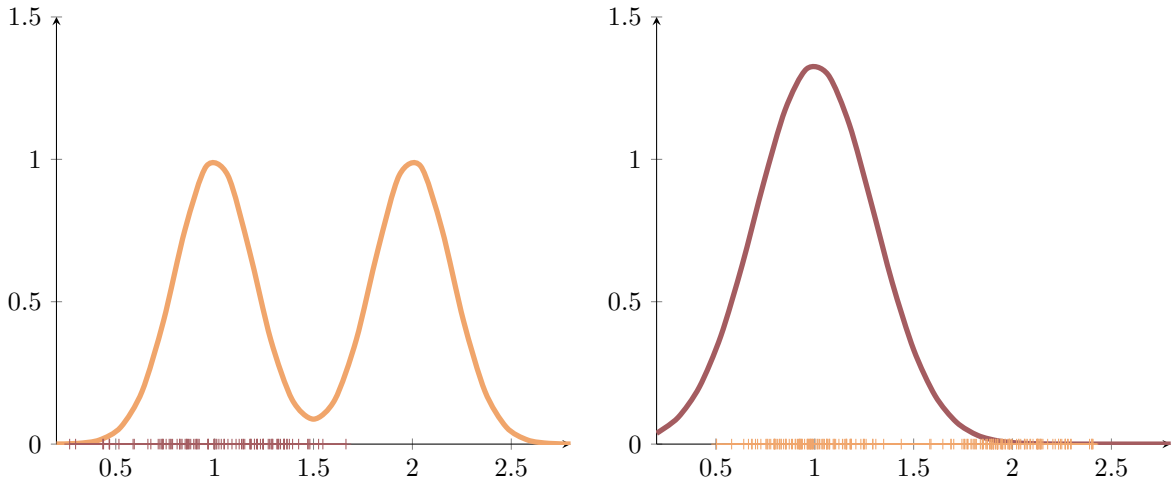


Figure 1.14: Reverse KL (left) and KL (right) divergence between the true orange distribution and the mode-collapsed purple distribution. When computing these distances, the reverse KL is lower, even if a missing mode is clearly visible.

reject these samples and therefore saturating the loss. A proposed solution (Goodfellow et al., 2014) was to slightly change the generator's loss function from $\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z})))$ to

$$Ł_{\mathrm{G}}(\mathrm{D}, \mathrm{G}) = - \mathop{\mathbb{E}}_{\mathbf{z} \sim p_{\mathrm{Z}}} \log(\mathrm{D}(\mathrm{G}(\mathbf{z}))) \ , \tag{1.20}$$

which helps considerably in avoiding failures of the training process (Radford et al., 2015). While this new loss function 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 and Bottou, 2017). More formally,

$$\mathop{\mathbb{E}}_{\mathbf{z} \sim p_{\mathrm{Z}}} \left[\nabla_{\mathrm{G}} \log \mathrm{D}^*(\mathrm{G}(\mathbf{z})) \right] = \nabla_{\mathrm{G}} \left[\mathrm{D}_{\mathrm{KL}} \left(p_{\mathrm{G_X}} \middle|\middle| p_{\mathrm{X}} \right) - 2\mathrm{D}_{\mathrm{JS}} \left(p_{\mathrm{G_X}} \middle|\middle| p_{\mathrm{X}} \right) \right] \ . \tag{1.21}$$

However, albeit an empirical reduction of the instability, this new loss has been proved to not solve the instability problem (Arjovsky and Bottou, 2017). One hypothesis for this issue is the degenerate behavior of the JS and KL divergences when the real distribution and the learned one does not share the same support. Several tricks can be applied to the training process in order to avoid this pitfall (Heusel et al., 2017; Salimans et al., 2016; Sønderby et al., 2017), such as adding noise on the discriminator input or using separate optimizers for the generator and the discriminator. Although more recent approaches (more thoroughly detailed in section 1.3), seem to help alleviate this issue, instability can still be observed in approaches that make full use of them (Brock et al., 2018). Even though several techniques aimed to solve this issue (Arjovsky et al., 2017; Li et al., 2017b; Nowozin et al., 2016), to our knowledge at the time of writing this thesis, there are neither clear consensus on the theoretical causes of this instability nor robust efficient solutions.

Mode collapse

Although the aforementioned change of loss from $\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z})))$ to $-\log(\mathrm{D}(\mathrm{G}(\mathbf{z})))$ can help solving the instability issues, using the reverse KL divergence is conjectured to be

one reason of the loss of statistical diversity. Reasons of the lack of diversity are twofold: the *mode collapse* problem that causes different $\mathbf{z}_1, \mathbf{z}_2$ to be mapped to samples $G(\mathbf{z}_1)$ and $G(\mathbf{z}_2)$ that are very close; and *mode dropping* which leads to missing modes in the generated samples, as only a localized support of the target distribution can actually be mapped to. Indeed, the reverse KL divergence does not penalize "missing" parts of the learned distribution $p_{G_\mathbf{x}}$, which correspond top some points in the support of p_X that have zero (or near-zero) probability on p_{G_X} (see Figure 1.14).

Another conjectured cause is raised by the alternate gradient descent. Indeed, this algorithm does not behave in the same way when formulating the problem as a minimax problem $G^* = \min_G \max_D L_{GAN}$ or maximin problem $G^* = \max_D \min_G L_{GAN}$. Most notably, using alternate gradient descent to solve the maximin problem can push the generator towards mapping every $\mathbf{z}$ to the single most probable $\mathbf{x}$, evaluated by the generator (Goodfellow, 2016).

As for the instability problem, there is, to our knowledge, no clear consensus on the origin mode collapse. However, a trade-off seems to emerge: using the original GAN creates instability which leads to a drop of visual quality, and using the non-saturating variant that stabilizes the training 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 impossible to draw diverse samples. In that case, the generated images $\hat{\mathbf{x}} = G(\mathbf{z})$ can be considered independent from $\mathbf{z}$. This loss of diversity, however, is not a severe issue for conditional tasks that consists in mapping an input to one of many feasible outputs, the most notable of these tasks being domain-transfer (see Section 1.2.3).

1.3 Improvements to Generative Adversarial Networks

Recently, Generative Adversarial Networks have made progress towards generating realistic high definition images (Brock et al., 2018; Karras et al., 2020; Wang et al., 2018a) (see Figure 1.15). These notable successes leverage an overwhelming amount of incremental enhancements and variations of the original GAN (Hindupur, 2017). In this section, a summary of some GAN variants is proposed. We consider two objectives: enhancing the visual quality of the generated samples and ensuring some diversity among the generated samples. We discuss three categories of approaches for this: changing the optimized divergences through alternative loss functions; regularization, normalization and auxiliary tasks and improvements to the training process; and the architecture of the neural networks.

1.3.1 Changing the divergence

As mentioned in Subsection 1.2.4, the original GAN loss (see Equation 1.13) as well as its non-saturating variation (Equation 1.20) show strong limitations, the former causes instability and the latter causes a loss in diversity. As potential solutions, several new loss terms are envisioned.

Figure 1.15: Samples generated with the BigGAN approach (Brock et al., 2018). By combining several recent techniques for improving GANs with very large models and datasets, GANs can generate nearly photo-realistic images.

An alternative to the objective function to the Jensen-Shannon and the reverse Kullback-Leibler divergences is the least-squares loss, which leads to the following discriminator error

$$\mathrm{L}_{\mathrm{LSGAN}}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}}\left[(1-\mathrm{D}(\mathbf{x}))^2\right] + \mathop{\mathbb{E}}_{\mathbf{z}\sim p_{\mathrm{Z}}}\left[(\mathrm{D}(\mathrm{G}(\mathbf{z})))^2\right] \ . \tag{1.22}$$

Such a loss term was considered by Mao et al. (2017) for their **Least-Squares GAN(LSGAN)**. While this loss function follow the same idea as in the original GAN method, LSGAN actually optimizes 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. The reason potentially resides in 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 and Vajda, 2006) defined as

$$\mathrm{D}_f(p_{\mathrm{X}}||q_{\mathrm{X}}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim q_{\mathrm{X}}} f\left(\frac{p_{\mathrm{X}}(\mathbf{x})}{q_{\mathrm{X}}(\mathbf{x})}\right) \ , \tag{1.23}$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex, lower-semi-continuous 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 ($f(u) = (u-1)^2$) divergences. Nowozin et al. (2016) proposed a generalized approach for these divergences as well as several new GAN formulations based on divergences such as the Squared Hellinger distance ($f(u) = (\sqrt{u}-1)^2$) or the Total Variation ($f(u) = \frac{1}{2}|u-1|$). The **hinge loss** variant for GANs (Lim and Ye, 2017), which redefines the loss of the discriminator as

$$\mathrm{L}_{hinge_{\mathrm{D}}}(\mathrm{D},\mathrm{G}) = -\mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}}\left[\min(0,\mathrm{D}(\mathbf{x})-1)\right] - \mathop{\mathbb{E}}_{\mathbf{z}\sim p_{\mathrm{Z}}}\left[\min(0,-\mathrm{D}(\mathrm{G}(\mathbf{z}))-1)\right] \ , \tag{1.24}$$

can also be expressed as the Reverse Kullback-Leibler divergence (Miyato et al., 2018).

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 behaviors, most notably when the distributions p_{X} and $p_{\mathrm{G_X}}$ have no shared support, causing the divergence to be non-continuous and non-differentiable. As a solution

Approach	Divergence
f-divergences	
GAN (Goodfellow et al., 2014)	Jensen-Shannon
NS-GAN (Goodfellow et al., 2014)	Reverse KL - 2·Jensen-Shannon
LSGAN (Mao et al., 2017)	Pearson χ^2
EBGAN* (Zhao et al., 2017)	Total variation
Geometric GAN (Lim and Ye, 2017)	Reverse Kullback-Leibler
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., 2017b)	Maximum Mean Discrepancy
Fisher GAN(Mroueh and 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.

to this issue Arjovsky et al. (2017) proposed the **Wasserstein GAN (WGAN)** by replacing the Jensen-Shannon divergence by the Wasserstein-1 (or Earth-Mover) distance which stems from optimal transport theory (Peyré and Cuturi, 2020). The Wasserstein distance, albeit having many different formulations, can be expressed in its dual form using the Kantorovich-Rubinstein duality (Kantorovich and Akilov, 1982) as

$$\mathrm{W}(p||q) = \sup_{f \in \mathscr{F}} \left[\mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\mathrm{X}}} f(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{x} \sim q_{\mathrm{X}}} f(\mathbf{x}) \right] , \tag{1.25}$$

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

$$\mathrm{L}_{\mathrm{WGAN}}(\mathrm{D}, \mathrm{G}) = \min_{\mathrm{G}} \max_{\mathrm{D} \in \mathscr{F}} \left[\mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\mathrm{X}}} \mathrm{D}(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{z} \sim p_{\mathrm{Z}}} \mathrm{D}(\mathrm{G}(\mathbf{z})) \right] . \tag{1.26}$$

This formulation, however, requires the discriminator to be 1-Lipschitz. This property is enforced done by clipping the weights **W** of the discriminator so that each weight $w_{i,j}$ is in a fixed interval $w_{i,j} \in [-c, c]$, with c being a hyper-parameter. Overall the solution proved to be quite harmful in terms of visual quality by Gulrajani et al. (2017), who proposed the **Wasserstein GAN with Gradient Penalty (WGAN-GP)**. WGAN-GP replaces the clipping by a penalty on the gradient, leading to an additional loss term that pushes the discriminator towards having a gradient with a norm close to 1. Hence, the resulting objective function is formulated as

$$\mathrm{W}_{\mathrm{GP}}(\mathrm{D}, \mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\mathrm{X}}} \mathrm{D}(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{z} \sim q_{\mathbf{z}}} \mathrm{D}(\mathrm{G}(\mathbf{z})) + \lambda \mathop{\mathbb{E}}_{\hat{\mathbf{x}} \sim p_{\hat{\mathrm{X}}}} \left[(||\nabla_{\hat{\mathbf{x}}} \mathrm{D}(\hat{\mathbf{x}})||_2 - 1)^2 \right] , \tag{1.27}$$

where $p_{\hat{\mathrm{X}}}$ is implicitly defined as an uniform distribution on straight lines between pairs of points sampled on p_{X} and $p_{\mathrm{G_X}}$. The forged artificial distribution is used to overcome the intractability of enforcing the gradient norm constraint.

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

$$\mathrm{D}_{\mathscr{F}}(p_{\mathrm{X}}||p_{\mathrm{G_X}}) = \sup_{f\in\mathscr{F}} \left[\mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}} f(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{G_X}}} f(\mathbf{x}) \right] , \tag{1.28}$$

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

Also part of the IPM family of metrics is the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), with the set $\mathscr{F} = \{f : ||f||_{\mathscr{H}} \leq 1\}$, where $\mathscr{H}$ is a Reproducing Kernel Hilbert Space (RKHS) of kernel $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$, thus relying on the choice of the kernel k. MMD was used to formulate different **MMDGAN** (Bińkowski et al., 2018; Dziugaite et al., 2015; Li et al., 2017b) approaches, which train GANs by estimating the MMD with gaussian or quadratic kernels. More recent approaches leverage gradient penalty similarly to WGAN-GP in order to learn the kernel k, which translates into special cases of MMD such as **Energy Distance GAN** (Bellemare et al., 2017; Szekely and Rizzo, 2004) or the so-called **Fisher GAN** (Mroueh and Sercu, 2017).

1.3.2 Improving the GAN framework and architectures

The original GAN approach (Goodfellow et al., 2014) used multi-layer perceptrons as discriminator and generator. While showing good performance on small image datasets such as MNIST (LeCun et al., 1998) or CIFAR10 (Krizhevsky, 2009), it struggled to scale up to higher-dimension images. These relatively simple models were however quickly enhanced with specific architectures and training techniques designed for GANs. They were later combined with more recent neural network architectures such as residual blocks[1] (He et al., 2015) or U-Net[1] encoder-decoder[1] architectures (Ronneberger et al., 2015).

On one hand, the **Laplacian Pyramid GAN (LAPGAN)** (Denton et al., 2015) approach first generates a low-resolution sample $\mathbf{x}_0 = \mathrm{G}_0(\mathbf{z})$, with $\mathbf{z} \sim p_{\mathrm{Z}}$, using a GAN model G_0 and then iteratively upscales it K times using Laplacian Pyramids (Burt and Adelson, 1983). In multi-resolution pyramids, an upscaling operator $u(\mathbf{x}, \mathbf{y})$ combines its input $\mathbf{x}$ with a *difference map* $\mathbf{y}$ to produce a higher resolution image $\mathbf{x}'$. In the LAPGAN approach, these difference maps $\mathbf{y}$ are generated by several conditional generative models $\{\mathrm{G}_1, ..., \mathrm{G}_{\mathrm{K}}\}$ as $\mathbf{y}_n = \mathrm{G}_n(\mathbf{z}, \mathbf{x}_{n-1}), \mathbf{z} \sim p_{\mathrm{Z}}$, then used to create an upscaled image $\mathbf{x}_n = u(\mathbf{x}_{n-1}, \mathbf{y}_n)$ which can in turn be used to generate a difference map $\mathbf{y}_{n+1} = \mathrm{G}_{n+1}(\mathbf{z}, \mathbf{x}_n)$. Each generator G_n is trained as a GAN, in pair with its own discriminator D_n, which makes the approach computationally expensive.

On the other hand, the **Deep Convolutional GAN (DCGAN)** (Radford et al., 2015) approach replaced the discriminator by a fully convolutional network (Springenberg et al., 2015) with strided convolutions and introduced deconvolutional[1] (or transposed convolutional) layers in the generator. It also introduced dropout (Srivastava et al., 2014) and Batch Normalization[1] (Ioffe et al., 2015), and used both ReLU (Nair and Hinton, 2010) and Leaky ReLU (Maas et al., 2013) as activation functions. DCGAN showed much better

[1] see Glossary, Appendix B

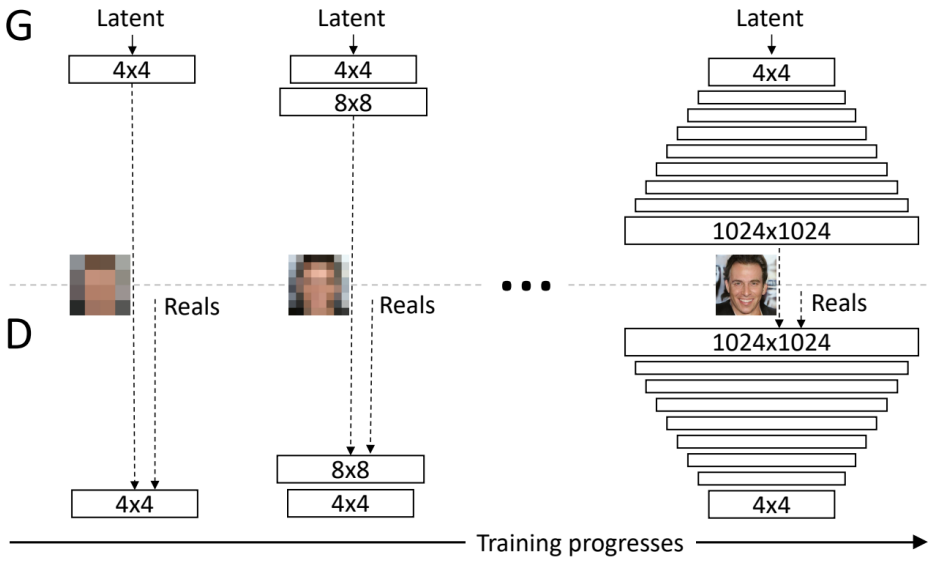


Figure 1.16: Progressive growing of Generative Adversarial Networks. Convolutional layers are added throughout the learning process, each time doubling the dimension of the generated images.

results than the original GAN and LAPGAN, while being more computationally efficient than the latter, and thus became a standard baseline for image generation.

This approach was extended with tricks for mitigating its instability (Salimans et al., 2016), such as adding Batch Normalization[1] (Ioffe et al., 2015), smoothing the 0/1 label used for training the discriminator or adding noise to the discriminator's input (Sønderby et al., 2017). They effectively helped stabilizing the training process. However, the DCGAN approach remains limited in both the visual quality of the generated samples and in its ability to generate high-dimension images.

Progressive GAN (Karras et al., 2017) first enabled high-dimensional image generation with GANs by progressively adding convolutional layers in the generator and the discriminator during training. Thus, the training starts with low-dimension images and progressively increase the dimension of the images throughout the learning process, which is illustrated in Figure 1.16. This approach yielded the first high-quality, high-definition images generated with GANs.

Self-Attention GAN (SAGAN) (Zhang et al., 2018a) implements attention-driven[1] mechanisms in GANs to model *long-range dependencies*. In most of the previous approaches, such as LAPGAN, DCGAN or Progressive GAN, the images are successively upscaled using convolutional layers. This causes issues in high-dimension, as some parts of the generated image could be inconsistent with each other due to the spatially local nature of convolutions. Self-attention propose to use non-local modeling as a solution to this issue. In the SAGAN approach, this is done by computing self-attention feature maps (see Figure 1.17) that are then added to the convolution feature maps.

[1] see Glossary, Appendix B

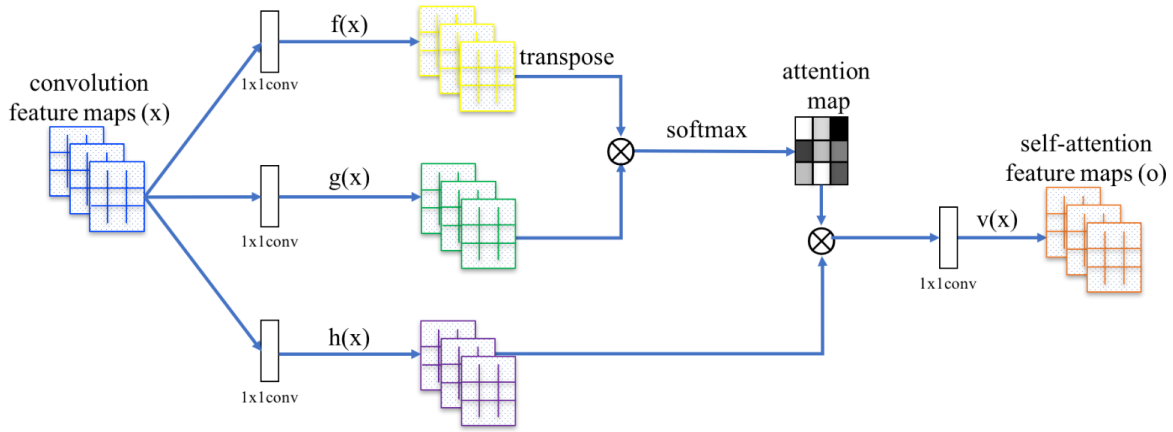


Figure 1.17: Self-attention module for the SAGAN approach. The $\otimes$ denotes matrix multiplication. Figure by Zhang et al. (2018a)

1.3.3 Augmenting the objective

Another common design approach for both stabilizing the training process and increasing diversity among the generated samples is to extend the GAN objective with additional conditioning costs.

Training a GAN in a supervised way with conditioning approaches such as Conditional GAN (Mirza and Osindero, 2014), Auxiliary Classifier GAN (Odena et al., 2016) or Triple-GAN (Li et al., 2017a) (see Section 1.2.2) can enhance both the visual quality as well as the diversity among the generated samples. Indeed, GANs seem to take profit from the supervision added by the use of labels during the training process, even though it requires labeled data.

Another category of approaches aim to include an inference process into the GAN framework to retrieve the input noise from a generated sample, that is obtaining a function I such that $\mathrm{I}(\mathrm{G}(\mathbf{z})) = \mathbf{z}, \mathbf{z} \in p_Z$. **Adversarially Learned Inference (ALI)** (Dumoulin et al., 2016) and **Bidirectional GAN (BiGAN)** (Donahue et al., 2017) are two similar approaches that aim to train a neural network $\mathrm{I} : \mathbb{X} \rightarrow \mathbb{Z}$ as an inference mechanism. By providing the discriminator with either the input noise $\mathbf{z}$ or the input noise $\mathrm{I}(\mathbf{x})$ inferred from the real sample $\mathbf{x}$ (see Figure 1.18), the networks G, D and the inference model I are trained simultaneously by solving the problem

$$\min_{\mathrm{G,I}} \max_{\mathrm{D}} \mathbb{E}_{\mathbf{x} \sim p_X} \Big[\log(\mathrm{D}(\mathbf{x}, \mathrm{I}(\mathbf{x}))) \Big] + \mathbb{E}_{\mathbf{z} \sim p_Z} \Big[\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z}), \mathbf{z})) \Big] \ . \tag{1.29}$$

The main interest of these approaches is that they increase the diversity among the generated samples, but the trained inference model serve several purposes. For example, since the optimal generator and inference models G^* and I^* are inverse of each others, as $\mathrm{I}^*(\mathrm{G}^*(\mathbf{z})) = \mathbf{z}, \mathbf{z} \sim p_Z$ and $\mathrm{G}^*(\mathrm{I}^*(\mathbf{x})) = \mathbf{x}, \mathbf{x} \sim p_X$, this allow for approximating the likelihood of a sample using the *change of variable formula* (see Equation 1.9) by considering $\mathrm{I}^* = \mathrm{G}^{*^{-1}}$.

