Quality and Diversity in Generative Models through the lens of f-divergences

Alexandre Vérine

May 06 2024

Version: Reviewer's Draft

Université Paris-Dauphine / Université PSL

Informatique
LAMSADE
Machine Intelligence and LEarning Systems - MILES

Machine Learning

Quality and Diversity in Generative Models through the lens of f-divergences

Alexandre Vérine

1. Reviewer Richard Nock
Honorary Professor, the Australian National University
Research Scientist, Google Research

2. Reviewer David Picard
École des Ponts
Senior Research Scientist

Supervisors Yann Chevaleyre
Fabrice Rossi
Benjamin Negrevergne

May 06 2024

Alexandre Vérine
Quality and Diversity in Generative Models through the lens of f-divergences
Machine Learning, May 06 2024
Reviewers: Richard Nock and David Picard
Supervisors: Yann Chevaleyre, Fabrice Rossi and Benjamin Negrevergne

Université Paris-Dauphine / Université PSL
Machine Intelligence and LEarning Systems - MILES
LAMSADE
Informatique
Place du Maréchal de Lattre de Tassigny
75 775, Paris, FRANCE

Abstract

Generative modeling have become an essential tool in machine learning for generating realistic samples from complex data distributions. Despite significant advancements in models such as Generative Adversarial Networks , Variational Autoencoders, Normalizing Flows, and Diffusion models, challenges persist in achieving a balance between sample quality and sample diversity. Precision and recall have emerged as crucial metrics for assessing the quality and diversity of generative models. Precision measures how many generated samples are coherent with the real data distribution, reflecting sample quality. Recall evaluates how many samples from the real data distribution can be generated, indicating sample diversity. This thesis addresses the fundamental problem of characterizing, tuning, and improving Precision and recall in generative models.

The first major contribution of this work is the unification of precision and recall definitions within the framework of f-divergences. By expressing the most popular metrics and their derivatives as f-divergences, we establish a cohesive and comprehensive evaluation system for generative models. This theoretical formulation allows for a clearer understanding and more precise measurement of model performance in terms of quality and diversity. Building upon this theoretical foundation, the thesis introduces a novel method for estimating the f-divergence in a tractable manner, facilitating its use as an objective function in the training of generative models. This approach enables the optimization of a specific trade-off between precision and recall, addressing a critical gap in the current literature where models often fail to achieve an optimal balance due to computational constraints. Furthermore, the thesis proposes an optimal rejection sampling method that enhances both precision and recall. This method is shown to be optimal in terms of any f-divergence, providing a robust technique for refining the outputs of pre-trained generative models. The rejection sampling algorithm is designed to operate under limited computational budgets, making it practical for real-world applications.

The experimental validation of the proposed methods is conducted on a variety of datasets, including MNIST, CIFAR-10, Fashion MNIST, CelebA, FFHQ, and ImageNet.

Using both Normalizing Flows, Generative Adversarial Networks and Diffusion Models, we demonstrate the effectiveness of our approaches in tuning the balance between quality and diversity of generated samples, and then in improving the quality. The results highlight the superiority of our methods compared to traditional metrics and existing techniques.

Résumé

Les modèles génératifs sont devenus un outil essentiel dans l'apprentissage automatique pour générer des échantillons réalistes à partir de distributions de données complexes. Malgré des avancées significatives dans les modèles tels que les Generative Adversarial Network, les Variational Autoencoders, les Normalizing Flows et les modèles de diffusion, des défis persistent pour régler le compromis entre la qualité et la diversité des échantillons. Cette thèse aborde le problème fondamental de la caractérisation, l'ajustemtn et de l'amélioration de la qualité et de la diversité dans les modèles génératifs.

La précision et le rappel ont émergé comme des métriques cruciales pour évaluer la qualité et la diversité des modèles génératifs. La précision mesure combien d'échantillons générés sont réalistes avec la distribution de données réelle, reflétant la qualité des échantillons. Le rappel évalue combien d'échantillons de la distribution de données réelle peuvent être générés, indiquant la diversité des échantillons.

