Generative Adversarial Networks : theory and practice

Ugo Tanielian

«Generative Adversarial Networks » : théorie et pratique

Generative Adversarial Networks: theory and practice

Ugo Tanielian

Laboratoire de Probabilités, Statistique et Modélisation - UMR 8001
Sorbonne Université

Thèse pour l'obtention du grade de :
Docteur de l'université Sorbonne Université

Sous la direction de : Gérard Biau et Maxime Sangnier

Rapportée par : Jérémie Bigot et Arnak Dalalyan

Présentée devant le jury suivant :
Chloé-Agathe Azencott, Mines ParisTech, *Examinateur*
Gérard Biau, Sorbonne Université, *Directeur*
Jérémie Bigot, Université de Bordeaux, *Rapporteur*
Arnak Dalalyan, ENSAE, *Rapporteur*
Patrick Gallinari, Sorbonne Université, *Examinateur*
Jérémie Mary, Criteo AI Lab, *Invité*
Eric Moulines, Ecole Polytechnique, *Président*
Maxime Sangnier, Sorbonne Université, *Directeur*
Flavian Vasile, Criteo AI Lab, *Co-encadrant*

École Doctorale de Sciences Mathématiques de
Paris-Centre
Section Mathématiques appliquées

23/04/2021

Remerciements

Mes premiers remerciements vont à mes directeurs de thèse Gérard, Maxime et Flavian qui m'ont toujours soutenu depuis le début de cette thèse. Je vous remercie pour votre investissement, votre bienveillance et vos connaissances sans lesquels cette thèse n'aurait jamais pu aboutir. Gérard, merci pour les milliers de Skype que l'on a fait, merci pour ton honnêteté toujours alliée d'humour et merci pour ta patience, plus d'une fois mise à rude épreuve! Maxime, merci pour toutes tes remarques minutieuses et nos nombreuses conversations, toujours enrichissantes. Flavian, merci pour ta confiance, tes idées, ton béret et tes chemises hawaïennes. Merci tous les trois pour votre oreille attentive et vos conseils avisés, tant pour mon développement professionnel que personnel. J'ai eu l'immense chance de vous avoir en tant que directeurs et je garderai d'excellents souvenirs de ces dernières années.

De plus, je tiens à remercier Sorbonne Université et tout particulièrement le LPSM pour m'avoir accueilli pendant ces trois années de thèse. Un grand merci pour les bons moments passés avec les membres du labo mais surtout ceux du Politbureau 206: Nicolas, Adeline, Sébastien, Clément et Taieb.

Je tiens également à remercier vivement Criteo pour m'avoir donné la possibilité de réaliser cette thèse Cifre et pour offrir aux chercheurs un cadre de travail incroyable. Quelle chance d'avoir pu évoluer chaque jour avec cette équipe du Criteo AI Lab! Un grand merci tout particulièrement à Anne-Marie Tousch (pour m'avoir recruté), Flavian, Mike, David, Liva, Alain, Eustache, Patrick, Vianney, Benjamin, Alex, Sergey, Martin, Amine, Thomas, Otmane, Lorenzo, Morgane et Matthieu. Au dynamique groupe des thésards. Une mention particulière au duo Clément et l'abeille pour leurs discussions sur la terasse. Egalement, un grand remerciement aux co-fondateurs de l'équipe GAN we do it: Jérémie et Thibaut. C'est vraiment un plaisir de travailler avec vous, j'espère que de nombreux 'best papers' nous attendent! Enfin, un grand merci à Loulou pour ces moments passés à bosser Machine Learning autour du baby-foot: on en aura fait du chemin depuis la 1ère S3.

Mon doctorat n'aurait pas été aussi agréable si je n'avais pas pu partager mes idées avec tous les chercheurs du CAIL et du LPSM. Vous m'avez tant apporté, merci. Tout particulierement, je tiens à remercier Jérémie Bigot et Arnak Dalalyan qui ont accepté de relire ma thèse. Merci pour vos commentaires enrichissants! J'adresse également mes remerciements à Chloé-Agathe

Azencott, Patrick Gallinari, Eric Moulines et Jérémie d'avoir accepté de faire partie de mon jury de soutenance.

Plus personnellement, je tiens également à remercier les taupally et les picheurs. Tels le yin et le yang, ce sont deux forces qui parfois se repoussent mais restent indissociables malgré tout. Je suis plus que chanceux de vous avoir à mes côtés! Ne changez surtout pas! Un remerciement tout particulier aux deux Nico (V & J) pour leurs relectures assidues! Sans oublier bien sur le Phil, le Jood, le Sach', Samoens Escape et tous les autres!

Enfin, un grand merci à toute ma famille! Merci à mes parents pour leurs sacrifices et leur soutien. Cette thèse, c'est clairement grâce à vous. A mes soeurs, qui sont toujours présentes, inconditionnellement. A mon père et ma soeur pour leurs relectures rigoureuses qui m'ont permis de peaufiner l'introduction. A ma grand-mère, dont l'oreille ne pourrait être plus attentive. A mon grand-père Dédé, qui fut le meilleur des profs de maths pendant près de 10 ans. Il reste encore aujourd'hui un fervent critique des réseaux de neurones, sûrement à raison. Enfin, last but not least, une dédicace toute particulière à notre Andréa adorée qui anime nos vies depuis maintenant plus de deux ans.

Abstract

Generative Adversarial Networks (GANs) were proposed in 2014 as a new method efficiently producing realistic images. Since their original formulation, GANs have triggered a surge of empirical studies, and have been successfully applied to different domains of machine learning: video, sound generation, and image editing. However, our theoretical understanding of GANs remains limited. This thesis aims to reduce the gap between theory and practice by studying several statistical properties of GANs. After reviewing the main applications of GANs in the introduction, we introduce a mathematical formalism necessary for a better understanding of GANs. This framework is then applied to the analysis of GANs defined by Goodfellow et al. (2014) and Wasserstein GANs, a variant proposed by Arjovsky et al. (2017), well-known in the scientific community for its strong empirical results. The rest of the thesis attempts to solve two practical problems often encountered by researchers: the approximation of Lipschitz functions with constrained neural networks and the learning of non-connected manifolds with GANs.

Key-words: GANs, generative models, adversarial training, deep learning theory, Wasserstein distance.

Résumé

Les Generative Adversarial Networks (GANs) ont été proposés en 2014 comme une nouvelle méthode pour produire efficacement des images réalistes. Les premiers travaux ont été suivis par de nombreuses études qui ont permis aux GANs de s'imposer dans des domaines variés de l'apprentissage automatique tels que la génération de vidéos, de sons, ou encore l'édition d'images. Cependant, les résultats empiriques de la communauté scientifique devancent largement leurs progrès théoriques. La présente thèse se propose de réduire cet écart en étudiant les propriétés statistiques des GANs. Après avoir rappelé succinctement l'état de l'art dans le chapitre introductif, le second chapitre présente un formalisme mathématique adapté à une

meilleure compréhension des GANs. Ce support théorique est appliqué à l'analyse des GANs définis par Goodfellow et al. (2014). Le troisième chapitre se concentre sur les Wasserstein GANs, variante proposée par Arjovsky et al. (2017), qui s'est imposée dans la communauté scientifique grâce à de très bons résultats empiriques. La suite de la thèse est plus appliquée et apporte des éléments de compréhension à deux problèmes souvent associés aux GANs : d'une part, l'approximation des fonctions Lipschitz avec des réseaux de neurones contraints et, d'autre part, l'apprentissage de variétés non connexes avec les GANs.

Mots-clés: GANs, modèles génératifs, entraînement antagoniste, théorie de l'apprentissage profond, distance de Wasserstein.

Contents

1 Introduction **1**

1.1 Contexte général 1
1.2 Introduction du problème 2
1.3 Tour d'horizon des GANs 9
1.4 Deux problèmes existants dans les GANs 13
1.5 Organisation du manuscrit et présentation des contributions 18

2 Some theoretical properties of GANs **25**

2.1 Introduction 26
2.2 Optimality properties 29
2.3 Approximation properties 34
2.4 Statistical analysis 35
2.5 Conclusion and perspectives 53
Appendix 2.A Technical results 55

3 Some theoretical properties of Wasserstein GANs **61**

3.1 Introduction 62
3.2 Wasserstein GANs 64
3.3 Optimization properties 71
3.4 Asymptotic properties 80
3.5 Understanding the performance of WGANs 85
Appendix 3.A Technical results 91

4 Approximating Lipschitz continuous functions with GroupSort neural networks **111**

4.1 Introduction 112
4.2 Mathematical context 113
4.3 Learning functions with a grouping size 2 115
4.4 Impact of the grouping size 120

4.5 Experiments 122
4.6 Conclusion 127
Appendix 4.A Technical results 127
Appendix 4.B Complementary experiments 134

5 Learning disconnected manifolds: a no GAN's land **139**
5.1 Introduction 140
5.2 Related work 142
5.3 Our approach 143
5.4 Experiments 150
5.5 Conclusion and future work 155
Appendix 5.A Technical results 158
Appendix 5.B Complementary experiments 169
Appendix 5.C Supplementary details 177

Conclusion **181**
5.4 Conclusion on the present thesis 181
5.5 Broader perspectives on GANs 183

References **185**

Chapter 1

Introduction

Contents

1.1 Contexte général 1
1.2 Introduction du problème 2
1.3 Tour d'horizon des GANs 9
1.4 Deux problèmes existants dans les GANs 13
1.5 Organisation du manuscrit et présentation des contributions 18

1.1 Contexte général

De janvier 2018 à décembre 2020, notre travail de recherche a été mené grâce à la collaboration du Criteo AI Lab (CAIL) et du laboratoire LPSM de Sorbonne Université. Criteo est un leader de la French Tech française qui s'est imposé dans l'industrie du digital. Spécialisée dans le ciblage publicitaire sur Internet, l'entreprise a pour cœur de métier l'analyse de très grandes bases de données. Chaque heure, Criteo suggère des dizaines de millions d'annonces publicitaires pour des dizaines de millions d'utilisateurs différents. Il s'agit d'être rapide, précis et efficace. Soucieuse d'améliorer la qualité de ses modèles de recommandation, Criteo développe une activité de recherche au sein de son AI Lab. Nous y étudions la recommandation, mais aussi les bandits manchots et l'efficacité des différents systèmes d'enchères. De son côté, le LPSM est un laboratoire réputé pour ses nombreux travaux dans le domaine des statistiques paramétriques, celui de l'apprentissage statistique et des valeurs extrêmes. Travailler pendant trois années au sein de ces deux institutions a été, pour moi, une expérience passionnante.

La présente thèse porte sur l'analyse théorique des Generative Adversarial Networks (GANs), un algorithme récent mais très prometteur. Depuis sa publication en 2014, le modèle

proposé par Goodfellow et al. (2014) a été largement étudié, modifié et amélioré, comme en témoignent les 25 000 citations sur Google scholar. Ian Goodfellow, son *inventeur*, est désormais devenu un pilier du Machine Learning. Pour la seule année 2018, plus de 11 000 publications ont traité du sujet des GANs, soit une trentaine quotidiennement. Le rapide succès des GANs s'est opéré dans des domaines divers et variés. Ce prompt déploiement s'explique par leur définition simple, leur utilisation ludique et leurs résultats saisissants.

En revanche, comme c'est souvent le cas dans le domaine de l'intelligence artificielle, les résultats empiriques de la communauté scientifique devancent largement leurs progrès théoriques. En effet, de nombreuses interrogations subsistent sur la compréhension théorique et bien des sujets restent encore inexplorés. Etant donné les importantes applications des GANs dans des domaines très visuels, la communauté scientifique a priorisé la performance empirique au détriment de la connaissance théorique. Six ans après la première publication sur le sujet, il existe de nombreuses architectures différentes pour entraîner un GAN mais aucune méthode d'évaluation fiable pour les comparer. Partant de cette observation, l'objectif de la thèse est donc de progresser vers une meilleure compréhension de cet algorithme et des enjeux qu'il représente. Pour mener ce projet à bien, les recherches se sont portées sur deux domaines distincts. Au LPSM, nous nous sommes concentrés sur une étude probabiliste et statistique tournée vers l'objectif d'élargir le formalisme mathématique des GANs. Au CAIL, la conception plus appliquée de la recherche nous a mené à examiner des problèmes concrets propres à l'entraînement des GANs. Cette double facette théorique et pratique a été à la fois enrichissante et prolifique - le formalisme permettant de mieux appréhender les problèmes.

1.2 Introduction du problème

1.2.1 Du Deep Learning aux Generative Adversarial Networks

Le début des années 2010 a marqué un véritable tournant pour le développement de l'apprentissage automatique (Machine Learning). D'un côté, les systèmes d'information des entreprises se sont améliorés, augmentant considérablement le nombre de données à disposition. D'un autre côté, la capacité de stockage et la puissance de calcul des ordinateurs a énormément progressé, facilitant le traitement de ces données. Cette conjonction entre l'augmentation de la quantité de données disponible et l'amélioration de traitement de ces mêmes données s'est traduite par une progression considérable des algorithmes de Machine Learning. Tombé en désuétude pendant plusieurs années, l'apprentissage profond (Deep Learning) a refait son entrée sur le devant de la scène. Dopés par ce surplus de données, les réseaux de neurones profonds se sont révélés particulièrement efficaces pour la résolution de problèmes complexes, dépassant tous les autres

algorithmes concurrents (modèles linéaires généralisés, forêts aléatoires, arbre de décision, machines à vecteurs de supports, etc.).

Le Deep Learning s'est montré extrêmement bénéfique dans le domaine de la classification multi-classe qui s'attache à distinguer des objets appartenant à différentes catégories. De nombreuses études empiriques ont montré l'efficacité de ces réseaux de neurones notamment sur des jeux de données complexes où la dimension des objets est grande. Dans le domaine de l'analyse d'images par exemple, (par example le jeu de données MNIST (LeCun et al., 1998) ou ImageNet (Krizhevsky et al., 2012)), les meilleurs modèles sont exclusivement des réseaux à convolution. La force de ces algorithmes est qu'il n'est maintenant plus nécessaire de traiter préalablement les données et de sélectionner les variables (feature engineering) puisque les modèles profonds façonnent automatiquement leurs propres variables. De manière plus informelle, l'abandon de la sélection manuelle des variables au profit de l'utilisation des modèles plus profonds est analysée avec humour par Frederick Jelinek : "Every time I fire a linguist, my performance goes up".

En revanche, le développement de modèles génératifs a connu un progrès plus tardif. Cela est principalement dû au fait que les méthodes d'entraînement existantes telles que l'estimation de densité n'étaient pas réalisables sur des données de grande dimension comme des images. Il a fallu attendre l'année 2014 et le développement de l'entraînement antagoniste proposé par les Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) pour voir émerger des réseaux de neurones capables de générer des images de haute qualité et extrêmement réalistes.

1.2.2 Présentation succincte des GANs

Les GANs (Goodfellow et al., 2014) font partie de la famille des modèles génératifs. A partir d'un ensemble de données, il s'agit d'être capable de générer des objets similaires sans pour autant qu'ils soient identiques à ceux déjà existants. Dans le contexte de visages humains, l'objectif des GANs est donc de générer des photos à la fois réalistes, uniques et diverses. Deux exemples des résultats obtenus par Karras et al. (2019) sont exhibés dans la Figure 1.1. Nous constatons que les résutats visuels sont impressionnants.

Les GANs se composent de deux fonctions paramétriques : le générateur et le discriminateur. En pratique, les modèles utilisés sont des réseaux de neurones - qu'ils soient à propagation avant (feed-forward), convolutionnels ou récurrents selon les applications. L'objectif du générateur est de créer les meilleures images possibles : prenant un bruit en entrée (Gaussien ou uniforme) il le transforme dans l'espace des images. Pour être correctement défini, le générateur nécessite donc un espace latent sur lequel une distribution est définie : c'est la distribution latente. Le discriminateur, quant à lui, apprend à distinguer les *fausses* images produites par le générateur des *vraies* données disponibles dans le jeu d'entraînement. Même si le discriminateur joue

Fig. 1.1 Exemples de visages humains générés à partir de la structure proposée par Karras et al. (2019). Source :thispersondoesexist.com.

un rôle de support, il n'en demeure pas moins essentiel car il transmet au générateur les informations nécessaires et suffisantes pour qu'il s'améliore.

Du point de vue de l'optimisation, le générateur essaie de tromper le discriminateur tandis que le discriminateur est entraîné de manière supervisée : il prend en entrée des images vraies et fausses et essaie de les classifier correctement. L'ensemble de cette structure est illustrée dans la Figure 1.2.

Du point de vue probabiliste, le générateur transfère la distribution latente sur l'espace d'arrivée et définit donc une mesure image. Le but des GANs est alors d'approcher la distribution cible à l'aide de cette mesure image. Quant au discriminateur, nous verrons plus tard qu'en discriminant entre les images vraies et fausses, il définit également une distance (ou divergence) entre les deux distributions de probabilité que sont la distribution cible et la distribution générée.

En pratique, à la fois le générateur et le discriminateur sont paramétrés par des réseaux de neurones. En fonction des domaines d'application et des tâches à réaliser, de nombreuses paramétrisations différentes ont été proposées pour l'entraînement. En ce qui concerne la génération d'images, c'est l'architecture DCGAN (Radford et al., 2015) qui a été largement répandue dans la communauté scientifique : cette dernière correspond en une simple série de convolutions pour le générateur. Gulrajani et al. (2017) propose l'utilisation de réseaux résiduels (He et al., 2016) pour améliorer la qualité des images générées. D'un point de vue purement qualitatif, c'est la structure proposée par Karras et al. (2019) qui a permis une véritable amélioration. Au lieu d'apprendre directement la transformation, Karras et al. (2019) proposent de rajouter un réseau de neurones à propagation avant (feedforward neural network)

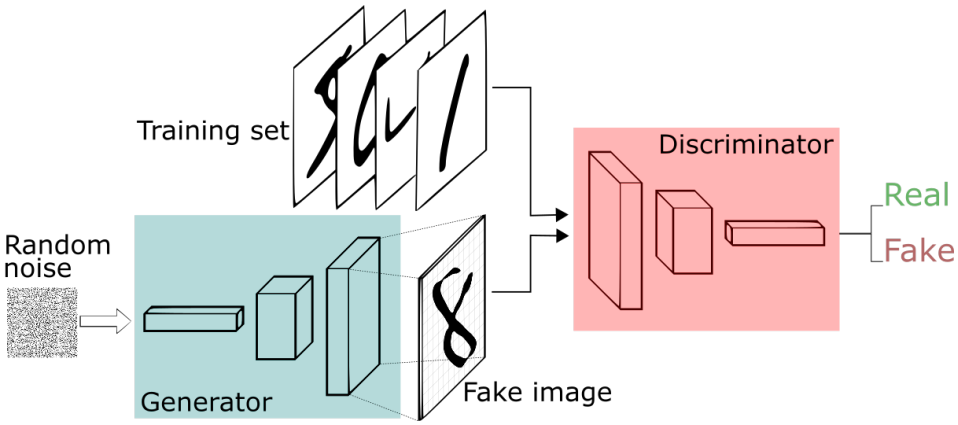


Fig. 1.2 Exemple d'architecture classique d'un GAN entraîné sur le jeu de données de digits MNIST. Source : Trending in AI capabilities.

afin d'intégrer une distribution latente plus complexe et mieux adaptée à la génération de visages humains.

Du fait de l'opposition entre le générateur et le discriminateur, l'entraînement des GANs est complexe et peut aboutir à des solutions non optimales. Goodfellow et al. (2014) ont confirmé que les gradients du discriminateur s'amenuisent lorsque celui-ci s'approche de l'optimalité. La procédure par gradients alternés utilisée pour entraîner les GANs complique la détection de convergence. Mertikopoulos et al. (2018) relèvent en effet que des cycles peuvent se répéter indéfiniment. Goodfellow et al. (2014) et Salimans et al. (2016) se sont rendus compte dès les premières études empiriques que le générateur pouvait finir par concentrer toute sa masse sur une portion minime de la distribution cible : c'est le phénomène de perte de modes (mode collapse). Dans le cas où la distribution cible est multimodale, cela signifie que le générateur ignore certains de ces modes. Il finit donc par générer un petit ensemble d'images très réalistes mais peu diversifiées. Comme nous le verrons par la suite, une grande partie des chercheurs tentent de comprendre et de minimiser ce phénomène.

1.2.3 Les divers domaines d'application des GANs

L'efficacité des GANs s'est d'abord révélée dans la génération d'images. Karras et al. (2018, 2019) ont perfectionné la génération de visages humains allant jusqu'à générer des images $1024x1024$ pixels. Brock et al. (2019) ont étendu cette réussite au jeu de données complexe ImageNet contenant plus de 1000 classes distinctes. Néanmoins, il est important de souligner que les GANs se sont révélés également efficaces pour toutes sortes de tâches qui dépassent largement le domaine de la génération d'images. Afin de mieux saisir l'engouement scientifique

créé par les GANs, la sous-section qui suit présente, succinctement et simplement, leurs différents domaines d'application.

L'analyse d'images. La littérature portant sur l'analyse d'images à partir des GANs est extrêmement variée. Shen et al. (2020) ont souligné comment les GANs pouvaient faciliter l'édition d'images. En se déplaçant selon certaines directions de l'espace latent, la Figure 1.3 illustre comment, partant d'un visage initial, il est possible de le vieillir, lui rajouter des lunettes ou changer son genre. Yi et al. (2017) sont parvenus à modifier une image en lui donnant le style d'un tableau ou d'une photo. Reed et al. (2016) ont appliqué les GANs à la génération d'images à partir d'un texte descriptif. Enfin, Ledig et al. (2017) ont décrit comment restaurer des images floutées en haute résolution avec une efficacité surpenante.

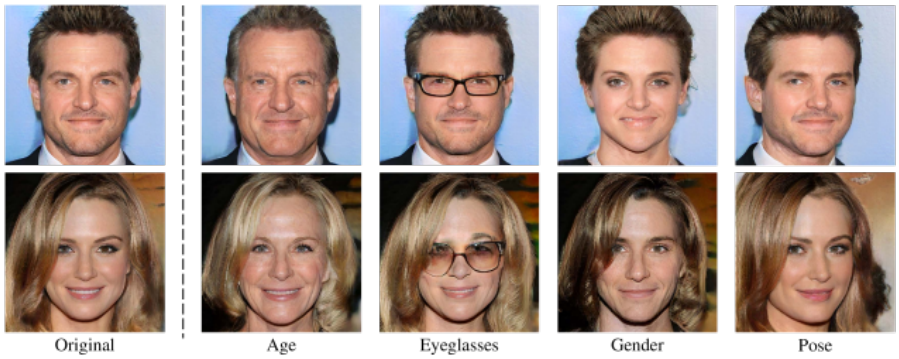


Fig. 1.3 Exemple d'éditions d'images en se déplaçant simplement dans certaines directions de l'espace latent. Source : Shen et al. (2020).

La génération de vidéos. Au delà de l'analyse d'images, les GANs ont été utilisés avec succès dans différents domaines de recherche. S'appuyant sur les récents progrès réalisés en analyse vidéo et en particulier la convolution 3D (Ji et al., 2013), les GANs se sont révélés particulièrement efficaces dans la génération de vidéos (Vondrick et al., 2016; Saito et al., 2017; Tulyakov et al., 2018) comme l'illustre la Figure 1.4.

Améliorer la robustesse des algorithmes de Deep Learning. En 2014, la communauté scientifique s'est rendue compte que les modèles profonds pouvaient facilement être dupés. S'ils sont performants dans le domaine de la classification supervisée, leurs prédictions peuvent être faussées par une perturbation aussi minime soit-elle (Goodfellow et al., 2015) : ce sont des "attaques adverses". Un exemple frappant est celui proposé par Su et al. (2019) qui ont réussi à tromper des réseaux de neurones en ne modifiant qu'un seul pixel. Une branche de

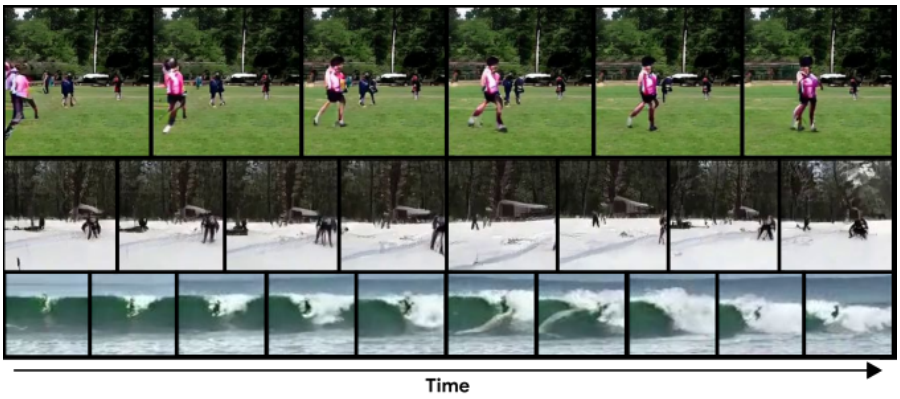


Fig. 1.4 Exemples de générations de vidéos à l'aide des GANs. Source : Clark et al. (2019).

la recherche s'est alors concentrée à améliorer la robustesse des réseaux profonds face à ces attaques adverses. Pour réaliser cette tache, les GANs se sont révélés très utiles.

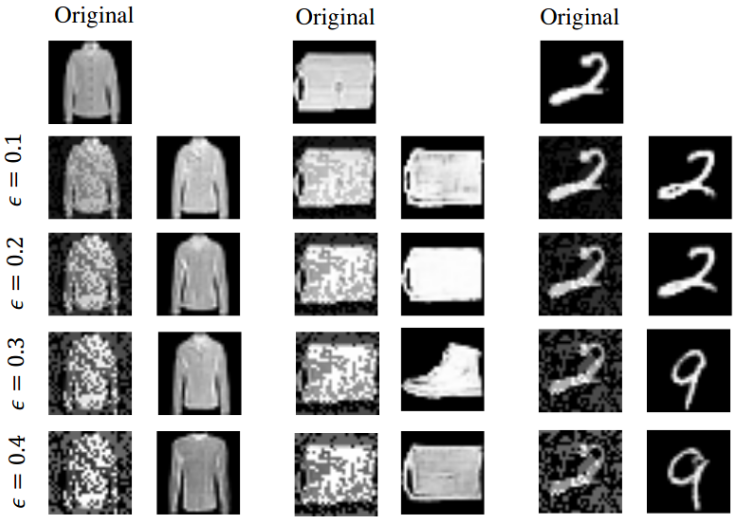
Tout d'abord, après avoir entraîné un GAN sur un jeu de données d'entraînement, le générateur peut maintenant étendre ce jeu d'entraînement, fournir un ensemble infini d'exemples supplémentaires labélisés permettant d'améliorer la généralisation du modèle. Ensuite, les GANs peuvent être spécifiquement utilisés pour permettre à des classifieurs extérieurs d'observer des exemples complexes sur lequel le classifieur est indécis. Xiao et al. (2018) ont utilisé les GANs pour générer directement les attaques adverses et faciliter l'amélioration du classifieur. Prenant un angle d'attaque différent, Samangouei et al. (2018) ont adopté les GANs comme moyen de défense : avant de faire une prévision avec le classifieur, chaque point de donnée corrompu est projeté sur la variété apprise par le GAN. Quelques exemples pour les jeux de données MNIST et Fashion-MNIST sont montrés dans la Figure 1.6a. Dans ce cas précis, le GAN peut être utilisé sur n'importe quel type de classifieurs et ce dernier n'a même pas besoin d'être ré-entraîné. Enfin, dans le domaine de la classification multi-classe, les GANs permettent aussi de générer des points dans les zones complexes où la donnée est plus rare. La Figure 1.6b illustre la faculté du GAN à produire des points au niveau de la frontière entre deux classes.

Comme nous pouvons le constater la faculté générative des GANs est tour à tour une finalité, quand il s'agit de produire des images ou des vidéos, ou bien un moyen, quand il s'agit de rendre plus robustes certains algorithmes.

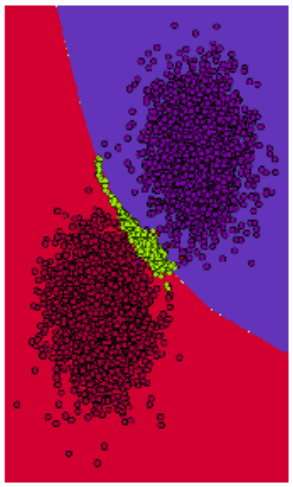
Le langage. Le langage (ou NLP, Natural Language Processing) est l'un des domaines où l'utilisation des GANs n'est pas directe. En effet, dans leur formulation initale, l'entraînement des GANs nécessite de pouvoir calculer les gradients de la sortie du générateur. Dans le domaine discret, dont fait partie le traitement naturel du langage, cette dernière opération n'est pas possible. En revanche, en apportant quelques modifications, il devient possible

(a) Source : Su et al. (2019).

Fig. 1.5 Exemple d'attaque adverse perturbant considérablement la réponse du réseau de neurones alors que seulement un pixel de l'image d'entrée a été modifié.



(a) Source : Samangouei et al. (2018).



(b) Source : Sun et al. (2019).

Fig. 1.6 Exemple d'utilisation des GANs dans le domaine de la robustesse des réseaux profonds. A gauche, les images corrompues sont projetées sur la variété apprise par le GAN. A droite, le GAN vient sampler au niveau de la frontière entre les deux classes pour diminuer l'indécision du classifieur.

de contourner ce problème. Kusner and Hernández-Lobato (2016) ont proposé d'utiliser un algorithme d'échantillonage basé sur une distribution de Gumbel. Yu et al. (2017) et Che et al. (2017) ont proposé une fonction de coût inspirée de l'apprentissage par renforcement (Reinforcement Learning). Ils suggèrent d'utiliser le discriminateur comme un agent externe et entraîne le générateur via policy gradient (Sutton et al., 2000).

1.3 Tour d'horizon des GANs

1.3.1 Contexte mathématique

Précisons tout d'abord le contexte mathématique dans lequel se place les Generative Adversarial Networks. Comme nous l'avons précisé précedemment, l'objectif des GANs est de pouvoir approcher avec un modèle paramétrique, une distribution cible, inconnue. Pour le reste de l'étude, cette dernière sera notée $\mu_\star$. Elle est définie sur un espace métrique $\mathbb{R}^D$, dont la dimension peut-être très grande : c'est notamment le cas de la génération d'images en haute résolution. L'espace de départ (espace latent) est également un espace métrique $\mathbb{R}^d$ dont la dimension est en pratique nettement plus petite que cele de l'espace d'arrivée. Cet espace latent est muni d'une variable aléatoire latente Z de mesure γ. Il s'agit le plus souvent d'une gaussienne multivariée ou de la mesure uniforme sur $[-1,1]^d$.

Formellement, le générateur est paramétré par une classe de fonctions mesurables de l'espace latent $\mathbb{R}^d$ dans l'espace d'arrivée $\mathbb{R}^D$, on note

$$\mathscr{G} = \{G_\theta : \theta \in \Theta\}, \quad \text{où} \quad \Theta \subseteq \mathbb{R}^P,$$

l'ensemble des paramètres décrivant le modèle. Chaque function G_θ prend en entrée un vecteur dans $\mathbb{R}^d$ échantilloné par Z et renvoie une fausse observation $G_\theta(Z)$ dont la loi est notée μ_θ. Par conséquent, la collection de mesures images $\mathscr{P} = \{\mu_\theta : \theta \in \Theta\}$ est la classe naturelle des distributions associée avec le générateur. Quant au discriminateur, il est décrit par une classe de fonctions mesurables de $\mathbb{R}^D$ dans $\mathbb{R}$, notée

$$\mathscr{D} = \{D_\alpha : \alpha \in \Lambda\}, \quad \text{où} \quad \Lambda \subseteq \mathbb{R}^Q$$

correspond à l'ensemble des paramètres du discriminateur. L'objectif des GANs est de trouver au sein de cette famille de distributions celle qui est la plus proche de la distribution cible $\mu_\star$ selon le critère donné par le discriminateur.

1.3.2 Les fonctions de coût

GANs originels. Dans leur définition initiale, Goodfellow et al. (2014) proposent les GANs comme une manière originale d'entraîner deux réseaux de manière antagoniste: le générateur cherche à tromper le discriminateur qui, quant à lui, cherche à classifier le vrai du faux. Considérons une variable aléatoire Y à valeurs dans $\{0,1\}$ et notons $X|Y=1$ la variable aléatoire de distribution $\mu_\star$ et $X|Y=0$ la variable aléatoire de distribution μ_θ. Alors l'objectif du discriminateur est le suivant :

$$D_\alpha(X) = \mathbb{P}(Y=1|X).$$

En choisissant le discriminateur comme une classe de fonctions mesurables, paramétriques à valeurs dans $[0,1]$, les auteurs définissent l'objectif plus général des GANs comme suit:

$$\inf_{\theta\in\Theta}\sup_{\alpha\in\Lambda}\mathbb{E}\log(D_\alpha(X|Y=1)) + \mathbb{E}\log(1-D_\alpha(X|Y=0)), \tag{1.3.1}$$

où le symbole $\mathbb{E}$ fait référence à l'espérance. Pour mieux comprendre cette fonction de coût, plaçons nous dans le contexte spécifique où :

1. les distributions $\mu_\star$ et μ_θ sont absolument continues par rapport à la mesure de Lebesgue μ. Notons respectivement $p_\star$ et p_θ leur densités par rapport à μ.
2. l'ensemble des fonctions discriminatives correspond à la classe non paramétrique $\mathscr{D}_\infty$ des fonctions mesurables de $\mathbb{R}^D$ dans $[0,1]$.

Dans ce cas précis, nous pouvons montrer que le problème des GANs revient à résoudre

$$\inf_{\theta\in\Theta} D_{\mathrm{JS}}(\mu_\star,\mu_\theta), \tag{1.3.2}$$

où D_{JS} correspond à la divergence de Jensen-Shannon définit comme suit :

$$D_{\mathrm{JS}}(\mu_\star,\mu_\theta) = \int p_\star \ln\big(\frac{2p_\star}{p_\star+p_\theta}\big)\mathrm{d}\mu + \int \frac{\mathrm{p}_\star+\mathrm{p}_\theta}{2}\ln\big(\frac{\mathrm{p}_\star+\mathrm{p}_\theta}{2\mathrm{p}_\star}\big)\mathrm{d}\mu.$$

Etant donné les propriétés d'approximation universelle des réseaux de neurones, nous comprenons bien le rôle joué par le discriminateur : c'est une approximation paramétrique de la divergence de Jensen-Shannon.

En modifiant la fonction de discrimination utilisée dans (1.3.1), Nowozin et al. (2016) et Mao et al. (2017) montrent que le problème des GANs peut s'étendre à l'objectif suivant :

$$\inf_{\theta\in\Theta} D_f(\mu_\star,\mu_\theta), \tag{1.3.3}$$

où $D_f(\mu_\star, \mu_\theta) = \int p_\star(x) f\left(\frac{p_\star(x)}{p_\theta(x)}\right) \mathrm{d}\mu(\mathrm{x})$ correspond à la f-divergence entre $\mu_\star$ et μ_θ.

Le défaut général des formulations impliquant des f-divergences est qu'elles nécessitent de fortes hypothèses. En effet, la f-divergence $D_f(\mu_\star, \mu_\theta)$ n'est définie que si l'on suppose la distribution μ_θ absolument continue par rapport à la distribution $\mu_\star$. En pratique, Arjovsky and Bottou (2017, Theorem 2.2) a montré qu'il est fort probable qu'en grande dimension, $\mu_\star$ et μ_θ ne soit pas absolument continue par rapport à la même mesure de base. Roth et al. (2017) appellent ce phénomène une erreur de dimensionnalité (dimensionality mispecification): la variété cible et la variété générée n'ont dans ce cas pas la même dimension. Dans ce cas précis, Arjovsky and Bottou (2017) ont montré que lorsque le discriminateur se rapproche de l'optimalité, les gradients renvoyés au générateur sont soit nuls, soit instables; empêchant l'apprentissage de la distribution cible et facilitant l'apparition du phénomène de perte de modes.

IPM GANs. Pour s'attaquer aux problèmes présentés ci-dessus, il est possible d'utiliser une différente famille de distances entre distributions de probabilité qui nécessite des hyptohèses plus faibles : ce sont les Integral Probability Metric (IPM) (Müller, 1997). Etant donné une classe de fonctions mesurables $\mathscr{F}$ définie de $\mathbb{R}^D$ dans $\mathbb{R}$, on définit l'IPM entre deux distributions de probabilité μ et ν de $\mathbb{R}^D$, comme suit :

$$d_{\mathscr{F}}(\mu, \nu) = \sup_{f \in \mathscr{F}} \mathbb{E}_\mu f - \mathbb{E}_\nu f. \tag{1.3.4}$$

Pour être définie, la distance $d_{\mathscr{F}}(\mu, \nu)$ ne nécessite que des hypothèses de finitude des moments sur les distributions de probabilité μ et ν. Les IPMs vérifient la propriété de symétrie $d_{\mathscr{F}}(\mu, \nu) = d_{\mathscr{F}}(\nu, \mu)$ ainsi que l'inégalité triangulaire $d_{\mathscr{F}}(\mu, \nu) \leqslant d_{\mathscr{F}}(\mu, \eta) + d_{\mathscr{F}}(\eta, \nu)$ (pour toute distribution de probabilité η). Elles sont fréquemment rencontrées en machine learning, notamment la distance de 1-Wasserstein W qui, en utilisant sa forme duale, s'écrit comme une IPM (Villani, 2008) :

$$W(\mu, \nu) = \sup_{f \in \mathrm{Lip}_1} \mathbb{E}_\mu f - \mathbb{E}_\nu f = d_{\mathrm{Lip}_1}(\mu, \nu), \tag{1.3.5}$$

où Lip_1 correspond à l'ensemble des fonctions 1-Lipschitz.

Pour corriger les défauts des f-GANs, Arjovsky et al. (2017) définissent les Wasserstein GANs comme une manière de minimiser la distance de Wasserstein entre la distribution cible $\mu_\star$ et la distribution modélisée μ_θ. Le nouveau problème des GANs devient :

$$\inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta), \tag{1.3.6}$$

En revanche, étant donné que la classe des fonctions 1-Lipschitz n'est pas paramétrable, les auteurs approximent cette dernière par un critique (ou discriminateur) paramétré par un réseau de neurones. Le véritable objectif des WGANs se formule comme suit :

$$\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \mathbb{E}_{\mu^\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha = \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta), \tag{1.3.7}$$

où $d_{\mathscr{D}}(\mu_\star, \mu_\theta) = \sup_{\alpha\in\Lambda} \mathbb{E}_{\mu^\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha$ correspond à l'IPM générée par $\mathscr{D}$. Comme l'illustre la Figure 1.7, Arjovsky et al. (2017) montrent l'intérêt de cette formulation en justifiant qu'elle stabilise l'entraînement des GANs : les gradients du discriminateur ne s'annulent pas. Au contraire, Gulrajani et al. (2017, Theorem 1) montrent que la norme du gradient du discriminateur optimal est égale à 1 presque partout sur chaque ligne du transport optimal.

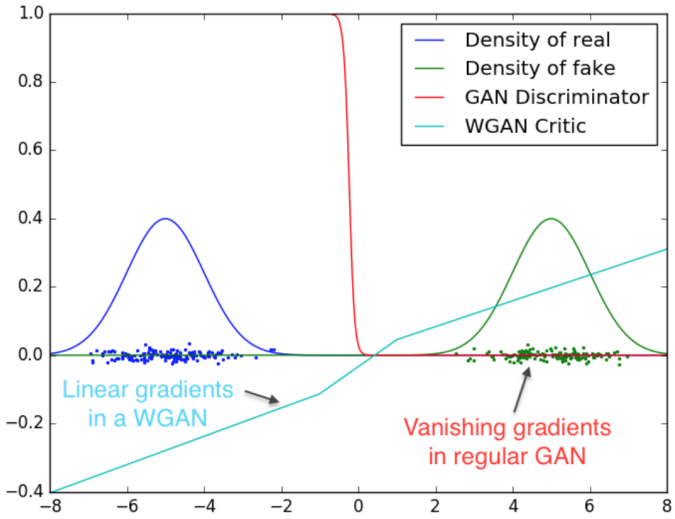


Fig. 1.7 Comparaison entre un discriminateur GAN optimal de classification et un discriminateur (critique) optimal WGAN (en bleu). On observe, en effet, que les gradients du discriminateur en rouge sont nuls presque partout contrairement au critique WGAN. Source : Arjovsky et al. (2017).

Il est important de noter qu'en jouant sur différentes classes paramétriques de fonctions, divers objectifs peuvent être proposés. Dans Li et al. (2015, 2017), le discriminateur $\mathscr{D}$ approxime la boule unité dans un espace de Hilbert à noyau reproduisant (RKHS, Reproducing Kernel Hilbert Space). Mroueh and Sercu (Fisher GANs, 2017) imposent des contraintes sur le moment d'ordre 2 du discriminateur et proposent un objectif qui approxime la distance du Khi-deux χ_2 (Mroueh and Sercu, 2017, Theorem 2).

Les formulations proposées en (1.3.1) et (1.3.6) repose donc sur une minimisation de distance (ou pseudo-distaces) paramétriques. Arora et al. (2017) parlent de distances neuronales

(neural net distances). Liu et al. (2017) font référence à des divergences adverses. Cette caractérisation des GANs comme minimisation de distances neuronales est à la base de notre réflexion.

Régularisation d'un GAN. Dans le cadre des WGANs, pour contraindre le discriminateur a une classe de fonctions 1-Lipschitz, Arjovsky et al. (2017) proposent de restreindre les poids du discriminateur (*weigth clipping*). Néanmoins, il existe d'autres manières plus efficaces pour implémenter cette contrainte sur le gradient du discriminateur. Gulrajani et al. (2017) ajoutent à la fonction de pertes, une pénalisation sur le gradient du discriminateur :

$$\inf_{\theta\in\Theta}\sup_{\alpha\in\Lambda}\mathbb{E}_{\mu^\star}D_\alpha-\mathbb{E}_{\mu_\theta}D_\alpha+\lambda\mathbb{E}_{\hat{\mu}}(\|\nabla_\alpha D_\alpha-1\|)^2, \tag{1.3.8}$$

où $\hat{\mu}$ est la distribution associée à la variable aléatoire $\hat{X}=\varepsilon X+(1-\varepsilon)G_\theta(Z)$ ($X\sim\mu_\star$ and $Z\sim\gamma$). Miyato et al. (2018), quant à eux, normalisent la norme spectrale des matrices apprises tandis que Anil et al. (2019) proposent de projeter chaque matrice de poids sur une boule unité en utilisant l'orthonormalisation de Björck (Bjorck and Bowie, 1971). Empiriquement, la régularisation du discriminateur a permis une amélioration significative de l'entraînement des GANs. Roth et al. (2017) ont montré que régulariser le gradient du discriminateur pouvait également améliorer les f-GANs. Kodali et al. (2017) ont, quant à eux, souligné le fait que l'utilisation de cette régularisation permettait de diminuer le nombre des minimums locaux associés à la perte de modes. La régularisation des GANs est maintenant largement utilisée.

1.4 Deux problèmes existants dans les GANs

1.4.1 L'apprentissage de variétés non connexes

Comme vu précédemment, dans leur formulation standard, les GANs sont définis comme la mesure image d'une distribution le plus souvent unimodale par un générateur continu. Il est alors facile de montrer que, dans ce cas précis, la loi apprise μ_θ aura un support connexe dans l'espace d'arrivée $\mathbb{R}^D$. Par conséquent, quand la distribution cible est complexe et à support sur une variété non connexe, Khayatkhoei et al. (2018) ont montré que, dans ce cas, les GANs pêchent par le problème suivant :

- soit le générateur concentre sa masse sur l'un des modes de la distribution cible et produit des points hautement réalistes mais très peu diversifiés : c'est le cas de la perte de modes.

- soit le générateur essaie de couvrir le plus de modes possibles et met, de ce fait, de la masse là où la distribution cible n'en met pas (entre deux modes). Le générateur est, dans ce cas précis, nécessairement amené à produire certains points de très faible qualité.

Pour résoudre ce problème, certaines recherches se sont concentrées sur le développement d'architectures qui améliorent l'apprentissage de lois au support non connexe. Cela sous-tend la question suivante : comment faire en sorte que la distribution apprise puisse avoir un support non connexe ?

Ensemble de GANs. Gurumurthy et al. (2017) transforment la distribution latente unimodale en un mélange de gaussiennes, ce qui permet de plus facilement gérer le cas où les données d'apprentissage sont non connexes, diverses et limitées. Au lieu de sur-paramétrer la distribution latente, Tolstikhin et al. (2017) proposent d'entraîner un mélange de générateurs suivant la méthode d'Adaboost. Egalement, Khayatkhoei et al. (2018) entraînent une famille de générateur mais, dans le but précis d'empêcher la perte de modes. En maximisant l'entropie croisée, chacun des générateurs du mélange se spécialise dans l'apprentissage de l'un des modes de la loi cible. Enfin, il faut bien entendu préciser que, si ces méthodes permettent d'améliorer significativement l'apprentissage de variétés non connexes, cela se fait avec un coût computationnel considérablement augmenté. Pour éviter cela, une série de travaux de recherches prend le parti, non pas de modifier la méthode d'entraînement des GANs, mais plutôt de sélectionner les points générés notamment à l'aide de méthodes de Monte-Carlo (Azadi et al., 2019; Turner et al., 2019).

GANs conditionnels. Pour améliorer la génération d'images au sein de jeux de données complexes avec un nombre important de classes différentes, plusieurs auteurs ont proposé l'utilisation des GANs conditionnels (Mirza and Osindero, 2014). Dans ce cas précis, la génération d'une image est conditionnée à la fois à un bruit gaussien et à une classe donnée, comme le montre la Figure 1.8. Brock et al. (2019) appliquent cette même méthode pour générer des images de haute qualité sur le jeu de données de grande dimension qu'est ImageNet (Krizhevsky et al., 2012). La génération conditionnée permet également de transformer la distribution cible au support non connexe en une famille de lois plus simples, au support connexe, et donc plus facilement approchable par un GAN. Pour réduire la perte de modes dans ce schéma précis, Chongxuan et al. (2017) couplent un GAN conditionnel avec un troisième réseau qui apprend la distribution conditionnelle.

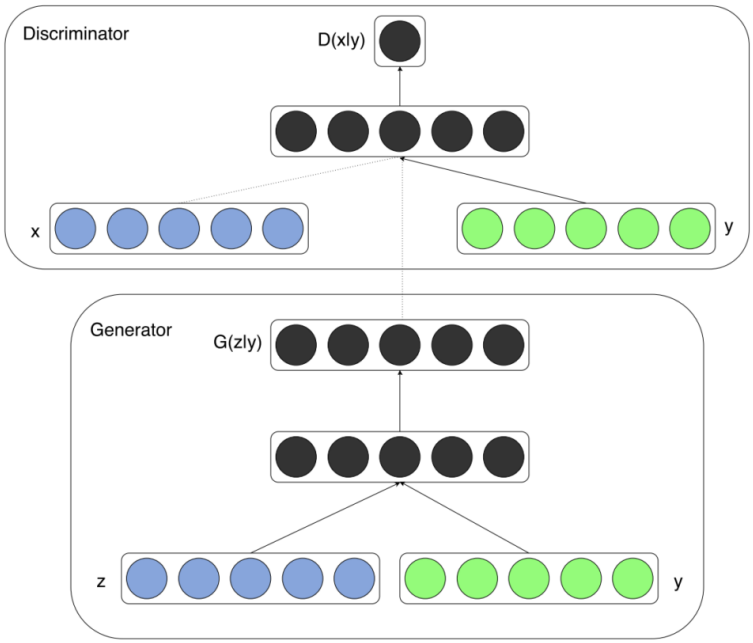


Fig. 1.8 Architecture d'un GAN conditionnel Source : Mirza and Osindero (2014).

1.4.2 L'évaluation des GANs : une question ouverte.

L'évaluation des GANs est toujours une question ouverte et complexe. La principale raison est due au fait, qu'à ce jour, le but final des GANs n'a pas encore été clairement défini. Selon les tâches, les méthodes d'évaluation peuvent donc varier. Le lecteur intéressé pourra se référer à l'étude menée par Borji (2019) qui présente une liste de 25 différentes méthodes d'évaluations des GANs. L'auteur de l'étude souligne lui-même qu'il n'y a à ce jour "pas de consensus quant à la mesure qui capturerait le mieux les forces et les limites d'un GAN et qui devrait être utilisée pour une comparaison équitable des différents modèles".

Il est clair qu'en fonction des différents objectifs choisis et/ou des différentes paramétrisations du discriminateur, les optimums globaux vérifiant l'équation (1.3.7), ne seront pas certainement pas identiques. La question de la comparaison des différents modèles génératifs μ_θ obtenus se pose. Par souci d'équité, ces différents modèles ne peuvent être comparés par exemple ni sur la divergence de Jensen-Shanon ou la distance de Wassertein, ce qui favoriserait respectivement les GANs standards (Goodfellow et al., 2014) ou les WGANs (Arjovsky et al., 2017). Lucic et al. (2018) ont mené une étude empirique importante comparant une grande variété de GANs différents. Ils concluent que la comparaison des différents modèles obtenus doit se faire sur un terrain neutre tel que l'Inception Score ou la distance de Fréchet (étudiés plus bas). Ils montrent que la plupart des modèles peuvent obtenir des scores similaires après avoir

joué sur les hyper-paramètres. De manière similaire, l'étude empirique menée par Mescheder et al. (2018) montre qu'aucun objectif n'est stabilisé sensiblement plus l'entraînement des GANs.

La mesure d'évaluation ne doit pas reposer sur des densités de probabilité. L'un des principaux problèmes des mesures d'évaluation des GANs réside dans le fait qu'elles ne peuvent pas reposer sur les densités de probabilité. Tout d'abord, la mesure cible est inconnue. Ensuite, il est fort possible que les mesures $\mu_\star$ et μ_θ ne soient pas absolument continue par rapport à la mesure de Lebesgue. Pour résoudre ce problème, certaines études proposent l'utilisation d'un 3ème réseau qui agit comme un juge. Par exemple, Salimans et al. (2016); Heusel et al. (2017) proposent d'utiliser InceptionNet (Szegedy et al., 2015) pour quantifier la qualité des GANs. D'autres métriques reposent plus spécifiquement, sur des approximations en échantillon fini qui permettent l'utilisation de méthodes non paramétriques telles que les plus proches voisins (Devroye and Wise, 1980).

La mesure d'évaluation doit évaluer à la fois la qualité et diversité. Le second enjeu est directement lié avec la finalité des GANs : doivent-ils être capables de générer des images de qualité ou bien avoir la plus grande diversité possible ? Salimans et al. (2016) utilisent l'Inception Score (IS) et un réseau préalablement entraîné pour mesurer la qualité des images générées. Si l'IS évalue à la fois le réalisme et la diversité des points générés, il n'évalue en revanche, pas correctement la diversité au sein d'une même classe. Sajjadi et al. (2018) argumentent que pour quantifier proprement la qualité et la diversité des images générées, une seule mesure ne suffit pas. Par conséquent, ils définissent la métrique *Précision/Rappel*. Pour améliorer la robustesse de cette métrique, en particulier quand le générateur s'effondre, Kynkäänniemi et al. (2019) ont proposé la métrique *Precision/Rappel améliorée* (Improved PR) basée sur une estimation non paramétrique du support. La précision évalue la proportion de la loi μ_θ qui appartient au support de la distribution cible. Réciproquement, le rappel s'intéresse à la mesure de la distribution cible qui peut être reconstruite par le générateur. La figure 1.9 illustre synthétiquement ces deux notions.

La mesure d'évaluation doit-elle être une distance entre lois de probabilité ou entre variétés topologiques ? Il est clair que le choix d'une mesure d'évaluation est intimement lié à l'objectif des GANs. De ce point de vue là, l'objectif des GANs est-il d'approcher la distribution cible ou seulement son support ? Succinctement, on distingue les distances entre lois de probabilités (*mesures probabilistes*) de celles entre variétés topologiques (*mesures topologiques*):

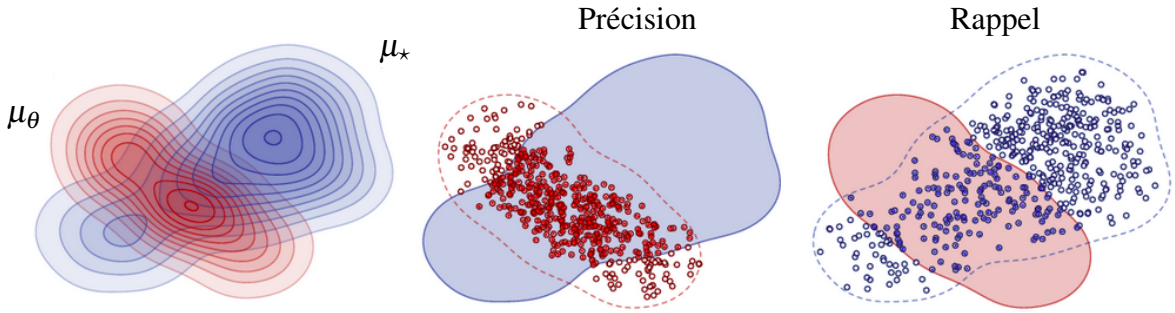


Fig. 1.9 A gauche, la distribution cible $\mu_\star$ et le modèle μ_θ. Au milieu, la mesure des points surlignés en rouge correspond à la précision du modèle. A droite, la mesure des points surlignés en bleu correspond à son rappel. Source : Kynkäänniemi et al. (2019).

- mesure probabiliste : Heusel et al. (2017) utilisent la distance de Fréchet. Ils estiment deux gaussiennes multivariées à partir des données d'entraînement et des données générées et comparent les moyennes et variances obtenues. Plus récemment, la distance de Wasserstein et son approximation, la Earth Mover's distance, basée sur des collections de points échantillonés par $\mu_\star$ et μ_θ a également été proposée.

- mesure topologique : la métrique *Précision/Rappel améliorée* proposée par Kynkäänniemi et al. (2019) est, quant à elle, basée sur une estimation non paramétrique du support. De même, la distance de Hausdhorff (Xiang and Li, 2017) mesure l'éloignement entre deux sous-ensembles d'un espace métrique. De manière similaire, Roth et al. (2017) mesurent pour chaque point présent dans le jeu de données, la distance à la variété crée par le générateur, c'est-à-dire que,

$$\forall x \in \text{supp}(\mu_\star), \inf_{y \in \text{supp}(\mu_\theta)} \|x - y\|,$$

  où $\|.\|$ correspond à la norme euclidienne et $\text{supp}(\mu)$ correspond au support d'une loi μ donnée. Enfin, Khrulkov and Oseledets (2018) définissent le score géométrique (geometry score), et comparent les similitudes entre deux variétés topologiques en utilisant des notions de topologie algébrique.

Evaluation de la généralisation d'un GAN. L'objectif des GANs est-il de générer des exemples qui représentent fidèlement le jeu de données ou, au contraire, doivent-ils être capables de générer des images qui n'ont jamais été observées pendant l'entraînement. Il est en effet extrêmement intéressant de se demander si les GANs apprennent la distribution cible ou mémorisent simplement le jeu d'entraînement observé. Arora and Zhang (2017) proposent d'utiliser le paradoxe des anniversaires pour évaluer le nombre d'images distinctes générer par

les GANs et répondre à la question. De manière plus générale, il n'existe encore à ce jour que très peu de travaux sur l'évaluation empirique des capacités de généralisation des GANs ? En effet, la communauté ne fait pas nécessairement la différence entre les données d'entraînement des données de test, signifiant que comprendre la généralisation des GANs n'est pas l'une des priorités de la communauté. Enfin, il faut noter que cette question a tout de même suscité quelques recherches théoriques (Zhang et al., 2018; Qi, 2019).

1.5 Organisation du manuscrit et présentation des contributions

Ce travail de thèse est structuré en cinq chapitres. Le Chapitre 2 vise à formaliser l'entraînement des GANs et s'intéresse principalement aux propriétés statistiques des GANs définis par Goodfellow et al. (2014). Ces travaux menés en collaboration avec Gérard Biau (LPSM), Benoit Cadre (IRMAR, Université Rennes 2) et Maxime Sangnier (LPSM) ont été publiés au journal Annals of Statistics. Le Chapitre 3 étend cette recherche aux Wasserstein GANs (Arjovsky et al., 2017), réputés plus stables. Mené conjointement avec Gérard Biau et Maxime Sangnier, ce travail a fait l'objet d'un article soumis pour publication. Le Chapitre 4 découle de l'utilisation de réseaux de neurones paramétrés avec la fonction d'activation GroupSort (Anil et al., 2019). Il se propose d'étudier l'expressivité de ces réseaux et sera proposé à une conférence. La suite de la thèse est axée autour d'un problème plus appliqué. Le Chapitre 5 traite en effet de la difficulté d'apprendre une variété non connexe avec les GANs. C'est le sujet de deux travaux de recherche menés conjointement avec Thibaut Issenhuth (Criteo) et Jérémie Mary (Criteo), dont l'un a été publié à ICML 2020 et le second est en cours de révision.

1.5.1 Chapitre 2 : Etude statistique des GANs

Ce chapitre propose une formalisation théorique des GANs et analyse certaines de leurs propriétés mathématiques et statistiques. Nous commençons par rappeler que pour un générateur G_θ et un discriminateur $D \in \mathscr{D}$, les GANs optimisent le critère probabiliste suivant :

$$L(\theta, D) = \int \ln(D) p_\star \mathrm{d}\mu + \int \ln(1-D) p_\theta \mathrm{d}\mu.$$

En particulier, les GANs cherchent à résoudre

$$\inf_{\theta \in \Theta} \sup_{D \in \mathscr{D}} L(\theta, D).$$

Nous commençons par étudier le cas où le discriminateur n'est pas restreint à un modèle paramétrique et où $\mathscr{D} = \mathscr{D}_\infty$ correspond à l'ensemble des fonctions mesurables de $\mathbb{R}^D$ dans $[0,1]$. Dans ce contexte non paramétrique, nous établissons le lien entre l'entraînement adverse des GANs et la divergence de Jensen-Shannon. Le Théorème 2.2.2 montre l'existence et l'unicité de l'optimum des GANs, c'est-à-dire que, sous certaines hypothèses,

$$\{\theta^\star\} = \arg\min_{\theta\in\Theta} \sup_{D\in\mathscr{D}_\infty} L(\theta, D) = \arg\min_{\theta\in\Theta} D_{\mathrm{JS}}(p_\star, p_\theta),$$

existe et est un singleton. Ici D_{JS} correspond à la divergence de Jensen-Shannon. Nous nous ramenons, par la suite, à un cas plus réaliste où la classe de fonctions discriminatives est paramétrée par un réseau de neurones. En utilisant la notation $L(\theta, D) = L(\theta, \alpha)$ dans le cas paramétrique, l'objectif des GANs consiste alors à trouver le modèle génératif suivant :

$$\bar{\Theta} = \arg\min_{\theta\in\Theta} \sup_{\alpha\in\Lambda} L(\theta,\alpha) = \arg\min_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \int \log(D_\alpha) p_\star \mathrm{d}\mu + \int \log(1 - D_\alpha) p_\theta \mathrm{d}\mu.$$

En particulier, le Théorème 2.3.1 montre, en supposant que le discriminateur optimal est approché à ε près, que pour chaque $\bar{\theta} \in \bar{\Theta}$, il existe une constante $c > 0$ (indépendante de ε) telle que :

$$0 \leq D_{\mathrm{JS}}(p_\star, p_{\bar{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq c\varepsilon^2.$$

En revanche, il est clair qu'en pratique nous n'avons uniquement accès qu'à un jeu de données de n échantillons $X_1, \ldots, X_n$ indépendants et identiquement distribués selon $p_\star$. Le critère empirique des GANs devient,

$$\hat{L}(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ln D(X_i) + \frac{1}{n}\sum_{i=1}^{n} \ln(1 - D \circ G_\theta(Z_i)),$$

où ln est le logarithme naturel et $Z_1, \ldots, Z_n$ sont des variables indépendantes et identiquement distribuées de loi Z. Par conséquent, l'ensemble des paramètres optimaux associés se définit comme suit :

$$\hat{\Theta} = \arg\min_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \hat{L}(\theta, \alpha).$$

L'un des principaux résultats du chapitre (Théorème 2.4.1) montre que, sous des hypothèses similaires à celles du Théorème 2.3.1, nous avons, pour tout $\hat{\theta} \in \hat{\Theta}$:

$$\mathbb{E} D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) = \mathrm{O}\Big(\varepsilon^2 + \frac{1}{\sqrt{n}}\Big).$$

1.5.2 Chapitre 3 : Extension et développement pour le cas des WGANs

Plusieurs études empiriques (Gulrajani et al., 2017; Roth et al., 2017) ont validé les bénéfices de l'approche cousine appelée Wasserstein GANs (WGANs) proposée par Arjovsky et al. (2017). Cette dernière apporte une stabilisation dans le processus d'entraînement. Il est donc important d'approfondir notre compréhension de cette architecture. De manière similaire, au chapitre précédent, pour bien comprendre le fonctionnement de ces WGANs, il est nécessaire de distinguer deux problèmes.

Tout d'abord, dans le cas où la classe de fonctions discriminatives $\mathscr{D}$ correspond à la classe non paramétrique des fonctions 1-Lipschitz, l'objectif des WGANs revient à minimiser la distance de Wasserstein entre la distribution cible $\mu_\star$ et le modèle $\mathscr{P}$. Plus formellement, l'objectif théorique des WGANs est le suivant :

$$\inf_{\theta\in\Theta} W(\mu_\star,\mu_\theta) = \inf_{\theta\in\Theta} \sup_{f\in \mathrm{Lip}_1} |\mathbb{E}_{\mu_\star} f - \mathbb{E}_{\mu_\theta} f|, \tag{1.5.1}$$

où W correspond à la distance de 1-Wasserstein et $\mathrm{Lip}_1 = \big\{f : E \to \mathbb{R} : |f(x)-f(y)| \leqslant \|x-y\|, (x,y) \in (\mathbb{R}^D)^2\big\}$.

Ensuite, le second problème, plus réaliste, vise à considérer une classe de fonctions discriminatives paramétrique plus restreinte, $\mathscr{D} = \{D_\alpha : \alpha \in \Lambda\}$. Dans cette approche, le véritable problème des WGANs s'écrit :

$$\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} |\mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha|. \tag{1.5.2}$$

En ré-écrivant les deux objectifs des WGANs théoriques (T-WGANs) et des WGANs sous forme d'Integral Probability Metric (Müller, 1997), nous obtenons :

$$\text{T-WGANs: } \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu^\star,\mu_\theta) \quad \text{et} \quad \text{WGANs: } \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu^\star,\mu_\theta),$$

où pour une classe de fonctions $\mathscr{F}$ donnée, l'IPM entre deux distributions μ et ν s'écrit $d_{\mathscr{F}}(\mu,\nu) = \sup_{f\in\mathscr{G}} |\mathbb{E}_\mu f - \mathbb{E}_\nu f|$.

Comme pour le chapitre précédent, nous nous intéressons à l'influence de l'échantillon et considérons le cas où nous n'avons accès qu'à un ensemble fini de points, représentés par la mesure empirique μ_n. Finalement, cela nous permet d'identifier les trois ensembles de paramètres correspondants :

$$\Theta^\star = \arg\min_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu^\star,\mu_\theta) \quad \& \quad \bar{\Theta} = \arg\min_{\theta\in\Theta} d_{\mathscr{D}}(\mu^\star,\mu_\theta) \quad \& \quad \hat{\Theta}_n = \arg\min_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta).$$

L'objectif du présent chapitre est de parvenir à étudier ces trois ensembles et de pouvoir comparer la performance des différents modèles génératifs obenus à partir de $\Theta^\star$, $\bar{\Theta}$ et $\hat{\Theta}_n$. Pour avoir une meilleure compréhension de la performance finale des WGANs $d_{\mathrm{Lip}_1}(\mu^\star, \mu_{\hat{\theta}_n})$ où $\hat{\theta}_n \in \hat{\Theta}_n$, nous proposons la décomposition suivante :

$$d_{\mathrm{Lip}_1}(\mu^\star, \mu_{\hat{\theta}_n}) \leqslant \varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}} + \varepsilon_{\mathrm{approx}}, \tag{1.5.3}$$

où

- $\varepsilon_{\mathrm{approx}} = \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)$ est liée à la capacité d'approximation du modèle génératif et la performance des paramètres $\theta^\star \in \Theta^\star$;
- $\varepsilon_{\mathrm{optim}} = \sup_{\bar{\theta} \in \bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)$ correspond à l'écart de performance entre un paramètre $\bar{\theta} \in \bar{\Theta}$ et $\theta^\star \in \Theta^\star$. L'analyse de cette erreur est menée dans la Section 3.3 et, en particulier, le Théorème 3.3.1 montre que, sous certaines hypothèses, elle peut être arbitrairement petite;
- $\varepsilon_{\mathrm{estim}}$ mesure l'écart de performance lié à l'obtention d'un paramètre $\hat{\theta}_n \in \hat{\Theta}_n$ plutôt que $\bar{\theta} \in \bar{\Theta}$. Notons que le Théorème 3.4.1 prouve que, sous certaines hypothèses, la somme $\varepsilon_{\mathrm{optim}} + \varepsilon_{\mathrm{estim}}$ peut être arbitrairement petite avec grande probabilité.

Des expériences sur données réelles et simulées viennent compléter les résultats théoriques.

1.5.3 Chapitre 4 : Etude des réseaux de neurones dits GroupSort et application aux GANs

Les récentes publications sur les attaques adverses liées aux réseaux profonds (Goodfellow et al., 2015) et le développement des WGANs ont préconisé l'utilisation de réseaux de neurones avec des constantes de Lipschitz restreintes. Motivés par ces observations, Anil et al. (2019) ont proposé l'utilisation de réseaux de neurones dits GroupSort avec des contraintes sur les poids. Les auteurs de cette publications ont notamment prouver que les réseaux GroupSort pouvaient approcher n'importe quelle fonction Lipschitz tout en garantissant le caractère Lipschitz de l'estimateur. Dans ce chapitre, nous visons à mieux comprendre l'intérêt des réseaux GroupSort, utilisés dont le chapitre précédent, et faisons un pas théorique vers une meilleure compréhension de leur expressivité.

Les réseaux GroupSort se caractérisent par leur fonction d'activation GroupSort qui sépare les entrées en groupes et les trie par ordre croissant. La fonction d'activation GroupSort avec une taille de regroupement (grouping size) $k \geqslant 2$ est appliquée sur un vecteur $x_1, \ldots, x_{kn}$. Tout

d'abord, elle sépare le vecteur en n groupes $G_1 = \{x_1, \dots, x_k\}, \dots G_n = \{x_{nk-k-1}, \dots, x_{nk}\}$. Puis, elle trie chaque groupe comme suit:

$$\sigma_k(x_1, \dots, x_k, \dots, x_{nk-(k-1)}, \dots, x_{nk}) = \left(x_{(k)}^{G_1}, \dots, x_{(1)}^{G_1}\right), \dots, \left(x_{(k)}^{G_n}, \dots, x_{(1)}^{G_n}\right),$$

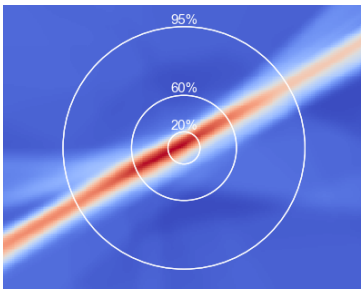
où $x_{(i)}^{G}$, la notation des statistiques d'ordre, correspond au $i$ème plus petit élément du groupe G. Tout comme les réseaux ReLU, les réseaux GroupSort paramètrent des fonctions linéaires par morceaux. L'étude d'expressivité de ces réseaux commencent par analyser leur faculté à représenter l'ensemble des fonctions linéaires continues par morceaux. Nous montrons, en particulier avec le Corollaire 4.3.1, que pour toute fonction Lipschitz f linéaires par morceaux sur m_f sous-domaines convexes $\Omega_1, \dots, \Omega_{m_f}$ ($m_f = k^n$ avec $n \geqslant 1$), il existe un réseau GroupSort avec une taille de regroupement k, une profondeur $2\lceil \log_k(m_f) \rceil + 1$ et une taille au plus $\frac{m_f^2 - 1}{k-1}$ qui reproduit la fonction f.

La faculté de ces réseaux à reproduire les fonctions linéaires par morceaux nous permet de passer au cas plus général de l'approximation des fonctions Lipschitz. Nous prouvons que pour tout $\varepsilon > 0$, et toute fonction f Lipschitz définie sur $[0,1]^d$, il existe un réseau de neurones GroupSort D avec une taille de groupement $\lceil \frac{2\sqrt{d}}{\varepsilon} \rceil$ tel que $\|f - D\|_\infty \leqslant \varepsilon$. De plus, la profondeur de D est $O(d^2)$. Pour conclure, nous illustrons l'efficacité des réseaux GroupSort par rapport à celles des réseaux ReLU sur un ensemble d'expériences synthétiques.