Structured GAN (Deng et al., 2017) combines the supervised label-based conditioning with the inference-based approaches by adding both an inference model I as well as a classifier C that are trained jointly with the GAN. To do so, Deng et al. (2017) introduce

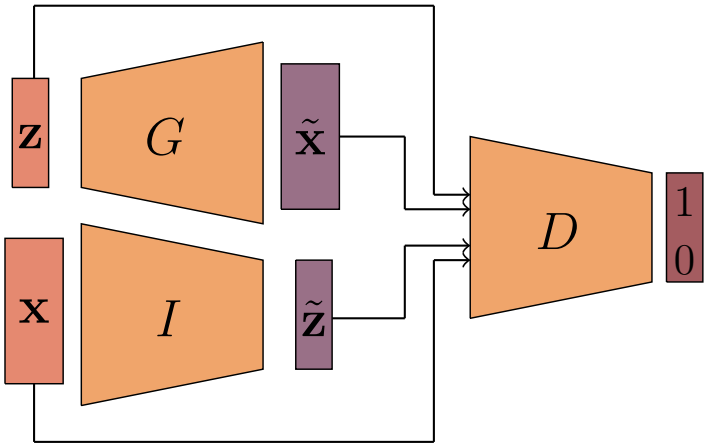


Figure 1.18: Adversarially Learned Inference / Bidirectional GAN

two separate discriminators $D_{\mathbf{y}}$ and $D_{\mathbf{z}}$ that take respectively an image and its label, or an image and its corresponding noise in the latent space. The Structured GAN framework then consists in solving

$$\min_{G,I,C} \max_{D} L_{\mathbf{x},\mathbf{y}}(G,D,C) + L_{\mathbf{x},\mathbf{z}}(G,D,I) + L_{\mathbf{y}}(G,D,C) + L_{\mathbf{z}}(G,D,I) \quad , \tag{1.30}$$

where $L_{\mathbf{x},\mathbf{y}}$ and $L_{\mathbf{x},\mathbf{z}}$ are adversarial losses, $L_{\mathbf{y}}$ is a classification loss and $L_{\mathbf{z}}$ is a disentanglement loss, that is a loss that aims to ensure that an inferred noise $I(G(\mathbf{y},\mathbf{z})), \mathbf{y} \sim p_Y, \mathbf{z} \sim p_Z$ is independent from the class label $\mathbf{y}$. These losses take the forms

$$L_{\mathbf{x},\mathbf{y}}(G,D,C) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\Big[\log D_{\mathbf{y}}(\mathbf{x}, C(\mathbf{x}))\Big] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}}\Big[\log(D_{\mathbf{y}}(G(\mathbf{y},\mathbf{z}),\mathbf{y}))\Big] \tag{1.31}$$

$$L_{\mathbf{x},\mathbf{z}}(G,D,I) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\Big[\log D_{\mathbf{z}}(\mathbf{x}, I(\mathbf{x}))\Big] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}}\Big[\log(D_{\mathbf{z}}(G(\mathbf{y},\mathbf{z}),\mathbf{z}))\Big]$$

$$L_{\mathbf{y}}(G,D,C) = -\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim p_{X,Y}}\Big[\sum_{i=1}^{n} \mathbf{y}_i, C(\mathbf{x})_i\Big] - \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_Y \\ \mathbf{z}\sim p_Z}}\Big[\sum_{i=1}^{n} \mathbf{y}_i C(G(\mathbf{z}))_i\Big]$$

$$L_{\mathbf{z}}(G,D,I) = -\mathop{\mathbb{E}}_{\substack{\mathbf{y}_1,\mathbf{y}_2\sim p_Y \\ \mathbf{z}\sim p_Z}}\Big[||I(G(\mathbf{y}_1,\mathbf{z})) - I(G(\mathbf{y}_2,\mathbf{z}))||_2^2\Big] \quad . \tag{1.32}$$

Another method for conditioning GANs is **Packing GAN (PacGAN)** (Lin et al., 2018). This technique, designed to help with the diversity issues of GANs, consists in using sets of samples as input to the discriminator (see Figure 1.19) instead of single ones

$$L_{\mathrm{P}ac\mathrm{GAN}} = \mathop{\mathbb{E}}_{(\mathbf{x}_1,\ldots,\mathbf{x}_l)\sim p_X}\Big[\log(D((\mathbf{x}_1,\ldots,\mathbf{x}_l)))\Big] + \mathop{\mathbb{E}}_{(\mathbf{z}_1,\ldots,\mathbf{z}_l)}\Big[\log(1 - D((G(\mathbf{z}_1),\ldots,G(\mathbf{z}_l))))\Big] \quad . \tag{1.33}$$

Since the real images $\mathbf{x}_1,\ldots,\mathbf{x}_l$ should always be different from each other, the task of the discriminator becomes very easy if the generator collapsed, that is if the generated samples $G(\mathbf{z}_1),\ldots,G(\mathbf{z}_l)$ are similar. This approach, albeit computationally more expensive than the classical GAN framework because of the l generated samples needed to train the discriminator, is very effective in preventing mode collapse. In practice, using two samples is usually enough to prevent the model from mode collapsing.

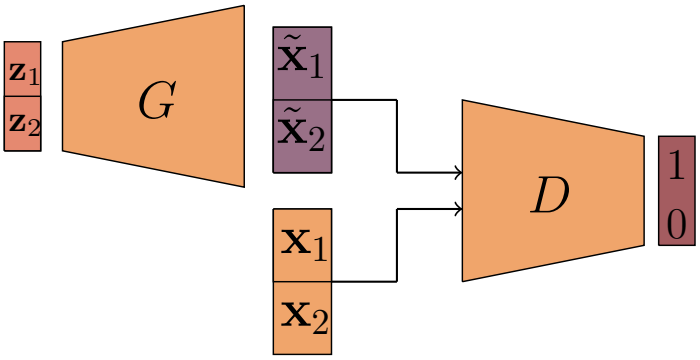


Figure 1.19: The PacGAN approach with two samples

Conversely to these conditioning approaches, several works showed that constraining the discriminator to be Lipschitz continuous (Arjovsky and Bottou, 2017; Arjovsky et al., 2017; Qi, 2018) improved the stability of the training process. **Spectral Normalization** (Miyato et al., 2018) show that normalizing the weights of the discriminator using the spectral norm of its weight matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$, as

$$\mathbf{W}' = \frac{\mathbf{W}}{\sigma(\mathbf{W})} \ , \tag{1.34}$$

with $\sigma(\mathbf{W})$ being the spectral norm[1] of $\mathbf{W}$, is sufficient to ensure that the discriminator is 1-Lipschitz. Since computing the spectral norm for each training step is computationally expensive, the authors propose to use the *power method* (Golub and van der Vorst, 2000) to compute a fast approximation of the spectral norm. This is done by sampling a random vector $\mathbf{u} \in \mathbb{R}^n$ and by iteratively computing

$$\mathbf{u} \leftarrow \frac{\mathbf{W}^\top \mathbf{W}\mathbf{u}}{||\mathbf{W}^\top \mathbf{W}\mathbf{u}||_2} \ , \tag{1.35}$$

from which the spectral norm is estimated as

$$\sigma(\mathbf{W}) = \sqrt{\mathbf{u}^\top (\mathrm{W}^\top \mathrm{W})\mathbf{u}} \ . \tag{1.36}$$

This technique showed excellent results in stabilizing the GAN training process, notably allowing for learning a conditional generative model that was able to generate samples from the 1000 classes of the ImageNet dataset (Deng et al., 2009).

1.4 A note on the evaluation of GANs

Unlike discriminative models, evaluating and comparing generative models is a non-trivial task. Two approaches can be envisioned: evaluating the *intrinsic* quality of generated samples with ad-hoc criteria or directly evaluating the likelihood of the generated samples. However, unlike VAEs and flow-based models, GANs offer no explicit way to

[1] The spectral norm $\sigma(\mathbf{W})$ of a matrix $\mathbf{W}$ is its maximum singular value

evaluate or approximate the likelihood of the generated samples. Thus, a significant part of the GAN literature resorts 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; the Inception Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017) were proposed, which both make use of a pre-trained Inception v3 model (Szegedy et al., 2016), a deep classifier trained on the ImageNet dataset (Deng et al., 2009).

Inception Score (IS)) (Salimans et al., 2016) is based on the evaluation of the entropy of the labels $\mathbf{y}$ predicted by the Inception classifier of generated data. High-fidelity samples should be easier to classify and therefore have a conditional label distribution $p_{\mathrm{G_{Y|X}}}$ with low entropy. In addition to the high quality, the samples should be diverse, therefore the marginal distribution

$$p_{\mathrm{G_Y}} = \int_{\mathcal{Z}} p_{\mathrm{G_{Y|X=G(z)}}}\, d\mathbf{z} \tag{1.37}$$

should have a high entropy. By combining these two requirements, the IS is formulated as

$$\mathrm{IS}(\mathbf{y}) = \exp\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{G_X}}} \mathrm{D_{KL}}\Big(p_{\mathrm{G_{Y|X}}} \Big|\Big| p_{\mathrm{G_Y}} \Big) \right] \ . \tag{1.38}$$

Although it has been widely used, IS has shown major issues (Barratt and 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

$$\mathrm{FID} = ||\mu – \mu_{\mathrm{G}}||^2 + \mathrm{T}r(\Sigma + \Sigma_{\mathrm{G}} – 2\sqrt{\Sigma \times \Sigma_{\mathrm{G}}}) \ , \tag{1.39}$$

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 (Barratt and Sharma, 2018) and has been either completing or replacing the use of IS in recent works.

However, while these two metrics are considered to be the gold standard for evaluating GANs, their reliance on the pre-trained Inception model may be an issue. Indeed, they behave well when used to compare models learned on natural images datasets such as ImageNet, but they cannot directly extend to other datasets such as medical images or 3D images. A solution to consider is the training of another classifier network on a more adapted dataset, provided labeled data is available.

For the sake of completeness, we can also refer to some other notable (albeit less commonly used) 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.

Figure 1.20: Evolution of the visual quality of generated images from 2014 to 2020, using the CelebA(-HQ) (Liu et al., 2015) datasets. Left: DCGAN (Radford et al., 2015), Center: ProgGAN (Karras et al., 2017), Right: StyleGAN2 (Karras et al., 2020)

Finally it is to note that for conditioned models, computing the aforementioned metrics does not inform on the quality of the conditioning. However, since the conditioning usually requires either labels or prior information, they can usually be used to evaluate conditioned models, 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.

1.5 Conclusion

In this chapter, we introduce the Generative Adversarial Network (GANs) framework, which consists in a pair of neural networks, namely the generator and discriminator, that conjointly learn to model a complex data distribution by minimizing a divergence between the real data distribution and this modeled one. We present some of the techniques for conditioning modeling with GANs, from simply providing the GAN models with labels to adding a supervised auxiliary task to the training process. We present domain-transfer approaches with GANs, that is transferring an image from a data domain to another (for example images of infra-red to color images), most notably the cycle-consistent approaches that do not rely on hard-to-obtain paired data.

We expose some of the limitations of the GAN framework, most notably the stability issues of the training process, the lack of diversity among the generated samples and the difficulty to produce high-quality, high-dimension images. We review recent research works that aim to solve these issues, among them variants of the loss functions, changes to the architectures and advanced conditioning techniques. We also expose the difficulties of evaluating generative models and present common solutions for assessing the visual quality of generated samples.

Although we restrict ourselves to image generation, it is worth noting that the application range of GANs covers text (Guo et al., 2018), video (Clark et al., 2020) or sound (Engel et al., 2018) generation. These application domains are beyond the scope of the thesis.

Chapter 2

Image reconstruction as an auxiliary task to generative modeling

Chapter abstract

While the Conditional GAN approach (Mirza and 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 an 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. Such a 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 propose 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 balance visual quality and 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 standard 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: *ESANN 2019 - Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 25–30

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (Nov. 27, 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing* 416, pp. 218–230

Contents

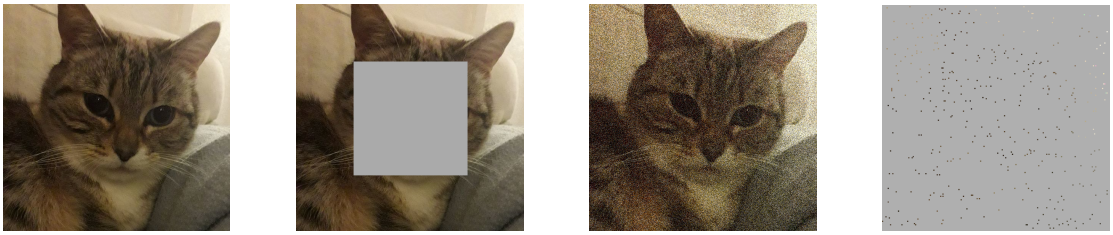
2.1 Introduction

Conditional GANs (Mirza and Osindero, 2014) are powerful methods for learning conditional generative models. By simply providing a label to both the generation and discrimination networks, CGAN is able to solve problems such as class-conditioned image generation (Mirza and 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 $\mathbf{y}$. 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. Our motivation stems from 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, 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. Hence the model we seek aims to match the distribution of the real images conditioned on a highly scarce constraint map. To explicitly enforce the generated images to honor 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, especially suited for texture generation. 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 (Laloy et al., 2018; Strebelle, 2002). 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



(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 the real sample depicted in sub-figure (2.1a).

discriminator during the training process instead of single images, for both the generated images and the images from the dataset (see Section 1.3.3). 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. Section 2.2 introduces the problem of image reconstruction. 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 and Tao, 2005) approaches and conditional generation methods. Section 2.4 introduces our approach for image reconstruction, and proposes some theoretical insight. In Section 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 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 (Figure 2.1d).

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) (Figure 2.1b) or image denoising (Goyal et al., 2020) (Figure 2.1c) which consists

in retrieving missing or altered parts of an image. 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, image denoising (Figure 2.1c) aims to remove alterations induced by some noise, which can be due to imperfections in the acquisition procedure or natural degradation, which finds applications such as raw image denoising in cameras (Kim, 2014) or medical image denoising (Gondara, 2016).

Image reconstruction (Figure 2.1d) however differs from these problems as most part of the original image is unavailable. Thus, in comparison to inpainting or denoising in which the altered parts of 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 observations. This can be done by leveraging on prior knowledge to train a generative model, while ensuring that the resulting image is coherent with the pixels given as input.

Before delving into the details, we introduce the notations related to the problem. Note that we use the matrix and vector formulations interchangeably, as most of the computation remains similar regardless of the number of dimensions. We denote by $\mathbf{X}$ a random variable and $\mathbf{x} \in \mathbb{R}^{n \times p \times c}$ its realization. Let $p_{\mathbf{X}}$ be the distribution measure of $\mathbf{X}$ over $\mathbb{X}$. Similarly $p_{\mathbf{X}|\mathbf{Y}}$ represents the distribution of $\mathbf{X}$ conditioned on the random variable Y, while $p_{\mathbf{X},\mathbf{Y}}$ 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

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

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

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 $n \times p$ image with only a few available pixels (less than 1% of $n \times p$). We will encode the spatial location of these pixels using a corresponding binary mask $\mathbf{M}_{\mathbf{y}} \in \{0,1\}^{n \times p}$.

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 $p_{\mathbf{X}}$ 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 $p_{\mathbf{X}}(\hat{\mathbf{x}})$ for a given constraint map $\mathbf{y}$. In other words, the problem consists in retrieving $\mathbf{x}$ such that $\mathbf{y} = \mathbf{M}_{\mathbf{y}} \odot \mathbf{x}$ and $\mathbf{x}$ is issued from to the data distribution $p_{\mathbf{X}}$. More formally, we aim at finding

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p_X(\mathbf{x}) \ \text{ subject to } \ \|\mathbf{y} - \mathbf{M}_{\mathbf{y}} \odot \mathbf{x}\|_2^2 \leq \delta \tag{2.2}$$

where $\odot$ stands for the Hadamard (or point-wise) product[1], $\|\mathbf{N}\|_2^2$ represents the squared

[1]Note that this expression can be formulated with the Hadamard product instead of a matrix product in Equation (2.1), since $\text{vect}(\mathbf{M}_{\mathbf{y}} \odot \mathbf{x}) = \text{Tr}(\text{Diag}(\text{vect}(\mathbf{M}_{\mathbf{y}}))\text{vect}(\mathbf{x}))$. $\text{vect}(\cdot)$ is the vectorisation operator that consists in stacking the pixels, with $\text{vect}(\mathbf{y}) \in \mathbb{R}^{n.m.c}$ for $\mathbf{y} \in \mathbb{R}^{n \times m \times c}$, $\text{Tr}(\cdot)$ is the trace of a matrix and $\text{Diag}(\cdot)$ is an operator which transforms a vector $\mathbf{x}$ into a diagonal matrix with $\mathbf{x}$ as its diagonal entries.

Frobenius norm of matrix $\mathbf{N}$ that is the sum of its squared entries, δ is a small constant and $\mathbf{M_y}$ the mask, a 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 weaknesses 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 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 $\mathbf{A}$, 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 problem (2.1) 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 $\mathbf{A}$, which states that for two samples $\mathbf{x}_1, \mathbf{x}_2 \sim p_{\mathrm{X}}$,

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

where α is a small constant. This states that distance between two samples is preserved when altered by $\mathbf{A}$. Candes and Tao (2005) used this property to show that if the matrix $\mathbf{A}$ enforces the RIP, samples $\hat{\mathbf{x}}$ retrieved by compressed sensing will follow with a high probability the real data distribution $p_{\mathbf{X}}$. Examples of matrices that enforce the RIP are random Gaussian or Fourier matrices (Candes et al., 2006; Candes and Tao, 2006). Under the RIP setting, the sparse signal $\mathbf{x}$ can be retrieved by solving for

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ subject to } \ \mathbf{A}\mathbf{x} = \mathbf{y} \ , \tag{2.4}$$

when the measurement process is assumed to be noiseless. In practice, the measurements processes are nearly always noisy, thus this problem must be reformulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ subject to } \ \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 \leq \delta \ , \tag{2.5}$$

with δ a small constant. This assumes that the reconstruction is the best possible while taking noise into account. In either cases, solving this problem is NP-hard due to the ℓ_0

norm, however a convex relaxation seeks the minimal ℓ_1-norm solution is also the sparsest solution (Donoho, 2006b). Thus we can instead recover the sparse signal $\mathbf{x}$ as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \leq \delta \ . \tag{2.6}$$

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 for natural 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 or quadratic programming, which can be solved in polynomial time, it is still computationally expensive. Finally, the third issue is that the measurement matrix $\mathbf{A}$ does not necessarily respect the RIP. This is detrimental since the RIP guarantees the coherency of the reconstructed sample. However, verifying that the matrix $\mathbf{A}$ respects the RIP is NP-hard in general. While several approaches for image compression use techniques for generating random matrices that have a high probability of respecting the RIP (Rauhut, 2010; Rudelson and Vershynin, 2008), there are no guarantees in the case when $\mathbf{A}$ is fixed, such as image reconstruction.

Compressed sensing with sparse coding

When aiming to recover high-dimensional signals such as natural images, the assumption of sparsity on $\mathbf{x}$ is unrealistic. This requirement can however be replaced by the more generic approach of considering sparsity in another basis. Let $\mathbf{B}$ be a basis such that $\mathbf{x} = \mathbf{B}\mathbf{s}$ and $\mathbf{s}$ a sparse vector. Thus, the problem becomes

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{B}\mathbf{s}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{B}\mathbf{s} - \mathbf{y}\|_2^2 \leq \delta \ . \tag{2.7}$$

Signal is then recovered as $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{s}}$. By either carefully selecting $\mathbf{B}$, such as Fourier or wavelet basis (Mallat, 2008), **compressed sensing with sparse basis** (Shaobing and Donoho, 1994) is much more robust and provides good results in real-world situations, for example in medical imaging (Lustig et al., 2008), image acquisition (Duarte et al., 2008; Kolev, 2011).

Another category of approaches is **dictionary learning for compressed sensing** (Tošić and Frossard, 2011), learns the basis $\hat{\mathbf{B}}$ as a dictionary using a dataset of samples $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_K\}, \mathbf{x}_i \in \mathbb{R}^{n \times m \times c}$ such that $\mathbf{x}_i = \hat{\mathbf{B}}\mathbf{s}_i$, where $\mathbf{s}_i$ is sparse. This can be formulated as solving for

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

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 relax the ℓ_1-norm solution and solve

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

This learned basis $\hat{\mathbf{B}}$ can then be used as a basis for compressed sensing. Several algorithms exist for solving this problem, usually by iteratively updating the basis $\hat{\mathbf{B}}$ and

the representations $\mathbf{s}_i$ alternatively. Examples of such algorithms are LASSO (Tibshirani, 1996), basis pursuit (Donoho, 2006a) 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 $\mathbf{A}$ 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 may not be necessarily close to the real data distribution p_{X}.

To overcome these problems, **Compressed Sensing with Meta-Learning** (Wu et al., 2019) extends 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 $\mathrm{G}(\mathbf{z})$ in an unconstrained way and optimizing in the latent space $\mathbb{Z}$ of the generative model G by minimizing $\|\mathbf{A}\mathrm{G}(\mathbf{z})-\mathbf{y}\|_2^2$, this method finds an image that, when altered as $\hat{\mathbf{y}} = \mathbf{A}\mathrm{G}(\mathbf{z})$ 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 $\mathbf{A}$. The overall problem induced by this approach is formulated as

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

The method tries to minimize the difference between the distances among samples (generated or real) and the distances among samples altered by $\mathbf{A}$. Solving this problem pushes the generator towards producing samples on which the RIP of $\mathbf{A}$ 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 $\|\mathbf{y}-\mathbf{A}\mathrm{G}(\mathbf{z})\|_2^2$, starting from a random $\mathbf{z}\sim p_{\mathrm{Z}}$.