La première contribution majeure de ce travail est l'unification des définitions de la précision et du rappel dans le cadre des f-divergences. En exprimant les métriques les plus populaires et leurs dérivés en tant qu'un famille de f-Divergnce, la PR-Divergence, nous établissons un système d'évaluation cohérent et complet pour les modèles génératifs. Cette formulation théorique permet une compréhension plus claire et une mesure plus précise des performances des modèles en termes de qualité et de diversité. En s'appuyant sur cette base théorique, la thèse introduit une méthode novatrice pour estimer la PR-Divergnce de manière differentiable, facilitant son utilisation comme fonction objective dans la formation des modèles génératifs. Cette approche permet d'optimiser n'importe quel compromis spécifique entre précision et rappel. Cette méthode se montre complémentaire aux méthodes existantes. De plus, la thèse propose une méthode optimale d'échantillonnage par rejet qui améliore à la fois la précision et le rappel. Cette méthode est démontrée comme étant optimale en termes de toute f-divergence, fournissant une technique robuste pour affiner les sorties des modèles génératifs pré-entraînés. L'algorithme d'échantillonnage par rejet est conçu pour fonctionner sous des budgets computationnels limités, le rendant pratique pour des applications réelles.

La validation expérimentale des méthodes proposées est réalisée sur une variété de jeux de données, incluant MNIST, CIFAR-10, Fashion MNIST, CelebA, FFHQ et ImageNet. En utilisant les Normalizing Flows, les Generative Adversarial Networks et les modèles de diffusion, nous démontrons l'efficacité de nos approches pour ajuster le compromis entre la qualité et la diversité des échantillons générés, puis pour améliorer la qualité. Les résultats soulignent la supériorité de nos méthodes par rapport aux métriques traditionnelles et aux techniques existantes.

Acknowledgement

Premièrement, je tiens à exprimer toute ma gratitude à Richard Nock et David Picard pour avoir accepté d'être les rapporteurs de ma thèse. Je tiens à les remercier pour leurs retours constructifs, mais aussi et surtout pour leur patience. J'aimerais également remercier Michèle Sebag, Guillaume Charpiat et Olivier Cappé pour avoir accepté de faire partie de mon jury de thèse. Je suis honoré d'avoir pu défendre mon travail devant vous.

Je souhaiterais remercier amicalement Yann Chevaleyre, Benjamin Negrevergne et Fabrice Rossi pour avoir respectivement dirigé, encadré et co-dirigé ma thèse. Je suis extrêmement reconnaissant d'avoir pu profiter d'une équipe d'encadrement aussi présente et complémentaire. Je tiens à remercier tout particulièrement Yann pour son savoir, sa confiance et son humour, et Benjamin pour sa sagesse, sa rigeur et son sens de l'effort. Vous avez su faire de cet encadrement, certes un moment d'apprentissage, mais surtout quatre années de joyeuse camaraderie et de complicité. J'espère sincèrement pouvoir trouver dans mon avenir des ambiances de travail aussi saines *et* productives.

Durant ma thèse, j'ai travaillé au sein de l'équipe MILES dont je remercie tous les membres pour leur bienveillance. Je voudrais exprimer ma sincère gratitude à Jamal Atif et Alexandre Allauzen pour toutes les opportunités qu'ils m'ont offertes. Ils ont su me faire confiance et m'inclure dans leur différents projets d'enseignement et de recherche. Je remercie également les doctorants de l'équipe pour leur soutien et leur amitié. Merci à Lucas pour sa bonne humeur et sa motivation d'escalade matinale inébranlable, à Florian pour nos collaborations fructeuses, aux frères Blaise et Paul pour leurs rires. A special thanks to Muni for his efficiency and his kindness. Enfin, j'aimerais remercier ceux qui m'ont accueilli dans cette équipe dès 2019. Un grand merci à Alexandre Araujo pour m'avoir si bien guidé et m'avoir permis de rejoindre l'équipe MILES, et bien sûr, Rafaël qui a été pour moi un mentor, un collègue et un ami. Merci pour ta patience, ton écoute et tes conseils.

Une thèse dans une aussi bonne ambiance ne serait pas possible sans un bon équilibre entre travail et vie personnelle. Je tiens donc à remercier ma famille et mes amis pour leur soutien indéfectible. Merci à mes parents pour avoir continué de m'encourager et de m'avoir accompagné jusqu'au dernier jour de thèse. Je ne saurais trop remercier Joséphine d'avoir partagé avec moi les moments de réussite et d'échec et merci à sa famille de s'être si bien occupé de moi.