1.5.4 Chapitre 5 : L'apprentissage de variétés non connexes avec les GANs

Dans la formulation standard des GANs, une distribution latente unimodale (unifome ou gaussienne) est transformée par un générateur continu dans l'espace des images. Par conséquent, dans le cas où la distribution cible a un support non connexe, aucune des distributions modélisées μ_θ ne pourra parfaitement approcher $\mu^\star$. Dans ce chapitre, nous formalisons ce cadre précis et établissons des résultats qui mesurent la quantité de données simulées se trouvant en dehors de la variété cible.

Notre étude part du constat suivant établi dans un contexte simple : pour apprendre un mélange de deux gaussiennes, les GANs divisent l'espace latent en deux zones, comme le montre la ligne de séparation en rouge sur la figure 1.10a. Plus important encore, chaque bruit gaussien à l'intérieur de cette zone rouge sur la figure 1.10a est ensuite envoyé dans l'espace de sortie entre les deux modes (voir Figure 1.10b) de la loi cible. En utilisant des résultats connus de l'inégalité gaussienne isopérimétrique, nous quantifions la quantité de données en dehors de la variété cible. La métrique choisie pour définir si un échantillon



(a) Heatmap de la norme du jacobien du générateur. Les cercles blancs correspondent aux quantiles de la distribution latente $\mathcal{N}(0, I)$.

(b) En vert : la distribution cible. Les points colorés correspondent aux échantillons générés par le générateur. Ils sont colorés en fonctions de la norme du jacobien de G_θ. La même heatmap que dans la figure (a) est utilisée.

Fig. 1.10 L'apprentissage d'une variété non connexe avec un GAN standard amène à l'apparition d'une zone à forts gradients dans l'espace de départ où chaque échantillon est envoyé en dehors de la variété.

donné appartient à la variété cible est donc primordiale. Pour la présente étude, nous avons choisi la métrique Précision/Rappel (PR) proposée par Sajjadi et al. (2018) et, en particulier, la version améliorée (Improved PR) (Kynkäänniemi et al., 2019) construite sur une estimation non paramétrique des supports. Comme précisé plus haut, la précision quantifie la part de la fausse distribution qui peut être générée par la distribution cbile $\mu_\star$, tandis que le rappel mesure la part de la vraie distribution qui peut être reconstruite par la distribution μ_θ du modèle. Plus formellement, soient $(X_1, \dots, X_n) \sim \mu_\theta^n$ (ensemble de données générées par le générateur) et $(Y_1, \dots, Y_n) \sim \mu_\theta^n$ (ensemble de données échantillonées par la distribution cible). Pour chaque X (ou respectivement chaque Y), on considère $(X_{(1)}, \dots, X_{(n-1)})$, l'arrangement des éléments dans $(X_1, \dots X_n) \setminus X$ selon leur distance croissante à X ($X_{(1)} = \underset{X_i \in (X_1, \dots X_n) \setminus X}{\arg\min} \|X_i - X\|$). Pour chaque $k \in \mathbb{N}$ et chaque X, la précision $\alpha_k^n(X)$ du point X est définie par

$$\text{Précision: } \alpha_k^n(X) = 1 \iff \exists Y \in (Y_1, \dots, Y_n), \|X - Y\| \leqslant \|Y_{(k)} - Y\|.$$

De manière similaire, le rappel $\beta_k^n(Y)$ d'un point $Y \in (Y_1, \dots, Y_n)$ est défini par

$$\text{Rappel: } \beta_k^n(Y) = 1 \iff \exists X \in (X_1, \dots, X_n), \|Y - X\| \leqslant \|X_{(k)} - X\|.$$

Après cette analyse théorique, nous poursuivons notre étude en définissant une méthode d'échantillonnage de rejet basée sur la norme du Jacobien du générateur. Nous montrons sa capacité à enlever les points de données de mauvaise qualité et ceux, à la fois sur des jeux

de données synthétiques (approximation de mélanges de gaussiennes), mais aussi sur de la génération d'images en grande dimension.

Chapter 2

Some theoretical properties of GANs

Abstract

Generative Adversarial Networks (GANs) are a class of generative algorithms that have been shown to produce state-of-the-art samples, especially in the domain of image creation. The fundamental principle of GANs is to approximate the unknown distribution of a given data set by optimizing an objective function through an adversarial game between a family of generators and a family of discriminators. In this paper, we offer a better theoretical understanding of GANs by analyzing some of their mathematical and statistical properties. We study the deep connection between the adversarial principle underlying GANs and the Jensen-Shannon divergence, together with some optimality characteristics of the problem. An analysis of the role of the discriminator family via approximation arguments is also provided. In addition, taking a statistical point of view, we study the large sample properties of the estimated distribution and prove in particular a central limit theorem. Some of our results are illustrated with simulated examples.

Contents

2.1 Introduction **26**

2.2 Optimality properties **29**

2.3 Approximation properties **34**

2.4 Statistical analysis **35**

2.5 Conclusion and perspectives **53**

Appendix 2.A Technical results **55**

2.1 Introduction

The fields of machine learning and artificial intelligence have seen spectacular advances in recent years, one of the most promising being perhaps the success of Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014). GANs are a class of generative algorithms implemented by a system of two neural networks contesting with each other in a zero-sum game framework. This technique is now recognized as being capable of generating photographs that look authentic to human observers (e.g., Salimans et al., 2016), and its spectrum of applications is growing at a fast pace, with impressive results in the domains of inpainting, speech, and 3D modeling, to name but a few. A survey of the most recent advances is given by Goodfellow (2016).

The objective of GANs is to generate fake observations of a target distribution $p_\star$ from which only a true sample (e.g., real-life images represented using raw pixels) is available. It should be pointed out at the outset that the data involved in the domain are usually so complex that no exhaustive description of $p_\star$ by a classical parametric model is appropriate, nor its estimation by a traditional maximum likelihood approach. Similarly, the dimension of the samples is often very large, and this effectively excludes a strategy based on nonparametric density estimation techniques such as kernel or nearest neighbor smoothing, for example. In order to generate according to $p_\star$, GANs proceed by an adversarial scheme involving two components: a family of generators and a family of discriminators, which are both implemented by neural networks. The generators admit low-dimensional random observations with a known distribution (typically Gaussian or uniform) as input, and attempt to transform them into fake data that can match the distribution $p_\star$; on the other hand, the discriminators aim to accurately discriminate between the true observations from $p_\star$ and those produced by the generators. The generators and the discriminators are calibrated by optimizing an objective function in such a way that the distribution of the generated sample is as indistinguishable as possible from that of the original data. In pictorial terms, this process is often compared to a game of cops and robbers, in which a team of counterfeiters illegally produces banknotes and tries to make them undetectable in the eyes of a team of police officers, whose objective is of course the opposite. The competition pushes both teams to improve their methods until counterfeit money becomes indistinguishable (or not) from genuine currency.

From a mathematical point of view, here is how the generative process of GANs can be represented. All the densities that we consider in the article are supposed to be dominated by a fixed, known, measure μ on E, where E is a Borel subset of $\mathbb{R}^d$. Depending on the practical

context, this dominating measure may be the Lebesgue measure, the counting measure, or more generally the Hausdorff measure on some submanifold of $\mathbb{R}^d$. We assume to have at hand an i.i.d. sample $X_1,\dots,X_n$, drawn according to some unknown density $p_\star$ on E. These random variables model the available data, such as images or video sequences; they typically take their values in a high-dimensional space, so that the ambient dimension d must be thought of as large. The generators as a whole have the form of a parametric family of functions from $\mathbb{R}^{d'}$ to E (usually, $d' \ll d$), say $\mathscr{G} = \{G_\theta\}_{\theta\in\Theta}$, $\Theta \subset \mathbb{R}^p$. Each function G_θ is intended to be applied to a d'-dimensional random variable Z (sometimes called the noise—in most cases Gaussian or uniform), so that there is a natural family of densities associated with the generators, say $\mathscr{P} = \{p_\theta\}_{\theta\in\Theta}$, where, by definition, $G_\theta(Z) \overset{\mathscr{L}}{=} p_\theta \mathrm{d}\mu$. In this model, each density p_θ is a potential candidate to represent $p_\star$. On the other hand, the discriminators are described by a family of Borel functions from E to $[0,1]$, say $\mathscr{D}$, where each $D \in \mathscr{D}$ must be thought of as the probability that an observation comes from $p_\star$ (the higher $D(x)$, the higher the probability that x is drawn from $p_\star$). At some point, but not always, we will assume that $\mathscr{D}$ is in fact a parametric class, of the form $\{D_\alpha\}_{\alpha\in\Lambda}$, $\Lambda \subset \mathbb{R}^q$, as is always the case in practice. In GANs algorithms, both parametric models $\{G_\theta\}_{\theta\in\Theta}$ and $\{D_\alpha\}_{\alpha\in\Lambda}$ take the form of neural networks, but this does not play a fundamental role in this paper. We will simply remember that the dimensions p and q are potentially very large, which takes us away from a classical parametric setting. We also insist on the fact that it is <u>not</u> assumed that $p_\star$ belongs to $\mathscr{P}$.

Let $Z_1,\dots,Z_n$ be an i.i.d. sample of random variables, all distributed as the noise Z. The objective is to solve in θ the problem

$$\inf_{\theta\in\Theta} \sup_{D\in\mathscr{D}} \Big[\prod_{i=1}^n D(X_i) \times \prod_{i=1}^n (1 - D\circ G_\theta(Z_i)) \Big], \tag{2.1.1}$$

or, equivalently, to find $\hat\theta \in \Theta$ such that

$$\sup_{D\in\mathscr{D}} \hat{L}(\hat\theta, D) \le \sup_{D\in\mathscr{D}} \hat{L}(\theta, D), \quad \forall \theta \in \Theta, \tag{2.1.2}$$

where

$$\hat{L}(\theta, D) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \ln D(X_i) + \frac{1}{n}\sum_{i=1}^n \ln(1 - D\circ G_\theta(Z_i))$$

(ln is the natural logarithm). The zero-sum game (2.1.1) is the statistical translation of making the distribution of $G_\theta(Z_i)$ (i.e., p_θ) as indistinguishable as possible from that of X_i (i.e., $p_\star$). Here, distinguishability is understood as the capability to determine from which distribution an observation x comes from. Mathematically, this is captured by the discrimination value $D(x)$, which represents the probability that x comes from $p_\star$ rather than from p_θ. Therefore,

for a given θ, the discriminator D is determined so as to be maximal on the X_i and minimal on the $G_\theta(Z_i)$. In the most favorable situation (that is, when the two samples are scattered by $\mathscr{D}$, $\sup_{D\in\mathscr{D}} \hat{L}(\theta, D)$ is zero, and the larger this quantity, the more distinguishable the two samples are. Hence, in order to make the distribution p_θ as indistinguishable as possible from $p_\star$, G_θ has to be driven so as to minimize $\sup_{D\in\mathscr{D}} \hat{L}(\theta, D)$.

This adversarial problem is often illustrated by the struggle between a police team (the discriminators), trying to distinguish true banknotes from false ones (respectively, the X_i and the $G_\theta(Z_i)$), and a counterfeiters team, slaving to produce banknotes as credible as possible and to mislead the police. Obviously, their objectives (represented by the quantity $\hat{L}(\theta, D)$) are exactly opposite. All in all, we see that the criterion seeks to find the right balance between the conflicting interests of the generators and the discriminators. The hope is that the $\hat{\theta}$ achieving equilibrium will make it possible to generate observations $G_{\hat{\theta}}(Z_1), \ldots, G_{\hat{\theta}}(Z_n)$ indistinguishable from reality, i.e., observations with a distribution close to the unknown $p_\star$.

The criterion $\hat{L}(\theta, D)$ involved in (2.1.2) is the criterion originally proposed in the adversarial framework of Goodfellow et al. (2014). Since then, the success of GANs in applications has led to a large volume of literature on variants, which all have many desirable properties but are based on different optimization criteria—examples are MMD-GANs (Li et al., 2017), f-GANs (Nowozin et al., 2016), Wasserstein-GANs (Arjovsky et al., 2017), and an approach based on scattering transforms (Angles and Mallat, 2018). All these variations and their innumerable algorithmic versions constitute the galaxy of GANs. That being said, despite increasingly spectacular applications, little is known about the mathematical and statistical forces behind these algorithms (e.g., Arjovsky et al., 2017; Liu et al., 2017; Zhang et al., 2018), and, in fact, nearly nothing about the primary adversarial problem (2.1.2). As acknowledged by Liu et al. (2017), basic questions on how well GANs can approximate the target distribution $p_\star$ remain largely unanswered. In particular, the role and impact of the discriminators on the quality of the approximation are still a mystery, and simple but fundamental questions regarding statistical consistency and rates of convergence remain open.

In the present article, we propose to take a small step towards a better theoretical understanding of GANs by analyzing some of the mathematical and statistical properties of the original adversarial problem (2.1.2). In Section 2.2, we study the deep connection between the population version of (2.1.2) and the Jensen-Shannon divergence, together with some optimality characteristics of the problem, often referred to in the literature but in fact poorly understood. Section 2.3 is devoted to a better comprehension of the role of the discriminator family via approximation arguments. Finally, taking a statistical point of view, we study in Section 2.4 the large sample properties of the distribution $p_{\hat{\theta}}$ and of $\hat{\theta}$, and prove in particular a central limit theorem for this parameter. Section 2.5 summarizes the main results and discusses research

directions for future work. For clarity, most technical proofs are gathered in Section 2.A. Some of our results are illustrated with simulated examples.

2.2 Optimality properties

We start by studying some important properties of the adversarial principle, emphasizing the role played by the Jensen-Shannon divergence. We recall that if P and Q are probability measures on E, and P is absolutely continuous with respect to Q, then the Kullback-Leibler divergence from Q to P is defined as $D_{\mathrm{KL}}(P \parallel Q) = \int \ln \frac{\mathrm{d}P}{\mathrm{d}Q} \mathrm{d}P$, where $\frac{\mathrm{d}P}{\mathrm{d}Q}$ is the Radon-Nikodym derivative of P with respect to Q. The Kullback-Leibler divergence is always nonnegative, with $D_{\mathrm{KL}}(P \parallel Q)$ zero if and only if $P = Q$. If $p = \frac{\mathrm{d}P}{\mathrm{d}\mu}$ and $q = \frac{\mathrm{d}Q}{\mathrm{d}\mu}$ exist (meaning that P and Q are absolutely continuous with respect to μ, with densities p and q), then the Kullback-Leibler divergence is given as

$$D_{\mathrm{KL}}(P \parallel Q) = \int p \ln \frac{p}{q} \mathrm{d}\mu,$$

and alternatively denoted by $D_{\mathrm{KL}}(p \parallel q)$. We also recall that the Jensen-Shannon divergence is a symmetrized version of the Kullback-Leibler divergence. It is defined for any probability measures P and Q on E by

$$D_{\mathrm{JS}}(P,Q) = \frac{1}{2} D_{\mathrm{KL}}\Big(P \,\Big\|\, \frac{P+Q}{2}\Big) + \frac{1}{2} D_{\mathrm{KL}}\Big(Q \,\Big\|\, \frac{P+Q}{2}\Big),$$

and satisfies $0 \leq D_{\mathrm{JS}}(P,Q) \leq \ln 2$. The square root of the Jensen-Shannon divergence is a metric often referred to as Jensen-Shannon distance (Endres and Schindelin, 2003). When P and Q have densities p and q with respect to μ, we use the notation $D_{\mathrm{JS}}(p,q)$ in place of $D_{\mathrm{JS}}(P,Q)$.

For a generator G_θ and an arbitrary discriminator $D \in \mathscr{D}$, the criterion $\hat{L}(\theta, D)$ to be optimized in (2.1.2) is but the empirical version of the probabilistic criterion

$$L(\theta, D) \stackrel{\text{def}}{=} \int \ln(D) p_\star \mathrm{d}\mu + \int \ln(1-D) p_\theta \mathrm{d}\mu.$$

We assume for the moment that the discriminator class $\mathscr{D}$ is not restricted and equals $\mathscr{D}_\infty$, the set of all Borel functions from E to $[0,1]$. We note however that, for all $\theta \in \Theta$,

$$0 \geq \sup_{D \in \mathscr{D}_\infty} L(\theta, D) \geq -\ln 2 \Big(\int p_\star \mathrm{d}\mu + \int p_\theta \mathrm{d}\mu \Big) = -\ln 4,$$

so that $\inf_{\theta\in\Theta}\sup_{D\in\mathscr{D}_\infty} L(\theta,D)\in[-\ln 4,0]$. Thus,

$$\inf_{\theta\in\Theta}\sup_{D\in\mathscr{D}_\infty} L(\theta,D)=\inf_{\theta\in\Theta}\sup_{D\in\mathscr{D}_\infty:L(\theta,D)>-\infty} L(\theta,D).$$

This identity points out the importance of discriminators such that $L(\theta,D)>-\infty$, which we call θ-admissible. In the sequel, in order to avoid unnecessary problems of integrability, we only consider such discriminators, keeping in mind that the others have no interest.

Of course, working with $\mathscr{D}_\infty$ is somehow an idealized vision, since in practice the discriminators are always parameterized by some parameter $\alpha\in\Lambda$, $\Lambda\subset\mathbb{R}^q$. Nevertheless, this point of view is informative and, in fact, is at the core of the connection between our generative problem and the Jensen-Shannon divergence. Indeed, taking the supremum of $L(\theta,D)$ over $\mathscr{D}_\infty$, we have

$$\begin{aligned}\sup_{D\in\mathscr{D}_\infty} L(\theta,D)&=\sup_{D\in\mathscr{D}_\infty}\int\big[\ln(D)p_\star+\ln(1-D)p_\theta\big]\mathrm{d}\mu\\&\leq\int\sup_{D\in\mathscr{D}_\infty}\big[\ln(D)p_\star+\ln(1-D)p_\theta\big]\mathrm{d}\mu\\&=L(\theta,D^\star_\theta),\end{aligned}$$

where

$$D^\star_\theta\stackrel{\text{def}}{=}\frac{p_\star}{p_\star+p_\theta}. \tag{2.2.1}$$

(We use throughout the convention $0/0=0$ and $\infty\times 0=0$.) By observing that $L(\theta,D^\star_\theta)=2D_{\mathrm{JS}}(p_\star,p_\theta)-\ln 4$, we conclude that, for all $\theta\in\Theta$,

$$\sup_{D\in\mathscr{D}_\infty} L(\theta,D)=L(\theta,D^\star_\theta)=2D_{\mathrm{JS}}(p_\star,p_\theta)-\ln 4.$$

In particular, $D^\star_\theta$ is θ-admissible. The fact that $D^\star_\theta$ realizes the supremum of $L(\theta,D)$ over $\mathscr{D}_\infty$ and that this supremum is connected to the Jensen-Shannon divergence between $p_\star$ and p_θ appears in the original article by Goodfellow et al. (2014). This remark has given rise to many developments that interpret the adversarial problem (2.1.2) as the empirical version of the minimization problem $\inf_\theta D_{\mathrm{JS}}(p_\star,p_\theta)$ over Θ. Accordingly, many GANs algorithms try to learn the optimal function $D^\star_\theta$, using for example stochastic gradient descent techniques and mini-batch approaches. However, it remains to prove that $D^\star_\theta$ is unique as a maximizer of $L(\theta,D)$ over all D. The following theorem, which completes a result of (Goodfellow et al., 2014), shows that this is the case in some situations.

Theorem 2.2.1. *Let $\theta \in \Theta$ and $D \in \mathscr{D}_\infty$ be such that $L(\theta, D) = L(\theta, D^\star_\theta)$. Then $D = D^\star_\theta$ on the complementary of the set $\{p_\star = p_\theta = 0\}$. In particular, if $\mu(\{p_\star = p_\theta = 0\}) = 0$, then the function $D^\star_\theta$ is the unique discriminator that achieves the supremum of the functional $D \mapsto L(\theta, D)$ over $\mathscr{D}_\infty$, i.e.,*

$$\{D^\star_\theta\} = \arg\max_{D \in \mathscr{D}_\infty} L(\theta, D).$$

Before proving the theorem, it is important to note that if we dot not assume that $\mu(\{p_\star = p_\theta = 0\}) = 0$, then we cannot conclude that $D = D^\star_\theta$ μ-almost everywhere. To see this, suppose that $p_\theta = p_\star$. Then, whatever $\bar{D} \in \mathscr{D}_\infty$ is, the discriminator $D^\star_\theta \mathbf{1}_{\{p_\theta > 0\}} + \bar{D}\mathbf{1}_{\{p_\theta = 0\}}$ satisfies

$$L(\theta, D^\star_\theta \mathbf{1}_{\{p_\theta > 0\}} + \bar{D}\mathbf{1}_{\{p_\theta = 0\}}) = L(\theta, D^\star_\theta).$$

This simple counterexample shows that uniqueness of the optimal discriminator does not hold in general.

Proof. Let $D \in \mathscr{D}_\infty$ be a discriminator such that $L(\theta, D) = L(\theta, D^\star_\theta)$. In particular, $L(\theta, D) > -\infty$ and D is θ-admissible. Thus, letting $A \stackrel{\text{def}}{=} \{p_\star = p_\theta = 0\}$ and $f_\alpha \stackrel{\text{def}}{=} p_\star \ln(\alpha) + p_\theta \ln(1-\alpha)$ for $\alpha \in [0,1]$, we see that

$$\int_{A^c} (f_D - f_{D^\star_\theta}) \mathrm{d}\mu = 0.$$

Since, on A^c,

$$f_D \leq \sup_{\alpha \in [0,1]} f_\alpha = f_{D^\star_\theta},$$

we have $f_D = f_{D^\star_\theta}$ μ-almost everywhere on A^c. By uniqueness of the maximizer of $\alpha \mapsto f_\alpha$ on A^c, we conclude that $D = D^\star_\theta$ μ-almost everywhere on A^c. □

By definition of the optimal discriminator $D^\star_\theta$, we have

$$L(\theta, D^\star_\theta) = \sup_{D \in \mathscr{D}_\infty} L(\theta, D) = 2D_{\mathrm{JS}}(p_\star, p_\theta) - \ln 4, \quad \forall \theta \in \Theta.$$

Therefore, it makes sense to let the parameter $\theta^\star \in \Theta$ be defined as

$$L(\theta^\star, D^\star_{\theta^\star}) \leq L(\theta, D^\star_\theta), \quad \forall \theta \in \Theta,$$

or, equivalently,

$$D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq D_{\mathrm{JS}}(p_\star, p_\theta), \quad \forall \theta \in \Theta. \tag{2.2.2}$$

The parameter $\theta^\star$ may be interpreted as the best parameter in Θ for approaching the unknown density $p_\star$ in terms of Jensen-Shannon divergence, in a context where all possible discriminators

are available. In other words, the generator $G_{\theta^\star}$ is the ideal generator, and the density $p_{\theta^\star}$ is the one we would ideally like to use to generate fake samples. Of course, whenever $p_\star \in \mathscr{P}$ (i.e., the target density is in the model), then $p_\star = p_{\theta^\star}$, $D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) = 0$, and $D^\star_{\theta^\star} = 1/2$. This is, however, a very special case, which is of no interest, since in the applications covered by GANs, the data are usually so complex that the hypothesis $p_\star \in \mathscr{P}$ does not hold.

In the general case, our next theorem provides sufficient conditions for the existence and uniqueness of $\theta^\star$. For P and Q probability measures on E, we let $\delta(P,Q) = \sqrt{D_{\mathrm{JS}}(P,Q)}$, and recall that δ is a distance on the set of probability measures on E (Endres and Schindelin, 2003). We let $\mathrm{d}p_\star = p_\star \mathrm{d}\mu$ and, for all $\theta \in \Theta$, $\mathrm{d}P_\theta = p_\theta \mathrm{d}\mu$.

Theorem 2.2.2. *Assume that the model $\mathscr{P} = \{P_\theta\}_{\theta\in\Theta}$ is convex and compact for the metric δ. If $p_\star > 0$ μ-almost everywhere, then there exists a unique $\bar{p} \in \mathscr{P}$ such that*

$$\{\bar{p}\} = \arg\min_{p\in\mathscr{P}} D_{\mathrm{JS}}(p_\star, p).$$

In particular, if the model $\mathscr{P}$ is identifiable, then

$$\{\theta^\star\} = \arg\min_{\theta\in\Theta} L(\theta, D^\star_\theta)$$

or, equivalently,

$$\{\theta^\star\} = \arg\min_{\theta\in\Theta} D_{\mathrm{JS}}(p_\star, p_\theta).$$

We note that the identifiability assumption in the second statement of the theorem is hardly satisfied in the high-dimensional context of (deep) neural networks. In this case, it is likely that several parameters θ yield the same function (generator), so that the parametric setting is potentially misspecified. However, if we think in terms of distributions instead of parameters, then the first part of Theorem 2.2.2 ensures existence and uniqueness of the optimum.

Proof. Assuming the first part of the theorem, the second one is obvious since $L(\theta, D^\star_\theta) = \sup_{D\in\mathscr{D}_\infty} L(\theta, D) = 2D_{\mathrm{JS}}(p_\star, p_\theta) - \ln 4$. Therefore, it is enough to prove that there exists a unique density $\bar{p}$ of $\mathscr{P}$ such that

$$\{\bar{p}\} = \arg\min_{p\in\mathscr{P}} D_{\mathrm{JS}}(p_\star, p).$$

Existence. Since $\mathscr{P}$ is compact for δ, it is enough to show that the function

$$\begin{array}{rcl} \mathscr{P} & \to & \mathbb{R}_+ \\ P & \mapsto & D_{\mathrm{JS}}(p_\star, P) \end{array}$$

is continuous. But this is clear since, for all $P_1, P_2 \in \mathscr{P}$, $|\delta(p_\star, P_1) - \delta(p_\star, P_2)| \leq \delta(P_1, P_2)$ by the triangle inequality. Therefore, $\arg\min_{p\in\mathscr{P}} D_{\mathrm{JS}}(p_\star, p) \neq \emptyset$.

Uniqueness. For $a \geq 0$, we consider the function F_a defined by

$$F_a(x) = a \ln\Big(\frac{2a}{a+x}\Big) + x \ln\Big(\frac{2x}{a+x}\Big), \quad x \geq 0,$$

with the convention $0 \ln 0 = 0$. Clearly, $F_a''(x) = \frac{a}{x(a+x)}$, which shows that F_a is strictly convex whenever $a > 0$. We now proceed to prove that $L^1(\mu) \supset \mathscr{P} \ni p \mapsto D_{\mathrm{JS}}(p_\star, p)$ is strictly convex as well. Let $\lambda \in (0,1)$ and $p_1, p_2 \in \mathscr{P}$ with $p_1 \neq p_2$, i.e., $\mu(\{p_1 \neq p_2\}) > 0$. Then

$$\begin{aligned} D_{\mathrm{JS}}(p_\star, \lambda p_1 + (1-\lambda)p_2) &= \int F_{p_\star}(\lambda p_1 + (1-\lambda)p_2) \mathrm{d}\mu \\ &= \int_{\{p_1 = p_2\}} F_{p_\star}(p_1) \mathrm{d}\mu + \int_{\{p_1 \neq p_2\}} F_{p_\star}(\lambda p_1 + (1-\lambda)p_2) \mathrm{d}\mu. \end{aligned}$$

By the strict convexity of $F_{p_\star}$ over $\{p_\star > 0\}$, we obtain

$$\begin{aligned} &D_{\mathrm{JS}}(p_\star, \lambda p_1 + (1-\lambda)p_2) \\ &\quad < \int_{\{p_1 = p_2\}} F_{p_\star}(p_1) \mathrm{d}\mu + \lambda \int_{\{p_1 \neq p_2\}} F_{p_\star}(p_1) \mathrm{d}\mu + (1-\lambda) \int_{\{p_1 \neq p_2\}} F_{p_\star}(p_2) \mathrm{d}\mu, \end{aligned}$$

which implies

$$D_{\mathrm{JS}}(p_\star, \lambda p_1 + (1-\lambda)p_2) < \lambda D_{\mathrm{JS}}(p_\star, p_1) + (1-\lambda) D_{\mathrm{JS}}(p_\star, p_2).$$

Consequently, the function $L^1(\mu) \supset \mathscr{P} \ni p \mapsto D_{\mathrm{JS}}(p_\star, p)$ is strictly convex, and its $\arg\min$ over the convex set $\mathscr{P}$ is either the empty set or a singleton. □

Remark 2.2.1. *There are simple conditions for the model $\mathscr{P} = \{P_\theta\}_{\theta\in\Theta}$ to be compact for the metric δ. It is for example enough to suppose that Θ is compact, $\mathscr{P}$ is convex, and*

(*i*) *For all $x \in E$, the function $\theta \mapsto p_\theta(x)$ is continuous on Θ;*

(*ii*) *One has $\sup_{(\theta,\theta')\in\Theta^2} |p_\theta \ln p_{\theta'}| \in L^1(\mu)$.*

Let us quickly check that under these conditions, $\mathscr{P}$ is compact for the metric δ. Since Θ is compact, by the sequential characterization of compact sets, it is enough to prove that if $\Theta \supset (\theta_n)_n$ converges to $\theta \in \Theta$, then $D_{\mathrm{JS}}(p_\theta, p_{\theta_n}) \to 0$. But,

$$D_{\mathrm{JS}}(p_\theta, p_{\theta_n}) = \int \Big[p_\theta \ln\Big(\frac{2p_\theta}{p_\theta + p_{\theta_n}}\Big) + p_{\theta_n} \ln\Big(\frac{2p_{\theta_n}}{p_\theta + p_{\theta_n}}\Big) \Big] \mathrm{d}\mu.$$

By the convexity of $\mathscr{P}$, using (i) and (ii), the Lebesgue dominated convergence theorem shows that $D_{\mathrm{JS}}(p_\theta, p_{\theta_n}) \to 0$, whence the result.

Interpreting the adversarial problem in connection with the optimization program $\inf_{\theta\in\Theta} D_{\mathrm{JS}}(p_\star, p_\theta)$ is a bit misleading, because this is based on the assumption that all possible discriminators are available (and in particular the optimal discriminator $D^\star_\theta$). In the end this means assuming that we know the distribution $p_\star$, which is eventually not acceptable from a statistical perspective. In practice, the class of discriminators is always restricted to be a parametric family $\mathscr{D} = \{D_\alpha\}_{\alpha\in\Lambda}$, $\Lambda \subset \mathbb{R}^q$, and it is with this class that we have to work. From our point of view, problem (2.1.2) is a likelihood-type problem involving two parametric families $\mathscr{G}$ and $\mathscr{D}$, which must be analyzed as such, just as we would do for a classical maximum likelihood approach. In fact, it takes no more than a moment's thought to realize that the key lies in the approximation capabilities of the discriminator class $\mathscr{D}$ with respect to the functions $D^\star_\theta$, $\theta \in \Theta$. This is the issue that we discuss in the next section.

2.3 Approximation properties

In the remainder of the article, we assume that $\theta^\star$ exists, keeping in mind that Theorem 2.2.2 provides us with precise conditions guaranteeing its existence and its uniqueness. As pointed out earlier, in practice only a parametric class $\mathscr{D} = \{D_\alpha\}_{\alpha\in\Lambda}$, $\Lambda \subset \mathbb{R}^q$, is available, and it is therefore logical to consider the parameter $\bar\theta \in \Theta$ defined by

$$\sup_{D\in\mathscr{D}} L(\bar\theta, D) \le \sup_{D\in\mathscr{D}} L(\theta, D), \quad \forall \theta \in \Theta.$$

(We assume for now that $\bar\theta$ exists—sufficient conditions for this existence, relating to compactness of Θ and regularity of the model $\mathscr{P}$, will be given in the next section.) The density $p_{\bar\theta}$ is thus the best candidate to imitate $p_{\theta^\star}$, given the parametric families of generators $\mathscr{G}$ and discriminators $\mathscr{D}$. The natural question is then: is it possible to quantify the proximity between $p_{\bar\theta}$ and the ideal $p_{\theta^\star}$ via the approximation properties of the class $\mathscr{D}$? In other words, if $\mathscr{D}$ is growing, is it true that $p_{\bar\theta}$ approaches $p_{\theta^\star}$, and in the affirmative, in which sense and at which speed? Theorem 2.3.1 below provides a first answer to this important question, in terms of excess of Jensen-Shannon error $D_{\mathrm{JS}}(p_\star, p_{\bar\theta}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star})$. To state the result, we will need an assumption.

Let $\|\cdot\|_2$ be the $L^2(\mu)$ norm. Our condition guarantees that the parametric class $\mathscr{D}$ is rich enough to approach the discriminator $D^\star_{\bar\theta}$ in the L^2 sense. In the remainder of the section, it is assumed that $D^\star_{\bar\theta} \in L^2(\mu)$.

Assumption (H_ε) There exist $\varepsilon > 0$, $m \in (0, 1/2)$, and $D \in \mathscr{D} \cap L^2(\mu)$ such that $m \leq D \leq 1-m$ and $\|D - D^\star_{\bar{\theta}}\|_2 \leq \varepsilon$.

We observe in passing that such a discriminator D is $\bar{\theta}$-admissible. We are now equipped to state our approximation theorem. For ease of reading, its proof is postponed to Section 2.A.

Theorem 2.3.1. *Assume that, for some $M > 0$, $p_\star \leq M$ and $p_{\bar{\theta}} \leq M$. Then, under Assumption (H_ε) with $\varepsilon < 1/(2M)$, there exists a positive constant c (depending only upon m and M) such that*

$$0 \leq D_{\mathrm{JS}}(p_\star, p_{\bar{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq c\varepsilon^2. \tag{2.3.1}$$

This theorem points out that if the class $\mathscr{D}$ is rich enough to approximate the discriminator $D^\star_{\bar{\theta}}$ in such a way that $\|D - D^\star_{\bar{\theta}}\|_2 \leq \varepsilon$ for some small ε, then working with a restricted class of discriminators $\mathscr{D}$ instead of the set of all discriminators $\mathscr{D}_\infty$ has an impact that is not larger than a $\mathrm{O}(\varepsilon^2)$ factor with respect to the excess of Jensen-Shannon error. It shows in particular that the Jensen-Shannon divergence is a suitable criterion for the problem we are examining.

2.4 Statistical analysis

The data-dependent parameter $\hat{\theta}$, achieves the infimum of the adversarial problem (2.1.2). Practically speaking, it is this parameter that will be used in the end for producing fake data, via the associated generator $G_{\hat{\theta}}$. We first study in Subsection 2.4.1 the large sample properties of the distribution $p_{\hat{\theta}}$ via the excess of Jensen-Shannon error $D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star})$, and then state in Subsection 2.4.2 the almost sure convergence and asymptotic normality of the parameter $\hat{\theta}$ as the sample size n tends to infinity. Throughout, the parameter sets Θ and Λ are assumed to be compact subsets of $\mathbb{R}^p$ and $\mathbb{R}^q$, respectively. To simplify the analysis, we also assume that $\mu(E) < \infty$. In this case, every discriminator is in $L^p(\mu)$ for all $p \geq 1$.

2.4.1 Asymptotic properties of GANs

As for now, we assume that we have at hand a parametric family of generators $\mathscr{G} = \{G_\theta\}_{\theta \in \Theta}$, $\Theta \subset \mathbb{R}^p$, and a parametric family of discriminators $\mathscr{D} = \{D_\alpha\}_{\alpha \in \Lambda}$, $\Lambda \subset \mathbb{R}^q$. We recall that the collection of probability densities associated with $\mathscr{G}$ is $\mathscr{P} = \{p_\theta\}_{\theta \in \Theta}$, where $G_\theta(Z) \overset{\mathscr{L}}{=} p_\theta \mathrm{d}\mu$ and Z is some low-dimensional noise random variable. In order to avoid any confusion, for a given discriminator $D = D_\alpha$ we use the notation $\hat{L}(\theta, \alpha)$ (respectively, $L(\theta, \alpha)$) instead of $\hat{L}(\theta, D)$ (respectively, $L(\theta, D)$) when useful. So,

$$\hat{L}(\theta, \alpha) = \frac{1}{n}\sum_{i=1}^{n} \ln D_\alpha(X_i) + \frac{1}{n}\sum_{i=1}^{n} \ln(1 - D_\alpha \circ G_\theta(Z_i)),$$

and

$$L(\theta,\alpha)=\int \ln(D_\alpha)p_\star \mathrm{d}\mu+\int \ln(1-D_\alpha)p_\theta \mathrm{d}\mu.$$

We will need the following regularity assumptions:

Assumptions (H_{reg})

(H_D) There exists $\kappa\in(0,1/2)$ such that, for all $\alpha\in\Lambda$, $\kappa\le D_\alpha\le 1-\kappa$. In addition, the function $(x,\alpha)\mapsto D_\alpha(x)$ is of class C^1, with a uniformly bounded differential.

(H_G) For all $z\in\mathbb{R}^{d'}$, the function $\theta\mapsto G_\theta(z)$ is of class C^1, uniformly bounded, with a uniformly bounded differential.

(H_p) For all $x\in E$, the function $\theta\mapsto p_\theta(x)$ is of class C^1, uniformly bounded, with a uniformly bounded differential.

Note that under (H_D), all discriminators in $\{D_\alpha\}_{\alpha\in\Lambda}$ are θ-admissible, whatever θ. All of these requirements are classic regularity conditions for statistical models, which imply in particular that the functions $\hat{L}(\theta,\alpha)$ and $L(\theta,\alpha)$ are continuous. Therefore, the compactness of Θ guarantees that $\hat{\theta}$ and $\bar{\theta}$ exist. Conditions for the existence of $\theta^\star$ are given in Theorem 2.2.2.

We have known since Theorem 2.3.1 that if the available class of discriminators $\mathscr{D}$ approaches the optimal discriminator $D^\star_{\bar{\theta}}$ by a distance not more than ε, then $D_{\mathrm{JS}}(p_\star,p_{\bar{\theta}})-D_{\mathrm{JS}}(p_\star,p_{\theta^\star})=\mathrm{O}(\varepsilon^2)$. It is therefore reasonable to expect that, asymptotically, the difference $D_{\mathrm{JS}}(p_\star,p_{\hat{\theta}})-D_{\mathrm{JS}}(p_\star,p_{\theta^\star})$ will not be larger than a term proportional to ε^2, in some probabilistic sense. This is precisely the result of Theorem 2.4.1 below. In fact, most articles to date have focused on the development and analysis of optimization procedures (typically, stochastic-gradient-type algorithms) to compute $\hat{\theta}$, without really questioning its convergence properties as the data set grows. Although our statistical results are theoretical in nature, we believe that they are complementary to the optimization literature, insofar as they offer guarantees on the validity of the algorithms.

In addition to the regularity hypotheses, we will need the following requirement, which is a stronger version of (H_ε):

Assumption (H'_ε) There exist $\varepsilon>0$ and $m\in(0,1/2)$ such that: for all $\theta\in\Theta$, there exists $D\in\mathscr{D}$ such that $m\le D\le 1-m$ and $\|D-D^\star_\theta\|_2\le\varepsilon$.

We are ready to state our first statistical theorem.

Theorem 2.4.1. *Assume that, for some $M>0$, $p_\star\le M$ and $p_\theta\le M$ for all $\theta\in\Theta$. Then, under Assumptions (H_{reg}) and (H'_ε) with $\varepsilon<1/(2M)$, one has*

$$\mathbb{E}D_{\mathrm{JS}}(p_\star,p_{\hat{\theta}})-D_{\mathrm{JS}}(p_\star,p_{\theta^\star})=\mathrm{O}\Big(\varepsilon^2+\frac{1}{\sqrt{n}}\Big).$$

Remark 2.4.1. *The constant hidden in the* O *term scales as* $p+q$*. Knowing that (deep) neural networks, and thus GANs, are often used in the so-called overparameterized regime (i.e., when the number of parameters exceeds the number of examples), this limits the impact of the result in the neural network context, at least when* $p+q$ *is large with respect to* $\sqrt{n}$*. For instance, successful applications of GANs on common datasets such as LSUN (*$\sqrt{n} \approx 1\,740$*) and FACES (*$\sqrt{n} \approx 590$*) make use of more than* $1\,500\,000$ *parameters (Radford et al., 2015).*

Proof. Fix $\varepsilon \in (0, 1/(2M))$ as in Assumption (H'_ε), and choose $\hat{D} \in \mathscr{D}$ such that $m \le \hat{D} \le 1-m$ and $\|\hat{D} - D^\star_{\hat{\theta}}\|_2 \le \varepsilon$. By repeating the arguments of the proof of Theorem 2.3.1 (with $\hat{\theta}$ instead of $\bar{\theta}$), we conclude that there exists a constant $c_1 > 0$ such that

$$2D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) \le c_1\varepsilon^2 + L(\hat{\theta}, \hat{D}) + \ln 4 \le c_1\varepsilon^2 + \sup_{\alpha\in\Lambda} L(\hat{\theta}, \alpha) + \ln 4.$$

Therefore,

$$\begin{aligned}
2D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) &\le c_1\varepsilon^2 + \sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| + \sup_{\alpha\in\Lambda} \hat{L}(\hat{\theta},\alpha) + \ln 4 \\
&= c_1\varepsilon^2 + \sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| + \inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \hat{L}(\theta,\alpha) + \ln 4 \\
&\quad \text{(by definition of } \hat{\theta}\text{)} \\
&\le c_1\varepsilon^2 + 2\sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| + \inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} L(\theta,\alpha) + \ln 4.
\end{aligned}$$

So,

$$\begin{aligned}
2D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) &\le c_1\varepsilon^2 + 2\sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| + \inf_{\theta\in\Theta} \sup_{D\in\mathscr{D}_\infty} L(\theta, D) + \ln 4 \\
&= c_1\varepsilon^2 + 2\sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| + L(\theta^\star, D^\star_{\theta^\star}) + \ln 4 \\
&\quad \text{(by definition of } \theta^\star\text{)} \\
&= c_1\varepsilon^2 + 2D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) + 2\sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)|.
\end{aligned}$$

Thus, letting $c_2 = c_1/2$, we have

$$D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \le c_2\varepsilon^2 + \sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)|. \tag{2.4.1}$$

Clearly, under Assumptions (H_D), (H_G), and (H_p), $(\hat{L}(\theta,\alpha) - L(\theta,\alpha))_{\theta\in\Theta,\alpha\in\Lambda}$ is a separable subgaussian process (e.g., van Handel, 2016, Chapter 5) for the distance $d = S\|\cdot\|/\sqrt{n}$, where $\|\cdot\|$ is the standard Euclidean norm on $\mathbb{R}^p \times \mathbb{R}^q$ and $S > 0$ depends only on the bounds in (H_D)

and (H_G). Let $N(\Theta \times \Lambda, \|\cdot\|, u)$ denote the u-covering number of $\Theta \times \Lambda$ for the distance $\|\cdot\|$. Then, by Dudley's inequality (van Handel, 2016, Corollary 5.25),

$$\mathbb{E} \sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)| \leq \frac{12S}{\sqrt{n}} \int_0^\infty \sqrt{\ln(N(\Theta \times \Lambda, \|\cdot\|, u))}\mathrm{d}u. \tag{2.4.2}$$

Since Θ and Λ are bounded, there exists $r > 0$ such that $N(\Theta \times \Lambda, \|\cdot\|, u) = 1$ for $u \geq r$ and

$$N(\Theta \times \Lambda, \|\cdot\|, u) = \mathrm{O}\Big(\Big(\frac{\sqrt{p+q}}{u}\Big)^{p+q}\Big) \quad \text{for } u < r.$$

Combining this inequality with (2.4.1) and (2.4.2), we obtain

$$\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq c_3\Big(\varepsilon^2 + \frac{1}{\sqrt{n}}\Big),$$

for some positive constant c_3 that scales as $p+q$. The conclusion follows by observing that, by (2.2.2),

$$D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}).$$

□

Theorem 2.4.1 is illustrated in Figure 2.1, which shows the approximate values of $\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$. We took $p_\star(x) = \frac{e^{-x/s}}{s(1+e^{-x/s})^2}$ (centered logistic density with scale parameter $s = 0.33$), and let $\mathscr{G}$ and $\mathscr{D}$ be two fully connected neural networks parameterized by weights and offsets. The noise random variable Z follows a uniform distribution on $[0,1]$, and the parameters of $\mathscr{G}$ and $\mathscr{D}$ are chosen in a sufficiently large compact set. In order to illustrate the impact of ε in Theorem 2.4.1, we fixed the sample size to a large $n = 100\,000$ and varied the number of layers of the discriminators from 2 to 5, keeping in mind that a larger number of layers results in a smaller ε. To diversify the setting, we also varied the number of layers of the generators from 2 to 3. The expectation $\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$ was estimated by averaging over 30 repetitions (the number of runs has been reduced for time complexity limitations). Note that we do not pay attention to the exact value of the constant term $D_{\mathrm{JS}}(p_\star, p_{\theta^\star})$, which is intractable in our setting.

Figure 2.1 highlights that $\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$ approaches the constant value $D_{\mathrm{JS}}(p_\star, p_{\theta^\star})$ as $\varepsilon \downarrow 0$, i.e., as the discriminator depth increases, given that the contribution of $1/\sqrt{n}$ is certainly negligible for $n = 100\,000$. Figure 2.2 shows the target density $p_\star$ vs. the histograms and kernel estimates of 100 000 data sampled from $G_{\hat{\theta}}(Z)$, in the two cases: (discriminator depth = 2, generator depth = 3) and (discriminator depth = 5, generator depth = 3). In accordance with the

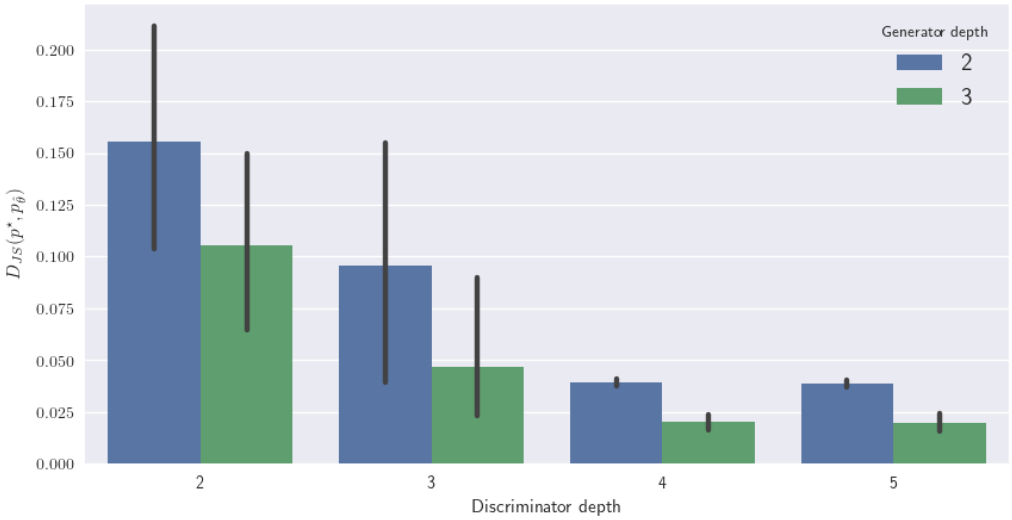
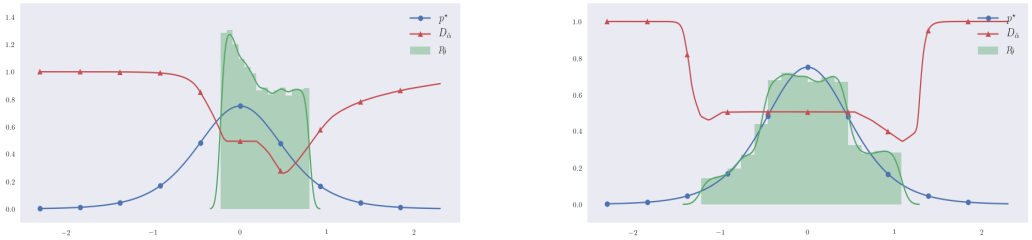


Fig. 2.1 Bar plots of the Jensen-Shannon divergence $D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$ with respect to the number of layers (depth) of both the discriminators and generators. The height of each rectangle estimates $\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$.

decrease of $\mathbb{E}D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}})$, the estimation of the true distribution $p_\star$ improves when ε becomes small.



(a) Discriminator depth = 2, generator depth = 3.(b) Discriminator depth = 5, generator depth = 3.

Fig. 2.2 True density $p_\star$, histograms, and kernel estimates (continuous line) of 100000 data sampled from $G_{\hat{\theta}}(Z)$. Also shown is the final discriminator $D_{\hat{\alpha}}$.

Some comments on the optimization scheme. Numerical optimization is quite a tough point for GANs, partly due to nonconvex-concavity of the saddle point problem described in equation (2.1.2) and the nondifferentiability of the objective function. This motivates a very active line of research (e.g., Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017), which aims at transforming the objective into a more convenient function and

devising efficient algorithms. In the present paper, since we are interested in original GANs, the algorithmic approach described by Goodfellow et al. (2014) is adopted, and numerical optimization is performed thanks to the machine learning framework `TensorFlow`, working with gradient descent based on automatic differentiation. As proposed by Goodfellow et al. (2014), the objective function $\theta \mapsto \sup_{\alpha\in\Lambda} \hat{L}(\theta,\alpha)$ is not directly minimized. We used instead an alternated procedure, which consists in iterating (a few hundred times in our examples) the following two steps:

(i) For a fixed value of θ and from a given value of α, perform 10 ascent steps on $\hat{L}(\theta,\cdot)$;

(ii) For a fixed value of α and from a given value of θ, perform 1 descent step on $\theta \mapsto -\sum_{i=1}^{n} \ln(D_\alpha \circ G_\theta(Z_i))$ (instead of $\theta \mapsto \sum_{i=1}^{n} \ln(1 - D_\alpha \circ G_\theta(Z_i))$).

This alternated procedure is motivated by two reasons. First, for a given θ, approximating $\sup_{\alpha\in\Lambda} \hat{L}(\theta,\alpha)$ is computationally prohibitive and may result in overfitting the finite training sample (Goodfellow et al., 2014). This can be explained by the shape of the function $\theta \mapsto \sup_{\alpha\in\Lambda} \hat{L}(\theta,\alpha)$, which may be almost piecewise constant, resulting in a zero gradient almost everywhere (or at best very low; see Arjovsky et al., 2017). Next, empirically, $-\ln(D_\alpha \circ G_\theta(Z_i))$ provides bigger gradients than $\ln(1 - D_\alpha \circ G_\theta(Z_i))$, resulting in a more powerful algorithm than the original version, while leading to the same minimizers.

In all our experiments, the learning rates needed in gradient steps were fixed and tuned by hand, in order to prevent divergence. In addition, since our main objective is to focus on illustrating the statistical properties of GANs rather than delving into optimization issues, we decided to perform mini-batch gradient updates instead of stochastic ones (that is, new observations of X and Z are not sampled at each step of the procedure). This is different of what is done in the original algorithm of Goodfellow et al. (2014).

All in all, we realize that our numerical approach—although widely adopted by the machine learning community—may fail to locate the desired estimator $\hat{\theta}$ (i.e., the exact minimizer in θ of $\sup_{\alpha\in\Lambda} \hat{L}(\theta,\alpha)$) in more complex contexts than those presented in the present paper. It is nevertheless sufficient for our objective, which is limited to illustrating the theoretical results with a few simple examples.

2.4.2 Asymptotic properties of $\hat{\theta}$

Theorem 2.4.1 states a result relative to the excess of Jensen-Shannon error $D_{\mathrm{JS}}(p_\star, p_{\hat{\theta}}) - D_{\mathrm{JS}}(p_\star, p_{\theta^\star})$. We now examine the convergence properties of the parameter $\hat{\theta}$ itself as the sample size n grows. We would typically like to find reasonable conditions ensuring that $\hat{\theta} \to \bar{\theta}$ almost surely as $n \to \infty$. To reach this goal, we first need to strengthen a bit the Assumptions (H_{reg}), as follows:

Assumptions (H'_{reg})

(H'_D) There exists $\kappa \in (0,1/2)$ such that, for all $\alpha \in \Lambda$, $\kappa \le D_\alpha \le 1-\kappa$. In addition, the function $(x,\alpha) \mapsto D_\alpha(x)$ is of class C^2, with differentials of order 1 and 2 uniformly bounded.

(H'_G) For all $z \in \mathbb{R}^{d'}$, the function $\theta \mapsto G_\theta(z)$ is of class C^2, uniformly bounded, with differentials of order 1 and 2 uniformly bounded.

(H'_p) For all $x \in E$, the function $\theta \mapsto p_\theta(x)$ is of class C^2, uniformly bounded, with differentials of order 1 and 2 uniformly bounded.

It is easy to verify that under these assumptions the partial functions $\theta \mapsto \hat{L}(\theta,\alpha)$ (respectively, $\theta \mapsto L(\theta,\alpha)$) and $\alpha \mapsto \hat{L}(\theta,\alpha)$ (respectively, $\alpha \mapsto L(\theta,\alpha)$) are of class C^2. Throughout, we let $\theta = (\theta_1,\dots,\theta_p)$, $\alpha = (\alpha_1,\dots,\alpha_q)$, and denote by $\frac{\partial}{\partial\theta_i}$ and $\frac{\partial}{\partial\alpha_j}$ the partial derivative operations with respect to θ_i and α_j. The next lemma will be of constant utility. In order not to burden the text, its proof is given in Section 2.A.

Lemma 2.4.1. *Under Assumptions* (H'_{reg}), $\forall (a,b,c,d) \in \{0,1,2\}^4$ *such that* $a+b \le 2$ *and* $c+d \le 2$, *one has*

$$\sup_{\theta\in\Theta,\alpha\in\Lambda} \left| \frac{\partial^{a+b+c+d}}{\partial\theta_i^a \partial\theta_j^b \partial\alpha_\ell^c \partial\alpha_m^d} \hat{L}(\theta,\alpha) - \frac{\partial^{a+b+c+d}}{\partial\theta_i^a \partial\theta_j^b \partial\alpha_\ell^c \partial\alpha_m^d} L(\theta,\alpha) \right| \to 0 \quad \textit{almost surely},$$

for all $(i,j) \in \{1,\dots,p\}^2$ *and* $(\ell,m) \in \{1,\dots,q\}^2$.

We recall that $\bar{\theta} \in \Theta$ is such that

$$\sup_{\alpha\in\Lambda} L(\bar{\theta},\alpha) \le \sup_{\alpha\in\Lambda} L(\theta,\alpha), \quad \forall \theta \in \Theta,$$

and insist that $\bar{\theta}$ exists under (H'_{reg}) by continuity of the function $\theta \mapsto \sup_{\alpha\in\Lambda} L(\theta,\alpha)$. Similarly, there exists $\bar{\alpha} \in \Lambda$ such that

$$L(\bar{\theta},\bar{\alpha}) \ge L(\bar{\theta},\alpha), \quad \forall \alpha \in \Lambda.$$

The following assumption ensures that $\bar{\theta}$ and $\bar{\alpha}$ are uniquely defined, which is of course a key hypothesis for our estimation objective. Throughout, the notation S° (respectively, ∂S) stands for the interior (respectively, the boundary) of the set S.

Assumption (H_1) The pair $(\bar{\theta},\bar{\alpha})$ is unique and belongs to $\Theta^\circ \times \Lambda^\circ$.

Finally, in addition to $\hat{\theta}$, we let $\hat{\alpha} \in \Lambda$ be such that

$$\hat{L}(\hat{\theta}, \hat{\alpha}) \geq \hat{L}(\hat{\theta}, \alpha), \quad \forall \alpha \in \Lambda.$$

Theorem 2.4.2. *Under Assumptions* (H'_{reg}) *and* (H_1)*, one has*

$$\hat{\theta} \to \bar{\theta} \quad \textit{almost surely} \quad \textit{and} \quad \hat{\alpha} \to \bar{\alpha} \quad \textit{almost surely}.$$

Proof. We write

$$\begin{aligned}
&|\sup_{\alpha\in\Lambda} L(\hat{\theta}, \alpha) - \sup_{\alpha\in\Lambda} L(\bar{\theta}, \alpha)| \\
&\quad \leq |\sup_{\alpha\in\Lambda} L(\hat{\theta}, \alpha) - \sup_{\alpha\in\Lambda} \hat{L}(\hat{\theta}, \alpha)| + |\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \hat{L}(\theta, \alpha) - \inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} L(\theta, \alpha)| \\
&\quad \leq 2 \sup_{\theta\in\Theta, \alpha\in\Lambda} |\hat{L}(\theta, \alpha) - L(\theta, \alpha)|.
\end{aligned}$$

Thus, by Lemma 2.4.1, $\sup_{\alpha\in\Lambda} L(\hat{\theta}, \alpha) \to \sup_{\alpha\in\Lambda} L(\bar{\theta}, \alpha)$ almost surely. In the lines that follow, we make more transparent the dependence of $\hat{\theta}$ in the sample size n and set $\hat{\theta}_n \stackrel{\text{def}}{=} \hat{\theta}$. Since $\hat{\theta}_n \in \Theta$ and Θ is compact, we can extract from any subsequence of $(\hat{\theta}_n)_n$ a subsequence $(\hat{\theta}_{n_k})_k$ such that $\hat{\theta}_{n_k} \to z \in \Theta$ (with $n_k = n_k(\omega)$, i.e., it is almost surely defined). By continuity of the function $\theta \mapsto \sup_{\alpha\in\Lambda} L(\theta, \alpha)$, we deduce that $\sup_{\alpha\in\Lambda} L(\hat{\theta}_{n_k}, \alpha) \to \sup_{\alpha\in\Lambda} L(z, \alpha)$, and so $\sup_{\alpha\in\Lambda} L(z, \alpha) = \sup_{\alpha\in\Lambda} L(\bar{\theta}, \alpha)$. Since $\bar{\theta}$ is unique by (H_1), we have $z = \bar{\theta}$. In conclusion, we can extract from each subsequence of $(\hat{\theta}_n)_n$ a subsequence that converges towards $\bar{\theta}$: this shows that $\hat{\theta}_n \to \bar{\theta}$ almost surely.

Finally, we have

$$\begin{aligned}
&|L(\bar{\theta}, \hat{\alpha}) - L(\bar{\theta}, \bar{\alpha})| \\
&\quad \leq |L(\bar{\theta}, \hat{\alpha}) - L(\hat{\theta}, \hat{\alpha})| + |L(\hat{\theta}, \hat{\alpha}) - \hat{L}(\hat{\theta}, \hat{\alpha})| + |\hat{L}(\hat{\theta}, \hat{\alpha}) - L(\bar{\theta}, \bar{\alpha})| \\
&\quad = |L(\bar{\theta}, \hat{\alpha}) - L(\hat{\theta}, \hat{\alpha})| + |L(\hat{\theta}, \hat{\alpha}) - \hat{L}(\hat{\theta}, \hat{\alpha})| + |\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \hat{L}(\theta, \alpha) - \inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} L(\theta, \alpha)| \\
&\quad \leq \sup_{\alpha\in\Lambda} |L(\bar{\theta}, \alpha) - L(\hat{\theta}, \alpha)| + 2 \sup_{\theta\in\Theta, \alpha\in\Lambda} |\hat{L}(\theta, \alpha) - L(\theta, \alpha)|.
\end{aligned}$$

Using Assumptions (H'_D) and (H'_p), and the fact that $\hat{\theta} \to \bar{\theta}$ almost surely, we see that the first term above tends to zero. The second one vanishes asymptotically by Lemma 2.4.1, and we conclude that $L(\bar{\theta}, \hat{\alpha}) \to L(\bar{\theta}, \bar{\alpha})$ almost surely. Since $\hat{\alpha} \in \Lambda$ and Λ is compact, we may argue as in the first part of the proof and deduce from the uniqueness of $\bar{\alpha}$ that $\hat{\alpha} \to \bar{\alpha}$ almost surely. □

To illustrate the result of Theorem 2.4.2, we undertook a series of small numerical experiments with three choices for the triplet (true $p_\star$ + generator model $\mathscr{P} = \{p_\theta\}_{\theta\in\Theta}$ + discriminator family $\mathscr{D} = \{D_\alpha\}_{\alpha\in\Lambda}$), which we respectively call the **Laplace-Gaussian**, **Claw-Gaussian**, and **Exponential-Uniform** model. They are summarized in Table 2.1. We are aware that more elaborate models (involving, for example, neural networks) can be designed and implemented. However, our objective is not to conduct a series of extensive simulations, but simply to illustrate our theoretical results with a few graphs to get some better intuition and provide a sanity check. We stress in particular that these experiments are in one dimension and are therefore very limited compared to the way GANs algorithms are typically used in practice.

Model	$p_\star$	$\mathscr{P} = \{p_\theta\}_{\theta\in\Theta}$	$\mathscr{D} = \{D_\alpha\}_{\alpha\in\Lambda}$
Laplace-Gaussian	$\frac{1}{2b}e^{-\frac{\lvert x\rvert}{b}}$	$\frac{1}{\sqrt{2\pi}\theta}e^{-\frac{x^2}{2\theta^2}}$	$\frac{1}{1+\frac{\alpha_1}{\alpha_0}e^{\frac{x^2}{2}(\alpha_1^{-2}-\alpha_0^{-2})}}$
	$b = 1.5$	$\Theta = [10^{-1}, 10^3]$	$\Lambda = \Theta\times\Theta$
Claw-Gaussian	$p_{\text{claw}}(x)$	$\frac{1}{\sqrt{2\pi}\theta}e^{-\frac{x^2}{2\theta^2}}$	$\frac{1}{1+\frac{\alpha_1}{\alpha_0}e^{\frac{x^2}{2}(\alpha_1^{-2}-\alpha_0^{-2})}}$
		$\Theta = [10^{-1}, 10^3]$	$\Lambda = \Theta\times\Theta$
Exponential-Uniform	$\lambda e^{-\lambda x}$	$\frac{1}{\theta}\mathbf{1}_{[0,\theta]}(x)$	$\frac{1}{1+\frac{\alpha_1}{\alpha_0}e^{\frac{x^2}{2}(\alpha_1^{-2}-\alpha_0^{-2})}}$
	$\lambda = 1$	$\Theta = [10^{-3}, 10^3]$	$\Lambda = \Theta\times\Theta$

Table 2.1 Triplets used in the numerical experiments.

Figure 2.3 shows the densities $p_\star$. We recall that the claw density on $[0,\infty)$ takes the form

$$p_{\text{claw}} = \frac{1}{2}\varphi(0,1) + \frac{1}{10}\big(\varphi(-1,0.1) + \varphi(-0.5,0.1) + \varphi(0,0.1) + \varphi(0.5,0.1) + \varphi(1,0.1)\big),$$

where $\varphi(\mu,\sigma)$ is a Gaussian density with mean μ and standard deviation σ (this density is borrowed from Devroye, 1997).

In the **Laplace-Gaussian** and **Claw-Gaussian** examples, the densities p_θ are centered Gaussian, parameterized by their standard deviation parameter θ. The random variable Z is uniform $[0,1]$ and the natural family of generators associated with the model $\mathscr{P} = \{p_\theta\}_{\theta\in\Theta}$ is $\mathscr{G} = \{G_\theta\}_{\theta\in\Theta}$, where each G_θ is the generalized inverse of the cumulative distribution function of p_θ (because $G_\theta(Z) \overset{\mathscr{L}}{=} p_\theta \mathrm{d}\mu$). The rationale behind our choice for the discriminators is based on the form of the optimal discriminator $D^\star_\theta$ described in (2.2.1): starting from

$$D^\star_\theta = \frac{p_\star}{p_\star + p_\theta}, \quad \theta\in\Theta,$$

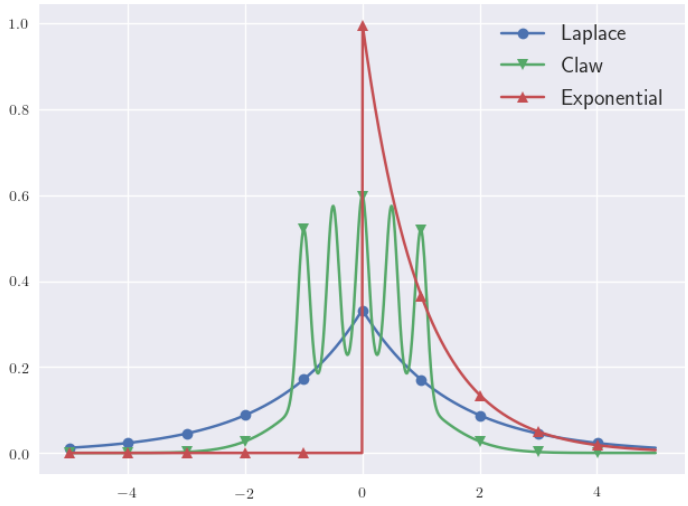


Fig. 2.3 Probability density functions $p_\star$ used in the numerical experiments.

we logically consider the following ratio

$$D_\alpha = \frac{p_{\alpha_1}}{p_{\alpha_1} + p_{\alpha_0}}, \quad \alpha = (\alpha_0, \alpha_1) \in \Lambda = \Theta \times \Theta.$$

Figure 2.4 (**Laplace-Gaussian**), Figure 2.5 (**Claw-Gaussian**), and Figure 2.6 (**Exponential-Uniform**) show the boxplots of the differences $\hat{\theta} - \bar{\theta}$ over 200 repetitions, for a sample size n varying from 10 to 10000. In these experiments, the parameter $\bar{\theta}$ is obtained by averaging the $\hat{\theta}$ for the largest sample size n. In accordance with Theorem 2.4.2, the size of the boxplots shrinks around 0 when n increases, thus showing that the estimated parameter $\hat{\theta}$ is getting closer and closer to $\bar{\theta}$. Before analyzing at which rate this convergence occurs, we may have a look at Figure 2.7, which plots the estimated density $p_{\hat{\theta}}$ (for $n = 10000$) vs. the true density $p_\star$. It also shows the discriminator $D_{\hat{\alpha}}$, together with the initial density $p_{\theta_{\text{init}}}$ and the initial discriminator $D_{\alpha_{\text{init}}}$ fed into the optimization algorithm. We note that in the three models, $D_{\hat{\alpha}}$ is almost identically $1/2$, meaning that it is impossible to discriminate between the original observations and those generated by $p_{\hat{\theta}}$.

In line with the above, our next step is to state a central limit theorem for $\hat{\theta}$. Although simple to understand, this result requires additional assumptions and some technical prerequisites. One first needs to ensure that the function $(\theta, \alpha) \mapsto L(\theta, \alpha)$ is regular enough in a neighborhood of $(\bar{\theta}, \bar{\alpha})$. This is captured by the following set of assumptions, which require in particular the uniqueness of the maximizer of the function $\alpha \mapsto L(\theta, \alpha)$ for a θ around $\bar{\theta}$. For a function $F : \Theta \to \mathbb{R}$ (respectively, $G : \Theta \times \Lambda \to \mathbb{R}$), we let $HF(\theta)$ (respectively, $H_1 G(\theta, \alpha)$

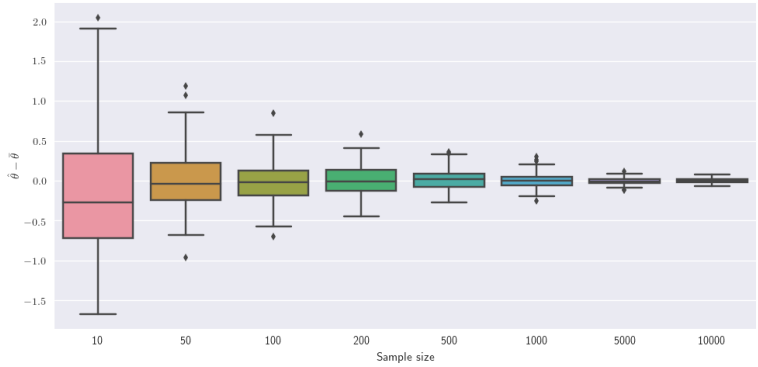


Fig. 2.4 Boxplots of $\hat{\theta} - \bar{\theta}$ for different sample sizes (**Laplace-Gaussian** model, 200 repetitions).

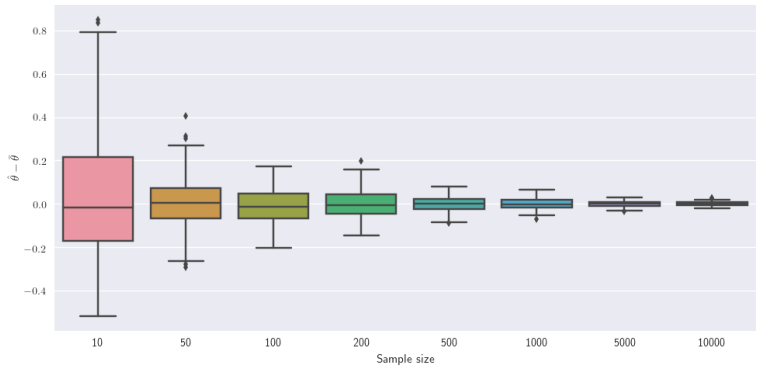


Fig. 2.5 Boxplots of $\hat{\theta} - \bar{\theta}$ for different sample sizes (**Claw-Gaussian** model, 200 repetitions).