Deep Compressed Sensing (Wu et al., 2019) extends even further compressed sensing by replacing the (usually random) measurement matrix $\mathbf{A}$ 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. The induced

(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 uses 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).

optimization problem is therefore

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

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

In the same fashion as the objective in Equation (2.10), solving problem (2.12) pushes f_θ towards respecting the RIP. Wu et al. (2019) showed that optimizing these two criteria trains both the generator G and the measurement function f_θ, thus replacing the discriminator of the 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 set 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 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 $p_{\mathrm{X}|\mathrm{Y}}$ is equivalent to taking a sample or

image $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{y}, \mathbf{z})$, with $\mathbf{z} \sim p_{\mathrm{Z}}$, using the generative model G solution to the problem

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{\mathrm{X,Y}}} [\log \mathrm{D}(\mathbf{x}, \mathbf{y})] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}} \\ \mathbf{z}\sim p_{\mathrm{Z}}}} [1 - \log \mathrm{D}(\mathrm{G}(\mathbf{y}, \mathbf{z}), \mathbf{y})] \ , \tag{2.13}$$

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 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

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{\mathrm{X,Y}}} [\log \mathrm{D}(\mathbf{x}, \mathbf{y})] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}} \\ \mathbf{z}\sim p_{\mathrm{Z}}}} [1 - \log \mathrm{D}(\mathrm{G}(\mathbf{y}, \mathbf{z}), \mathbf{y})] + \|\mathbf{M}_{\mathbf{y}} \odot \mathrm{G}(\mathbf{y}, \mathbf{z}) - \mathbf{y}\|_{\mathrm{F}}^2 \ . \tag{2.14}$$

These approaches are often extended with techniques such as using multiple discriminators (Armanious et al., 2019; Yu et al., 2018), extending the training with extra information and features (Armanious et al., 2019) as for medical imaging modalities, or using style losses (Guo et al., 2019) (See Section 1.3.3). However, several of these CGAN-based inpainting methods (Demir and 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 from very sparse and unstructured observations $\mathbf{y}$.

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 $\mathbf{y} \sim p_{\mathrm{Y}}$. This problem is different from the one we tackle, since ours supposes that a dataset of unaltered samples $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_s\}, \mathbf{x}_i \in \mathbb{R}^{n\times m\times c}$ 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 $\mathscr{Y} = \{\mathbf{y}_1, ...\mathbf{y}_t\}, \mathbf{y}_i \in \mathbb{R}^{n\times m\times c}$. 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 this purpose, Ambient GAN considers lossy measurements such as a blurred image, an image 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 measurement matrix $\mathbf{A}$

$$\mathbf{y} = f_\theta(\mathbf{x}) \ . \tag{2.15}$$

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

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D}, \mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{Y}}} \Big[\log(\mathrm{D}(\mathbf{y})) \Big] + \mathop{\mathbb{E}}_{\substack{\mathbf{z}\sim p_{\mathrm{Z}} \\ \theta\sim p_\theta}} \Big[\log(1 - \mathrm{D}(f_\theta(\mathrm{G}(\mathbf{z})))) \Big] \ . \tag{2.16}$$

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 $\mathrm{G}(\mathbf{z})$ on which the alteration

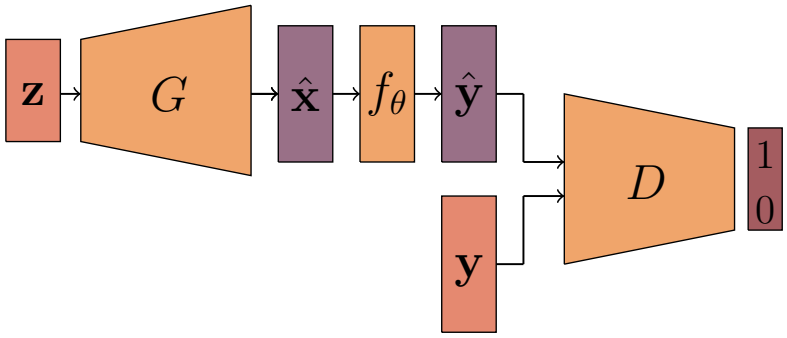


Figure 2.3: Overview of the **Ambient GAN** framework for learning generative models using altered samples only.

function f_θ is applied. Thus, the Ambient GAN generator network G actually learns to generate samples $\hat{\mathbf{x}} = \mathrm{G}(\mathbf{z})$ 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 Ambient GAN approach by adding a conditioning to the model, which allows for the reconstruction of an image $\mathbf{x}$ from an altered image $\mathbf{y} \sim p_{\mathrm{Y}}$, 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 $\tilde{\mathbf{x}} = \mathrm{G}(\mathbf{y})$ and applying the alteration function f_θ to the generated image $\tilde{\mathbf{x}}$ to get $\tilde{\mathbf{y}} = f_\theta(\mathrm{G}(\mathbf{y}))$, then re-generating an image as $\hat{\mathbf{x}} = \mathrm{G}(f_\theta(\mathrm{G}(\mathbf{y})))$ and finally re-applying f_θ 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

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) = \mathop{\mathbb{E}}_{\mathbf{y}\sim p_{\mathrm{Y}}} \Big[\log(\mathrm{D}(\mathbf{y})) \Big] + \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}} \\ \theta\sim p_\theta}} \Big[\log(1-\mathrm{D}(\hat{\mathbf{y}})) \Big] + \|\hat{\mathbf{y}} - \tilde{\mathbf{y}}\|_{\mathrm{F}}^2 \ . \tag{2.17}$$

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 a 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 surrounding 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 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

$$\min_{\mathbf{z}} \|\mathbf{A}\mathrm{G}(\mathbf{z}) - \mathbf{y}\|_2^2 + \lambda \log(1-\mathrm{D}(\mathrm{G}(\mathbf{z}))) \ , \tag{2.18}$$

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.

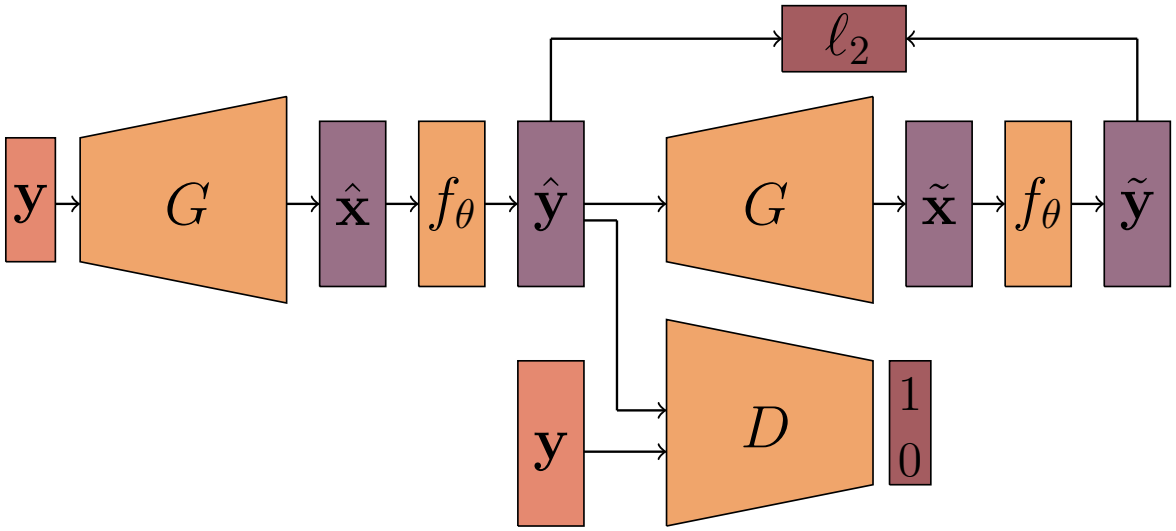


Figure 2.4: Overview of the **Unsupervised Image Reconstruction** framework for learning image reconstruction models using altered samples only.

A summary of the main features of the presented related work is provided in Table 2.1. These features are the need for a dataset of samples, with only the need for altered samples for the Ambient GAN and UNIR approaches; the need for solving a reconstruction problem for each generated image, which is computationally expensive; the ability to sample multiple images on the conditional distribution $p_{\mathrm{Y|X}}$ for a given $\mathbf{y}$, and the different ways to control the enforcement of the constraints.

Approach	Dataset free	One-step reconstruction	Sampling mechanism	Constraints enforcement
Compressed sensing-based				
Compressed sensing (Candes and 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 and 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 advantages and limitations of 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 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 $p_{X|Y}$ to reconstruct the image by sampling on this distribution. Among these approaches are CGAN-based methods, Ambient GAN 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 require a dataset to train on and do not provide any means of controlling the trade-off between the visual quality and the respect of the constraints.

As a main contribution in 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. This method also provides a control parameter λ that balances the visual quality and the respect of the constraints.

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 replacing the implicit conditioning of the CGAN with a reconstruction loss is an approach that naturally emerges from the denoising formulation Equation (2.15). This provides a rationale for the use of these losses in the similar aforementioned approaches (Section 2.3.2).

Such a generative model 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 images, 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 consists in recovering $\mathbf{x}$, assuming the constraint map $\mathbf{y}$ is resulting from applying $\mathbf{M}_\mathbf{y}$ on the image $\mathbf{x}$, as

$$\mathbf{y} = \mathbf{M}_\mathbf{y} \odot \mathbf{x} \ . \tag{2.19}$$

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

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \log p_{\mathrm{X|Y}}(\mathbf{x}|\mathbf{y}) + \log p_{\mathrm{Y}}(\mathbf{y}) \ , \tag{2.20}$$

$$= \arg\max_{\mathbf{x}} \log p_{\mathrm{Y|X}}(\mathbf{y}|\mathbf{x}) + \log p_{\mathrm{X}}(\mathbf{x}) \ . \tag{2.21}$$

$p_{\mathrm{Y|X}}(\mathbf{y}|\mathbf{x})$ is the likelihood that the constrained pixels $\mathbf{y}$ are issued from image $\mathbf{x}$ while $p_{\mathrm{X|Y}}(\mathbf{x}|\mathbf{y})$ is the likelihood of an image knowing the constrained pixels $\mathbf{y}$. p_{X} and p_{Y} represent the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$. Thus, we can introduce a conditional generative model $\mathrm{G} : (\mathbb{Y}, \mathbb{Z}) \to \mathbb{X}$ to replace the conditional distribution $p_{\mathrm{X|Y}}$. This changes the original problem (Equation (2.19)) to

$$\mathbf{y} = \mathbf{M_y} \odot \mathrm{G}(\mathbf{y}, \mathbf{z}) + \epsilon \ , \tag{2.22}$$

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

$$\max_{\mathrm{G}} \mathop{\mathbb{E}}_{\substack{\mathbf{y}\sim p_{\mathrm{Y}} \\ \mathbf{z}\sim p_{\mathrm{Z}}}} \log p_{\mathrm{Y|X}}(\mathbf{y}|\mathrm{G}(\mathbf{y}, \mathbf{z})) + \log p_{\mathrm{X}}(\mathrm{G}(\mathbf{y}, \mathbf{z})) \ . \tag{2.23}$$

The first term in Problem (2.23) 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 (2.22) as $\mathrm{vect}(\mathbf{y}) = \mathrm{vect}(\mathbf{M_y} \odot \mathbf{x}) + \mathrm{vect}(\epsilon)$ where $\mathrm{vect}(\cdot)$ is the vectorisation operator that consists in stacking the pixels, with $\mathrm{vect}(\mathbf{y}) \in \mathbb{R}^{n.m.c}$ for $\mathbf{y} \in \mathbb{R}^{n \times m \times c}$. Therefore, assuming $\mathrm{vect}(\epsilon)$ is i.i.d and follows a Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathrm{I})$, the conditional likelihood of $\mathbf{y}$ knowing $\mathbf{x}$ reads as

$$\log p_{\mathrm{Y|X}}(\mathbf{y}, \mathrm{G}(\mathbf{y}|\mathbf{z})) \propto -\left\| \mathrm{vect}(\mathbf{y}) - \mathrm{vect}(\mathbf{M_y} \odot \mathrm{G}(\mathbf{y}, \mathbf{z})) \right\|_2^2 \ . \tag{2.24}$$

It evaluates the Euclidean distance between the conditioning pixels and their predictions by G. In other words, using a matrix notation of Equation (2.22), the latter conditional likelihood given a generated image equivalently writes

$$\log p_{\mathrm{Y|X}}(\mathbf{y}, \mathrm{G}(\mathbf{y}, \mathbf{z}) \propto -\left\| \mathbf{y} - \mathbf{M_y} \odot \mathrm{G}(\mathbf{y}, \mathbf{z}) \right\|_{\mathrm{F}}^2 \ . \tag{2.25}$$

The second term in Problem (2.23) is the likelihood of the generated image under the true but unknown data distribution p_{X}. Maximizing this term can be equivalently achieved by minimizing the distance between p_{X} and the marginal distribution of the generated samples $\mathrm{G}(\mathbf{y}, \mathbf{z})$. This amounts to minimizing with respect to G, the GAN-like objective function $\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{X}}} \log(\mathrm{D}(\mathbf{x})) + \mathbb{E}_{\substack{\mathbf{z}\sim p_{\mathrm{Z}} \\ \mathbf{y}\sim p_{\mathrm{Y}}}} \log(1 - \mathrm{D}(\mathrm{G}(\mathbf{y}, \mathbf{z})))$ (Goodfellow et al., 2014).

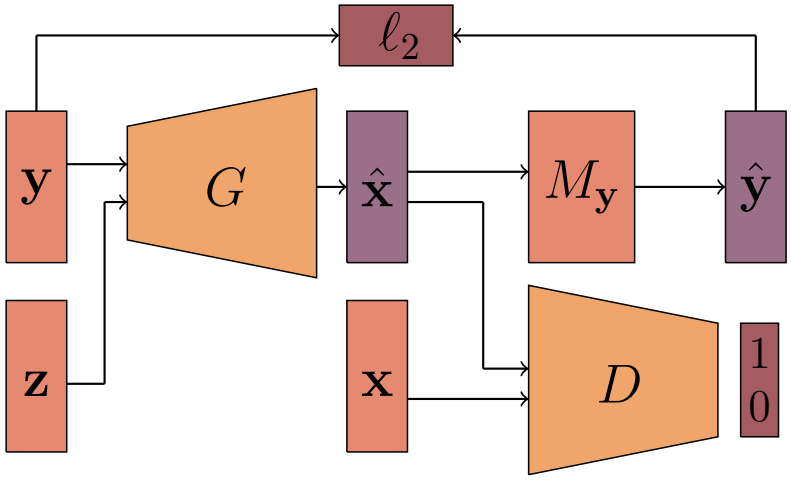


Figure 2.5: Overview of our formulation of the maximum a posteriori approach for image reconstruction using GANs.

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

$$\begin{aligned} \min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}(\mathrm{D},\mathrm{G}) \quad &= \quad \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}} \Big[\log(\mathrm{D}(\mathbf{x})) \Big] \\ &+ \quad \mathop{\mathbb{E}}_{\substack{\mathbf{z}\sim p_{\mathrm{Z}} \\ \mathbf{y}\sim p_{\mathrm{Y}}}} \Big[\log(1-\mathrm{D}(\mathrm{G}(\mathbf{y},\mathbf{z}))) + \lambda \left\| \mathbf{y} - \mathbf{M}_{\mathbf{y}} \odot \mathrm{G}(\mathbf{y},\mathbf{z}) \right\|_{\mathrm{F}}^2 \Big] \ . \end{aligned} \tag{2.26}$$

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 $\mathbf{M}_{\mathbf{y}}$. The trade-off between the distribution matching loss and the constraint enforcement is assessed by the regularization parameter $\lambda \geq 0$. Figure 2.5 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 noise. 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 (2.26), we use stochastic gradient descent. The overall training procedure is detailed in Algorithm 3 and ends 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

Algorithm 3 Proposed training algorithm

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

repeat

sample a mini-batch $\{\mathbf{x}_i\}_{i=1}^m$ of real images from $\mathscr{D}_{\mathrm{X}}$

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

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

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

until a stopping condition is met

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 1.3.3, Equation 1.33, which consists in passing a small set of samples to the discrimination function instead of a single one. PacGAN is intended to tackle the mode collapse problem in GAN training (see Section 1.2.4) . 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.

Algorithm 4 Our training algorithm including PacGAN

Require: $\mathscr{D}_{\mathrm{X}}$ the set of unaltered images, 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 $\{\mathbf{M}_{\mathbf{y}_i}\}_{i=1}^m$ and compute the constraint maps $\{\mathbf{y}_i = \mathbf{M}_{\mathbf{y}_i} \odot \mathbf{x}_i^a\}$

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 $\{\mathbf{M}_{\mathbf{y}_i}\}_{i=1}^m$ and compute the labels $\mathbf{y}_i = \mathbf{M}_{\mathbf{y}_i} \odot \mathbf{x}_i$

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

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

until a stopping condition is met

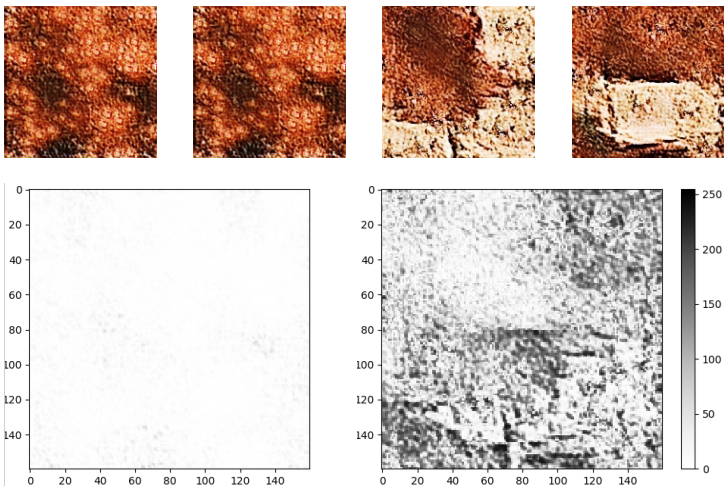


Figure 2.6: An example of a loss of diversity when generating brick texture samples (see Section 2.4.2) using two different random noises $\mathbf{z}$ and a single constraint map $\mathbf{y}$. The two samples on the top left are generated using the classical GAN discriminator whereas the samples on the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

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. We compare our approach to the CGAN approach exclusively, as it is the only approach among the methods reviewed in Section 2.2 to provide a sampling mechanism while not requiring to solve a computationally expensive optimization problem for each generated sample (see Table 2.1).

Datasets

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

FashionMNIST (Xiao et al., 2017) consists of 60,000 28×28 small gray-scale 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 color 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. Samples are shown in Figure 2.9

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 test-bed of our approach on fully-convolutional networks for constrained texture generation task (Figure 2.8).

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 (Figure 2.10).

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 uniformly 0.5% of the pixels to compose 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 neural network architectures for the GAN generator and discriminator in order to adapt to the different scales and image sizes of our datasets. The detailed configuration of these architectures are exposed in Appendix C.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[1] (Yu and Koltun, 2015), which behave well on texture datasets (Ruffino et al., 2017), or encoder-decoder[1] architectures that are commonly used in domain-transfer applications such as CycleGAN (Zhu et al., 2017c).

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[1] (the up-scaling part), and a set of dilated convolutional layers (Yu and Koltun, 2015), while the Up-EncDec has an up-scaling part followed by an encoder-decoder section with skip-connections,

[1] see Glossary, Appendix B

where the constraints are down-scaled, concatenated to the noise, and re-up-scaled to the output size.

The UNet (Ronneberger et al., 2015) architecture is an encoder-decoder[1] where skip-connections[1] are added between the encoder and the decoder. The Res architecture is an encoder-decoder[1] where residual blocks[1] (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)

$$\mathrm{MSE} = \frac{1}{\mathrm{L}} \sum_{i=1}^{\mathrm{L}} \left\| \mathbf{y}_i - \mathbf{M}_{\mathbf{y}_i} \odot \mathrm{G}(\mathbf{y}_i, \mathbf{z}_i) \right\|_{\mathrm{F}}^2 \quad . \tag{2.27}$$

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 1.4 of the chapter 1. We employ FID since the Inception Score has been shown to be less reliable (Barratt and 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 are neither 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 and 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 within 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 C.3.

[1] see Glossary, Appendix B

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 $G(\mathbf{y}, \mathbf{z}_1), \ldots, G(\mathbf{y}, \mathbf{z}_n)$ 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 that minimizes the cost

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

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

Empirical evidences (highlighted in Figure 2.7) 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 (Isola et al., 2016; Zhu et al., 2017c) 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 up-scaling-dilatation network, the encoder-decoder[1] one and the ResNet[1]-like architectures.

Our training algorithm ran for 40 epochs on all reported results. We provide a comparison to CGAN (Mirza and 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,

[1] see Glossary, Appendix B

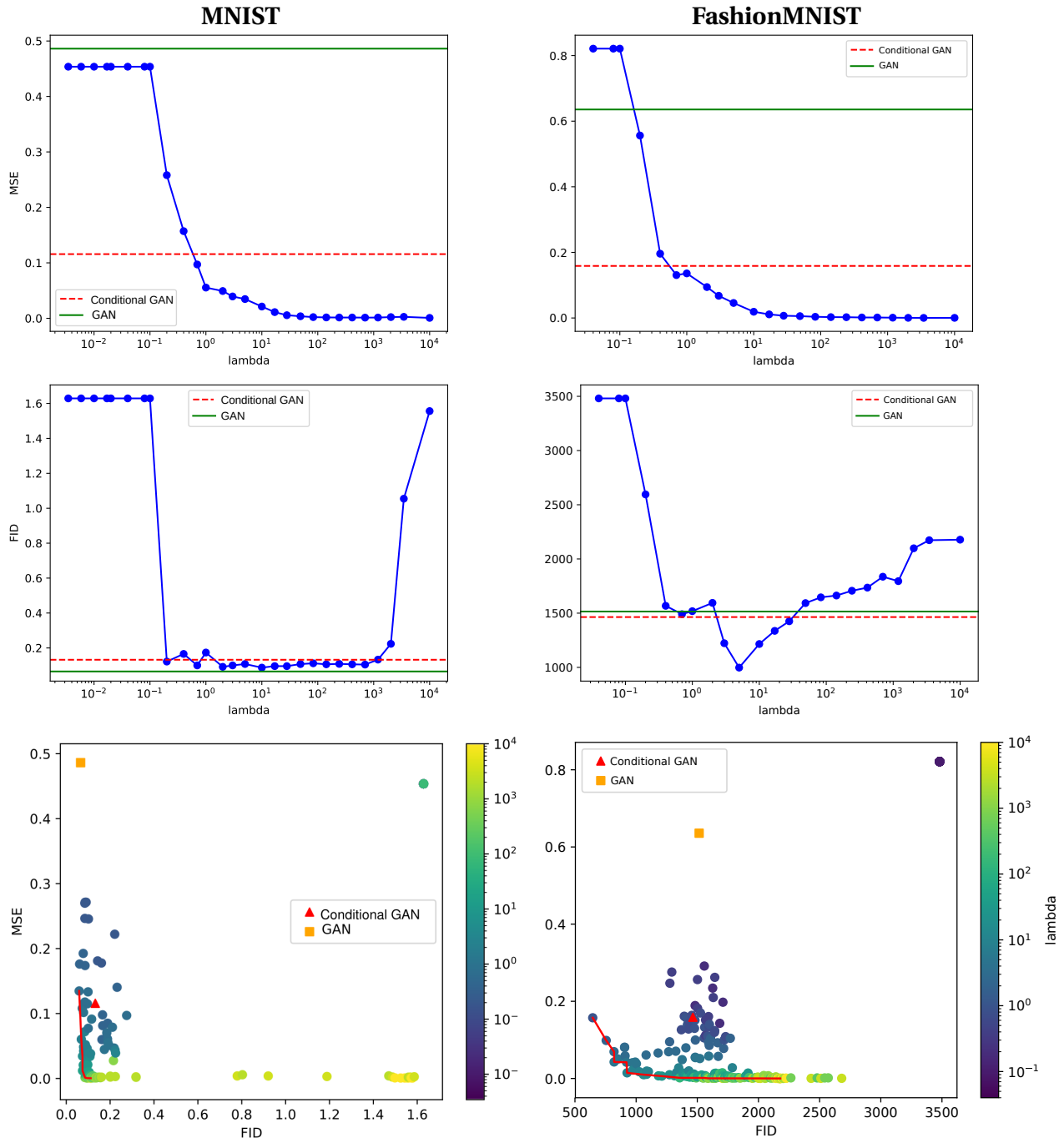


Figure 2.7: Our approach compared to the GAN and CGAN baselines. MSE (Top) and FID (center) w.r.t. the regularization parameter λ; MSE w.r.t the FID (bottom), on the MNIST (left) and Fashion MNIST (right) datasets. Note that the different orders of magnitude for the FID is due to the different classifiers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

the MSE is reported at epoch when the FID is the lowest. The obtained performances are detailed in Table 2.2.