Je clôture ces mots en espérant pouvoir transmettre à l'avenir la générosité et le soutien que j'ai reçus pendant ces années heureuses et enrichissantes de thèse, impatient de voir ce que le futur nous réserve.

Overview and Contributions

Funding and Grants

This thesis was prepared in the Machine Intelligence and LEarning Systems (MILES) team of the LAMSADE research lab at Université Paris-Dauphine / Univeristé PSL from September 2020 to June 2024. It was first funded by the Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation (MESRI) through a Contrat Doctoral Spécifique Normalien (CDSN). This work was granted access to the HPC resources of IDRIS under the allocations 2020-AD011011296, 2023-AD011014053, 2023-AD011014053 made by GENCI.

Focus of the Manuscript

The work during the Ph.D. was focused on the trade-off between quality and diversity in generative models. In the manuscript, we focus on the theoretical evaluation of quality and diversity, and to practically showcase our findings, we focus on models for image generation. We list below the contributions made during the Ph.D. in two categories: the one included in the manuscript and the one that is not included.

Contributions included in the manuscript:

- *In Chapter 4:*
  Alexandre Verine et al. "On the expressivity of bi-Lipschitz normalizing flows". en. In: Proceedings of The 14th Asian Conference on Machine Learning. *ISSN: 2640-3498. PMLR, Apr. 2023, pp. 1054–1069*
- *In Chapter 4 and Chapter 5:*
  Alexandre Verine et al. "Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows". en. In: Advances in Neural Information Processing Systems *36 (Dec. 2023), pp. 32539–32573*
- *In Chapter 6:*
  Alexandre Verine et al. "Optimal Budgeted Rejection Sampling for Generative

Models". *In:* Proceedings of The 27th International Conference on Artificial Intelligence and Statistics *(Mar. 2024). arXiv:2311.00460 [cs]*

Contributions not included in the manuscript:

- *Florian Le Bronnec et al. "Exploring Precision and Recall to assess the quality and diversity of LLMs".* *In:* Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics *(Feb. 2024). arXiv:2402.10693 [cs]*

Scientific Outreach

During this thesis, we have been involved in highlighting the importance of the trade-off between quality and diversity in generative models for a wider audience. We have presented our work on several occasions including the following:

Larger audience presentations:

- ***On the expressivity of bi-Lipschitz Normalizing*** *(Poster),*
  Prix du meilleur poster, Journée du LAMSADE, 2020
- ***PR-Divergence Optimization for generative modeling*** *(Poster),*
  Prix du meilleur poster, Dauphine Digital Days, 2023
- ***Comment Evaluer la Qualité et la Diversité des IA Génératives ?****, (talk)*
  Le Cercle Dauphine, 2024
- ***Entre réalisme et inventivité : comment cadrer les IA Génératives ?****, (media article)*
  Journal Dauphine Eclairages, 2024
- ***On the trade-off between Quality and Diversity****, (talk)*
  Conference on AI for Science, PSL DATA Program, 2024

Pedagogical Outreach

Teaching has been an essential part of the Ph.D. experience. First, teaching has been an excellent way to consolidate my pedagogical skills. I had the opportunity to teach in various programs with different audiences, pushing me to adapt my teaching methods. Second, teaching has been a way to share my knowledge and passion for the field of machine learning. I have been involved in the following teaching activities:

Teaching activities:

- ***Mathematics for Data Science*** *- Master M1 IDD*
  Université Paris-Dauphine, 2020-2023
  Lecturer: Alexandre Vérine
- ***Trustworthy Machine Learning*** *- Executive Master IASD*
  Université Paris-Dauphine, 2021-Present
  Lecturer: Alexandre Vérine, Geovanni Rikz, Blaise Delattre
- ***Deep Learning Lab*** *- Master M2 IASD*
  Université Paris-Dauphine, 2022-Present
  Lecturer: Alexandre Vérine, Benjamin Negrevergne
- ***Introduction to Deep Learning*** *- Executive Master IASD*
  Université Paris-Dauphine, 2022-Present
  Lecturer: Alexandre Vérine
- ***Hands-on Generative Modeling*** *- Certificate AI for Creative Business Dirigeants*
  We Are School - PSL, 2024-Present
  Lecturer: Alexandre Vérine, Florian Le Bronnec

Reading Guide

The manuscript has been written to be read chapter by chapter. Chapter 2 provides the necessary background on generative models, and Chapter 4 introduces the related works on quality and diversity evaluation. Even if the reader is familiar with the topic, we recommend reading these chapters to become familiar with the notation and definitions used in the manuscript. The technical chapters are Chapter 4, Chapter 5, and Chapter 6. The chapters start either with an insight, a specific related work section, or a naive approach to the problem. This section is important in order to better understand the contributions made in the chapter.