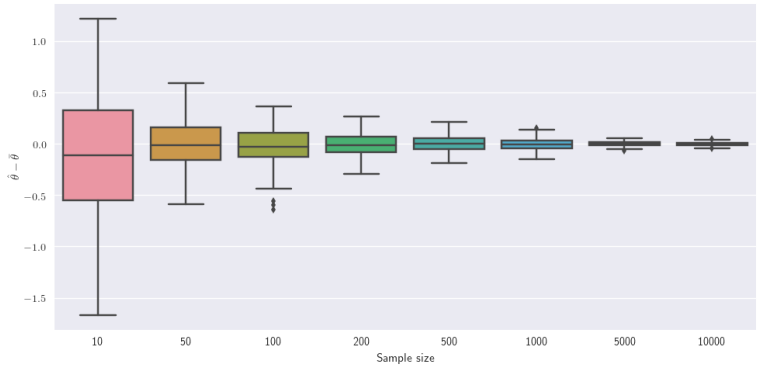


Fig. 2.6 Boxplots of $\hat{\theta} - \bar{\theta}$ for different sample sizes (**Exponential-Uniform** model, 200 repetitions).

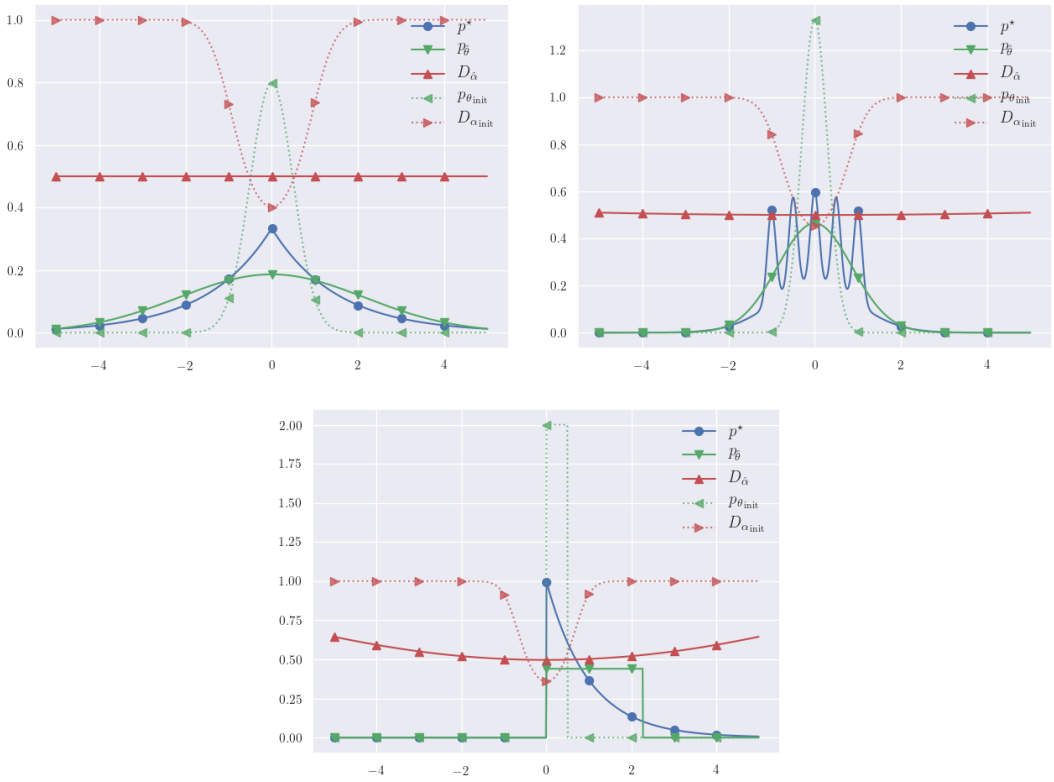


Fig. 2.7 True density $p_\star$, estimated density $p_{\hat\theta}$, and discriminator $D_{\hat\alpha}$ for $n = 10\,000$ (from left to right: **Laplace-Gaussian**, **Claw-Gaussian**, and **Exponential-Uniform** model). Also shown are the initial density $p_{\theta_{\text{init}}}$ and the initial discriminator $D_{\alpha_{\text{init}}}$ fed into the optimization algorithm.

and $H_2G(\theta,\alpha)$) be the Hessian matrix of the function $\theta \mapsto F(\theta)$ (respectively, $\theta \mapsto G(\theta,\alpha)$ and $\alpha \mapsto G(\theta,\alpha)$) computed at θ (respectively, at θ and α).

Assumptions (H_{loc})

(H_U) There exists a neighborhood U of $\bar\theta$ and a function $\alpha : U \to \Lambda$ such that

$$\arg\max_{\alpha\in\Lambda} L(\theta,\alpha) = \{\alpha(\theta)\}, \quad \forall \theta \in U.$$

(H_V) The Hessian matrix $HV(\bar\theta)$ is invertible, where $V(\theta) \stackrel{\text{def}}{=} L(\theta,\alpha(\theta))$.

(H_H) The Hessian matrix $H_2L(\bar\theta,\bar\alpha)$ is invertible.

We stress that under Assumption (H_U), there is for each $\theta \in U$ a unique $\alpha(\theta) \in \Lambda$ such that $L(\theta,\alpha(\theta)) = \sup_{\alpha\in\Lambda} L(\theta,\alpha)$. We also note that $\alpha(\bar\theta) = \bar\alpha$ under (H_1). We still need some notation before we state the central limit theorem. For a function $f(\theta,\alpha)$, $\nabla_1 f(\theta,\alpha)$ (respectively, $\nabla_2 f(\theta,\alpha)$) means the gradient of the function $\theta \mapsto f(\theta,\alpha)$ (respectively, the function $\alpha \mapsto f(\theta,\alpha)$) computed at θ (respectively, at α). For a function $g(t)$, $J(g)_t$ is the Jacobian matrix of g computed at t. Observe that by the envelope theorem,

$$HV(\bar\theta) = H_1L(\bar\theta,\bar\alpha) + J(\nabla_1 L(\bar\theta,\cdot))_{\bar\alpha} J(\alpha)_{\bar\theta},$$

where, by the chain rule,

$$J(\alpha)_{\bar\theta} = -H_2L(\bar\theta,\bar\alpha)^{-1} J(\nabla_2 L(\cdot,\bar\alpha))_{\bar\theta}.$$

Therefore, in Assumption(H_V), the Hessian matrix $HV(\bar\theta)$ can be computed with the sole knowledge of L. Finally, we let

$$\ell_1(\theta,\alpha) = \ln D_\alpha(X_1) + \ln(1 - D_\alpha \circ G_\theta(Z_1)),$$

and denote by $\stackrel{\mathscr{L}}{\to}$ the convergence in distribution.

Theorem 2.4.3. *Under Assumptions* (H'_{reg}), (H_1), *and* (H_{loc}), *one has*

$$\sqrt{n}(\hat\theta - \bar\theta) \stackrel{\mathscr{L}}{\to} Z,$$

where Z is a Gaussian random variable with mean 0 *and covariance matrix*

$$\mathbf{V} = \mathrm{Var}\big[-HV(\bar\theta)^{-1}\nabla_1\ell_1(\bar\theta,\bar\alpha) + HV(\bar\theta)^{-1} J(\nabla_1 L(\bar\theta,\cdot))_{\bar\alpha} H_2L(\bar\theta,\bar\alpha)^{-1}\nabla_2\ell_1(\bar\theta,\bar\alpha)\big].$$

The expression of the covariance is relatively complex and, unfortunately, cannot be simplified, even for a dimension of the parameter equal to 1. We note however that if Y is a random vector of $\mathbb{R}^p$ whose components are bounded in absolute value by some $\delta > 0$, then the Euclidean norm of the covariance matrix of Y is bounded by $4p\delta^2$. But each component of the random vector of $\mathbb{R}^p$ involved in the covariance matrix $\mathbf{V}$ is bounded in absolute value by Cpq^2, for some positive constant C resulting from Assumption (H'_{reg}). We conclude that the Euclidean norm of $\mathbf{V}$ is bounded by $4C^2p^3q^4$. Thus, our statistical approach reveals that in the overparameterized regime (i.e, when p and q are very large compared to n), the estimator $\hat{\theta}$ has a large dispersion around $\bar{\theta}$, which may affects the performance of the algorithm.

Nevertheless, the take-home message of Theorem 2.4.3 is that the estimator $\hat{\theta}$ is asymptotically normal, with a convergence rate of $\sqrt{n}$. This is illustrated in Figures 2.8, 2.9, and 2.10, which respectively show the histograms and kernel estimates of the distribution of $\sqrt{n}(\hat{\theta} - \bar{\theta})$ for the **Laplace-Gaussian**, the **Claw-Gaussian**, and the **Exponential-Uniform** model in function of the sample size n (200 repetitions).

Proof. By technical Lemma 2.A.1, we can find under Assumptions (H'_{reg}) and (H_1) an open set $V \subset U \subset \Theta^\circ$ containing $\bar{\theta}$ such that, for all $\theta \in V$, $\alpha(\theta) \in \Lambda^\circ$. In the sequel, to lighten the notation, we assume without loss of generality that $V = U$. Thus, for all $\theta \in U$, we have $\alpha(\theta) \in \Lambda^\circ$ and $L(\theta, \alpha(\theta)) = \sup_{\alpha \in \Lambda} L(\theta, \alpha)$ (with $\alpha(\bar{\theta}) = \bar{\alpha}$ by (H_1)). Accordingly, $\nabla_2 L(\theta, \alpha(\theta)) = 0, \forall \theta \in U$. Also, since $H_2 L(\bar{\theta}, \bar{\alpha})$ is invertible by (H_H) and since the function $(\theta, \alpha) \mapsto H_2 L(\theta, \alpha)$ is continuous, there exists an open set $U' \subset U$ such that $H_2 L(\theta, \alpha)$ is invertible as soon as $(\theta, \alpha) \in (U', \alpha(U'))$. Without loss of generality, we assume that $U' = U$. Thus, by the chain rule, the function α is of class C^2 in a neighborhood $U' \subset U$ of $\bar{\theta}$, say $U' = U$, with Jacobian matrix given by

$$J(\alpha)_\theta = -H_2 L(\theta, \alpha(\theta))^{-1} J\big(\nabla_2 L(\cdot, \alpha(\theta))\big)_\theta, \quad \forall \theta \in U.$$

We note that $H_2 L(\theta, \alpha(\theta))^{-1}$ is of format $q \times q$ and $J(\nabla_2 L(\cdot, \alpha(\theta)))_\theta$ of format $q \times p$.

Now, for each $\theta \in U$, we let $\hat{\alpha}(\theta)$ be such that $\hat{L}(\theta, \hat{\alpha}(\theta)) = \sup_{\alpha \in \Lambda} \hat{L}(\theta, \alpha)$. Clearly,

$$\begin{aligned} |L(\theta, \hat{\alpha}(\theta)) - L(\theta, \alpha(\theta))| &\leq |L(\theta, \hat{\alpha}(\theta)) - \hat{L}(\theta, \hat{\alpha}(\theta))| + |\hat{L}(\theta, \hat{\alpha}(\theta)) - L(\theta, \alpha(\theta))| \\ &\leq \sup_{\alpha \in \Lambda} |L(\theta, \alpha) - \hat{L}(\theta, \alpha)| + |\sup_{\alpha \in \Lambda} \hat{L}(\theta, \alpha) - \sup_{\alpha \in \Lambda} L(\theta, \alpha)| \\ &\leq 2 \sup_{\alpha \in \Lambda} |\hat{L}(\theta, \alpha) - L(\theta, \alpha)|. \end{aligned}$$

Therefore, by Lemma 2.4.1, $\sup_{\theta \in U} |L(\theta, \hat{\alpha}(\theta)) - L(\theta, \alpha(\theta))| \to 0$ almost surely. The event on which this convergence holds does not depend upon $\theta \in U$, and, arguing as in the proof of

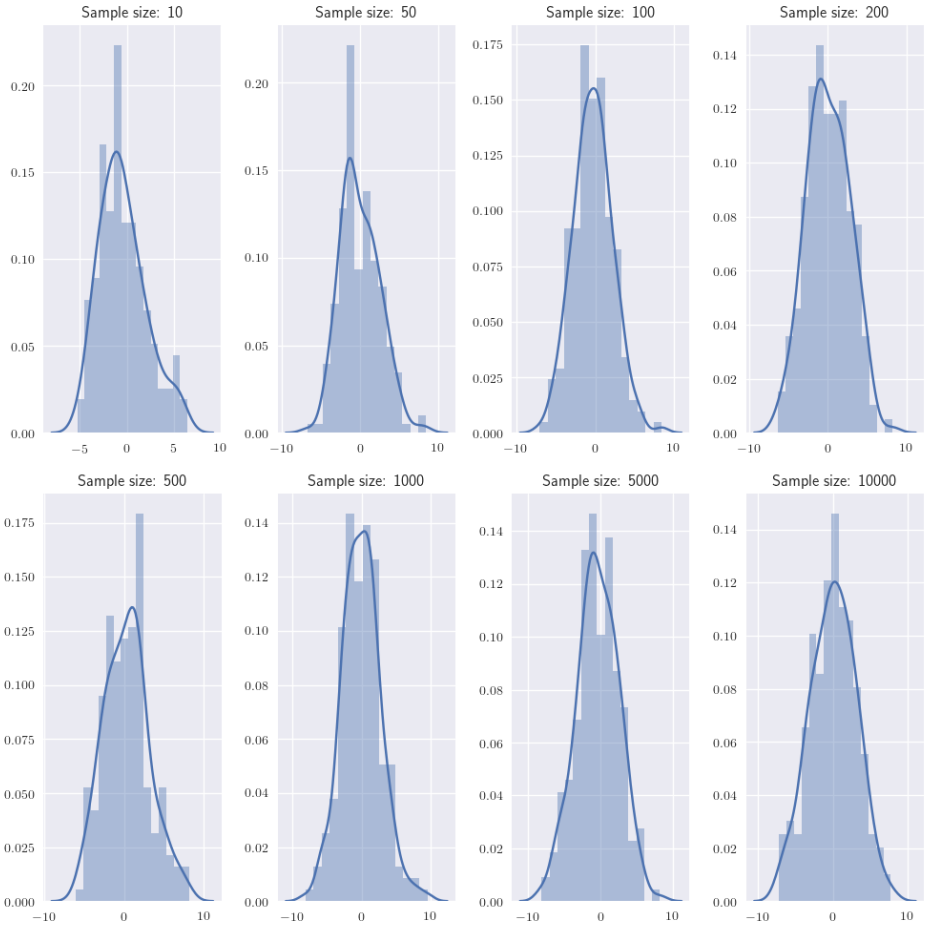


Fig. 2.8 Histograms and kernel estimates (continuous line) of the distribution of $\sqrt{n}(\hat{\theta}-\bar{\theta})$ for different sample sizes n (**Laplace-Gaussian** model, 200 repetitions).

Theorem 2.4.2, we deduce that under (H_1), $\mathbb{P}(\hat{\alpha}(\theta)\to\alpha(\theta)\forall\theta\in U)=1$. Since $\alpha(\theta)\in\Lambda^\circ$ for all $\theta\in U$, we also have $\mathbb{P}(\hat{\alpha}(\theta)\in\Lambda^\circ\forall\theta\in U)\to 1$ as $n\to\infty$. Thus, in the sequel, it will be assumed without loss of generality that, for all $\theta\in U$, $\hat{\alpha}(\theta)\in\Lambda^\circ$.

Still by Lemma 2.4.1, $\sup_{\theta\in\Theta,\alpha\in\Lambda}\|H_2\hat{L}(\theta,\alpha)-H_2L(\theta,\alpha)\|\to 0$ almost surely. Since $H_2L(\theta,\alpha)$ is invertible on $U\times\alpha(U)$, we have

$$\mathbb{P}\big(H_2\hat{L}(\theta,\alpha)\text{ invertible }\forall(\theta,\alpha)\in U\times\alpha(U)\big)\to 1.$$

Thus, we may and will assume that $H_2\hat{L}(\theta,\alpha)$ is invertible for all $(\theta,\alpha)\in U\times\alpha(U)$.

Next, since $\hat{\alpha}(\theta)\in\Lambda^\circ$ for all $\theta\in U$, one has $\nabla_2\hat{L}(\theta,\hat{\alpha}(\theta))=0$. Therefore, by the chain rule, $\hat{\alpha}$ is of class C^2 on U, with Jacobian matrix

$$J(\hat{\alpha})_\theta=-H_2\hat{L}(\theta,\hat{\alpha}(\theta))^{-1}J\big(\nabla_2\hat{L}(\cdot,\hat{\alpha}(\theta))\big)_\theta,\quad\forall\theta\in U.$$

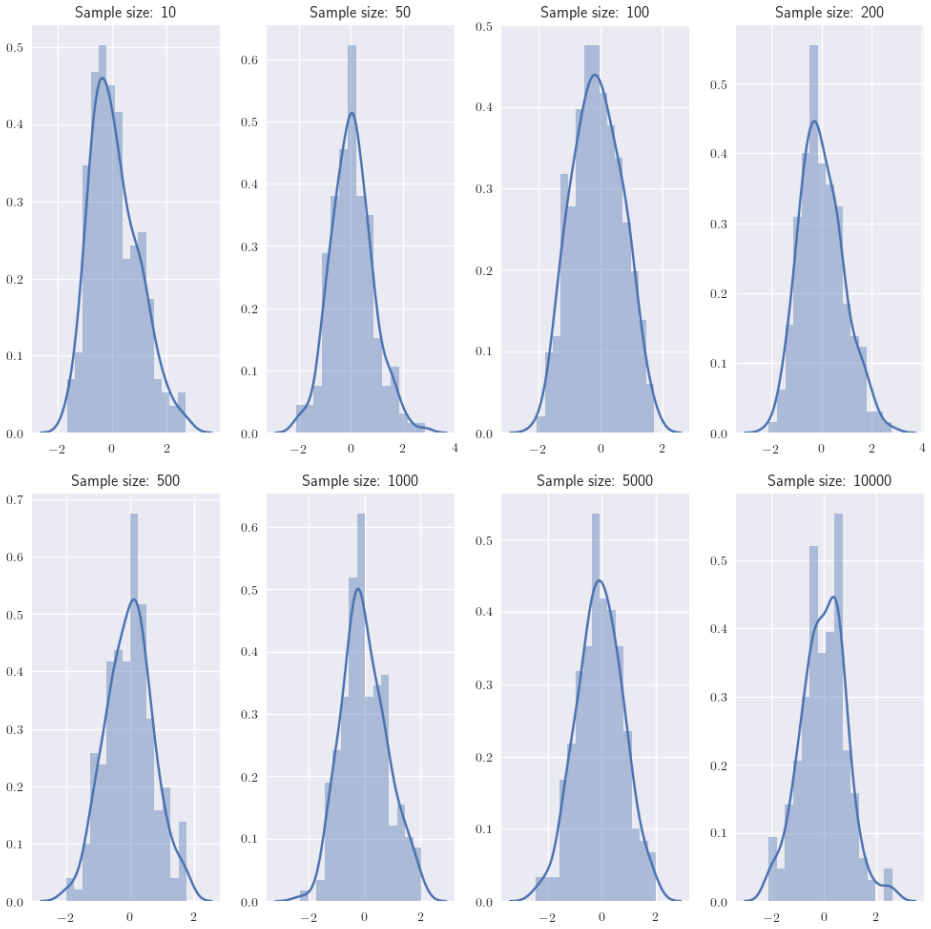


Fig. 2.9 Histograms and kernel estimates (continuous line) of the distribution of $\sqrt{n}(\hat{\theta}-\bar{\theta})$ for different sample sizes n (**Claw-Gaussian** model, 200 repetitions).

Let $\hat{V}(\theta) \stackrel{\text{def}}{=} \hat{L}(\theta,\hat{\alpha}(\theta)) = \sup_{\alpha\in\Lambda}\hat{L}(\theta,\alpha)$. By the envelope theorem, $\hat{V}$ is of class C^2, $\nabla\hat{V}(\theta) = \nabla_1\hat{L}(\theta,\hat{\alpha}(\theta))$, and $H\hat{V}(\theta) = H_1\hat{L}(\theta,\hat{\alpha}(\theta)) + J(\nabla_1\hat{L}(\theta,\cdot))_{\hat{\alpha}(\theta)}J(\hat{\alpha})_\theta$. Recall that $\hat{\theta}\to\bar{\theta}$ almost surely by Theorem 2.4.2, so that we may assume that $\hat{\theta}\in\Theta^\circ$ by (H_1). Moreover, we can also assume that $\hat{\theta}+t(\hat{\theta}-\bar{\theta})\in U$, $\forall t\in[0,1]$. Thus, by a Taylor series expansion with integral remainder, we have

$$0 = \nabla\hat{V}(\hat{\theta}) = \nabla\hat{V}(\bar{\theta}) + \int_0^1 H\hat{V}(\hat{\theta}+t(\hat{\theta}-\bar{\theta}))\mathrm{d}t(\hat{\theta}-\bar{\theta}). \tag{2.4.3}$$

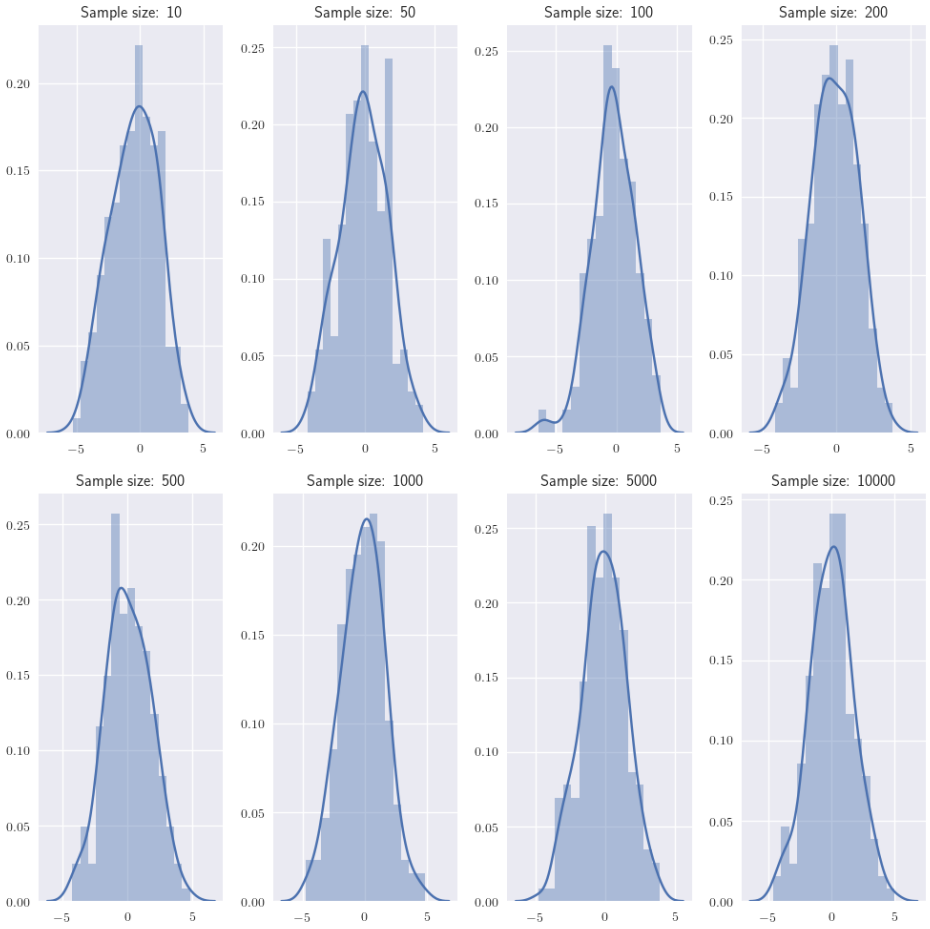


Fig. 2.10 Histograms and kernel estimates (continuous line) of the distribution of $\sqrt{n}(\hat{\theta}-\bar{\theta})$ for different sample sizes n (**Exponential-Uniform** model, 200 repetitions).

Since $\hat{\alpha}(\bar{\theta}) \in \Lambda^{\circ}$ and $\hat{L}(\bar{\theta}, \hat{\alpha}(\bar{\theta})) = \sup_{\alpha \in \Lambda} \hat{L}(\bar{\theta}, \alpha)$, one has $\nabla_2 \hat{L}(\bar{\theta}, \hat{\alpha}(\bar{\theta})) = 0$. Thus,

$$\begin{aligned} 0 &= \nabla_2 \hat{L}(\bar{\theta}, \hat{\alpha}(\bar{\theta})) \\ &= \nabla_2 \hat{L}(\bar{\theta}, \alpha(\bar{\theta})) + \int_0^1 H_2 \hat{L}\big(\bar{\theta}, \alpha(\bar{\theta}) + t(\hat{\alpha}(\bar{\theta}) - \alpha(\bar{\theta}))\big) \mathrm{d}t \, (\hat{\alpha}(\bar{\theta}) - \alpha(\bar{\theta})). \end{aligned}$$

By Lemma 2.4.1, since $\hat{\alpha}(\bar{\theta}) \to \alpha(\bar{\theta})$ almost surely, we have

$$\hat{I}_1 \stackrel{\text{def}}{=} \int_0^1 H_2 \hat{L}\big(\bar{\theta}, \alpha(\bar{\theta}) + t(\hat{\alpha}(\bar{\theta}) - \alpha(\bar{\theta}))\big) \mathrm{d}t \to H_2 L(\bar{\theta}, \bar{\alpha}) \quad \text{almost surely.}$$

Because $H_2L(\bar{\theta},\bar{\alpha})$ is invertible, $\mathbb{P}(\hat{I}_1 \text{ invertible}) \to 1$ as $n \to \infty$. Therefore, we may assume, without loss of generality, that $\hat{I}_1$ is invertible. Hence,

$$\hat{\alpha}(\bar{\theta}) - \alpha(\bar{\theta}) = -\hat{I}_1^{-1}\nabla_2\hat{L}(\bar{\theta},\alpha(\bar{\theta})). \tag{2.4.4}$$

Furthermore,

$$\nabla\hat{V}(\bar{\theta}) = \nabla_1\hat{L}(\bar{\theta},\hat{\alpha}(\bar{\theta})) = \nabla_1\hat{L}(\bar{\theta},\alpha(\bar{\theta})) + \hat{I}_2(\hat{\alpha}(\bar{\theta}) - \alpha(\bar{\theta})),$$

where

$$\hat{I}_2 \overset{\text{def}}{=} \int_0^1 J(\nabla_1\hat{L}(\bar{\theta},\cdot))_{\alpha(\bar{\theta})+t(\hat{\alpha}(\bar{\theta})-\alpha(\bar{\theta}))}\mathrm{d}t.$$

By Lemma 2.4.1, $\hat{I}_2 \to J(\nabla_1 L(\bar{\theta},\cdot))_{\alpha(\bar{\theta})}$ almost surely. Combining (2.4.3) and (2.4.4), we obtain

$$0 = \nabla_1\hat{L}(\bar{\theta},\alpha(\bar{\theta})) - \hat{I}_2\hat{I}_1^{-1}\nabla_2\hat{L}(\bar{\theta},\alpha(\bar{\theta})) + \hat{I}_3(\hat{\theta} - \bar{\theta}),$$

where

$$\hat{I}_3 \overset{\text{def}}{=} \int_0^1 H\hat{V}(\hat{\theta} + t(\hat{\theta} - \bar{\theta}))\mathrm{d}t.$$

By technical Lemma 2.A.2, we have $\hat{I}_3 \to HV(\bar{\theta})$ almost surely. So, by (H_V), it can be assumed that $\hat{I}_3$ is invertible. Consequently,

$$\hat{\theta} - \bar{\theta} = -\hat{I}_3^{-1}\nabla_1\hat{L}(\bar{\theta},\alpha(\bar{\theta})) + \hat{I}_3^{-1}\hat{I}_2\hat{I}_1^{-1}\nabla_2\hat{L}(\bar{\theta},\alpha(\bar{\theta})),$$

or, equivalently, since $\alpha(\bar{\theta}) = \bar{\alpha}$,

$$\hat{\theta} - \bar{\theta} = -\hat{I}_3^{-1}\nabla_1\hat{L}(\bar{\theta},\bar{\alpha}) + \hat{I}_3^{-1}\hat{I}_2\hat{I}_1^{-1}\nabla_2\hat{L}(\bar{\theta},\bar{\alpha}).$$

Using Lemma 2.4.1, we conclude that $\sqrt{n}(\hat{\theta} - \bar{\theta})$ has the same limit distribution as

$$S_n \overset{\text{def}}{=} -\sqrt{n}HV(\bar{\theta})^{-1}\nabla_1\hat{L}(\bar{\theta},\bar{\alpha}) + \sqrt{n}HV(\bar{\theta})^{-1}J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}}H_2L(\bar{\theta},\bar{\alpha})^{-1}\nabla_2\hat{L}(\bar{\theta},\bar{\alpha}).$$

Let

$$\ell_i(\theta,\alpha) = \ln D_\alpha(X_i) + \ln(1 - D_\alpha \circ G_\theta(Z_i)), \quad 1 \le i \le n.$$

With this notation, we have

$$S_n = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\Big(-HV(\bar{\theta})^{-1}\nabla_1\ell_i(\bar{\theta},\bar{\alpha}) + HV(\bar{\theta})^{-1}J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}}H_2L(\bar{\theta},\bar{\alpha})^{-1}\nabla_2\ell_i(\bar{\theta},\bar{\alpha})\Big).$$

One has $\nabla V(\bar{\theta}) = 0$, since $V(\bar{\theta}) = \inf_{\theta\in\Theta} V(\theta)$ and $\bar{\theta} \in \Theta^\circ$. Therefore, under (H'_{reg}), $\mathbb{E}\nabla_1\ell_i(\bar{\theta},\bar{\alpha}) = \nabla_1\mathbb{E}\ell_i(\bar{\theta},\bar{\alpha}) = \nabla_1 L(\bar{\theta},\bar{\alpha}) = \nabla V(\bar{\theta}) = 0$. Similarly, $\mathbb{E}\nabla_2\ell_i(\bar{\theta},\bar{\alpha}) = \nabla_2\mathbb{E}\ell_i(\bar{\theta},\bar{\alpha}) = \nabla_2 L(\bar{\theta},\bar{\alpha}) = 0$, since $L(\bar{\theta},\bar{\alpha}) = \sup_{\alpha\in\Lambda} L(\bar{\theta},\alpha)$ and $\bar{\alpha} \in \Lambda^\circ$. Using the central limit theorem, we conclude that

$$\sqrt{n}(\hat{\theta} - \bar{\theta}) \xrightarrow{\mathscr{L}} Z,$$

where Z is a Gaussian random variable with mean 0 and covariance matrix

$$\mathbf{V} = \mathrm{Var}\big[-HV(\bar{\theta})^{-1}\nabla_1\ell_1(\bar{\theta},\bar{\alpha}) + HV(\bar{\theta})^{-1}J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}}H_2L(\bar{\theta},\bar{\alpha})^{-1}\nabla_2\ell_1(\bar{\theta},\bar{\alpha})\big].$$

□

2.5 Conclusion and perspectives

In this paper, we have presented a theoretical study of the original Generative Adversarial Networks (GAN) algorithm, which consists in building a generative model of an unknown distribution from samples from that distribution. The key idea of the procedure is to simultaneously train the generative model (the generators) and an adversary (the discriminators) that tries to distinguish between real and generated samples. We made a small step towards a better understanding of this generative process by analyzing some optimality properties of the problem in terms of Jensen-Shannon divergence in Section 2.2, and explored the role of the discriminator family via approximation arguments in Section 2.3. Finally, taking a statistical view, we studied in Section 2.4 some large sample properties (convergence and asymptotic normality) of the parameter describing the empirically selected generator. Some numerical experiments were conducted to illustrate the results.

The point of view embraced in the article is statistical, in that it takes into account the variability of the data and its impact on the quality of the estimators. This point of view is different from the classical approach encountered in the literature on GANs, which mainly focuses on the effective computation of the parameters using optimization procedures. In this sense, our results must be thought of as a complementary insight. We realize however that the simplified context in which we have placed ourselves, as well as some of the assumptions we have made, are quite far from the typical situations in which GANs algorithms are used. Thus, our work should be seen as a first step towards a more realistic understanding of GANs, and certainly not as a definitive explanation for their excellent practical performance. We give below three avenues of theoretical research that we believe should be explored as a priority.

1. One of the basic assumptions is that the family of densities $\{p_\theta\}_{\theta\in\Theta}$ (associated with the generators $\{G_\theta\}_{\theta\in\Theta}$) and the unknown density $p_\star$ are dominated by the same measure

μ on the same subset E of $\mathbb{R}^d$. In a way, this means that we already have some kind of information on the support of $p_\star$, which will typically be a manifold in $\mathbb{R}^d$ of dimension smaller than d' (the dimension of Z). Therefore, the random variable Z, the dimension d' of the so-called latent space $\mathbb{R}^{d'}$, and the parametric model $\{G_\theta\}_{\theta\in\Theta}$ should be carefully tuned in order to match this constraint. From a practical perspective, the original article of Goodfellow et al. (2014) suggests using for Z a uniform or Gaussian distribution of small dimension, without further investigation. Mirza and Osindero (2014) and Radford et al. (2015), who have surprisingly good practical results with a deep convolutional generator, both use a 100-dimensional uniform distribution to represent respectively 28×28 and 64×64 pixel images. Many papers have been focusing on either decomposing the latent space $\mathbb{R}^{d'}$ to force specified portions of this space to correspond to different variations (as, e.g., in Donahue et al., 2018) or inverting the generators (e.g., Lipton and Tripathi, 2017; Srivastava et al., 2017; Bojanowski et al., 2018). However, to the best of our knowledge, there is to date no theoretical result tackling the impact of d' and Z on the performance of GANs, and it is our belief that a thorough mathematical investigation of this issue is needed for a better understanding of the generating process. Similarly, whenever the $\{G_\theta\}_{\theta\in\Theta}$ are neural networks, the link between the networks (number of layers, dimensionality of Θ, etc.) and the target $p_\star$ (support, dominating measure, etc.) is also a fundamental question, which should be addressed at a theoretical level.

2. Assumptions (H_ε) and (H'_ε) highlight the essential role played by the discriminators to approximate the optimal functions $D^\star_\theta$. We believe that this point is critical for the theoretical analysis of GANs, and that it should be further developed in the context of neural networks, with a potentially large number of hidden layers.

3. Theorem 2.4.2 (convergence of the estimated parameter) and Theorem 2.4.3 (asymptotic normality) hold under the assumption that the model is identifiable (uniqueness of $\bar{\theta}$ and $\bar{\alpha}$). This identifiability assumption is hardly satisfied in the high-dimensional context of (deep) neural networks, where the function to be optimized displays a very wild landscape, without immediate convexity or concavity. Thus, to take one more step towards a more realistic model, it would be interesting to shift the parametric point of view and move towards results concerning the convergence of distributions not parameters.

Appendix 2.A Technical results

2.A.1 Proof of Theorem 2.3.1

Let $\varepsilon \in (0, 1/(2M))$, $m \in (0, 1/2)$, and $D \in \mathscr{D}$ be such that $m \leq D \leq 1-m$ and $\|D - D^\star_{\bar\theta}\|_2 \leq \varepsilon$. Observe that

$$\begin{aligned} L(\bar\theta, D) &= \int \ln(D) p_\star \mathrm{d}\mu + \int \ln(1-D) p_{\bar\theta} \mathrm{d}\mu \\ &= \int \ln\Big(\frac{D}{D^\star_{\bar\theta}}\Big) p_\star \mathrm{d}\mu + \int \ln\Big(\frac{1-D}{1-D^\star_{\bar\theta}}\Big) p_{\bar\theta} \mathrm{d}\mu + 2D_{\mathrm{JS}}(p_\star, p_{\bar\theta}) - \ln 4. \end{aligned} \tag{2.A.1}$$

We first derive a lower bound on the quantity

$$\begin{aligned} I &\stackrel{\text{def}}{=} \int \ln\Big(\frac{D}{D^\star_{\bar\theta}}\Big) p_\star \mathrm{d}\mu + \int \ln\Big(\frac{1-D}{1-D^\star_{\bar\theta}}\Big) p_{\bar\theta} \mathrm{d}\mu \\ &= \int \ln\Big(\frac{D(p_\star + p_{\bar\theta})}{p_\star}\Big) p_\star \mathrm{d}\mu + \int \ln\Big(\frac{(1-D)(p_\star + p_{\bar\theta})}{p_{\bar\theta}}\Big) p_{\bar\theta} \mathrm{d}\mu. \end{aligned}$$

Let $\mathrm{d}p_\star = p_\star \mathrm{d}\mu$, $\mathrm{d}P_{\bar\theta} = p_{\bar\theta}\mathrm{d}\mu$,

$$\mathrm{d}\kappa = \frac{D(p_\star + p_{\bar\theta})}{\int D(p_\star + p_{\bar\theta})\mathrm{d}\mu}\mathrm{d}\mu, \quad \text{and} \quad \mathrm{d}\kappa' = \frac{(1-D)(p_\star + p_{\bar\theta})}{\int (1-D)(p_\star + p_{\bar\theta})\mathrm{d}\mu}\mathrm{d}\mu.$$

Observe, since $m \leq D \leq 1-m$, that $p_\star \ll \kappa$ and $P_{\bar\theta} \ll \kappa'$. With this notation, we have

$$I = -D_{\mathrm{KL}}(p_\star \parallel \kappa) - D_{\mathrm{KL}}(P_{\bar\theta} \parallel \kappa') + \ln\Big[\int D(p_\star + p_{\bar\theta})\mathrm{d}\mu \Big(2 - \int D(p_\star + p_{\bar\theta})\mathrm{d}\mu\Big)\Big]. \tag{2.A.2}$$

Since

$$\int D(p_\star + p_{\bar\theta})\mathrm{d}\mu = \int (D - D^\star_{\bar\theta})(p_\star + p_{\bar\theta})\mathrm{d}\mu + 1,$$

the Cauchy-Schwartz inequality leads to

$$\begin{aligned} \Big|\int D(p_\star + p_{\bar\theta})\mathrm{d}\mu - 1\Big| &\leq \|D - D^\star_{\bar\theta}\|_2 \|p_\star + p_{\bar\theta}\|_2 \\ &\leq 2M\varepsilon, \end{aligned} \tag{2.A.3}$$

because both $p_\star$ and $p_{\bar\theta}$ are bounded by M. Thus,

$$\ln\Big[\int D(p_\star+p_{\bar\theta})\mathrm{d}\mu\big(2-\int D(p_\star+p_{\bar\theta})\mathrm{d}\mu\big)\Big] \geq \ln(1-4M^2\varepsilon^2) \geq -\frac{4M^2\varepsilon^2}{1-4M^2\varepsilon^2}, \tag{2.A.4}$$

using the inequality $\ln(1-x)\geq -x/(1-x)$ for $x\in[0,1)$. Moreover, recalling that the Kullback-Leibler divergence is smaller than the chi-square divergence, and letting $\overline{F}=F/(\int F\mathrm{d}\mu)$ for $F\in L^1(\mu)$, we have

$$D_{\mathrm{KL}}(p_\star \parallel \kappa) \leq \int \Big(\frac{p_\star}{\overline{D(p_\star+p_{\bar\theta})}}-1\Big)^2\overline{D(p_\star+p_{\bar\theta})}\mathrm{d}\mu.$$

Hence, letting $J\stackrel{\mathrm{def}}{=}\int D(p_\star+p_{\bar\theta})\mathrm{d}\mu$, we see that

$$\begin{aligned}
&D_{\mathrm{KL}}(p_\star \parallel \kappa)\\
&\quad\leq \frac{1}{J}\int\Big(p_\star\int D(p_\star+p_{\bar\theta})\mathrm{d}\mu - D(p_\star+p_{\bar\theta})\Big)^2\frac{1}{D(p_\star+p_{\bar\theta})}\mathrm{d}\mu\\
&\quad= \frac{1}{J}\int\Big(p_\star\int (D-D^\star_{\bar\theta})(p_\star+p_{\bar\theta})\mathrm{d}\mu + (D^\star_{\bar\theta}-D)(p_\star+p_{\bar\theta})\Big)^2\frac{1}{D(p_\star+p_{\bar\theta})}\mathrm{d}\mu.
\end{aligned}$$

Since $\varepsilon<1/(2M)$, inequality (2.A.3) gives $1/J\leq c_1$ for some constant $c_1>0$. By Cauchy-Schwarz and $(a+b)^2\leq 2(a^2+b^2)$, we obtain

$$\begin{aligned}
&D_{\mathrm{KL}}(p_\star \parallel \kappa)\\
&\quad\leq 2c_1\Big(\int\Big(\int (D-D^\star_{\bar\theta})(p_\star+p_{\bar\theta})\mathrm{d}\mu\Big)^2\frac{(p_\star)^2}{D(p_\star+p_{\bar\theta})}\mathrm{d}\mu + \int (D^\star_{\bar\theta}-D)^2\frac{p_\star+p_{\bar\theta}}{D}\mathrm{d}\mu\Big)\\
&\quad\leq 2c_1\Big(\|D-D^\star_{\bar\theta}\|_2^2\|p_\star+p_{\bar\theta}\|_2^2\int\frac{(p_\star)^2}{D(p_\star+p_{\bar\theta})}\mathrm{d}\mu + \int (D^\star_{\bar\theta}-D)^2\frac{p_\star+p_{\bar\theta}}{D}\mathrm{d}\mu\Big).
\end{aligned}$$

Therefore, since $p_\star\leq M$, $p_{\bar\theta}\leq M$, and $D\geq m$,

$$D_{\mathrm{KL}}(p_\star \parallel \kappa)\leq 2c_1\Big(\frac{4M^2}{m}+\frac{2M}{m}\Big)\varepsilon^2.$$

One proves with similar arguments that

$$D_{\mathrm{KL}}(P_{\bar\theta} \parallel \kappa')\leq 2c_1\Big(\frac{4M^2}{m}+\frac{2M}{m}\Big)\varepsilon^2.$$

Combining these two inequalities with (2.A.2) and (2.A.4), we see that $I \geq -c_2\varepsilon^2$ for some constant $c_2 > 0$ that depends only upon M and m. Getting back to identity (2.A.1), we conclude that

$$2D_{\mathrm{JS}}(p_\star, p_{\bar{\theta}}) \leq c_2\varepsilon^2 + L(\bar{\theta}, D) + \ln 4.$$

But

$$\begin{aligned} L(\bar{\theta}, D) \leq \sup_{D\in\mathscr{D}} L(\bar{\theta}, D) &\leq \sup_{D\in\mathscr{D}} L(\theta^\star, D) \\ &\quad \text{(by definition of } \bar{\theta}) \\ &\leq \sup_{D\in\mathscr{D}_\infty} L(\theta^\star, D) \\ &= L(\theta^\star, D^\star_{\theta^\star}) = 2D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) - \ln 4. \end{aligned}$$

Thus,

$$2D_{\mathrm{JS}}(p_\star, p_{\bar{\theta}}) \leq c_2\varepsilon^2 + 2D_{\mathrm{JS}}(p_\star, p_{\theta^\star}).$$

This shows the right-hand side of inequality (2.3.1). To prove the left-hand side, just note that by inequality (2.2.2),

$$D_{\mathrm{JS}}(p_\star, p_{\theta^\star}) \leq D_{\mathrm{JS}}(p_\star, p_{\bar{\theta}}).$$

2.A.2 Proof of Lemma 2.4.1

To simplify the notation, we set

$$\Delta = \frac{\partial^{a+b+c+d}}{\partial\theta_i^a \partial\theta_j^b \partial\alpha_\ell^c \partial\alpha_m^d}.$$

Using McDiarmid's inequality (McDiarmid, 1989), we see that there exists a constant $c > 0$ such that, for all $\varepsilon > 0$,

$$\mathbb{P}\Big(\Big|\sup_{\theta\in\Theta,\alpha\in\Lambda} |\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha)| - \mathbb{E}\sup_{\theta\in\Theta,\alpha\in\Lambda} |\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha)|\Big| \geq \varepsilon\Big) \leq 2e^{-cn\varepsilon^2}.$$

Therefore, by the Borel-Cantelli lemma,

$$\sup_{\theta\in\Theta,\alpha\in\Lambda} |\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha)| - \mathbb{E}\sup_{\theta\in\Theta,\alpha\in\Lambda} |\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha)| \to 0 \quad \text{almost surely.} \tag{2.A.5}$$

It is also easy to verify that under Assumptions (H'_{reg}), the process $(\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha))_{\theta\in\Theta,\alpha\in\Lambda}$ is subgaussian. Thus, as in the proof of Theorem 2.4.1, we obtain via

Dudley's inequality that

$$\mathbb{E} \sup_{\theta\in\Theta,\alpha\in\Lambda} |\Delta\hat{L}(\theta,\alpha) - \Delta L(\theta,\alpha)| = \mathrm{O}\Big(\frac{1}{\sqrt{n}}\Big), \tag{2.A.6}$$

since $\mathbb{E}\Delta\hat{L}(\theta,\alpha) = \Delta L(\theta,\alpha)$. The result follows by combining (2.A.5) and (2.A.6).

2.A.3 Some technical lemmas

Lemma 2.A.1. *Under Assumptions* (H'_{reg}) *and* (H_1)*, there exists an open set* $V \subset \Theta^\circ$ *containing* $\bar{\theta}$ *such that, for all* $\theta \in V$*,* $\arg\max_{\alpha\in\Lambda} L(\theta,\alpha) \cap \Lambda^\circ \neq \emptyset$*.*

Proof. Assume that the statement is not true. Then there exists a sequence $(\theta_k)_k \subset \Theta$ such that $\theta_k \to \bar{\theta}$ and, for all k, $\alpha_k \in \partial\Lambda$, where $\alpha_k \in \arg\max_{\alpha\in\Lambda} L(\theta_k,\alpha)$. Thus, since Λ is compact, even if this means extracting a subsequence, one has $\alpha_k \to z \in \partial\Lambda$ as $k \to \infty$. By the continuity of L, $L(\bar{\theta},\alpha_k) \to L(\bar{\theta},z)$. But

$$\begin{aligned}
|L(\bar{\theta},\alpha_k) - L(\bar{\theta},\bar{\alpha})| &\leq |L(\bar{\theta},\alpha_k) - L(\theta_k,\alpha_k)| + |L(\theta_k,\alpha_k) - L(\bar{\theta},\bar{\alpha})| \\
&\leq \sup_{\alpha\in\Lambda} |L(\bar{\theta},\alpha) - L(\theta_k,\alpha)| + |\sup_{\alpha\in\Lambda} L(\theta_k,\alpha) - \sup_{\alpha\in\Lambda} L(\bar{\theta},\alpha)| \\
&\leq 2\sup_{\alpha\in\Lambda} |L(\bar{\theta},\alpha) - L(\theta_k,\alpha)|,
\end{aligned}$$

which tends to zero as $k \to \infty$ by (H'_D) and (H'_p). Therefore, $L(\bar{\theta},z) = L(\bar{\theta},\bar{\alpha})$ and, in turn, $z = \bar{\alpha}$ by (H_1). Since $z \in \partial\Delta$ and $\bar{\alpha} \in \Delta^\circ$, this is a contradiction. □

Lemma 2.A.2. *Under Assumptions* (H'_{reg})*,* (H_1)*, and* (H_{loc})*, one has* $\hat{I}_3 \to HV(\bar{\theta})$ *almost surely.*

Proof. We have

$$\hat{I}_3 = \int_0^1 H\hat{V}(\hat{\theta} + t(\hat{\theta} - \bar{\theta}))\mathrm{d}t = \int_0^1 \big(H_1\hat{L}(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t)) + J(\nabla_1\hat{L}(\hat{\theta}_t,\cdot))_{\hat{\alpha}(\hat{\theta}_t)} J(\hat{\alpha})_{\hat{\theta}_t}\big)\mathrm{d}t,$$

where we set $\hat{\theta}_t = \hat{\theta} + t(\hat{\theta} - \bar{\theta})$. Note that $\hat{\theta}_t \in U$ for all $t \in [0,1]$. By Lemma 2.4.1,

$$\begin{aligned}
&\sup_{t\in[0,1]} \|H_1\hat{L}(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t)) - H_1 L(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t))\| \\
&\quad\leq \sup_{\theta\in\Theta,\alpha\in\Lambda} \|H_1\hat{L}(\theta,\alpha) - H_1 L(\theta,\alpha)\| \to 0 \quad \text{almost surely.}
\end{aligned}$$

Also, by Theorem 2.4.2, for all $t \in [0,1]$, $\hat{\theta}_t \to \bar{\theta}$ almost surely. Besides,

$$\begin{aligned}|L(\bar{\theta},\hat{\alpha}(\hat{\theta}_t)) - L(\bar{\theta},\alpha(\bar{\theta}))| &\leq |L(\bar{\theta},\hat{\alpha}(\hat{\theta}_t)) - L(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t))| + |L(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t)) - L(\bar{\theta},\alpha(\bar{\theta}))| \\ &\leq \sup_{\alpha\in\Lambda} |L(\bar{\theta},\alpha) - L(\hat{\theta}_t,\alpha)| + 2 \sup_{\theta\in\Theta,\alpha\in\Lambda} |\hat{L}(\theta,\alpha) - L(\theta,\alpha)|.\end{aligned}$$

Thus, via (H'_{reg}), (H_1), and Lemma 2.4.1, we conclude that almost surely, for all $t \in [0,1]$, $\hat{\alpha}(\hat{\theta}_t) \to \alpha(\bar{\theta}) = \bar{\alpha}$. Accordingly, almost surely, for all $t \in [0,1]$, $H_1 L(\hat{\theta}_t, \hat{\alpha}(\hat{\theta}_t)) \to H_1 L(\bar{\theta},\bar{\alpha})$. Since $H_1 L(\theta,\alpha)$ is bounded under (H'_D) and (H'_p), the Lebesgue dominated convergence theorem leads to

$$\int_0^1 H_1 \hat{L}(\hat{\theta}_t, \hat{\alpha}(\hat{\theta}_t))\mathrm{d}t \to H_1 L(\bar{\theta},\bar{\alpha}) \quad \text{almost surely.} \tag{2.A.7}$$

Furthermore,

$$J(\hat{\alpha})_\theta = -H_2\hat{L}(\theta,\hat{\alpha}(\theta))^{-1} J\big(\nabla_2 \hat{L}(\cdot,\hat{\alpha}(\theta))\big)_\theta, \quad \forall (\theta,\alpha) \in U \times \alpha(U),$$

where U is the open set defined in the proof of Theorem 2.4.3. By the cofactor method, $H_2\hat{L}(\theta,\alpha)^{-1}$ takes the form

$$H_2\hat{L}(\theta,\alpha)^{-1} = \frac{\hat{c}(\theta,\alpha)}{\det(H_2\hat{L}(\theta,\alpha))},$$

where $\hat{c}(\theta,\alpha)$ is the matrix of cofactors associated with $H_2\hat{L}(\theta,\alpha)$. Thus, each component of $-H_2\hat{L}(\theta,\alpha)^{-1} J(\nabla_2\hat{L}(\cdot,\alpha))_\theta$ is a quotient of a multilinear form of the partial derivatives of $\hat{L}$ evaluated at (θ,α) divided by $\det(H_2\hat{L}(\theta,\alpha))$, which is itself a multilinear form in the $\frac{\partial^2 \hat{L}}{\partial\alpha_i\partial\alpha_j}(\theta,\alpha)$. Hence, by Lemma 2.4.1, we have

$$\sup_{\theta\in U,\alpha\in\alpha(U)} \|H_2\hat{L}(\theta,\alpha)^{-1} J(\nabla_2\hat{L}(\cdot,\alpha))_\theta - H_2 L(\theta,\alpha)^{-1} J(\nabla_2 L(\cdot,\alpha))_\theta\| \to 0 \text{ almost surely.}$$

So, for all n large enough,

$$\begin{aligned}&\sup_{t\in[0,1]} \|J(\hat{\alpha})_{\hat{\theta}_t} + H_2 L(\hat{\theta}_t,\hat{\alpha}(\hat{\theta}_t))^{-1} J\big(\nabla_2 L(\cdot,\hat{\alpha}(\hat{\theta}_t))\big)_{\hat{\theta}_t}\| \\ &\quad\leq \sup_{\theta\in U,\alpha\in\alpha(U)} \|H_2\hat{L}(\theta,\alpha)^{-1} J(\nabla_2\hat{L}(\cdot,\alpha))_\theta - H_2 L(\theta,\alpha)^{-1} J(\nabla_2 L(\cdot,\alpha))_\theta\| \\ &\quad\to 0 \quad \text{almost surely.}\end{aligned}$$

We know that almost surely, for all $t \in [0,1]$, $\hat{\alpha}(\hat{\theta}_t) \to \bar{\alpha}$. Thus, since the function $U \times \alpha(U) \ni (\theta, \alpha) \mapsto H_2 L(\theta,\alpha)^{-1} J(\nabla_2 L(\cdot,\alpha))_\theta$ is continuous, we have almost surely, for all $t \in [0,1]$,

$$H_2\hat{L}(\hat{\theta}_t, \hat{\alpha}(\hat{\theta}_t))^{-1} J\big(\nabla_2 \hat{L}(\cdot, \hat{\alpha}(\hat{\theta}_t))\big)_{\hat{\theta}_t} \to H_2 L(\bar{\theta},\bar{\alpha})^{-1} J(\nabla_2 L(\cdot,\bar{\alpha}))_{\bar{\theta}}.$$

Therefore, almost surely, for all $t \in [0,1]$, $J(\hat{\alpha})_{\hat{\theta}_t} \to J(\alpha)_{\bar{\theta}}$. Similarly, almost surely, for all $t \in [0,1]$, $J(\nabla_1 \hat{L}(\hat{\theta}_t,\cdot))_{\hat{\alpha}(\hat{\theta}_t)} \to J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}}$. All involved quantities are uniformly bounded in t, and so, by the Lebesgue dominated convergence theorem, we conclude that

$$\int_0^1 J(\nabla_1 \hat{L}(\hat{\theta}_t,\cdot))_{\hat{\alpha}(\hat{\theta}_t)} J(\hat{\alpha})_{\hat{\theta}_t} \mathrm{d}t \to J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}} J(\alpha)_{\bar{\theta}} \quad \text{almost surely.} \tag{2.A.8}$$

Consequently, by combining (2.A.7) and (2.A.8),

$$\hat{I}_3 \to H_1 L(\bar{\theta},\bar{\alpha}) + J(\nabla_1 L(\bar{\theta},\cdot))_{\bar{\alpha}} J(\alpha)_{\bar{\theta}} = HV(\bar{\theta}) \quad \text{almost surely,}$$

as desired. □

Acknowledgments

We thank Flavian Vasile (Criteo) and Antoine Picard-Weibel (ENS Ulm) for stimulating discussions and insightful suggestions. We also thank the Associate Editor and two anonymous referees for their careful reading of the paper and constructive comments, which led to a substantial improvement of the document.

Chapter 3

Some theoretical properties of Wasserstein GANs

Abstract

Generative Adversarial Networks (GANs) have been successful in producing outstanding results in areas as diverse as image, video, and text generation. Building on these successes, a large number of empirical studies have validated the benefits of the cousin approach called Wasserstein GANs (WGANs), which brings stabilization in the training process. In the present paper, we add a new stone to the edifice by proposing some theoretical advances in the properties of WGANs. First, we properly define the architecture of WGANs in the context of integral probability metrics parameterized by neural networks and highlight some of their basic mathematical features. We stress in particular interesting optimization properties arising from the use of a parametric 1-Lipschitz discriminator. Then, in a statistically-driven approach, we study the convergence of empirical WGANs as the sample size tends to infinity, and clarify the adversarial effects of the generator and the discriminator by underlining some trade-off properties. These features are finally illustrated with experiments using both synthetic and real-world datasets.

Contents

3.1 Introduction **62**

3.2 Wasserstein GANs **64**

3.3 Optimization properties **71**

3.4 Asymptotic properties **80**

3.5 Understanding the performance of WGANs **85**

Appendix 3.A Technical results **91**

3.1 Introduction

Generative Adversarial Networks (GANs) is a generative framework proposed by Goodfellow et al. (2014), in which two models (a generator and a discriminator) act as adversaries in a zero-sum game. Leveraging the recent advances in deep learning, and specifically convolutional neural networks (LeCun et al., 1998), a large number of empirical studies have shown the impressive possibilities of GANs in the field of image generation (Radford et al., 2015; Ledig et al., 2017; Karras et al., 2018; Brock et al., 2019). Lately, Karras et al. (2019) proposed an architecture able to generate hyper-realistic fake human faces that cannot be differentiated from real ones (see the website thispersondoesnotexist.com). The recent surge of interest in the domain also led to breakthroughs in video (Acharya et al., 2018), music (Mogren, 2016), and text generation (Yu et al., 2017; Fedus et al., 2018), among many other potential applications.

The aim of GANs is to generate data that look "similar" to samples collected from some unknown probability measure $\mu_\star$, defined on a Borel subset E of $\mathbb{R}^D$. In the targeted applications of GANs, E is typically a submanifold (possibly hard to describe) of a high-dimensional $\mathbb{R}^D$, which therefore prohibits the use of classical density estimation techniques. GANs approach the problem by making two models compete: the generator, which tries to imitate $\mu_\star$ using the collected data, vs. the discriminator, which learns to distinguish the outputs of the generator from the samples, thereby forcing the generator to improve its strategy.

Formally, the generator has the form of a parameterized class of Borel functions from $\mathbb{R}^d$ to E, say $\mathscr{G} = \{G_\theta : \theta \in \Theta\}$, where $\Theta \subseteq \mathbb{R}^P$ is the set of parameters describing the model. Each function G_θ takes as input a d-dimensional random variable Z—it is typically uniform or Gaussian, with d usually small—and outputs the "fake" observation $G_\theta(Z)$ with distribution μ_θ. Thus, the collection of probability measures $\mathscr{P} = \{\mu_\theta : \theta \in \Theta\}$ is the natural class of distributions associated with the generator, and the objective of GANs is to find inside this class the distribution that generates the most realistic samples, closest to the ones collected from the unknown $\mu_\star$. On the other hand, the discriminator is described by a family of Borel functions from E to $[0,1]$, say $\mathscr{D} = \{D_\alpha : \alpha \in \Lambda\}$, $\Lambda \subseteq \mathbb{R}^Q$, where each D_α must be thought of as the probability that an observation comes from $\mu_\star$ (the higher $D(x)$, the higher the probability that x is drawn from $\mu_\star$).

In the original formulation of Goodfellow et al. (2014), GANs make $\mathscr{G}$ and $\mathscr{D}$ fight each other through the following objective:

$$\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} \Big[\mathbb{E}\log(D_\alpha(X)) + \mathbb{E}\log(1 - D_\alpha(G_\theta(Z)))\Big], \tag{3.1.1}$$

where X is a random variable with distribution $\mu^\star$ and the symbol $\mathbb{E}$ denotes expectation. Since one does not have access to the true distribution, $\mu_\star$ is replaced in practice with the empirical measure μ_n based on independent and identically distributed (i.i.d.) samples $X_1, \ldots, X_n$ distributed as X, and the practical objective becomes

$$\inf_{\theta \in \Theta} \sup_{\alpha \in \Lambda} \left[\frac{1}{n} \sum_{i=1}^{n} \log(D_\alpha(X_i)) + \mathbb{E} \log(1 - D_\alpha(G_\theta(Z))) \right]. \tag{3.1.2}$$

In the literature on GANs, both $\mathscr{G}$ and $\mathscr{D}$ take the form of neural networks (either feed-forward or convolutional, when dealing with image-related applications). This is also the case in the present paper, in which the generator and the discriminator will be parameterized by feed-forward neural networks with, respectively, rectifier (Glorot et al., 2011) and GroupSort (Chernodub and Nowicki, 2016) activation functions. We also note that from an optimization standpoint, the minimax optimum in (3.1.2) is found by using stochastic gradient descent alternatively on the generator's and the discriminator's parameters.

In the initial version (3.1.1), GANs were shown to reduce, under appropriate conditions, the Jensen-Shanon divergence between the true distribution and the class of parameterized distributions (Goodfellow et al., 2014). This characteristic was further explored by Biau et al. (2020), who stressed some theoretical guarantees regarding the approximation and statistical properties of problems (3.1.1) and (3.1.2). However, many empirical studies (e.g., Metz et al., 2016; Salimans et al., 2016) have described cases where the optimal generative distribution computed by solving (3.1.2) collapses to a few modes of the distribution $\mu_\star$. This phenomenon is known under the term of mode collapse and has been theoretically explained by Arjovsky et al. (2017). As a striking result, in cases where both $\mu_\star$ and μ_θ lie on disjoint supports, these authors proved the existence of a perfect discriminator with null gradient on both supports, which consequently does not convey meaningful information to the generator.

To cancel this drawback and stabilize training, Arjovsky et al. (2017) proposed a modification of criterion (3.1.1), with a framework called Wasserstein GANs (WGANs). In a nutshell, the objective of WGANs is to find, inside the class of parameterized distributions $\mathscr{P}$, the one that is the closest to the true $\mu_\star$ with respect to the Wasserstein distance (Villani, 2008). In its dual form, the Wasserstein distance can be considered as an integral probability metric (IPM, Müller, 1997) defined on the set of 1-Lipschitz functions. Therefore, the proposal of Arjovsky et al. (2017) is to replace the 1-Lipschitz functions with a discriminator parameterized by neural networks. To practically enforce this discriminator to be a subset of 1-Lipschitz functions, the authors use a weight clipping technique on the set of parameters. A decisive step has been taken by Gulrajani et al. (2017), who stressed the empirical advantage of the WGANs architecture by replacing the weight clipping with a gradient penalty. Since then, WGANs have been largely

recognized and studied by the Machine Learning community (e.g., Roth et al., 2017; Petzka et al., 2018; Wei et al., 2018; Karras et al., 2019).

A natural question regards the theoretical ability of WGANs to learn $\mu^\star$, considering that one only has access to the parametric models of generative distributions and discriminative functions. Previous works in this direction are those of Liang (2018) and Zhang et al. (2018), who explore generalization properties of WGANs. In the present paper, we make one step further in the analysis of mathematical forces driving WGANs and contribute to the literature in the following ways:

- (i) We properly define the architecture of WGANs parameterized by neural networks. Then, we highlight some properties of the IPM induced by the discriminator, and finally stress some basic mathematical features of the WGANs framework (Section 3.2).
- (ii) We emphasize the impact of operating with a parametric discriminator contained in the set of 1-Lipschitz functions. We introduce in particular the notion of monotonous equivalence and discuss its meaning in the mechanism of WGANs. We also highlight the essential role played by piecewise linear functions (Section 3.3).
- (iii) In a statistically-driven approach, we derive convergence rates for the IPM induced by the discriminator, between the target distribution $\mu^\star$ and the distribution output by the WGANs based on i.i.d. samples (Section 3.4).
- (iv) Building upon the above, we clarify the adversarial effects of the generator and the discriminator by underlining some trade-off properties. These features are illustrated with experiments using both synthetic and real-world datasets (Section 3.5).

For the sake of clarity, proofs of the most technical results are gathered in the Appendix.

3.2 Wasserstein GANs

The present section is devoted to the presentation of the WGANs framework. After having given a first set of definitions and results, we stress the essential role played by IPMs and study some optimality properties of WGANs.

3.2.1 Notation and definitions

Throughout the paper, E is a Borel subset of $\mathbb{R}^D$, equipped with the Euclidean norm $\|\cdot\|$, on which $\mu_\star$ (the target probability measure) and the μ_θ's (the candidate probability measures)

are defined. Depending on the practical context, E can be equal to $\mathbb{R}^D$, but it can also be a submanifold of it. We emphasize that there is no compactness assumption on E.

For $K \subseteq E$, we let $C(K)$ (respectively, $C_b(K)$) be the set of continuous (respectively, continuous bounded) functions from K to $\mathbb{R}$. We denote by Lip_1 the set of 1-Lipschitz real-valued functions on E, i.e.,

$$\mathrm{Lip}_1 = \big\{f : E \to \mathbb{R} : |f(x) - f(y)| \leqslant \|x-y\|,\ (x,y) \in E^2\big\}.$$

The notation $P(E)$ stands for the collection of Borel probability measures on E, and $P_1(E)$ for the subset of probability measures with finite first moment, i.e.,

$$P_1(E) = \Big\{\mu \in P(E) : \int_E \|x_0 - x\|\mu(\mathrm{dx}) < \infty\Big\},$$

where $x_0 \in E$ is arbitrary (this set does not depend on the choice of the point x_0). Until the end, it is assumed that $\mu_\star \in P_1(E)$. It is also assumed throughout that the random variable $Z \in \mathbb{R}^d$ is a sub-Gaussian random vector (Jin et al., 2019), i.e., Z is integrable and there exists $\gamma > 0$ such that

$$\forall v \in \mathbb{R}^d,\ \mathbb{E}e^{v\cdot(Z-\mathbb{E}Z)} \leqslant e^{\frac{\gamma^2\|v\|^2}{2}},$$

where $\cdot$ denotes the dot product in $\mathbb{R}^d$ and $\|\cdot\|$ the Euclidean norm. The sub-Gaussian property is a constraint on the tail of the probability distribution. As an example, Gaussian random variables on the real line are sub-Gaussian and so are bounded random vectors. We note that Z has finite moments of all nonnegative orders (Jin et al., 2019, Lemma 2). Assuming that Z is sub-Gaussian is a mild requirement since, in practice, its distribution is most of the time uniform or Gaussian.

As highlighted earlier, both the generator and the discriminator are assumed to be parameterized by feed-forward neural networks, that is,

$$\mathscr{G} = \{G_\theta : \theta \in \Theta\} \qquad \text{and} \qquad \mathscr{D} = \{D_\alpha : \alpha \in \Lambda\}$$

with $\Theta \subseteq \mathbb{R}^P$, $\Lambda \subseteq \mathbb{R}^Q$, and, for all $z \in \mathbb{R}^d$,

$$G_\theta(z) = \underset{D\times u_{p-1}}{U_p}\ \sigma\Big(\underset{u_{p-1}\times u_{p-2}}{U_{p-1}}\ \cdots\sigma\Big(\underset{u_2\times u_1}{U_2}\ \sigma\Big(\underset{u_1\times d}{U_1}\ z + \underset{u_1\times 1}{b_1}\Big) + \underset{u_2\times 1}{b_2}\Big)\cdots + \underset{u_{p-1}\times 1}{b_{p-1}}\Big) + \underset{D\times 1}{b_p}, \quad (3.2.1)$$

for all $x \in E$,

$$D_\alpha(x) = \underset{1\times v_{q-1}}{V_q}\ \tilde{\sigma}\Big(\underset{v_{q-1}\times v_{q-2}}{V_{q-1}}\ \cdots\tilde{\sigma}\Big(\underset{v_2\times v_1}{V_2}\ \tilde{\sigma}\Big(\underset{v_1\times D}{V_1}\ x + \underset{v_1\times 1}{c_1}\Big) + \underset{v_2\times 1}{c_2}\Big) + \cdots + \underset{v_{q-1}\times 1}{c_{q-1}}\Big) + \underset{1\times 1}{c_q}, \quad (3.2.2)$$

where $p, q \geqslant 2$ and the characters below the matrices indicate their dimensions (lines $\times$ columns). Some comments on the notation are in order. Networks in $\mathscr{G}$ and $\mathscr{D}$ have, respectively, $(p-1)$ and $(q-1)$ hidden layers. Hidden layers from depth 1 to $(p-1)$ (for the generator) and from depth 1 to $(q-1)$ (for the discriminator) are assumed to be of respective even widths u_i, $i = 1, \ldots, p-1$, and v_i, $i = 1, \ldots, q-1$. The matrices U_i (respectively, V_i) are the matrices of weights between layer i and layer $(i+1)$ of the generator (respectively, the discriminator), and the b_i's (respectively, the c_i's) are the corresponding offset vectors (in column format). We let $\sigma(x) = \max(x, 0)$ be the rectifier activation function (applied componentwise) and

$$\tilde{\sigma}(x_1, x_2, \ldots, x_{2n-1}, x_{2n}) = (\max(x_1, x_2), \min(x_1, x_2), \ldots, \max(x_{2n-1}, x_{2n}), \min(x_{2n-1}, x_{2n}))$$

be the GroupSort activation function with a grouping size equal to 2 (applied on pairs of components, which makes sense in (3.2.2) since the widths of the hidden layers are assumed to be even). GroupSort has been introduced in Chernodub and Nowicki (2016) as a 1-Lipschitz activation function that preserves the gradient norm of the input. This activation can recover the rectifier, in the sense that $\tilde{\sigma}(x, 0) = (\sigma(x), -\sigma(-x))$, but the converse is not true. The presence of GroupSort is critical to guarantee approximation properties of Lipschitz neural networks (Anil et al., 2019), as we will see later.