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 up-scaling 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.

For the encoder-decoder[1] models, we can notice that the models using ResNet blocks perform better than just using a UNet[1] generator. A trade-off can also be seen between the FID and MSE for the ResNet models and the UNet[1]-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[1] models perform the best, they tend to lose diversity in the generated samples (see Figure 2.6), whereas the up-scaling-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[1] 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 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. Figure 2.8 show some samples generated with our approach.

High-dimension image reconstruction

We extend the comparison of our approach to CGAN on the CIFAR10 and CelebA datasets (Table 2.4). We also compare generation times (Table 2.5) and visual quality on the CelebA dataset (Table 2.4) with the Semantic Inpainting by Constrained Image Generation approach (Yeh et al., 2017), in order to show the difference in generation times with the approaches that use optimization at generation time.

[1] see Glossary, Appendix B

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.

Dataset	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**
	Yeh et al. (2017)	2.44e-4	$\leq$ **0.0001**	/	/

Table 2.4: Results on the CIFAR10 and CelebA datasets. The reported performances compare CGAN to our proposed GAN conditioned on scarce constraint map.

According to the results obtained in Section 2.4.2 and 2.4.2, we used the *UNetResPac* architecture and 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.

We compare the computation times (in seconds) for generating a set number of mini-batches, composed of 16 images each, using respectively our proposed GAN conditioned on scarce constraint map and the Semantic Inpainting approach (Yeh et al., 2017). The Semantic Inpainting by Constrained Image Generation model is fully trained with the parameters used by Yeh et al. (2017) [1].

The experiments on both datasets show that although CGAN provides better results in terms of visual quality, our approach greatly outperforms it according to the respect of the pixel constraints. They also show that, even if they greatly increase the respect of the constraints (which is to be expected, since these approach optimizes the constraints at generation time), for roughly equivalent visual qualities, solving an optimization problem at generation times is computationally expensive. These computation times make them prohibitive for generating a very large number of images.

Samples generated with our approach are shown in Figure 2.9

[1] Computation times are reported using the author's code and a NVIDIA GTX 1080Ti.

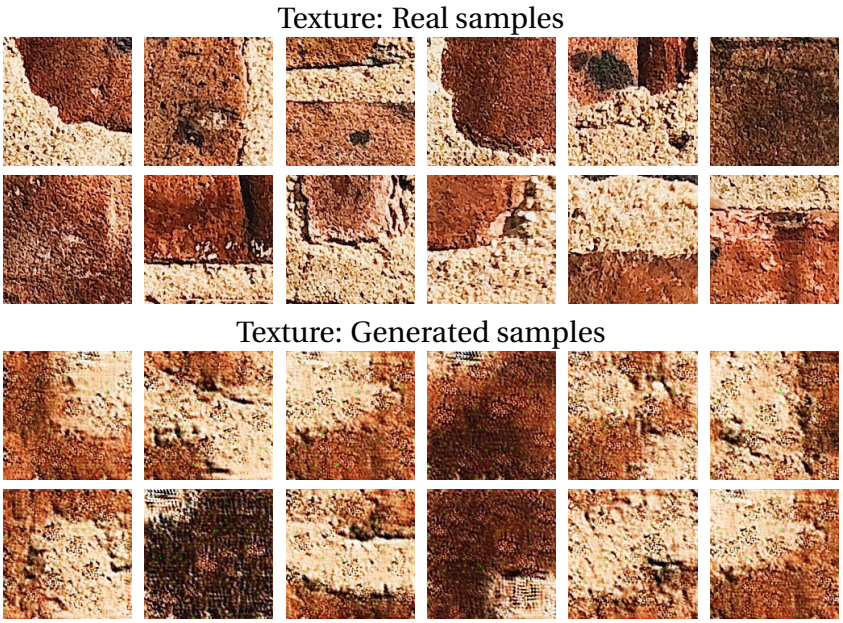


Figure 2.8: Real and generated samples from the Texture dataset.

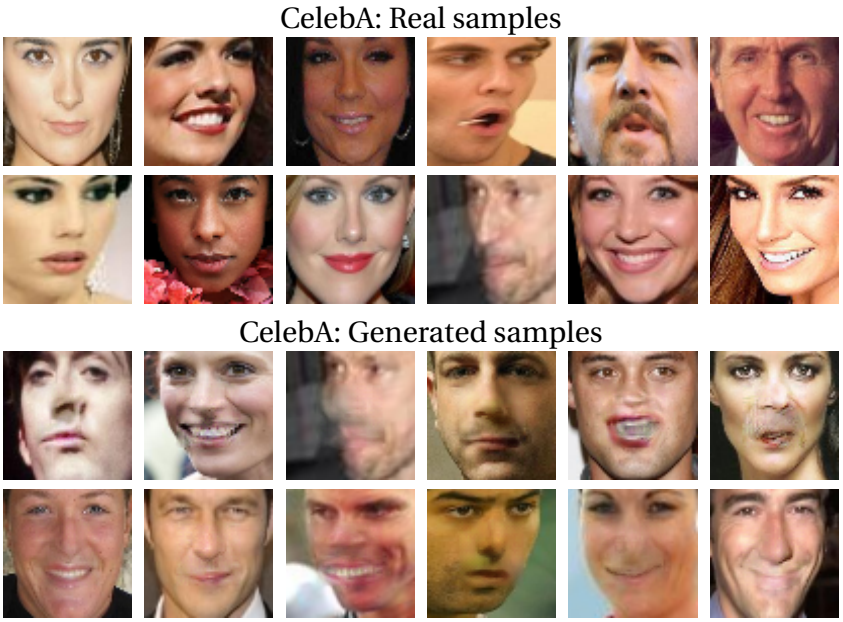


Figure 2.9: Real and generated samples from the CelebA dataset.

Minibatches	Ours	Yeh et al. (2017)
1	0.39s	75.73s
2	0.72s	152.05s
4	0.88s	316.09s
8	1.56s	744.25s
16	2.20s	1056.87s
32	4.48s	2211.45s

Table 2.5: Time comparison (in seconds) on the CelebA datasets with the Semantic Inpainting with Constrained Generative Models (Yeh et al., 2017) for set numbers of mini-batches, composed of 16 images each.

Dataset	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 2.6: 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 labeled data in order to learn a deep network classifier from which the FID score can be computed.

The obtained results are summarized in Tables 2.6 and 2.7. 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 C.2 in the supplementary materials. Also examples of generated images by our approach pictured in Figure 2.10 show that we preserve the visual quality and honor the constraints.

Dataset	Model	Best HOG	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 2.7: Evaluation of the visual quality between the CGAN approach and ours on the Subsurface dataset using several metrics.

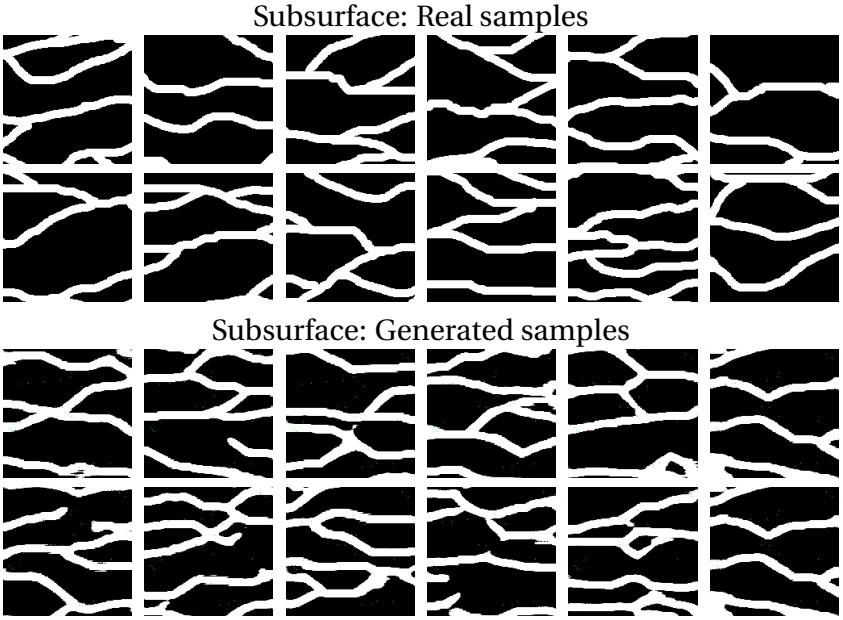


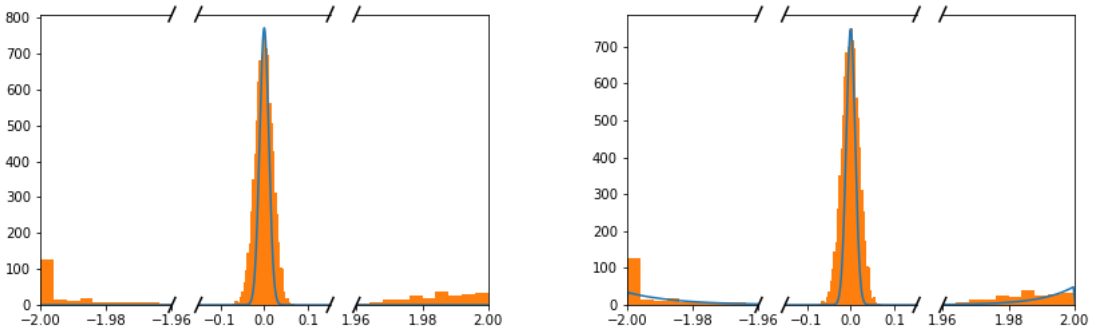
Figure 2.10: Real and generated samples from the Subsurface dataset.

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 prior 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 Section 2.4.1, we assume that the reconstruc-



(a) Gaussian best fit

(b) Mixture distribution fit

Figure 2.11: 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).

tion error of the model is Gaussian. However this may not be 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 tends to be close to either 0, -2, or 2 (see Figure 2.11a). For this application, a mixture distribution could be more appropriate as it could better model both 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 2.11b).

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 with with auxiliary tasks for generative modeling

Chapter abstract

In this chapter, we tackle 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 image domain constraints for converting RGB images to polarimetry-encoded ones with constraints derived from the physics of polarimetry. Our work is driven by an application in road-scene object detection in polarimetric images. This is motivated by the application of deep learning frameworks to 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 object detection in road scenes images, their use may be hindered by reduced number of labeled training images. This issue could be resolved by data augmentation. Moreover, polarization modality is subject to some physical feasibility constraints that could be impeded standard classical data augmentation techniques. Hence we propose a polarimetric image generation framework based on the CycleGAN approach to transfer RGB images to polarimetry-encoded ones, in order to convert full labeled datasets to the polarimetric domain. 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 at training stage, our GAN learns to generate high-quality polarimetric images that follow the physics of polarimetry. This allows for transferring existing labeled RGB datasets to the polarimetric domain without re-labeling of 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 images are indeed coherent from a physics perspective. Further experiments on road object detection show that by training a detection model using a polarimetric images dataset that includes generated polarimetric images, the detection of cars and pedestrian are improved. We end the chapter with some perspectives on training CycleGANs for generating polarimetric images using proximal methods. We formulate a projection operator and propose two algorithms that could potentially allow for generating realistic polarimetric images.

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

- Rachel Blin, Cyprien Ruffino, Samia Ainouz, Romain Hérault, Gilles Gasso, Fabrice Mériaudeau, and Stéphane Canu (2021). "Generating Polarimetric-Encoded Images Using Constrained Cycle-Consistent Generative Adversarial Networks". In: *Currently in Preparation*

Contents

3.1 Introduction

In the previous chapters, we have seen that Generative adversarial networks (Goodfellow et al., 2014) are powerful deep generative models, able to 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 less common imaging domains. Among these works, the task of generating images from the RGB domain to these other imaging domains, using domain-translation approaches such as CycleGAN (Zhu et al., 2017a). For instance, methods to generate infrared road scenes from RGB counterpart images (Zhang et al., 2018b), 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 on non-conventional imaging techniques. Specifically we investigate a generative model framework to produce realistic polarimetric images from RGB 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 and Andreou, 1995). Applications of polarimetric imaging range from indoor autonomous navigation (Berger et al., 2017), depth map estimation (Zhu and Smith, 2019), 3D object 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. As such, 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. We study this problem in a fully unsupervised context, meaning that we do not have access to datasets of paired or labeled samples. Techniques b based on cycle-consistency (Zhu et al., 2017a) enabled to achieve unpaired image-to-image translation with a relatively few number of images. They allow to circumvent the expensive labeling issue in deep learning 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 proposed framework based on CycleGAN (Zhu et al., 2017a) is able to handle the physical polarization constraints during training. We demonstrate the effectiveness of our constrained-output CycleGAN on the KITTI[1] (Geiger et al., 2012) and BDD100K datasets[2] (Yu et al., 2020), two common datasets used for object detection in road scenes. Using the generated polarization-encoded images to train a deep object detector, we witness an improvement of the detection performances of cars and pedestrians which are of great interest for autonomous driving applications.

[1] Karlsruhe Institute of Technology and Toyota Technological Institute

[2] the Berkeley Deep Drive dataset

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 a CycleGAN-based model 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: the polarization formalism and the physical constraints it involves are first presented in Section 3.2. Then, in Section 3.3, the formulation of the image-to-image translation from RGB images to the polarimetric domain is described, and we review different approaches to tackle this problem, as well as their limitations. In Section 3.4 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 in Section 3.4.2, in which we aim to translate RGB images of KITTI and BDD100K datasets into polarimetric images. We evaluate our approach as a data augmentation technique using an object detection network trained on the generated images. Section 3.5 discusses prospective optimization approaches to handle the polarimetric constraints. Most notably, we formulate a projector operator on the space of the constraints and propose two algorithm based on this projector for handling polarimetric constraints. The last section concludes the chapter.

3.2 Polarimetric imaging: formalism and constraints

As most of this chapter revolves around polarimetric image generation, we first introduce the formalism of polarization that stems from the physics of polarimetry. Polarization is a property of light that represents the direction of propagation of the electrical field of the light wave. Polarimetric imaging defines the polarization state of light waves reflected by each part of the scene. When an un-polarized light wave is being reflected, it becomes partially linearly polarized and 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 it does not represent the wavelength of light, but contains rich information about the surfaces that the light reflected on, most notably information about the materials of these surfaces (Gross et al., 2012). In this section, we first propose an overview of the mathematical formulation of polarimetric imaging and then review the different physical constraints that apply to this imaging paradigm.

3.2.1 Polarimetry-encoded images and Stokes vectors as parameters for polarization

Similarly to color images, several encoding formats exist for polarimetric images. The acquisition principle of a polarimetric camera is based on a set of polarizers located between the object and the sensors (Bass et al., 1995). In this work, we rely on a polarimetric image encoding format that consists in four channel images respectively obtained with four

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°.

different linear polarizers oriented at $\alpha_\theta, \theta \in \{1,...,4\} = (0°, 45°, 90°, 135°)$. The polarimetric camera captures an image $\mathbf{y} \in \mathbb{Y} \subset \mathbb{R}^{n \times p \times 4}$ consisting in the light intensities $\mathbf{y}_{i,j_{\alpha_\theta}}$ of the scene for each angle α_θ for each pixel $\mathbf{y}_{i,j} = \begin{bmatrix} \mathbf{y}_{i,j_0} & \mathbf{y}_{i,j_{45}} & \mathbf{y}_{i,j_{90}} & \mathbf{y}_{i,j_{135}} \end{bmatrix}^\top, \forall i \leq n, j \leq p$. An example of the four different intensities for the same scene is shown in Figure 3.1.

The linearly-polarized reflected light can be described by measurable parameters, specifically by linear Stokes vectors[1]. These parameters are encoded as an image $\mathbf{s} \in \mathbb{S} \subset \mathbb{R}^{n \times p \times 3}$ such that each pixel $\mathbf{s}_{i,j}$ is a Stokes vector $\mathbf{s}_{i,j} = \begin{bmatrix} \mathbf{s}_{i,j_0} & \mathbf{s}_{i,j_1} & \mathbf{s}_{i,j_2} \end{bmatrix}^\top \in \mathbb{R}^3$, $1 \leq i \leq n, 1 \leq j \leq p$. Here, $\mathbf{s}_0 > 0$ represents the total light intensity, $\mathbf{s}_1$ the amount of horizontally and vertically linearly polarized light and $\mathbf{s}_2$ the amount of linearly polarized light at ± 45°.

Associated with each polarimetry encoding format is its so-called calibration matrix $\mathbf{A}$ that allows for computing Stokes vectors. In this work, the calibration matrix is set by the manufacturer of the polarimetric camera we use (a Polarcam™4D Technology[2]) as

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

Using $\mathbf{A} \in \mathbb{R}^{4\times3}$, we define[3] the relationship between Stokes vectors $\mathbf{s} \in \mathbb{R}^{n \times p \times 3}$ and the light intensities $\mathbf{y} \in \mathbb{R}^{n \times p \times 4}$ reaching the camera as

$$\mathbf{y} = \mathbf{A}\mathbf{s} \ . \tag{3.1}$$

To compute Stokes parameters from the measured intensities (equation 3.1), we require $\mathbf{A}^\dagger = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top \in \mathbb{R}^{3\times4}$ the pseudo-inverse (or Moore-Penrose inverse) of the matrix $\mathbf{A}$. The relationship between $\mathbf{s}$ and $\mathbf{y}$ is defined for each pixel as

$$\mathbf{s}_{i,j} = \mathbf{A}^\dagger\mathbf{y}_{i,j} \forall\, i \leq n, j \leq p \ .$$

[1] https://en.wikipedia.org/wiki/Stokes_parameters

[2] https://www.4dtechnology.com

[3] To ease the notation for the rest of this chapter, we use the matrix product notation $\mathbf{Mt}$ between a tensor $\mathbf{t} \in \mathbb{R}^{n\times p\times a}$ and a matrix $\mathbf{M} \in \mathbb{R}^{a\times b}$ as computing a tensor $\mathbf{t}' \in \mathbb{R}^{n\times p\times b}$ such that each of its elements $\mathbf{t}'_{i,j} = \mathbf{M}\mathbf{t}_{i,j}$, $1 \leq i \leq n, 1 \leq j \leq p$.

In our work, the pseudo-inverse $\mathbf{A}_\dagger$ of the calibration matrix $\mathbf{A}$ is

$$\mathbf{A}_\dagger = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} , \tag{3.2}$$

thus we have the relation

$$\mathbf{s}_{i,j} = \mathbf{A}^\dagger \mathbf{y}_{i,j} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{i,j_0} \\ \mathbf{y}_{i,j_{45}} \\ \mathbf{y}_{i,j_{90}} \\ \mathbf{y}_{i,j_{135}} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{0_{i,j}} + \mathbf{y}_{i,j_{90}} \\ \mathbf{y}_{0_{i,j}} - \mathbf{y}_{i,j_{90}} \\ \mathbf{y}_{i,j_{45}} - \mathbf{y}_{i,j_{135}} \end{bmatrix} 1 \leq i \leq n, 1 \leq j \leq p \ .$$

3.2.2 Physical constraints of polarimetry

A polarimetry-encoded image $\mathbf{y}$ is deemed valid if its Stokes vectors satisfy two main conditions: they must be physically admissible and they must be the result of an acquisition process that uses the right calibration. Since we are interested in generating new polarimetric images, they will have to comply with these essential constraints.

The three components of Stokes vectors represent respectively the total light intensity, the intensity of the vertically and horizontally polarized light, and the intensity of the diagonally polarized light. To be physically admissible, the total light intensity $\mathbf{s}_0$ of Stokes vectors $\mathbf{s}$ should be at least superior to the sum of the intensities of the diagonally, vertically and horizontally polarized light. Thus, we have

$$\mathbf{s}_0 \geq \sqrt{\mathbf{s}_1^2 + \mathbf{s}_2^2} \ . \tag{3.3}$$

Additionally, since $\mathbf{s}_0$ represents the total light intensity, it cannot be 0. Thus, to be physically admissible, a Stokes vector has to meet the conditions

$$\mathbf{s}_0 > 0 \quad \text{and} \quad \mathbf{s}_0^2 \geqslant \mathbf{s}_1^2 + \mathbf{s}_2^2 \ . \tag{3.4}$$

Then, an additional check has to be done to ensure that a polarimetric image $\mathbf{y}$ has been obtained using a given calibration matrix $\mathbf{A}$. To do so, we evaluate if the image $\mathbf{y}$ can be reconstructed from Stokes vectors computed using the pseudo-inverse $\mathbf{A}^\dagger$ of the calibration matrix. By using equations (3.1) and (3.2), we can formulate the condition

$$\mathbf{y} = \mathbf{A}\mathbf{A}^\dagger \mathbf{y} \ . \tag{3.5}$$

Note that in the case where $\mathbf{A}$ is invertible, $\mathbf{A}^\dagger = \mathbf{A}^{-1}$ so $\mathbf{A}\mathbf{A}^\dagger = \mathrm{I}d$, thus this constraint is always enforced. In general, this constraint is satisfied if and only if $\mathbf{y} \in \ker(\mathbf{A}\mathbf{A}^\dagger - \mathrm{I}d)$ In the specific case where the calibration matrix $\mathbf{A}$ defined in Equation 3.1 is used, the solution to this constraint is

$$\left\{ \mathbf{y} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_{45} & \mathbf{y}_{90} & \mathbf{y}_{135} \end{bmatrix}^\top \middle| \mathbf{y}_0 + \mathbf{y}_{90} = \mathbf{y}_{45} + \mathbf{y}_{135} \right\} \ . \tag{3.6}$$

The proof of this result is deferred to Appendix D. We finally obtain a set of three polarimetric constraints $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_3$ formulated as

$$\begin{aligned} \mathscr{C}_1 &: \mathbf{y} = \mathbf{A}\mathbf{A}^\dagger \mathbf{y}, \\ \mathscr{C}_2 &: \mathbf{s}_0^2 \geqslant \mathbf{s}_1^2 + \mathbf{s}_2^2 \ , \\ \mathscr{C}_3 &: \mathbf{s}_0 > 0 \ . \end{aligned} \tag{3.7}$$

In this chapter, we consider images that satisfy this set of constraints to be physically admissible.