Note that the chapters are written to be easy to read. When mathematical proofs are either too long or do not add much to the understanding of the chapter, they are moved to the Appendix. The reader can refer to Appendix B for more details on the proofs. Moreover, the experimental details are moved to the Appendix C to improve the readability of the manuscript.

Contents

Notation, Symbols and Abbrevations

We denote vectors and functions with multidimensional outputs using bold lower-case letters, scalars and real-value functions with standard lower-case letters, and occasionally with upper-case letters. Matrices are denoted with upper-case bold letters. We use calligraphic fonts to indicate ensembles or subsets.The following is a non-exhaustive list of symbols and abbreviations used in this manuscript.

Notation

d, k, l n, m, K, N	Integers
u, v, w, x, y, z, a, b, c, μ, λ, σ	Scalars
$\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$, $\boldsymbol{z}$, $\boldsymbol{y}$, $\boldsymbol{z}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$	Vectors
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{I}_d$, $\boldsymbol{\Sigma}$	Matrices
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{G}$, $\mathcal{M}$, $\mathcal{P}$, $\mathcal{T}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$,	Ensembles
P, $\widehat{P}$, $\widetilde{P}$, Q	Probability measures
p, $\widehat{p}$, $\widetilde{p}$, q	Density functions

Algebra

$\mathbb{R}$	Set of real scalar
$\mathbb{N}$	Set of natural integers
$\mathbb{R}^d$	Set of real values d–dimensional vectors
$[\![0, K]\!]$	Set of integers between 0 and K
x_i	ith component of the vector $\boldsymbol{x}$
$\boldsymbol{x}_{i:j}$	Vector composed of the ith to the jth components of $\boldsymbol{x}$
$\lvert x \rvert$	Absolute value of the scalar x
$\lVert \boldsymbol{x} \rVert_p$	ℓ_p-norm of the vector $\boldsymbol{x}$
$\lVert \boldsymbol{x} \rVert_\infty$	Infinite norm of the vector $\boldsymbol{x}$
$\lVert \boldsymbol{x} \rVert$	Euclidian norm (ℓ_2) of the vector $\boldsymbol{x}$
$\det(\mathbf{A})$	Determinant of the matrix $\mathbf{A}$
$\odot$	Hadamart/Piece-wise product
I_d	Identity matrix of size $d \times d$
$\mathbf{0}_d$	Null matrix of size $d \times d$
$B_{R,\boldsymbol{x}}$	ℓ_2 Ball of center $\boldsymbol{x}$ and radius R
B_R	ℓ_2 Ball of center $\mathbf{0}$ and radius R
$\mathrm{vol}\,(\mathcal{A})$	Volume of the subset $\mathcal{A}$

Probability

$\mathbb{P}(\cdot)$	Probability of a random event
$\mathcal{P}(\mathcal{X})$	Set of all probability measure on the space $\mathcal{X}$
$\mathcal{T}$	Set of all measurable functions $\mathcal{X} \mapsto \mathbb{R}$
$\mathcal{G}$	Set of all measurable functions $\mathcal{Z} \mapsto \mathcal{X}$
$H(P)$	Shannon Entropy of the distribution P
$\mathrm{Supp}(P)$	Support of the distribution P
$\mathbb{E}_P[\cdot]$	Expected value under the distribution P
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$G\#Q$	Push-forward distribution of $Q \in \mathcal{P}(\mathcal{Z})$ by a measurable function $G \in \mathcal{G}$

Functions

$\mathbb{1}_{\{\mathcal{A}\}}$	Indicator function of the event $\mathcal{A}$
$\mathrm{sign}(x)$	Indicator function of the event $x > 0$
$\mathrm{dom}(f)$	Domain of the function f
f^*	Convex conjugate of the function f
$\nabla_x f$	Gradient of the function f with respect to x
Jac_G	Jacobian matrix of mapping G