Therefore, denoting by $\mathscr{M}_{(j,k)}$ the space of matrices with j rows and k columns, we have $U_1 \in \mathscr{M}_{(u_1,d)}$, $V_1 \in \mathscr{M}_{(v_1,D)}$, $b_1 \in \mathscr{M}_{(u_1,1)}$, $c_1 \in \mathscr{M}_{(v_1,1)}$, $U_p \in \mathscr{M}_{(D,u_{p-1})}$, $V_q \in \mathscr{M}_{(1,v_{q-1})}$, $b_p \in \mathscr{M}_{(D,1)}$, $c_q \in \mathscr{M}_{(1,1)}$. All the other matrices U_i, $i = 2, \ldots, p-1$, and V_i, $i = 2, \ldots, q-1$, belong to $\mathscr{M}_{(u_i,u_{i-1})}$ and $\mathscr{M}_{(v_i,v_{i-1})}$, and vectors b_i, $i = 2, \ldots, p-1$, and c_i, $i = 2, \ldots, q-1$, belong to $\mathscr{M}_{(u_i,1)}$ and $\mathscr{M}_{(v_i,1)}$. So, altogether, the vectors $\theta = (U_1, \ldots, U_p, b_1, \ldots, b_p)$ (respectively, the vectors $\alpha = (V_1, \ldots, V_q, c_1, \ldots, c_q)$) represent the parameter space Θ of the generator $\mathscr{G}$ (respectively, the parameter space Λ of the discriminator $\mathscr{D}$). We stress the fact that the outputs of networks in $\mathscr{D}$ are not restricted to $[0, 1]$ anymore, as is the case for the original GANs of Goodfellow et al. (2014). We also recall the notation $\mathscr{P} = \{\mu_\theta : \theta \in \Theta\}$, where, for each θ, μ_θ is the probability distribution of $G_\theta(Z)$. Since Z has finite first moment and each G_θ is piecewise linear, it is easy to see that $\mathscr{P} \subset P_1(E)$.

Throughout the manuscript, the notation $\|\cdot\|$ (respectively, $\|\cdot\|_\infty$) means the Euclidean (respectively, the supremum) norm on $\mathbb{R}^k$, with no reference to k as the context is clear. For $W = (w_{i,j})$ a matrix in $\mathscr{M}_{(k_1,k_2)}$, we let $\|W\|_2 = \sup_{\|x\|=1} \|Wx\|$ be the 2-norm of W. Similarly, the ∞-norm of W is $\|W\|_\infty = \sup_{\|x\|_\infty=1} \|Wx\|_\infty = \max_{i=1,\ldots,k_1} \sum_{j=1}^{k_2} |w_{i,j}|$. We will also use the $(2,\infty)$-norm of W, i.e., $\|W\|_{2,\infty} = \sup_{\|x\|=1} \|Wx\|_\infty$. We shall constantly need the following assumption:

Assumption 1. *For all* $\theta = (U_1, \ldots, U_p, b_1, \ldots, b_p) \in \Theta$,

$$\max(\|U_i\|_2, \|b_i\|_2 : i = 1, \ldots, p) \leqslant K_1,$$

where $K_1 > 0$ *is a constant. Besides, for all* $\alpha = (V_1, \ldots, V_q, c_1, \ldots, c_q) \in \Lambda$,

$$\|V_1\|_{2,\infty} \leqslant 1, \ \max(\|V_2\|_\infty, \ldots, \|V_q\|_\infty) \leqslant 1, \text{ and } \max(\|c_i\|_\infty : i = 1, \ldots, q) \leqslant K_2,$$

where $K_2 \geqslant 0$ *is a constant.*

This compactness requirement is classical when parameterizing WGANs (e.g., Arjovsky et al., 2017; Zhang et al., 2018; Anil et al., 2019). In practice, one can satisfy Assumption 1 by clipping the parameters of neural networks as proposed by Arjovsky et al. (2017). An alternative approach to enforce $\mathscr{D} \subseteq \mathrm{Lip}_1$ consists in penalizing the gradient of the discriminative functions, as proposed by Gulrajani et al. (2017), Kodali et al. (2017), Wei et al. (2018), and Zhou et al. (2019). This solution was empirically found to be more stable. The usefulness of Assumption 1 is captured by the following lem.

Lemma 3.2.1. *Assume that Assumption 1 is satisfied. Then, for each* $\theta \in \Theta$, *the function* G_θ *is* K_1^p*-Lipschitz on* $\mathbb{R}^d$. *In addition,* $\mathscr{D} \subseteq \mathrm{Lip}_1$.

Recall (e.g., Dudley, 2004) that a sequence of probability measures (μ_k) on E is said to converge weakly to a probability measure μ on E if, for all $\varphi \in C_b(E)$,

$$\int_E \varphi \, \mathrm{d}\mu_k \underset{k\to\infty}{\to} \int_E \varphi \, \mathrm{d}\mu.$$

In addition, the sequence of probability measures (μ_k) in $P_1(E)$ is said to converge weakly in $P_1(E)$ to a probability measure μ in $P_1(E)$ if (i) (μ_k) converges weakly to μ and if (ii) $\int_E \|x_0 - x\| \mu_k(\mathrm{d}x) \to \int_E \|x_0 - x\| \mu(\mathrm{d}x)$, where $x_0 \in E$ is arbitrary (Villani, 2008, Definition 6.7). The next proposition offers a characterization of our collection of generative distributions $\mathscr{P}$ in terms of compactness with respect to the weak topology in $P_1(E)$. This result is interesting as it gives some insight into the class of probability measures generated by neural networks.

Proposition 3.2.1. *Assume that Assumption 1 is satisfied. Then the function* $\Theta \ni \theta \mapsto \mu_\theta$ *is continuous with respect to the weak topology in* $P_1(E)$, *and the set of generative distributions* $\mathscr{P}$ *is compact with respect to the weak topology in* $P_1(E)$.

3.2.2 The WGANs and T-WGANs problems

We are now in a position to formally define the WGANs problem. The Wasserstein distance (of order 1) between two probability measures μ and ν in $P_1(E)$ is defined by

$$W_1(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{E\times E} \|x-y\|\pi(\mathrm{d}x,\mathrm{d}y),$$

where $\Pi(\mu,\nu)$ denotes the collection of all joint probability measures on $E\times E$ with marginals μ and ν (e.g., Villani, 2008). It is a finite quantity. In the present article, we will use the dual representation of $W_1(\mu,\nu)$, which comes from the duality theorem of Kantorovich and Rubinstein (1958):

$$W_1(\mu,\nu) = \sup_{f\in \mathrm{Lip}_1} |\mathbb{E}_\mu f - \mathbb{E}_\nu f|,$$

where, for a probability measure π, $\mathbb{E}_\pi f = \int_E f\mathrm{d}\pi$ (note that for $f\in\mathrm{Lip}_1$ and $\pi\in P_1(E)$, the function f is Lebesgue integrable with respect to π).

In this context, it is natural to define the theoretical-WGANs (T-WGANs) problem as minimizing over Θ the Wasserstein distance between $\mu^\star$ and the μ_θ's, i.e.,

$$\inf_{\theta\in\Theta} W_1(\mu_\star,\mu_\theta) = \inf_{\theta\in\Theta} \sup_{f\in\mathrm{Lip}_1} |\mathbb{E}_{\mu_\star} f - \mathbb{E}_{\mu_\theta} f|. \tag{3.2.3}$$

In practice, however, one does not have access to the class of 1-Lipschitz functions, which cannot be parameterized. Therefore, following Arjovsky et al. (2017), the class Lip_1 is restricted to the smaller but parametric set of discriminators $\mathscr{D} = \{D_\alpha : \alpha\in\Lambda\}$ (it is a subset of Lip_1, by Lemma 3.2.1), and this defines the actual WGANs problem:

$$\inf_{\theta\in\Theta} \sup_{\alpha\in\Lambda} |\mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha|. \tag{3.2.4}$$

Problem (3.2.4) is the Wasserstein counterpart of problem (3.1.1). Provided Assumption 1 is satisfied, $\mathscr{D}\subseteq\mathrm{Lip}_1$, and the IPM (Müller, 1997) $d_\mathscr{D}$ is defined for $(\mu,\nu)\in P_1(E)^2$ by

$$d_\mathscr{D}(\mu,\nu) = \sup_{f\in\mathscr{D}} |\mathbb{E}_\mu f - \mathbb{E}_\nu f|. \tag{3.2.5}$$

With this notation, $d_{\mathrm{Lip}_1} = W_1$ and problems (3.2.3) and (3.2.4) can be rewritten as the minimization over Θ of, respectively, $d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta)$ and $d_\mathscr{D}(\mu_\star,\mu_\theta)$. So,

$$\text{T-WGANs: } \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) \quad \text{and} \quad \text{WGANs: } \inf_{\theta\in\Theta} d_\mathscr{D}(\mu_\star,\mu_\theta).$$

Similar objectives have been proposed in the literature, in particular neural net distances (Arora et al., 2017) and adversarial divergences (Liu et al., 2017). These two general approaches include f-GANs (Goodfellow et al., 2014; Nowozin et al., 2016), but also WGANs (Arjovsky et al., 2017), MMD-GANs (Li et al., 2017), and energy-based GANs (Zhao et al., 2016). Using the terminology of Arora et al. (2017), $d_{\mathscr{D}}$ is called a neural IPM. If the theoretical properties of the Wasserstein distance d_{Lip_1} have been largely studied (e.g., Villani, 2008), the story is different for neural IPMs. This is why our next subsection is devoted to the properties of $d_{\mathscr{D}}$.

3.2.3 Some properties of the neural IPM

The study of the neural IPM $d_{\mathscr{D}}$ is essential to assess the driving forces of WGANs architectures. Let us first recall that a mapping $\ell : P_1(E) \times P_1(E) \to [0,\infty)$ is a metric if it satisfies the following three requirements:

(i) $\ell(\mu,\nu) = 0 \iff \mu = \nu$ (discriminative property)

(ii) $\ell(\mu,\nu) = \ell(\nu,\mu)$ (symmetry)

(iii) $\ell(\mu,\nu) \leqslant \ell(\mu,\pi) + \ell(\pi,\nu)$ (triangle inequality).

If (i) is replaced by the weaker requirement $\ell(\mu,\mu) = 0$ for all $\mu \in P_1(E)$, then one speaks of a pseudometric. Furthermore, the (pseudo)metric ℓ is said to metrize weak convergence in $P_1(E)$ (Villani, 2008) if, for all sequences (μ_k) in $P_1(E)$ and all μ in $P_1(E)$, one has $\ell(\mu,\mu_k) \to 0 \iff \mu_k$ converges weakly to μ in $P_1(E)$ as $k \to \infty$. According to Villani (2008, Theorem 6.8), d_{Lip_1} is a metric that metrizes weak convergence in $P_1(E)$.

As far as $d_{\mathscr{D}}$ is concerned, it is clearly a pseudometric on $P_1(E)$ as soon as Assumption 1 is satisfied. Moreover, an elementary application of Dudley (2004, Lemma 9.3.2) shows that if $\mathrm{span}(\mathscr{D})$ (with $\mathrm{span}(\mathscr{D}) = \{\gamma_0 + \sum_{i=1}^n \gamma_i D_i : \gamma_i \in \mathbb{R}, D_i \in \mathscr{D}, n \in \mathbb{N}\}$) is dense in $C_b(E)$, then $d_{\mathscr{D}}$ is a metric on $P_1(E)$, which, in addition, metrizes weak convergence. As in Zhang et al. (2018), Dudley's result can be exploited in the case where the space E is compact to prove that, whenever $\mathscr{D}$ is of the form (3.2.2), $d_{\mathscr{D}}$ is a metric metrizing weak convergence. However, establishing the discriminative property of the pseudometric $d_{\mathscr{D}}$ turns out to be more challenging without an assumption of compactness on E, as is the case in the present study. Our result is encapsulated in the following proposition.

Proposition 3.2.2. *Assume that Assumption 1 is satisfied. Then there exists a discriminator of the form (3.2.2) (i.e., a depth q and widths $v_1,\dots,v_{q-1}$) such that $d_{\mathscr{D}}$ is a metric on $\mathscr{P} \cup \{\mu_\star\}$. In addition, $d_{\mathscr{D}}$ metrizes weak convergence in $\mathscr{P} \cup \{\mu_\star\}$.*

Standard universal approximation theorems (Cybenko, 1989; Hornik et al., 1989; Hornik, 1991) state the density of neural networks in the family of continuous functions defined on compact sets but do not guarantee that the approximator respects a Lipschitz constraint. The proof of Proposition 3.2.2 uses the fact that, under Assumption 1, neural networks of the form (3.2.2) are dense in the space of Lipschitz continuous functions on compact sets, as revealed by Anil et al. (2019).

We deduce from Proposition 3.2.2 that, under Assumption 1, provided enough capacity, the pseudometric $d_{\mathscr{D}}$ can be topologically equivalent to d_{Lip_1} on $\mathscr{P} \cup \{\mu_\star\}$, i.e., the convergent sequences in $(\mathscr{P} \cup \{\mu_\star\}, d_{\mathscr{D}})$ are the same as the convergent sequences in $(\mathscr{P} \cup \{\mu_\star\}, d_{\mathrm{Lip}_1})$ with the same limit—see O'Searcoid (2006, Corollary 13.1.3). We are now ready to discuss some optimality properties of the T-WGANs and WGANs problems, i.e., conditions under which the infimum in $\theta \in \Theta$ and the supremum in $\alpha \in \Lambda$ are reached.

3.2.4 Optimality properties

Recall that for T-WGANs, we minimize over Θ the distance

$$d_{\mathrm{Lip}_1}(\mu^\star, \mu_\theta) = \sup_{f \in \mathrm{Lip}_1} |\mathbb{E}_{\mu^\star} f - \mathbb{E}_{\mu_\theta} f|,$$

whereas for WGANs, we use

$$d_{\mathscr{D}}(\mu^\star, \mu_\theta) = \sup_{\alpha \in \Lambda} |\mathbb{E}_{\mu^\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha|.$$

A first natural question is to know whether for a fixed generator parameter $\theta \in \Theta$, there exists a 1-Lipschitz function (respectively, a discriminative function) that achieves the supremum in $d_{\mathrm{Lip}_1}(\mu^\star, \mu_\theta)$ (respectively, in $d_{\mathscr{D}}(\mu^\star, \mu_\theta)$) over all $f \in \mathrm{Lip}_1$ (respectively, all $\alpha \in \Lambda$). For T-WGANs, Villani (2008, Theorem 5.9) guarantees that the maximum exists, i.e.,

$$\{f \in \mathrm{Lip}_1 : |\mathbb{E}_{\mu_\star} f - \mathbb{E}_{\mu_\theta} f| = d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)\} \neq \varnothing. \tag{3.2.6}$$

For WGANs, we have the following:

Lemma 3.2.2. *Assume that Assumption 1 is satisfied. Then, for all $\theta \in \Theta$,*

$$\{\alpha \in \Lambda : |\mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha| = d_{\mathscr{D}}(\mu_\star, \mu_\theta)\} \neq \varnothing.$$

Thus, provided Assumption 1 is verified, the supremum in α in the neural IPM $d_{\mathscr{D}}$ is always reached. A similar result is proved by Biau et al. (2020) in the case of standard GANs.

We now turn to analyzing the existence of the infimum in θ in the minimization over Θ of $d_{\mathrm{Lip}_1}(\mu^\star, \mu_\theta)$ and $d_{\mathscr{D}}(\mu^\star, \mu_\theta)$. Since the optimization scheme is performed over the parameter set Θ, it is worth considering the following two functions:

$$\xi_{\mathrm{Lip}_1} : \Theta \to \mathbb{R} \qquad \text{and} \qquad \xi_{\mathscr{D}} : \Theta \to \mathbb{R}$$
$$\theta \mapsto d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \qquad\qquad \theta \mapsto d_{\mathscr{D}}(\mu_\star, \mu_\theta).$$

Theorem 3.2.1. *Assume that Assumption 1 is satisfied. Then ξ_{Lip_1} and $\xi_{\mathscr{D}}$ are Lipschitz continuous on Θ, and the Lipschitz constant of $\xi_{\mathscr{D}}$ is independent of $\mathscr{D}$.*

Theorem 3.2.1 extends Arjovsky et al. (2017, Theorem 1), which states that $d_{\mathscr{D}}$ is locally Lipschitz continuous under the additional assumption that E is compact. In contrast, there is no compactness hypothesis in Theorem 3.2.1 and the Lipschitz property is global. The lipschitzness of the function $\xi_{\mathscr{D}}$ is an interesting property of WGANS, in line with many recent empirical works that have shown that gradient-based regularization techniques are efficient for stabilizing the training of GANs and preventing mode collapse (Kodali et al., 2017; Roth et al., 2017; Miyato et al., 2018; Petzka et al., 2018).

In the sequel, we let $\Theta^\star$ and $\bar{\Theta}$ be the sets of optimal parameters, defined by

$$\Theta^\star = \arg\min_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \quad \text{and} \quad \bar{\Theta} = \arg\min_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta).$$

An immediate but useful corollary of Theorem 3.2.1 is as follows:

Corollary 3.2.1. *Assume that Assumption 1 is satisfied. Then $\Theta^\star$ and $\bar{\Theta}$ are non empty.*

Thus, any $\theta^\star \in \Theta^\star$ (respectively, any $\bar{\theta} \in \bar{\Theta}$) is an optimal parameter for the T-WGANs (respectively, the WGANs) problem. Note however that, without further restrictive assumptions on the models, we cannot ensure that $\Theta^\star$ or $\bar{\Theta}$ are reduced to singletons.

3.3 Optimization properties

We are interested in this section in the error made when minimizing over Θ the pseudo-metric $d_{\mathscr{D}}(\mu^\star, \mu_\theta)$ (WGANs problem) instead of $d_{\mathrm{Lip}_1}(\mu^\star, \mu_\theta)$ (T-WGANs problem). This optimization error is represented by the difference

$$\varepsilon_{\mathrm{optim}} = \sup_{\bar{\theta} \in \bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta).$$

It is worth pointing out that we take the supremum over all $\bar{\theta} \in \bar{\Theta}$ since there is no guarantee that two distinct elements $\bar{\theta}_1$ and $\bar{\theta}_2$ of $\bar{\Theta}$ lead to the same distances $d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_1})$ and $d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_2})$. The quantity $\varepsilon_{\mathrm{optim}}$ captures the largest discrepancy between the scores achieved by distributions solving the WGANs problem and the scores of distributions solving the T-WGANs problem. We emphasize that the scores are quantified by the Wasserstein distance d_{Lip_1}, which is the natural metric associated with the problem. We note in particular that $\varepsilon_{\mathrm{optim}} \geqslant 0$. A natural question is whether we can upper bound the difference and obtain some control of $\varepsilon_{\mathrm{optim}}$.

3.3.1 Approximating d_{Lip_1} with $d_{\mathscr{D}}$

As a warm-up, we observe that in the simple but unrealistic case where $\mu^\star \in \mathscr{P}$, provided Assumption 1 is satisfied and the neural IPM $d_{\mathscr{D}}$ is a metric on $\mathscr{P}$ (see Proposition 3.2.2), then $\Theta^\star = \bar{\Theta}$ and $\varepsilon_{\mathrm{optim}} = 0$. However, in the high-dimensional context of WGANs, the parametric class of distributions $\mathscr{P}$ is likely to be "far" from the true distribution $\mu^\star$. This phenomenon is thoroughly discussed in Arjovsky and Bottou (2017, Lemma 2 and Lemma 3) and is often referred to as dimensional misspecification (Roth et al., 2017).

From now on, we place ourselves in the general setting where we have no information on whether the true distribution belongs to $\mathscr{P}$, and start with the following simple observation. Assume that Assumption 1 is satisfied. Then, clearly, since $\mathscr{D} \subseteq \mathrm{Lip}_1$,

$$\inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta). \tag{3.3.1}$$

Inequality (3.3.1) is useful to upper bound $\varepsilon_{\mathrm{optim}}$. Indeed,

$$\begin{aligned}
0 \leqslant \varepsilon_{\mathrm{optim}} &= \sup_{\bar{\theta} \in \bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\
&\leqslant \sup_{\bar{\theta} \in \bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \\
&= \sup_{\bar{\theta} \in \bar{\Theta}} \left[d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - d_{\mathscr{D}}(\mu_\star, \mu_{\bar{\theta}}) \right] \\
&\quad \left(\text{since } \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) = d_{\mathscr{D}}(\mu_\star, \mu_{\bar{\theta}}) \text{ for all } \bar{\theta} \in \bar{\Theta} \right) \\
&\leqslant T_{\mathscr{P}}(\mathrm{Lip}_1, \mathscr{D}),
\end{aligned} \tag{3.3.2}$$

where, by definition,

$$T_{\mathscr{P}}(\mathrm{Lip}_1, \mathscr{D}) = \sup_{\theta \in \Theta} \left[d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) \right] \tag{3.3.3}$$

is the maximum difference in distances on the set of candidate probability distributions in $\mathscr{P}$. Note, since Θ is compact (by Assumption 1) and ξ_{Lip_1} and $\xi_{\mathscr{D}}$ are Lipschitz continuous (by Theorem 3.2.1), that $T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D}) < \infty$. Thus, the loss in performance when comparing T-WGANs and WGANs can be upper-bounded by the maximum difference over $\mathscr{P}$ between the Wasserstein distance and $d_{\mathscr{D}}$.

Observe that when the class of discriminative functions is increased (say $\mathscr{D} \subset \mathscr{D}'$) while keeping the generator fixed, then the bound (3.3.3) gets reduced since $d_{\mathscr{D}}(\mu_\star, \cdot) \leqslant d_{\mathscr{D}'}(\mu_\star, \cdot)$. Similarly, when increasing the class of generative distributions (say $\mathscr{P} \subset \mathscr{P}'$) with a fixed discriminator, then the bound gets bigger, i.e., $T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D}) \leqslant T_{\mathscr{P}'}(\text{Lip}_1, \mathscr{D})$. It is important to note that the conditions $\mathscr{D} \subset \mathscr{D}'$ and/or $\mathscr{P} \subset \mathscr{P}'$ are easily satisfied for classes of functions parameterized with neural networks using either rectifier or GroupSort activation functions, just by increasing the width and/or the depth of the networks.

Our next theorem states that, as long as the distributions of $\mathscr{P}$ are generated by neural networks with bounded parameters (Assumption 1), then one can control $T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D})$.

Theorem 3.3.1. *Assume that Assumption 1 is satisfied. Then, for all $\varepsilon > 0$, there exists a discriminator $\mathscr{D}$ of the form (3.2.2) such that*

$$0 \leqslant \varepsilon_{\text{optim}} \leqslant T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D}) \leqslant c\varepsilon,$$

where $c > 0$ is a constant independent from ε.

Theorem 3.3.1 is important because it shows that for any collection of generative distributions $\mathscr{P}$ and any approximation threshold ε, one can find a discriminator such that the loss in performance $\varepsilon_{\text{optim}}$ is (at most) of the order of ε. In other words, there exists $\mathscr{D}$ of the form (3.2.2) such that $T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D})$ is arbitrarily small. We note however that Theorem 3.3.1 is an existence theorem that does not give any particular information on the depth and/or the width of the neural networks in $\mathscr{D}$. The key argument to prove Theorem 3.3.1 is Anil et al. (2019, Theorem 3), which states that the set of Lipschitz neural networks are dense in the set of Lipschitz continuous functions on a compact space.

3.3.2 Equivalence properties

The quantity $T_{\mathscr{P}}(\text{Lip}_1, \mathscr{D})$ is of limited practical interest, as it involves a supremum over all $\theta \in \Theta$. Moreover, another caveat is that the definition of $\varepsilon_{\text{optim}}$ assumes that one has access to $\bar{\Theta}$. Therefor, our next goal is to enrich Theorem 3.3.1 by taking into account the fact that numerical procedures do not reach $\bar{\theta} \in \bar{\Theta}$ but rather an ε-approximation of it.

One way to approach the problem is to look for another form of equivalence between d_{Lip_1} and $d_{\mathscr{D}}$. As one is optimizing $d_{\mathscr{D}}$ instead of d_{Lip_1}, we would ideally like that the two IPMs behave "similarly", in the sense that minimizing $d_{\mathscr{D}}$ leads to a solution that is still close to the true distribution with respect to d_{Lip_1}. Assuming that Assumption 1 is satisfied, we let, for any $\mu \in P_1(E)$ and $\varepsilon > 0$, $\mathscr{M}_\ell(\mu, \varepsilon)$ be the set of ε-solutions to the optimization problem of interest, that is the subset of Θ defined by

$$\mathscr{M}_\ell(\mu, \varepsilon) = \big\{\theta \in \Theta : \ell(\mu, \mu_\theta) - \inf_{\theta \in \Theta} \ell(\mu, \mu_\theta) \leqslant \varepsilon\big\},$$

with $\ell = d_{\mathrm{Lip}_1}$ or $\ell = d_{\mathscr{D}}$.

Definition 3.3.1. *Let $\varepsilon > 0$. We say that d_{Lip_1} can be ε-substituted by $d_{\mathscr{D}}$ if there exists $\delta > 0$ such that*

$$\mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta) \subseteq \mathscr{M}_{d_{\mathrm{Lip}_1}}(\mu_\star, \varepsilon).$$

In addition, if d_{Lip_1} can be ε-substituted by $d_{\mathscr{D}}$ for all $\varepsilon > 0$, we say that d_{Lip_1} can be fully substituted by $d_{\mathscr{D}}$.

The rationale behind this definition is that by minimizing the neural IPM $d_{\mathscr{D}}$ close to optimality, one can be guaranteed to be also close to optimality with respect to the Wasserstein distance d_{Lip_1}. In the sequel, given a metric d, the notation $d(x, F)$ denotes the distance of x to the set F, that is, $d(x, F) = \inf_{f \in F} d(x, f)$.

Proposition 3.3.1. *Assume that Assumption 1 is satisfied. Then, for all $\varepsilon > 0$, there exists $\delta > 0$ such that, for all $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$, one has $d(\theta, \bar{\Theta}) \leqslant \varepsilon$.*

Corollary 3.3.1. *Assume that Assumption 1 is satisfied and that $\Theta^\star = \bar{\Theta}$. Then d_{Lip_1} can be fully substituted by $d_{\mathscr{D}}$.*

Proof. Let $\varepsilon > 0$. By Theorem 3.2.1, we know that the function $\Theta \ni \theta \mapsto d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)$ is Lipschitz continuous. Thus, there exists $\eta > 0$ such that, for all $(\theta, \theta') \in \Theta^2$ satisfying $\|\theta - \theta'\| \leqslant \eta$, one has $|d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta'})| \leqslant \varepsilon$. Besides, using Proposition 3.3.1, there exists $\delta > 0$ such that, for all $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$, one has $d(\theta, \bar{\Theta}) \leqslant \eta$.

Now, let $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$. Since $d(\theta, \bar{\Theta}) \leqslant \eta$ and $\bar{\Theta} = \Theta^\star$, we have $d(\theta, \Theta^\star) \leqslant \eta$. Consequently, $|d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)| \leqslant \varepsilon$, and so, $\theta \in \mathscr{M}_{d_{\mathrm{Lip}_1}}(\mu_\star, \varepsilon)$. □

Corollary 3.3.1 is interesting insofar as when both $d_{\mathscr{D}}$ and d_{Lip_1} have the same minimizers over Θ, then minimizing one close to optimality is the same as minimizing the other. The requirement $\Theta^\star = \bar{\Theta}$ can be relaxed by leveraging what has been studied in the previous subsection about $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D})$.

Lemma 3.3.1. *Assume that Assumption 1 is satisfied, and let $\varepsilon > 0$. If $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D}) \leqslant \varepsilon$, then d_{Lip_1} can be $(\varepsilon + \delta)$-substituted by $d_{\mathscr{D}}$ for all $\delta > 0$.*

Proof. Let $\varepsilon > 0$, $\delta > 0$, and $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$, i.e., $d_{\mathscr{D}}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant \delta$. We have

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) &\leqslant d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \\ &\quad \text{(by inequality (3.3.1))} \\ &\leqslant d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) + \delta \\ &\leqslant T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D}) + \delta \leqslant \varepsilon + \delta. \end{aligned}$$

□

Lemma 3.3.1 stresses the importance of $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D})$ in the performance of WGANs. Indeed, the smaller $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D})$, the closer we will be to optimality after training. Moving on, to derive sufficient conditions under which d_{Lip_1} can be substituted by $d_{\mathscr{D}}$ we introduce the following definition:

Definition 3.3.2. *We say that d_{Lip_1} is monotonously equivalent to $d_{\mathscr{D}}$ on $\mathscr{P}$ if there exists a continuously differentiable, strictly increasing function $f : \mathbb{R}^+ \to \mathbb{R}^+$ and $(a, b) \in (\mathbb{R}_+^\star)^2$ such that*

$$\forall \mu \in \mathscr{P},\ a f(d_{\mathscr{D}}(\mu_\star, \mu)) \leqslant d_{\mathrm{Lip}_1}(\mu_\star, \mu) \leqslant b f(d_{\mathscr{D}}(\mu_\star, \mu)).$$

Here, it is assumed implicitly that $\mathscr{D} \subseteq \mathrm{Lip}_1$. At the end of the subsection, we stress, empirically, that Definition 3.3.2 is easy to check for simple classes of generators. A consequence of this definition is encapsulated in the following lem.

Lemma 3.3.2. *Assume that Assumption 1 is satisfied, and that d_{Lip_1} and $d_{\mathscr{D}}$ are monotonously equivalent on $\mathscr{P}$ with $a = b$ (that is, $d_{\mathrm{Lip}_1} = f \circ d_{\mathscr{D}}$). Then $\Theta^\star = \bar{\Theta}$ and d_{Lip_1} can be fully substituted by $d_{\mathscr{D}}$.*

To complete Lemma 3.3.2, we now tackle the case $a < b$.

Proposition 3.3.2. *Assume that Assumption 1 is satisfied, and that d_{Lip_1} and $d_{\mathscr{D}}$ are monotonously equivalent on $\mathscr{P}$. Then, for any $\delta \in (0, 1)$, d_{Lip_1} can be ε-substituted by $d_{\mathscr{D}}$ with $\varepsilon = (b - a) f(\inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta)) + O(\delta)$.*

Proposition 3.3.2 states that we can reach ε-minimizers of d_{Lip_1} by solving the WGANs problem up to a precision sufficiently small, for all ε larger than a bias induced by the model $\mathscr{P}$ and by the discrepancy between d_{Lip_1} and $d_{\mathscr{D}}$.

In order to validate Definition 3.3.2, we slightly depart from the WGANs setting and run a series of small experiments in the simplified setting where both $\mu^\star$ and $\mu \in \mathscr{P}$ are bivariate mixtures of independent Gaussian distributions with K components ($K = 1, 2, 3, 25$). We consider two classes of discriminators $\{\mathscr{D}_q : q = 2, 6\}$ of the form (3.2.2), with growing depth q (the width of the hidden layers is kept constant equal to 20). Our goal is to exemplify the relationship between the distances d_{Lip_1} and $d_{\mathscr{D}_q}$ by looking whether d_{Lip_1} is monotonously equivalent to $d_{\mathscr{D}_q}$.

First, for each K, we randomly draw 40 different pairs of distributions $(\mu_\star, \mu)$ among the set of mixtures of bivariate Gaussian densities with K components. Then, for each of these pairs, we compute an approximation of d_{Lip_1} by averaging the Wasserstein distance between finite samples of size 4096 over 20 runs. This operation is performed using the Python package by Flamary and Courty (2017). For each pair of distributions, we also calculate the corresponding IPMs $d_{\mathscr{D}_q}(\mu_\star, \mu)$. We finally compare d_{Lip_1} and $d_{\mathscr{D}_q}$ by approximating their relationship with a parabolic fit. Results are presented in Figure 3.1, which depicts in particular the best parabolic fit, and shows the corresponding Least Relative Error (LRE) together with the width $(b-a)$ from Definition 3.3.2. In order to enforce the discriminator to verify Assumption 1, we use the orthonormalization of Bjorck and Bowie (1971), as done for example in Anil et al. (2019).

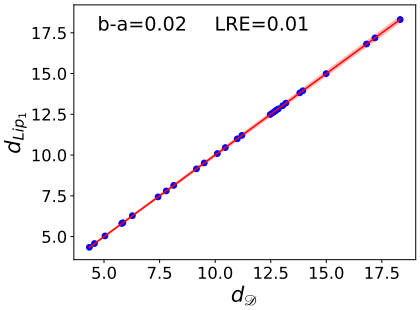
Interestingly, we see that when the class of discriminative functions gets larger (i.e., when q increases), then both metrics start to behave similarly (i.e., the range $(b-a)$ gets thinner), independently of K (Figure 3.1a to Figure 3.1f). This tends to confirm that d_{Lip_1} can be considered as monotonously equivalent to $d_{\mathscr{D}_q}$ for q large enough. On the other hand, for a fixed depth q, when allowing for more complex distributions, the width $(b-a)$ increases. This is particularly clear in Figure 3.1g and Figure 3.1h, which show the fits obtained when merging all pairs for $K = 1, 4, 9, 25$ (for both $\mu_\star$ and $\mathscr{P}$).

These figures illustrate the fact that, for a fixed discriminator, the monotonous equivalence between d_{Lip_1} and $d_{\mathscr{D}}$ seems to be a more demanding assumption when the class of generative distributions becomes too large.

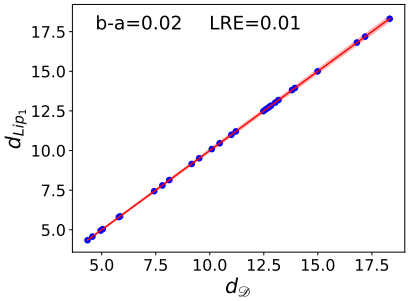
3.3.3 Motivating the use of deep GroupSort neural networks

The objective of this subsection is to provide some justification for the use of deep GroupSort neural networks in the field of WGANs. This short discussion is motivated by the observation of Anil et al. (2019, Theorem 1), who stress that norm-constrained ReLU neural networks are not well-suited for learning non-linear 1-Lipschitz functions.

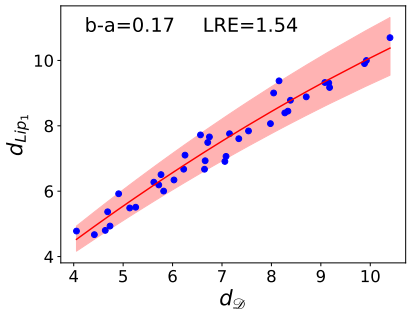
The next lemma shows that networks of the form (3.2.2), which use GroupSort activations, can recover any 1-Lipschitz function belonging to the class AFF of real-valued affine functions on E.



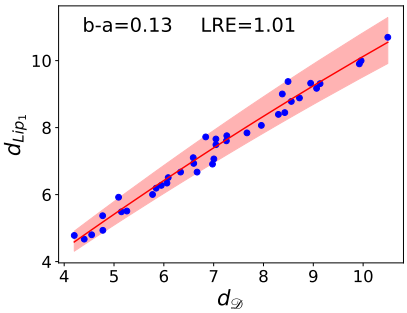
(a) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=2$, $K=1$.



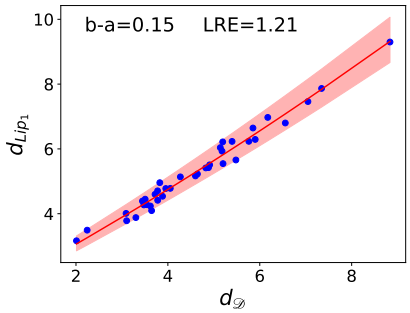
(b) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=5$, $K=1$.



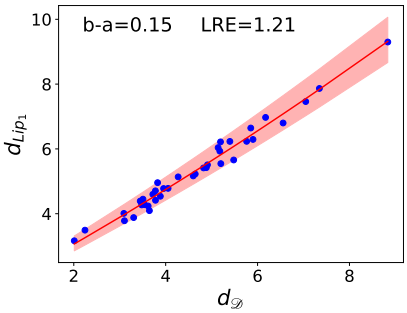
(c) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=2$, $K=4$.



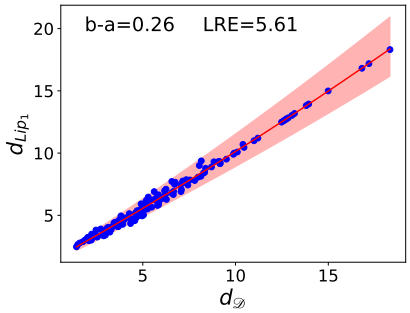
(d) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=5$, $K=4$.



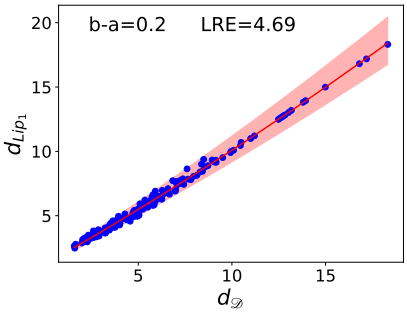
(e) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=2$, $K=9$.



(f) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=5$, $K=9$.



(g) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=2$, $K=1,4,9,25$.



(h) $d_{\mathscr{D}_q}$ vs. d_{Lip_1}, $q=5$, $K=1,4,9,25$.

Fig. 3.1 Scatter plots of 40 pairs of distances simultaneously measured with d_{Lip_1} and $d_{\mathscr{D}_q}$, for $q=2,5$ and $K=1,4,9,25$. The red curve is the optimal parabolic fitting and LRE refers to the Least Relative Error. The red zone is the envelope obtained by stretching the optimal curve from b to a.

Lemma 3.3.3. *Let $f : E \to \mathbb{R}$ be in $\mathrm{AFF} \cap \mathrm{Lip}_1$. Then there exists a neural network of the form (3.2.2) verifying Assumption 1, with $q = 2$ and $v_1 = 2$, that can represent f.*

Motivated by Lemma 3.3.3, we show that, in some specific cases, the Wasserstein distance d_{Lip_1} can be approached by only considering affine functions, thus motivating the use of neural networks of the form (3.2.2). Recall that the support S_μ of a probability measure μ is the smallest subset of μ-measure 1.

Lemma 3.3.4. *Let μ and ν be two probability measures in $P_1(E)$. Assume that S_μ and S_ν are one-dimensional disjoint intervals included in the same line. Then $d_{\mathrm{Lip}_1}(\mu,\nu) = d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu)$.*

Lemma 3.3.4 is interesting insofar as it describes a specific case where the discriminator can be restricted to affine functions while keeping the identity $d_{\mathrm{Lip}_1} = d_{\mathscr{D}}$. We consider in the next lemma a slightly more involved setting, where the two distributions μ and ν are multivariate Gaussian with the same covariance matrix.

Lemma 3.3.5. *Let $(m_1, m_2) \in (\mathbb{R}^D)^2$, and let $\Sigma \in \mathscr{M}_{(D,D)}$ be a positive semi-definite matrix. Assume that μ is Gaussian $\mathscr{N}(m_1,\Sigma)$ and that ν is Gaussian $\mathscr{N}(m_2,\Sigma)$. Then $d_{\mathrm{Lip}_1}(\mu,\nu) = d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu)$.*

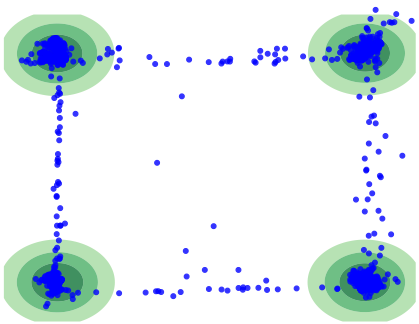
Yet, assuming multivariate Gaussian distributions might be too restrictive. Therefore, we now assume that both distributions lay on disjoint compact supports sufficiently distant from one another. Recall that for a set $S \subseteq E$, the diameter of S is $\mathrm{diam}(S) = \sup_{(x,y)\in S^2} \|x-y\|$, and that the distance between two sets S and T is defined by $d(S,T) = \inf_{(x,y)\in S\times T} \|x-y\|$.

Proposition 3.3.3. *Let $\varepsilon > 0$, and let μ and ν be two probability measures in $P_1(E)$ with compact convex supports S_μ and S_ν. Assume that $\max(\mathrm{diam}(S_\mu), \mathrm{diam}(S_\nu)) \leqslant \varepsilon d(S_\mu, S_\nu)$. Then*

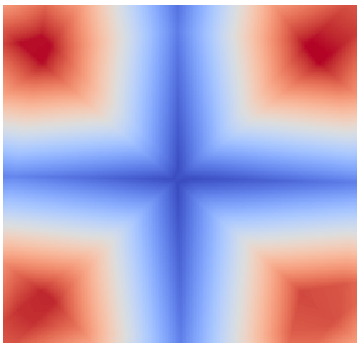
$$d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu) \leqslant d_{\mathrm{Lip}_1}(\mu,\nu) \leqslant (1+2\varepsilon) d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu).$$

Observe that in the case where neither μ nor ν are Dirac measures, then the assumption of the lemma imposes that $S_\mu \cap S_\nu = \emptyset$. In the context of WGANs, it is highly likely that the generator badly approximates the true distribution $\mu_\star$ at the beginning of training. The setting of Proposition 3.3.3 is thus interesting insofar as $\mu_\star$ and the generative distribution will most certainly verify the assumption on the diameters at this point in the learning process. However, in the common case where the true distribution lays on disconnected manifolds, the assumptions of the proposition are not valid anymore, and it would therefore be interesting to show a similar result using the broader set of piecewise linear functions on E.

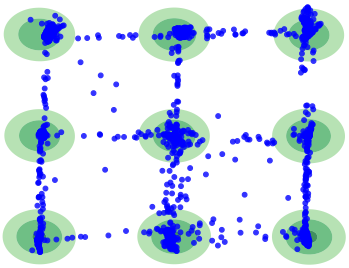
As an empirical illustration, consider the synthetic setting where one tries to approximate a bivariate mixture of independent Gaussian distributions with respectively 4 (Figure 3.2a) and 9 (Figure 3.2c) modes. As expected, the optimal discriminator takes the form of a piecewise linear function, as illustrated by Figure 3.2b and Figure 3.2d, which display heatmaps of the discriminator's output. Interestingly, we see that the number of linear regions increases with the number K of components of $\mu_\star$.



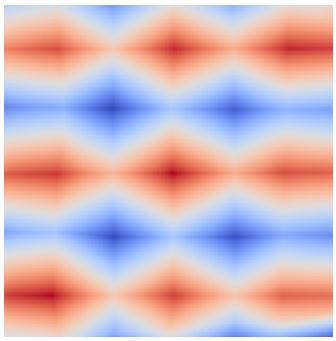
(a) True distribution $\mu_\star$ (mixture of $K = 4$ bivariate Gaussian densities, green circles) and 2000 data points sampled from the generator $\mu_{\tilde{\theta}}$ (blue dots).



(b) Heatmap of the discriminator's output on a mixture of $K = 4$ bivariate Gaussian densities.



(c) True distribution $\mu_\star$ (mixture of $K = 9$ bivariate Gaussian densities, green circles) and 2000 data points sampled from the generator $\mu_{\tilde{\theta}}$ (blue dots).



(d) Heatmap of the discriminator's output on a mixture of $K = 9$ bivariate Gaussian densities.

Fig. 3.2 Illustration of the usefulness of GroupSort neural networks when dealing with the learning of mixtures of Gaussian distributions. In both cases, we have $p = q = 3$.

These empirical results stress that when $\mu_\star$ gets more complex, if the discriminator ought to correctly approximate the Wasserstein distance, then it should parameterize piecewise linear functions with growing numbers of regions. While we enlighten properties of Groupsort networks, many recent theoretical works have been studying the number of regions of deep

ReLU neural networks (Pascanu et al., 2013; Montúfar et al., 2014; Arora et al., 2018; Serra et al., 2018). In particular, Montúfar et al. (2014, Theorem 5) states that the number of linear regions of deep models grows exponentially with the depth and polynomially with the width. This, along with our observations, is an interesting avenue to choose the architecture of the discriminator.

3.4 Asymptotic properties

In practice, one never has access to the distribution $\mu_\star$ but rather to a finite collection of i.i.d. observations $X_1,\dots,X_n$ distributed according to $\mu_\star$. Thus, for the remainder of the article, we let μ_n be the empirical measure based on $X_1,\dots,X_n$, that is, for any Borel subset A of E, $\mu_n(A) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{X_i\in A}$. With this notation, the empirical counterpart of the WGANs problem is naturally defined as minimizing over Θ the quantity $d_{\mathscr{D}}(\mu_n,\mu_\theta)$. Equivalently, we seek to solve the following optimization problem:

$$\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta) = \inf_{\theta\in\Theta}\sup_{\alpha\in\Lambda}\Big[\frac{1}{n}\sum_{i=1}^n D_\alpha(X_i) - \mathbb{E}D_\alpha(G_\theta(Z))\Big]. \tag{3.4.1}$$

Assuming that Assumption 1 is satisfied, we have, as in Corollary 3.2.1, that the infimum in (3.4.1) is reached. We therefore consider the set of empirical optimal parameters

$$\hat{\Theta}_n = \operatorname*{arg\,min}_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta),$$

and let $\hat{\theta}_n$ be a specific element of $\hat{\Theta}_n$ (note that the choice of $\hat{\theta}_n$ has no impact on the value of the minimum). We note that $\hat{\Theta}_n$ (respectively, $\hat{\theta}_n$) is the empirical counterpart of $\bar{\Theta}$ (respectively, $\bar{\theta}$). Section 3.3 was mainly devoted to the analysis of the difference $\varepsilon_{\text{optim}}$. In this section, we are willing to take into account the effect of having finite samples. Thus, in line with the above, we are now interested in the generalization properties of WGANs and look for upper-bounds on the quantity

$$0 \leqslant d_{\mathrm{Lip}_1}(\mu_\star,\mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta). \tag{3.4.2}$$

Arora et al. (2017, Theorem 3.1) states an asymptotic result showing that when provided enough samples, the neural IPM $d_{\mathscr{D}}$ generalizes well, in the sense that for any pair $(\mu,\nu)\in P_1(E)^2$, the difference $|d_{\mathscr{D}}(\mu,\nu) - d_{\mathscr{D}}(\mu_n,\nu_n)|$ can be arbitrarily small with high probability. However, this result does not give any information on the quantity of interest $d_{\mathrm{Lip}_1}(\mu_\star,\mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta)$. Closer to our current work, Zhang et al. (2018) provide bounds for

$d_{\mathscr{D}}(\mu_\star, \mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta)$, starting from the observation that

$$0 \leqslant d_{\mathscr{D}}(\mu_\star, \mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant 2d_{\mathscr{D}}(\mu_\star, \mu_n). \tag{3.4.3}$$

In the present article, we develop a complementary point of view and measure the generalization properties of WGANs on the basis of the Wasserstein distance d_{Lip_1}, as in equation (3.4.2). Our approach is motivated by the fact that the neural IPM $d_{\mathscr{D}}$ is only used for easing the optimization process and, accordingly, that the performance should be assessed on the basis of the distance d_{Lip_1}, not $d_{\mathscr{D}}$.

Note that $\hat{\theta}_n$, which minimizes $d_{\mathscr{D}}(\mu_n, \mu_\theta)$ over Θ, may not be unique. Besides, there is no guarantee that two distinct elements $\theta_{n,1}$ and $\theta_{n,2}$ of $\hat{\Theta}_n$ lead to the same distance $d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_{n,1}})$ and $d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_{n,2}})$ (again, $\hat{\theta}_n$ is computed with $d_{\mathscr{D}}$, not with d_{Lip_1}). Therefore, in order to upper-bound the error in (3.4.2), we let, for each $\theta_n \in \hat{\Theta}_n$,

$$\bar{\theta}_n \in \operatorname*{arg\,min}_{\bar{\theta}\in\bar{\Theta}} \|\theta_n - \bar{\theta}\|.$$

The rationale behind the definition of $\bar{\theta}_n$ is that we expect it to behave "similarly" to θ_n. Following our objective, the error can be decomposed as follows:

$$\begin{aligned}
0 &\leqslant d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\
&\leqslant \sup_{\theta_n\in\hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\
&= \sup_{\theta_n\in\hat{\Theta}_n} \left[d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_n}) + d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_n})\right] - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\
&\leqslant \sup_{\theta_n\in\hat{\Theta}_n} \left[d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_n})\right] + \sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\
&= \varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}},
\end{aligned} \tag{3.4.4}$$

where we set $\varepsilon_{\mathrm{estim}} = \sup_{\theta_n\in\hat{\Theta}_n} [d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_n})]$. Notice that this supremum can be positive or negative. However, it can be shown to converge to 0 almost surely when $n \to \infty$.

Lemma 3.4.1. *Assume that Assumption 1 is satisfied. Then* $\lim_{n\to\infty} \varepsilon_{\mathrm{estim}} = 0$ *almost surely.*

Going further with the analysis of (3.4.2), the sum $\varepsilon_{\text{estim}} + \varepsilon_{\text{optim}}$ is bounded as follows:

$$\begin{aligned}\varepsilon_{\text{estim}} + \varepsilon_{\text{optim}} &\leqslant \sup_{\theta_n \in \hat{\Theta}_n} \left[d_{\text{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\text{Lip}_1}(\mu_\star, \mu_{\tilde{\theta}_n}) \right] + T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}) \\ &\qquad \text{(by inequality (3.3.2))} \\ &\leqslant \sup_{\theta_n \in \hat{\Theta}_n} \left[d_{\text{Lip}_1}(\mu_\star, \mu_{\theta_n}) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \right] + T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}).\end{aligned}$$

Hence,

$$\begin{aligned}&\varepsilon_{\text{estim}} + \varepsilon_{\text{optim}} \\ &\quad \leqslant \sup_{\theta_n \in \hat{\Theta}_n} \left[d_{\text{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathscr{D}}(\mu_\star, \mu_{\theta_n}) + d_{\mathscr{D}}(\mu_\star, \mu_{\theta_n}) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \right] + T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}) \\ &\quad \leqslant \sup_{\theta_n \in \hat{\Theta}_n} \left[d_{\text{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathscr{D}}(\mu_\star, \mu_{\theta_n}) \right] + 2 d_{\mathscr{D}}(\mu_\star, \mu_n) + T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}) \\ &\qquad \text{(upon noting that inequality (3.4.3) is also valid for any } \theta_n \in \hat{\Theta}_n) \\ &\quad \leqslant 2 T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}) + 2 d_{\mathscr{D}}(\mu_\star, \mu_n). \qquad (3.4.5)\end{aligned}$$

The above bound is a function of both the generator and the discriminator. The term $T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D})$ is increasing when the capacity of the generator is increasing. The discriminator, however, plays a more ambivalent role, as already pointed out by Zhang et al. (2018). On the one hand, if the discriminator's capacity decreases, the gap between $d_{\mathscr{D}}$ and d_{Lip_1} gets bigger and $T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D})$ increases. On the other hand, discriminators with bigger capacities ought to increase the contribution $d_{\mathscr{D}}(\mu_\star, \mu_n)$. In order to bound $d_{\mathscr{D}}(\mu_\star, \mu_n)$, Proposition 3.4.1 below extends Zhang et al. (2018, Theorem 3.1), in the sense that it does not require the set of discriminative functions nor the space E to be bounded. Recall that, for $\gamma > 0$, $\mu_\star$ is said to be γ sub-Gaussian (Jin et al., 2019) if

$$\forall v \in \mathbb{R}^d, \ \mathbb{E} e^{v \cdot (T - \mathbb{E}T)} \leqslant e^{\frac{\gamma^2 \|v\|^2}{2}},$$

where T is a random vector with probability distribution $\mu^\star$ and $\cdot$ denotes the dot product in $\mathbb{R}^D$.

Proposition 3.4.1. *Assume that Assumption 1 is satisfied, let $\eta \in (0,1)$, and let $\mathscr{D}$ be a discriminator of the form (3.2.2).*

(*i*) *If $\mu_\star$ has compact support with diameter B, then there exists a constant $c_1 > 0$ such that, with probability at least $1-\eta$,*

$$d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \frac{c_1}{\sqrt{n}} + B\sqrt{\frac{\log(1/\eta)}{2n}}.$$

(*ii*) *More generally, if $\mu_\star$ is γ sub-Gaussian, then there exists a constant $c_2 > 0$ such that, with probability at least $1-\eta$,*

$$d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \frac{c_2}{\sqrt{n}} + 8\gamma\sqrt{eD}\sqrt{\frac{\log(1/\eta)}{n}}.$$

The result of Proposition 3.4.1 has to be compared with convergence rates of the Wasserstein distance. According to Fournier and Guillin (2015, Theorem 1), when the dimension D of E is such that $D > 2$, if $\mu_\star$ has a second-order moment, then there exists a constant c such that

$$0 \leqslant \mathbb{E} d_{\mathrm{Lip}_1}(\mu_\star, \mu_n) \leqslant \frac{c}{n^{1/D}}.$$

Thus, when the space E is of high dimension (e.g., in image generation tasks), under the conditions of Proposition 3.4.1, the pseudometric $d_{\mathscr{D}}$ provides much faster rates of convergence for the empirical measure. However, one has to keep in mind that both constants c_1 and c_2 grow in $O(qQ^{3/2}(D^{1/2}+q))$.

Our Theorem 3.3.1 states the existence of a discriminator such that $\varepsilon_{\text{optim}}$ can be arbitrarily small. It is therefore reasonable, in view of inequality (3.4.5), to expect that the sum $\varepsilon_{\text{estim}} + \varepsilon_{\text{optim}}$ can also be arbitrarily small, at least in an asymptotic sense. This is encapsulated in Theorem 3.4.1 below.

Theorem 3.4.1. *Assume that Assumption 1 is satisfied, and let $\eta \in (0,1)$.*

(*i*) *If $\mu_\star$ has compact support with diameter B, then, for all $\varepsilon > 0$, there exists a discriminator $\mathscr{D}$ of the form (3.2.2) and a constant $c_1 > 0$ (function of ε) such that, with probability at least $1-\eta$,*

$$0 \leqslant \varepsilon_{\text{estim}} + \varepsilon_{\text{optim}} \leqslant 2\varepsilon + \frac{2c_1}{\sqrt{n}} + 2B\sqrt{\frac{\log(1/\eta)}{2n}}.$$

(*ii*) *More generally, if $\mu_\star$ is γ sub-Gaussian, then, for all $\varepsilon > 0$, there exists a discriminator $\mathscr{D}$ of the form (3.2.2) and a constant $c_2 > 0$ (function of ε) such that, with probability at least $1-\eta$,*

$$0 \leqslant \varepsilon_{\text{estim}} + \varepsilon_{\text{optim}} \leqslant 2\varepsilon + \frac{2c_2}{\sqrt{n}} + 16\gamma\sqrt{eD}\sqrt{\frac{\log(1/\eta)}{n}}.$$

Theorem 3.4.1 states that, asymptotically, the optimal parameters in $\hat{\Theta}_n$ behave properly. A caveat is that the definition of $\varepsilon_{\text{estim}}$ uses $\hat{\Theta}_n$. However, in practice, one never has access to $\hat{\theta}_n$, but rather to an approximation of this quantity obtained by gradient descent algorithms. Thus, in line with Definition 3.3.1, we introduce the concept of empirical substitution:

Definition 3.4.1. *Let $\varepsilon > 0$ and $\eta \in (0,1)$. We say that d_{Lip_1} can be empirically ε-substituted by $d_{\mathscr{D}}$ if there exists $\delta > 0$ such that, for all n large enough, with probability at least $1-\eta$,*

$$\mathscr{M}_{d_{\mathscr{D}}}(\mu_n, \delta) \subseteq \mathscr{M}_{d_{\text{Lip}_1}}(\mu_\star, \varepsilon). \tag{3.4.6}$$

The rationale behind this definition is that if (3.4.6) is satisfied, then by minimizing the IPM $d_{\mathscr{D}}$ close to optimality in (3.4.1), one can be guaranteed to be also close to optimality in (3.2.3) with high probability. We stress that Definition 3.4.1 is the empirical counterpart of Definition 3.3.1.

Proposition 3.4.2. *Assume that Assumption 1 is satisfied and that $\mu_\star$ is sub-Gaussian. Let $\varepsilon > 0$. If $T_{\mathscr{D}}(\text{Lip}_1, \mathscr{D}) \leqslant \varepsilon$, then d_{Lip_1} can be empirically $(\varepsilon + \delta)$-substituted by $d_{\mathscr{D}}$ for all $\delta > 0$.*

This proposition is the empirical counterpart of Lemma 3.3.1. It underlines the fact that by minimizing the pseudometric $d_{\mathscr{D}}$ between the empirical measure μ_n and the set of generative distributions $\mathscr{P}$ close to optimality, one can control the loss in performance under the metric d_{Lip_1}.

Let us finally mention that it is also possible to provide asymptotic results on the sequences of parameters $(\hat{\theta}_n)$, keeping in mind that $\hat{\Theta}_n$ and $\bar{\Theta}$ are not necessarily reduced to singletons.

Lemma 3.4.2. *Assume that Assumption 1 is satisfied. Let $(\hat{\theta}_n)$ be a sequence of optimal parameters that converges almost surely to $z \in \Theta$. Then $z \in \bar{\Theta}$ almost surely.*

Proof. Let the sequence $(\hat{\theta}_n)$ converge almost surely to some $z \in \Theta$. By Theorem 3.2.1, the function $\Theta \ni \theta \mapsto d_{\mathscr{D}}(\mu_\star, \mu_\theta)$ is continuous, and therefore, almost surely, $\lim_{n\to\infty} d_{\mathscr{D}}(\mu_\star, \mu_{\hat{\theta}_n}) = d_{\mathscr{D}}(\mu_\star, \mu_z)$. Using inequality (3.4.3), we see that, almost surely,

$$\begin{aligned} 0 \leqslant d_{\mathscr{D}}(\mu_\star, \mu_z) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) &= \lim_{n\to\infty} d_{\mathscr{D}}(\mu_\star, \mu_{\hat{\theta}_n}) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \\ &\leqslant \liminf_{n\to\infty} 2 d_{\mathscr{D}}(\mu_\star, \mu_n). \end{aligned}$$

Using Dudley (2004, Theorem 11.4.1) and the strong law of large numbers, we have that the sequence of empirical measures (μ_n) almost surely converges weakly to $\mu_\star$ in $P_1(E)$. Besides, since $d_{\mathscr{D}}$ metrizes weak convergence in $P_1(E)$ (by Proposition 3.2.2), we conclude that $z \in \bar{\Theta}$ almost surely. □

3.5 Understanding the performance of WGANs

In order to better understand the overall performance of the WGANs architecture, it is instructive to decompose the final loss $d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\hat{\theta}_n})$ as in (3.4.4):

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\hat{\theta}_n}) &\leqslant \varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}} + \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \\ &= \varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}} + \varepsilon_{\mathrm{approx}}, \end{aligned} \tag{3.5.1}$$

where

- (i) $\varepsilon_{\mathrm{estim}}$ matches up with the use of a data-dependent optimal parameter $\hat{\theta}_n$, based on the training set $X_1, \ldots, X_n$ drawn from $\mu_\star$;
- (ii) $\varepsilon_{\mathrm{optim}}$ corresponds to the loss in performance when using $d_{\mathscr{D}}$ as training loss instead of d_{Lip_1} (this term has been thoroughly studied in Section 3.3);
- (iii) and $\varepsilon_{\mathrm{approx}}$ stresses the capacity of the parametric family of generative distributions $\mathscr{P}$ to approach the unknown distribution $\mu_\star$.

Close to our work are the articles by Liang (2018), Singh et al. (2018), and Uppal et al. (2019), who study statistical properties of GANs. Liang (2018) and Singh et al. (2018) exhibit rates of convergence under an IPM-based loss for estimating densities that live in Sobolev spaces, while Uppal et al. (2019) explore the case of Besov spaces. Remarkably, Liang (2018) discusses bounds for the Kullback-Leibler divergence, the Hellinger divergence, and the Wasserstein distance between $\mu_\star$ and $\mu_{\hat{\theta}_n}$. These bounds are based on a different decomposition of the loss and offer a complementary point of view. We emphasize that, in the present article, no density assumption is made neither on the class of generative distributions $\mathscr{P}$ nor on the target distribution $\mu_\star$.

3.5.1 Synthetic experiments

Our goal in this subsection is to illustrate (3.5.1) by running a set of experiments on synthetic datasets. The true probability measure $\mu_\star$ is assumed to be a mixture of bivariate Gaussian distributions with either 1, 4, or 9 components. This simple setting allows us to control the complexity of $\mu_\star$, and, in turn, to better assess the impact of both the generator's and discriminator's capacities. We use growing classes of generators of the form (3.2.1), namely $\{\mathscr{G}_p : p = 2,3,5,7\}$, and growing classes of discriminators of the form (3.2.2), namely $\{\mathscr{D}_q : q = 2,3,5,7\}$. For both the generator and the discriminator, the width of the hidden layers is kept constant equal to 20.

Two metrics are computed to evaluate the behavior of the different generative models. First, we use the Wasserstein distance between the true distribution (either $\mu_\star$ or its empirical version μ_n) and the generative distribution (either $\mu_{\bar{\theta}}$ or $\mu_{\hat{\theta}_n}$). This distance is calculated by using the Python package by Flamary and Courty (2017), via finite samples of size 4096 (average over 20 runs). Second, we use the recall metric (the higher, the better), proposed by Kynkäänniemi et al. (2019). Roughly, this metric measures "how much" of the true distribution (either $\mu_\star$ or μ_n) can be reconstructed by the generative distribution (either $\mu_{\bar{\theta}}$ or $\mu_{\hat{\theta}_n}$). At the implementation level, this score is based on k-nearest neighbor nonparametric density estimation. It is computed via finite samples of size 4096 (average over 20 runs).

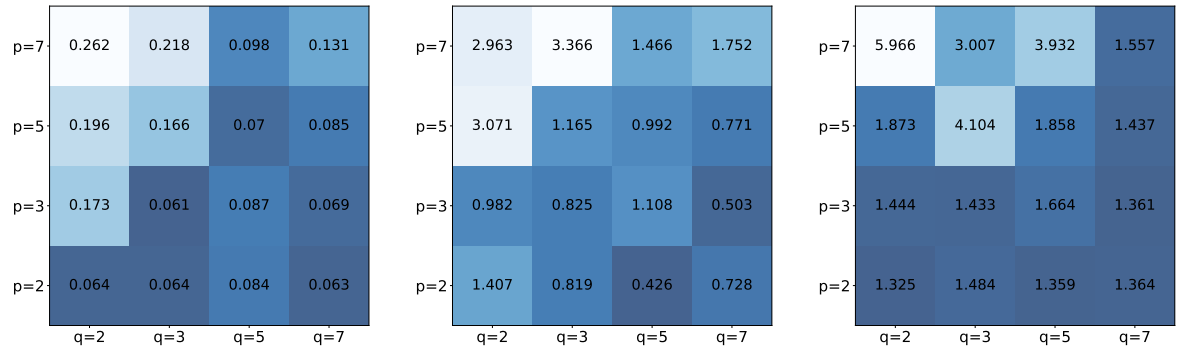
Our experiments were run in two different settings:

Asymptotic setting: in this first experiment, we assume that $\mu_\star$ is known from the experimenter (so, there is no dataset). At the end of the optimization scheme, we end up with one $\bar{\theta} \in \bar{\Theta}$. Thus, in this context, the performance of WGANs is captured by

$$\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}}) = \varepsilon_{\mathrm{optim}} + \varepsilon_{\mathrm{approx}}.$$

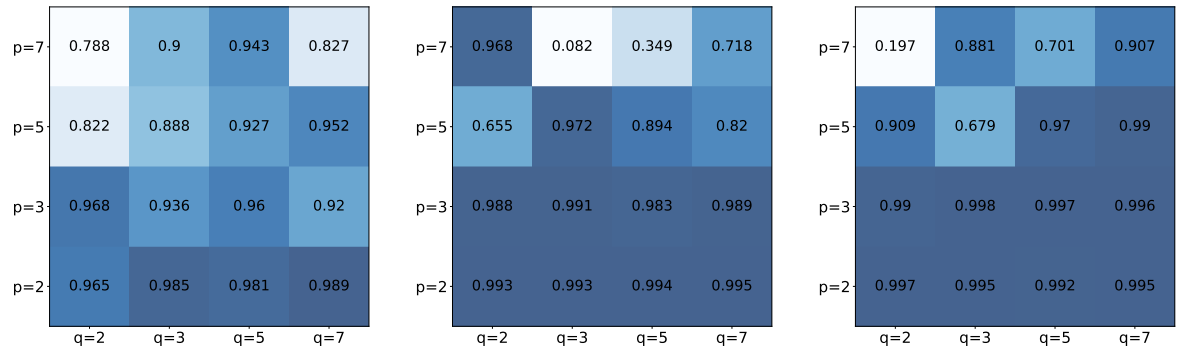
For a fixed discriminator, when increasing the generator's depth p, we expect $\varepsilon_{\mathrm{approx}}$ to decrease. Conversely, as discussed in Subsection 3.3.1, we anticipate an augmentation of $\varepsilon_{\mathrm{optim}}$, since the discriminator must now differentiate between larger classes of generative distributions. In this case, it is thus difficult to predict how $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$ behaves when p increases. On the contrary, in accordance with the results of Section 3.3, for a fixed p we expect the performance to increase with a growing q since, with larger discriminators, the pseudometric $d_{\mathscr{D}}$ is more likely to behave similarly to the Wasserstein distance d_{Lip_1}.

These intuitions are validated by Figure 3.3 and Figure 3.4 (the bluer, the better). The first one shows an approximation of $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$ computed over 5 different seeds as a function of p and q. The second one depicts the average recall of the estimator $\mu_{\bar{\theta}}$ with respect to $\mu_\star$, as a function of p and q, again computed over 5 different seeds. In both figures, we observe that for a fixed p, incrementing q leads to better results. On the opposite, for a fixed discriminator's depth q, increasing the depth p of the generator seems to deteriorate both scores (Wasserstein distance and recall). This consequently suggests that the term $\varepsilon_{\mathrm{optim}}$ dominates $\varepsilon_{\mathrm{approx}}$.



(a) $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$, $K = 1$. (b) $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$, $K = 9$. (c) $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$, $K = 25$.

Fig. 3.3 Influence of the generator's depth p and the discriminator's depth q on the maximal Wasserstein distance $\sup_{\bar{\theta}\in\bar{\Theta}} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}})$.



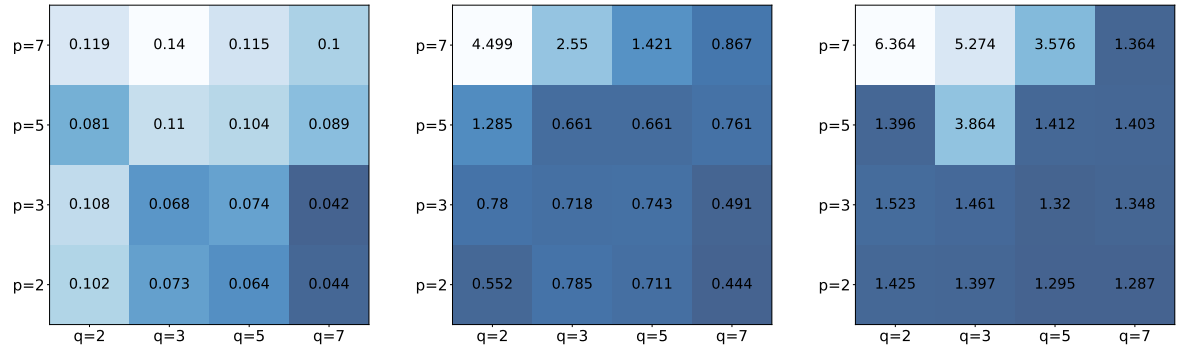
(a) Av. recall of $\mu_{\bar{\theta}}$ w.r.t. $\mu_\star$, $K = 1$. (b) Av. recall of $\mu_{\bar{\theta}}$ w.r.t. $\mu_\star$, $K = 9$. (c) Av. recall of $\mu_{\bar{\theta}}$ w.r.t. $\mu_\star$, $K = 25$.

Fig. 3.4 Influence of the generator's depth p and the discriminator's depth q on the average recall of the estimators $\mu_{\bar{\theta}}$ w.r.t. $\mu_\star$.

Finite-sample setting: in this second experiment, we consider the more realistic situation where we have at hand finite samples $X_1, \ldots, X_n$ drawn from $\mu_\star$ ($n = 5000$).

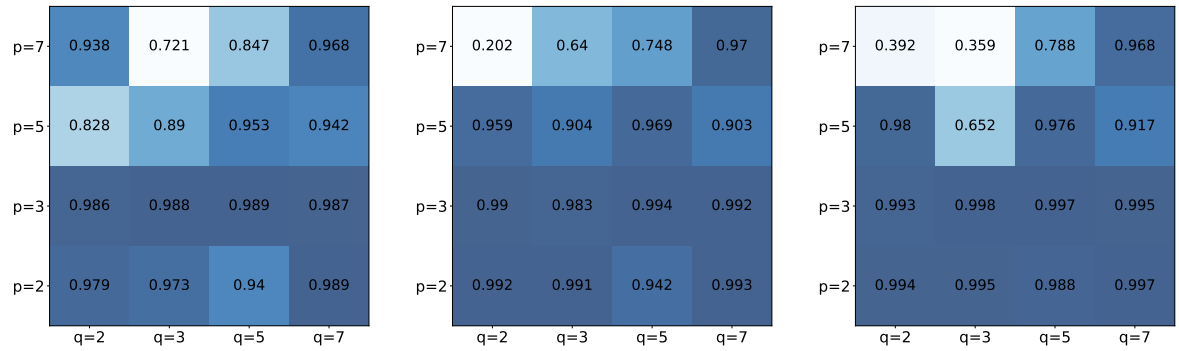
Recalling that $\sup_{\theta_n\in\hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) \leqslant \varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}} + \varepsilon_{\mathrm{approx}}$, we plot in Figure 3.5 the maximal Wasserstein distance $\sup_{\theta_n\in\hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$, and in Figure 3.6 the average recall of the estimators μ_{θ_n} with respect to $\mu_\star$, as a function of p and q. Anticipating the behavior of $\sup_{\theta_n\in\hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$ when increasing the depth q is now more involved. Indeed, according to inequality (3.4.5), which bounds $\varepsilon_{\mathrm{estim}} + \varepsilon_{\mathrm{optim}}$, a larger $\mathscr{D}$ will make $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D})$ smaller but will, on the opposite, increase $d_{\mathscr{D}}(\mu_\star, \mu_n)$. Figure 3.5 clearly shows that, for a fixed p, the maximal Wasserstein distance seems to be improved when q increases. This suggests that the term $T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D})$ dominates $d_{\mathscr{D}}(\mu_\star, \mu_n)$. Similarly to the asymptotic setting, we also make the

observation that bigger p require a higher depth q since larger class of generative distributions are more complex to discriminate.