3.3 Unsupervised color to polarimetric image translation

In this section, we propose a formulation of the polarimetric image generation as a constrained domain-transfer problem. We examine the limits of the classical domain-transfer approaches and propose an overview of some recent methods that overcome these limits.

3.3.1 Polarimetric image generation as a constrained conditional image generation problem

The problematic studied in this chapter is learning a model for generating physically realistic polarimetry-encoded images from color images, using neither paired data nor labeled data. A polarimetric image generated that way should remain semantically consistent with the input color image, i.e it should represent the same scene and objects but in a different modality. Thus, there are two important aspects to this problem. First, we aim to learn a generative model G_{XY} such that, for an RGB image $\mathbf{x} \in \mathbb{X}$ issued from the distribution p_X, the generated images $G_{XY}(\mathbf{x}) = \hat{\mathbf{y}} \in \mathbb{Y}$ are issued from p_Y the distribution of the real polarimetric images. Hence, the generated images $\hat{\mathbf{y}}$ and their Stokes vectors $\hat{\mathbf{s}} = \mathbf{A}^\dagger \hat{\mathbf{y}}$ must respect the constraints $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_3$ (see 3.7).

We can formulate this as

$$\begin{aligned} \max_{G_{XY}} \quad & L(G_{XY}) = \mathbb{E}_{\mathbf{x}\sim p_X}\left[\log(p_Y(G_{XY}(\mathbf{x}))\right] \\ \text{s.c.} \quad & G_{XY}(\mathbf{x}_i) = \mathbf{A}\mathbf{s}_i \;\; ; \;\; \mathbf{s}_{0_i}^2 \geq \mathbf{s}_{1_i}^2 + \mathbf{s}_{2_i}^2 \;\text{ and }\; \mathbf{s}_{0_i}^2 > 0 \\ \text{with} \quad & \mathbf{s}_i = \mathbf{A}^\dagger(G_{XY}(\mathbf{x}_i)) \quad \forall i \;\; . \end{aligned} \tag{3.8}$$

These constraints enforce the physical admissibility of the generated polarimetric images, however they do not guarantee that the objects pictured in the generated images $G_{XY(\mathbf{x}_i)}$ will be the same as in the original images $\mathbf{x}_i$. This property is called **semantic consistency** between the input $\mathbf{x}$ and the generated image $\hat{\mathbf{y}}$ and is essential to the task of domain-transfer. Indeed, a model that is not semantically consistent could generate realistic images that are completely different from the provided inputs.

One final requirement is that the model should be trainable in an unpaired and unsupervised way. This implies that the only available datasets consist in unpaired and unlabeled samples $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_s\}, \mathbf{x}_i \in \mathbb{X}$ and $\mathscr{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_s\}, \mathbf{y}_i \in \mathbb{Y}$ from the two domains $\mathbb{X}$ and $\mathbb{Y}$.

These two requirements can be approached using unsupervised domain-transfer methods.

3.3.2 Approaches for unsupervised conditional domain-transfer

In Section 1.2.3, we reviewed different approaches for unsupervised domain-transfer. Most notably, we introduced the cycle-consistency losses used in models such as CycleGAN

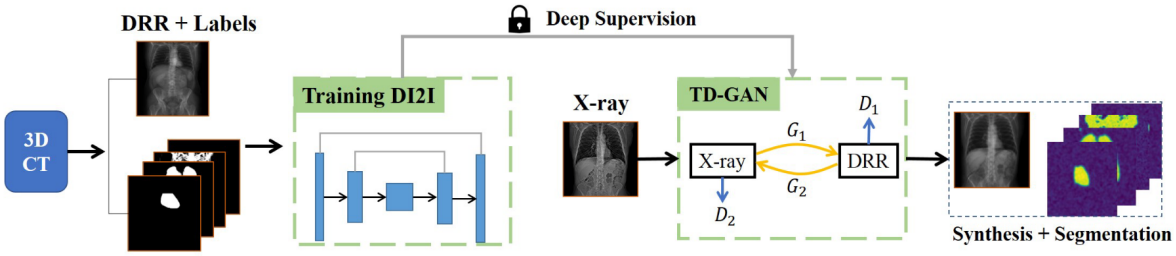


Figure 3.2: Overview of the TD-GAN approach. Figure from Zhang et al. (2018c).

(Zhu et al., 2017a). These approaches consists in training two conditional GAN models, $G_{XY} : \mathbb{X} \rightarrow \mathbb{Y}$ and $G_{YX} : \mathbb{Y} \rightarrow \mathbb{X}$, that maps samples between the distributions p_X and p_Y of the two domains, then training them with both the classical GAN losses and the cycle-consistency loss, formulated as

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim p_\mathbf{x}} ||\mathbf{x} - G_{YX}(G_{XY}(\mathbf{x}))||_1 + \mathop{\mathbb{E}}_{y\sim p_\mathbf{y}} ||\mathbf{y} - G_{XY}(G_{YX}(\mathbf{y}))||_1 \ .$$

The full CycleGAN problem can be summed up as

$$\begin{aligned}
&\min_{G_{XY},G_{YX}} \max_{D_X,D_Y} L_{CycGAN}(G_{XY}, G_{YX}, D_X, D_Y) = \\
&\min_{G_{XY},G_{YX}} \max_{D_X,D_Y} \mathop{\mathbb{E}}_{\mathbf{x}\sim p_\mathbf{x}} \left[(1 - D_X(\mathbf{x})^2) + (D_Y(G_{XY}(\mathbf{x})))^2\right] + \mathop{\mathbb{E}}_{y\sim p_\mathbf{y}} \left[(1 - D_Y(\mathbf{y}))^2 + (D_X(G_{YX}(\mathbf{y})))^2\right] \\
&\qquad + \lambda \left[\mathop{\mathbb{E}}_{\mathbf{x}\sim p_\mathbf{x}} ||\mathbf{x} - G_{YX}(G_{XY}(\mathbf{x}))||_1 + \mathop{\mathbb{E}}_{y\sim p_\mathbf{y}} ||\mathbf{y} - G_{XY}(G_{YX}(\mathbf{y}))||_1 \right] \ .
\end{aligned}$$

While these approaches allow for efficient domain-translation (and notably image-to-image translation), they do not integrate domain-specific knowledge, for example the polarimetric constraints mentioned in Section 3.2.

To constrain the domain-transfer process, several approaches rely on adding a task specific loss to the CycleGAN objective. This can be done in a supervised way, leveraging on labeled data by training a task model and minimizing its error, or in an unsupervised way with an explicit task loss.

TD-GAN (Zhang et al., 2018c) integrates a supervised conditioning process in a task of semantic segmentation from organ X-ray images. To do so, the authors rely on existing labeled datasets of digitally reconstructed radiographs (DRRs) and use them to train a Dense Image-to-Image (DI2I) (Huang et al., 2018) semantic segmentation model. They use this model to condition a CycleGAN-like model that translates images from the X-ray to the DRR domain by adding the segmentation loss to the CycleGAN objective. In other words, DRR images are translated to X-ray, then back to DRRs, segmented using the pre-trained model and then compared to the ground truth segmentation using binary cross-entropy. The full procedure is described in Figure 3.2.

CyCADA (Hoffman et al., 2018) also implements the idea of using pre-trained classifiers or segmentation models to condition domain-transfer. They achieve this conditioning by comparing the classes (or segmentation maps) of the source and generated images, and then adding the fitting term of these supervision models to the objective loss of a CycleGAN-like model. While this conditioning process requires a model pre-trained in a

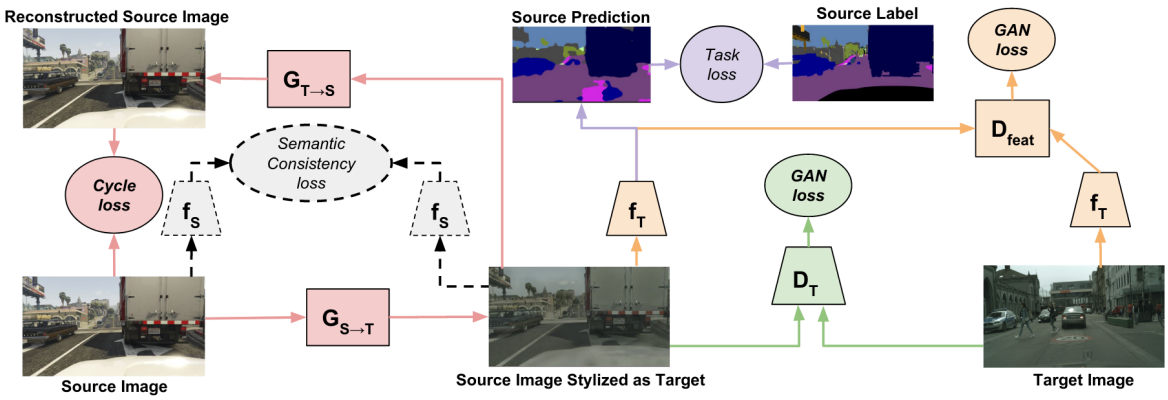


Figure 3.3: Cycle-consistent adversarial adaptation overview. By directly remapping source training data into the target domain, they remove the low-level differences between the domains, ensuring that their task model is well-conditioned on target data. They depict here the image-level adaptation as composed of the pixel GAN loss (green), the source cycle loss (red), and the source and target semantic consistency losses (black dashed) – used when needed to prevent label flipping. For clarity the target cycle is omitted. The feature-level adaptation is depicted as the feature GAN loss (orange) and the source task loss (purple). Figure from Hoffman et al. (2018).

supervised way, training the CyCADA model does not require labeled data. They demonstrate this approach for domain-transfer with datasets such as MNIST (LeCun et al., 1998) and Street View House Numbers (SVHN) (Netzer et al., 2011) for a digit classification task, and the SYNTHIA (Ros et al., 2016), GTA (Richter et al., 2016) and Cityscapes (Cordts et al., 2015) datasets for road scenes semantic segmentation The full method is illustrated in Figure 3.3.

Several methods also leverage on this conditioning approach to enhance the performance of the domain-transfer task. **VIGAN** (Shang et al., 2017) includes a denoising auto-encoder; the aforementioned **CyCADA** an uses adversarial loss on the features extracted by a pre-trained classifier and **Attention-GAN** (**Chen2018b**) leverages on an attention mechanism, both to increase the visual quality of the generated samples.

3.4 Generating polarimetric images with auxiliary tasks for domain-transfer modeling

The problem studied in this chapter is image-to-image translation from RGB images to the polarimetry domain. In the previous sections, we formalized the constraints of polarimetric imaging and reviewed the different approaches for image-to-image translation based on generative modeling, as well as conditioning mechanism based on task-specific losses.

As a main contribution in this chapter, we propose a CycleGAN-based approach for conditioning the domain translation task with the constraints of polarimetry. We formulate a relaxation of both the calibration constraint (see Equation 3.5) and the constraint of physical admissibility (see Equation 3.8) and add the related costs to the CycleGAN losses.

We evaluate this methods on a road-scene RGB to polarimetric image translation. We

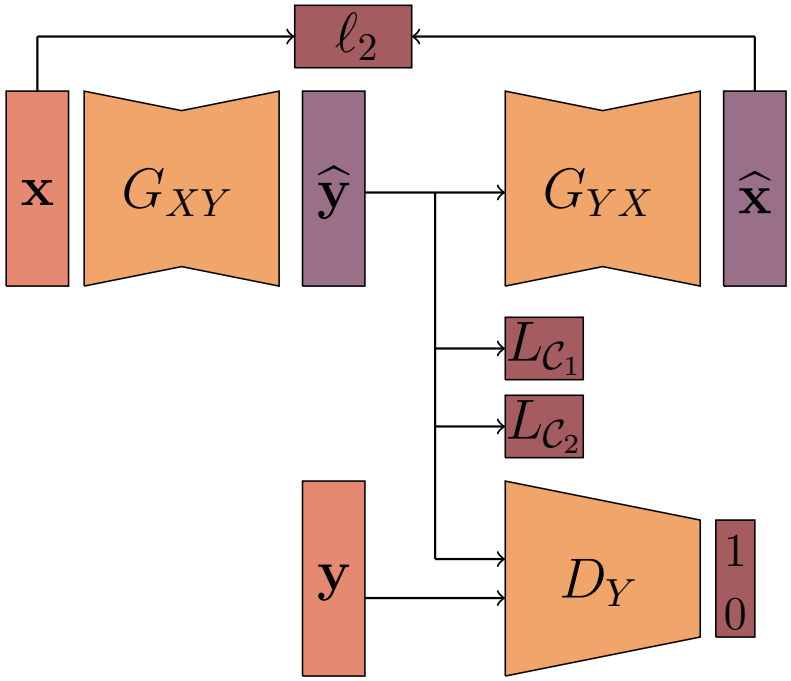


Figure 3.4: Overview of the CycleGAN training process extended with $L_{\mathscr{C}_1}$ and $L_{\mathscr{C}_2}$.

compare the visual quality of the samples and the respect of the polarimetric constraints. We further examine the physical admissibility by studying the impact of using our generated images as a dataset for a road-scene detection task. We observe that our approach outperforms CycleGAN on the used criterions and that using generated polarimetric images contributes to enhancing the performances on the road-scene detection task.

3.4.1 Auxiliary tasks for color to polarimetric images domain-transfer

As discussed above, to generate a realistic polarimetric image from an RGB image, we propose to use the CycleGAN approach to learn the translation models G_{YX} between $\mathbb{Y}$ the space of the polarimetric images and $\mathbb{X}$ the RGB image domain. Let $\hat{\mathbf{y}} \in \mathbb{R}^{n\times p\times 4}$ be a generated polarimetric image. To be physically admissible, it has to satisfy the admissibility constraints (3.4) and the calibration constraint (3.5).

By design, the first component of Stokes vector is always positive as it represents the total intensity reflected from an object. Indeed the last layer of the generation models customary uses the hyperbolic tangent as activation function, each output intensity $\hat{\mathbf{y}}$ is within the range $]-1,1[$ which we scale to $]0,255[$. Hence $\hat{\mathbf{s}}_{0_{i,j}} = \hat{\mathbf{y}}_{0_{i,j}} + \hat{\mathbf{y}}_{90_{i,j}}$, $1 \leq i \leq n$, $1 \leq j \leq p$ (see equation (3.2)) is ensured to be strictly positive pixel-wise. 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 (1.19) 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. 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, as in Equation (3.8), 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 $\mathrm{G_{YX}}$ and $\mathbf{A\hat{s}}$ is proposed. It reads

$$\mathrm{L}_{\mathscr{C}_1}(\mathrm{G_{XY}}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}} ||\mathrm{G_{XY}}(\mathbf{x}) - \mathbf{A}\mathbf{A}^{\dagger}\mathrm{G_{XY}}(\mathbf{x})||_2 \ . \tag{3.9}$$

Similarly, to enforce the constraint $\mathscr{C}_2$, a rectified linear penalty $\mathrm{L}_{\mathscr{C}_2}$ is considered. It is defined by

$$\mathrm{L}_{\mathscr{C}_2}(\mathrm{G_{XY}}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\mathrm{X}}} \max\left(\hat{\mathbf{s}}_1^2 + \hat{\mathbf{s}}_2^2 - \hat{\mathbf{s}}_0^2, 0\right) \ , \tag{3.10}$$

with $\hat{\mathbf{s}} = \begin{bmatrix}\hat{\mathbf{s}_0} & \hat{\mathbf{s}_1} & \hat{\mathbf{s}_2}\end{bmatrix}^{\top} = \mathbf{A}^{\dagger}\mathrm{G_{XY}}(\mathbf{x})$.

The loss $\mathrm{L}_{\mathscr{C}_1}$ translates the respect of the acquisition conditions according to the calibration matrix A while $\mathrm{L}_{\mathscr{C}_2}$ 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

$$\mathrm{L}_{final}(\mathrm{G_{XY}}, \mathrm{G_{YX}}, \mathrm{D_X}, \mathrm{D_Y}) = \mathrm{L}_{Cycle\mathrm{GAN}}(\mathrm{G_{XY}}, \mathrm{G_{YX}}, \mathrm{D_X}, \mathrm{D_Y}) + \mu\mathrm{L}_{\mathscr{C}_1}(\mathrm{G_{XY}}) + \nu\mathrm{L}_{\mathscr{C}_2}(\mathrm{G_{XY}}) \ . \tag{3.11}$$

The non-negative hyper-parameters μ and $\nu \in \mathbb{R}^+$ control respectively the balance of calibration and admissibility constraints according to the CycleGAN loss $\mathrm{L}_{Cycle\mathrm{GAN}}$ (see equation (1.19)). As the values of $\mathrm{L}_{\mathscr{C}_1}$ and $\mathrm{L}_{\mathscr{C}_2}$ are computed pixel-wise, we consider their averages over the whole images in the objective function. The training principle of the proposed generative model is illustrated in Figure 3.4 and detailed in Algorithm 5.

Algorithm 5 CycleGAN with relaxed constraints training algorithm

Require: $\mathscr{D}_{\mathrm{X}}$ and $\mathscr{D}_{\mathrm{Y}}$ two unpaired datasets, $\mathrm{G_{XY}}$ and $\mathrm{G_{YX}}$ the mapping networks, $\mathrm{D_X}$ and $\mathrm{D_Y}$ the discrimination models, b the mini-batch size, $\mathbf{A}$ the calibration matrix and $\mathbf{A}^{\dagger}$ its pseudo-inverse, λ, μ, ν hyperparameters

repeat

sample a mini-batch $\{\mathbf{x}_i\}_{i=1}^{b}$ from the RGB $\mathscr{D}_{\mathrm{X}}$

sample a mini-batch $\{\mathbf{y}_i\}_{i=1}^{b}$ from the polarimetric $\mathscr{D}_{\mathrm{Y}}$

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

$\sum_{i=1}^{m}\left(\mathrm{D_X}(\mathbf{x}_i) - 1\right)^2 + \left(\mathrm{D_X}(\mathrm{G_{YX}}(\mathbf{y}_i))\right)^2$

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

$\sum_{i=1}^{m}\left(\mathrm{D_Y}(\mathbf{y}_i) - 1\right)^2 + \left(\mathrm{D_Y}(\mathrm{G_{XY}}(\mathbf{x}_i))\right)^2$

for $i = 1$ to b, compute $\hat{\mathbf{s}}_i = \begin{bmatrix}\hat{\mathbf{s}_0}_i & \hat{\mathbf{s}_1}_i & \hat{\mathbf{s}_2}_i\end{bmatrix}^{\top} = \mathbf{A}^{\dagger}\mathrm{G_{XY}}(\mathbf{x}_i)$.

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

$\sum_{i=1}^{n}\left(\mathrm{D_Y}(\mathrm{G_{XY}}(\mathbf{x}_i)) - 1\right)^2 + \lambda\left(||\mathbf{x}_i - \mathrm{G_{YX}}(\mathrm{G_{XY}}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - \mathrm{G_{XY}}(\mathrm{G_{YX}}(\mathbf{y}_i))||_1\right)$

$+\mu\left(\|\mathrm{G_{XY}}(\mathbf{x}_i) - \mathbf{A}\mathbf{A}^{\dagger}\mathrm{G_{XY}}(\mathbf{x}_i)\|_{\mathrm{F}}^2\right) + \nu\left(\max(\mathbf{s}_{1_i}^2 + \mathbf{s}_{2_i}^2 - \mathbf{s}_{0_i}^2, 0\right)$

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

$\sum_{i=1}^{n}\left(\mathrm{D_X}(\mathrm{G_{YX}}(\mathbf{y}_i)) - 1\right)^2 + \lambda\left(||\mathbf{x}_i - \mathrm{G_{YX}}(\mathrm{G_{XY}}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - \mathrm{G_{XY}}(\mathrm{G_{YX}}(\mathbf{y}_i))||_1\right)$

until a stopping condition is met

Figure 3.5: Examples of images in the polarimetric dataset (Blin et al., 2020). Only the intensities $\mathbf{y}_0$ are shown here.

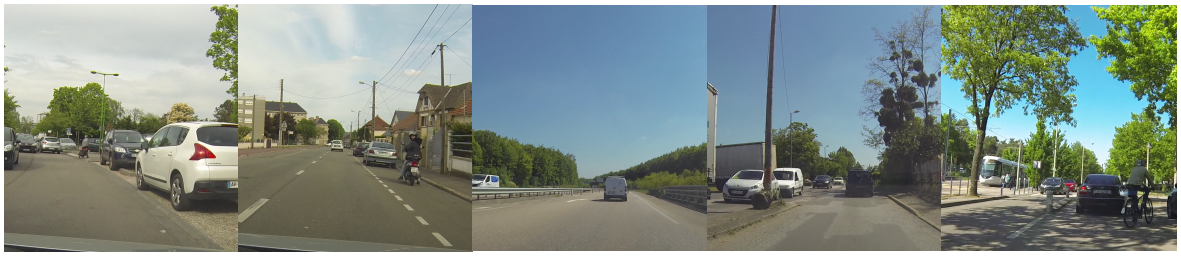


Figure 3.6: Examples of images in the RGB dataset.

3.4.2 Experimental evaluation

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

Experimental setup

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 3.5 and 3.6 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 $\mathbf{y}_0, \mathbf{y}_{45}, \mathbf{y}_{90}$ and $\mathbf{y}_{135}$. The RGB images are of dimension $906 \times 945 \times 3$.

Our CycleGANs were trained for 400 epochs on randomly cropped patches of size

	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.