(a) $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$, $K = 1$. (b) $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$, $K = 4$. (c) $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$, $K = 9$.

Fig. 3.5 Influence of the generator's depth p and the discriminator's depth q on the maximal Wasserstein distance $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n})$, with $n = 5000$.

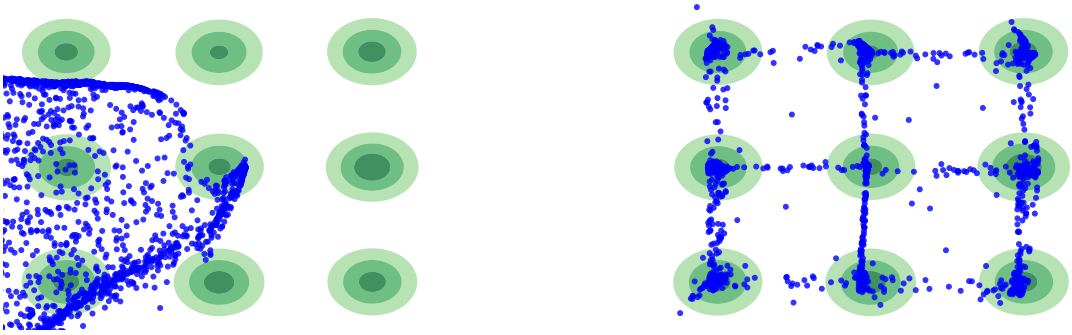


(a) Av. recall of $\mu_{\hat{\theta}_n}$ w.r.t. $\mu_\star$, $K = 1$. (b) Av. recall of $\mu_{\hat{\theta}_n}$ w.r.t. $\mu_\star$, $K = 9$. (c) Av. recall of $\mu_{\hat{\theta}_n}$ w.r.t. $\mu_\star$, $K = 25$.

Fig. 3.6 Influence of the generator's depth p and the discriminator's depth q on the average recall of the estimators μ_{θ_n} w.r.t. $\mu_\star$, with $n = 5000$.

We end this subsection by pointing out a recurring observation across different experiments. In Figure 3.4 and Figure 3.6, we notice, as already stressed, that the average recall of the estimators is prone to decrease when the generator's depth p increases. On the opposite, the average recall increases when the discriminator's depth q increases. This is interesting because the recall metric is a good proxy for a stabilized training, insofar as a high recall means the absence of mode collapse. This is also confirmed in Figure 3.7, which compares two densities: in Figure 3.7a, the discriminator has a small capacity ($q = 3$) and the generator a large capacity ($p = 7$), whereas in Figure 3.7b, the discriminator has a large capacity ($q = 7$) and

the generator a small capacity ($p = 3$). We observe that the first WGAN architecture behaves poorly compared to the second one. We therefore conclude that larger discriminators seem to bring some stability in the training of WGANS both in the asymptotic and finite sample regimes.



(a) $p = 7$ and $q = 3$.

(b) $p = 3$ and $q = 7$.

Fig. 3.7 True distribution $\mu_\star$ (mixture of $K = 9$ bivariate Gaussian densities, green circles) and 2000 data points sampled from the generator $\mu_{\bar{\theta}}$ (blue dots).

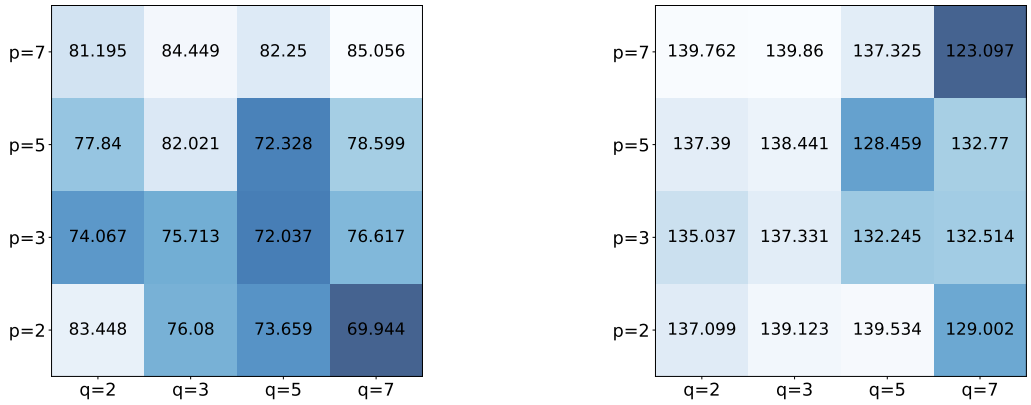
3.5.2 Real-world experiments

In this subsection, we further illustrate the impact of the generator's and the discriminator's capacities on two high-dimensional datasets, namely MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017). MNIST contains images in $\mathbb{R}^{28\times 28}$ with 10 classes representing the digits. Fashion-MNIST is a 10-class dataset of images in $\mathbb{R}^{28\times 28}$, with slightly more complex shapes than MNIST. Both datasets have a training set of 60,000 examples.

To measure the performance of WGANs when dealing with high-dimensional applications such as image generation, Brock et al. (2019) have advocated that embedding images into a feature space with a pre-trained convolutional classifier provides more meaningful information. Therefore, in order to assess the quality of the generator $\mu_{\hat{\theta}_n}$, we sample images both from the empirical measure μ_n and from the distribution $\mu_{\hat{\theta}_n}$. Then, instead of computing the Wasserstein (or recall) distance directly between these two samples, we use as a substitute their embeddings output by an external classifier and compute the Wasserstein (or recall) between the two new collections. Such a transformation is also done, for example, in Kynkäänniemi et al. (2019). Practically speaking, for any pair of images (a, b), this operation amounts to using the Euclidean distance $\|\phi(a) - \phi(b)\|$ in the Wasserstein and recall criteria, where ϕ is a pre-softmax layer of a supervised classifier, trained specifically on the datasets MNIST and Fashion-MNIST.

For these two datasets, as usual, we use generators of the form (3.2.1) and discriminators of the form (3.2.2), and plot the performance of $\mu_{\hat{\theta}_n}$ as a function of both p and q. The results

of Figure 3.8 confirm the fact that the worst results are achieved for generators with a large depth p combined with discriminators with a small depth q. They also corroborate the previous observations that larger discriminators are preferred.



(a) $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_n, \mu_{\theta_n})$, MNIST dataset.

(b) $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_n, \mu_{\theta_n})$, FMNIST dataset.

Fig. 3.8 Influence of the generator's depth p and the discriminator's depth q on the maximal Wasserstein distance $\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathrm{Lip}_1}(\mu_n, \mu_{\theta_n})$ for the MNIST and F-MNIST datasets.

Appendix 3.A Technical results

3.A.1 Proof of Lemma 3.2.1

Recall that the notation $\|\cdot\|$ (respectively, $\|\cdot\|_\infty$) means the Euclidean (respectively, the supremum) norm, with no specific mention of the underlying space on which it acts. For $(z,z') \in (\mathbb{R}^d)^2$, we have

$$\begin{aligned}
\|f_1(z) - f_1(z')\| &\leqslant \|U_1 z + b_1 - U_1 z' - b_1\| \\
&\quad \text{(since } \sigma \text{ is 1-Lipschitz)} \\
&= \|U_1(z - z')\| \\
&\leqslant \|U_1\|_2 \, \|z - z'\| \\
&\leqslant K_1 \|z - z'\| \\
&\quad \text{(by Assumption 1).}
\end{aligned}$$

Repeating this for $i = 2, \dots, p$, we thus have, for all $(z,z') \in (\mathbb{R}^d)^2$, $\|G_\theta(z) - G_\theta(z')\| \leqslant K_1^p \|z - z'\|$. We conclude that, for each $\theta \in \Theta$, the function G_θ is K_1^p-Lipschitz on $\mathbb{R}^d$.

Let us now prove that $\mathscr{D} \subseteq \mathrm{Lip}_1$. Fix $D_\alpha \in \mathscr{D}$, $\alpha \in \Lambda$. According to (3.2.2), we have, for $x \in E$, $D_\alpha(x) = f_q \circ \cdots \circ f_1(x)$, where $f_i(t) = \tilde{\sigma}(V_i t + c_i)$ for $i = 1, \dots, q-1$ ($\tilde{\sigma}$ is applied on pairs of components), and $f_q(t) = V_q t + c_q$.

Consequently, for $(x,y) \in E^2$,

$$\begin{aligned}
\|f_1(x) - f_1(y)\|_\infty &\leqslant \|V_1 x - V_1 y\|_\infty \\
&\quad \text{(since } \tilde{\sigma} \text{ is 1-Lipschitz)} \\
&= \|V_1(x - y)\|_\infty \\
&\leqslant \|V_1\|_{2,\infty} \, \|x - y\| \\
&\leqslant \|x - y\| \\
&\quad \text{(by Assumption 1).}
\end{aligned}$$

Thus,

$$\begin{aligned}\|f_2\circ f_1(x)-f_2\circ f_1(y)\|_\infty &\leqslant \|V_2 f_1(x)-V_2 f_1(y)\|_\infty\\ &\quad \text{(since } \tilde{\sigma} \text{ is 1-Lipschitz)}\\ &\leqslant \|V_2\|_\infty \, \|f_1(x)-f_1(y)\|_\infty\\ &\leqslant \|f_1(x)-f_1(y)\|_\infty\\ &\quad \text{(by Assumption 1)}\\ &\leqslant \|x-y\|.\end{aligned}$$

Repeating this, we conclude that, for each $\alpha\in\Lambda$ and all $(x,y)\in E^2$, $|D_\alpha(x)-D_\alpha(y)|\leqslant\|x-y\|$, which is the desired result.

3.A.2 Proof of Proposition 3.2.1

We first prove that the function $\Theta\ni\theta\mapsto\mu_\theta$ is continuous with respect to the weak topology in $P_1(E)$. Let G_θ and $G_{\theta'}$ be two elements of $\mathscr{G}$, with $(\theta,\theta')\in\Theta^2$. Using (3.2.1), we write $G_\theta(z)=f_p\circ\cdots\circ f_1(z)$ (respectively, $G_{\theta'}(z)=f'_p\circ\cdots\circ f'_1(z)$), where $f_i(x)=\max(U_ix+b_i,0)$ (respectively, $f'_i(x)=\max(U'_ix+b'_i,0)$) for $i=1,\dots,p-1$, and $f_p(x)=U_px+b_p$ (respectively, $f'_p(x)=U'_px+b'_p$).

Clearly, for $z\in\mathbb{R}^d$,

$$\begin{aligned}\|f_1(z)-f'_1(z)\| &\leqslant \|U_1z+b_1-U'_1z-b'_1\|\\ &\leqslant \|(U_1-U'_1)z\|+\|b_1-b'_1\|\\ &\leqslant \|U_1-U'_1\|_2\,\|z\|+\|b_1-b'_1\|\\ &\leqslant (\|z\|+1)\|\theta-\theta'\|.\end{aligned}$$

Similarly, for any $i\in\{2,\dots,p\}$ and any $x\in\mathbb{R}^{u_i}$,

$$\|f_i(x)-f'_i(x)\|\leqslant(\|x\|+1)\|\theta-\theta'\|.$$

Observe that

$$\begin{aligned}&\|G_\theta(z)-G_{\theta'}(z)\|\\ &=\|f_p\circ\cdots\circ f_1(z)-f'_p\circ\cdots\circ f'_1(z)\|\\ &\leqslant\|f_p\circ\cdots\circ f_1(z)-f_p\circ\cdots\circ f_2\circ f'_1(z)\|+\cdots+\|f_p\circ f'_{p-1}\circ\cdots\circ f'_1(z)-f'_p\circ\cdots\circ f'_1(z)\|.\end{aligned}$$

As in the proof of Lemma 3.2.1, one shows that for any $i \in \{1,\dots,p\}$, the function $f_p \circ \cdots \circ f_i$ is K_1^{p-i+1}-Lipschitz with respect to the Euclidean norm. Therefore,

$$\begin{aligned}
&\|G_\theta(z) - G_{\theta'}(z)\| \\
&\quad \leqslant K_1^{p-1}\|f_1(z) - f_1'(z)\| + \cdots + K_1^0\|f_p \circ f_{p-1}' \circ \cdots \circ f_1'(z) - f_p' \circ \cdots \circ f_1'(z)\| \\
&\quad \leqslant K_1^{p-1}(\|z\|+1)\|\theta - \theta'\| + \cdots + (\|f_{p-1}' \circ \cdots \circ f_1'(z)\| + 1)\|\theta - \theta'\| \\
&\quad \leqslant K_1^{p-1}(\|z\|+1)\|\theta - \theta'\| + \cdots + (K_1^{p-1}\|z\| + \|f_{p-1}' \circ \cdots \circ f_1'(0)\| + 1)\|\theta - \theta'\|.
\end{aligned}$$

Using the architecture of neural networks in (3.2.1), a quick check shows that, for each $i \in \{1,\dots,p\}$,

$$\|f_i' \circ \cdots \circ f_1'(0)\| \leqslant \sum_{k=1}^{i} K_1^k.$$

We are led to

$$\|G_\theta(z) - G_{\theta'}(z)\| = (\ell_1\|z\| + \ell_2)\|\theta - \theta'\|, \tag{3.A.1}$$

where

$$\ell_1 = pK_1^{p-1} \quad \text{and} \quad \ell_2 = \sum_{i=1}^{p-1} K_1^{p-(i+1)} \sum_{k=1}^{i} K_1^k + \sum_{i=0}^{p-1} K_1^i.$$

Denoting by ν the probability distribution of the sub-Gaussian random variable Z, we note that $\int_{\mathbb{R}^d}(\ell_1\|z\| + \ell_2)\nu(\mathrm{d}z) < \infty$. Now, let (θ_k) be a sequence in Θ converging to $\theta \in \Theta$ with respect to the Euclidean norm. Clearly, for a given $z \in \mathbb{R}^d$, by continuity of the function $\theta \mapsto G_\theta(z)$, we have $\lim_{k\to\infty} G_{\theta_k}(z) = G_\theta(z)$ and, for any $\varphi \in C_b(E)$, $\lim_{k\to\infty} \varphi(G_{\theta_k}(z)) = \varphi(G_\theta(z))$. Thus, by the dominated convergence theorem,

$$\lim_{k\to\infty} \int_E \varphi(x)\mu_{\theta_k}(\mathrm{d}x) = \lim_{k\to\infty} \int_{\mathbb{R}^d} \varphi(G_{\theta_k}(z))\nu(\mathrm{d}z) = \int_{\mathbb{R}^d} \varphi\big(G_\theta(z)\big)\nu(\mathrm{d}z) = \int_E \varphi(x)\mu_\theta(\mathrm{d}x). \tag{3.A.2}$$

This shows that the sequence (μ_{θ_k}) converges weakly to μ_θ. Besides, for an arbitrary x_0 in E, we have

$$\begin{aligned}
&\limsup_{k\to\infty} \int_E \|x_0 - x\| \mu_{\theta_k}(\mathrm{d}x) \\
&\quad = \limsup_{k\to\infty} \int_{\mathbb{R}^d} \|x_0 - G_{\theta_k}(z)\| \nu(\mathrm{d}z) \\
&\quad \leqslant \limsup_{k\to\infty} \int_{\mathbb{R}^d} \big(\|G_{\theta_k}(z) - G_\theta(z)\| + \|G_\theta(z) - x_0\| \big) \nu(\mathrm{d}z) \\
&\quad \leqslant \limsup_{k\to\infty} \int_{\mathbb{R}^d} (\ell_1 \|z\| + \ell_2) \|\theta_k - \theta\| \nu(\mathrm{d}z) + \int_{\mathbb{R}^d} \|G_\theta(z) - x_0\| \nu(\mathrm{d}z) \\
&\qquad \text{(by inequality (3.A.1)).}
\end{aligned}$$

Consequently,

$$\limsup_{k\to\infty} \int_E \|x_0 - x\| \mu_{\theta_k}(\mathrm{d}x) \leqslant \int_{\mathbb{R}^d} \|G_\theta(z) - x_0\| \nu(\mathrm{d}z) = \int_E \|x_0 - x\| \mu_\theta(\mathrm{d}x).$$

One proves with similar arguments that

$$\liminf_{k\to\infty} \int_E \|x_0 - x\| \mu_{\theta_k}(\mathrm{d}x) \geqslant \int_E \|x_0 - x\| \mu_\theta(\mathrm{d}x).$$

Therefore, putting all the pieces together, we conclude that

$$\lim_{k\to\infty} \int_E \|x_0 - x\| \mu_{\theta_k}(\mathrm{d}x) = \int_E \|x_0 - x\| \mu_\theta(\mathrm{d}x).$$

This, together with (3.A.2), shows that the sequence (μ_{θ_k}) converges weakly to μ_θ in $P_1(E)$, and, in turn, that the function $\Theta \ni \theta \mapsto \mu_\theta$ is continuous with respect to the weak topology in $P_1(E)$, as desired.

The second assertion of the proposition follows upon noting that $\mathscr{P}$ is the image of the compact set Θ by a continuous function.

3.A.3 Proof of Proposition 3.2.2

To show the first statement, we are to exhibit a specific discriminator, say $\mathscr{D}_{\max}$, such that, for all $(\mu, \nu) \in (\mathscr{P} \cup \{\mu_\star\})^2$, the identity $d_{\mathscr{D}_{\max}}(\mu, \nu) = 0$ implies $\mu = \nu$.

Let $\varepsilon > 0$. According to Proposition 3.2.1, under Assumption 1, $\mathscr{P}$ is a compact subset of $P_1(E)$ with respect to the weak topology in $P_1(E)$. Let $x_0 \in E$ be arbitrary. For any $\mu \in \mathscr{P}$ there exists a compact $K_\mu \subseteq E$ such that $\int_{K_\mu^\complement} \|x_0 - x\| \mu(\mathrm{d}x) \leqslant \varepsilon/4$. Also, for any such K_μ, the

function $P_1(E) \ni \rho \mapsto \int_{K_\mu^\complement} \|x_0 - x\| \rho(\mathrm{d}x)$ is continuous. Therefore, there exists an open set $U_\mu \subseteq P_1(E)$ containing μ such that, for any $\rho \in U_\mu$, $\int_{K_\mu^\complement} \|x_0 - x\| \rho(\mathrm{d}x) \leqslant \varepsilon/2$.

The collection of open sets $\{U_\mu : \mu \in \mathscr{P}\}$ forms an open cover of $\mathscr{P}$, from which we can extract, by compactness, a finite subcover $U_{\mu_1}, \ldots, U_{\mu_n}$. Letting $K_1 = \cup_{i=1}^n K_{\mu_i}$, we deduce that, for all $\mu \in \mathscr{P}$, $\int_{K_1^\complement} \|x_0 - x\| \mu(\mathrm{d}x) \leqslant \varepsilon/2$. We conclude that there exists a compact $K \subseteq E$ and $x_0 \in K$ such that, for any $\mu \in \mathscr{P} \cup \{\mu_\star\}$,

$$\int_{K^\complement} ||x_0 - x|| \mu(\mathrm{d}x) \leqslant \varepsilon/2.$$

By Arzelà-Ascoli theorem, it is easy to see that $\mathrm{Lip}_1(K)$, the set of 1-Lipschitz real-valued functions on K, is compact with respect to the uniform norm $\|\cdot\|_\infty$ on K. Let $\{f_1, \ldots, f_{\mathscr{N}_\varepsilon}\}$ denote an ε-covering of $\mathrm{Lip}_1(K)$. According to Anil et al. (2019, Theorem 3), for each $k = 1, \ldots, \mathscr{N}_\varepsilon$ there exists under Assumption 1 a discriminator $\mathscr{D}_k$ of the form (3.2.2) such that

$$\inf_{g \in \mathscr{D}_k} \|f_k - g \mathbb{1}_K\|_\infty \leqslant \varepsilon.$$

Since the discriminative classes of functions use GroupSort activations, one can find a neural network of the form (3.2.2) satisfying Assumption 1, say $\mathscr{D}_{\max}$, such that, for all $k \in \{1, \ldots, \mathscr{N}_\varepsilon\}$, $\mathscr{D}_k \subseteq \mathscr{D}_{\max}$. Consequently, for any $f \in \mathrm{Lip}_1(K)$, letting $k_0 \in \operatorname*{arg\,min}_{k \in \{1, \ldots, \mathscr{N}_\varepsilon\}} \|f - f_k\|_\infty$, we have

$$\inf_{g \in \mathscr{D}_{\max}} \|f - g \mathbb{1}_K\|_\infty \leqslant \|f - f_{k_0}\|_\infty + \inf_{g \in \mathscr{D}_{\max}} \|f_{k_0} - g \mathbb{1}_K\|_\infty \leqslant 2\varepsilon.$$

Now, let $(\mu, \nu) \in (\mathscr{P} \bigcup \{\mu_\star\})^2$ be such that $d_{\mathscr{D}_{\max}}(\mu, \nu) = 0$, i.e., $\sup_{f \in \mathscr{D}_{\max}} |\mathbb{E}_\mu f - \mathbb{E}_\nu f| = 0$. Let $f^\star$ be a function in Lip_1 such that $\mathbb{E}_\mu f^\star - \mathbb{E}_\nu f^\star = d_{\mathrm{Lip}_1}(\mu, \nu)$ (such a function exists according to (3.2.6)) and, without loss of generality, such that $f^\star(x_0) = 0$. Clearly,

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu, \nu) &= \mathbb{E}_\mu f^\star - \mathbb{E}_\nu f^\star \\ &\leqslant \left| \int_K f^\star \mathrm{d}\mu - \int_K f^\star \mathrm{d}\nu \right| + \left| \int_{K^\complement} f^\star \mathrm{d}\mu - \int_{K^\complement} f^\star \mathrm{d}\nu \right| \\ &\leqslant \left| \int_K f^\star \mathrm{d}\mu - \int_K f^\star \mathrm{d}\nu \right| + \varepsilon. \end{aligned}$$

Letting $g_{f^\star} \in \mathscr{D}_{\max}$ be such that

$$\|(f^\star - g_{f^\star}) \mathbb{1}_K\|_\infty \leqslant \inf_{g \in \mathscr{D}_{\max}} \|(f^\star - g) \mathbb{1}_K\|_\infty + \varepsilon \leqslant 3\varepsilon,$$

we are thus led to

$$d_{\mathrm{Lip}_1}(\mu,\nu) \leqslant \left| \int_K (f^\star - g_{f^\star})\mathrm{d}\mu - \int_K (f^\star - g_{f^\star})\mathrm{d}\nu + \int_K g_{f^\star}\mathrm{d}\mu - \int_K g_{f^\star}\mathrm{d}\nu \right| + \varepsilon.$$

Observe, since $x_0 \in K$, that $|g_{f^\star}(x_0)| \leqslant 3\varepsilon$ and that, for any $x \in E$, $|g_{f^\star}(x)| \leqslant \|x_0 - x\| + 3\varepsilon$. Exploiting $\mathbb{E}_\mu g_{f^\star} - \mathbb{E}_\nu g_{f^\star} = 0$, we obtain

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu,\nu) &\leqslant 7\varepsilon + \left| \int_{K^\complement} g_{f^\star}\mathrm{d}\mu - \int_{K^\complement} g_{f^\star}\mathrm{d}\nu \right| \\ &\leqslant 7\varepsilon + \int_{K^\complement} \|x_0 - x\|\mu(\mathrm{d}x) + \int_{K^\complement} \|x_0 - x\|\nu(\mathrm{d}x) + 6\varepsilon \\ &\leqslant 14\varepsilon. \end{aligned}$$

Since ε is arbitrary and d_{Lip_1} is a metric on $P_1(E)$, we conclude that $\mu = \nu$, as desired.

To complete the proof, it remains to show that $d_{\mathscr{D}_{\max}}$ metrizes weak convergence in $\mathscr{P} \cup \{\mu_\star\}$. To this aim, we let (μ_k) be a sequence in $\mathscr{P} \cup \{\mu_\star\}$ and μ be a probability measure in $\mathscr{P} \cup \{\mu_\star\}$.

If (μ_k) converges weakly to μ in $P_1(E)$, then $d_{\mathrm{Lip}_1}(\mu,\mu_k) \to 0$ (Villani, 2008, Theorem 6.8), and, accordingly, $d_{\mathscr{D}_{\max}}(\mu,\mu_k) \to 0$.

Suppose, on the other hand, that $d_{\mathscr{D}_{\max}}(\mu,\mu_k) \to 0$, and fix $\varepsilon > 0$. There exists $M > 0$ such that, for all $k \geqslant M$, $d_{\mathscr{D}_{\max}}(\mu,\mu_k) \leqslant \varepsilon$. Using a similar reasoning as in the first part of the proof, it is easy to see that for any $k \geqslant M$, we have $d_{\mathrm{Lip}_1}(\mu,\mu_k) \leqslant 15\varepsilon$. Since the Wasserstein distance metrizes weak convergence in $P_1(E)$ and ε is arbitrary, we conclude that (μ_k) converges weakly to μ in $P_1(E)$.

3.A.4 Proof of Lemma 3.2.2

Using a similar reasoning as in the proof of Proposition 3.2.1, one easily checks that for all $(\alpha,\alpha') \in \Lambda^2$ and all $x \in E$,

$$\begin{aligned} |D_\alpha(x) - D_{\alpha'}(x)| &\leqslant Q^{1/2}\big(q\|x\| + K_2 \sum_{i=1}^{q-1} i + q\big)\|\alpha - \alpha'\| \\ &\leqslant Q^{1/2}\big(q\|x\| + \frac{q(q-1)K_2}{2} + q\big)\|\alpha - \alpha'\|, \end{aligned}$$

where q refers to the depth of the discriminator. Thus, since $\mathscr{D} \subset \mathrm{Lip}_1$ (by Lemma 3.2.1), we have, for all $\alpha \in \Lambda$, all $x \in E$, and any arbitrary $x_0 \in E$,

$$\begin{aligned}
|D_\alpha(x)| &\leqslant |D_\alpha(x) - D_\alpha(x_0)| + |D_\alpha(x_0)| \\
&\leqslant \|x_0 - x\| + Q^{1/2}\Big(q\|x_0\| + \frac{q(q-1)K_2}{2} + q\Big)\|\alpha\| \\
&\quad \text{(upon noting that } D_0(x_0) = 0\text{)} \\
&\leqslant \|x_0 - x\| + Q^{1/2}\Big(q\|x_0\| + \frac{q(q-1)K_2}{2} + q\Big)Q^{1/2}\max(K_2, 1),
\end{aligned}$$

where Q is the dimension of Λ. Thus, since $\mu^\star$ and the μ_θ's belong to $P_1(E)$ (by Lemma 3.2.1), we deduce that all $D_\alpha \in \mathscr{D}$ are dominated by a function independent of α and integrable with respect to $\mu_\star$ and μ_θ. In addition, for all $x \in E$, the function $\alpha \mapsto D_\alpha(x)$ is continuous on Λ. Therefore, by the dominated convergence theorem, the function $\Lambda \ni \alpha \mapsto |\mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha|$ is continuous. The conclusion follows from the compactness of the set Λ (Assumption 1).

3.A.5 Proof of Theorem 3.2.1

Let $(\theta, \theta') \in \Theta^2$, and let γ_Z be the joint distribution of the pair $(G_\theta(Z), G_{\theta'}(Z))$. We have

$$\begin{aligned}
|\xi_{\mathrm{Lip}_1}(\theta) - \xi_{\mathrm{Lip}_1}(\theta')| &= |d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta'})| \\
&\leqslant d_{\mathrm{Lip}_1}(\mu_\theta, \mu_{\theta'}) \\
&= \inf_{\gamma \in \Pi(\mu_\theta, \mu_{\theta'})} \int_{E^2} \|x - y\| \gamma(\mathrm{d}x, \mathrm{d}y),
\end{aligned}$$

where $\Pi(\mu_\theta, \mu_{\theta'})$ denotes the collection of all joint probability measures on $E \times E$ with marginals μ_θ and $\mu_{\theta'}$. Thus,

$$\begin{aligned}
|\xi_{\mathrm{Lip}_1}(\theta) - \xi_{\mathrm{Lip}_1}(\theta')| &\leqslant \int_{E^2} \|x - y\| \gamma_Z(\mathrm{d}x, \mathrm{d}y) \\
&= \int_{\mathbb{R}^d} \|G_\theta(z) - G_{\theta'}(z)\| \nu(\mathrm{d}z) \\
&\quad \text{(where } \nu \text{ is the distribution of } Z\text{)} \\
&\leqslant \|\theta - \theta'\| \int_{\mathbb{R}^d} (\ell_1 \|z\| + \ell_2) \nu(\mathrm{d}z) \\
&\quad \text{(by inequality (3.A.1)).}
\end{aligned}$$

This shows that the function $\theta \ni \Theta \mapsto \xi_{\mathrm{Lip}_1}(\theta)$ is L-Lipschitz, with $L = \int_{\mathbb{R}^d}(\ell_1\|z\| + \ell_2)\nu(\mathrm{d}z)$. For the second statement of the theorem, just note that

$$\begin{aligned}
|\xi_{\mathscr{D}}(\theta) - \xi_{\mathscr{D}}(\theta')| &= |d_{\mathscr{D}}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_{\theta'})| \\
&\leqslant d_{\mathscr{D}}(\mu_\theta, \mu_{\theta'}) \\
&\leqslant d_{\mathrm{Lip}_1}(\mu_\theta, \mu_{\theta'}) \\
&\quad \text{(since } \mathscr{D} \subseteq \mathrm{Lip}_1\text{)} \\
&\leqslant L\|\theta - \theta'\|.
\end{aligned}$$

3.A.6 Proof of Theorem 3.3.1

The proof is divided into two parts. First, we show that under Assumption 1, for all $\varepsilon > 0$ and $\theta \in \Theta$, there exists a discriminator $\mathscr{D}$ (function of ε and θ) of the form (3.2.2) such that

$$d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant 10\varepsilon.$$

Let $f^\star$ be a function in Lip_1 such that $\mathbb{E}_{\mu_\star} f^\star - \mathbb{E}_{\mu_\theta} f^\star = d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)$ (such a function exists according to (3.2.6)). We may write

$$\begin{aligned}
d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) &= \mathbb{E}_{\mu_\star} f^\star - \mathbb{E}_{\mu_\theta} f^\star - \sup_{f \in \mathscr{D}} |\mathbb{E}_{\mu_\star} f - \mathbb{E}_{\mu_\theta} f| \\
&= \mathbb{E}_{\mu_\star} f^\star - \mathbb{E}_{\mu_\theta} f^\star - \sup_{f \in \mathscr{D}} (\mathbb{E}_{\mu_\star} f - \mathbb{E}_{\mu_\theta} f) \\
&= \inf_{f \in \mathscr{D}} (\mathbb{E}_{\mu_\star} f^\star - \mathbb{E}_{\mu_\theta} f^\star - \mathbb{E}_{\mu_\star} f + \mathbb{E}_{\mu_\theta} f) \\
&= \inf_{f \in \mathscr{D}} (\mathbb{E}_{\mu_\star}(f^\star - f) - \mathbb{E}_{\mu_\theta}(f^\star - f)) \\
&\leqslant \inf_{f \in \mathscr{D}} (\mathbb{E}_{\mu_\star}|f^\star - f| + \mathbb{E}_{\mu_\theta}|f^\star - f|). \qquad (3.A.3)
\end{aligned}$$

Next, for any $f \in \mathscr{D}$ and any compact $K \subseteq E$,

$$\begin{aligned}
\mathbb{E}_{\mu_\star}|f^\star - f| &= \mathbb{E}_{\mu_\star}|f^\star - f|\mathbb{1}_K + \mathbb{E}_{\mu_\star}|f^\star - f|\mathbb{1}_{K^\complement} \\
&\leqslant \|(f^\star - f)\mathbb{1}_K\|_\infty + \mathbb{E}_{\mu_\star}|f^\star|\mathbb{1}_{K^\complement} + \mathbb{E}_{\mu_\star}|f|\mathbb{1}_{K^\complement}.
\end{aligned}$$

For the rest of the proof, we will assume, without loss of generality, that $f^\star(0) = 0$ and thus $|f^\star(x)| \leqslant |x|$. Therefore, there exists a compact set K such that $0 \in K$ and

$$\max\left(\mathbb{E}_{\mu_\star}|f^\star|\mathbb{1}_{K^\complement},\ \mathbb{E}_{\mu_\theta}|f^\star|\mathbb{1}_{K^\complement}\right) \leqslant \varepsilon.$$

Besides, according to Anil et al. (2019, Theorem 3), under Assumption 1, for any compact K, we can find a discriminator of the form (3.2.2) such that $\inf_{f\in\mathscr{D}} \|(f^\star - f)\mathbb{1}_K\|_\infty \leqslant \varepsilon$. So, choose $f \in \mathscr{D}$ such that $\|(f^\star - f)\mathbb{1}_K\|_\infty \leqslant 2\varepsilon$. For such a choice of f, we have, for any $x \in E$, $|f(x)| \leqslant |f(x) - f(0)| + |f(0)| \leqslant |x| + 2\varepsilon$, and thus, recalling that $f^\star(0) = 0$,

$$\max\left(\mathbb{E}_{\mu_\star}|f|\mathbb{1}_{K^\complement},\ \mathbb{E}_{\mu_\theta}|f|\mathbb{1}_{K^\complement}\right) \leqslant 3\varepsilon.$$

Consequently,

$$\mathbb{E}_{\mu_\star}|f^\star - f| \leqslant \|(f^\star - f)\mathbb{1}_K\|_\infty + 4\varepsilon.$$

Similarly,

$$\mathbb{E}_{\mu_\theta}|f^\star - f| \leqslant \|(f^\star - f)\mathbb{1}_K\|_\infty + 4\varepsilon.$$

Plugging the two inequalities above in (3.A.3), we obtain

$$d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant 2 \inf_{f\in\mathscr{D}} \|(f^\star - f)\mathbb{1}_K\|_\infty + 8\varepsilon.$$

We conclude that, for this choice of $\mathscr{D}$ (function of ε and θ),

$$d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant 10\varepsilon, \tag{3.A.4}$$

as desired.

For the second part of the proof, we fix $\varepsilon > 0$ and let, for each $\theta \in \Theta$ and each discriminator of the form (3.2.2),

$$\hat{\xi}_{\mathscr{D}}(\theta) = d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_\star, \mu_\theta).$$

Arguing as in the proof of Theorem 3.2.1, we see that $\hat{\xi}_{\mathscr{D}}(\theta)$ is $2L$-Lipschitz in θ, where $L = \int_{\mathbb{R}^d}(\ell_1\|z\| + \ell_2)\nu(\mathrm{d}z)$ and ν is the probability distribution of Z.

Now, let $\{\theta_1, \ldots, \theta_{\mathscr{N}_\varepsilon}\}$ be an ε-covering of the compact set Θ, i.e., for each $\theta \in \Theta$, there exists $k \in \{1, \ldots, \mathscr{N}_\varepsilon\}$ such that $\|\theta - \theta_k\| \leqslant \varepsilon$. According to (3.A.4), for each such k, there exists a discriminator $\mathscr{D}_k$ such that $\hat{\xi}_{\mathscr{D}_k}(\theta_k) \leqslant 6\varepsilon$. Since the discriminative classes of functions use GroupSort activation functions, one can find a neural network of the form (3.2.2) satisfying Assumption 1, say $\mathscr{D}_{\max}$, such that, for all $k \in \{1, \ldots, \mathscr{N}_\varepsilon\}$, $\mathscr{D}_k \subseteq \mathscr{D}_{\max}$. Clearly, $\hat{\xi}_{\mathscr{D}_{\max}}(\theta)$ is $2L$-Lipschitz, and, for all $k \in \{1, \ldots, \mathscr{N}_\varepsilon\}$, $\hat{\xi}_{\mathscr{D}_{\max}}(\theta_k) \leqslant 6\varepsilon$. Hence, for all $\theta \in \Theta$, letting

$$\hat{k} \in \operatorname*{arg\,min}_{k\in\{1,\ldots,\mathscr{N}_\varepsilon\}} \|\theta - \theta_k\|,$$

we have

$$\hat{\xi}_{\mathscr{D}_{\max}}(\theta) \leqslant \left|\hat{\xi}_{\mathscr{D}_{\max}}(\theta) - \hat{\xi}_{\mathscr{D}_{\max}}(\theta_{\hat{k}})\right| + \hat{\xi}_{\mathscr{D}_{\max}}(\theta_{\hat{k}}) \leqslant (2L+6)\varepsilon.$$

Therefore,

$$T_{\mathscr{P}}(\mathrm{Lip}_1, \mathscr{D}_{\max}) = \sup_{\theta\in\Theta}\left[d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - d_{\mathscr{D}_{\max}}(\mu_\star, \mu_\theta)\right] = \sup_{\theta\in\Theta} \hat{\xi}_{\mathscr{D}_{\max}}(\theta) \leqslant (2L+6)\varepsilon.$$

We have just proved that, for all $\varepsilon > 0$, there exists a discriminator $\mathscr{D}_{\max}$ of the form (3.2.2) and a positive constant c (independent of ε) such that

$$T_{\mathscr{P}}(\mathrm{Lip}_1, \mathscr{D}_{\max}) \leqslant c\varepsilon.$$

This is the desired result.

3.A.7 Proof of Proposition 3.3.1

Let us assume that the statement is not true. If so, there exists $\varepsilon > 0$ such that, for all $\delta > 0$, there exists $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$ satisfying $d(\theta, \bar{\Theta}) > \varepsilon$. Consider $\delta_n = 1/n$, and choose a sequence of parameters (θ_n) such that

$$\theta_n \in \mathscr{M}_{d_{\mathscr{D}}}\left(\mu_\star, \frac{1}{n}\right) \quad \text{and} \quad d(\theta_n, \bar{\Theta}) > \varepsilon.$$

Since Θ is compact by Assumption 1, we can find a subsequence (θ_{φ_n}) that converges to some $\theta_{\mathrm{acc}} \in \Theta$. Thus, for all $n \geqslant 1$, we have

$$d_{\mathscr{D}}(\mu_\star, \mu_{\theta_{\varphi_n}}) \leqslant \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) + \frac{1}{n},$$

and, by continuity of the function $\Theta \ni \theta \mapsto d_{\mathscr{D}}(\mu_\star, \mu_\theta)$ (Theorem 3.2.1),

$$d_{\mathscr{D}}(\mu_\star, \theta_{\mathrm{acc}}) \leqslant \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta).$$

We conclude that θ_{acc} belongs to $\bar{\Theta}$. This contradicts the fact that $d(\theta_{\mathrm{acc}}, \bar{\Theta}) \geqslant \varepsilon$.

3.A.8 Proof of Lemma 3.3.2

Since $a = b$, according to Definition 3.3.2, there exists a continuously differentiable, strictly increasing function $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that, for all $\mu \in \mathscr{P}$,

$$d_{\mathrm{Lip}_1}(\mu_\star, \mu) = f(d_{\mathscr{D}}(\mu_\star, \mu)).$$

For $(\theta, \theta') \in \Theta^2$ we have, as f is strictly increasing,

$$d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant d_{\mathscr{D}}(\mu_\star, \mu_{\theta'}) \iff f(d_{\mathscr{D}}(\mu_\star, \mu_\theta)) \leqslant f(d_{\mathscr{D}}(\mu_\star, \mu_{\theta'})).$$

Therefore,

$$d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant d_{\mathscr{D}}(\mu_\star, \mu_{\theta'}) \iff d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \leqslant d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta'}).$$

This proves the first statement of the lemma.

Let us now show that d_{Lip_1} can be fully substituted by $d_{\mathscr{D}}$. Let $\varepsilon > 0$. Then, for $\delta > 0$ (function of ε, to be chosen later) and $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$, we have

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) &= f(d_{\mathscr{D}}(\mu_\star, \mu_\theta)) - \inf_{\theta \in \Theta} f(d_{\mathscr{D}}(\mu_\star, \mu_\theta)) \\ &= f(d_{\mathscr{D}}(\mu_\star, \mu_\theta)) - f(\inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta)) \\ &\leqslant \sup_{\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)} \left| f(d_{\mathscr{D}}(\mu_\star, \mu_\theta)) - f(\inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta)) \right|. \end{aligned}$$

According to Theorem 3.2.1, there exists a nonnegative constant c such that for any $\theta \in \Theta$, $d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant c$. Therefore, using the definition of $\mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$ and the fact that f is continuously differentiable, we are led to

$$d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta) \leqslant \delta \sup_{x \in [0,c]} \left| \frac{\partial f(x)}{\partial x} \right|.$$

The conclusion follows by choosing δ such that $\delta \sup_{x \in [0,c]} |\frac{\partial f(x)}{\partial x}| \leqslant \varepsilon$.

3.A.9 Proof of Proposition 3.3.2

Let $\delta \in (0,1)$ and $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_\star, \delta)$, i.e., $d_{\mathscr{D}}(\mu_\star, \mu_\theta) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \leqslant \delta$. As d_{Lip_1} is monotonously equivalent to $d_{\mathscr{D}}$, there exists a continuously differentiable, strictly increasing

function $f : \mathbb{R}^+ \to \mathbb{R}^+$ and $(a,b) \in (\mathbb{R}_+^\star)^2$ such that

$$\forall \mu \in \mathscr{P},\ af(d_{\mathscr{D}}(\mu_\star,\mu)) \leqslant d_{\mathrm{Lip}_1}(\mu_\star,\mu) \leqslant bf(d_{\mathscr{D}}(\mu_\star,\mu)).$$

So,

$$\begin{aligned} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) &\leqslant bf(\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta)+\delta) \\ &\leqslant bf(\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta))+O(\delta). \end{aligned}$$

Also,

$$\inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) \geqslant af(\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta)).$$

Therefore,

$$d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) \leqslant (b-a)f(\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta))+O(\delta).$$

3.A.10 Proof of Lemma 3.3.3

Let $f : \mathbb{R}^D \to \mathbb{R}$ be in $\mathrm{AFF}\cap\mathrm{Lip}_1$. It is of the form $f(x) = x\cdot u + b$, where $u = (u_1,\ldots,u_D)$, $b \in \mathbb{R}$, and $\|u\| \leqslant 1$. Our objective is to prove that there exists a discriminator of the form (3.2.2) with $q = 2$ and $v_1 = 2$ that contains the function f. To see this, define $V_1 \in \mathscr{M}_{(2,D)}$ and the offset vector $c_1 \in \mathscr{M}_{(2,1)}$ as

$$V_1 = \begin{bmatrix} u_1 & \cdots & u_D \\ u_1 & \cdots & u_D \end{bmatrix} \quad \text{and} \quad c_1 = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Letting $V_2 \in \mathscr{M}_{(1,2)}$, $c_2 \in \mathscr{M}_{(1,1)}$ be

$$V_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad c_2 = \begin{bmatrix} b \end{bmatrix},$$

we readily obtain $V_2\tilde{\sigma}(V_1x + c_1) + c_2 = f(x)$. Besides, it is easy to verify that $\|V_1\|_{2,\infty} \leqslant 1$.

3.A.11 Proof of Lemma 3.3.4

Let μ and ν be two probability measures in $P_1(E)$ with supports S_μ and S_ν satisfying the conditions of the lemma. Let π be an optimal coupling between μ and ν, and let (X,Y) be a

random pair with distribution π such that

$$d_{\mathrm{Lip}_1}(\mu,\nu) = \mathbb{E}\|X-Y\|.$$

Clearly, any function $f_0 \in \mathrm{Lip}_1$ satisfying $f_0(X) - f_0(Y) = \|X-Y\|$ almost surely will be such that

$$d_{\mathrm{Lip}_1}(\mu,\nu) = |\mathbb{E}_\mu f_0 - \mathbb{E}_\nu f_0|.$$

The proof will be achieved if we show that such a function f_0 exists and that it may be chosen linear. Since S_μ and S_ν are disjoint and convex, we can find a unit vector u of $\mathbb{R}^D$ included in the line containing both S_μ and S_ν such that $(x_0-y_0)\cdot u > 0$, where (x_0,y_0) is an arbitrary pair of $S_\mu \times S_\nu$. Letting $f_0(x) = x\cdot u$ ($x \in E$), we have, for all $(x,y) \in S_\mu \times S_\nu$, $f_0(x) - f_0(y) = (x-y)\cdot u = \|x-y\|$. Since f_0 is a linear and 1-Lipschitz function on E, this concludes the proof.

3.A.12 Proof of Lemma 3.3.5

For any pair of probability measures (μ,ν) on E with finite moment of order 2, we let $W_2(\mu,\nu)$ be the Wasserstein distance of order 2 between μ and ν. Recall (Villani, 2008, Definition 6.1) that

$$W_2(\mu,\nu) = \Big(\inf_{\pi\in\Pi(\mu,\nu)} \int_{E\times E} \|x-y\|^2 \pi(\mathrm{d}x,\mathrm{d}y)\Big)^{1/2},$$

where $\Pi(\mu,\nu)$ denotes the collection of all joint probability measures on $E\times E$ with marginals μ and ν. By Jensen's inequality,

$$d_{\mathrm{Lip}_1}(\mu,\nu) = W_1(\mu,\nu) \leqslant W_2(\mu,\nu).$$

Let $\Sigma \in \mathscr{M}_{(D,D)}$ be a positive semi-definite matrix, and let μ be Gaussian $\mathscr{N}(m_1,\Sigma)$ and ν be Gaussian $\mathscr{N}(m_2,\Sigma)$. Denoting by (X,Y) a random pair with marginal distributions μ and ν such that

$$\mathbb{E}\|X-Y\| = W_1(\mu,\nu),$$

we have

$$\|m_1-m_2\| = \|\mathbb{E}(X-Y)\| \leqslant \mathbb{E}\|X-Y\| = W_1(\mu,\nu) \leqslant W_2(\mu,\nu) = \|m_1-m_2\|,$$

where the last equality follows from Givens and Shortt (1984, Proposition 7). Thus, $d_{\mathrm{Lip}_1}(\mu,\nu) = \|m_1 - m_2\|$. The proof will be finished if we show that

$$d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu) \geqslant \|m_1 - m_2\|.$$

To see this, consider the linear and 1-Lipschitz function $f : E \ni x \mapsto x \cdot \frac{(m_1-m_2)}{\|m_1-m_2\|}$ (with the convention $0 \times \infty = 0$), and note that

$$\begin{aligned} d_{\mathrm{AFF}\cap\mathrm{Lip}_1}(\mu,\nu) &\geqslant \left| \int_E x \cdot \frac{(m_1-m_2)}{\|m_1-m_2\|}\mu(\mathrm{d}x) - \int_E y \cdot \frac{(m_1-m_2)}{\|m_1-m_2\|}\nu(\mathrm{d}y) \right| \\ &= \left| \int_E x \cdot \frac{(m_1-m_2)}{\|m_1-m_2\|}\mu(\mathrm{d}x) - \int_E (x - m_1 + m_2) \cdot \frac{(m_1-m_2)}{\|m_1-m_2\|}\mu(\mathrm{d}x) \right| \\ &= \|m_1 - m_2\|. \end{aligned}$$

3.A.13 Proof of Proposition 3.3.3

Let $\varepsilon > 0$, and let μ and ν be two probability measures in $P_1(E)$ with compact supports S_μ and S_ν such that $\max(\mathrm{diam}(S_\mu), \mathrm{diam}(S_\nu)) \leqslant \varepsilon d(S_\mu, S_\nu)$. Throughout the proof, it is assumed that $d(S_\mu, S_\nu) > 0$, otherwise the result is immediate. Let π be an optimal coupling between μ and ν, and let (X,Y) be a random pair with distribution π such that

$$d_{\mathrm{Lip}_1}(\mu,\nu) = \mathbb{E}\|X - Y\|.$$

Any function $f_0 \in \mathrm{Lip}_1$ satisfying $\|X - Y\| \leqslant (1+2\varepsilon)(f_0(X) - f_0(Y))$ almost surely will be such that

$$d_{\mathrm{Lip}_1}(\mu,\nu) \leqslant (1+2\varepsilon)|\mathbb{E}_\mu f_0 - \mathbb{E}_\nu f_0|.$$

Thus, the proof will be completed if we show that such a function f_0 exists and that it may be chosen affine.

Since S_μ and S_ν are compact, there exists $(x^\star, y^\star) \in S_\mu \times S_\nu$ such that $\|x^\star - y^\star\| = d(S_\mu, S_\nu)$. By the hyperplane separation theorem, there exists a hyperplane $\mathscr{H}$ orthogonal to the unit vector $u = \frac{x^\star - y^\star}{\|x^\star - y^\star\|}$ such that $d(x^\star, \mathscr{H}) = d(y^\star, \mathscr{H}) = \frac{\|x^\star - y^\star\|}{2}$. For any $x \in E$, we denote by $p_{\mathscr{H}}(x)$ the projection of x onto $\mathscr{H}$. We thus have $d(x, \mathscr{H}) = \|x - p_{\mathscr{H}}(x)\|$, and $\frac{x^\star + y^\star}{2} = p_{\mathscr{H}}(\frac{x^\star + y^\star}{2}) = p_{\mathscr{H}}(x^\star) = p_{\mathscr{H}}(y^\star)$. In addition, by convexity of S_μ and S_ν, for any $x \in S_\mu$, $\|x - p_{\mathscr{H}}(x)\| \geqslant \|x^\star - p_{\mathscr{H}}(x^\star)\|$. Similarly, for any $y \in S_\nu$, $\|y - p_{\mathscr{H}}(y)\| \geqslant \|y^\star - p_{\mathscr{H}}(y^\star)\|$.

Let the affine function f_0 be defined for any $x \in E$ by

$$f_0(x) = (x - p_{\mathscr{H}}(x)) \cdot u.$$

Observe that $f_0(x) = f_0(x + \frac{x^\star+y^\star}{2})$. Clearly, for any $(x,y) \in E^2$, one has

$$\begin{aligned}
|f_0(x) - f_0(y)| &= \big|f_0\big(x-y+\frac{x^\star+y^\star}{2}\big)\big| \\
&= \big|\big(\big(x-y+\frac{x^\star+y^\star}{2}\big) - p_{\mathscr{H}}\big(x-y+\frac{x^\star+y^\star}{2}\big)\big).u\big| \\
&\leqslant \big\|\big(x-y+\frac{x^\star+y^\star}{2}\big) - p_{\mathscr{H}}\big(x-y+\frac{x^\star+y^\star}{2}\big)\big\| \\
&\leqslant \big\|x-y+\frac{x^\star+y^\star}{2} - \frac{x^\star+y^\star}{2}\big\| \\
&\quad \text{(since } \frac{x^\star+y^\star}{2} \in \mathscr{H}\text{)} \\
&= \|x-y\|.
\end{aligned}$$

Thus, f_0 belongs to Lip_1. Besides, for any $(x,y) \in S_\mu \times S_\nu$, we have

$$\begin{aligned}
\|x-y\| &\leqslant \|x - p_{\mathscr{H}}(x)\| + \|p_{\mathscr{H}}(x) - p_{\mathscr{H}}(y)\| + \|p_{\mathscr{H}}(y) - y\| \\
&\leqslant (x - p_{\mathscr{H}}(x)) \cdot u - (y - p_{\mathscr{H}}(y)) \cdot u + \big\|p_{\mathscr{H}}(x) - \frac{x^\star+y^\star}{2}\big\| + \big\|p_{\mathscr{H}}(y) - \frac{x^\star+y^\star}{2}\big\| \\
&= (x - p_{\mathscr{H}}(x)) \cdot u - (y - p_{\mathscr{H}}(y)) \cdot u + \|p_{\mathscr{H}}(x) - p_{\mathscr{H}}(x^\star)\| + \|p_{\mathscr{H}}(y) - p_{\mathscr{H}}(y^\star)\|.
\end{aligned}$$

Thus,

$$\begin{aligned}
\|x-y\| &\leqslant (x - p_{\mathscr{H}}(x)) \cdot u - (y - p_{\mathscr{H}}(y)) \cdot u + 2\max(\mathrm{diam}(S_\mu), \mathrm{diam}(S_\nu)) \\
&\leqslant f_0(x) - f_0(y) + 2\varepsilon d(S_\mu, S_\nu) \\
&= f_0(x) - f_0(y) + 2\varepsilon(f_0(x^\star) - f_0(y^\star)) \\
&= f_0(x) - f_0(y) + 2\varepsilon(f_0(x^\star) - f_0(x) + f_0(x) - f_0(y) + f_0(y) - f_0(y^\star)) \\
&\leqslant (1+2\varepsilon)(f_0(x) - f_0(y)) \\
&\quad \text{(using the fact that } f_0(x^\star) - f_0(x) \leqslant 0 \text{ and } f_0(y^\star) - f_0(y) \geqslant 0\text{)}.
\end{aligned}$$

Since $f_0 \in \mathrm{Lip}_1$, we conclude that, for any $(x,y) \in S_\mu \times S_\nu$,

$$|f_0(x) - f_0(y)| \leqslant \|x-y\| \leqslant (1+2\varepsilon)(f_0(x) - f_0(y)).$$

3.A.14 Proof of Lemma 3.4.1

Using Dudley (2004, Theorem 11.4.1) and the strong law of large numbers, the sequence of empirical measures (μ_n) almost surely converges weakly in $P_1(E)$ to $\mu_\star$. Thus, we have $\lim_{n\to\infty} d_{\mathrm{Lip}_1}(\mu_\star, \mu_n) = 0$ almost surely, and so $\lim_{n\to\infty} d_{\mathscr{D}}(\mu_\star, \mu_n) = 0$ almost surely. Hence, recalling

inequality (3.4.3), we conclude that

$$\sup_{\theta_n \in \hat{\Theta}_n} d_{\mathscr{D}}(\mu_\star, \mu_{\theta_n}) - \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \to 0 \quad \text{almost surely.} \tag{3.A.5}$$

Now, fix $\varepsilon > 0$ and recall that, by our Theorem 3.2.1, the function $\Theta \ni \theta \mapsto d_{\mathrm{Lip}_1}(\mu_\star, \mu_\theta)$ is L-Lipschitz, for some $L > 0$. According to (3.A.5) and Proposition 3.3.1, almost surely, there exists an integer $N > 0$ such that, for all $n \geqslant N$, for all $\theta_n \in \hat{\Theta}_n$, the companion $\bar{\theta}_n \in \bar{\Theta}$ is such that $\|\theta_n - \bar{\theta}_n\| \leqslant \frac{\varepsilon}{L}$. We conclude by observing that $|\varepsilon_{\text{estim}}| \leqslant \sup_{\theta_n \in \hat{\Theta}_n} |d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\theta_n}) - d_{\mathrm{Lip}_1}(\mu_\star, \mu_{\bar{\theta}_n})| \leqslant L \times \frac{\varepsilon}{L}$.

3.A.15 Proof of Proposition 3.4.1

Let μ_n be the empirical measure based on n i.i.d. samples $X_1, \ldots, X_n$ distributed according to $\mu_\star$. Recall (equation (3.2.5)) that

$$d_{\mathscr{D}}(\mu_\star, \mu_n) = \sup_{\alpha \in \Lambda} |\mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_n} D_\alpha| = \sup_{\alpha \in \Lambda} \left| \mathbb{E}_{\mu_\star} D_\alpha - \frac{1}{n} \sum_{i=1}^n D_\alpha(X_i) \right|.$$

Let g be the real-valued function defined on E^n by

$$g(x_1, \ldots, x_n) = \sup_{\alpha \in \Lambda} \left| \mathbb{E}_{\mu_\star} D_\alpha - \frac{1}{n} \sum_{i=1}^n D_\alpha(x_i) \right|.$$

Observe that, for $(x_1, \ldots, x_n) \in E^n$ and $(x'_1, \ldots, x'_n) \in E^n$,

$$\begin{aligned} |g(x_1, \ldots, x_n) - g(x'_1, \ldots, x'_n)| &\leqslant \sup_{\alpha \in \Lambda} \left| \frac{1}{n} \sum_{i=1}^n D_\alpha(x_i) - \frac{1}{n} \sum_{i=1}^n D_\alpha(x'_i) \right| \\ &\leqslant \frac{1}{n} \sup_{\alpha \in \Lambda} \sum_{i=1}^n |D_\alpha(x_i) - D_\alpha(x'_i)| \\ &\leqslant \frac{1}{n} \sum_{i=1}^n \|x_i - x'_i\|. \end{aligned} \tag{3.A.6}$$

We start by examining statement (i), where $\mu_\star$ has compact support with diameter B. In this case, letting X'_i be an independent copy of X_i, we have, almost surely,

$$|g(X_1, \ldots, X_n) - g(X_1, \ldots, X'_i, \ldots, X_n)| \leqslant \frac{B}{n}.$$

An application of McDiarmid's inequality (McDiarmid, 1989) shows that for any $\eta \in (0,1)$, with probability at least $1-\eta$,

$$d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \mathbb{E} d_{\mathscr{D}}(\mu_\star, \mu_n) + B\sqrt{\frac{\log(1/\eta)}{2n}}. \tag{3.A.7}$$

Next, for each $\alpha \in \Lambda$, let Y_α denote the random variable defined by

$$Y_\alpha = \mathbb{E}_{\mu_\star} D_\alpha - \frac{1}{n}\sum_{i=1}^n D_\alpha(X_i).$$

Using a similar reasoning as in the proof of Proposition 3.2.1, one shows that for any $(\alpha, \alpha') \in \Lambda^2$ and any $x \in E$,

$$|D_\alpha(x) - D_{\alpha'}(x)| \leqslant Q^{1/2}\big(q\|x\| + \frac{q(q-1)K_2}{2} + q\big)\|\alpha - \alpha'\|,$$

where we recall that q is the depth of the discriminator. Since $\mu_\star$ has compact support,

$$\ell = \int_E Q^{1/2}\big(q\|x\| + \frac{q(q-1)K_2}{2} + q\big)\mu_\star(\mathrm{d}x) < \infty.$$

Observe that

$$|Y_\alpha - Y_{\alpha'}| \leqslant \frac{1}{n}\|\alpha - \alpha'\|\ |\xi(n)|,$$

where

$$\xi_n = \sum_{i=1}^n Q^{1/2}\big(\ell + q\|X_i\| + \frac{q(q-1)K_2}{2} + q\big).$$

Thus, using Vershynin (2018, Proposition 2.5.2), there exists a positive constant $c = O(qQ^{1/2}(D^{1/2}+q))$ such that, for all $\lambda \in \mathbb{R}$,

$$\mathbb{E} e^{\lambda(Y_\alpha - Y_{\alpha'})} \leqslant \mathbb{E} e^{\lambda \frac{1}{n}\|\alpha-\alpha'\|\ |\xi_n|} \leqslant e^{c^2 \frac{1}{n}\|\alpha-\alpha'\|^2\lambda^2}.$$

We conclude that the process (Y_α) is sub-Gaussian (van Handel, 2016, Definition 5.20) for the distance $d(\alpha,\alpha') = \frac{c\|\alpha-\alpha'\|}{\sqrt{n}}$. Therefore, using van Handel (2016, Corollary 5.25), we have

$$\mathbb{E} d_{\mathscr{D}}(\mu_\star, \mu_n) = \mathbb{E}\sup_{\alpha\in\Lambda}\Big|\mathbb{E}_{\mu_\star} D_\alpha - \frac{1}{n}\sum_{i=1}^n D_\alpha(X_i)\Big| \leqslant \frac{12c}{\sqrt{n}}\int_0^\infty \sqrt{\log \mathscr{N}(\Lambda, \|\cdot\|, u)}\mathrm{d}u,$$

where $\mathscr{N}(\Lambda, \|\cdot\|, u)$ is the u-covering number of Λ for the norm $\|\cdot\|$. Since Λ is bounded, there exists $r > 0$ such that $\mathscr{N}(\Lambda, \|\cdot\|, u) = 1$ for $u \geqslant rQ^{1/2}$ and

$$\mathscr{N}(\Lambda, \|\cdot\|, u) \leq \left(\frac{rQ^{1/2}}{u}\right)^{Q} \quad \text{for } u < rQ^{1/2}.$$

Thus,

$$\mathbb{E}d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \frac{c_1}{\sqrt{n}}$$

for some positive constant $c_1 = O(qQ^{3/2}(D^{1/2} + q))$. Combining this inequality with (3.A.7) shows the first statement of the lemma.

We now turn to the more general situation (statement (ii)) where $\mu^\star$ is γ sub-Gaussian. According to inequality (3.A.6), the function g is $\frac{1}{n}$-Lipschitz with respect to the 1-norm on E^n. Therefore, by combining Kontorovich (2014, Theorem 1) and Vershynin (2018, Proposition 2.5.2), we have that for any $\eta \in (0,1)$, with probability at least $1-\eta$,

$$d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \mathbb{E}d_{\mathscr{D}}(\mu_\star, \mu_n) + 8\gamma\sqrt{eD}\sqrt{\frac{\log(1/\eta)}{n}}. \tag{3.A.8}$$

As in the first part of the proof, we let

$$Y_\alpha = \mathbb{E}_{\mu_\star} D_\alpha - \frac{1}{n}\sum_{i=1}^{n} D_\alpha(X_i),$$

and recall that for any $(\alpha, \alpha') \in \Lambda^2$ and any $x \in E$,

$$|D_\alpha(x) - D_{\alpha'}(x)| \leqslant Q^{1/2}\big(q\|x\| + \frac{q(q-1)K_2}{2} + q\big)\|\alpha - \alpha'\|.$$

Since $\mu_\star$ is sub-Gaussian, we have (see, e.g., Jin et al., 2019, Lemma 1),

$$\ell = \int_E Q^{1/2}\big(q\|x\| + \frac{q(q-1)K_2}{2} + q\big)\mu_\star(\mathrm{d}x) < \infty.$$

Thus,

$$|Y_\alpha - Y_{\alpha'}| \leqslant \frac{1}{n}\|\alpha - \alpha'\|\ |\xi(n)|,$$

where

$$\xi_n = \sum_{i=1}^{n} Q^{1/2}\big(\ell + q\|X_i\| + \frac{q(q-1)K_2}{2} + q\big).$$

According to Jin et al. (2019, Lemma 1), the real-valued random variable ξ_n is sub-Gaussian. We obtain that, for some positive constant $c_2 = O(qQ^{3/2}(D^{1/2}+q))$,

$$\mathbb{E}d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \frac{c_2}{\sqrt{n}},$$

and the conclusion follows by combining this inequality with (3.A.8).

3.A.16 Proof of Theorem 3.4.1

Let $\varepsilon > 0$ and $\eta \in (0,1)$. According to Theorem 3.3.1, there exists a discriminator $\mathscr{D}$ of the form (3.2.2) (i.e., a collection of neural networks) such that

$$T_{\mathscr{D}}(\mathrm{Lip}_1, \mathscr{D}) \leqslant \varepsilon.$$

We only prove statement (i) since both proofs are similar. In this case, according to Proposition 3.4.1, there exists a constant $c_1 > 0$ such that, with probability at least $1-\eta$,

$$d_{\mathscr{D}}(\mu_\star, \mu_n) \leqslant \frac{c_1}{\sqrt{n}} + B\sqrt{\frac{\log(1/\eta)}{2n}}.$$

Therefore, using inequality (3.4.5), we have, with probability at least $1-\eta$,

$$0 \leqslant \varepsilon_{\text{estim}} + \varepsilon_{\text{optim}} \leqslant 2\varepsilon + \frac{2c_1}{\sqrt{n}} + 2B\sqrt{\frac{\log(1/\eta)}{2n}}.$$

3.A.17 Proof of Proposition 3.4.2

Observe that, for $\theta \in \Theta$,

$$\begin{aligned}
0 &\leqslant d_{\mathscr{D}}(\mu_\star, \mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \\
&= d_{\mathscr{D}}(\mu_\star, \mu_\theta) - d_{\mathscr{D}}(\mu_n, \mu_\theta) + d_{\mathscr{D}}(\mu_n, \mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n, \mu_\theta) \\
&\quad + \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n, \mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta) \\
&\leqslant d_{\mathscr{D}}(\mu_\star, \mu_n) + d_{\mathscr{D}}(\mu_n, \mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n, \mu_\theta) + d_{\mathscr{D}}(\mu_\star, \mu_n) \\
&= 2d_{\mathscr{D}}(\mu_\star, \mu_n) + d_{\mathscr{D}}(\mu_n, \mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n, \mu_\theta),
\end{aligned}$$

where we used respectively the triangle inequality and

$$|\inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta)| \leqslant \sup_{\theta\in\Theta} |d_{\mathscr{D}}(\mu_\star,\mu_\theta) - d_{\mathscr{D}}(\mu_n,\mu_\theta)| \leqslant d_{\mathscr{D}}(\mu_\star,\mu_n).$$

Thus, assuming that $T_{\mathscr{D}}(\mathrm{Lip}_1,\mathscr{D}) \leqslant \varepsilon$, we have

$$\begin{aligned}
0 &\leqslant d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) \\
&\leqslant d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) - d_{\mathscr{D}}(\mu_\star,\mu_\theta) + d_{\mathscr{D}}(\mu_\star,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta) \\
&\leqslant T_{\mathscr{D}}(\mathrm{Lip}_1,\mathscr{D}) + d_{\mathscr{D}}(\mu_\star,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_\star,\mu_\theta) \\
&\leqslant \varepsilon + 2d_{\mathscr{D}}(\mu_\star,\mu_n) + d_{\mathscr{D}}(\mu_n,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta). \qquad (3.A.9)
\end{aligned}$$

Let $\delta > 0$ and $\theta \in \mathscr{M}_{d_{\mathscr{D}}}(\mu_n,\delta/2)$, that is,

$$d_{\mathscr{D}}(\mu_n,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathscr{D}}(\mu_n,\mu_\theta) \leqslant \delta/2.$$

For $\eta \in (0,1)$, we know from the second statement of Proposition 3.4.1 that there exists $N \in \mathbb{N}^\star$ such that, for all $n \geqslant N$, $2d_{\mathscr{D}}(\mu_\star,\mu_n) \leqslant \delta/2$ with probability at least $1-\eta$. Therefore, we conclude from (3.A.9) that for $n \geqslant N$, with probability at least $1-\eta$,

$$d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) - \inf_{\theta\in\Theta} d_{\mathrm{Lip}_1}(\mu_\star,\mu_\theta) \leqslant \varepsilon + \delta.$$

Chapter 4

Approximating Lipschitz continuous functions with GroupSort neural networks

Abstract

Recent advances in adversarial attacks and Wasserstein GANs have advocated for use of neural networks with restricted Lipschitz constants. Motivated by these observations, we study the recently introduced GroupSort neural networks, with constraints on the weights, and make a theoretical step towards a better understanding of their expressive power. We show in particular how these networks can represent any Lipschitz continuous piecewise linear functions. We also prove that they are well-suited for approximating Lipschitz continuous functions and exhibit upper bounds on both the depth and size. To conclude, the efficiency of GroupSort networks compared with more standard ReLU networks is illustrated in a set of synthetic experiments.

Contents

4.1 Introduction **112**
4.2 Mathematical context **113**
4.3 Learning functions with a grouping size 2 . . . . . **115**
4.4 Impact of the grouping size **120**
4.5 Experiments **122**
4.6 Conclusion **127**
Appendix 4.A Technical results **127**
Appendix 4.B Complementary experiments **134**

4.1 Introduction

In the past few years, developments in deep learning have highlighted the benefits of operating neural networks with restricted Lipschitz constants. An important illustration is provided by robust machine learning, where networks with large Lipschitz constants are prone to be more sensitive to adversarial attacks, in the sense that small perturbations of the inputs can lead to significant misclassification errors (e.g., Goodfellow et al., 2015). In order to circumvent these limitations, Gao et al. (2017), Esfahani and Kuhn (2018), and Blanchet et al. (2019) studied a new regularization scheme based on penalizing the gradients of the networks. Constrained neural networks also play a key role in the different but not less important domain of Wasserstein GANs (Arjovsky et al., 2017), which take advantage of the dual form of the 1-Wasserstein distance expressed as a supremum over the set of 1-Lipschitz functions (Villani, 2008). This formulation has been shown to bring training stability and is empirically efficient (Gulrajani et al., 2017). In this context, many different ways have been explored to restrict the Lipschitz constants of the discriminator. One possibility is to clip their weights, as advocated by Arjovsky et al. (2017). Other solutions involve enforcing a gradient penalty (Gulrajani et al., 2017) or penalizing norms of the matrices of the weights (Miyato et al., 2018).