200×200, as recommended for CycleGAN (Zhu et al., 2017a). As for the constraints, we found experimentally that setting the hyper-parameters $\mu = 1$ and $\nu = 1$ in equation (3.11) provides the best performances. As for the original CycleGAN, the hyper-parameter λ, controlling the reconstruction cost, was set to $\lambda = 10$. The learning rate is decreased linearly from 2×10^{-4} to 2×10^{-6} during the epochs.

To evaluate the effectiveness of the generative model, we consider KITTI (Geiger et al., 2012) and BDD100K (Yu et al., 2020)(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 CycleGANs we train are 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.

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

Overall, the trained CycleGANs and detection networks under these settings are evaluated in qualitative and quantitative ways. The end goal is to check the ability of the generated images to help learning polarimetry-based features for object detection, and 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) (see Section 1.4). 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 Gaussian distributions (thoroughly explained in Section 1.4). We compute this distance using 500 images from each generated polarimetric dataset and from the test set as described in Table 3.1.

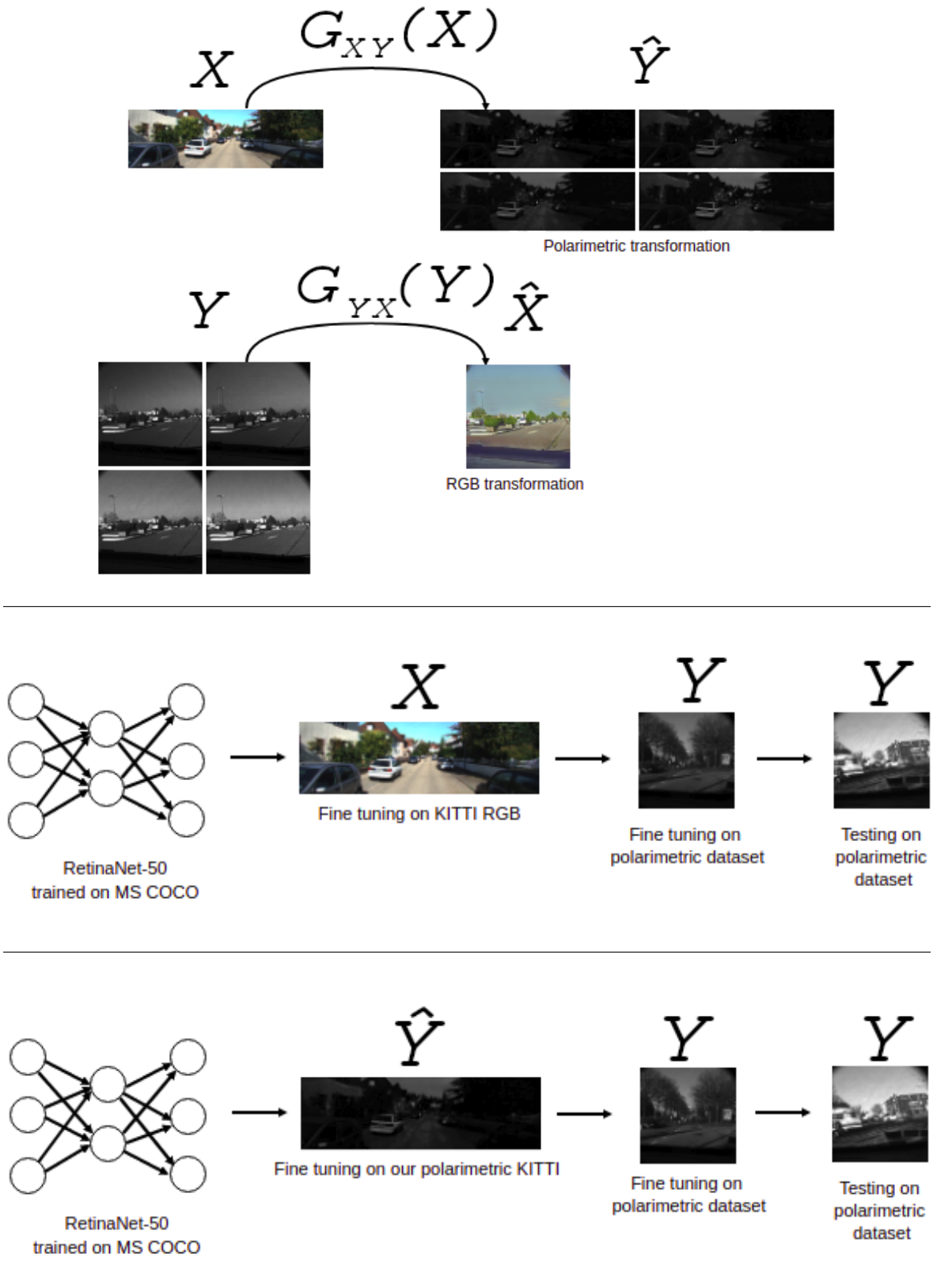


Figure 3.7: Setup of the detection evaluation experiment. The procedure is illustrated with the KITTI dataset and straightforwardly extends to the BDD100K dataset. Top: domain-transfer procedure with our model; Center: baseline setup; Bottom: our setup

Figure 3.8: Examples of polarimetric image reconstruction. From left to right: $\mathbf{y}_0$, $\mathbf{y}_{45}$, $\mathbf{y}_{90}$ and $\mathbf{y}_{135}$ ground truth, RGB image generated with a trained model and $\mathbf{y}_0$, $\mathbf{y}_{45}$, $\mathbf{y}_{90}$ and $\mathbf{y}_{135}$ generated from RGB image using the model trained with relaxed constraints.

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:

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

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

Results and discussion

First we evaluate whether the generated images are qualitatively coherent or not. For the sake, we reconstruct the polarimetric images from their RGB generation, which refers to $\mathrm{G_{XY}} \circ \mathrm{G_{YX}}$. The reconstruction of these RGB images is shown in Figure 3.8.

As for the constraints, Table 3.2 shows how including them to the CycleGAN's loss helps to generate 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 (which corresponds to acquisition errors), 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 (see table 3.2) indicates that taking the constraints into account improves visual quality and physical admissibility of the generated samples on the test set.

Datasets	$\mathscr{C}$	Mean	Median	FID
Real polar	$\mathscr{C}_1$	0.06 ± 0.04	0.04	N/A
	$\mathscr{C}_2$	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	6022.7
	$\mathscr{C}_2$	27.31 ± 43.5%	2.15%	
	$\mathscr{C}_3$	0%	0%	
Relaxed constraints	$\mathscr{C}_1$	0.12 ± 0.04	0.12	**4485.1**
	$\mathscr{C}_2$	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, and the visual quality using the Fréchet Inception Distance (FID). 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. Here, the column $\mathscr{C}$ indicates the evaluated constraint. $\mathscr{C}_1$ refers to the constraints $\mathbf{y} = \mathbf{A}\mathbf{A}^\dagger\mathbf{y}$, $\mathscr{C}_2$ to $\mathbf{s}_0^2 \geqslant \mathbf{s}_1^2 + \mathbf{s}_2^2$ and $\mathscr{C}_3$ to $\mathbf{s}_0 > 0$. 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 $||\mathbf{y} - \mathbf{A}\mathbf{A}^\dagger\mathbf{y}|| / (||\mathbf{y}|| + ||\mathbf{A}\mathbf{A}^\dagger\mathbf{y}||)$.

Next, we show the benefit of the generated images in object detection task, enabling to verify that objects within them are globally physically coherent. The RetinaNet-based detection model were trained according to the setups previously described (see Figure 3.7) 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 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 an overall improvement of 9% in the detection, using Polar-KITTI with constraints. Similarly for Polar-BDD100K dataset, we notice an improvement of 10% for pedestrian detection which leads to an increased mAP of 5% (pedestrians and cars). However, we 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 do not perform well on small objects. To verify that, we compare the evolution of the detection 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 3.9.The results of this experiment illustrate 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.

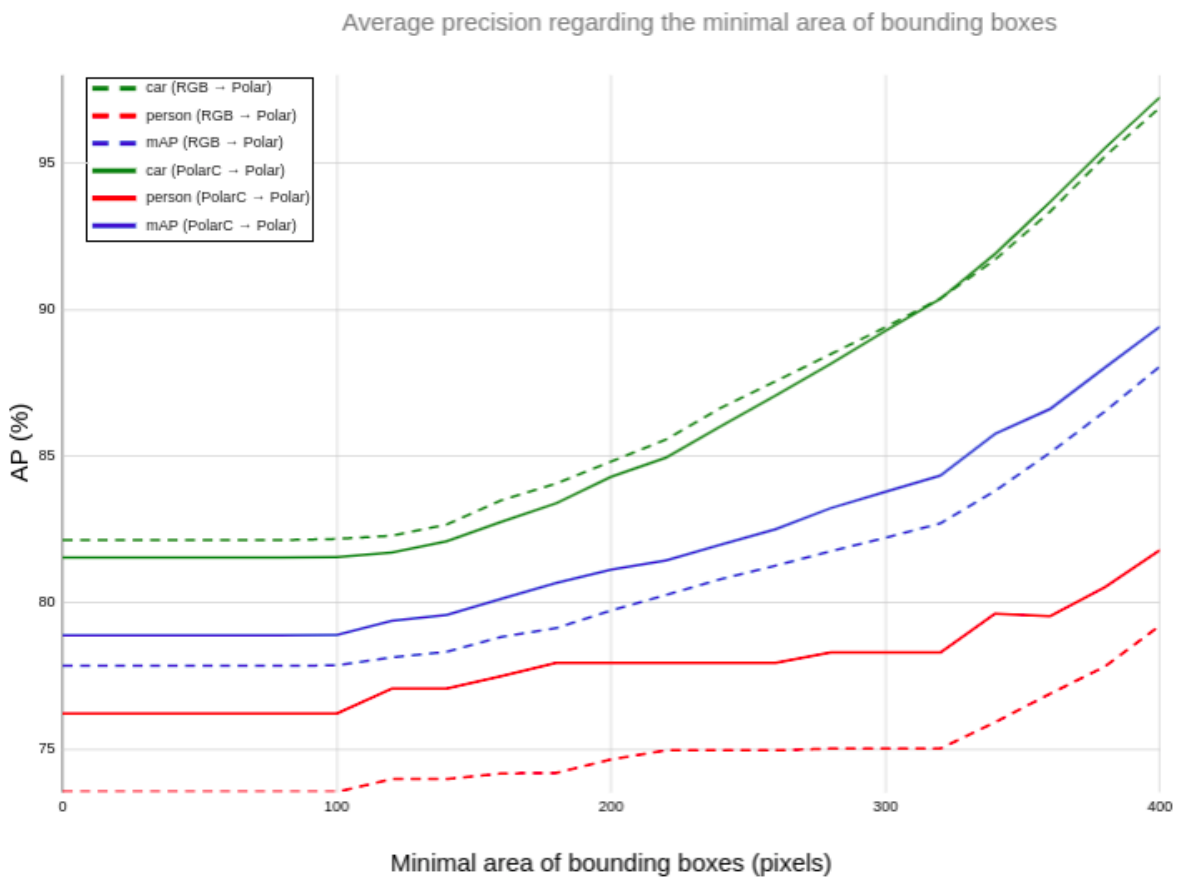


Figure 3.9: Evolution of the average precision when setting a minimal area of the bounding boxes to be detected. Here green lines refer to the evolution of cars' detection, blue lines to the evolution of the *m*AP and red lines to the evolution of person's detection. The dashed lines refer to the training including the BDD100K RGB and the solid lines to the training including Polar-BDD100K.

Databases	Class	Test	ER_o	Databases	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	person	0.673	-0.03	Polar-BDD100K	person	0.720	0.06
no $\mathscr{C}$ + real polar	car	0.786	-0.01	no $\mathscr{C}$ + real polar	car	0.816	0.03
	*m*AP	0.730	-0.02		*m*AP	0.768	0.05
Polar-KITTI with	person	**0.704**	-0.12	Polar-BDD100K	person	**0.762**	-0.10
relaxed $\mathscr{C}$	car	**0.794**	-0.04	with relaxed $\mathscr{C}$	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 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 physical constraints and the bottom row represents the detection model fine-tuned on Polar-KITTI or Polar-BDD100K with enforced constraints. Every model is finally fine-tuned on the real polarimetric dataset.

3.5 Perspectives

In this section, we propose to explore some perspectives and new approaches for transferring color images to the polarimetric domain. We formulate an operator for projecting generated images onto the space delimited by the constraints and propose two algorithms based on this projector. Since these approaches are, at the writing this thesis, work-in-progress, their experimental evaluations are deferred to future work.

3.5.1 Generating polarimetric images with a projector operator

In the same fashion as in Section 3.4, we aim to generate images $\hat{\mathbf{y}} = \mathrm{G}_{\mathrm{XY}}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{X}$ is a sample from the RGB domain, such that $\hat{\mathbf{s}} = \mathbf{A}^{\dagger}\hat{\mathbf{y}} \in \mathbb{S}$ the space of Stokes vectors. Each of the vectors must respect $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_3$, which in fact correspond to a second-order cone, or Lorentz cone (Boyd et al., 2004). Thus, let

$$\mathscr{C} = \left\{ (\mathbf{s}_0, \mathbf{s}_{1,2}) \in \mathbb{S} \;\middle|\; \|\mathbf{s}_{1,2}\|_2 \leq \mathbf{s}_0,\ \mathbf{s}_{1,2} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \right\} \ , \tag{3.12}$$

a convex set whose vectors satisfy the aforementioned constraints. As such, we can reformulate the Problem (3.8) as

$$\begin{aligned} \max_{\mathrm{G}} \quad & \mathrm{L}(\mathrm{G}) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{X}}} \left[\log(p_{\mathrm{Y}}(\mathrm{G}_{\mathrm{XY}}(\mathbf{x})) \right] \\ \text{s.c.} \quad & \mathbf{A}^{\dagger}(\mathrm{G}_{\mathrm{XY}}(\mathbf{x}_i)) \in \mathscr{C}, \forall\, i \ . \end{aligned} \tag{3.13}$$

With such a membership constraint, the projection operator $\Pi_{\mathscr{C}}$ on $\mathscr{C}$ can be defined as

Algorithm 6 Training algorithm for CycleGAN with projected images

Require: $\mathscr{D}_X$ and $\mathscr{D}_Y$ two unpaired datasets, G_{XY} and G_{YX} the mapping networks, D_X and D_Y the discrimination models, m the mini-batch size, $\mathbf{A}$ the calibration matrix and $\mathbf{A}^\dagger$ its pseudo-inverse, λ, μ hyperparameters

repeat

sample a mini-batch $\{\mathbf{x}_i\}_{i=1}^m$ from $\mathscr{D}_X$

sample a mini-batch $\{\mathbf{y}_i\}_{i=1}^m$ from $\mathscr{D}_Y$

update D_X by stochastic gradient descent of

$\sum_{i=1}^m \left(D_X(\mathbf{x}_i) - 1\right)^2 + \left(D_X(G_{YX}(\mathbf{y}_i))\right)^2$

update D_Y by stochastic gradient descent of

$\sum_{i=1}^m \left(D_Y(\mathbf{y}_i) - 1\right)^2 + \left(D_Y(\mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)))\right)^2$

update G_{XY} by stochastic gradient descent of

$\sum_{i=1}^n \left(D_Y(G_{XY}(\mathbf{x}_i)) - 1\right)^2 + \mu\left(\mathbf{A}\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)\right)$

$+\lambda\left(||\mathbf{x}_i - G_{YX}(\mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)))||_1 + ||\mathbf{y}_i - \mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(G_{YX}(\mathbf{y}_i)))||_1\right)$

$+\mu\left(\|\mathbf{x}_i - \mathbf{A}\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)\|_F^2\right)$

update G_{YX} by stochastic gradient descent of

$\sum_{i=1}^n \left(D_X(G_{YX}(\mathbf{y}_i)) - 1\right)^2$

$+\lambda\left(||\mathbf{x}_i - G_{YX}(\mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)))||_1 + ||\mathbf{y}_i - \mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(G_{YX}(\mathbf{y}_i))||_1\right)$

until a stopping condition is met

the solution to the optimization problem

$$\min_{(\mathbf{r},\mathbf{u})\in\mathscr{C}} \frac{1}{2}\|(\mathbf{s}_0, \mathbf{s}_{1,2}) - (\mathbf{r}, \mathbf{u})\|_2^2 \ , \tag{3.14}$$

which has a closed-form (Parikh and Boyd, 2014) as

$$\Pi_{\mathscr{C}}(\mathbf{s}_0, \mathbf{s}_{1,2}) = \begin{cases} (\mathbf{s}_0, \mathbf{s}_{1,2}) & \text{if} \quad \|\mathbf{s}_{1,2}\|_2 \leq \mathbf{s}_0 \\ \frac{1+\mathbf{s}_0/\|\mathbf{s}_{1,2}\|_2}{2}(\|\mathbf{s}_{1,2}\|_2, \mathbf{s}_{1,2}) & \text{if} \quad \|\mathbf{s}_{1,2}\|_2 > \mathbf{s}_0 \end{cases} \tag{3.15}$$

We can introduce this projection operator to the training algorithm instead of the loss term $L_{\mathscr{C}_2}$ obtain by the relaxation of $\mathscr{C}_2$. To do so, the output of G_{XY} is systematically projected onto $\mathscr{C}$ using $\Pi_{\mathscr{C}}$ as

$$\hat{\mathbf{y}}_{\Pi_{\mathscr{C}}} = \mathbf{A}\Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY(\mathbf{x})}) \ , \tag{3.16}$$

with $\mathbf{x} \in \mathbb{X}$ an RGB image. This process is summed up in Algorithm 6 and illustrated in Figure 3.10.

3.5.2 Proximal method for generating polarimetric images

Another solution is to formulate an alternative version to the loss induced by relaxation (3.10) that measures the distance between the Stokes vectors of the generated image and their projection on the constraint space, as

$$L_{prox}(G_{XY}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\left[\|\mathbf{A}^\dagger G_{XY}(\mathbf{x}) - \Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{x}))\|^2\right] \ , \tag{3.17}$$

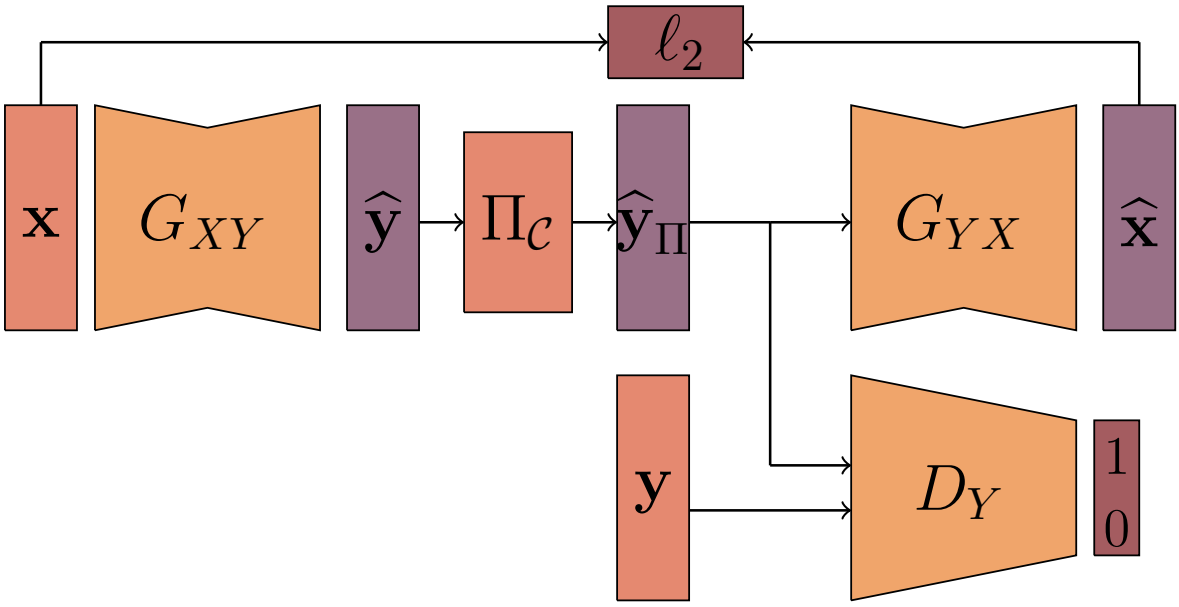


Figure 3.10: Overview of the CycleGAN training process with extended the projection operator. The $L_{\mathscr{C}_1}$ term is omitted.

with λ a regularization parameter. This loss can be substituted to $L_{\mathscr{C}_2}$ in Equation 3.11, thus the problem becomes

$$L_{final}(G_{XY}, G_{YX}, D_X, D_Y) = L_{CycleGAN}(G_{XY}, G_{YX}, D_X, D_Y) + \mu L_{\mathscr{C}_1}(G_{XY}) + \nu L_{prox}(G_{XY}) \ . \quad (3.18)$$

This process is summed up in Algorithm 7 and illustrated in Figure 3.11. Note that the gradient of the distance $\Omega_{\mathscr{C}} = \|\mathbf{A}^\dagger G_{XY}(\mathbf{y}) - \Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{y}))\|^2$ can be expressed (Parikh and Boyd, 2014) as

$$\nabla_{G_{XY}} \Omega_{\mathscr{C}}(\mathbf{s}) = (\mathbf{s} - \Pi_{\mathscr{C}}(\mathbf{s})) \times \begin{cases} 0 & \text{if} \quad \|\mathbf{s}_{1,2}\|_2 \leq \mathbf{s}_0 \\ \nabla_{G_{XY}} \mathbf{s} - \nabla_{G_{XY}} \frac{1}{2}\left[(1 + \frac{\mathbf{s}_0}{\|\mathbf{s}_{1,2}\|_2})(\|\mathbf{s}_{1,2}\|_2, \mathbf{s}_{1,2})\right] & \text{if} \quad \|\mathbf{s}_{1,2}\|_2 > \mathbf{s}_0 \end{cases} \quad (3.19)$$

Such an approach for learning models with constraints has been used, for example, by Kervadec et al. (2019) as an alternative to the Lagrangian version of an image segmentation problem under volume constraints determined by a convolutional neural network (Pathak et al., 2015).