However, all of these operations are delicate and may significantly affect the expressive power of the neural networks. For example, Huster et al. (2018) show that ReLU neural networks with constraints on the weights cannot represent even the simplest functions, such as the absolute value. In fact, little is known regarding the expressive power of such restricted networks, since most studies interested in the expressiveness of neural networks (e.g., Hornik et al., 1989; Cybenko, 1989; Raghu et al., 2017) do not take into account eventual constraints on their architectures. As far as we know, the most recent attempt to tackle this issue is by Anil et al. (2019). These authors exhibit a family of neural networks, with constraints on the weights, which is dense in the set of Lipschitz continuous functions on a compact set. To show this result, Anil et al. (2019) make critical use of GroupSort activations.

Motivated by the above, our objective in the present article is to make a step towards a better mathematical understanding of the approximation properties of Lipschitz feedforward neural networks using GroupSort activations. Our contributions are threefold:

(i) We show that GroupSort neural networks, with constraints on the weights, can represent any Lipschitz continuous piecewise linear function and exhibit upper bounds on both their depth and size. We make a connection with the literature on the depth and size of ReLU networks (in particular Arora et al., 2018; He et al., 2018).

(ii) Building on the work of Anil et al. (2019), we offer upper bounds on the depth and size of GroupSort neural networks that approximate 1-Lipschitz continuous functions on

compact sets. We also show that increasing the grouping size may significantly improve the expressivity of GroupSort networks.

(*iii*) We empirically compare the performances of GroupSort and ReLU networks in the context of function regression estimation and Wasserstein distance approximation.

The mathematical framework together with the necessary notation is provided in Section 4.2. Section 4.3 is devoted to the problem of representing Lipschitz continuous functions with GroupSort networks of grouping size 2. The extension to any arbitrary grouping size is discussed in Section 4.4 and numerical illustrations are given in Section 4.5. For the sake of clarity, all proofs are gathered in the Appendix.

4.2 Mathematical context

We introduce in this section the mathematical context of the article and describe more specifically the GroupSort neural networks, which, as we will see, play a key role in representing and approximating Lipschitz continuous functions.

Throughout the paper, the ambient space $\mathbb{R}^d$ is assumed to be equipped with the Euclidean norm $\|\cdot\|$. For E a subset of $\mathbb{R}^d$, we denote by $\mathrm{Lip}_1(E)$ the set of 1-Lipschitz real-valued functions on E, i.e.,

$$\mathrm{Lip}_1(E) = \left\{ f : E \to \mathbb{R} : |f(x) - f(y)| \leqslant \|x - y\|,\ (x,y) \in E^2 \right\}$$

Let $k \geqslant 2$ be an integer. We let $\mathscr{D}_k = \{D_{k,\alpha} : \alpha \in \Lambda\}$ be the class of functions from $\mathbb{R}^d$ to $\mathbb{R}$ parameterized by feedforward neural networks of the form

$$\begin{aligned} D_{k,\alpha}(x) = \underset{1\times v_{q-1}}{V_q}\ \sigma_k\Big(\underset{v_{q-1}\times v_{q-2}}{V_{q-1}}\ \cdots \sigma_k\Big(\underset{v_2\times v_1}{V_2}\ \sigma_k\Big(\underset{v_1\times D}{V_1}\, x + \underset{v_1\times 1}{c_1}\Big) \\ + \underset{v_2\times 1}{c_2}\Big) + \underset{v_{q-1}\times 1}{c_{q-1}}\Big) + \underset{1\times 1}{c_q}\,, \end{aligned} \tag{4.2.1}$$

where $q \geqslant 2$ and the characters below the matrices indicate their dimensions (lines $\times$ columns). For $q = 1$, we simply let $D_{k,\alpha}(x) = V_1 x + c_1$ be a simple linear regression in $\mathbb{R}$ without hidden layers. Thus, a network in $\mathscr{D}_k$ has $(q-1)$ hidden layers, and hidden layers from depth 1 to $(q-1)$ are assumed to be of respective widths v_i, $i = 1,\ldots,q-1$, *divisible by k*. Such a network is said to be of depth q and of size $v_1 + \cdots + v_{q-1}$. The matrices V_i are the matrices of weights between layer i and layer $(i+1)$ and the c_i's are the corresponding offset vectors (in column format). So, altogether, the vectors $\alpha = (V_1,\ldots,V_q,c_1,\ldots,c_q)$ represent the parameter space Λ of the functions in $\mathscr{D}_k$.

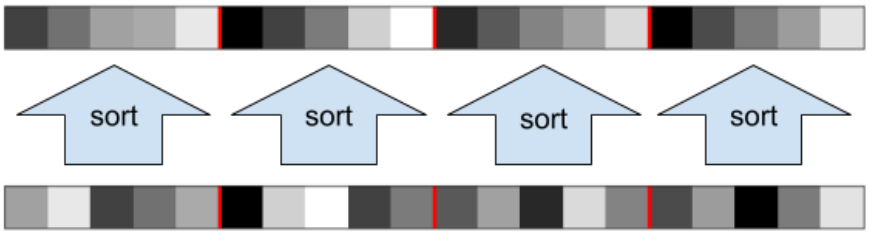


Fig. 4.1 GroupSort activation with a grouping size 5. Source: Anil et al. (2019).

With respect to the activation functions σ_k, we propose to use the GroupSort activation, which separates the pre-activations into groups and then sorts each group into ascending order. First, the GroupSort function splits the input into n different groups of k elements each: $G_1 = \{x_1, \ldots, x_k\}, \ldots G_n = \{x_{nk-k-1}, \ldots, x_{nk}\}$. Then, it orders each group by decreasing order as follows:

$$\sigma_k(x_1, \ldots, x_k, \ldots, x_{nk-(k-1)}, \ldots, x_{nk}) = \left(x_{(k)}^{G_1}, \ldots, x_{(1)}^{G_1}\right), \ldots, \left(x_{(k)}^{G_n}, \ldots, x_{(1)}^{G_n}\right).$$

where $x_{(1)}^{G}$, corresponding to the ordering statistics, is the smallest element of the group G.

This activation is applied on groups of k components, which makes sense in (4.2.1) since the widths of the hidden layers are assumed to be divisible by k. GroupSort has been introduced in Anil et al. (2019) as a 1-Lipschitz activation function that preserves the gradient norm of the input. An example with a grouping size $k = 5$ is given in Figure 4.1. With a slight abuse of vocabulary, we call a neural network of the form (4.2.1) a GroupSort neural network. We note that the GroupSort activation can recover the standard rectifier function. For example, $\sigma_2(x, 0) = (\mathrm{ReLU}(x), -\mathrm{ReLU}(-x))$, but the converse is not true.

Throughout the manuscript, the notation $\|\cdot\|$ (respectively, $\|\cdot\|_\infty$) means the Euclidean (respectively, the supremum) norm on $\mathbb{R}^p$, with no reference to p as the context is clear. For $W = (w_{i,j})$ a matrix of size $p_1 \times p_2$, we let $\|W\|_2 = \sup_{\|x\|=1} \|Wx\|$ be the 2-norm of W. Similarly, the ∞-norm of W is $\|W\|_\infty = \sup_{\|x\|_\infty=1} \|Wx\|_\infty = \max_{i=1,\ldots,p_1} \sum_{j=1}^{p_2} |w_{i,j}|$. We will also use the $(2,\infty)$-norm of W, i.e., $\|W\|_{2,\infty} = \sup_{\|x\|=1} \|Wx\|_\infty$. The following ass plays a central role in our approach:

Assumption 2. *For all* $\alpha = (V_1, \ldots, V_q, c_1, \ldots, c_q) \in \Lambda$,

$$\|V_1\|_{2,\infty} \leqslant 1, \ \max(\|V_2\|_\infty, \ldots, \|V_q\|_\infty) \leqslant 1,$$
$$\text{and } \max(\|c_i\|_\infty : i = 1, \ldots, q) \leqslant K_2,$$

where $K_2 \geqslant 0$ *is a constant.*

This type of compactness requirement has already been suggested in the statistical and machine learning community (e.g., Arjovsky et al., 2017; Anil et al., 2019; Biau et al., 2020). In the setting of this article, its usefulness is captured in the following simple but essential lemma:

Lemma 4.2.1. *Assume that Assumption 2 is satisfied. Then, for any $k \geqslant 2$, $\mathscr{D}_k \subseteq \mathrm{Lip}_1(\mathbb{R}^d)$.*

Combining Lemma 4.2.1 with Arzelà-Ascoli theorem, it is easy to see that, under Assumption 2, the class $\mathscr{D}_k$ restricted to any compact $K \subseteq \mathbb{R}^d$ is compact in the set of continuous functions on K with respect to the uniform norm. From this point of view, Assumption 2 is therefore somewhat restrictive. On the other hand, it is essential in order to guarantee that all neural networks in $\mathscr{D}_k$ are indeed 1-Lipschitz. Practically speaking, various approaches have been explored in the literature to enforce this 1-Lipschitz constraint. Gulrajani et al. (2017), Kodali et al. (2017), Wei et al. (2018), and Zhou et al. (2019) proposed a gradient penalty term, Miyato et al. (2018) applied spectral normalization, while Anil et al. (2019) have shown the empirical efficiency of the orthonormalization of Bjorck and Bowie (1971).

Importantly, Anil et al. (2019, Theorem 3) states that, under Assumption 2, GroupSort neural networks are universal Lipschitz approximators on compact sets. More precisely, for any Lipschitz continuous function f defined on a compact, one can find a neural network of the form (4.2.1) verifying Assumption 2 and arbitrarily close to f with respect to the uniform norm. Our objective in the present article is to explore the properties of these networks. We start in the next section by examining the case of piecewise linear functions.

4.3 Learning functions with a grouping size 2

For this section, we only consider GroupSort neural networks with a grouping size 2 and aim at studying their expressivity. The capacity of GroupSort networks to approximate continuous functions is studied via the representation of piecewise linear functions. For feedforward ReLU networks, their ability to represent such functions has been largely studied. In particular, Arora et al. (2018, Theorem 2.1) reveals that any piecewise linear function from $\mathbb{R}^d \to \mathbb{R}$ can be represented by a ReLU network of depth at most $\lceil \log_2(d+1) \rceil$ (the symbol $\lceil \cdot \rceil$ stands for the ceiling function), whereas He et al. (2018) specify an upper bound on their size. In the present section, we extend these results and first tackle the problem of representing piecewise linear functions with constrained GroupSort networks. Then we move to the non-linear case.

4.3.1 Representation of piecewise linear functions

Let us start gently by fixing the vocabulary.

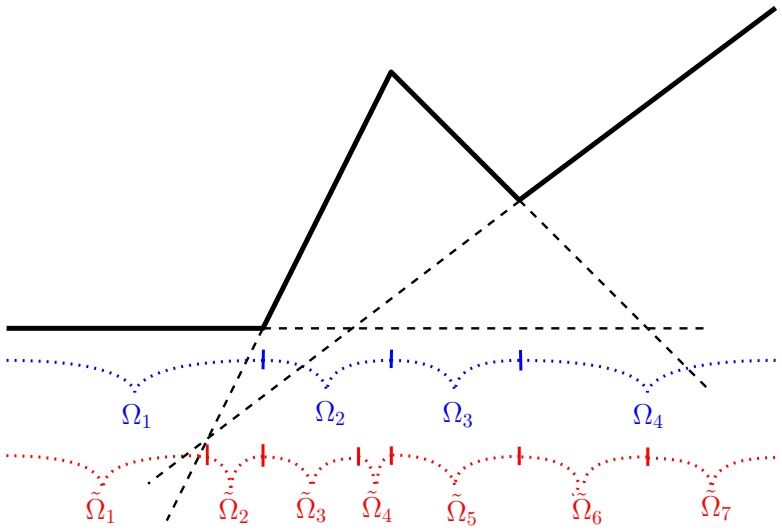


Fig. 4.2 A 4-piecewise linear function on the real line and the associated partitions $\Omega = \{\Omega_1, \ldots, \Omega_4\}$ and $\tilde{\Omega} = \{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_7\}$. The partition $\tilde{\Omega}$ is finer than Ω.

Definition 4.3.1. *A continuous function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be (continuous) m_f-piecewise linear ($m_f \geqslant 2$) if there exist a partition $\Omega = \{\Omega_1, \ldots, \Omega_{m_f}\}$ of $\mathbb{R}^d$ into polytopes and a collection $\ell_1, \ldots, \ell_{m_f}$ of affine functions such that, for all $x \in \Omega_i$, $i = 1, \ldots, m_f$, $f(x) = \ell_i(x)$.*

At this stage no further assumption is made on the sets $\Omega_1, \ldots, \Omega_{m_f}$, which are just assumed to be polytopes in $\mathbb{R}^d$. An example of piecewise linear function on the real line with $m_f = 4$ is depicted in Figure 4.2. As this figure suggests, the ambient space $\mathbb{R}^d$ can be further covered by a second partition $\tilde{\Omega} = \{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_{M_f}\}$ of M_f polytopes ($M_f \geqslant 1$), in such a way that the sign of the differences $\ell_i - \ell_j$, $(i, j) \in \{1, \ldots, m_f\}^2$, does not change on the subsets $\tilde{\Omega}_1, \ldots, \tilde{\Omega}_{M_f}$. It is easy to see that the partition $\tilde{\Omega}$ is finer than Ω since, for each $i \in \{1, \ldots, M_f\}$ there exists $j \in \{1, \ldots, m_f\}$ such that $\tilde{\Omega}_i \subseteq \Omega_j$. This implies in particular that $M_f \geqslant m_f$.

The usefulness of the partition $\tilde{\Omega}$ is demonstrated by He et al. (2018, Theorem 5.1), which states that any m_f-piecewise linear function f can be written as

$$f = \max_{1 \leqslant k \leqslant M_f} \min_{i \in S_k} \ell_i, \tag{4.3.1}$$

where each S_k is a non-empty subset of $\{1, \ldots, m_f\}$. This characterization of the function f is interesting, since it shows that any m_f-piecewise linear function can be computed using only a finite number of max and min operations. As identity (4.3.1) is essential for our approach, this justifies spending some time examining it.

Lemma 4.3.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an m_f-piecewise linear function. Then $m_f \leqslant M_f \leqslant \min(2^{m_f^2/2}, (m_f/\sqrt{2})^{2d})$.*

Lemma 4.3.1 is an improvement of He et al. (2018, Lemma 5.1), which shows that $M_f \leqslant m_f!$. Our proof method exploits the inequality $M_f \leqslant C_{m_f(m_f-1)/2,d}$, where $C_{n,d}$ denotes the number of arrangements of n hyperplanes in a space of dimension d (Devroye et al., 1996, Chapter 5). Another application of (4.3.1) is encapsulated in Proposition 4.3.1 below, which will be useful for later analysis, in combining maxima and minima in neural networks of the form (4.2.1).

Proposition 4.3.1. *Let $f_1,\ldots,f_m : \mathbb{R}^d \to \mathbb{R}$ be a collection of functions ($m \geqslant 2$), each represented by a neural network of the form (4.2.1), with common depth q and sizes s_i, $i = 1,\ldots,m$.*

In the specific case where $m = 2^n$ for some $n \geqslant 1$, there exist neural networks of the form (4.2.1) (with grouping size 2) with depth $q+\log_2(m)$ and size at most $s_1+\cdots+s_m+m-1$ that represent the functions $f = \max(f_1,\ldots,f_m)$ and $g = \min(f_1,\ldots,f_m)$.

If m is arbitrary, then there exist neural networks of the form (4.2.1) with depth $q+\lceil\log_2(m)\rceil$ and size at most $s_1+\cdots+s_m+2m-1$ that represent the functions f and g.

Interestingly, Arora et al. (2018, Lemma D.3), which is the analog of Proposition 4.3.1 asserts that the size with ReLU activations is at most $s_1+\cdots+s_m+8m-4$. For the specific computation of maxima/minima of functions, it should be stressed that GroupSort activations slightly reduces the size of the networks. By combining Lemma 4.3.1, Proposition 4.3.1, and identity (4.3.1), we are led to the following theorem, which reveals the ability of GroupSort networks for representing 1-Lipschitz piecewise linear functions.

Theorem 4.3.1. *Let $f \in \mathrm{Lip}_1(\mathbb{R}^d)$ that is also m_f-piecewise linear. Then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Besides, its depth is $\lceil\log_2(M_f)\rceil+\lceil\log_2(m_f)\rceil+1$ and its size is at most $3m_fM_f+M_f-1$.*

This result should be compared with state-of-the-art results known for ReLU neural networks. In particular, Arora et al. (2018, Theorem 2.1) reveals that any m_f-piecewise linear function f can be represented by a ReLU network with depth at most $\lceil\log_2(d+1)\rceil$. The upper bound of Theorem 4.3.1 can be larger since it involves both M_f and m_f. On the other hand, the upper bound $O(m_fM_f)$ on the size significantly improves on He et al. (2018, Theorem 5.2), which is at least $O(d2^{m_fM_f})$. This improvement in terms of size can be roughly explained by the depth/size trade-off results known in deep learning theory. As a matter of fact, many theoretical research papers have underlined the benefits of depth relatively to width for parameterizing complex functions (as, for example, in Telgarsky, 2015, 2016). For a fixed number of neurons, when comparing two neural networks, the deepest is the most expressive one (Lu et al., 2017).

It turns out that Theorem 4.3.1 can be significantly refined when the partition Ω satisfies some geometrical properties. Our next proposition examines the case where the sets $\Omega_1,\ldots,\Omega_{m_f}$ are convex.

Corollary 4.3.1. *Let $f \in \mathrm{Lip}_1(\mathbb{R}^d)$ that is also m_f-piecewise linear with convex subdomains $\Omega_1, \dots, \Omega_{m_f}$. Then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Besides, its depth is $2\lceil \log_2(m_f) \rceil + 1$ and its size is at most $3m_f^2 + m_f - 1$.*

Corollary 4.3.1 offers a significant improvement over Theorem 4.3.1, since in general $M_f \gg m_f$. We note in passing that the result of this proposition is dimension-free.

4.3.2 GroupSort neural networks on the real line

Piecewise linear functions defined on $\mathbb{R}$ deserve a special treatment, since in this case, any connected subset is convex.

Proposition 4.3.2. *Let $f \in \mathrm{Lip}_1(\mathbb{R})$ that is also m_f-piecewise linear. Then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Besides, its depth is $2\lceil \log_2(m_f) \rceil + 1$ and its size is at most $3m_f^2 + m_f - 1$.*

In the specific case where f is convex (or concave), then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Its depth is $\lceil \log_2(m_f) \rceil + 1$ and its size is at most $3m_f - 1$.

When f is convex (or concave) and $m_f = 2^n$ for some $n \geqslant 1$, then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Its depth is $\log_2(m_f) + 1$ and its size is at most $2m_f - 1$.

This proposition is the counterpart of Arora et al. (2018, Theorem 2.2), which states that any m_f-piecewise linear function from $\mathbb{R} \to \mathbb{R}$ can be represented by a 2-layer ReLU neural network with a size at least $m_f - 1$. He et al. (2018, Theorem 5.2) shows that the upper-bound on the size of ReLU networks is $O(2^{m^2+2(m-1)})$. Thus, for the representation of piecewise linear functions on the real line, GroupSort networks require larger depths but smaller sizes. Besides, bear in mind that the obtained ReLU neural networks do not necessarily verify a requirement similar to the one of Assumption 2.

Regarding the number of linear regions of GroupSort networks on the real line, we have the following result:

Lemma 4.3.2. *Any neural network of the form (4.2.1) on the real line, with depth q and widths $\nu_1, \dots, \nu_{q-1}$, parameterizes a piecewise linear function with at most $2^{q-2} \times (\nu_1/2 + 1) \times \nu_2 \times \cdots \times \nu_{q-1}$ linear subdomains.*

We deduce from this lemma that for a neural network of the form (4.2.1) with depth $q \geqslant 2$ and constant width ν, the maximum number of linear regions is $O(2^{q-3}\nu^{q-1})$. Similarly to ReLU networks (Montúfar et al., 2014; Arora et al., 2018), the maximum number of linear

regions for GroupSort networks with grouping size 2 is also likely to grow polynomially in ν and exponentially in q.

Our next corollary now illustrates the trade-off between depth and width for GroupSort neural networks.

Corollary 4.3.2. *Let $f \in \mathrm{Lip}_1(\mathbb{R})$ be an m_f-piecewise linear function. Then, any neural network of the form (4.2.1) verifying Assumption 2 and representing f with a depth q, has a size s at least $\frac{1}{2}(q-1)m_f^{1/(q-1)}$.*

The lower bound highlighted in Corollary 4.3.2 is dependent on the depth q of the neural network. By looking at the minimum of the function, we get that any neural network representing f has a size $s \geqslant \frac{e\ln(m_f)}{2}$. Thus, merging this result with Proposition 4.3.2, we have that for any m_f-piecewise linear function from $\mathbb{R} \to \mathbb{R}$, there exists a GroupSort network verifying Assumption 2 with a size s satisfying

$$\frac{e\ln(m_f)}{2} \leqslant s \leqslant 3m_f^2 - m_f - 3.$$

We realize that this inequality is large but, up to our knowledge, this is first of this type for GroupSort neural networks.

4.3.3 Approximating Lipschitz continuous functions on compact sets

Following our plan, we tackle in this subsection the task of approximating Lipschitz continuous functions on compact sets using GroupSort neural networks. The space of continuous functions on $[0,1]^d$ is equipped with the uniform norm

$$\|f-g\|_\infty = \max_{x\in[0,1]^d} |f(x)-g(x)|.$$

The main result of the section, and actually of the article, is that GroupSort neural networks are well suited for approximating functions in $\mathrm{Lip}_1([0,1]^d)$.

Theorem 4.3.2. *Let $\varepsilon > 0$ and $d \geqslant 2$, $f \in \mathrm{Lip}_1([0,1]^d)$. Then there exists a neural network D of the form (4.2.1) verifying Assumption 2 such that $\|f-D\|_\infty \leqslant \varepsilon$. The depth of D is $O(d^2\log_2(\frac{2\sqrt{d}}{\varepsilon}))$ and its size is $O((\frac{2\sqrt{d}}{\varepsilon})^{d^2})$.*

To the best of our knowledge, Theorem 4.3.2 is the first one that provides an upper bound on the depth and size of neural networks, with constraints on the weights, that approximate Lipschitz continuous functions.

As for the representation of piecewise linear functions, one can, for the sake of completeness, compare this bound with those previously found in the literature of ReLU neural networks. Yarotsky (2017) establishes the density of ReLU networks in Sobolev spaces, using a different technique of proof. In particular, Theorem 1 of this paper states that for any $f \in \mathrm{Lip}_1([0,1]^d)$ continuously differentiable, there exists a ReLU neural network approximating f with precision ε, with depth at most $c(\ln(1/\varepsilon)+1)$ and size at most $c\varepsilon^{-d}(\ln(1/\varepsilon)+1)$ (with a constant c function of d). Comparing this result with our Theorem 4.3.2, we see that, with respect to ε, both depths are similar but ReLU networks are smaller in size. However, one has to keep in mind that both lines of proof largely differ. Besides, our formulation ensures that the approximator is also a 1-Lipschitz function, a feature that cannot be guaranteed under the formulation of Yarotsky (2017).

It turns out however that our framework provides smaller neural networks as soon as $d = 1$.

Proposition 4.3.3. *Let $\varepsilon > 0$ and $f \in \mathrm{Lip}_1([0,1])$. Then there exists a neural network D of the form (4.2.1) verifying Assumption 2 such that $\|f - D\|_\infty \leqslant \varepsilon$. The depth of D is $2\lceil \log_2(1/\varepsilon)\rceil + 1$ and its size is $O((\frac{1}{\varepsilon})^2)$.*

Besides, if f is assumed to be convex or concave, then the depth of D is $\lceil \log_2(1/\varepsilon)\rceil + 1$ and its size is $O(\frac{1}{\varepsilon})$.

4.4 Impact of the grouping size

The previous section paved the way for a better understanding of GroupSort neural networks and their ability to approximate Lipschitz continuous functions. As mentioned in Section 4.2, one can play with the grouping size k of the neural network when defining its architecture. However, it is not clear how changing this parameter might influence the expressivity of the network. The present section aims at bringing some understanding. Following a similar reasoning as in Section 4.3, we start by analyzing how GroupSort networks with an arbitrary grouping size $k \geqslant 2$ can represent any piecewise linear functions:

Proposition 4.4.1 (Extension of Proposition 4.3.1)**.** *Let $f_1, \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ be a collection of functions ($m \geqslant 2$), each represented by a neural network of the form (4.2.1), with common depth q and sizes s_i, $i = 1, \ldots, m$.*

In the specific case where $m = k^n$ for some $n \geqslant 1$, there exist neural networks of the form (4.2.1) (with grouping size k) with depth $q + \log_k(m)$ and size at most $s_1 + \cdots + s_m + \frac{m-1}{k-1} - 1$ that represent the functions $f = \max(f_1, \ldots, f_m)$ and $g = \min(f_1, \ldots, f_m)$.

Similarly to Section 4.3, this leads to the following corollary:

Methods	Up Depth	Up Size	Down Size	Reference
Representing $m = k^n$-PWL functions in $\mathbb{R}^d$ with a constant width ν				
ReLU	$\lceil \log_2(d+1) \rceil + 1$	$O(d2^{m^2})$	$O(m)$	He et al. (2018)
GroupSort $GS = k$	$\lceil 2\log_k(m) \rceil + 1$	$\frac{m^2-1}{k-1}$	$\frac{\nu \log_k(m)}{2\log_k(\nu)}$	present article
Approximating 1-Lipschitz continuous functions in $[0,1]^d$				
ReLU	$O(\ln(\frac{1}{\varepsilon}))$	$O(\frac{\ln(1/\varepsilon)}{\varepsilon^d})$	\	Yarotsky (2017)
GroupSort $GS = \lceil \frac{2\sqrt{d}}{\varepsilon} \rceil$	$O(d^2)$	$O((\frac{2\sqrt{d}}{\varepsilon})^{d^2-1})$	\	present article
Approximating 1-Lipschitz continuous functions in $[0,1]$				
ReLU (PWL representation)	2	$O(2^{1/\varepsilon^2+2/\varepsilon})$	\	He et al. (2018)
ReLU (different approach)	$O(\ln(\frac{1}{\varepsilon}))$	$O(\frac{\ln(1/\varepsilon)}{\varepsilon})$	\	Yarotsky (2017)
Adaptative ReLU	6	$O(\frac{1}{\varepsilon \ln(1/\varepsilon)})$	\	Yarotsky (2017)
GroupSort $GS = \lceil \frac{1}{\varepsilon} \rceil$	3	$O(\frac{1}{\varepsilon})$	\	present article

Table 4.1 Summary of the results shown in the present paper together with results previously found for ReLU networks. "Up Depth" refers to upper bounds on the depths, "Up Size" to upper bounds on the sizes, and "Down Size" to lower bounds on the sizes. The symbol "\" means that no result is known (up to our knowledge).

Corollary 4.4.1 (Extension of Corollary 4.3.1)**.** *Let $f \in \mathrm{Lip}_1(\mathbb{R}^d)$ that is also m_f-piecewise linear with convex subdomains $\Omega_1, \ldots, \Omega_{m_f}$ such that $m_f = k^n$ for some $n \geqslant 1$. Then there exists a neural network of the form (4.2.1) verifying Assumption 2 that represents f. Besides, its depth is $2\lceil \log_k(m_f) \rceil + 1$ and its size is at most $\frac{m_f^2-1}{k-1}$.*

Proposition 4.4.1 and Corollary 4.4.1 exhibit the nice properties of using larger grouping sizes. Indeed, for a given $q \geqslant 1$, there exists a neural network with depth $2q+1$ and grouping size k representing a function with k^q pieces. Consequently, the use of larger grouping sizes helps have more expressive neural networks. The efficiency of larger grouping sizes may also be explained by the following result for GroupSort networks on the real line:

Lemma 4.4.1 (Extension of Lemma 4.3.2)**.** *Any neural network of the form (4.2.1) on the real line, with depth q, widths $\nu_1, \ldots, \nu_{q-1}$, and grouping size k, parameterizes a piecewise linear function with at most $k^{q-2} \times (\frac{(k-1)\nu_1}{2} + 1) \times \nu_2 \times \cdots \times \nu_{q-1}$ linear subdomains.*

Thus, the number of linear regions of a GroupSort network is likely to increase polynomially with the grouping size, which highlights the benefits of using larger groups. Similarly to Section 4.3, when moving to the approximation of Lipschitz continuous functions on $[0,1]^d$, we are lead to the following theorem:

Theorem 4.4.1 (Extension Theorem 4.3.2)**.** *Let $\varepsilon > 0$, $d \geqslant 2$, and $f \in \mathrm{Lip}_1([0,1]^d)$. Then there exists a neural network D of the form (4.2.1) verifying Assumption 2 with grouping size $\lceil \frac{2\sqrt{d}}{\varepsilon} \rceil$ such that $\|f - D\|_\infty \leqslant \varepsilon$. The depth of D is $O(d^2)$ and its size is $O((\frac{2\sqrt{d}}{\varepsilon})^{d^2-1})$.*

Using a grouping size proportional to $1/\varepsilon$, we thus have a bound on the depth that is independent from the error rate. The uni-dimensional case leads to a different result:

Proposition 4.4.2 (Extension of Proposition 4.3.3)**.** *Let $\varepsilon > 0$ and $f \in \mathrm{Lip}_1([0,1])$. Then there exists a neural network D of the form (4.2.1) verifying Assumption 2 (with grouping size k) such that $\|f - D\|_\infty \leqslant \varepsilon$. The depth of D is $2\lceil \log_k(\frac{1}{\varepsilon})\rceil + 1$ and its size is at most $O(\frac{1}{k\varepsilon^2})$.*

In particular, if k is chosen to be equal to $\lceil \frac{1}{\varepsilon}\rceil$, then the depth of D is 3 and its size is $O(\frac{1}{\varepsilon})$.

When approximating real-valued functions, the use of larger grouping sizes can significantly decrease the required size since it goes from $O(1/\varepsilon^2)$ in Proposition 4.3.3 to $O(1/\varepsilon)$ in Proposition 4.4.2. When f is assumed to be convex or concave, the depth of the network D can further be reduced to 2.

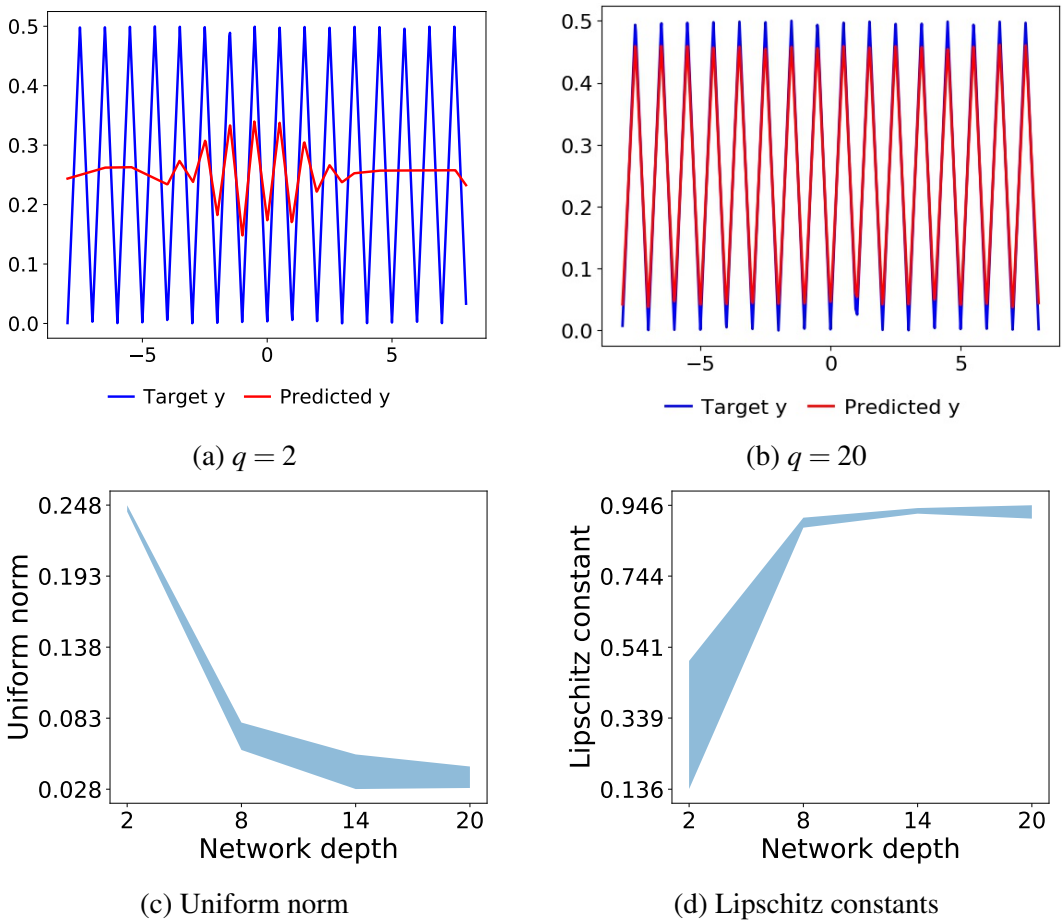
Using a different approach for approximating Lipschitz continuous functions in $[0,1]$, Yarotsky (2017, Theorem 1) shows that ReLU networks with a depth of $O(\ln(1/\varepsilon))$ is needed together with a size $O(\frac{\ln(1/\varepsilon)}{\varepsilon})$ to approximate with an error rate ε. To sum-up, when compared with ReLU networks, GroupSort neural networks with well-chosen grouping size can be significantly more expressive.

Table 4.1 summarizes the results shown in the present paper together with results previously found for ReLU networks. Bear in mind that GroupSort neural networks also have the supplementary condition that any parameterized function verifies the 1-Lipschitz continuity.

4.5 Experiments

Anil et al. (2019) have already compared the performances of GroupSort neural networks with their ReLU counterparts, both with constraints on the weights. In particular, they showed that ReLU neural networks are more sensitive to adversarial attacks while stressing the fact that if their weights are limited, then these networks lose their expressive power. Building on these observations, we further illustrate the good behavior of GroupSort neural networks in the context of estimating a Lipschitz continuous regression function and in approximating the Wasserstein distance (via its dual form) between pairs of distributions.

Impact of the depth. We start with the problem of learning a function f in the model $Y = f(X)$, where X follows a uniform distribution on $[-8,8]$ and f is 32-piecewise linear. To this aim, we use neural networks of the form (4.2.1) with respective depth $q = 2$, 8, 14, 20, and a constant width $\nu = 50$. Since we are only interested in the approximation properties of the networks, we assume to have at hand an infinite number of pairs $(X_i, f(X_i))$ and train the models by minimizing the mean squared error. We give in the Appendix, the full details of our



(a) $q = 2$

(b) $q = 20$

(c) Uniform norm

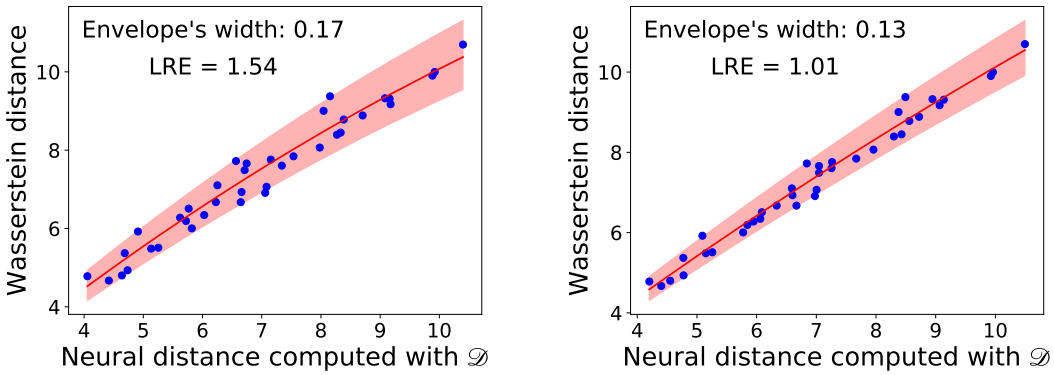
(d) Lipschitz constants

Fig. 4.3 Reconstruction of a 32-piecewise linear function on $[-8,8]$ with a GroupSort neural network of the form (4.2.1) with depth $q = 2$, 8, 14, 20, and a constant width $v = 50$ (the thickness of the line represents a 95%-confidence interval).

experimental setting. The quality of the estimation is evaluated using the uniform norm between the target function f and the output network. In order to enforce Assumption 2, GroupSort neural networks are constrained using the orthonormalization of Bjorck and Bowie (1971). The results are presented in Figure 4.3. Note that throughout this section, confidence intervals are computed over 20 runs. In line with Theorem 4.3.1, which states that f is representable by a neural network of the form (4.2.1) with size at most $3 \times 32^2 + 32 - 1 = 3104$, we clearly observe that, as the depth of the networks increases, the uniform norm decreases and the Lipschitz constant of the network converges to 1. The reconstruction of this piecewise linear function is even almost perfect for the depth $q = 20$, i.e., with a network of size only $20 \times 60 = 1200$, a value significantly smaller than the upper bound of the theorem.

We also illustrate the behavior of GroupSort neural networks in the context of WGANs (Arjovsky et al., 2017). We run a series of small experiments in the simplified setting where we

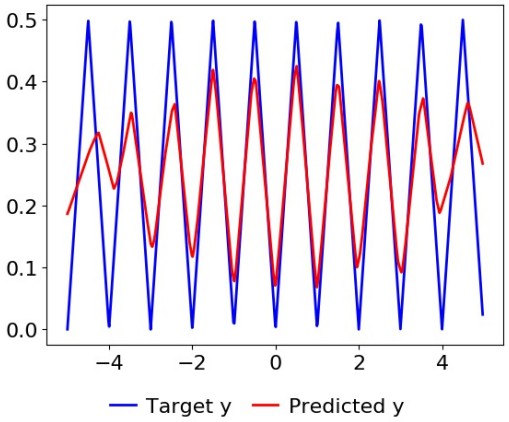
try to approximate the 1-Wasserstein distance between two bivariate mixtures of independent Gaussian distributions with 4 components. We consider networks of the form (4.2.1) with grouping size 2, a depth $q = 2$ and $q = 5$, and a constant width $v = 20$. For a pair of distributions (μ, ν), our goal is to exemplify the relationship between the 1-Wasserstein distance $\sup_{f \in \mathrm{Lip}_1(\mathbb{R}^2)} (\mathbb{E}_\mu - \mathbb{E}_\nu)$ (approximated with the Python package by Flamary and Courty, 2017) and the neural distance $\sup_{f \in \mathscr{D}_2} (\mathbb{E}_\mu - \mathbb{E}_\nu)$ (Arora et al., 2017) computed over the class of functions $\mathscr{D}_2$. To this aim, we randomly draw 40 different pairs of distributions. Then, for each of these pairs, we compute an approximation of the 1-Wasserstein distance and calculate the corresponding neural distance. Figure 4.4 depicts the best parabolic fit between 1-Wasserstein and neural distances, and shows the corresponding Least Relative Error (LRE) together with the width of the envelope. The take-home message of this figure is that both the LRE and the width are significantly smaller for deeper GroupSort neural networks.



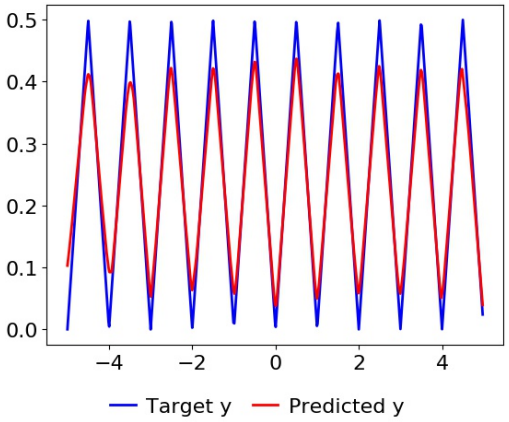
(a) $q = 2$ (b) $q = 5$

Fig. 4.4 Scatter plots of 40 pairs of Wasserstein and neural distances computed with GroupSort neural networks, for $q = 2, 5$. The underlying distributions are bivariate Gaussians. The red curve is the optimal parabolic fitting and LRE refers to the Least Relative Error. The red zone is the envelope obtained by stretching the optimal curve.

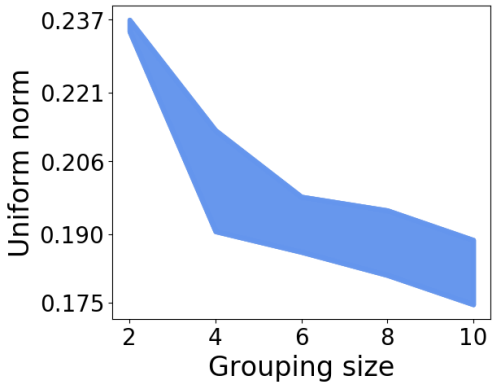
Impact of the grouping size. To highlight the benefits of using larger grouping sizes, we show the impact of increasing the grouping size from 2 in Figure 4.5a to 5 in Figure 4.5b for the representation of a 20-piecewise linear function. This is corroborated by Figure 4.5c, which illustrates that the uniform norm with a 64-piecewise linear function decreases when the grouping size increases. As already underlined in Lemma 4.4.1, this may be explained by the fact that the number of linear regions significantly grows with the grouping size—see Figure 4.5d.



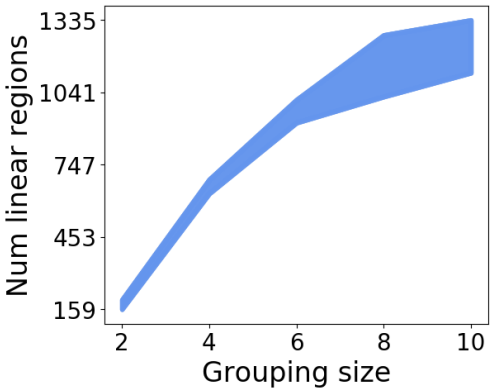
(a) Grouping size = 2



(b) Grouping size = 6



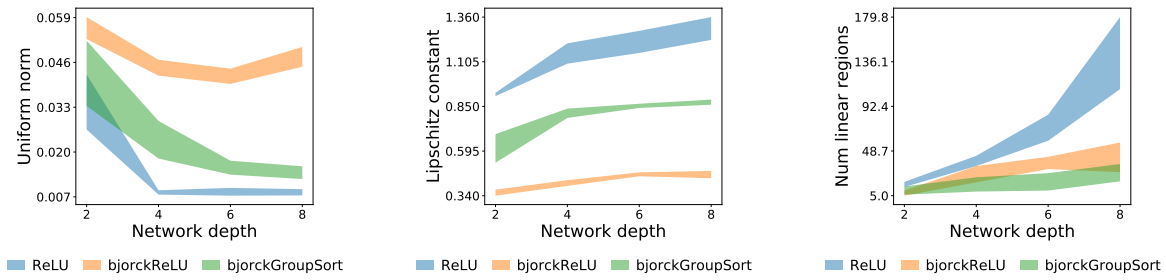
(c) Uniform norm



(d) Linear regions

Fig. 4.5 Reconstruction of a 20-piecewise linear function on $[-5,5]$ (top line) and a 64-piecewise linear function (bottom line) with GroupSort neural networks of the form (4.2.1) with depth $q = 4$ and varying grouping sizes $k = 2,4,6,8,10$.

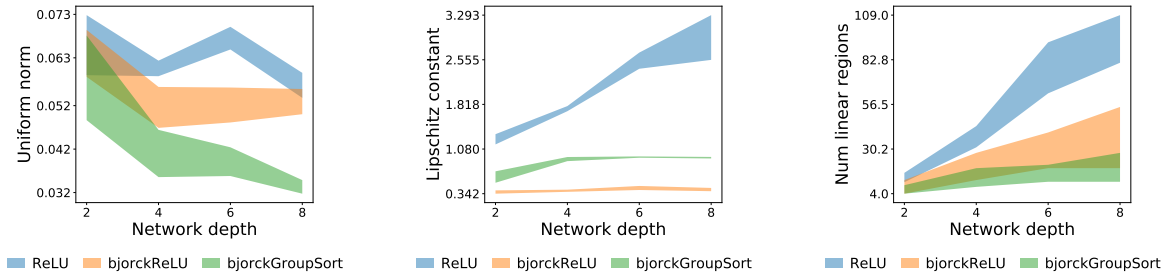
Comparison with ReLU neural networks. Next, in a second series of experiments, we compare the performances of GroupSort networks against two baselines: ReLU neural networks without constraints on the weights (dense in the set of continuous functions on a compact set; see Yarotsky, 2017), and ReLU neural networks with orthonormalization of Bjorck and Bowie (1971). The architecture of the ReLU neural networks in terms of depth and width is the same as for GroupSort networks: $q = 2$, 4, 6 ,8, and $w = 20$. The task is now to approximate the 1-Lipschitz continuous function $f(x) = (1/15)\sin(15x)$ on $[0,1]$ in the models $Y = f(X)$ (noiseless case) and $Y = f(X) + \varepsilon$ (noisy case), where X is uniformly distributed on $[0,1]$ and ε follows a Gaussian distribution with standard deviation 0.05. In both cases, we assume to have at hand a finite sample of size $n = 100$ and fit the models by minimizing the mean squared error.



(a) Prediction quality

(b) Lipschitz constants

(c) Number of linear regions



(d) Prediction quality

(e) Lipschitz constant

(f) Number of linear regions

Fig. 4.6 (Top line) Estimating the function $f(x) = (1/15)\sin(15x)$ on $[0,1]$ in the model $Y = f(X)$, with a dataset of size $n = 100$. (Bottom line) Estimating the function $f(x) = (1/15)\sin(15x)$ on $[0,1]$ in the model $Y = f(X) + \varepsilon$, with a dataset of size $n = 100$ (the thickness of the line represents a 95%-confidence interval).

Both results (noiseless case and noisy case) are presented in Figure 4.6. We observe that in the noiseless setting Figure 4.6a, 4.6b, and 4.6c, ReLU neural networks without normalization have a slightly better performance with respect to the uniform norm with, however, a Lipschitz constant larger than 1. On the other hand, in the noisy case, ReLU neural networks without constraints have a tendency to overfitting (a high Lipschitz constant close to 2.7), leading to a deteriorated performance, contrary to GroupSort neural networks. Furthermore, in both cases (noiseless and noisy), ReLU with constraints are found to perform worse (due to a Lipschitz constant much smaller than 1) than their GroupSort counterparts in terms of prediction. Interestingly, we see in the two examples shown in Figure 4.6e and Figure 4.6f, that the number of linear regions for GroupSort neural networks is smaller than for ReLU networks.

Finally, we quickly show in Appendix a comparison between GroupSort and ReLU networks when approximating Wasserstein distances. The take home message is that, on this specific task, GroupSort networks perform better.

4.6 Conclusion

The results presented in this article show the advantage of using GroupSort neural networks over standard ReLU networks. On the one hand, ReLU neural networks without any constraints are sensitive to adversarial attacks (as they may have a large Lipschitz constant) and, on the other hand, lose expressive power when enforcing limits on their weights. On the opposite, GroupSort neural networks with constrained weights are proved to be both robust and expressive, and are therefore an interesting alternative. Moreover, by allowing larger grouping sizes for GroupSort networks, one can further increase their expressivity. These properties open new perspectives for broader use of GroupSort networks.

Appendix 4.A Technical results

4.A.1 Proof of Lemma 4.2.1

We prove the result for $\mathscr{D}_2$. The result for $\mathscr{D}_k$ holds following a similar argument.

Fix $D_{2,\alpha} \in \mathscr{D}_2$, $\alpha \in \Lambda$. According to (4.2.1), we have, for $x \in \mathbb{R}^d$, $D_{2,\alpha}(x) = f_q \circ \cdots \circ f_1(x)$, where $f_i(t) = \sigma_2(V_i t + c_i)$ for $i = 1, \ldots, q-1$ (σ_2 is applied on pairs of components), and

$f_q(t) = V_q t + c_q$. Therefore, for $(x,y) \in (\mathbb{R}^d)^2$,

$$\begin{aligned}
\|f_1(x) - f_1(y)\|_\infty &\leqslant \|V_1 x - V_1 y\|_\infty \\
&\quad \text{(since } \sigma_2 \text{ is 1-Lipschitz)} \\
&= \|V_1(x-y)\|_\infty \\
&\leqslant \|V_1\|_{2,\infty} \, \|x-y\| \\
&\leqslant \|x-y\| \\
&\quad \text{(by Assumption 2).}
\end{aligned}$$

Thus,

$$\begin{aligned}
\|f_2 \circ f_1(x) - f_2 \circ f_1(y)\|_\infty &\leqslant \|V_2 f_1(x) - V_2 f_1(y)\|_\infty \\
&\quad \text{(since } \sigma_2 \text{ is 1-Lipschitz)} \\
&\leqslant \|V_2\|_\infty \, \|f_1(x) - f_1(y)\|_\infty \\
&\leqslant \|f_1(x) - f_1(y)\|_\infty \\
&\quad \text{(by Assumption 2)} \\
&\leqslant \|x-y\|.
\end{aligned}$$

Repeating this, we conclude that, for each $\alpha \in \Lambda$ and all $(x,y) \in (\mathbb{R}^d)^2$, $|D_{2,\alpha}(x) - D_{2,\alpha}(y)| \leqslant \|x-y\|$, which is the desired result.

4.A.2 Proof of Lemma 4.3.1

Recall that $m_f \geqslant 2$. Throughout the proof, we let $\cdot$ refer to the dot product in $\mathbb{R}^d$. Let $(i,j) \in \{1,\dots,m_f\}^2$, $i \neq j$. There exist $(a_i,b_i) \in \mathbb{R}^d \times \mathbb{R}$ and $(a_j,b_j) \in \mathbb{R}^d \times \mathbb{R}$ such that $\ell_i = a_i \cdot x + b_i$ and $\ell_j = a_j \cdot x + b_j$. Therefore,

$$\ell_i(x) - \ell_j(x) \leqslant 0 \iff x \cdot (a_i - a_j) \leqslant b_j - b_i.$$

So, there exist two subdomains $\tilde{\Omega}_1$ and $\tilde{\Omega}_2$, separated by an affine hyperplane, in which $\ell_i - \ell_j$ does not change sign. By repeating this operation for the $m_f(m_f-1)/2$ different pairs (ℓ_i,ℓ_j), we get that the number M_f of subdomains on which any pair $\ell_i - \ell_j$ does not change sign is smaller than the maximal number of arrangements of $m_f(m_f-1)/2$ hyperplanes.

Denoting by $C_{n,d}$ the maximal number of arrangements of n hyperplanes in $\mathbb{R}^d$, we know that when $d > n$ then $C_{n,d} = 2^n$, whereas if $n > d$ the upper bound $C_{n,d} \leqslant (1+n)^d$ becomes

preferable (Devroye et al., 1996, Chapter 30). Thus, we have

$$m_f \leqslant M_f \leqslant \min\left(2^{m_f^2/2}, (m_f/\sqrt{2})^{2d}\right).$$

4.A.3 Proof of Proposition 4.3.1

We prove the first part of the proposition by using an induction on n. The case where $n = 1$ and thus $m = 2^1$ is clear since the function $f = \max(f_1, f_2)$ can be represented by a neural network of the form (4.2.1) with depth $q+1$ and size $s_1 + s_2 + 1$. Now, let $m = 2^n$ with $n > 1$. We have that $m/2 = 2^{n-1}$. By the induction hypothesis, $g_1 = \max(f_1, \ldots, f_{m/2})$ and $g_2 = \max(f_{m/2+1}, \ldots, f_m)$ can be represented by neural networks of the form (4.2.1) with depths $q+n-1$, and sizes at most $s_1 + \cdots + s_{m/2} + m/2 - 1$ and $s_{m/2+1} + \cdots + s_m + m/2 - 1$, respectively. Consequently, the function $G(x) = (g_1(x), g_2(x))$ can be implemented by a neural network of the form (4.2.1) with depth $q+n-1$ and size $s_1 + \cdots + s_m + m - 2$. Finally, by concatenating a one neuron layer, we have that the function $f = \max(g_1, g_2)$ can be represented by a neural network of the form (4.2.1) with depth $q+n = q+\log_2(m)$ and size at most $s_1 + \cdots + s_m + m - 1$.

Now, let us prove the case where m is arbitrary. Let $f_1, \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ be a collection of functions ($m \geqslant 2$), each represented by a neural network of the form (4.2.1) with depth q and size s_i, $i = 1, \ldots, m$. We prove below by an induction on n that there exists a neural network of the form (4.2.1) with depth $q + \lceil \log_2(m) \rceil$, a final layer of width $v_{q-1} = 2$, and a size at most $s_1 + \cdots + s_m + 2^{\lceil \log_2(m) \rceil} - 1$ that represents the functions $f = \max(f_1, \ldots, f_m)$ and $g = \min(f_1, \ldots, f_m)$ (the symbol $\lceil \cdot \rceil$ stands for the ceiling function and the symbol $\lfloor \cdot \rfloor$ stands for the integer function).

The base case $m = 2$ is clear using the GroupSort activation and $v_1 = 2$. For $m > 2$, let $n \geqslant 2$ be such that $2^{n-1} \leqslant m < 2^n$. Let $g_1 = \max(f_1, \ldots, f_{2^{n-1}})$ and $g_2 = \max(f_{2^{n-1}+1}, \ldots, f_m)$. From the first part of the proof, we know that g_1 can be represented by a neural network of the form (4.2.1) with depth $q_1 = q + \lfloor \log_2 m \rfloor = q + n - 1$ and size $s_1 + \cdots + s_{2^{n-1}} + 2^{n-1} - 1$. Also, by the induction hypothesis, g_2 can be represented by a neural network of the form (4.2.1) with depth $q_2 = q + \lceil \log_2(m - 2^{n-1}) \rceil$ and size at most $s_{2^{n-1}+1} + \cdots + s_m + 2^{\lceil \log_2(m-2^{n-1}) \rceil} - 1$. Therefore, by padding identity matrices with two neurons (recall that $v_{q_2-1} = 2$) on layers from

$q + \lceil \log_2(m - 2^{n-1}) \rceil$ to $q + n - 1$, we have:

$$2^{\lceil \log_2(m-2^{n-1}) \rceil} - 1 + 2(n - 2 - \lceil \log_2(m - 2^{n-1}) \rceil) = \sum_{k=0}^{k=\lceil \log_2(m-2^{n-1}) \rceil - 1} 2^k + \sum_{k=\lceil \log_2(m-2^{n-1}) \rceil}^{k=n-2} 2^1$$
$$\leqslant \sum_{k=0}^{k=n-2} 2^k = 2^{n-1} - 1.$$

Thus, g_2 can be represented by a neural network of the form (4.2.1) with depth $q_2 = q + \lfloor \log_2 m \rfloor$ and size at most $s_{2^{n-1}+1} + \cdots + s_m + 2^{n-1} - 1$. Now, the bivariate function $G(x) = (g_1(x), g_2(x))$ can be implemented by a neural network of the form (4.2.1) with depth $q + \lfloor \log_2(m) \rfloor$ and size s such that

$$s \leqslant s_1 + \cdots + s_m + 2(2^{n-1} - 1) = s_1 + \cdots + s_m + 2^n - 2.$$

By concatenating a one neuron layer, we have that the function $f = \max(g_1, g_2)$ can be represented by a neural network of the form (4.2.1) with depth $q + \lceil \log_2(m) \rceil$ and size at most $s_1 + \cdots + s_m + 2^n - 1 = s_1 + \cdots + s_m + 2^{\lceil \log_2 m \rceil} - 1$. The conclusion follows using the inequality $2^{\lceil \log_2 m \rceil} \leqslant 2m$.

4.A.4 Proof of Theorem 4.3.1

Let $f \in \mathrm{Lip}_1(\mathbb{R}^d)$ that is also m_f-piecewise linear. We know that each linear function can be represented by a 1-neuron neural network verifying Assumption 2 (no need for hidden layers). Combining (4.3.1) with Proposition 4.3.1, for each $k \in \{1, \ldots, M_f\}$ there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing the function $\min_{i \in S_k} \ell_i$, with depth equal to $\lceil \log_2(m_f) \rceil + 1$ (since $|S_k| \leqslant m_f$) and size at most $3m_f - 1$.

Using again Proposition 4.3.1, we conclude that there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f, with depth $\lceil \log_2(M_f) \rceil + \lceil \log_2(m_f) \rceil + 1$ and size at most $3m_f M_f + M_f - 1$.

4.A.5 Proof of Corollary 4.3.1

According to He et al. (2018, Theorem A.1), the function f can be written as

$$f = \max_{1 \leqslant k \leqslant m_f} \min_{i \in S_k} \ell_i,$$

where $|S_k| \leqslant m_f$. Using the same technique of proof as for Theorem 4.3.1, we find that there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f, with depth equal to $2\lceil \log_2(m_f) \rceil + 1$ and size at most $3m_f^2 + m_f - 1$.

4.A.6 Proof of Proposition 4.3.2

Let $f \in \mathrm{Lip}_1(\mathbb{R})$ that is also m_f-piecewise linear. The proof of the first statement is an immediate consequence of Corollary 4.3.1 since connected subsets of $\mathbb{R}$ are also convex.

As for the second claim of the proposition, considering the case where f is convex, we know from He et al. (2018, Theorem A.1) that f can be written as

$$f = \max_{1 \leqslant k \leqslant m_f} \ell_k.$$

Each function ℓ_k, $k = 1, \ldots, m_f$, can be represented by a 1-neuron neural network verifying Assumption 2. Hence, by Proposition 4.3.1, there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f, with depth $\lceil \log_2(m_f) \rceil + 1$ and size at most $3m_f - 1$.

The last claim of the proposition for $m = 2^n$ is clear using Proposition 4.3.1.

4.A.7 Proof of Lemma 4.3.2

The result is proved by induction on q. To begin with, in the case $q = 2$ we have a neural network with one hidden layer. When applying the GroupSort function with a grouping size 2, every activation node is defined as the max or min between two different linear functions. The maximum number of breakpoints is equal to the maximum number of intersections, that is $v_1/2$. Thus, there is at most $v_1/2 + 1$ pieces.

Now, let us assume that the property is true for a given $q \geqslant 3$. Consider a neural network with depth q and widths $v_1, \ldots, v_{q-1}$. Observe that the input to any node in the last layer is the output of a $\mathbb{R} \to \mathbb{R}$ GroupSort neural network with depth $(q-1)$ and widths $v_1, \ldots, v_{q-2}$. Using the induction hypothesis, the input to this node is a function from $\mathbb{R} \to \mathbb{R}$ with at most $2^{q-3} \times (v_1/2+1) \times \cdots \times v_{q-2}$ pieces. Thus, after applying the GroupSort function with a grouping size 2, each node output is a function with at most $2 \times (2^{q-3} \times (v_1/2+1) \times v_2 \times \cdots \times v_{q-2})$. With the final layer, we take an affine combination of v_{q-1} functions, each with at most $2^{q-2} \times (v_1/2+1) \times v_2 \times \cdots \times v_{q-2}$ pieces. In all, we therefore get at most $2^{q-2} \times (v_1/2+1) \times v_2 \times \cdots \times v_{q-1}$ pieces. The induction step is completed.

4.A.8 Proof of Corollary 4.3.2

Let f be an m_f-piecewise linear function. For a neural network of depth q and widths $\nu_1, \dots, \nu_q$ representing f, we have, by Lemma 4.3.2,

$$2^{q-1} \times (\nu_1/2+1) \times \cdots \times \nu_{q-1} \geqslant m_f.$$

By the inequality of arithmetic and geometric means, minimizing the size $s = \nu_1/2 + \cdots + \nu_k$ subject to this constraint, means setting $\nu_1/2+1 = \nu_2 = \cdots = \nu_k$. This implies that $s \geqslant \frac{1}{2}(q-1)m_f^{1/(q-1)}$.

4.A.9 Proof of Theorem 4.3.2

The proof follows the one from Cooper (1995, Theorem 3). Tesselate $[0,1]^d$ by cubes of side $s = \varepsilon/(2\sqrt{d})$ and denote by $n = (\lceil 1/s \rceil)^d$ the number of cubes in the tesselation. Choose n data points, one in each different cube. Then any Delaunay sphere will have a radius $R < \varepsilon/2M_f$. Now, construct $\tilde{f}$ by linearly interpolating between values of f over the Delaunay simplices. According to Seidel (1995), the number m_f of subdomains is $O(n^{d/2})$ and each of them is convex. Besides, by Cooper (1995, Lemma 2), $\tilde{f}$ guarantees an approximation error $\|f - \tilde{f}\|_\infty \leqslant \varepsilon$.

Using Corollary 4.3.1, we know that there exists a neural network of the form (4.2.1) verifying Assumption 2 and representing $\tilde{f}$. Besides, its depth is $2\lceil \log_2(m_f) \rceil + 1$ and its size is at most $3m_f^2 + m_f - 1$. Consequently, we have that the depth of the neural network is $2\lceil \log_2(m_f) \rceil + 1 = O(d^2 \log_2(\frac{2\sqrt{d}}{\varepsilon}))$ and the size at most $O(m^2) = O((\frac{2\sqrt{d}}{\varepsilon})^{d^2})$.

4.A.10 Proof of Proposition 4.3.3

Let $f \in \mathrm{Lip}_1([0,1])$ and f_m be the piecewise linear interpolation of f with the following 2^m+1 breakpoints: $k/2^m$, $k = 0, \dots, 2^m$. We know that the function f_m approximates f with an error $\varepsilon_m \leqslant 2^{-m}$. In particular, for any $m \geqslant \log_2(1/\varepsilon)$, we have $\varepsilon_m \leqslant \varepsilon$. Besides, for any m, f_m is a 1-Lipschitz function defined on $[0,1]$, piecewise linear on 2^m subdomains. Thus, according to Proposition 4.3.2, there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f_m, with depth $2m+1$ and size at most $3 \times 2^{2m} + 2^m - 1$. Taking $m = \lceil \log_2(1/\varepsilon) \rceil$ shows the desired result.

Let $\varepsilon > 0$, let f be a convex (or concave) function in $\mathrm{Lip}_1([0,1])$, and let f_m be the piecewise linear interpolation of f with the following 2^m+1 breakpoints: $k/2^m$, $k = 0, \dots, 2^m$. The function f_m approximates f with an error $\varepsilon_m = 2^{-m}$. In particular, for any $m \geqslant \log_2(1/\varepsilon)$, we have $\varepsilon_m \leqslant \varepsilon$. Besides, for any m, f_m is a 2^m-piecewise linear convex function defined on

$[0,1]$. Hence, by Proposition 4.3.2, there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f_m, with depth $m+1$ and size at most $2\times 2^m-1$. Taking $m=\lceil \log_2(1/\varepsilon)\rceil$ leads to the desired result.

4.A.11 Proof of Proposition 4.4.1

We prove the result by using an induction on n. The case where $n=1$ and thus $m=k^1$ is true since the function $f=\max(f_1,\dots,f_k)$ can be represented by a neural network of the form (4.2.1) with grouping size k, depth $q+1$, and size $s_1+\cdots+s_k+1$. Now, let $m=k^n$ with $n>1$. We have that $\lfloor m/k\rfloor=\lceil m/k\rceil=m/k=k^{n-1}$. Let $g_1=\max(f_1,\dots,f_{m/k}), g_2=\max(f_{m/k+1},\dots,f_{2m/k}),\dots,g_k=\max(f_{((k-1)m/k)+1},\dots,f_m)$. By the induction hypothesis, $g_1,\dots,g_k$ can all be represented by neural networks of the form (4.2.1) with grouping size k, width depths equal to $q+n-1$ and sizes at most $s_1+\cdots+s_{m/k}+\frac{k^{n-1}-1}{k-1},\dots,s_{(k-1)m/k+1}+\cdots+s_m+\frac{k^{n-1}-1}{k-1}$, respectively.

Consequently, the function $G(x)=(g_1(x),\dots,g_k(x))$ can be implemented by a neural network of the form (4.2.1) with grouping size k, depth $q+n-1$, and size at most $s_1+\cdots+s_m+m-2$. Finally, by concatenating a one neuron layer, we see that the function $f=\max(g_1,\dots,g_k)$ can be represented by a neural network of the form (4.2.1) with depth $q+n=q+\log_k(m)$ and size at most

$$s_1+\cdots+s_m+k\Big(\frac{k^{n-1}-1}{k-1}\Big)+1=s_1+\cdots+s_m+\frac{k^n-1}{k-1}=s_1+\cdots+s_m+\frac{m-1}{k-1}.$$

4.A.12 Proof of Corollary 4.4.1

According to He et al. (2018, Theorem A.1), the function f can be written as

$$f=\max_{1\leqslant k\leqslant m_f}\min_{i\in S_k}\ell_i,$$

where $|S_k|\leqslant m_f$ and $m_f=k^n$ for some $n\geqslant 1$. From Proposition 4.4.1, there exists a neural network verifying Assumption 2 with grouping size k representing $\min_{i\in S_k}\ell_i$ with depth $\log_k(m)+1$ and size at most $\frac{m_f-1}{k-1}$.

Using again Proposition 4.4.1, we find that there exists a neural network, verifying Assumption 2, with grouping size k, representing f with depth $2\log_k(m_f)+1$ and size at most

$$m_f\Big(\frac{m_f-1}{k-1}\Big)+\frac{m_f-1}{k-1}=\frac{m_f^2-1}{k-1}.$$

4.A.13 Proof of Lemma 4.4.1

The result is proved by induction on q. To begin with, in the case $q = 2$ we have a neural network with one hidden layer. When applying the GroupSort function with a grouping size k, the maximum number of breakpoints is equal to the maximum number of intersections of linear functions. In each group of k functions, there are at most $\frac{k(k-1)}{2}$ intersections. Thus, there are at most $\frac{k(k-1)}{2} \times \frac{v_1}{k} = \frac{(k-1)v_1}{2}$ breakpoints, that is $\frac{(k-1)v_1}{2} + 1$ pieces.

Now, let us assume that the property is true for a given $q \geqslant 3$. Consider a neural network with depth q and widths $v_1, \dots, v_{q-1}$. Observe that the input to any node in the last layer is the output of a $\mathbb{R} \to \mathbb{R}$ GroupSort neural network with depth $(q-1)$ and widths $v_1, \dots, v_{q-2}$. Using the induction hypothesis, the input to this node is a function from $\mathbb{R} \to \mathbb{R}$ with at most $k^{q-3} \times (\frac{(k-1)v_1}{2} + 1) \times \cdots \times v_{q-2}$ pieces. Thus, after applying the GroupSort function with a grouping size k, each node output is a function with at most $k \times (k^{q-3} \times (\frac{(k-1)v_1}{2} + 1) \times v_2 \times \cdots \times v_{q-2})$. With the final layer, we take an affine combination of v_{q-1} functions, each with at most $k^{q-2} \times (\frac{(k-1)v_1}{2} + 1) \times v_2 \times \cdots \times v_{q-2}$ pieces. In all, we therefore get at most $k^{q-2} \times (\frac{(k-1)v_1}{2} + 1) \times v_2 \times \cdots \times v_{q-1}$ pieces. The induction step is completed.

4.A.14 Proof of Theorem 4.4.1

The proof of Theorem 4.4.1 is straightforward and follows the one of Theorem 4.3.2 combined with the result obtained in Corollary 4.4.1.

4.A.15 Proof of Proposition 4.4.2

Let $f \in \mathrm{Lip}_1([0,1])$ and f_m be the piecewise linear interpolation of f with the following $k^n + 1$ breakpoints: i/k^n, $k = 0, \dots, k^n$. We know that the function f_m approximates f with an error $\varepsilon_m \leqslant k^{-n}$. In particular, for any $n \geqslant \log_k(1/\varepsilon)$, we have $\varepsilon_n \leqslant \varepsilon$. Besides, for any n, f_{k^n} is a 1-Lipschitz function defined on $[0,1]$, piecewise linear on k^n subdomains. Thus, according to Corollary 4.4.1, there exists a neural network of the form (4.2.1), verifying Assumption 2 and representing f_{k^n}, with grouping size k, depth $2n+1$, and size at most $\frac{k^{2n}-1}{k-1}$. Taking $n = \lceil \log_k(1/\varepsilon) \rceil$ shows the desired result.

Appendix 4.B Complementary experiments

4.B.1 Extended comparison between GroupSort and ReLU networks

We provide in this section further results and details on the experiments ran in Section 4.5.

4.B.1.1 Task 1: Approximating functions

Piecewise linear functions. We complete the experiments of Section 4.5 by estimating the 6-piecewise linear function f in the model $Y = f(X)$ (noiseless case, see Figure 4.7 and Figure 4.8) and in the model $Y = f(X) + \varepsilon$ (noisy case, see Figure 4.9 and Figure 4.10). Recall that in both cases, X follows a uniform distribution on $[-1.5, 1.5]$ and the sample size is $n = 100$.