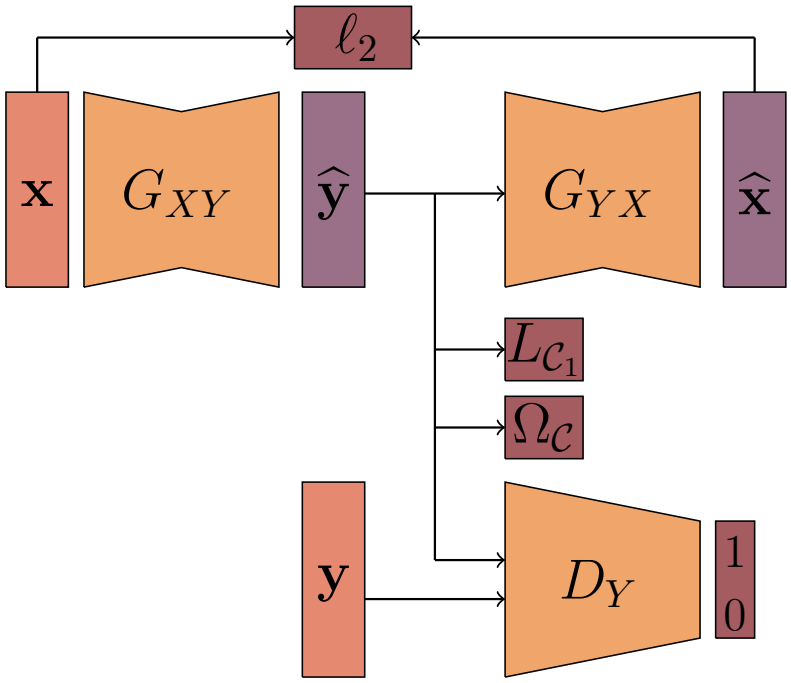


Figure 3.11: Overview of the CycleGAN training process extended with the $L_{\mathcal{C}_1}$ and proximal losses

Algorithm 7 CycleGAN with proximal training algorithm

Require: $\mathscr{D}_X$ and $\mathscr{D}_Y$ two unpaired datasets, G_{XY} and G_{YX} the mapping networks, D_X and D_Y the discrimination models, m the mini-batch size, $\mathbf{A}$ the calibration matrix and $\mathbf{A}^\dagger$ its pseudo-inverse, λ, μ, ν hyperparameters

repeat

sample a mini-batch $\{\mathbf{x}_i\}_{i=1}^m$ from $\mathscr{D}_X$

sample a mini-batch $\{\mathbf{y}_i\}_{i=1}^m$ from $\mathscr{D}_Y$

update D_X by stochastic gradient descent of

$$\sum_{i=1}^m \left(D_X(\mathbf{x}_i) - 1\right)^2 + \left(D_X(G_{YX}(\mathbf{y}_i))\right)^2$$

update D_Y by stochastic gradient descent of

$$\sum_{i=1}^m \left(D_Y(\mathbf{y}_i) - 1\right)^2 + \left(D_Y(G_{XY}(\mathbf{x}_i))\right)^2$$

for $i = 1$ to n, compute $\hat{\mathbf{s}}_i = \begin{bmatrix}\hat{\mathbf{s}_0}_i & \hat{\mathbf{s}_1}_i & \hat{\mathbf{s}_2}_i\end{bmatrix}^\top = \mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)$.

update G_{XY} by stochastic gradient descent of

$$\sum_{i=1}^n \left(D_Y(G_{XY}(\mathbf{x}_i)) - 1\right)^2 + \lambda\left(||\mathbf{x}_i - G_{YX}(G_{XY}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - G_{XY}(G_{YX}(\mathbf{y}_i))||_1\right)$$
$$+\mu\left(\|\mathbf{x}_i - \mathbf{A}\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i)\|_F^2\right) + \nu\|\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i) - \Pi_{\mathscr{C}}(\mathbf{A}^\dagger G_{XY}(\mathbf{x}_i))\|^2$$

update G_{YX} by stochastic gradient descent of

$$\sum_{i=1}^n \left(D_X(G_{YX}(\mathbf{y}_i)) - 1\right)^2 + \lambda\left(||\mathbf{x}_i - G_{YX}(G_{XY}(\mathbf{x}_i))||_1 + ||\mathbf{y}_i - G_{XY}(G_{YX}(\mathbf{y}_i))||_1\right)$$

until a stopping condition is met

3.6 Conclusion

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.

As a perspective, we characterize the set of the constraints and formulate a projection operator on this set. Using this operator, we propose two additional algorithms for transferring color images into the polarimetric domain: the first one consists in projecting the output of the generator to the space of the constraints and giving these projected images to the discriminator. The second algorithm consists in formulating a proximal distance between the projected samples and the generated ones, and adding this distance as an auxiliary cost function to the CycleGAN's objective.

Another future work direction would be to improve the quality of the small objects in generated images, in order to enhance the performances of the road-scene analysis models on, for example, pedestrian detection. Using better architectures and introducing more recent techniques (see Section 1.3) could lead to a notable increase in the visual quality of the generated samples. Extension of the generation procedure to road scene images under adverse weather conditions may help improving object detection in these situations.

It would also be interesting to extend the generation of polarimetric images to other domains such as medical and Synthetic-Aperture Radar (van Zyl and Kim, 2011) imaging.

Chapter 4

Conclusion and Perspectives

In this chapter, we sum up the contributions proposed in this thesis and discuss interesting directions for future work.

Contributions

In this thesis, we study the conditioning of Generative Adversarial Networks (GANs) using of auxiliary tasks. We focus on two real-world applications, the task of reconstructing images of underground water channels from a limited set of points and the task of transferring images from the color domain to a polarimetry-encoded image modality. Through these applications, we propose dedicated auxiliary tasks for conditioning both image-reconstruction and domain-transfer models.

Image reconstruction as an auxiliary task to generative modeling

The task of image reconstruction consists in recovering an image from very noisy or sparse measurements. In Chapter 2 of this thesis, we study the case in which only a few pixels of the image are available, usually less than a percent. We propose to use a GAN combined with a reconstruction task to learn to recover images from the very low amount of pixels. The first benefit of this method is that, similarly to the Conditional GAN approach (see Section 1.2.2), a model trained with our method is able to generate new samples in a single neural network forward pass. This allows for quickly sampling a high number of potential image reconstructions from a single set of pixels. For this, the approach introduces a hyperparameter λ that weights the impact of the reconstruction task. Through a large-scale study on the MNIST and FashionMNIST datasets, we empirically show that this hyperparameter allows for controlling a trade-off between the visual quality of the generated samples and the fidelity of the generation process with respect to the initial image. We evaluate our method on several datasets of natural images, namely CIFAR10, CelebA and a texture dataset and show that our method provides equal or better results than a conditional GAN without auxiliary task, while endowed with the added benefit of the control hyperparameter. Finally, we show that our method performs well on the real-world application of reconstructing underground terrain from few measurements by evaluating it on a dataset of image-like 2D slices of underground terrain.

Polarimetric image generation with auxiliary tasks for generative modeling

As a second main contribution, we investigate the conditioning of GAN-based domain-transfer approaches using auxiliary tasks. We focus on the task of image modality transfer, from the color domain to polarimetric images. Such images bear strong constraints that directly stem from both the physics of polarimetry and the configuration of the acquisition device. We design a set of auxiliary tasks that directly aim to push the transferred images towards enforcing the aforementioned constraints. We propose to integrate these new auxiliary tasks to a CycleGAN, a domain-transfer approach based on *cyclic consistency*. We show that our method produces high-quality polarimetric images that enforce both the physical and configuration constraints and generally performs better than unconditioned methods. As a further test of our method, we propose to transfer existing two road-scene color images datasets , BDD100K and KITTI, to the polarimetric domain and train a polarimetric variant of the RetinaNet detection network on the generated data. We show that this approach performs better than the existing approaches.

Perspectives

We now discuss some interesting perspectives of our contributions that could be addressed in future work.

On image reconstruction as an auxiliary task

In Chapter 2, we have studied the problem of image reconstruction and have proposed an approach based on an auxiliary reconstruction task for conditioning generative adversarial networks. Even though this approach provides good results, several directions could be explored in order to enhance it.

Better modeling of the prior distribution of the model error

To formulate the problem as a maximum a-posteriori estimation, the error of the model is assumed to be of a Gaussian distribution. This, however, is not necessarily true and we provide a real-word example of the mismatch between the assumed and actual errors of the generator (see Subsection 2.4.2). A solution to this problem could be to use adapted distributions to model these errors, typically a distribution (or mixture of distributions) from the exponential family (Brown, 1986), as we proposed in Chapter 2 with a mixture of Gaussian and exponential distributions. This would allow to reformulate the maximum a-posteriori estimation and provide with specialized auxiliary tasks for a given image reconstruction problem.

Better architectures and techniques

Although the different architectures employed in our experiments were common for their time, they are nowadays outdated. This is not necessarily compromising, since the main

result of this work is to show that the introduced auxiliary task allows both for high-fidelity image reconstruction and introduces a controllable trade-off between the visual quality and the respect of the constraints. Indeed, these results are independent from the architecture choices but, for real-world applications, the highest possible quality is desired. In Chapter 1, we reviewed several recent techniques and architectures that are far more efficient and could give way to better results, both for visual quality and respect of the constraints. These techniques could be directly implemented in our approach without any major changes in order to increase the overall performance of the trained models.

Application to other domains

In this thesis, we have applied image reconstruction to a task of underground terrain reconstruction. Thus, we focus our work on texture images, but our approach could be applied to a number of applications. Similarly to compressed sensing-based approaches, we could envision applications in medical imaging, image compression, or tasks on different types of signals such as audio inpainting (Marafioti et al., 2018). These tasks that require to efficiently sample potential reconstructions.

On color-to-polarimetric domain transfer for data augmentation

In Chapter 3, we studied the problem of transferring color images to the polarimetric domain using a CycleGAN-based approach with domain-specific auxiliary tasks. Motivated by the lack of labeled polarimetric images datasets, we aim to train such a domain-transfer model in order to convert large labeled color images datasets to the polarimetric domain. This method yielded good results, most notably increasing the performances of an object detection model for road-scene analysis. While these results are already satisfying, several extensions can be envisioned.

Projection-based methods

In Section 3.5, we proposed to study this problem as generating images that belong to a well-defined set. We formulated a projector operator for this set and proposed two algorithms based on projection for generating polarimetric images with CycleGAN-based approaches. Thus, we plan to evaluate these approaches with the same experimental setup as the main contribution of this chapter and propose a comparison of all these approaches.

Increasing quality of the small objects in the generated images

As mentioned in Section 3.4.2, the visual quality of the images generated with our approach is not sufficient keeping smaller details. This harms the performances of the road-scene analysis models used to evaluate our approach, especially for detecting smaller objects such as pedestrians. Thus, working on enhancing the architectures and objectives, using for example some techniques mentioned in Chapter 1 could yield better performance.

Stochastic modeling

As opposed to the methods studied in Chapter 2, our CycleGAN-based approach is not stochastic, which implies that it is not possible to generate different polarimetric images for a given RGB image. However, due to the ill-posed nature of the problem, to a unique color image corresponds an non-finite set of polarimetric images that belong to the constrained set. Thus, providing a sampling mechanism with stochastic variants of the CycleGAN such as BiCycleGAN (Zhu et al., 2017b) could further extend the potential of our approach as a data-augmentation technique.

Better metrics for compared acquisition

In order to evaluate the physical realism of our approach, we evaluate the generated images by measuring the error relative to the constraints and by evaluating the impact on the performance of a road-scene analysis model. Since polarimetric images contains rich information about the nature of the captured objects, most notably on the materials of the objects, this could be used to provide better metrics for evaluating our approach. By comparing the statistics of a given type of objects in the generated images, for example cars, to actual cars in the real data, we could assess the physical realism of the generated images.

Other domains of application for polarimetric images

Finally, another interesting perspective would be to apply this domain-transfer approaches to different domains. Indeed, polarimetric data are widely used in, for example, medical imaging (Kupinski et al., 2018; Rehbinder et al., 2016) and Synthetic-Aperture Radar (van Zyl and Kim, 2011) imaging for topological data. Since these domains also lack large labeled datasets, applying our approach as a data-augmentation technique could help increase the performances of models in these domains. (**Paetzold2019**)

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 (cit. on p. 44).

Almahairi, Amjad, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville (2018). *Augmented Cyclegan: Learning Many-to-Many Mappings from Unpaired Data* (cit. on p. 67).

Antipov, G., M. Baccouche, and J. Dugelay (Sept. 2017). "Face Aging with Conditional Generative Adversarial Networks". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017 IEEE International Conference on Image Processing (ICIP), pp. 2089–2093 (cit. on pp. 1, 7).

Arjovsky, Martin and Léon Bottou (2017). "Towards Principled Methods for Training Generative Adversarial Networks". In: (cit. on pp. 26, 34).

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein GAN". In: (cit. on pp. 26, 28, 29, 34).

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 (cit. on p. 46).

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. 67).

Barratt, Shane and Rishi Sharma (2018). *A Note on the Inception Score* (cit. on pp. 35, 56).

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. 68).

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 (cit. on p. 41).

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* (cit. on pp. 29, 30).

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. 67).

Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester (July 1, 2000). "Image Inpainting". In: *Proceedings of the 27th Annual Conference on Computer Graph-*

ics and Interactive Techniques. SIGGRAPH '00. USA: ACM Press/Addison-Wesley Publishing Co., pp. 417–424 (cit. on p. 40).

Bińkowski, Miko\ laj, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton (Jan. 2018). "Demystifying MMD GANs". In: (cit. on pp. 30, 35).

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 p. 76).

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* (cit. on pp. 67, 79).

Blin, Rachel, Cyprien Ruffino, Samia Ainouz, Romain Hérault, Gilles Gasso, Fabrice Meriaudeau, and Stéphane Canu (2021). "Generating Polarimetric-Encoded Images Using Constrained Cycle-Consistent Generative Adversarial Networks". In: *Currently in Preparation* (cit. on pp. 5, 11, 66).

Bogachev, V. (2007). *Measure Theory*. Berlin Heidelberg: Springer-Verlag (cit. on p. 19).

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. 46, 49).

Borji, Ali (2018). "Pros and Cons of GAN Evaluation Measures". In: (cit. on p. 35).

Bousmalis, K., A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke (May 2018). "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 4243–4250 (cit. on p. 25).

Boyd, Stephen, Stephen P. Boyd, and Lieven Vandenberghe (Mar. 8, 2004). *Convex Optimization*. Cambridge University Press. 744 pp. (cit. on p. 82).

Brock, Andrew, Jeff Donahue, and Karen Simonyan (Sept. 2018). "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: (cit. on pp. 26–28).

Brown, Lawrence D (1986). "Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory". In: Ims (cit. on pp. 52, 88).

Burt, Peter J and Edward H Adelson (1983). "The Laplacian Pyramid as a Compact Image Code". In: p. 9 (cit. on p. 30).

Candes, E. J., J. Romberg, and T. Tao (Feb. 2006). "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information". In: *IEEE Transactions on Information Theory* 52.2, pp. 489–509 (cit. on p. 42).

Candes, Emmanuel and Terence Tao (Dec. 4, 2006). "Near Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" In: *IEEE Transactions on Information Theory* 52.12, IEEE Transactions on Information Theory (cit. on p. 42).

Candes, Emmanuel J. and Terrence Tao (Dec. 2005). "Decoding by Linear Programming". In: *IEEE Transactions on Information Theory* 51.12, pp. 4203–4215 (cit. on pp. 40, 42, 49).

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 (cit. on p. 42).

Chen, Cheng, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng (July 17, 2019). "Synergistic Image and Feature Adaptation: Towards Cross-Modality Domain Adaptation for Medical Image Segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (01), pp. 865–872 (cit. on p. 25).

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. 56).

Clark, Aidan, Jeff Donahue, and Karen Simonyan (2020). "Adversarial Video Generation on Complex Datasets". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (ICLR 2020) (cit. on p. 36).

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2015). "The Cityscapes Dataset". In: *CVPR Workshop on the Future of Datasets in Vision*. Vol. 2 (cit. on p. 73).

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 (cit. on p. 41).

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 (cit. on p. 56).

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*. URL: http://arxiv.org/abs/1705.05263 (visited on 05/23/2020) (cit. on p. 25).

Demir, Ugur and Gozde Unal (Mar. 2018). "Patch-Based Image Inpainting with Generative Adversarial Networks". In: (cit. on p. 46).

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 (cit. on p. 15).

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 (cit. on pp. 34, 35, 77).

Deng, Zhijie, Hao Zhang, Xiaodan Liang, Luona Yang, Shizhen Xu, Jun Zhu, and Eric P. Xing (Nov. 2017). "Structured Generative Adversarial Networks". In: (cit. on p. 32).

Denton, Emily, Soumith Chintala, Arthur Szlam, and Rob Fergus (June 18, 2015). *Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks*. URL: http://arxiv.org/abs/1506.05751 (visited on 05/22/2020) (cit. on p. 30).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (Feb. 27, 2017). *Density Estimation Using Real NVP*. URL: http://arxiv.org/abs/1605.08803 (visited on 05/11/2020) (cit. on pp. 14, 18).

Donahue, Jeff, Philipp Krähenbühl, and Trevor Darrell (2017). "Adversarial Feature Learning". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on p. 32).

Donoho, D.L. (Apr. 2006a). “Compressed Sensing”. In: *IEEE Transactions on Information Theory* 52.4, pp. 1289–1306 (cit. on pp. 44, 49).

Donoho, David L. (2006b). “For Most Large Underdetermined Systems of Linear Equations the Minimal \ell_1-Norm Solution Is Also the Sparsest Solution”. In: *Communications on Pure and Applied Mathematics* 59.6, pp. 797–829 (cit. on p. 43).

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 (cit. on p. 43).

Dumoulin, Vincent, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville (2016). “Adversarially Learned Inference”. In: (cit. on p. 32).

Dziugaite, Gintare Karolina, Daniel M. Roy, and Zoubin Ghahramani (2015). “Training Generative Neural Networks via Maximum Mean Discrepancy Optimization”. In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. Conference on Uncertainty in Artificial Intelligence (cit. on p. 30).

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 (cit. on p. 44).

Engel, Jesse, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts (Sept. 27, 2018). “GANSynth: Adversarial Neural Audio Synthesis”. In: International Conference on Learning Representations (cit. on p. 36).

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. 67).

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 pp. 67, 77).

Golub, Gene H. and Henk A. van der Vorst (Nov. 1, 2000). “Eigenvalue Computation in the 20th Century”. In: *Journal of Computational and Applied Mathematics*. Numerical Analysis 2000. Vol. III: Linear Algebra 123.1, pp. 35–65 (cit. on p. 34).

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 (cit. on p. 41).

Goodfellow, Ian (2016). “NIPS 2016 Tutorial: Generative Adversarial Networks”. In: (cit. on p. 27).

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: (cit. on pp. 1, 7, 13, 18, 20, 21, 26, 29, 30, 51, 66).

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 (cit. on p. 40).

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 (cit. on p. 30).

Gross, Herbert, Bernd Dörband, and Henriette Müller (2012). "Polarimetry". In: *Handbook of Optical Systems*. John Wiley & Sons, Ltd, pp. 559–642 (cit. on p. 68).

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville (2017). "Improved Training of Wasserstein GANs". In: (cit. on p. 29).

Guo, Jiaxian, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang (2018). "Long Text Generation via Adversarial Training with Leaked Information". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*. AAAI Conference on Artificial Intelligence (AAAI-18) (cit. on p. 36).

Guo, Zongyu, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu (Sept. 18, 2019). *Progressive Image Inpainting with Full-Resolution Residual Network*. URL: http://arxiv.org/abs/1907.10478 (visited on 09/30/2020) (cit. on p. 46).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Deep Residual Learning for Image Recognition". In: (cit. on pp. 30, 56).

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: (cit. on pp. 26, 35, 56, 77).

Hindupur, Avinash (2017). *The GAN Zoo*. URL: https://github.com/hindupuravinash/the-gan-zoo (visited on 05/21/2020) (cit. on p. 27).

Hoffman, Judy, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell (July 10–15, 2018). "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 1989–1998 (cit. on pp. 72, 73).

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger (2018). "Densely Connected Convolutional Networks". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. Computer Vision and Pattern Recognition (CVPR) (cit. on p. 72).

Ioffe, Sergey, Christian Szegedy, and Sergey Ioffe (Feb. 2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: (cit. on pp. 18, 30, 31, 107).

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2016). "Image-to-Image Translation with Conditional Adversarial Networks". In: (cit. on pp. 23, 39, 55, 57).

Jetchev, Nikolay, Urs Bergmann, Roland Vollgraf, and Zalando Research (2017). "Texture Synthesis with Spatial Generative Adversarial Networks". In: (cit. on pp. 39, 55, 57).

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 (cit. on p. 35).

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 (cit. on pp. 1, 7).

Kantorovich, L. V. and G. P. Akilov (1982). *Functional Analysis*. Elsevier. 605 pp. (cit. on p. 29).

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen (2017). *Progressive Growing of GANs for Improved Quality, Stability and Variation* (cit. on pp. 31, 36).

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila (Mar. 23, 2020). "Analyzing and Improving the Image Quality of StyleGAN". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. Conference on Computer Vision and Pattern Recognition (cit. on pp. 27, 36, 39).

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 (cit. on p. 84).

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 (cit. on p. 41).

Kim, Taeksoo, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim (2017). "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks". In: *Proceedings of The34th International Conference on Machine Learning*. International Conference on Machine Learning (cit. on pp. 23, 25).

Kingma, Diederik and Max Welling (2014). "Auto-Encoding Variational Bayes". In: (cit. on pp. 14, 16).

Kingma, Durk P. and Prafulla Dhariwal (2018). *Glow: Generative Flow with Invertible 1x1 Convolutions*. 10236–10245 (cit. on pp. 14, 18).

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. 66).

Kolev, Vasil (2011). "Compressed Sensing of Astronomical Images: Orthogonal Wavelets Domains". In: p. 8 (cit. on p. 43).

Krizhevsky, Alex (2009). "Learning Multiple Layers of Features from Tiny Images". In: p. 60 (cit. on pp. 30, 39, 54).

Kupinski, Meredith, Matthieu Boffety, François Goudail, Razvigor Ossikovski, Angelo Pierangelo, Jean Rehbinder, Jérémy Vizet, and Tatiana Novikova (2018). "Polarimetric Measurement Utility for Pre-Cancer Detection from Uterine Cervix Specimens". In: *Biomedical optics express* 9.11, pp. 5691–5702 (cit. on p. 90).

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. 39).

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?" In: *Computers and Geosciences* 133 (cit. on pp. 5, 11, 37, 39, 50).

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 (cit. on pp. 30, 39, 54, 73).

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. 56, 117).

Li, Chongxuan, Kun Xu, Jun Zhu, and Bo Zhang (2017a). "Triple Generative Adversarial Nets". In: (cit. on pp. 22, 32).

Li, Chun-Liang, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos (2017b). "MMD GAN: Towards Deeper Understanding of Moment Matching Network". In: *Proceedings of the 31st Conference on Neural Information Processing*. Conference on Neural Information Processing (cit. on pp. 26, 29, 30).

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 (cit. on p. 28).

Lim, Jae Hyun and Jong Chul Ye (May 8, 2017). *Geometric GAN*. URL: `http://arxiv.org/abs/1705.02894` (visited on 05/21/2020) (cit. on pp. 28, 29).

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. 77).

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. 77).

Lin, Zinan, Ashish Khetan, Giulia Fanti, and Sewoong Oh (2018). *PacGAN: The Power of Two Samples in Generative Adversarial Networks*, pp. 1505–1514 (cit. on pp. 33, 37, 39, 40, 53).

Liu, Ming-Yu, Thomas Breuel, and Jan Kautz (July 22, 2018). "Unsupervised Image-to-Image Translation Networks". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. International Conference on Neural Information Processing Systems (cit. on p. 23).

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. 36, 39, 54).

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully Convolutional Networks for Semantic Segmentation Ppt". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (cit. on p. 108).

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 (cit. on p. 43).

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: p. 6 (cit. on p. 30).

Mallat, Stéphane (2008). *A Wavelet Tour of Signal Processing*. 3rd Edition. Elsevier (cit. on p. 43).

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: (cit. on pp. 28, 29).

Marafioti, Andrés, Nathanaël Perraudin, Nicki Holighaus, and Piotr Majdak (2018). *A Context Encoder for Audio Inpainting* (cit. on pp. 64, 89).

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. 67).

Mescheder, Lars, Andreas Geiger, and Sebastian Nowozin (2018). “Which Training Methods for GANs Do Actually Converge?” In: (cit. on p. 25).

Mirza, Mehdi and Simon Osindero (2014). “Conditional Generative Adversarial Nets”. In: (cit. on pp. 21, 32, 37, 39, 49, 57).

Miyato, Takeru, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida (2018). “Spectral Normalization for Generative Adversarial Networks”. In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on pp. 28, 34).

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. 67).

Mroueh, Youssef and Tom Sercu (Nov. 3, 2017). “Fisher GAN”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Neural Information Processing Systems (cit. on pp. 29, 30).

Müller, Alfred (June 1997). “Integral Probability Metrics and Their Generating Classes of Functions”. In: *Advances in Applied Probability* 29.2, pp. 429–443 (cit. on p. 30).

Nair, Vinod and Geoffrey E Hinton (2010). “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: p. 8 (cit. on p. 30).

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng (2011). “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: p. 9 (cit. on p. 73).

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. 67).

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). *F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization* (cit. on pp. 26, 28, 29).

Odena, Augustus, Christopher Olah, and Jonathon Shlens (2016). “Conditional Image Synthesis with Auxiliary Classifier GANs”. In: (cit. on pp. 21, 32).

Oliveira, Manuel M, Brian Bowen, Richard McKenna, and Yu-Sung Chang (2001). “Fast Digital Image Inpainting”. In: p. 7 (cit. on p. 41).

Pajot, Arthur, Emmanuel de Bezenac, and Patrick Gallinari (2019). “Unsupervised Adversarial Image Reconstruction”. In: *International Conference on Learning Representations* (cit. on pp. 47, 49).

Parikh, Neal and Stephen Boyd (Jan. 13, 2014). “Proximal Algorithms”. In: *Foundations and Trends in Optimization* 1.3, pp. 127–239 (cit. on pp. 83, 84).

Parzen, Emanuel (1962). “On Estimation of a Probability Density Function and Mode”. In: *Annals of Mathematical Statistics* 33.3, pp. 1065–1076 (cit. on p. 35).

Pathak, Deepak, Philipp Krahenbuhl, and Trevor Darrell (Dec. 2015). “Constrained Convolutional Neural Networks for Weakly Supervised Segmentation”. In: *2015 IEEE Inter-*

national Conference on Computer Vision (ICCV). 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile: IEEE, pp. 1796–1804 (cit. on p. 84).

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. 39, 46).

Perone, Christian S., Pedro Ballester, Rodrigo C. Barros, and Julien Cohen-Adad (July 1, 2019). "Unsupervised Domain Adaptation for Medical Imaging Segmentation with Self-Ensembling". In: *NeuroImage* 194, pp. 1–11 (cit. on p. 25).

Peyré, Gabriel and Marco Cuturi (Mar. 18, 2020). *Computational Optimal Transport*. Foundations and Trends in Machine Learning (cit. on p. 29).

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 (cit. on p. 56).

Qi, Guo-Jun (2018). "Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities". In: *International Journal of Computer Vision* 128 (cit. on p. 34).

Radford, Alec, Luke Metz, and Soumith Chintala (2015). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on pp. 26, 30, 36, 55).

Rauhut, Holger (2010). "Compressive Sensing and Structured Random Matrices". In: p. 94 (cit. on p. 43).

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 pp. 67, 90).

Richter, Stephan R, Vibhav Vineet, Stefan Roth, and Vladlen Koltun (2016). "Playing for Data: Ground Truth from Computer Games". In: *European Conference on Computer Vision*. Springer, pp. 102–118 (cit. on p. 73).

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 pp. 30, 56, 109).

Ros, German, Simon Stent, Pablo F. Alcantarilla, and Tomoki Watanabe (Apr. 6, 2016). *Training Constrained Deconvolutional Networks for Road Scene Semantic Segmentation*. URL: http://arxiv.org/abs/1604.01545 (visited on 12/01/2020) (cit. on p. 73).

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 (cit. on p. 43).

Ruffino, Cyprien, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: Conférence Sur l'Apprentissage (cit. on pp. 5, 11, 37, 55, 56, 62).

Ruffino, Cyprien, Romain Hérault, Eric Laloy, and Gilles Gasso (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 (cit. on pp. 5, 11, 37).

— (Nov. 27, 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing* 416, pp. 218–230 (cit. on pp. 5, 11, 37).

Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen (June 2016). "Improved Techniques for Training GANs". In: (cit. on pp. 26, 31, 35, 56).

Sallab, Ahmad El, Ibrahim Sobh, Mohamed Zahran, and Nader Essam (2019). *LiDAR Sensor Modeling and Data Augmentation with GANs for Autonomous Driving* (cit. on p. 67).

Shang, Chao, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi (2017). "VIGAN: Missing View Imputation with Generative Adversarial Networks". In: *Proceedings of the IEEE International Conference on Big Data*. IEEE International Conference on Big Data (cit. on p. 73).

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 (cit. on p. 43).

Sønderby, Casper Kaae, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár (Feb. 21, 2017). "Amortised MAP Inference for Image Super-Resolution". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on pp. 26, 31).

Springenberg, Jost Tobias, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller (Apr. 13, 2015). "Striving for Simplicity: The All Convolutional Net". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on p. 30).

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*. URL: `http://arxiv.org/abs/0901.2698` (visited on 05/22/2020) (cit. on p. 30).

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 (cit. on p. 30).

Strebelle, Sebastien (Jan. 1, 2002). "Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics". In: *Mathematical Geology* 34.1, pp. 1–21 (cit. on pp. 39, 55).

Sun, T., C. Jung, Q. Fu, and Q. Han (2019). "NIR to RGB Domain Translation Using Asymmetric Cycle Generative Adversarial Networks". In: *IEEE Access* 7, pp. 112459–112469 (cit. on p. 25).

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 (cit. on pp. 35, 77).

Szekely, Gabor J and Maria L Rizzo (2004). "Testing for Equal Distributions in High Dimension". In: p. 15 (cit. on p. 30).

Taigman, Yaniv, Adam Polyak, and Lior Wolf (2017). "Unsupervised Cross-Domain Image Generation". In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on p. 23).

Theis, Lucas, Aäron Van Den Oord, and Matthias Bethge (2015). "A Note on the Evaluation of Generative Models". In: (cit. on p. 56).

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 (cit. on p. 44).

Tošić, I. and P. Frossard (Mar. 2011). "Dictionary Learning". In: *IEEE Signal Processing Magazine* 28.2, pp. 27–38 (cit. on p. 43).

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2016). *Instance Normalization: The Missing Ingredient for Fast Stylization* (cit. on pp. 108, 112).

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 (cit. on pp. 1, 7).

Van Zyl, Jakob and Yunjin Kim (Oct. 28, 2011). *Synthetic Aperture Radar Polarimetry*. Hoboken, NJ, USA: John Wiley & Sons, Inc. (cit. on pp. 86, 90).

Vaswani, Ashish, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \ Lukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need". In: (cit. on p. 107).

Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba (2016). "Generating Videos with Scene Dynamics". In: *Proceedings of Neural Information Processing Systems*. Neural Information Processing Systems (cit. on pp. 1, 7).

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*. URL: `http://arxiv.org/abs/1808.06601` (visited on 05/21/2020) (cit. on p. 27).

Wang, Yi, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia (2018b). *Image Inpainting via Generative Multi-Column Convolutional Neural Networks*. 329–338 (cit. on p. 39).

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 (cit. on pp. 1, 7, 39).

Wolff, Lawrence B and Andreas G Andreou (1995). "Polarization Camera Sensors". In: *Image and Vision Computing* 13.6, pp. 497–510 (cit. on p. 67).

Wu, Jiajun, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum (2017). "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". In: *Proceedings of Neural Information Processing Systems*. Neural Information Processing Systems (cit. on pp. 1, 7).

Wu, Yan, Mihaela Rosca, and Timothy Lillicrap (2019). "Deep Compressed Sensing". In: *Proceedings of the 36th International Conference on Machine Learning* (cit. on pp. 44, 45, 49).

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 (cit. on p. 46).

Xiao, Han, Kashif Rasul, and Roland Vollgraf (Aug. 2017). “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: (cit. on pp. 39, 54, 55, 57).

Yang, Dingdong, Seunghoon Hong, Yunseok Jang, Tiangchen Zhao, and Honglak Lee (2019). “Diversity-Sensitive Conditional Generative Adversarial Networks”. In: *International Conference on Learning Representations* (cit. on pp. 39, 53).

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: (cit. on pp. 47, 49, 59, 60, 62).

Yi, Zili, Hao Zhang, Ping Tan, and Minglun Gong (2017). “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *Proceedings of the International Conference on Computer Vision*. International Conference on Computer Vision (cit. on pp. 23, 25).

Yu, Fisher, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell (2020). “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. Conference on Computer Vision and Pattern Recognition (cit. on pp. 67, 77).

Yu, Fisher and Vladlen Koltun (2015). “Multi-Scale Context Aggregation by Dilated Convolutions”. In: (cit. on pp. 55, 108).

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. 46).

Zhang, Han, Ian Goodfellow, Google Brain, Dimitris Metaxas, and Augustus Odena (2018a). “Self-Attention Generative Adversarial Networks”. In: (cit. on pp. 31, 32).

Zhang, Lichao, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan (2018b). “Synthetic Data Generation for End-to-End Thermal Infrared Tracking”. In: *IEEE Transactions on Image Processing* 28.4, pp. 1837–1850 (cit. on p. 66).

Zhang, Yue, Shun Miao, Tommaso Mansi, and Rui Liao (2018c). “Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-Ray Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 599–607 (cit. on p. 72).

Zhao, Junbo, Michael Mathieu, and Yann LeCun (Mar. 6, 2017). “Energy-Based Generative Adversarial Network”. In: *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations (cit. on p. 29).

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. 67).

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: (cit. on pp. 23, 24, 66, 67, 72, 77).

Zhu, Jun-Yan, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman (2017b). “Toward Multimodal Image-to-Image Translation”. In:

Proceedings of the 31st Conference on Neural Information Processing System (NeurIPS 2017). Neural Information Processing Systems (NeurIPS), p. 12 (cit. on p. 90).

Zhu, Xinyue, Yifan Liu, Zengchang Qin, and Jiahong Li (2017c). *Emotion Classification with Data Augmentation Using Generative Adversarial Networks* (cit. on pp. 23, 55, 57).

Appendix A

Publications

Conference papers

- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (May 2017). "Dilated Spatial Generative Adversarial Networks for Ergodic Image Generation". In: Conférence Sur l'Apprentissage
- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (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

Journal papers

- Eric Laloy, 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?" In: *Computers and Geosciences* 133
- Cyprien Ruffino, Romain Hérault, Eric Laloy, and Gilles Gasso (Nov. 27, 2020). "Pixel-Wise Conditioned Generative Adversarial Networks for Image Synthesis and Completion". In: *Neurocomputing* 416, pp. 218–230

In preparation

- Rachel Blin, Cyprien Ruffino, Samia Ainouz, Romain Hérault, Gilles Gasso, Fabrice Mériaudeau, and Stéphane Canu (2021). "Generating Polarimetric-Encoded Images Using Constrained Cycle-Consistent Generative Adversarial Networks". In: *Currently in Preparation*

Appendix B

Deep learning glossary

Attention

Attention (Vaswani et al., 2017) is, informally, a technique that allows a neural network to "focus" on a subset on the inputs by masking parts of the input vector. The most common type of attention is computed using a dot product: let $\mathbf{x} \in \mathbb{R}^n$ be the input of a neural network layer $l(.)$. Dot product attention consists in weighting the output and $l(\mathbf{x}) \in \mathbb{R}^m$ of the layer l with a feature vector $f_\theta(\mathbf{x}) \in [0,1]^m$, where f_θ is typically a neural network, as

$$l'(\mathbf{x}) = l(\mathbf{x}) \odot f_\theta(\mathbf{x}) \quad . \tag{B.1}$$

Auto-encoder

Auto-encoders AE(.) are a family of neural network architectures that are trained to copy its input to its output, as $\text{AE}(\mathbf{x}) = \mathbf{x}$. Since this is normally a trivial task, auto-encoders are constrained by their architecture (usually an encoder-decoder[2] architecture with a low-dimension latent representation) or through regularization. The aim of auto-encoders is usually to learn a good representation model of the data.

Batch Normalization

Batch Normalization (Ioffe et al., 2015) is a normalization technique that consists in re-centering and re-scaling the input $\mathbf{x}$ of a neural network layer. To approximate the mean and covariance of the full dataset, Batch Normalization computes μ_{mb} the mean and σ_{mb} the variance of the data in present in the mini-batch. It also keeps two parameters, β and γ, updated during the training of the neural network, and uses them to shift and scale the input as

$$\text{BN}(\mathbf{x}|\beta,\gamma) = \gamma \frac{\mathbf{x} - \mu_{mb}}{\sqrt{\sigma_{mb}}} + \beta \quad . \tag{B.2}$$

[2] see Glossary, Appendix B

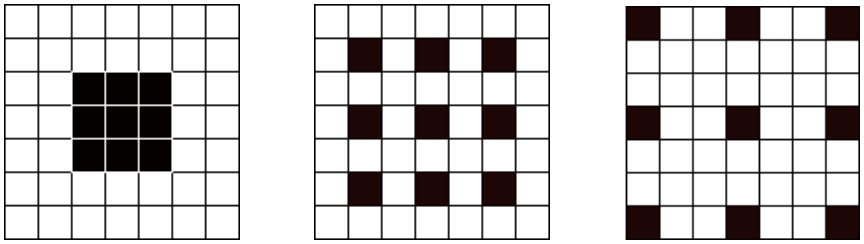


Figure B.1: Dilated filters with dilation rate of 1, 2, 3

Deconvolutional layer / Transposed convolution

Deconvolutions (Long et al., 2015), or transposed convolutions, are the inverse operation of convolutions. As opposed to convolutions in which striding decreases the dimension of the feature maps, deconvolutions allows for filters to upscale the feature maps using striding.

Dilated convolution

Dilated convolutions (Yu and Koltun, 2015) (or "A trous" convolutions) are convolutions in which the size of the filters receptive fields is artificially increased, without increasing the number of parameters, by using sparse filters (see Figure B.1).

Encoder-decoder architecture

An encoder-decoder architecture is a neural network that consists in two parts: the encoder which downscales the input to a small dimension representation; and a decoder which upscales this small dimension representation to obtain a high-dimension output.

Instance Normalization

Instance Normalization (Ulyanov et al., 2016) is a variant of Batch Normalization which does not compute the means and variances on the full mini-batch, but instead standardizes the input using the means and variances for a single input, across all dimensions.

Residual block

A residual block is a set of several layers with a skip-connection that links the first layer of the block to the last one (see Figure B.2). Residual blocks allow for having very deep neural network architectures while mitigating vanishing gradient issues.

Skip connection

Used in auto-encoder architectures and residual blocks, a skip-connection allow for connecting the input of a layer L_n to the input of another layer L_m further up the network (see Figure B.2). This bypassed information can be aggregated to the input of L_m using addition or concatenation. Skip-connections are used to allow for information to flow more easily up the network and is a way to mitigate vanishing gradient problems.

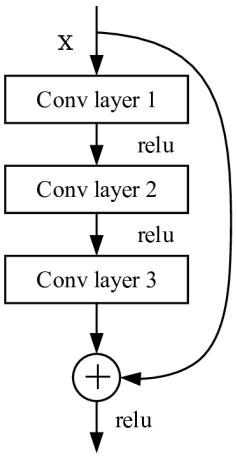


Figure B.2: 3 layer residual block

UNet

A UNet (Ronneberger et al., 2015) is an Encoder-Decoder architecture with skip-connections between layers of the encoder and the decoder.

Appendix C

Experiment details for the Pixel-Wise Conditionned GAN

C.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.

C.2 Detailed deep architectures

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

Table C.1: DCGAN for FashionMNIST

Additional information:

- A Gaussian noise is applied to the input of the discriminator
- Instance normalization[1] (Ulyanov et al., 2016) is applied across all the layers. This is involved by the use of the PacGAN technique.
- The layers noted with an asterisk are linked with a skip-connection[1]

"

[1] see Glossary, Appendix B

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

Table C.2: UNet-Res for CIFAR10

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

Table C.3: UNet-Res for CelebA

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

Table C.4: PatchGAN discriminator

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

Table C.5: 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 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

Table C.6: UpEncDec 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

Table C.7: 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
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

Table C.8: 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

Table C.9: UNet-Res Texture

C.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 C.1 and C.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 2.6).

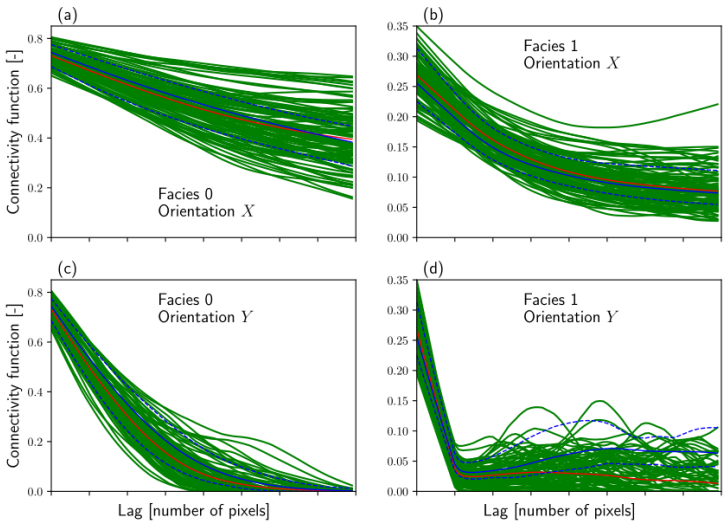


Figure C.1: Connectivity curves obtained on 100 samples generated with the CGAN approach.

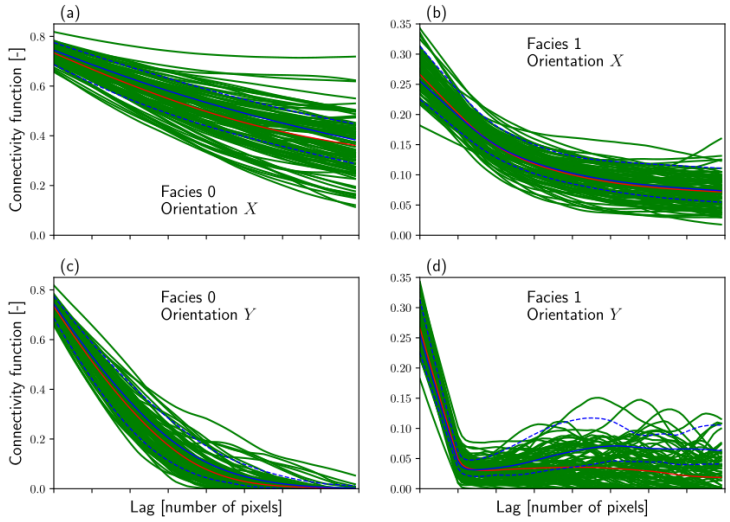


Figure C.2: Connectivity curves obtained on 100 samples generated with our approach.

Appendix D

Proof for the space of polarimetric constraints

Proof. We call $\hat{\mathbf{y}}$ and $\tilde{\mathbf{y}}$ respectively an image that we want to evaluate and the intensities computed by equation (3.5). From this equation, the calibration matrix $\mathbf{A}$ and its pseudo-inverse $\mathbf{A}^\dagger$, we have the following equality:

$$\begin{bmatrix} \tilde{\mathbf{y}}_0 \\ \tilde{\mathbf{y}}_{45} \\ \tilde{\mathbf{y}}_{90} \\ \tilde{\mathbf{y}}_{135} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} \hat{\mathbf{y}}_0 \\ \hat{\mathbf{y}}_{45} \\ \hat{\mathbf{y}}_{90} \\ \hat{\mathbf{y}}_{135} \end{bmatrix} . \tag{D.1}$$

Let $\mathbf{M} = \mathbf{A}\mathbf{A}^\dagger$, then we have:

$$\tilde{\mathbf{y}} = \mathbf{M}\hat{\mathbf{y}} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 \\ 0 & 0 & 2 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \hat{\mathbf{y}}_0 \\ \hat{\mathbf{y}}_{45} \\ \hat{\mathbf{y}}_{90} \\ \hat{\mathbf{y}}_{135} \end{bmatrix} .$$

The set $\mathscr{X}$ such that its elements are solutions to Problem D.1 is

$$\mathscr{X} = \{\mathbf{y} | \mathbf{y} = \mathbf{M}\mathbf{y}\} = \{\mathbf{y} | (\mathbf{M} - \mathbf{I})\mathbf{y} = 0\} = Ker(\mathbf{M} - \mathbf{I}) .$$

Lets first compute the matrix $\mathbf{M} - \mathbf{I}$:

$$\mathbf{M} - \mathrm{I} = \frac{1}{2} \left(\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 \\ 0 & 0 & 2 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} ,$$

with $\mathbf{I}$ the identity matrix. Lets now find $\mathbf{y}$ such that $(\mathbf{M} - \mathbf{I})\mathbf{y} = 0$:

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus we have $\mathbf{y}_1 - \mathbf{y}_2 + \mathbf{y}_3 - \mathbf{y}_4 = 0$. Hence $\mathscr{X}$ comprises vectors $\mathbf{y} \in \mathbb{R}^4$ with the constraint $\mathbf{y}_1 + \mathbf{y}_3 = \mathbf{y}_2 + \mathbf{y}_4$, leading to $\mathscr{X} = \left\{ \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \end{bmatrix}^\top \middle| \ \mathbf{y}_1 + \mathbf{y}_3 = \mathbf{y}_2 + \mathbf{y}_4 \right\}$. □