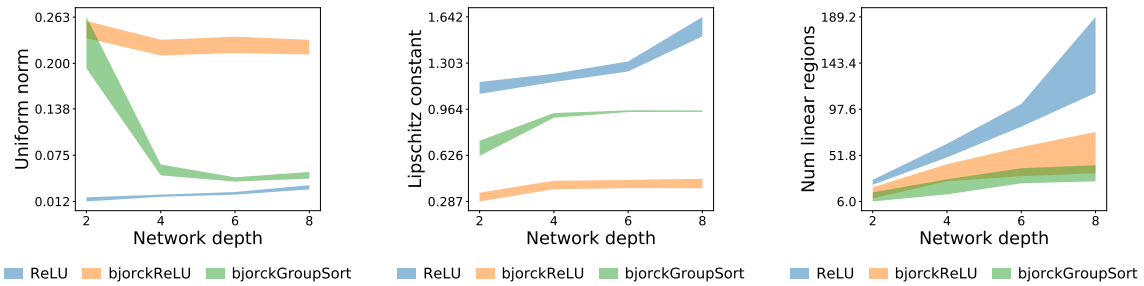
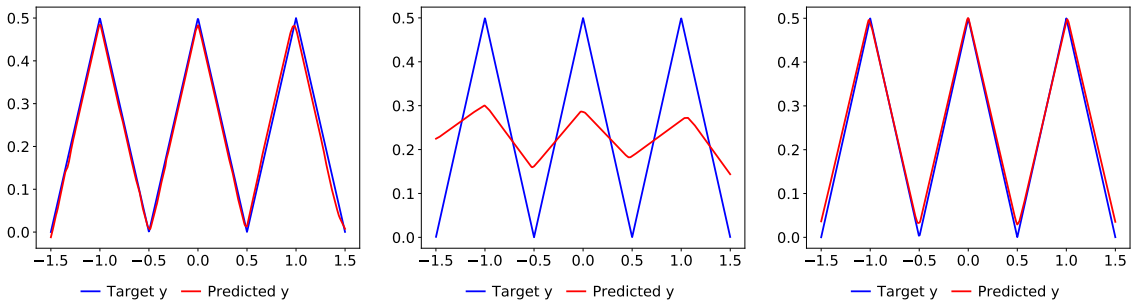


Fig. 4.7 Estimating the 6-piecewise linear function in the model $Y = f(X)$, with a dataset of size $n = 100$ (the thickness of the line represents a 95%-confidence interval).



(a) ReLU (b) bjorckReLU (c) bjorckGroupSort

Fig. 4.8 Reconstructing the 6-piecewise linear function in the model $Y = f(X)$, with a dataset of size $n = 100$.

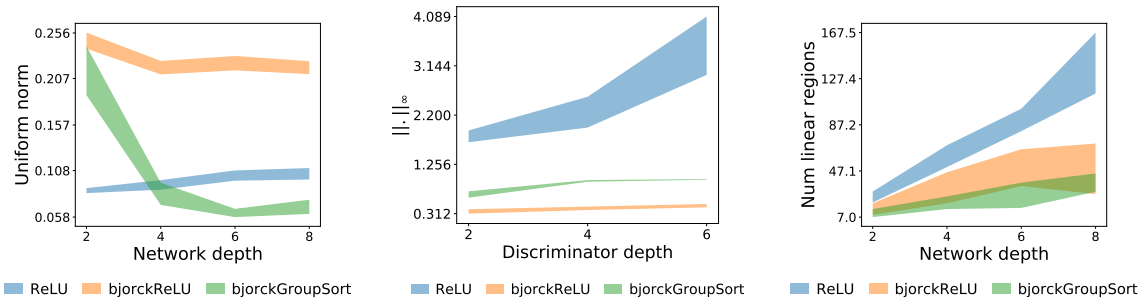
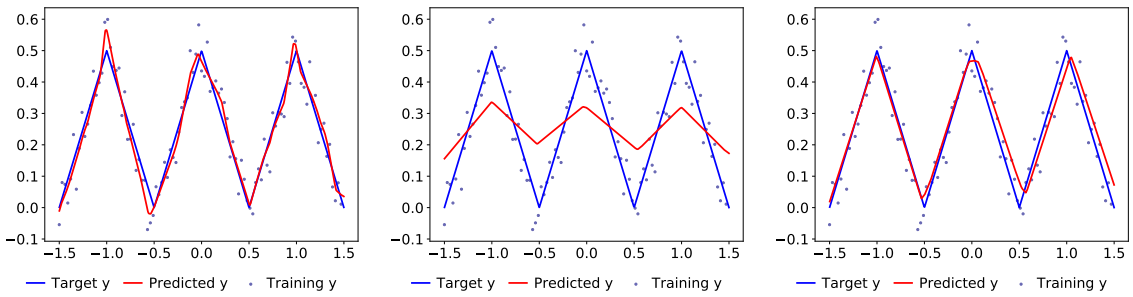


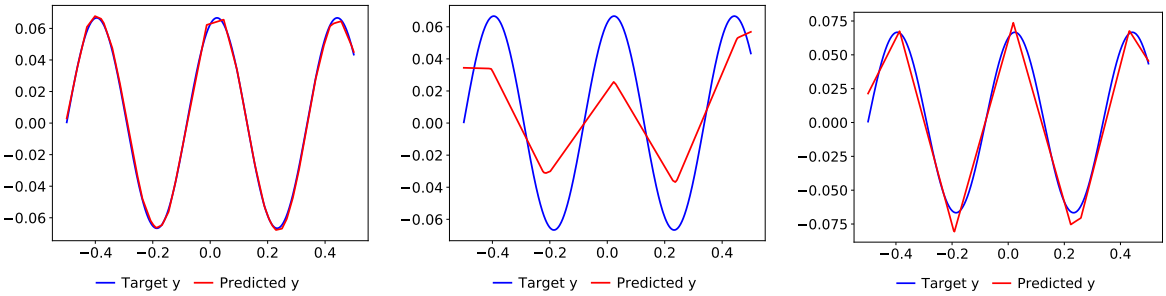
Fig. 4.9 Estimating the 6-piecewise linear function in the model $Y = f(X) + \varepsilon$, with a dataset of size $n = 100$ (the thickness of the line represents a 95%-confidence interval).



(a) ReLU (b) bjorckReLU (c) bjorckGroupSort

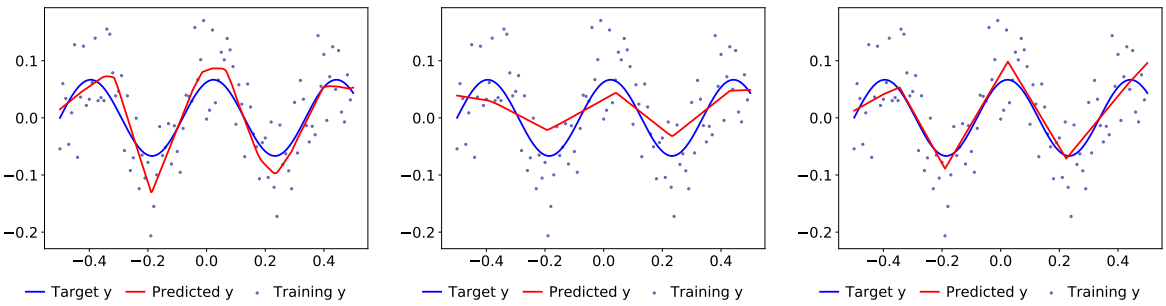
Fig. 4.10 Reconstructing the 6-piecewise linear function in the model $Y = f(X) + \varepsilon$, with a dataset of size $n = 100$.

The sinus function. We provide in this subsection additional details for the learning of the sinus function $f(x) = (1/15)\sin(15x)$ defined on $[0,1]$ (see Section 4.5). Figure 4.11 is the case without noise while Figure 4.12 is the case with noise.



(a) ReLU (b) bjorckReLU (c) bjorckGroupSort

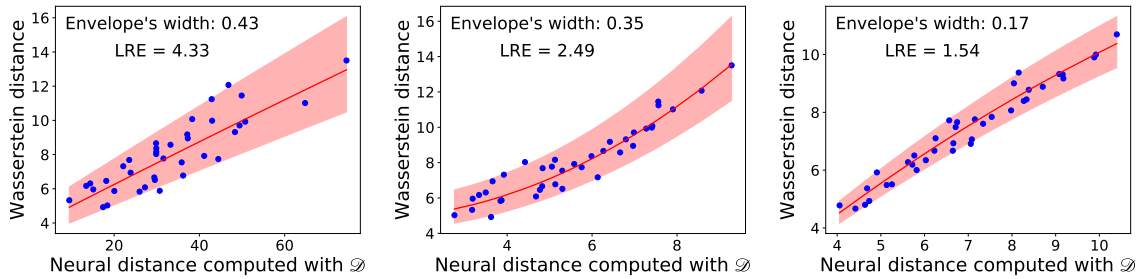
Fig. 4.11 Reconstructing the function $f(x) = (1/15)\sin(15x)$ in the model $Y = f(X)$, with a dataset of size $n = 100$.



(a) ReLU (b) bjorckReLU (c) bjorckGroupSort

Fig. 4.12 Reconstructing the function $f(x) = (1/15)\sin(15x)$ in the model $Y = f(X) + \varepsilon$, with a dataset of size $n = 100$.

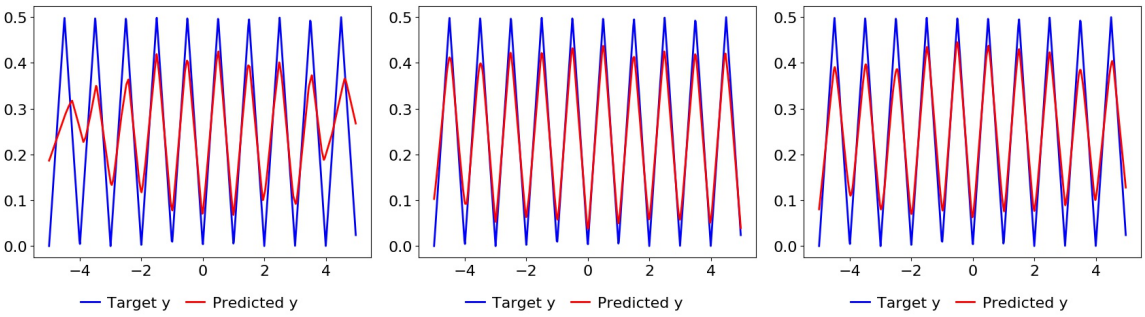
4.B.1.2 Task 2: Calculating Wasserstein distances



(a) $\mathscr{D}$ = ReLU network (b) $\mathscr{D}$ = bjorckReLU network (c) $\mathscr{D}$ = bjorckGroupSort network

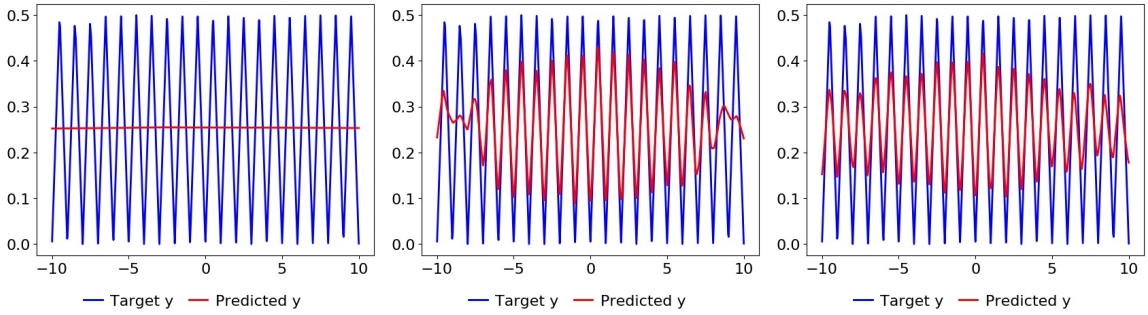
Fig. 4.13 Scatter plots of 40 pairs of Wasserstein and neural distances, for $q = 2$. The underlying distributions are bivariate Gaussian distributions with 4 components. The red curve is the optimal parabolic fitting and LRE refers to the Least Relative Error. The red zone is the envelope obtained by stretching the optimal curve.

4.B.2 Impact of the grouping size



(a) Grouping size = 2 (b) Grouping size = 5 (c) Grouping size = 10

Fig. 4.14 Reconstruction of a 20-piecewise linear function with varying grouping sizes ($k = 2, 5, 10$).



(a) Grouping size = 2 (b) Grouping size = 5 (c) Grouping size = 10

Fig. 4.15 Reconstruction of a 40-piecewise linear function with varying grouping sizes ($k = 2, 5, 10$).

4.B.3 Architecture for both GroupSort and ReLU networks

Operation	Feature Maps	Activation
$D(x)$		
Fully connected - q layers	width w	{GroupSort, ReLU}
Width w	{50}	
Depth q	{2, 4, 6, 8}	
Batch size	256	
Learning rate	0.0025	
Optimizer	Adam: $\beta_1 = 0.5$	$\beta_2 = 0.5$

Table 4.2 Hyperparameters used for the training of all neural networks

Chapter 5

Learning disconnected manifolds: a no GAN's land

Abstract

Typical architectures of Generative Adversarial Networks make use of a unimodal latent/input distribution transformed by a continuous generator. Consequently, the modeled distribution always has connected support which is cumbersome when learning a disconnected set of manifolds. We formalize this problem by establishing a "no free lunch" theorem for the disconnected manifold learning stating an upper-bound on the precision of the targeted distribution. This is done by building on the necessary existence of a low-quality region where the generator continuously samples data between two disconnected modes. Finally, we derive a rejection sampling method based on the norm of generator's Jacobian and show its efficiency on several generators including BigGAN.

Contents

5.1 Introduction **140**

5.2 Related work **142**

5.3 Our approach **143**

5.4 Experiments **150**

5.5 Conclusion and future work **155**

Appendix 5.A Technical results **158**

Appendix 5.B Complementary experiments **169**

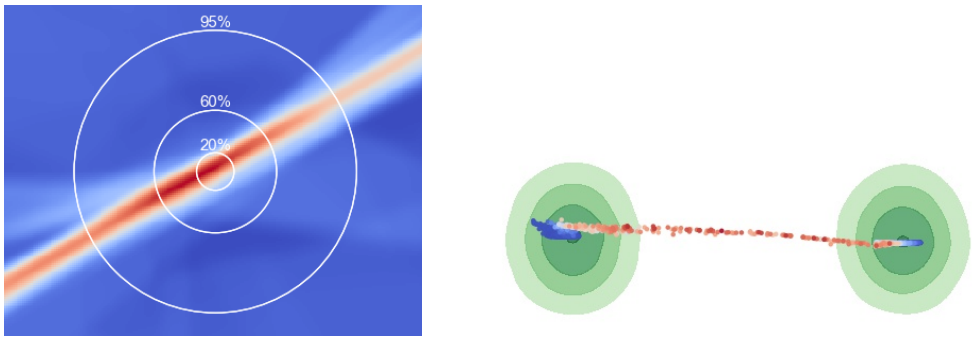
Appendix 5.C Supplementary details **177**

5.1 Introduction

GANs Goodfellow et al. (2014) provide a very effective tool for the unsupervised learning of complex probability distributions. For example, Karras et al. (2019) generate very realistic human faces while Yu et al. (2017) match state-of-the-art text corpora generation. Despite some early theoretical results on the stability of GANs Arjovsky and Bottou (2017) and on their approximation and asymptotic properties Biau et al. (2020), their training remains challenging. More specifically, GANs raise a mystery formalized by Khayatkhoei et al. (2018): *how can they fit disconnected manifolds when they are trained to continuously transform a unimodal latent distribution?* While this question remains widely open, we will show that studying it can lead to some improvements in the sampling quality of GANs. Indeed, training a GAN with the objective of continuously transforming samples from an unimodal distribution into a disconnected requires balancing between two caveats. On one hand, the generator could just ignore all modes but one, producing a very limited variety of high quality samples: this is an extreme case of the well known mode collapse Arjovsky and Bottou (2017). On the other hand, the generator could cover the different modes of the target distribution and necessarily generates samples out of the real data manifold as previously explained by Khayatkhoei et al. (2018).

As brought to the fore by Roth et al. (2017), there is a density mis-specification between the true distribution and the model distribution. Indeed, one cannot find parameters such that the model density function is arbitrarily close to the true distribution. To solve this issue, many empirical works have proposed to over-parameterize the generative distributions, as for instance, using a mixture of generators to better fit the different target modes. Tolstikhin et al. (2017) rely on boosting while Khayatkhoei et al. (2018) force each generator to target different sub-manifolds thanks to a criterion based on mutual information. Another direction is to add complexity in the latent space using a mixture of Gaussian distributions Gurumurthy et al. (2017).

To better visualize this phenomenon, we consider a simple 2D motivational example where the real data lies on two disconnected manifolds. Empirically, when learning the distribution, GANs split the Gaussian latent space into two modes, as highlighted by the separation line in red in Figure 5.1a. More importantly, each sample drawn inside this red area in Figure 5.1a is then mapped in the output space in between the two modes (see Figure 5.1b). For the quantitative evaluation of the presence of out-of-manifold samples, a natural metric is the Precision-Recall (PR) proposed by Sajjadi et al. (2018) and its improved version (Improved



(a) Heatmap of the generator's Jacobian norm. White circles: quantiles of the latent distribution $\mathcal{N}(0, I)$.

(b) Green: target distribution. Coloured dots: generated samples colored w.r.t. the Jacobian Norm using same heatmap than (a).

Fig. 5.1 Learning disconnected manifolds leads to the apparition of an area with high gradients and data sampled in between modes.

PR) (Kynkäänniemi et al., 2019). A first contribution of this paper is to formally link them. Then, taking advantage of these metrics, we lower bound the measure of this out-of-manifold region and formalize the impossibility of learning disconnected manifolds with standard GANs. We also extend this observation to the multi-class generation case and show that the volume of off-manifold areas increases with the number of covered manifolds. In the limit, this increase drives the precision to zero.

To solve this issue and increase the precision of GANs, we argue that it is possible to remove out-of-manifold samples using a truncation method. Building on the work of Arvanitidis et al. (2017) who define a Riemaniann metric that significantly improves clustering in the latent space, our truncation method is based on information conveyed by the Jacobian's norm of the generator. We empirically show that this rejection sampling scheme enables us to better fit disconnected manifolds without over-parametrizing neither the generative class of functions nor the latent distribution. Finally, in a very large high dimensional setting, we discuss the advantages of our rejection method and compare it to the truncation trick introduced by Brock et al. (2019).

In a nutshell, our contributions are the following:

- We discuss evaluation of GANs and formally link the PR measure Sajjadi et al. (2018) and its Improved PR version Kynkäänniemi et al. (2019).
- We upper bound the precision of GANs with Gaussian latent distribution and formalize an impossibility result for disconnected manifolds learning.
- Using toy datasets, we illustrate the behavior of GANs when learning disconnected manifolds and derive a new truncation method based on the Jacobian's Frobenius norm

of the generator. We confirm its empirical performance on state-of-the-art models and datasets.

5.2 Related work

Fighting mode collapse. Goodfellow et al. (2014) were the first to raise the problem of mode collapse in the learning of disconnected manifolds with GANs. They observed that when the generator is trained too long without updating the discriminator, the output distribution collapses to a few modes reducing the diversity of the samples. To tackle this issue, Salimans et al. (2016); Lin et al. (2018) suggested feeding several samples to the discriminator. Srivastava et al. (2017) proposed the use of a reconstructor network, mapping the data to the latent space to increase diversity.

In a different direction, Arjovsky and Bottou (2017) showed that training GANs using the original formulation Goodfellow et al. (2014) leads to instability or vanishing gradients. To solve this issue, they proposed a Wasserstein GAN architecture Arjovsky et al. (2017) where they restrict the class of discriminative functions to 1-Lipschitz functions using weight clipping. Pointing to issues with this clipping, Gulrajani et al. (2017); Miyato et al. (2018) proposed relaxed ways to enforce the Lipschitzness of the discriminator, either by using a gradient penalty or a spectral normalization. Albeit not exactly approximating the Wasserstein's distance (Petzka et al., 2018), both implementations lead to good empirical results, significantly reducing mode collapse. Building on all of these works, we will further assume that generators are now able to cover most of the modes of the target distribution, leaving us the problem of out-of-manifold samples (*a.k.a.* low-quality pictures).

Generation of disconnected manifolds. When learning complex manifolds in high dimensional spaces using deep generative models, Fefferman et al. (2016) highlighted the importance of understanding the underlying geometry. More precisely, the learning of disconnected manifold requires the introduction of disconnectedness in the model. Gurumurthy et al. (2017) used a multi-modal entry distribution, making the latent space disconnected, and showed better coverage when data is limited and diverse. Alternatively, Khayatkhoei et al. (2018) studied the learning of a mixture of generators. Using a mutual information term, they encourage each generator to focus on a different submanifold so that the mixture covers the whole support. This idea of using an ensemble of generators is also present in the work of Tolstikhin et al. (2017) and Zhong et al. (2019), though they were primarily interested in the reduction of mode collapse.

In this paper, we propose a truncation method to separate the latent space into several disjoint areas. It is a way to learn disconnected manifolds without relying on the previously introduced over-parameterization techniques. As our proposal can be applied without retraining the whole architecture, we can use it successfully on very larges nets. Close to this idea, Azadi et al. (2019) introduced a rejection strategy based on the output of the discriminator. However, this rejection sampling scheme requires the discriminator to be trained with a classification loss while our proposition can be applied to any generative models.

Evaluating GANs. The evaluation of generative models is an active area of research. Some of the proposed metrics only measure the quality of the generated samples such as the Inception score Salimans et al. (2016) while others define distances between probability distributions. This is the case of the Frechet Inception distance Heusel et al. (2017), the Wasserstein distance Arjovsky et al. (2017) or kernel-based metrics Gretton et al. (2012). The other main caveat for evaluating GANs lies in the fact that one does not have access to the true density nor the model density, prohibiting the use of any density based metrics. To solve this issue, the use of a third network that acts as an objective referee is common. For instance, the Inception score uses outputs from InceptionNet while the Fréchet Inception Distance compares statistics of InceptionNet activations. Since our work focuses on out-of-manifold samples, a natural measure is the PR measure (Sajjadi et al., 2018) and its Improved PR version (Kynkäänniemi et al., 2019), extensively discussed in the next section.

In the following, alongside precise definitions, we exhibit an upper bound on the precision of GANs with high recall (*i.e.* no mode collapse) and present a new truncation method.

5.3 Our approach

We start with a formal description of the framework of GANs and the relevant metrics. We later show a "no free lunch" theorem proving the necessary existence of an area in the latent space that generates out-of-manifold samples. We name this region the *no GAN's land* since any data point sampled from this area will be in the frontier in between two different modes. We claim that dealing with it requires special care. Finally, we propose a rejection sampling procedure to avoid points out of the true manifold.

5.3.1 Notations

In the original setting of Generative Adversarial Networks (GANs), one tries to generate data that are "similar" to samples collected from some unknown probability measure $\mu_\star$. To

do so, we use a parametric family of generative distribution where each distribution is the push-forward measure of a latent distribution Z and a continuous function modeled by a neural network.

Assumption 3 (Z Gaussian)**.** *The latent distribution Z is a standard multivariate Gaussian.*

Note that for any distribution μ, S_μ refers to its support. Assumption 3 is common for GANs as in many practical applications, the random variable Z defined on a low dimensional space $\mathbb{R}^d$ is either a multivariate Gaussian. Practicioners also studied distribution or uniform distribution defined on a compact.

The measure $\mu_\star$ is defined on a subset E of $\mathbb{R}^D$ (potentially a highly dimensional space), equipped with the norm $\|\cdot\|$. The generator has the form of a parameterized class of functions from $\mathbb{R}^d$ (a space with a much lower dimension) to E, say $\mathscr{G} = \{G_\theta : \theta \in \Theta\}$, where $\Theta \subseteq \mathbb{R}^p$ is the set of parameters describing the model. Each function G_θ thus takes input from a d-dimensional random variable Z (Z is associated with probability distribution γ) and outputs "fake" observations with distribution μ_θ. Thus, the class of probability measures $\mathscr{P} = \{\mu_\theta : \theta \in \Theta\}$ is the natural class of distributions associated with the generator, and the objective of GANs is to find inside this class of candidates the one that generates the most realistic samples, closest to the ones collected from the unknown distribution $\mu_\star$.

Assumption 4. *Let $L > 0$. The generator G_θ takes the form of a neural network whose Lipchitz constant is smaller than L, i.e. for all (z, z'), we have $\|G_\theta(z') - G_\theta(z)\| \leqslant L\|z - z'\|$.*

This is a reasonable assumption, since Virmaux and Scaman (2018) present an algorithm that upper-bounds the Lipschitz constant of deep neural networks. Initially, 1-Lipschitzness was enforced only for the discriminator by clipping the weigths Arjovsky et al. (2017), adding a gradient penalty Gulrajani et al. (2017); Roth et al. (2017); Petzka et al. (2018), or penalizing the spectral norms Miyato et al. (2018). Nowadays, state-of-the-art architectures for large scale generators such as SAGAN Zhang et al. (2019) and BigGAN Brock et al. (2019) also make use of spectral normalization for the generator.

5.3.2 Evaluating GANs with Precision and Recall

When learning disconnected manifolds, Srivastava et al. (2017) proved the need of measuring simultaneously the quality of the samples generated and the mode collapse. Sajjadi et al. (2018) proposed the use of a PR metric to measure the quality of GANs. The key intuition is that precision should quantify how much of the fake distribution can be generated by the true distribution while recall measures how much of the true distribution can be re-constructed by the model distribution. More formally, it is defined as follows:

Definition 5.3.1. *Sajjadi et al. (2018) Let X, Y be two random variables. For $\alpha, \beta \in (0,1]$, X is said to have an attainable precision α at recall β w.r.t. Y if there exists probability distributions μ, ν_X, ν_Y such that*

$$Y = \beta\mu + (1-\beta)\nu_Y \quad \textit{and} \quad X = \alpha\mu + (1-\alpha)\nu_X.$$

The component ν_Y denotes the part of Y that is "missed" by X, whereas, ν_X denotes the "noise" part of X. We denote $\bar{\alpha}$ (respectively $\bar{\beta}$) the maximum attainable precision (respectively recall). Th. 1 of Sajjadi et al. (2018) states:

$$X(S_Y) = \bar{\alpha} \quad \text{and} \quad Y(S_X) = \bar{\beta}.$$

Improved PR metric. Kynkäänniemi et al. (2019) highlighted an important drawback of the PR metric proposed by Sajjadi et al. (2018): it cannot correctly interpret situations when a large numbers of samples are packed together. To better understand this situation, consider a case where the generator slightly collapses on a specific data point, i.e. there exists $x \in E, \mu_\theta(x) > 0$. We show in Appendix 5.A.1 that if $\mu_\star$ is a non-atomic probability measure and μ_θ is highly precise (*i.e.* $\alpha = 1$), then the recall β must be 0.

To solve these issues, Kynkäänniemi et al. (2019) proposed an *Improved Precision-Recall* (Improved PR) metric built on a nonparametric estimation of support of densities.

Definition 5.3.2. *Kynkäänniemi et al. (2019) Let X, Y be two random variables and D_X, D_Y two finite sample datasets such that $D_X \sim X^n$ and $D_Y \sim Y^n$. For any $x \in D_X$ (respectively for any $y \in D_Y$), we consider $(x_{(1)}, \ldots, x_{(n-1)})$, the re-ordering of elements in $D_X \setminus x$ given their euclidean distance with x. For any $k \in \mathbb{N}$ and $x \in D_X$, the precision $\alpha_k^n(x)$ of point x is defined as:*

$$\alpha_k^n(x) = 1 \iff \exists y \in D_Y, \|x-y\| \leqslant \|y_{(k)} - y\|.$$

Similarly, the recall $\beta_k^n(y)$ of any given $y \in D_Y$ is:

$$\beta_k^n(y) = 1 \iff \exists x \in D_X, \|y-x\| \leqslant \|x_{(k)} - x\|.$$

Improved precision (respectively recall) are defined as the average over D_X (respectively D_Y) as follows:

$$\alpha_k^n = \frac{1}{n} \sum_{x_i \in D_X} \alpha_k^n(x_i) \qquad \beta_k^n = \frac{1}{n} \sum_{y_i \in D_Y} \beta_k^n(y_i).$$

A first contribution is to formalize the link between PR and Improved PR with the following theorem:

Theorem 5.3.1. *Let X, Y two random variables with probability distributions μ and ν. Assume that both μ and ν are associated with uniformly continuous probability density functions f_μ and f_ν. Besides, there exists constants $a_1 > 0, a_2 > 0$ such that for all $x \in E$ we have $a_1 < f_{\mu_\star}(x) \leqslant a_2$ and $a_1 < f_{\mu_\theta}(x) \leqslant a_2$ for some $c > 0$. Also, (k,n) are such that $\frac{k}{\log(n)} \to +\infty$ and $\frac{k}{n} \to 0$. Then,*

$$\alpha_k^n \to \bar{\alpha} \text{ in probability} \quad \text{and} \quad \beta_k^n \to \bar{\beta} \text{ in probability}.$$

This theorem, whose proof is delayed to Appendix 5.A.2, underlines the nature of the Improved PR metric: the metric compares the supports of the modeled probability distribution μ_θ and of the true distribution $\mu_\star$. This means that Improved PR is a tuple made of both maximum attainable precision $\bar{\alpha}$ and recall $\bar{\beta}$ (e.g. Theorem 1 of Sajjadi et al. (2018)). As Improved PR is shown to have a better performance evaluating GANs sample quality, we use this metric for both the following theoretical results and experiments.

5.3.3 Learning disconnected manifolds

In this section, we aim to stress the difficulties of learning disconnected manifolds with standard GANs architectures. To begin with, we recall the following lemma.

Lemma 5.3.1. *Assume that Assumptions 3 and 4 are satisfied. Then, for any $\theta \in \Theta$, the support S_{μ_θ} is connected.*

There is consequently a discrepancy between the connectedness of S_{μ_θ} and the disconnectedness of $S_{\mu_\star}$. In the case where the manifold lays on two disconnected components, our next theorem exhibit a no free lunch theorem:

Theorem 5.3.2. *("No free lunch" theorem) Assume that Assumptions 3 and 4 are satisfied. Assume also that true distribution $\mu_\star$ lays on two equally measured disconnected manifolds distant from a distance $D > 0$. Then, any estimator μ_θ that samples equally in both modes must have a precision $\bar{\alpha}$ such that $\bar{\alpha} + \frac{D}{\sqrt{2\pi}L} e^{\frac{-\Phi^{-1}(\frac{\bar{\alpha}}{2})^2}{2}} \leqslant 1$, where Φ is the c.d.f. of a standard normal distribution.*

Besides, if $\bar{\alpha} \geqslant 3/4$, $\bar{\alpha} \lesssim 1 - \sqrt{\frac{2}{\pi}} W(\frac{D^2}{4L^2})$ where W is the Lambert W function.

The proof of this theorem is delayed to Appendix 5.A.3. It is mainly based on the Gaussian isoperimetric inequality Borell (1975); Sudakov and Tsirelson (1978) that states that among all sets of given Gaussian measure in any finite dimensional Euclidean space, half-spaces have the minimal Gaussian boundary measure. If in Fig. 5.1, the generator has thus learned the optimal

separation, it is yet not known, to the limit of our knowledge, how to enforce such geometrical properties in the latent space.

In real world applications, when the number of distinct sub-manifolds increases, we expect the volume of these boundaries to increase with respect to the number of different classes covered by the modeled distribution μ_θ. Going in this direction, we better formalize this situation, and show an extended "no free lunch theorem" by expliciting an upper-bound of the precision $\bar{\alpha}$ in this broader framework.

Assumption 5. *The true distribution $\mu_\star$ lays on M equally-measured disconnected components at least distant from some constant $D > 0$.*

This is likely to be true for datasets made of symbol designed to be highly distinguishable (*e.g.* digits in the MNIST dataset). In very high dimension, this assumption also holds for complex classes of objects appearing in many different contexts (*e.g.* the bubble class in ImageNet, see Appendix).

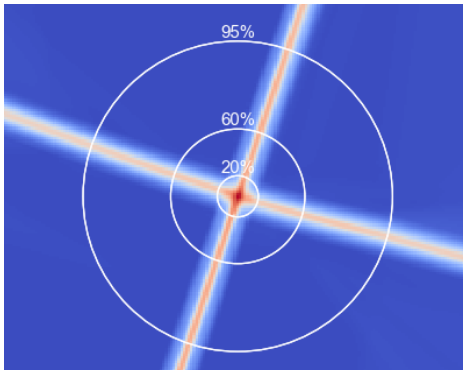
To better apprehend the next theorem, note A_m the pre-image in the latent space of mode m and A_m^r its r-enlargement: $A_m^r := \{z \in \mathbb{R}^d \mid \mathsf{dist}(\mathsf{z}, \mathsf{A_m}) \leq \mathsf{r}\}, \mathsf{r} > 0$.

Theorem 5.3.3. *(Generalized "no free lunch" theorem) Assume that Assumptions 3, 4, and 5 are satisfied, and that the pre-image enlargements A_m^ε, with $\varepsilon = \frac{D}{2L}$, form a partition of the latent space with equally measured elements.*

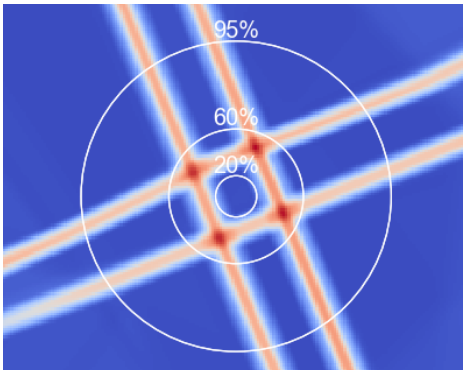
Then, any estimator μ_θ with recall $\bar{\beta} > \frac{1}{M}$ must have a precision $\bar{\alpha}$ at most $\frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x}$ where $x = \Phi^{-1}(1 - \frac{1}{\bar{\beta}M})$ and Φ is the c.d.f. of a standard normal distribution.

Theorem 5.3.3, whose proof is delayed to Appendix 5.A.4, states a lower-bound the measure of samples mapped out of the true manifold. We expect our bound to be loose since no theoretical results are known, to the best of our knowledge, on the geometry of the separation that minimizes the boundary between different classes (when $M \geqslant 3$). Finding this optimal cut would be an extension of the honeycomb theorem Hales (2001). In Appendix 5.A.4.2 we give a more technical statement of Theorem 5.3.3 without assuming equality of measure of the sets A_m^ε.

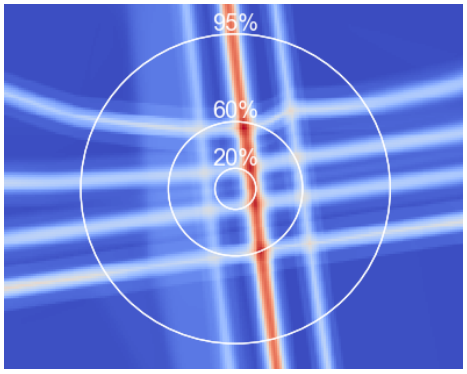
The idea of the proof is to consider the border of an individual cell with the rest of the partition. It is clear that at least half of the frontier will be inside this specific cell. Then, to get to the final result, we sum the measures of the frontiers contained inside all of the different cells. Remark that our analysis is fine enough to keep a dependency in M which translates into a maximum precision that goes to zero when M goes to the infinity and all the modes are covered. More precisely, in this scenario where all pre-images have equal measures in the latent



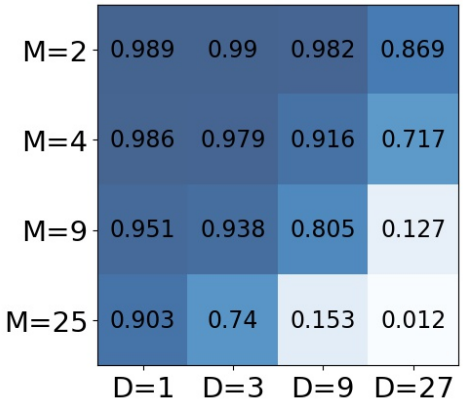
(a) WGAN 4 classes: visualisation of $\|J_G(z)\|_F$.



(b) WGAN 9 classes: visualisation of $\|J_G(z)\|_F$.



(c) WGAN 25 classes: visualisation of $\|J_G(z)\|_F$.



(d) Precision w.r.t. D (mode distance) and M (classes).

Fig. 5.2 Illustration of Theorem 5.3.3. If the number of classes $M \to \infty$ or the distance $D \to \infty$, then the precision $\bar{\alpha} \to 0$. We provide in appendix heatmaps for more values of M.

space, one can derive the following bound, when the recall $\bar{\beta}$ is kept fixed and M increases:

$$\bar{\alpha} \overset{M\to\infty}{\leqslant} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon\sqrt{2\log(\bar{\beta}M)}} \quad \text{where } \varepsilon = \frac{D}{2L}. \tag{5.3.1}$$

For a fixed generator, this equation illustrates that the precision $\bar{\alpha}$ decreases when either the distance D (equivalently ε) or the number of classes M increases. For a given ε, $\bar{\alpha}$ converges to 0 with a speed $O(\frac{1}{(\bar{\beta}M)^{\sqrt{2}\varepsilon}})$. To better illustrate this asymptotic result, we provide results from a 2D synthetic setting. In this toy dataset, we control both the number M of disconnected manifolds and the distance D. Figure 5.2 clearly corroborates (5.3.1) as we can easily get the maximum precision close to 0 ($M = 25$, $D = 27$).

5.3.4 Jacobian-based truncation (JBT) method

The analysis of the deformation of the latent space offers a grasp on the behavior of GANs. For instance, Arvanitidis et al. (2017) propose a distance accounting for the distortions made by the generator. For any pair of points $(z_1, z_2) \sim Z^2$, the distance is defined as the length of the geodesic $d(z_1, z_2) = \int_{[0,1]} \|J_{G_\theta}(\gamma_t)\frac{d\gamma_t}{dt}\| dt$ where γ is the geodesic parameterized by $t \in [0,1]$ and $J_{G_\theta}(z)$ denotes the Jacobian matrix of the generator at point z. Authors have shown that the use of this distance in the latent space improves clustering and interpretability. We make a similar observation that the generator's Jacobian Frobenius norm provides meaningful information.

Indeed, the frontiers highlighted in Figures 5.2a, 5.2b, and 5.2c correspond to areas of low precision mapped out of the true manifold: this is the *no GAN's land*. We argue that when learning disconnected manifolds, the generator tries to minimize the number of samples that do not belong to the support of the true distribution and that this can only be done by making paths steeper in the *no GAN's land*. Consequently, data points $G_\theta(z)$ with high Jacobian Frobenius norm (JFN) are more likely to be outside the true manifold. To improve the precision of generative models, we thus define a new truncation method by removing points with highest JFN.

However, note that computing the generators's JFN is expensive to compute for neural networks, since being defined as follows,

$$\|J_{G_\theta}(z)\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{\partial G_\theta(z)_i}{\partial z_j}\right)^2,$$

it requires a number of backward passes equal to the output dimension. To make our truncation method tractable, we use a stochastic approximation of the Jacobian Frobenius norm based on

the following result from Rifai et al. (2011):

$$\|J_{G_\theta}(z)\|^2 = \lim_{\substack{N\to\infty \\ \sigma\to 0}} \frac{1}{N}\sum_{\varepsilon_i}^{N} \frac{1}{\sigma^2}\|G_\theta(z+\varepsilon_i) - G_\theta(z)\|^2,$$

where $\varepsilon_i \sim\sim \mathcal{N}(0, \sigma^2 I$ and I is the identity matrix of dimension d. The variance σ of the noise and the number of samples are used as hyper-parameters. In practice, σ in $[1e{-}4; 1e{-}2]$ and $N = 10$ give consistent results.

Based on the preceding analysis, we propose a new **Jacobian-based truncation** (JBT) method that rejects a certain ratio of the generated points with highest JFN. This truncation ratio is considered as an hyper-parameter for the model. We show in our experiments that our JBT can be used to to detect samples outside the real data manifold and that it consequently improves the precision of the generated distribution as measured by the Improved PR metric.

5.4 Experiments

In the following, we show that our truncation method, JBT, can significantly improve the performances of generative models on several models, metrics and datasets. Furthermore, we compare JBT with over-parametrization techniques specifically designed for disconnected manifold learning. We show that our truncation method reaches or surpasses their performance, while it has the benefit of not modifying the training process of GANs nor using a mixture of generators, which is computationally expensive. Finally, we confirm the efficiency of our method by applying it on top of BigGAN Brock et al. (2019).

Except for BigGAN, for all our experiments, we use Wasserstein GAN with gradient penalty Gulrajani et al. (2017), called WGAN for conciseness. We give in Appendix 5.C the full details of our experimental setting. The use of WGAN is motivated by the fact that it was shown to stabilize the training and significantly reduce mode collapse Arjovsky and Bottou (2017). However, we want to emphasise that our method can be plugged on top of any generative model fitting disconnected components.

5.4.1 Evaluation metrics

To measure performances of GANs when dealing with low dimensional applications - as with synthetic datasets - we equip our space with the standard Euclidean distance. However, for high dimensional applications such as image generation, Brock et al. (2019); Kynkäänniemi et al. (2019) have shown that embedding images into a feature space with a pre-trained convolutional classifier provides more semantic information. In this setting, we consequently use the euclidean

distance between the images' embeddings from a classifier. For a pair of images (a,b), we define the distance $d(a,b)$ as $d(a,b) = \|\phi(a) - \phi(b)\|_2$ where ϕ is a pre-softmax layer of a supervised classifier, trained specifically on each dataset. Doing so, they will more easily separate images sampled from the true distribution $\mu_\star$ from the ones sampled by the distribution μ_θ.

We compare performances using Improved PR Kynkäänniemi et al. (2019). We also report the *Marginal Precision* which is the precision of newly added samples when increasing the ratio of kept samples. Besides, for completeness, we report FID Heusel et al. (2017) and recall precise definitions in Appendix 5.B.2. Note that FID was not computed with InceptionNet, but a classifier pre-trained on each dataset.

5.4.2 Synthetic dataset

We first consider the true distribution to be a 2D Gaussian mixture of 9 components. Both the generator and the discriminator are modeled with feed-forward neural networks.

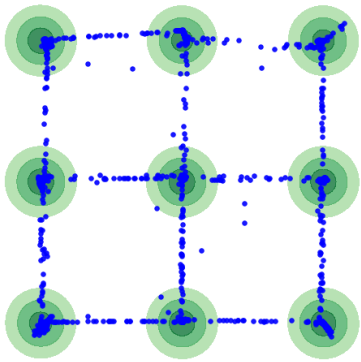
Interestingly, the generator tries to minimize the sampling of off-manifolds data during training until its JFN gets saturated (see Appendix 5.B.3). One way to reduce the number of off-manifold samples is to use JBT. Indeed, off-manifold data points progressively disappear when being more and more selective, as illustrated in Figure 5.3c. We quantitatively confirm that our truncation method (JBT) improves the precision. On Fig. 5.3d, we observe that keeping the 70% of lowest JFN samples leads to an almost perfect precision of the support of the generated distribution. Thus, off-manifold samples are in the 30% samples with highest JFN.

5.4.3 Image datasets

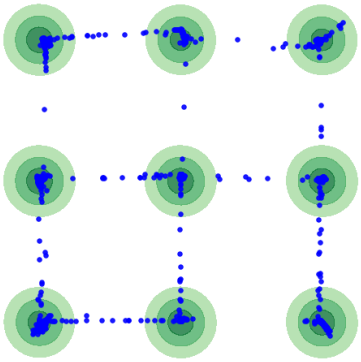
We further study JBT on three different datasets: MNIST LeCun et al. (1998), FashionMNIST Xiao et al. (2017) and CIFAR10 Krizhevsky et al. (2009). Following Khayatkhoei et al. (2018) implementation, we use a standard CNN architecture for MNIST and FashionMNIST while training a ResNet-based model for CIFAR10 (Gulrajani et al., 2017).

Figure 5.4 highlights that JBT also works on high dimensional datasets as the marginal precision plummets for high truncation ratios. Furthermore, when looking at samples ranked by increasing order of their JFN, we notice that samples with highest JFN are standing in-between manifolds. For example, those are ambiguous digits resembling both a "0" and a "6" or shoes with unrealistic shapes.

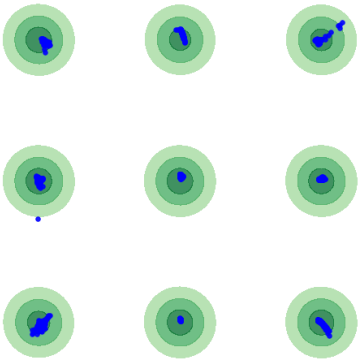
To further assess the efficiency of our truncation method, we also compare its performances with two state-of-the-art over-parameterization techniques that were designed for disconnected manifold learning. First, Gurumurthy et al. (2017) propose DeliGAN, a reparametrization trick



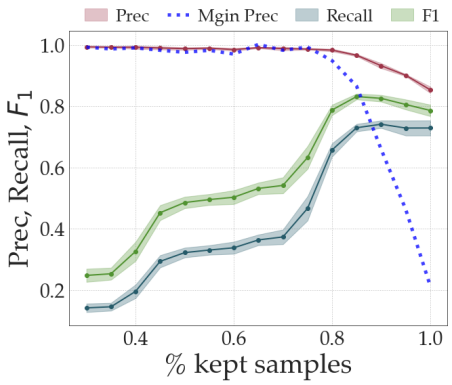
(a) WGAN - 2500 samples



(b) WGAN 90% JBT.



(c) WGAN 70% JBT.

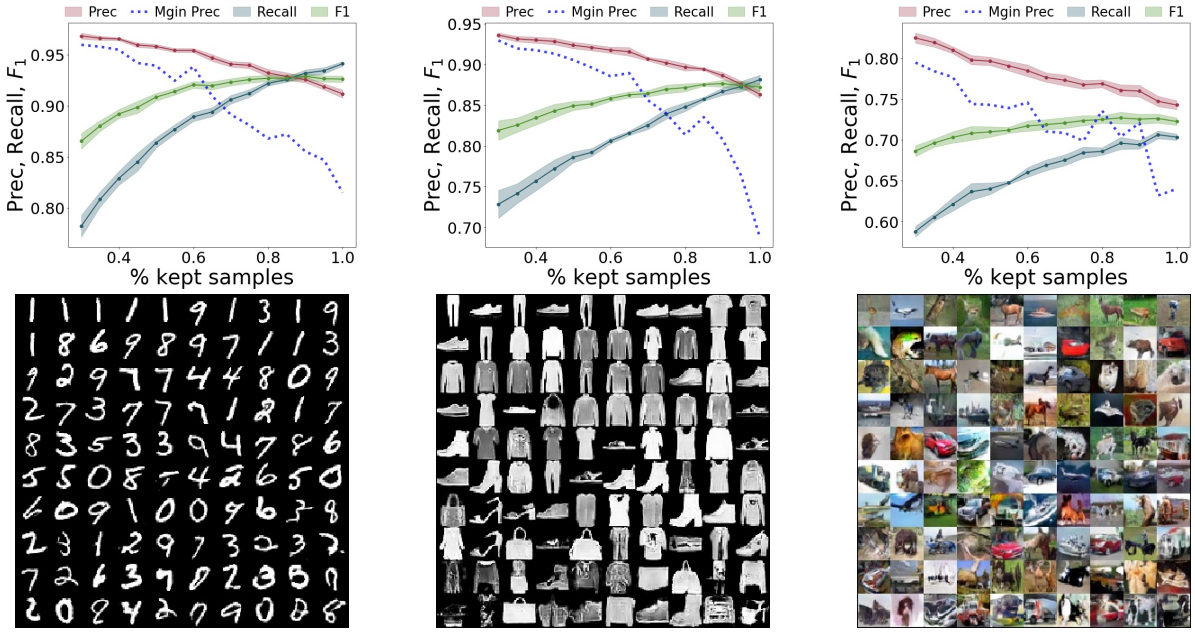


(d) 97% confidence intervals .

Fig. 5.3 Mixture of 9 Gaussians in green, generated points in blue. Our truncation method (JBT) removes least precise data points as marginal precision plummets.

to transform the unimodal Gaussian latent distribution into a mixture. The different mixture components are later learnt by gradient descent. For fairness, the re-parametrization trick is used on top of WGAN. Second, Khayatkhoei et al. (2018) define DMLGAN, a mixture of generators to better learn disconnected manifolds. In this architecture, each generator is encouraged to target a different submanifold by enforcing high mutual information between generated samples and generator's ids. Keep in mind that for DeliGAN (respectively DMLGAN), the optimal number of components (respectively generators) is not known and is a hyper-parameter of the model that has to be cross-validated.

The results of the comparison are presented in Table 5.1. In both datasets, JBT 80 % outperforms DeliGAN and DMLGAN in terms of precision while keeping a reasonnable recall. This confirms our claim that over-parameterization techniques are unnecessary. As noticed by Kynkäänniemi et al. (2019), we also observe that FID does not correlate properly with the Improved PR metric. Based on the Frechet distance, only a distance between multivariate

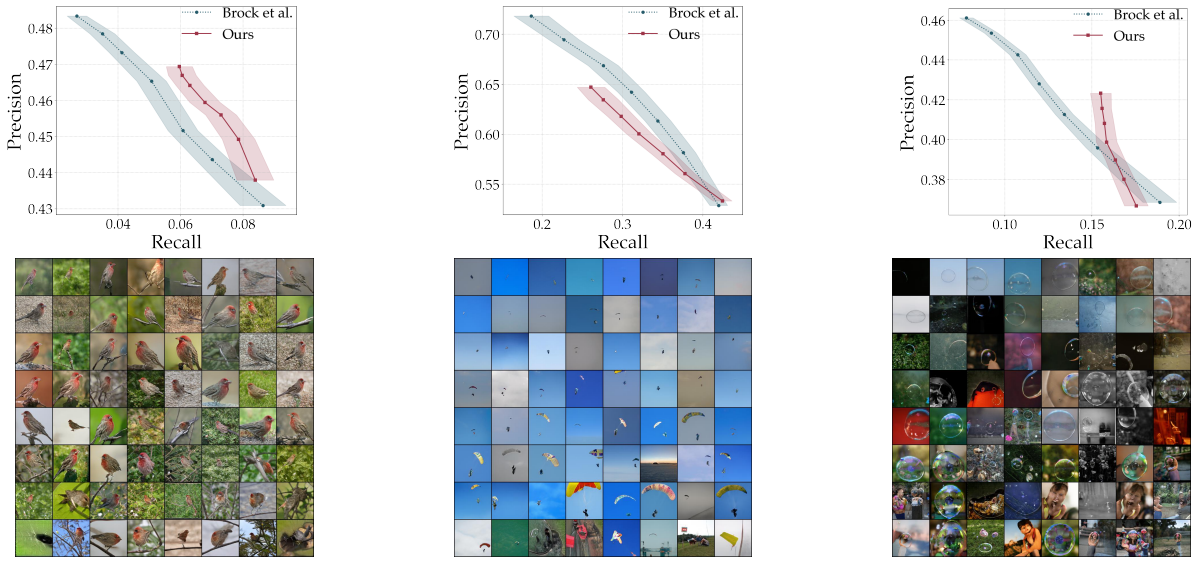


(a) MNIST dataset. (b) F-MNIST dataset. (c) CIFAR10 datatset.

Fig. 5.4 For high levels of kept samples, the marginal precision plummets of newly added samples, underlining the efficiency of our truncation method (JBT). Reported confidence intervals are 97% confidence intervals. On the second row, generated samples ordered by their JFN (left to right, top to bottom). In the last row, the data points generated are blurrier and outside the true manifold.

MNIST	Prec.	Rec.	FID
WGAN	91.2±0.3	**93.7±0.5**	24.3±0.3
WGAN JBT 90%	92.5±0.5	92.9±0.3	26.9±0.5
WGAN JBT 80%	**93.3±0.3**	91.8±0.4	33.1±0.3
W-Deligan	89.0±0.6	**93.6±0.3**	31.7±0.5
DMLGAN	**93.4±0.2**	92.3±0.2	**16.8±0.4**
F-MNIST			
WGAN	86.3±0.4	**88.2±0.2**	259.7±3.5
WGAN JBT 90%	88.6±0.6	86.6±0.5	**257.4±3.0**
WGAN JBT 80%	**89.8±0.4**	84.9±0.5	396.2±6.4
W-Deligan	88.5±0.3	85.3±0.6	310.9±3.1
DMLGAN	87.4±0.3	**88.1±0.4**	**253.0±2.8**

Table 5.1 JBT $x\%$ means we keep the $x\%$ samples with lowest Jacobian norm. Our truncation method (JBT) matches over-parameterization techniques. $\pm$ is 97% confidence interval.



(a) House finch. (b) Parachute. (c) Bubble.

Fig. 5.5 On the first row, per-class precision-recall curves comparing Brock et al. (2019)'s truncation trick and our truncation method (JBT), on three ImageNet classes generated by BigGAN. We show better results on complex and disconnected classes (*e.g.* bubble). Reported confidence intervals are 97% confidence intervals. On the second row, generated samples ordered by their JFN (left to right, top to bottom). We observe a concentration of off-manifold samples for images on the bottom row, confirming the soundness of JBT.

Gaussians, we argue that FID is not suited for disconnected manifold learning as it approximates distributions with unimodal ones and looses many information.

5.4.4 Spurious samples rejections on BigGAN

Thanks to the simplicity of JBT, we can also apply it on top of any trained generative model. In this subsection, we use JBT to improve the precision of a pre-trained BigGAN model Brock et al. (2019), which generates class-conditionned ImageNet Krizhevsky et al. (2012) samples. The class-conditioning lowers the problem of off-manifold samples, since it reduces the disconnectedness in the output distribution. However, we argue that the issue can still exist on high-dimensional natural images, in particular complex classes can still be multi-modal (*e.g.* the bubble class). The bottom row in Figure 5.5 shows a random set of 128 images for three different classes ranked by their JFN in ascending order (left to right, top to bottom). We observe a clear concentration of spurious samples on the bottom row images.

To better assess the Jacobian based truncation method, we compare it with the truncation trick from Brock et al. (2019). This truncation trick aims to reduce the variance of the latent space distribution using truncated Gaussians. While easy and effective, this truncation has

some issues: it requires to complexify the loss to enforce orthogonality in weight matrices of the network. Moreover, as explained by Brock et al. (2019) *"only 16% of models are amenable to truncation, compared to 60% when trained with Orthogonal Regularization"*. For fairness of comparison, the pre-trained network we use is optimized for their truncation method. On the opposite, JBT is simpler to apply since 100% of the tested models were amenable to the proposed truncation.

Results of this comparison are shown in the upper row of Figure 5.5. Our method can outperform their truncation trick on difficult classes with high intra-class variation, *e.g.* bubble and house finch. This confirms our claim that JBT can detect outliers within a class. However, one can note that their trick is particularly well suited for simpler unimodal classes, *e.g.* parachute and reaches high precision levels.

5.5 Conclusion and future work

In this paper, we provide insights on the learning of disconnected manifolds with GANs. Our analysis shows the existence of an off-manifold area with low precision. We empirically show on several datasets and models that we can detect these areas and remove samples located in between two modes thanks to a newly proposed truncation method.

Similarly to what has been proposed in this chapter, we want to briefly stress another possible solution that aims at improving the quality of trained generators. Note that this is also part of an ongoing work.

There is an already existing research that post-processes GANs' samples heavily relying on a variety of Monte-Carlo algorithms: Azadi et al. (2019) use the Rejection Sampling algorithm, Turner et al. (2019) the Metropolis-Hastings method, and Grover et al. (2019) the Sampling importance re-sampling method. These methods aim at sampling from a target distribution, while having only access to samples generated from a proposal distribution. This idea was successfully applied to GANs, using the previously learned generative distribution μ_θ as a proposal distribution. However, one of the main drawback is that Monte-Carlo algorithms only guarantee to sample from the target distribution under strong assumptions. First, we need access to the density ratios between the proposal and target distributions or equivalently to a perfect discriminator (Azadi et al., 2019). Second, the support of the proposal distribution must fully cover the one of the target distribution, which means no mode collapse. This is known to be very demanding in high dimension since the intersection of supports between the proposal and target distribution is likely to be negligible (Arjovsky and Bottou, 2017, Lemma 3). In this setting, an optimal discriminator would give null acceptance probabilities for almost any generated points, leading to a lower performance.

To tackle the aforementioned issue, we propose a novel method aiming at reducing the Wasserstein distance between the previously trained generative model and the target distribution. This is done via the adversarial training of a third network that learns importance weights in the latent space. The goal is to learn the redistribution of mass of the modeled distribution that best fits the target distribution. More formally, we propose to over parameterize the class of generative distributions and define a parametric class $\Omega = \{w_\varphi, \varphi \in \Phi\}$ of importance weighters. Each function w_φ learns importance weights in the latent space and is consequently defined from $\mathbb{R}^d$ to $\mathbb{R}^+$. For any given importance weighter w_θ, we impose the constraint $\mathbb{E}_{\mu_\theta}\, w_\varphi = 1$ and define μ_θ^φ the new weighted probability distribution defined as follows:

$$\text{for all } x \in \mathbb{R}^D,\ \mathrm{d}\mu_\theta^\varphi(\mathrm{x}) = \mathrm{w}_\varphi(\mathrm{x})\mathrm{d}\mu_\theta(\mathrm{x}).$$

Denoting by Lip_1 the set of 1-Lipschitz real-valued functions on $\mathbb{R}^D$, i.e.,

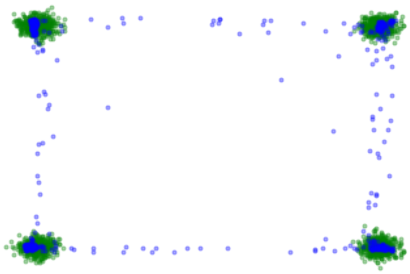
$$\mathrm{Lip}_1 = \left\{ f : \mathbb{R}^D \to \mathbb{R} : |f(x) - f(y)| \leqslant \|x - y\|,\ (x,y) \in (\mathbb{R}^D)^2 \right\},$$

the objective is to find the optimal importance weighter w_φ such that:

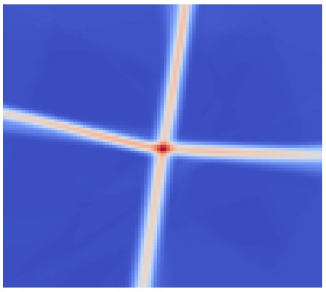
$$\begin{aligned} \arg\min_{\varphi \in \Phi} W(\mu_\star, \mu_\theta^\varphi) &= \arg\min_{w_\varphi \in \Omega} \sup_{D \in \mathrm{Lip}_1} \mathbb{E}_{\mu_\star} D - \mathbb{E}_{\mu_\theta^\varphi} D \\ &= \arg\min_{w_\varphi \in \Omega} \sup_{D \in \mathrm{Lip}_1} \mathbb{E}_{\mu_\star} D - \mathbb{E}_{\mu_\theta} w_\varphi D. \end{aligned}$$

To better understand our approach, we first consider a simple 2D motivational example where the real data lies on four disconnected manifolds. To approximate this, the generator splits the latent space into four distinct areas and maps data points located in the frontiers, areas in orange in Figure 5.6b, out of the true manifold (see Figure 5.6a). Our method consequently aims at learning latent importance weights that can identify these frontiers and simply avoid them. This is highlighted in Figure 5.6d where the importance weighter has identified these four frontiers. When sampling from the new latent distribution, we can now perfectly fit the mixture of four gaussians (see Figure 5.6c).

Consequently, future would thorougly compare the proposed method with a large set of previous approaches such as Azadi et al. (2019), Grover et al. (2019) and Tanaka (2019). These experiments should be ran on a variety of datasets and distributions.

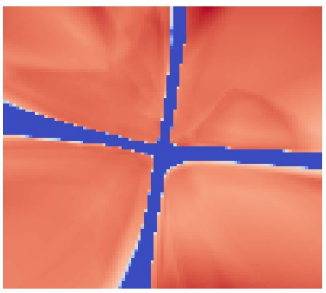


(a) WGAN: real samples in green and fake ones in blue.



(b) Latent space: heatmap of the distance between a generated sample and its nearest real sample.

(c) WGAN with latent rejection sampling: real samples in green and fake ones in blue.



(d) Latent space: heatmap of the learned importance weights. The blue frontiers have zero weights.

Fig. 5.6 Learning disconnected manifolds leads to the apparition of an area in the latent space generating points outside the target manifold. With the use of the importance weighter, one can avoid this specific area and better fit the target distribution.

Appendix 5.A Technical results

5.A.1 Highlighting drawbacks of the Precision/Recall metric

Lemma 5.A.1. *Assume that the modeled distribution μ_θ slightly collapses on a specific data point, i.e. there exists $x \in E, \mu_\theta(x) > 0$. Assume also that $\mu_\star$ is a continuous probability measure and that μ_θ has a recall $\beta = 1$. Then the precision must be such that $\alpha = 0$.*

Proof. Using Definition 5.3.1, we have that there exists μ such that

$$\mu_\star = \alpha\mu + (1-\alpha)\nu_{\mu_\star} \quad \text{and} \quad \mu_\theta = \mu.$$

Thus, $0 = \mu_\star(x) \geqslant \alpha\mu(x) = \alpha\mu_\theta(x)$. Which implies that $\alpha = 0$. □

5.A.2 Proof of Theorem 5.3.1

The proof of Theorem 5.3.1 relies on theoretical results from non-parametric estimation of the supports of probability distribution studied by Devroye and Wise (1980).

For the following proofs, we will require the following notation: let φ be a strictly monotonous function be such that $\lim_{n\to\infty} \frac{\varphi(n)}{n} = 0$ and $\lim_{n\to\infty} \frac{\varphi(n)}{\log(n)} = \infty$. We note $B(x,r) \subseteq E$, the open ball centered in x and of radius r. For a given probability distribution μ, S_μ refers to its support. We recall that for any x in a dataset D, $x_{(k)}$ denotes its k nearest neighbor in D. Finally, for a given probability distribution μ and a dataset D_μ sampled from μ^n, we note $R_{\min}$ and $R_{\max}$ the following:

$$R_{\min} = \min_{x\in E} \|x - x_{(\varphi(n))}\|, \quad R_{\max} = \max_{x\in E} \|x - x_{(\varphi(n))}\|. \tag{5.A.1}$$

In the following lemma, we show asymptotic behaviours for both $R_{\min}$ and $R_{\max}$.

Lemma 5.A.2. *Let μ be a probability distribution associated with a uniformly continuous probability density function f_μ. Assume that there exists constants $a_1 > 0, a_2 > 0$ such that for all $x \in E$, we have $a_1 < f_\mu(x) \leqslant a_2$. Then,*

$$R_{min} \underset{n\to\infty}{\longrightarrow} 0 \text{ a.s.} \quad \text{and} \quad R^d_{min} \underset{n\to\infty}{\longrightarrow} \infty \text{ a.s.}$$
$$R_{max} \underset{n\to\infty}{\longrightarrow} 0 \text{ a.s.} \quad \text{and} \quad R^d_{max} \underset{n\to\infty}{\longrightarrow} \infty \text{ a.s.}$$

Proof. We will only prove that $R_{\max} \underset{n\to\infty}{\longrightarrow} 0$ a.s. and and $R^d_{\min} \underset{n\to\infty}{\longrightarrow} \infty$ a.s. as the rest follows.

The result is based on a nearest neighbor result from Biau and Devroye (2015). Considering the $\varphi(n)$ nearest neighbor density estimate $f_n^{\varphi(n)}$ based on a finite sample dataset D_μ, Theorem

4.2 states that if f_μ is uniformly continuous then:

$$\sup_{x\in E} \|f_n^{\varphi(n)}(x) - f_\mu(x)\| \to 0.$$

where $f_n^{\varphi(n)}(x) = \frac{\varphi(n)}{nV_d\|x-x_{\varphi(n)}\|^d}$ with V_d being the volume of the unit ball in $\mathbb{R}^d$.

Let $\varepsilon > 0$ such that $\varepsilon < a_1/2$. There exists $N \in \mathbb{N}$ such that for all $n \geqslant N$, we have, almost surely, for all $x \in E$:

$$a_1 - \varepsilon \leqslant f_n^{\varphi(n)}(x) \leqslant a_2 + \varepsilon$$
$$a_1 - \varepsilon \leqslant \frac{\varphi(n)}{nV_d\|x-x_{\varphi(n)}\|^d} \leqslant a_2 + \varepsilon$$

Consequently, for all $n \geqslant N$, for all $x \in E$ almost surely:

$$\|x - x_{\varphi(n)}\| \leqslant \Big(\frac{\varphi(n)}{nV_d(a_1-\varepsilon)}\Big)^{1/d}$$
$$\text{Thus,} \quad \sup_{x\in E} \|x - x_{\varphi(n)}\| \to 0 \quad \text{a.s..}$$

Also, almost surely

$$n\|x - x_{\varphi(n)}\|^d \geqslant \frac{\varphi(n)}{V_d(a_2+\varepsilon)}$$
$$\text{Thus,} \quad \inf_{x\in E} \|x - x_{\varphi(n)}\| \to \infty \quad \text{a.s..}$$

□

Lemma 5.A.3. *Let μ, ν be two probability distributions associated with uniformly continuous probability density functions f_μ and f_ν. Assume that there exists constants $a_1 > 0, a_2 > 0$ such that for all $x \in E$, we have $a_1 < f_\mu(x) \leqslant a_2$ and $a_1 < f_\nu \leqslant a_2$. Also, let D_μ, D_ν be datasets sampled from ν^n, μ^n. If μ is an estimator for ν, then*

$$(i)\ \textit{for all } x \in D_\mu,\ \alpha^n_{\varphi(n)}(x) \underset{n\to\infty}{\to} \mathbb{1}_{\text{supp}(\nu)}(x) \quad \textit{in proba.}$$
$$(ii)\ \textit{for all } y \in D_\nu,\ \beta^n_{\varphi(n)}(y) \underset{n\to\infty}{\to} \mathbb{1}_{\text{supp}(\mu)}(x) \quad \textit{in proba.}$$

Proof. We will only show the result for (i), since a similar proof holds for (ii).

Thus, we want to show that

$$\text{for all } x \in D_\mu,\ \alpha^n_{\varphi(n)}(x) \underset{n\to\infty}{\to} \mathbb{1}_{supp(\nu)}(x) \quad \text{a. s.}$$

First, let's assume that $x \notin S_\nu$. Biau and Devroye (2015, Lemma 2.2) have shown that

$$\lim_{n\to\infty} \|x_{(\varphi(n))} - x\| = \inf\{\|x-y\| \mid y \in S_\nu\} \quad \text{a.s.}$$

As S_ν is a closed set - e.g. (Kallenberg, 2006) - we have

$$\lim_{n\to\infty} \|x - x_{(\varphi(n))}\| > 0 \quad \text{a.s.}$$

and

$$\text{for all } y \in D_\nu, \lim_{n\to\infty} \|y - y_{(\varphi(n))}\| = 0 \quad \text{a.s.}$$

Thus, $\lim_{n\to\infty} \alpha^n_{\varphi(n)}(x) = 0$ a.s..

Now, let's assume that $x \in S_\nu$. Using Definition 5.3.2, the precision of a given data point x can be rewritten as follows:

$$\alpha^n_{\varphi(n)}(x) = 1 \iff \exists y \in D_\nu, x \in B(y, \|y - y_{(\varphi(n))}\|)$$

Using notation from (5.A.1), we note

$$R_{\min} = \min_{y\in} \|y - y_{(\varphi(n))}\|, \quad R_{\max} = \max_{y\in E} \|y - y_{(\varphi(n))}\|.$$

It is clear that :

$$\bigcup_{y\in D_\nu} B(y, R_{\min}) \subseteq S^n_\nu \subseteq \bigcup_{y\in D_\nu} B(y, R_{\max}), \tag{5.A.2}$$

where $S^n_\nu = \bigcup_{y\in D_\nu} B(y, \|y - y_{(\varphi(n))}\|))$.

Besides, combining Lemma 5.A.2 with Devroye and Wise (1980, Theorem 1), we have that:

$$\nu(S_\nu \Delta \bigcup_{y\in D_\nu} B(y, R_{\min})) \underset{n\to 0}{\longrightarrow} 0 \quad \text{in proba.}$$
$$\nu(S_\nu \Delta \bigcup_{y\in D_\nu} B(y, R_{\max})) \underset{n\to 0}{\longrightarrow} 0 \quad \text{in proba.}$$

where Δ here refers to the symmetric difference.

Thus, using (5.A.2), it is now clear that, $\mu(S_\nu \Delta S^n_\nu) \to 0$ in probability. Finally, given $x \in S_\mu$, we have $\mu(x \in S^n_\nu) = \nu(\alpha^n_{\varphi(n)}(x) = 1) \to 1$ in probability. □

We can now finish the proof for Theorem 5.3.1. Recall that $\bar{\alpha} = \mu(S_\nu)$ and similarly, $\bar{\beta} = \nu(S_\mu)$.

Proof. We have that

$$|\alpha^n_{\varphi(n)} - \bar{\alpha}| = |\frac{1}{n}\sum_{x_i \in D_\mu} \alpha^n_{\varphi(n)}(x_i) - \int_E \mathbb{1}_{x \in S_\nu} \mu(\mathrm{d}x)|$$

Then,

$$\begin{aligned}
|\alpha^n_{\varphi(n)} - \bar{\alpha}| &= |\frac{1}{n}\sum_{x_i \in D_\mu} (\alpha^n_{\varphi(n)}(x_i) - \mathbb{1}_{x_i \in S_\nu}) \\
&\quad + (\frac{1}{n}\sum_{x_i \in D_\mu} \mathbb{1}_{x_i \in S_\nu} - \int_E \mathbb{1}_{x \in S_\nu} \mu(\mathrm{d}x))| \\
&= |\mathbb{E}_{x_i \sim \mu_n}(\alpha^n_{\varphi(n)}(x_i) - \mathbb{1}_{x_i \in S_\nu}) && \text{(5.A.3)} \\
&\quad + (\mathbb{E}_{\mu_n} \mathbb{1}_{S_\nu} - \mathbb{E}_\mu \mathbb{1}_{S_\nu})| && \text{(5.A.4)}
\end{aligned}$$

where μ_n is the empirical distribution of μ. As μ_n converges weakly to μ almost surely (e.g. Dudley (2004, Theorem 11.4.1)) and since $\mathbb{1}_{x \in S_\nu}$ is bounded, we can bound (5.A.4) as follows:

$$\lim_{n \to \infty} \quad \mathbb{E}_{x \sim \mu_n} \mathbb{1}_{x \in \mathrm{supp}(\mu)} - \mathbb{E}_{x \sim \mu} \mathbb{1}_{x \in \mathrm{supp}(\mu)} = 0 \quad \text{a. s.}$$

Now, to bound (5.A.3), we use the fact that for any $x \in D_\mu$, the random variable $\alpha^n_{\varphi(n)}(x)$ converges to $\mathbb{1}_{x \in S_\nu}$ in probability (Lemma 5.A.3) and that for all $x \in D_\mu$, both $\alpha^n_{\varphi(n)}(x) \leqslant 1$ and $\mathbb{1}_{x \in S_\nu} \leqslant 1$. Consequently, using results from the weak law for triangular arrays, we have that

$$\lim_{n \to \infty} \quad \frac{1}{n}\sum_{x_i \in D_\mu} (\alpha^n_{\varphi(n)}(x_i) - \mathbb{1}_{x_i \in S_\nu}) = 0 \quad \text{in proba.}$$

Finally,

$$|\alpha^n_{\varphi(n)} - \bar{\alpha}| \underset{n \to \infty}{\to} 0 \quad \text{in proba.},$$

which proves the result. The same proof works for $\lim_{k \to \infty} \beta^n_k = \bar{\beta}$. □

5.A.3 Proof of Theorem 5.3.2

This proof is based on the Gaussian isoperimetric inequality historically shown by Borell (1975); Sudakov and Tsirelson (1978).

Proof. Let $\mu_\star$ be a distribution defined on E laying on two disconnected manifolds M_1 and M_2 such that $\mu_\star(M_1) = \mu_\star(M_2) = \frac{1}{2}$ and $d(M_1, M_2) = D$. Note that for any subsets $A \subseteq E$ and $B \subseteq E$, $d(A,B) := \inf_{(x,y)\in A\times B} \|x-y\|$.

Let $G_\theta^{-1}(M_1)$ (respectively $G_\theta^{-1}(M_2)$ be the subset in $\mathbb{R}^d$ be the pre-images of M_1 (respectively M_2).

Consequently, we have for all $k \in [1,n]$

$$\gamma(G_\theta^{-1}(M_1)) = \mu_\theta(M_1) = \gamma(G_\theta^{-1}(M_2)) \geqslant \frac{\bar{\alpha}}{2}$$

We consider $(G_\theta^{-1}(M_1))^\varepsilon$ (respectively $(G_\theta^{-1}(M_2))^\varepsilon$) the ε enlargement of $G_\theta^{-1}(M_1)$ (respectively $G_\theta^{-1}(M_2)$ where $\varepsilon = \frac{D}{2L}$. We know that $(G_\theta^{-1}(M_1))^\varepsilon \bigcap (G_\theta^{-1}(M_2))^\varepsilon = \emptyset$.

Thus, we have that:

$$\gamma\big((G_\theta^{-1}(M_1))^\varepsilon\big) + \gamma\big((G_\theta^{-1}(M_2))^\varepsilon\big) \leqslant 1$$

Besides, by denoting Φ the function defined for any $t \in \mathbb{R}$ by $\Phi(t) = \int_{-\infty}^{t} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} ds$, we have

$$\begin{aligned}
&\gamma\big((G_\theta^{-1}(M_1))^\varepsilon\big) + \gamma\big((G_\theta^{-1}(M_2))^\varepsilon\big) \geqslant 2\Phi\big(\Phi^{-1}(\frac{\alpha}{2}) + \varepsilon\big) \\
&\quad \text{(using Theorem 1.3 from Ledoux (1996))} \\
&\quad \geqslant \alpha + \frac{2\varepsilon}{\sqrt{2\pi}} e^{-\Phi^{-1}(\frac{\alpha}{2})^2/2} \\
&\text{(since } \Phi^{-1}(\frac{\alpha}{2}) + \varepsilon < 0 \text{ and } \Phi \text{ convex on }]-\infty, 0]\text{)}
\end{aligned}$$

Thus, we have that

$$\alpha + \frac{2\varepsilon}{\sqrt{2\pi}} e^{-\Phi^{-1}(\frac{\alpha}{2})^2/2} \leqslant 1$$

Thus, by noting

$$\alpha^\star = \sup\{\alpha \in [0,1] \mid \alpha + \frac{2\varepsilon}{\sqrt{2\pi}} e^{\frac{-\Phi^{-1}(\frac{\alpha}{2})^2}{2}} \leqslant 1\},$$

we have our result.

For $\alpha \geqslant 3/4$. By noting $\alpha = 1 - x$, we have

$$\begin{aligned}
\Phi^{-1}(\frac{\alpha}{2}) &= \frac{\sqrt{2\pi}x}{2} + O(x^3) \\
\text{And, } e^{\frac{-\Phi^{-1}(\frac{\alpha}{2})^2}{2}} &= e^{\frac{-\pi x^2}{4}} + O(e^{-x^4}) \\
\text{Thus, } 1 - x + \frac{2\varepsilon}{\sqrt{2\pi}} e^{\frac{-\pi x^2}{4}} + O(e^{-x^4}) &\leqslant 1 \\
\iff x &\geqslant \frac{2\varepsilon}{\sqrt{2\pi}} e^{\frac{-\pi x^2}{4}} + O(e^{-x^4}) \\
\implies x &\geqslant \sqrt{\frac{2}{\pi}} W(\varepsilon^2)
\end{aligned}$$

where W is the product log function. Thus, $\alpha \leqslant 1 - \sqrt{\frac{2}{\pi}} W(\varepsilon^2)$. □

As an example, in the case where $\varepsilon = 1$, we have that $W(1) \approx 0.5671$, $x > 0.4525$ and $\alpha < 0.5475$.

5.A.4 Proof of Theorem 5.3.3

5.A.4.1 Equitable setting

This result is a consequence of Theorem 5.A.1 that we will assume true in this section.

We consider that the unknown true distribution $\mu_\star$ lays on M disjoint manifolds of equal measure. As specified in Section 5.3, the latent distribution γ is a multivariate Gaussian defined on $\mathbb{R}^d$. For each $k \in [1, M]$, we consider in the latent space, the pre-images A_k.

It is clear that $A_1, \ldots, A_M$ are pairwise disjoint Borel subsets of $\mathbb{R}^d$. We denote $\bar{M}$, the number of classes covered by the estimator μ_θ, such that for all $i \in [1, \bar{M}]$, we have $\gamma(A_i) > 0$. We know that $\bar{M} \geqslant M\bar{\beta} > 1$.

For each $i \in [1, \bar{M}]$, we denote A_i^ε, the ε-enlargement of A_i. For any pair (i, j) it is clear that $A_i^\varepsilon \bigcap A_j^\varepsilon = 0$ where $\varepsilon = \frac{D}{2L}$ (D being the minimum distance between two sub-manifolds and L being the Lipschitz constant of the generator).

As assumed, we know that $A_i^\varepsilon, i \in [1, \bar{M}]$ partition the latent space in equal measure, consequently, we assume that

$$\sum_{i=1}^{n} \gamma(A_i^\varepsilon) = 1 \quad \text{and} \quad \gamma(A_1) = \ldots = \gamma(A_{\bar{M}}) = 1/\bar{M} \tag{5.A.5}$$

Thus, we have that

$$\bar{\alpha} = \sum_{i=1}^{\bar{M}} \gamma(A_i^\varepsilon) = 1 - \gamma(\Delta^{-\varepsilon}(A_1^\varepsilon, \dots, A_{\bar{M}}^\varepsilon))$$

Using Theorem 5.A.1, we have

$$\gamma(\Delta^{-\varepsilon}(A_1^\varepsilon, \dots, A_n^\varepsilon)) \geqslant 1 - \frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x}$$

$$\text{Thus, } \bar{\alpha} \leqslant \frac{1+y^2}{y^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon y}$$

where $y = \Phi^{-1}\left(1 - \max_{k \in [\bar{M}]} \gamma(A_k^\varepsilon)\right) = \Phi^{-1}(\frac{\bar{M}-1}{\bar{M}})$ and $\Phi(t) = \int_{-\infty}^{t} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} ds$.

Knowing that $\bar{M} \geqslant \bar{\beta} M$ we have that

$$\Phi^{-1}(1 - \frac{1}{\bar{M}}) \geqslant \Phi^{-1}(1 - \frac{1}{\bar{\beta} M})$$

We conclude by saying that the function $x \mapsto \frac{1+x^2}{x^2} e^{-\varepsilon x}$ is decreasing for $x > 0$. Thus,

$$\bar{\alpha} \leqslant \frac{1+y^2}{y^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon y} \tag{5.A.6}$$

where $y = \Phi^{-1}(1 - \frac{1}{\bar{\beta} M})$ and $\Phi(t) = \int_{-\infty}^{t} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} ds$.

For further analysis, when $\bar{M} \to \infty$, refer to subsection 5.A.5 and note using the result in (5.A.14) that one obtains the desired upper-bound on $\bar{\alpha}$

$$\bar{\alpha} \overset{\bar{M} \to \infty}{\leqslant} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon\sqrt{2\log(\bar{M})}}$$

5.A.4.2 More general setting

As done previously, we denote $\bar{M}$, the number of classes covered by the estimator μ_θ, such that for all $i \in [1, \bar{M}]$, we have $\gamma(A_i) > 0$. We still assume that $\bar{M} > 1$. However, we now relax the previous assumption made in (5.A.5) and assume the milder assumption that there exists $w_1, \dots, w_M \in [0,1]^M$ such that for all $m \in [1, M], \gamma(A_m^\varepsilon) = w_m$, $\sum_m w_m \leqslant 1$ and $\max_{i \in [1,M]} w_m = w^{\max} < 1$.

Consider, $A^{\complement} = \left(\bigcup_{i=1}^{\bar{M}} A_i^{\varepsilon}\right)^{\complement}$ and denote $w^c = \gamma(A^{\complement}) \leqslant 1 - \bar{\alpha}$. Consequently, we have

$$\begin{aligned}
&\sum_{i=1}^{n} \gamma(A_i^{\varepsilon}) + \gamma(A^{\complement}) = 1 \\
&\gamma(\Delta^{-\varepsilon}(A_1^{\varepsilon}, \ldots, A_M^{\varepsilon}, A^{\complement})) + \sum_{i=1}^{M} \gamma(A_i^{\varepsilon}) = 1 - \gamma(A^{\complement}) \\
&\bar{\alpha} = 1 - w^{\complement} - \gamma(\Delta^{-\varepsilon}(A_1^{\varepsilon}, \ldots, A_M^{\varepsilon}, A^{\complement}))
\end{aligned}$$

In this setting, it is clear that $A_1, \ldots, A_{\bar{M}}, A^{\complement}$ is a a partition of $\mathbb{R}^d$ under the measure γ. Using, result from Theorem 5.A.1, we have

$$\gamma(\Delta^{-\varepsilon}(A_1^{\varepsilon}, \ldots, A_M^{\varepsilon}, A^{\complement})) \geqslant 1 - \frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x}$$

where $x = \Phi^{-1}\left(1 - \max(w^{\complement}, w^{\max})\right)$ and $\Phi(t) = \int_{-\infty}^{t} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} ds$.

Finally, we have that

$$\bar{\alpha} \leqslant \frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x} - w^{\complement} \tag{5.A.7}$$

In the case where $\gamma(A^{\complement}) = 0$, we find a result similar to (5.A.6).

5.A.5 Lower-bounding boundaries of partitions in a Gaussian space

Notations and preliminaries Given $\varepsilon \geq 0$ and a subset A of euclidean space $\mathbb{R}^d = (\mathbb{R}^d, \|\cdot - \cdot\|)$, let $A^{\varepsilon} := \{z \in \mathbb{R}^d \mid \mathsf{dist}(\mathsf{z}, \mathsf{A}) \leq \varepsilon\}$ be its ε-enlargement, where $\mathsf{dist}(\mathsf{z}, \mathsf{A}) := \inf_{\mathsf{z}' \in \mathsf{A}} \|\mathsf{z}' - \mathsf{z}\|_2$ is the distance of the point $z \in \mathbb{R}^d$ from A. Let γ be the standard Gaussian distribution in $\mathbb{R}^d$ and let $A_1, \ldots, A_K$ be $K \geq 2$ pairwise disjoint Borel subsets of $\mathbb{R}^d$ whose union has unit (i.e full) Gaussian measure $\sum_{k=1}^{K} w_k = 1$, where $w_k := \gamma(A_k)$. Such a collection $\{A_1, \ldots, A_K\}$ will be called an $(w_1, \ldots, w_K)$-partition of standard d-dimensional Gaussian space $(\mathbb{R}^d, \gamma)$.

For each $k \in [\![K]\!]$, define the compliment $A_{-k} := \cup_{k' \neq k} A_{k'}$, and let $\partial^{-\varepsilon} A_k := \{z \in A_k \mid \mathsf{dist}(\mathsf{z}, \mathsf{A}_{-\mathsf{k}}) \leq \varepsilon\}$ be the *inner ε-boundary* of A_k, i.e the points of A_k which are within distance ε of some other $A_{k'}$. For every $(k, k') \in [\![K]\!]^2$ with $k' \neq k$, it is an easy exercise to show that

$$\begin{aligned}
\partial^{-\varepsilon} A_k \cap \partial^{-\varepsilon} A_{k'} &= \emptyset \\
\partial^{-\varepsilon} A_k \cap A_{-k} &= \emptyset \\
A_{-k}^{\varepsilon} &= \partial^{-\varepsilon} A_k \cup A_{-k}
\end{aligned} \tag{5.A.8}$$

Now, let $\Delta^{-\varepsilon}(A_1,\dots,A_K) := \cup_{k=1}^K \partial^{-\varepsilon} A_k$ be the union of all the inner ε-boundaries. This is $\Delta^{-\varepsilon}(A_1,\dots,A_K)$ the set of points of $\cup_{k=1}^K A_k$ which are on the boundary between some two distinct A_k and $A_{k'}$. We want to find a lower bound in the measure $\gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K))$.

Theorem 5.A.1. *Given $K \geq 4$ and $w_1,\dots,w_K \in (0,1/4]$ such that $\sum_{k=1}^K w_k = 1$, we have the bound:*

$$\inf_{A_1,\dots,A_K} \gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) \geq 1 - \frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x}$$

where the infinimum is taken over all $(w_1,\dots,w_k)$-partitions of standard Gaussian space $(\mathbb{R}^d,\gamma)$, and $x := \Phi^{-1}\left(1 - \max_{k\in[\![M]\!]} w_k\right)$.

Proof. By (5.A.8), we have the formula

$$\gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) = \sum_{k=1}^K \gamma(\partial^{-\varepsilon} A_k) \tag{5.A.9}$$

$$= \sum_{k=1}^K \gamma(A_{-k}^{\varepsilon}) - \gamma(A_{-k}). \tag{5.A.10}$$

Let $w_{-k} := \gamma(A_{-k}) = 1 - w_k$, and assume $w_{-k} \geq 3/4$, i.e $w_k \leq 1/4$, for all $k \in [\![K]\!]$.

For example, this condition holds in the equitable scenario where $w_k = 1/K$ for all k.

Now, by standard *Gaussian Isoperimetric Inequality* (see Boucheron et al. (2013) for example), one has

$$\begin{aligned}\gamma(A_{-k}^{\varepsilon}) &\geq \Phi(\Phi^{-1}(\gamma(A_{-k}) + \varepsilon) \\ &= \Phi(\Phi^{-1}(1-w_k) + \varepsilon).\end{aligned} \tag{5.A.11}$$

Using the bound $\frac{x}{1+x^2}\varphi(x) < 1 - \Phi(x) < \frac{1}{x}\varphi(x)\ \forall x > 0$ where φ is the density of the standard Gaussian law. We can further find that

$$\begin{aligned}\Phi(\Phi^{-1}(1-w_k)+\varepsilon) &\geq 1 - w_k \frac{1+\Phi^{-1}(1-w_k)^2}{\Phi^{-1}(1-w_k)^2} \times \\ & e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon\Phi^{-1}(1-w_k)} \\ &\geq 1 - w_k \frac{1+x^2}{x^2} e^{-\frac{1}{2}\varepsilon^2} e^{-\varepsilon x} > 0\end{aligned} \tag{5.A.12}$$

(since the function $x \mapsto \frac{1+x^2}{x^2} e^{-\varepsilon x}$ is decreasing for $x > 0$)

where $x := \min_{k\in[\![K]\!]} \Phi^{-1}(1-w_k) = \Phi^{-1}\left(1-\max_{k\in[\![K]\!]} w_k\right) \geq \Phi^{-1}(3/4) > 0.67$. Combining (5.A.9), (5.A.11), and (5.A.12) yields the following

$$\begin{aligned}\gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) &\geq \sum_{k=1}^{K}\Big(1-w_k\frac{1+x^2}{x^2}e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon x}\\ &\quad -(1-w_k)\Big)\\ &= \sum_{k=1}^{K}\left(1-\frac{1+x^2}{x^2}e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon x}\right)w_k\\ &= 1-\frac{1+x^2}{x^2}e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon x},\end{aligned}$$

Asymptotic analysis In the limit, it is easy to check that in the case where $\max_{k\in[\![K]\!]} w_k \longrightarrow 0$, we have that $x \longrightarrow \infty$. In this setting, we thus have $\frac{1+x^2}{x^2} \longrightarrow 1$ and can now derive the following bound:

$$\inf_{A_1,\dots,A_K} \gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) \overset{\max_{k\in[\![K]\!]} w_k\to 0}{\longrightarrow} 1-e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon x}.$$

Equitable scenario In the equitable scenario where $w_k = 1/K$ for all k, we have

$$\inf_{A_1,\dots,A_K} \gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) \geqslant 1-\frac{1+x^2}{x^2}e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon x}$$

where $x = \Phi^{-1}(1-1/K)$. When $K \geq 8$ we have:

$$\Phi^{-1}(1-1/K) \geqslant \sqrt{2\log\left(\frac{K\,(q(K)^2-1)}{\sqrt{2\pi}q(K)^3}\right)} \tag{5.A.13}$$

where $q(K) = \sqrt{2\log(\sqrt{2\pi}K)}$.

Consequently, we have when $K \to \infty$, the following behavior:

$$\gamma(\Delta^{-\varepsilon}(A_1,\dots,A_K)) \overset{K\to\infty}{\leqslant} 1-e^{-\frac{1}{2}\varepsilon^2}e^{-\varepsilon\sqrt{2\log(K)}} \tag{5.A.14}$$

□

Proof of the inequality (5.A.13). Set $p := 1/K$. First, for any $x > 0$, we have the following upper:

$$\int_x^\infty e^{-y^2/2}dy = \int_x^\infty \frac{y}{y}e^{-y^2/2}dy \leq \frac{1}{x}\int_x^\infty ye^{-y^2/2}dy = \frac{e^{-x^2/2}}{x}.$$

For a lower bound:

$$\int_x^\infty e^{-y^2/2}dy = \int_x^\infty \frac{y}{y}e^{-y^2/2}dy = \frac{e^{-x^2/2}}{x} - \int_x^\infty \frac{1}{y^2}e^{-y^2/2}dy$$

and

$$\int_x^\infty \frac{1}{y^2}e^{-y^2/2}dy = \int_x^\infty \frac{y}{y^3}e^{-y^2/2}dy \leq \frac{e^{-x^2/2}}{x^3}$$

and combining these gives

$$\int_x^\infty e^{-y^2/2}dy \geq \left(\frac{1}{x} - \frac{1}{x^3}\right)e^{-x^2/2}.$$

Thus

$$\frac{1}{\sqrt{2\pi}}\left(\frac{1}{x} - \frac{1}{x^3}\right)e^{-x^2/2} \leq 1 - \Phi(x) \leq \frac{1}{\sqrt{2\pi}}\frac{1}{x}e^{-x^2/2},$$

from where

$$\frac{1}{\sqrt{2\pi}}\left(\frac{1}{\Phi^{-1}(1-p)} - \frac{1}{\Phi^{-1}(1-p)^3}\right)e^{-\Phi^{-1}(1-p)^2/2} \tag{5.A.15}$$

$$\leq p \leq \frac{1}{\sqrt{2\pi}}\frac{1}{\Phi^{-1}(1-p)}e^{-\Phi^{-1}(1-p)^2/2} \tag{5.A.16}$$

Using (5.A.16), when $\Phi^{-1}(1-p) \geq 1$ (that is $p \leqslant 0.15$ or equivalently $K \geq 8$), we have the following upper bound $\Phi^{-1}(1-p) \leqslant q(p)$ where $q(p) := \sqrt{2\log(\sqrt{2\pi}/p)}$. Then, injecting $q(p)$ in (5.A.15):

$$\frac{1}{\sqrt{2\pi}}\left(\frac{1}{q(p)} - \frac{1}{q(p)^3}\right)e^{-\Phi^{-1}(1-p)^2/2} \leq p.$$

Now when $q(p) \geq 1$ you have:

$$e^{-\Phi^{-1}(1-p)^2/2} \leq \frac{\sqrt{2\pi}pq(p)^3}{q(p)^2 - 1}$$

and

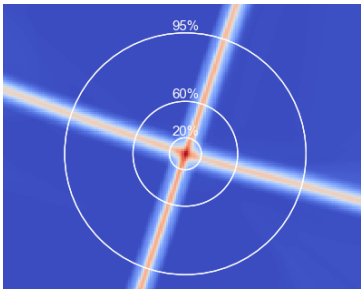
$$\Phi^{-1}(1-p) \geq \sqrt{2\log\left(\frac{q(p)^2-1}{\sqrt{2\pi}pq(p)^3}\right)}.$$

There is one additional requirement on p which is simply that the argument of the log should be ≥ 1 i.e. $q(p)^2 - 1 \geq \sqrt{2\pi}pq(p)^3$, which is true as soon as $K \geq 8$. □

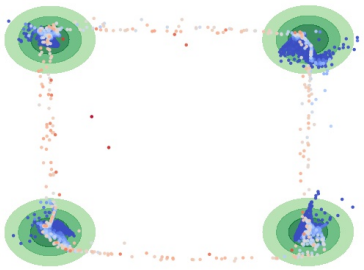
Appendix 5.B Complementary experiments

5.B.1 Visualization of Theorem 5.3.3

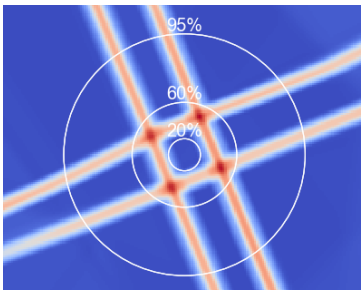
To further understand and illustrate Theorem 5.3.3, we propose in Figure 5.7, an interesting visualization where we plot the manifold learned by a WGANs architecture and its corresponding latent space configuration. As expected, we observe that when the number of distinct modes increase, the number of data generated out of the manifolds increase too.



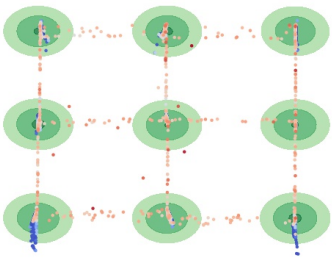
(a) WGAN 4 classes: visualisation of $\|J_G(z)\|_F$.



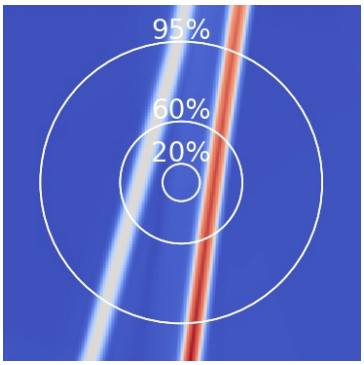
(b) Green blobs: true densities. Dots: generated points.



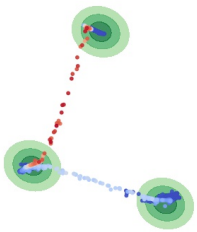
(c) WGAN 9 classes: visualisation of $\|J_G(z)\|_F$.



(d) Green blobs: true densities. Dots: generated points.



(e) WGAN 3 classes: visualisation of $\|J_G(z)\|_F$.



(f) Green blobs: true densities. Dots: generated points.

Fig. 5.7 Learning disconnected manifolds: visualization of the gradient of the generator (JFN) in the latent space and densities in the output space.

5.B.2 Definition of the different metrics used

In the sequel, we present the different metrics used in Section 5.4 of the paper to assess performances of GANs. We have:

- Improved Precision/Recall (PR) metric Kynkäänniemi et al. (2019): it has been presented in Definition 5.3.2. Intuitively, Based on a k-NN estimation of the manifold of real (resp. generated) data, it assesses whether generated (resp. real) points belong in the real (resp. generated) data manifold or not. The proportion of generated (resp. real) points that are in the real (resp. generated) data manifold is the precision (resp. recall).

- the Hausdorff distance: it is defined by

$$\text{Haus}(A,B) = \max\left\{\max_{a\in A}\min_{b\in B}\|a-b\|, \max_{b\in B}\min_{a\in A}\|a-b\|\right\}$$

  Such a distance is useful to evaluate the closeness of two different supports from a metric space, but is sensitive to outliers because of the max operation. It has been recently used for theoretical purposes by Pandeva and Schubert (2019).

- the Frechet Inception distance: first proposed by Dowson and Landau (1982), the Frechet distance was applied in the setting of GANs by Heusel et al. (2017). This distance between mutlivariate Gaussians compares statistic of generated samples to real samples as follows

$$\text{FID} = \|\nu_\star - \nu_\theta\|^2 + Tr\left(\Sigma_\star + \Sigma_\theta + 2(\Sigma_\star\Sigma_\theta)^{\frac{1}{2}}\right)$$

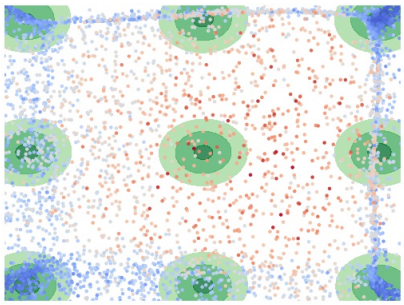
  where $X_\star = \mathcal{N}(\nu_\star, \Sigma_\star)$ and $X_\theta = \mathcal{N}(\nu_\theta, \Sigma_\theta)$ are the activations of a pre-softmax layer. However, when dealing with disconnected manifolds, we argue that this distance is not well suited as it approximates the distributions with unimodal one, thus loosing many information.

The choice of such metrics is motivated by the fact that metrics measuring the performances of GANs should not rely on relative densities but should rather be point sets based metrics.

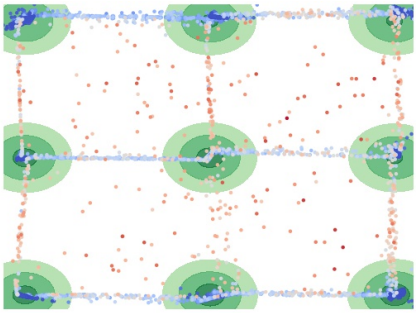
5.B.3 Saturation of a MLP neural network

In Section 5.4.2, we claim that the generator reduces the sampling of off-manifold data points up to a saturation point. Figure 5.8 below provides a visualization of this phenomenon. In this synthetic case, we learn a 9-component mixture of Gaussians using simple GANs architecture (both the generator and the discriminator are MLP with two hidden layers). The minimal

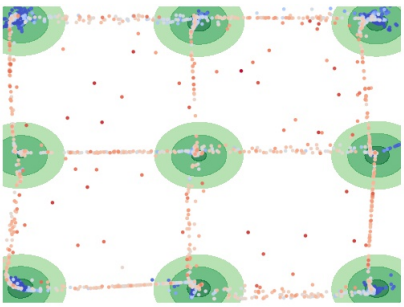
distance between two modes is set to 9. We clearly see in Figure 5.8d that the precision saturates around 80%.



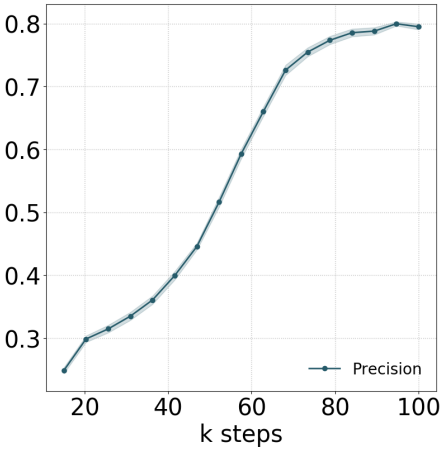
(a) Data points sampled after 5,000 steps of training.



(b) Data points sampled after 50,000 steps of training.



(c) Data points sampled after 100,000 steps of training.



(d) Evolution of the precision $\bar{\alpha}$ during training.

Fig. 5.8 Learning 9 disconnected manifolds with a standard GANs architecture.

5.B.4 More results and visualizations on MNIST/F-MNIST/CIFAR10

Additionally to those in Section 5.4.3, we provide in Figure 5.9 and Table 5.2 supplementary results for MNIST, F-MNIST and CIFAR-10 datasets.

(a) MNIST: examples of data points selected by our JBT with a truncation ratio of 90% (we thus removed the 10% highest gradients).

(b) MNIST: examples of data points removed by our JBT with a truncation ratio of 90% (these are the 10% highest gradients data points).

(c) F-MNIST: examples of data points selected by our JBT with a truncation ratio of 90% (we thus removed the 10% highest gradients)..

(d) F-MNIST: examples of data points removed by our JBT with a truncation ratio of 90% (these are the 10% highest gradients data points).

(e) CIFAR-10: examples of data points selected by our JBT with a truncation ratio of 90% (we thus removed the 10% highest gradients).

(f) MNIST: examples of data points removed by our JBT with a truncation ratio of 90% (these are the 10% highest gradients data points).

Fig. 5.9 Visualization of our truncation method (JBT) on three real-world datasets: MNIST, F-MNIST and CIFAR-10.

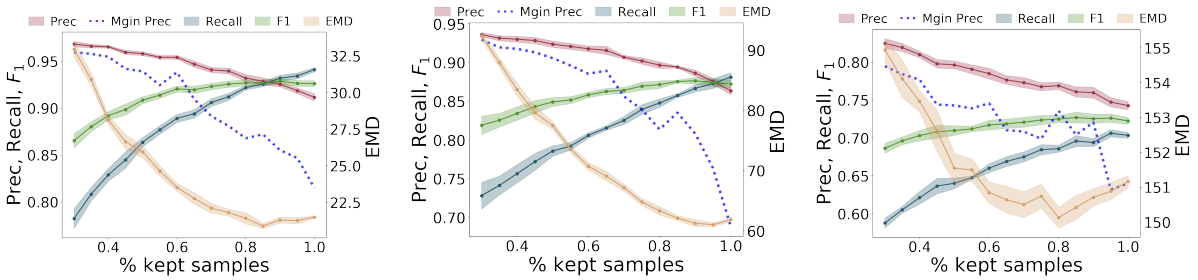


Fig. 5.10 For high levels of kept samples, the marginal precision plummets of newly added samples, underlining the efficiency of our truncation method (JBT). Reported confidence intervals are 97% confidence intervals. On the second row, generated samples ordered by their JFN (left to right, top to bottom). In the last row, the data points generated are blurrier and outside the true manifold.

MNIST	Prec.	Rec.	F1	Haus.	FID	EMD
WGAN	91.2 ± 0.3	$\mathbf{93.7 \pm 0.5}$	$\mathbf{92.4 \pm 0.4}$	49.7 ± 0.2	24.3 ± 0.3	21.5 ± 0.1
WGAN 90% JFN	92.5 ± 0.5	92.9 ± 0.3	$\mathbf{92.7 \pm 0.4}$	$\mathbf{48.1 \pm 0.2}$	26.9 ± 0.5	21.3 ± 0.2
WGAN 80% JFN	$\mathbf{93.3 \pm 0.3}$	91.8 ± 0.4	$\mathbf{92.6 \pm 0.4}$	50.6 ± 0.4	33.1 ± 0.3	21.4 ± 0.4
W-Deligan	89.0 ± 0.6	$\mathbf{93.6 \pm 0.3}$	91.2 ± 0.5	50.7 ± 0.3	31.7 ± 0.5	22.4 ± 0.1
DMLGAN	$\mathbf{93.4 \pm 0.2}$	92.3 ± 0.2	$\mathbf{92.8 \pm 0.2}$	$\mathbf{48.2 \pm 0.3}$	$\mathbf{16.8 \pm 0.4}$	$\mathbf{20.7 \pm 0.1}$
Fashion-MNIST						
WGAN	86.3 ± 0.4	$\mathbf{88.2 \pm 0.2}$	$\mathbf{87.2 \pm 0.3}$	140.6 ± 0.7	259.7 ± 3.5	61.9 ± 0.3
WGAN 90% JFN	88.6 ± 0.6	86.6 ± 0.5	$\mathbf{87.6 \pm 0.5}$	$\mathbf{138.7 \pm 0.9}$	$\mathbf{257.4 \pm 3.0}$	$\mathbf{61.3 \pm 0.6}$
WGAN 80% JFN	$\mathbf{89.8 \pm 0.4}$	84.9 ± 0.5	$\mathbf{87.3 \pm 0.4}$	146.3 ± 1.1	396.2 ± 6.4	63.3 ± 0.7
W-Deligan	88.5 ± 0.3	85.3 ± 0.6	86.9 ± 0.4	141.7 ± 1.1	310.9 ± 3.1	$\mathbf{60.9 \pm 0.4}$
DMLGAN	87.4 ± 0.3	$\mathbf{88.1 \pm 0.4}$	$\mathbf{87.7 \pm 0.4}$	141.9 ± 1.2	$\mathbf{253.0 \pm 2.8}$	$\mathbf{60.9 \pm 0.4}$
CIFAR10						
WGAN	74.3 ± 0.5	$\mathbf{70.3 \pm 0.4}$	$\mathbf{72.3 \pm 0.5}$	334.7 ± 3.5	$\mathbf{634.8 \pm 4.6}$	151.2 ± 0.2
WGAN 90% JFN	$\mathbf{76.0 \pm 0.7}$	69.4 ± 0.5	$\mathbf{72.5 \pm 0.6}$	$\mathbf{318.1 \pm 3.7}$	$\mathbf{631.3 \pm 4.5}$	150.7 ± 0.2
WGAN 80% JFN	$\mathbf{76.9 \pm 0.5}$	68.6 ± 0.5	$\mathbf{72.5 \pm 0.5}$	$\mathbf{323.5 \pm 4.0}$	725.0 ± 3.5	$\mathbf{150.1 \pm 0.3}$
W-Deligan	71.5 ± 0.7	$\mathbf{69.8 \pm 0.7}$	70.6 ± 0.7	328.7 ± 2.1	727.8 ± 3.9	154.0 ± 0.3
DMLGAN	74.1 ± 0.5	65.7 ± 0.6	69.7 ± 0.6	328.6 ± 2.7	967.2 ± 4.1	152.0 ± 0.4

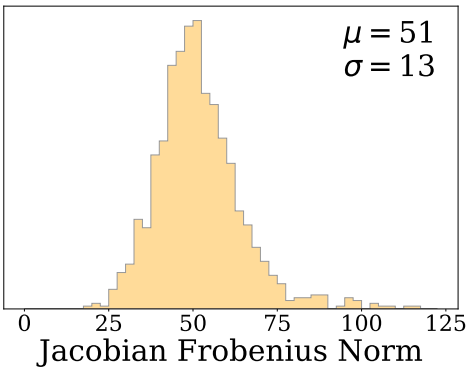
Table 5.2 Scores on MNIST and Fashion-MNIST. JFN stands for Jacobian Frobenius norm. $\pm$ is 97% confidence interval.

5.B.5 More results on BigGAN and ImageNet

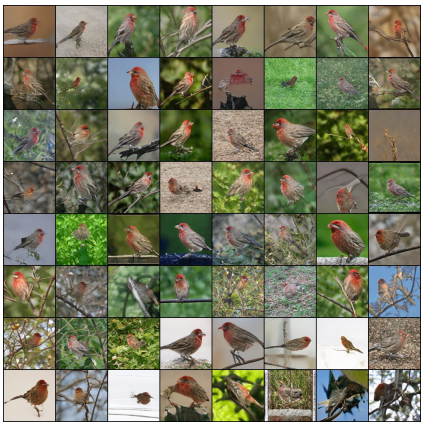
In Figure 5.11, we show images from the Bubble class of ImageNet. It supports our claim of manifold disconectedness, even within a class, and outlines the importance of studying the learning of disconnected manifolds in generative models. Then, in Figure 5.12 and Figure 5.13, we give more exemples from BigGAN 128x128 class-conditionned generator. We plot in the same format than in 5.4.4. Specifically, for different classes, we plot 128 images ranked by JFN. Here again, we see a concentration of off-manifold samples on the last row, proving the efficiency of our method. Example of classes responding particularly well to our ranking are House Finch c, Monnarch Butterfly e or Wood rabbit c. For each class, we also show an histogram of JFN based on 1024 samples. It shows that the JFN is a good indicator of the complexity of the class. For example, classes such as Cornet (see Figure 5.13e) or Football helmet (see Figure 5.13a) are very diverse and disconnected, resulting in high JFNs.

Fig. 5.11 Images from the Bubble class of ImageNet showing that the class is complex and slightly multimodal.

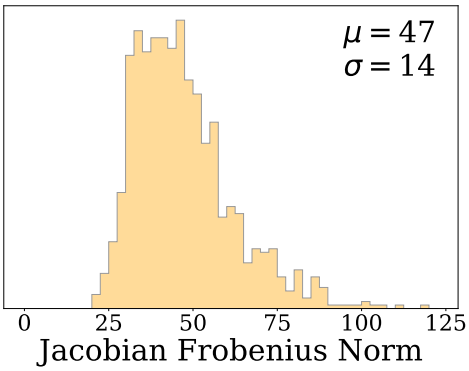
(a) 'Black swan' class.



(b) 'Black swan' class histogram.

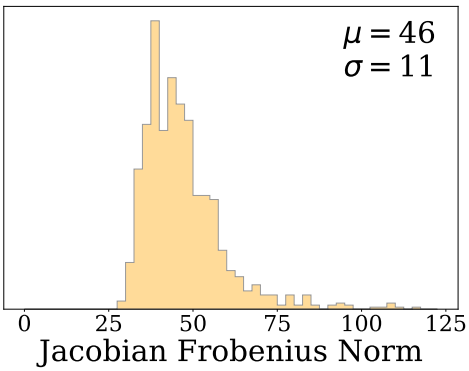


(c) 'House finch' class.



(d) 'House finch' class histogram.

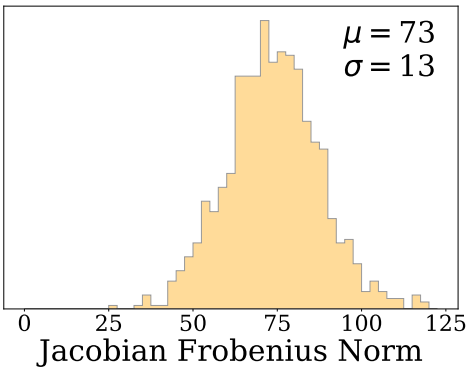
(e) 'Monarch butterfly' class.



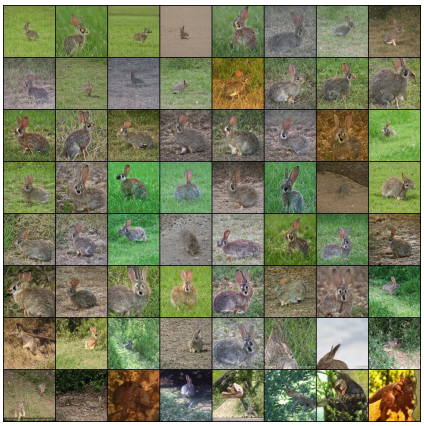
(f) 'Monarch butterfly' class histogram.

Fig. 5.12 Images

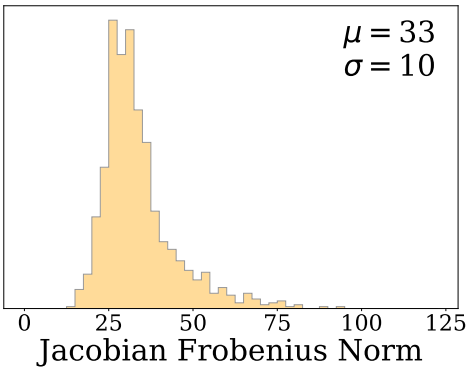
(a) 'Football helmet' class.



(b) 'Football helmet' class histogram.

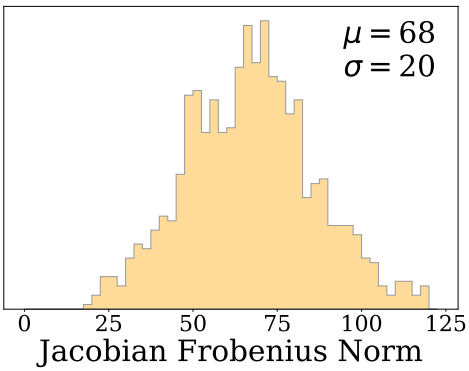


(c) 'Wood rabbit' class.



(d) 'wood rabbit' class histogram.

(e) 'Cornet' class.



(f) 'Cornet' class histogram.

Fig. 5.13 Images

Appendix 5.C Supplementary details

We now provide the different network's architecture used and their corresponding hyperparameters.

Table 5.3 Models for Synthetic datasets

Operation	Feature Maps	Activation
G(z): $z \sim \mathcal{N}(0,1)$	2	
Fully Connected - layer1	20	ReLU
Fully Connected - layer2	20	ReLU
D(x)		
Fully Connected - layer1	20	ReLU
Fully Connected - layer2	20	ReLU
Batch size	32	
Leaky ReLU slope	0.2	
Gradient Penalty weight	10	
Learning Rate	0.0002	
Optimizer	Adam: $\beta_1 = 0.5$	$\beta_2 = 0.5$

For DeliGan, we use the same architecture and simply add 50 Gaussians for the reparametrization trick. For DMLGAN, we re-use the architecture of the authors.

Table 5.4 WGAN for MNIST/Fashion MNIST

Operation	Kernel	Strides	Feature Maps	Activation
G(z): $z \sim \mathrm{N}(0, Id)$			100	
Fully Connected			$7 \times 7 \times 128$	
Convolution	3×3	1×1	$7 \times 7 \times 64$	LReLU
Convolution	3×3	1×1	$7 \times 7 \times 64$	LReLU
Nearest Up Sample			$14 \times 14 \times 64$	
Convolution	3×3	1×1	$14 \times 14 \times 32$	LReLU
Convolution	3×3	1×1	$14 \times 14 \times 32$	LReLU
Nearest Up Sample			$14 \times 14 \times 64$	
Convolution	3×3	1×1	$28 \times 28 \times 16$	LReLU
Convolution	5×5	1×1	$28 \times 28 \times 1$	Tanh
D(x)			$28 \times 28 \times 1$	
Convolution	4×4	2×2	$14 \times 14 \times 32$	LReLU
Convolution	3×3	1×1	$14 \times 14 \times 32$	LReLU
Convolution	4×4	2×2	$7 \times 7 \times 64$	LReLU
Convolution	3×3	1×1	$7 \times 7 \times 64$	LReLU
Fully Connected			1	-
Batch size	256			
Leaky ReLU slope	0.2			
Gradient Penalty weight	10			
Learning Rate	0.0002			
Optimizer	Adam	$\beta_1 : 0.5$	$\beta_2 : 0.5$	

Table 5.5 DMLGAN for MNIST/Fashion MNIST

Operation	Kernel	Strides	Feature Maps	BN	Activation
G(z): $z \sim \mathrm{N}(0, Id)$			100		
Fully Connected			$7 \times 7 \times 128$	-	
Convolution	3×3	1×1	$7 \times 7 \times 64$	-	Leaky ReLU
Convolution	3×3	1×1	$7 \times 7 \times 64$	-	Leaky ReLU
Nearest Up Sample			$14 \times 14 \times 64$	-	
Convolution	3×3	1×1	$14 \times 14 \times 32$	-	Leaky ReLU
Convolution	3×3	1×1	$14 \times 14 \times 32$	-	Leaky ReLU
Nearest Up Sample			$14 \times 14 \times 64$	-	
Convolution	3×3	1×1	$28 \times 28 \times 16$	-	Leaky ReLU
Convolution	5×5	1×1	$28 \times 28 \times 1$	-	Tanh
Encoder Q(x), Discriminator D(x)			$28 \times 28 \times 1$		
Convolution	4×4	2×2	$14 \times 14 \times 32$	-	Leaky ReLU
Convolution	3×3	1×1	$14 \times 14 \times 32$	-	Leaky ReLU
Convolution	4×4	2×2	$7 \times 7 \times 64$	-	Leaky ReLU
Convolution	3×3	1×1	$7 \times 7 \times 64$	-	Leaky ReLU
D Fully Connected			1	-	-
Q Convolution	3×3		$7 \times 7 \times 64$	Y	Leaky ReLU
Q Convolution	3×3		$7 \times 7 \times 64$	Y	Leaky ReLU
Q Fully Connected			$n_g = 10$	-	Softmax
Batch size	256				
Leaky ReLU slope	0.2				
Gradient Penalty weight	10				
Learning Rate	0.0002				
Optimizer	Adam	$\beta_1 = 0.5$	$\beta_2 = 0.5$		

Table 5.6 WGAN for CIFAR10, from Gulrajani et al. (2017)

Operation	Kernel	Strides	Feature Maps	BN	Activation
G(z): $z \sim \mathrm{N}(0, Id)$			128		
Fully Connected			$4 \times 4 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$4 \times 4 \times 128$	Y	ReLU
Nearest Up Sample			$8 \times 8 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	Y	ReLU
Nearest Up Sample			$16 \times 16 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$16 \times 16 \times 128$	Y	ReLU
Nearest Up Sample			$32 \times 32 \times 128$	-	
Convolution	3×3	1×1	$32 \times 32 \times 3$	-	Tanh
Discriminator D(x)			$32 \times 32 \times 3$		
ResBlock	$[3 \times 3] \times 2$	1×1	$32 \times 32 \times 128$	-	ReLU
AvgPool	2×2	1×1	$16 \times 16 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$16 \times 16 \times 128$	-	ReLU
AvgPool	2×2	1×1	$8 \times 8 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	-	ReLU
ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	-	ReLU
Mean pooling (spatial-wise)	-	-	128	-	
Fully Connected			1	-	-
Batch size	64				
Gradient Penalty weight	10				
Learning Rate	0.0002				
Optimizer	Adam	$\beta_1 = 0.$	$\beta_2 = 0.9$		
Discriminator steps	5				

Table 5.7 DMLGAN for CIFAR10, from Gulrajani et al. (2017)

Operation	Kernel	Strides	Feature Maps	BN	Activation
G(z): $z \sim \mathrm{N}(0, Id)$			128		
Fully Connected			$4 \times 4 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$4 \times 4 \times 128$	Y	ReLU
Nearest Up Sample			$8 \times 8 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	Y	ReLU
Nearest Up Sample			$16 \times 16 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$16 \times 16 \times 128$	Y	ReLU
Nearest Up Sample			$32 \times 32 \times 128$	-	
Convolution	3×3	1×1	$32 \times 32 \times 3$	-	Tanh
Encoder Q(x), Discriminator D(x)			$32 \times 32 \times 3$		
ResBlock	$[3 \times 3] \times 2$	1×1	$32 \times 32 \times 128$	-	ReLU
AvgPool	2×2	1×1	$16 \times 16 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$16 \times 16 \times 128$	-	ReLU
AvgPool	2×2	1×1	$8 \times 8 \times 128$	-	
ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	-	ReLU
D ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	-	ReLU
D Mean pooling (spatial-wise)	2×2	1×1	128	-	
D Fully Connected			1	-	-
Q ResBlock	$[3 \times 3] \times 2$	1×1	$8 \times 8 \times 128$	-	ReLU
Q Mean pooling (spatial-wise)	2×2	1×1	128	-	
Q Fully Connected			$n_g = 10$	-	Softmax
Batch size	64				
Gradient Penalty weight	10				
Learning Rate	0.0002				
Optimizer	Adam	$\beta_1 = 0.$	$\beta_2 = 0.9$		
Discriminator steps	5				

Conclusion

5.4 Conclusion on the present thesis

The present thesis is intended to develop methodological tools to study Generative Adversarial Networks and provide some theoretical results on GANs. More informally, it is an attempt at narrowing the gap between theory and practice.

5.4.1 Statistical study

Chapter 2 and Chapter 3 analyze properties of respectively GANs (Goodfellow et al., 2014) and Wasserstein GANs (Arjovsky et al., 2017).

To understand these two frameworks, one has to distinguish the theoretical non-parametric objectives from the practical ones involving parametric classes of discriminative functions. More formally, when allowing discriminative functions to be any measurable functions from $\mathbb{R}^D \to [0,1]$ (in GANs) or any 1-Lipschitz functions $\mathbb{R}^D \to \mathbb{R}$ (WGANs) we have the following set of optimal parameters:

$$\text{In Chapter 2, } \Theta^\star = \{\theta \in \Theta, D_{\text{JS}}(\mu_\star, \mu_\theta) = \inf_{\theta \in \Theta} D_{\text{JS}}(\mu_\star, \mu_\theta)\}.$$
$$\text{In Chapter 3, } \Theta^\star = \{\theta \in \Theta, W(\mu_\star, \mu_\theta) = \inf_{\theta \in \Theta} W(\mu_\star, \mu_\theta)\}.$$

where W is the Wasserstein distance and d_{JS} is the Jensen-Shanon divergence. In Chapter 2, Theorem 2.2.2 shows the existence and the uniqueness of the solution. However, the practitioner has to rely on a parametric discriminator $\mathscr{D}$. The main consequence of optimizing using this neural net distance is that one can only expect to find the following set of parameters:

$$\text{In Chapter 2, } \bar{\Theta} = \{\theta \in \Theta, D_{\mathscr{D}}(\mu_\star, \mu_\theta) = \inf_{\theta \in \Theta} D_{\mathscr{D}}(\mu_\star, \mu_\theta)\}.$$
$$\text{In Chapter 3, } \bar{\Theta} = \{\theta \in \Theta, d_{\mathscr{D}}(\mu_\star, \mu_\theta) = \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_\star, \mu_\theta)\}.$$

where $D_{\mathscr{D}} = \sup_{\alpha \in \lambda} \mathbb{E}_{\mu_\star} \log(D_\alpha) + \mathbb{E}_{\mu_\theta} \log(1 - D_\alpha)$ and $d_{\mathscr{D}} = \sup_{\alpha \in \lambda} \mathbb{E}_{\mu_\star} D_\alpha - \mathbb{E}_{\mu_\theta} D_\alpha$.

Both chapters study these approximation properties. Ideally, one would like $\bar{\Theta} \subset \Theta^\star$ such that any parameter obtained by the practitioner is also an optimal parameter. However, this condition is highly demanding and depends on Θ, Λ, and the global shape of the chosen loss function. Alternatively, one would like to exhibit specific parameterization under which any parameter $\bar{\theta} \in \bar{\Theta}$ has a performance ε-away to the optimal performance of a parameter $\theta^\star \in \Theta^\star$ (ε being arbitrary). Respectively, Theorem 2.3.1 and Theorem 3.3.1 show that for any given generative parameterization Θ, there exists a discriminative parameterization Λ such that this condition is verified for respectively the GANs and WGANs framework.

Interestingly, apprehending the relationship between $\bar{\Theta}$ and $\Theta^\star$ could be made possible via a better understanding of the behavior of the neural net distances $D_{\mathscr{D}}$ and $d_{\mathscr{D}}$. Chapter 3 attempted to study the question and proposed the monotonous equivalence as a way to link the Wasserstein distance d_{Lip_1} with the neural net distance $d_{\mathscr{D}}$. It is clear that many trade-offs are at play when training GANs: the more generative capacity the model has, the more demanding the monotonous equivalence is; conversely, the more discriminative capacity the model has, the more realistic the monotonous equivalence is.

As recalled in both Chapter 2 and Chapter 3, one only has access to the empirical measure μ_n of the target distribution $\mu_\star$. Therefore, due to this estimation error, one ends up with the following parameters:

$$\text{In Chapter 2, } \hat{\Theta} = \{\theta \in \Theta, D_{\mathscr{D}}(\mu_n, \mu_\theta) = \inf_{\theta \in \Theta} D_{\mathscr{D}}(\mu_n, \mu_\theta)\}.$$
$$\text{In Chapter 3, } \hat{\Theta} = \{\theta \in \Theta, d_{\mathscr{D}}(\mu_n, \mu_\theta) = \inf_{\theta \in \Theta} d_{\mathscr{D}}(\mu_n, \mu_\theta)\}.$$

Interestingly, Theorem 2.4.1 and Theorem 3.4.1 prove that both formulations are consistent. However, many questions remain unanswered:

- **Finite-sample analysis of $\varepsilon_{\text{estim}}$:** in Theorem 3.4.1, we were able to find convergence rates for $\varepsilon_{\text{optim}} + \varepsilon_{\text{estim}}$. Lemma 3.4.1 has shown that $\lim_{n \to \infty} \varepsilon_{\text{estim}} = 0$, but it would be interesting to find a finite sample upper-bound for $\varepsilon_{\text{estim}}$? This would also enable us to better understand the impact of the capacity of the discriminator D_α on this quantity $\varepsilon_{\text{estim}}$ and thus correctly parameterize it.

- **Bounding $\varepsilon_{\text{approx}} + \varepsilon_{\text{optim}} + \varepsilon_{\text{estim}}$:** this specific analysis is missing in the present thesis. A first theoretical step was made by Uppal et al. (2019) who exhibited convergence rates. However, their study specifically studied the case where both $\mu_\star$ and μ_θ are absolutely continuous with respect to the Lebesgue measure and their density lie in the same Besov ball. In high dimension, such study would not hold since it is highly likely

that none of these probability distributions would have densities (Fefferman et al., 2016). Similarly, Schreuder et al. (2020) also bring an interesting contribution in the case where the target distribution can be written as the push-forward between a multivariate uniform distribution and a smooth generator.

5.4.2 Post-processing trained generative networks

Another line of research present in this thesis has to do with the learning of disconnected manifolds with standard Generative Adversarial Networks. Using results from the Gaussian Isoperimetric inequality, Theorem 5.3.2 and Theorem 5.3.3 give lower bounds on the portion of generated samples that are mapped out of the target manifold. To solve this issue, previous works recommended to over-parameterize the model using either latent mixtures of Gaussians (Gurumurthy et al., 2017) or an ensemble set of generators (Tolstikhin et al., 2017; Khayatkhoei et al., 2018). In Chapter 5, we exhibited a simple heuristic, based on the generator's Jacobian norm, that efficiently remove off-manifold data points. Finally, another line of work relies on on the use of Monte-Carlo methods to post-process pre-trained generators (Azadi et al., 2019; Grover et al., 2019; Turner et al., 2019).

Interestingly, what we have observed so far is that these methods are very efficient when it comes to removing items that are located in between two modes of the target manifold. However, to detect blurry items within a class (or respectively when there is only one distinct class), all these methods seem to behave poorly. One of the possible hypothesis is that these two tasks might have to be tackled separately. Indeed, a single discriminator network may not be able to efficiently detect simultaneously fake items both within a given class and in-between two different classes.

5.5 Broader perspectives on GANs

Recent studies have been efficient at opening a series of broad questions on GANs. To close the present thesis, we propose a small discussion on three of the main challenges ahead for a better understanding of GANs:

- **Innovation in GANs:** there might be in GANs a discrepancy between expectations of the generative model and the chosen objective function. Since the model is trained to minimize the Wasserstein distance to the empirical distribution, it is a legitimate question to determine and understand how are generated samples linked to the training dataset. For example, could we identify what is *new* when generating the face of a person that does not exist (see, thispersondoesnotexist.com)?

- **Generalization in GANs:** theoretical research aim at understanding the ability of GANs at approximating the target distribution from finite samples. In order to improve our understanding of GANs and efficiently compare different formulations, it would be beneficial to have a clear evaluation protocol, measuring both the quality and diversity of the generated images. The Improved Precision/Recall metric (Kynkäänniemi et al., 2019) is a first step. However, the downside is that when evaluating empirically GANs, it is still not a common practice in the community to consistently have a train/test split.

- **Trade-off properties in GANs:** when it comes to training GANs, it is well-known that a tricky competition is at play between the generator and the discriminator. On the one hand, a close-to-optimality discriminator can lead to vanishing gradients (Arjovsky and Bottou, 2017) and on the other, if the discriminator does not have enough capacity, it could be easily "fooled" and one could have $d_{\mathscr{D}}(\mu_\star, \mu_\theta) = 0$ even though both probability distributions are significantly different. Similarly, Section 3.5 highlighted that increasing the capacity of the generator alone does not necessarily lead to an improved performance, mostly because it also makes the task harder for the discriminator. Interestingly, Liang (2018) also hypothesized diagrams on how the discriminator and the generator should be simultaneously parameterized. To enable further improvements, future research will have to thoroughly study the intricacies at play between both networks' capacity.

References

Acharya, D., Huang, Z., Paudel, D., and Van Gool, L. (2018). Towards high resolution video generation with progressive growing of sliced Wasserstein GANs. *arXiv.1810.02419*.

Angles, T. and Mallat, S. (2018). Generative networks as inverse problems with scattering transforms. In *International Conference on Learning Representations*.

Anil, C., Lucas, J., and Grosse, R. (2019). Sorting out Lipschitz function approximation. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 291–301. PMLR.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Precup, D. and Teh, Y., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. (2018). Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*.

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (gans). *CoRR*, abs/1703.00573.

Arora, S. and Zhang, Y. (2017). Do gans actually learn the distribution? an empirical study. *arXiv:1706.08224*.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2017). Latent space oddity: on the curvature of deep generative models. In *ICLR*.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. (2019). Discriminator rejection sampling. In *International Conference on Learning Representations*.

Biau, G., Cadre, B., Sangnier, M., and Tanielian, U. (2020). Some theoretical properties of GANs. *The Annals of Statistics*, in press.

Biau, G. and Devroye, L. (2015). *Lectures on the nearest neighbor method*. Springer.

Bjorck, A. and Bowie, C. (1971). An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8:358–364.

Blanchet, J., Kang, Y., and Murthy, K. (2019). Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56:830–857.

Bojanowski, P., Joulin, A., Lopez-Paz, D., and Szlam, A. (2018). Optimizing the latent space of generative networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 600–609. Proceedings of Machine Learning Research.

Borell, C. (1975). The brunn-minkowski inequality in gauss space. *Inventiones mathematicae*, 30(2):207–216.

Borji, A. (2019). Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, address="Oxford".

Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*.

Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. (2017). Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*.

Chernodub, A. and Nowicki, D. (2016). Norm-preserving Orthogonal Permutation Linear Unit activation functions (OPLU). *arXiv.1604.02313*.

Chongxuan, L., Xu, T., Zhu, J., and Zhang, B. (2017). Triple generative adversarial nets. In *Advances in neural information processing systems*, pages 4088–4098.

Clark, A., Donahue, J., and Simonyan, K. (2019). Adversarial video generation on complex datasets. *arXiv*, pages arXiv–1907.

Cooper, D. (1995). Learning Lipschitz functions. *International Journal of Computer Mathematics*, 59:15–26.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

Devroye, L. (1997). Universal smoothing factor selection in density estimation: Theory and practice. *TEST*, 6:223–320.

Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

Devroye, L. and Wise, G. (1980). Detection of abnormal behavior via nonparametric estimation of the support. *SIAM Journal on Applied Mathematics*, 38:480–488.

Donahue, C., Lipton, Z., Balsubramani, A., and McAuley, J. (2018). Semantically decomposing the latent spaces of generative adversarial networks. In *International Conference on Learning Representations*.

Dowson, D. and Landau, B. (1982). The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, pages 450–455.

Dudley, R. (2004). *Real Analysis and Probability*. Cambridge University Press, Cambridge, 2 edition.

Endres, D. and Schindelin, J. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49:1858–1860.

Esfahani, P. and Kuhn, D. (2018). Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171:115–166.

Fedus, W., Goodfellow, I., and Dai, A. M. (2018). Maskgan: Better text generation via filling in the _. *arXiv:1801.07736*.

Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29:983–1049.

Flamary, R. and Courty, N. (2017). POT: Python Optimal Transport library.

Fournier, N. and Guillin, A. (2015). On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162:707–738.

Gao, R., Chen, X., and Kleywegt, A. (2017). Wasserstein distributional robustness and regularization in statistical learning. *arXiv:1712.06050*.

Givens, C. and Shortt, R. (1984). A class of Wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31:231–240.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 315–323. PMLR.

Goodfellow, I. (2016). *NIPS 2016 Tutorial: Generative Adversarial Networks*. arXiv:1701.00160.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, J. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773.

Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E. J., and Ermon, S. (2019). Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11056–11068.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of Wasserstein GANs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc.

Gurumurthy, S., Kiran Sarvadevabhatla, R., and Venkatesh Babu, R. (2017). Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hales, T. C. (2001). The honeycomb conjecture. *Discrete & Computational Geometry*, pages 1–22.

He, J., Li, L., Xu, J., and Zheng, C. (2018). Relu deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.

Huster, T., Chiang, C.-Y. J., and Chadha, R. (2018). Limitations of the Lipschitz constant as a defense against adversarial examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 16–29. Springer.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

Jin, C., Netrapalli, P., Ge, R., Kakade, S., and Jordan, M. (2019). A short note on concentration inequalities for random vectors with subGaussian norm. *arXiv.1902.03736*.

Kallenberg, O. (2006). *Foundations of modern probability*. Springer Science & Business Media.

Kantorovich, L. and Rubinstein, G. (1958). On a space of completely additive functions. *Vestnik Leningrad University Mathematics*, 13:52–59.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410.

Khayatkhoei, M., Singh, M. K., and Elgammal, A. (2018). Disconnected manifold learning for generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 7343–7353.

Khrulkov, V. and Oseledets, I. (2018). Geometry score: A method for comparing generative adversarial networks. In *International Conference on Machine Learning*, pages 2621–2629.

Kodali, N., Abernethy, J., Hays, J., and Kira, Z. (2017). On convergence and stability of GANs. *arXiv.1705.07215*.

Kontorovich, A. (2014). Concentration in unbounded metric spaces and algorithmic stability. In Xing, E. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 28–36. PMLR.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Kusner, M. J. and Hernández-Lobato, J. M. (2016). Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 3927–3936. Curran Associates, Inc.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690.

Ledoux, M. (1996). Isoperimetry and gaussian analysis. In *Lectures on probability theory and statistics*, pages 165–294. Springer.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Poczos, B. (2017). MMD GAN: Towards deeper understanding of moment matching network. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2203–2213. Curran Associates, Inc.

Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727.

Liang, T. (2018). On how well generative adversarial networks learn densities: Nonparametric and parametric results. *arXiv.1811.03179*.

Lin, Z., Khetan, A., Fanti, G., and Oh, S. (2018). Pacgan: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1498–1507.

Lipton, Z. and Tripathi, S. (2017). *Precise recovery of latent vectors from generative adversarial networks*. arxiv:1702.04782.

Liu, S., Bousquet, O., and Chaudhuri, K. (2017). Approximation and convergence properties of generative adversarial learning. In Guyon, I., Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5551–5559. Curran Associates, Inc., Red Hook.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, pages 6231–6239.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 697–706.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE.

McDiarmid, C. (1989). On the method of bounded differences. In Siemons, J., editor, *Surveys in Combinatorics*, London Mathematical Society Lecture Note Series 141, pages 148–188. Cambridge University Press, Cambridge.

Mertikopoulos, P., Papadimitriou, C., and Piliouras, G. (2018). Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2703–2717. SIAM.

Mescheder, L., Geiger, A., and Nowozin, S. (2018). Which training methods for gans do actually converge? In *International Conference on Machine learning*, pages 3481–3490. PMLR.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv.1611.02163*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, 1411.1784.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

Mogren, O. (2016). C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv.1611.09904*.

Montúfar, G., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 2924–2932. Curran Associates, Inc.

Mroueh, Y. and Sercu, T. (2017). Fisher gan. In *Advances in Neural Information Processing Systems*, pages 2513–2523.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29:429–443.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-GAN: Training generative neural samplers using variational divergence minimization. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 271–279. Curran Associates, Inc., Red Hook.

O'Searcoid, M. (2006). *Metric Spaces*. Springer, Dublin.

Pandeva, T. and Schubert, M. (2019). Mmgan: Generative adversarial networks for multi-modal distributions. *arXiv:1911.06663*.

Pascanu, R., Montúfar, G., and Bengio, Y. (2013). On the number of response regions of deep feed forward networks with piece-wise linear activations. In *International Conference on Learning Representations*.

Petzka, H., Fischer, A., and Lukovnikov, D. (2018). On the regularization of Wasserstein GANs. In *International Conference on Learning Representations*.

Qi, G. (2019). Loss-sensitive generative adversarial networks on Lipschitz densities. *International Journal of Computer Vision*, pages 1–23.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Dickstein, J. (2017). On the expressive power of deep neural networks. In Precup, D. and Teh, Y., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2847–2854. PMLR.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069.

Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., and Glorot, X. (2011). Higher order contractive auto-encoder. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 645–660. Springer.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. (2017). Stabilizing training of generative adversarial networks through regularization. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2018–2028. Curran Associates, Inc.

Saito, M., Matsumoto, E., and Saito, S. (2017). Temporal generative adversarial nets with singular value clipping. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2849–2858.

Sajjadi, M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing generative models via precision and recall. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5228–5237. Curran Associates, Inc.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In Lee, D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc.

Samangouei, P., Kabkab, M., and Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*.

Schreuder, N., Brunel, V.-E., and Dalalyan, A. (2020). Statistical guarantees for generative models without domination. *arXiv:2010.09237*.

Seidel, R. (1995). The upper bound theorem for polytopes: An easy proof of its asymptotic version. *Computational Geometry*, 5:115–116.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pages 4565–4573.

Shen, Y., Gu, J., Tang, X., and Zhou, B. (2020). Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252.

Singh, S., Uppal, A., Li, B., Li, C.-L., Zaheer, M., and Poczos, B. (2018). Nonparametric density estimation under adversarial losses. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 10225–10236. Curran Associates, Inc.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318.

Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841.

Sudakov, V. N. and Tsirelson, B. S. (1978). Extremal properties of half-spaces for spherically invariant measures. *Journal of Mathematical Sciences*, pages 9–18.

Sun, K., Zhu, Z., and Lin, Z. (2019). Enhancing the robustness of deep neural networks by boundary conditional gan. *arXiv preprint arXiv:1902.11029*.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tanaka, A. (2019). Discriminator optimal transport. In *Advances in Neural Information Processing Systems*, pages 6813–6823.

Telgarsky, M. (2015). Representation benefits of deep feedforward networks. *arXiv 1509.08101*.

Telgarsky, M. (2016). Benefits of depth in neural networks. In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49, pages 1517–1539. PMLR.

Tolstikhin, I., Gelly, S., Bousquet, O., Simon-Gabriel, C.-J., and Schölkopf, B. (2017). Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems*, pages 5424–5433.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535.

Turner, R., Hung, J., Frank, E., Saatchi, Y., and Yosinski, J. (2019). Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353.

Uppal, A., Singh, S., and Poczos, B. (2019). Nonparametric density estimation and convergence rates for GANs under Besov IPM losses. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 9089–9100. Curran Associates, Inc.

van Handel, R. (2016). *Probability in High Dimension*. APC 550 Lecture Notes, Princeton University.

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, Cambridge.

Villani, C. (2008). *Optimal Transport: Old and New*. Springer, Berlin.

Virmaux, A. and Scaman, K. (2018). Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844.

Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621.

Wei, X., Gong, B., Liu, Z., Lu, W., and Wang, L. (2018). Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv:1803.01541*.

Xiang, S. and Li, H. (2017). On the effects of batch and weight normalization in generative adversarial networks. *arXiv:1704.03971*.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. (2018). Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3905–3911.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*.

Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114.

Yi, Z., Zhang, H., Tan, P., and Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-attention generative adversarial networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7354–7363.

Zhang, P., Liu, Q., Zhou, D., Xu, T., and He, X. (2018). On the discriminative-generalization tradeoff in GANs. In *International Conference on Learning Representations*.

Zhao, J., Mathieu, M., and LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv:1609.03126*.

Zhong, P., Mo, Y., Xiao, C., Chen, P., and Zheng, C. (2019). Rethinking generative mode coverage: A pointwise guaranteed approach. In *Advances in Neural Information Processing Systems*, pages 2086–2097.

Zhou, Z., Liang, J., Song, Y., Yu, L., Wang, H., Zhang, W., Yu, Y., and Zhang, Z. (2019). Lipschitz generative adversarial nets. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7584–7593. PMLR.