Generative Modeling

$\mathcal{X}$	Input space
d	Dimension of the input space $\mathcal{X} \subset \mathbb{R}^d$
$\mathcal{Z}$	Latent space
m	Dimension of the latent space $\mathcal{Z} \subset \mathbb{R}^m$
P	Target distribution in $\mathcal{P}(\mathcal{X})$
Q	Latent distribution in $\mathcal{P}(\mathcal{Z})$
Θ	Set of parameters
θ	Parameters vector
G_θ, G	Mapping function
$\widehat{P}$, $\widehat{P}_G$, $\widehat{P}_\theta$	Approximated distribution
$D(P\|\widehat{P})$	Dissimilarity measure between P and $\widehat{P}$
$\mathcal{D}_f(P\|\widehat{P})$	f-divergence between P and $\widehat{P}$

Abreviation

DRS	**D**iscriminator **R**ejection **S**ampling (Sampling Method)
DOT	**D**iscriminator **O**ptimal **T**ransport (Sampling Method)
e.g.	***e***xampli ***g***ratia
Eq.	**Eq**uation
Fig.	**Fig**ure
GAN	**G**enerative **A**dversatial **N**etwork (Generative Model)
i.e.	***i***d ***e***st
MH	**M**etropolis **H**asting (Sampling Method)
NF	**N**ormalizing **F**low (Generative Model)
OBRS	**O**ptimal **B**udgted **R**ejection **S**ampling (Sampling Method)
s.t.	**s**uch **t**hat
Tab	**Tab**le
w.r.t.	**w**ith **r**espect **t**o

Introduction

1

All that glitters is not gold; often have you heard that told.”

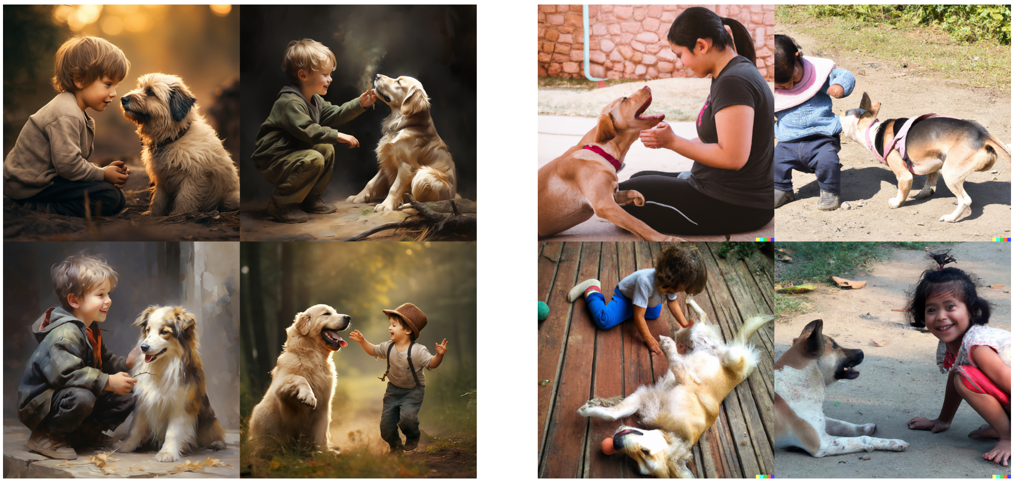
— **William Shakespeare**
The Merchant of Venice

1.1 Context and Motivation

Artificial Intelligence (AI) and Machine Learning (ML) have spread across various sectors, sparking revolutionary changes across industries. The potential of AI systems to learn patterns from data and make intelligent decisions has driven advancements in areas such as image analysis, natural language processing, and autonomous driving. A crucial task of ML, generative modeling, has become a central focus, capable of creating new data instances that resemble real-world examples.

Generative models seek to reproduce the underlying distribution of a dataset to generate new and coherent samples. This task has generated a surge of interest in various creative and practical applications, including image synthesis for computer graphics [16], style transfer in art [42], data augmentation for machine learning [107], drug molecule design in pharmaceuticals [49], and speech synthesis in natural language processing [92]. In the domain of image processing, prominent examples of models, including Generative Adversarial Networks (GANs) [44], Variational Autoencoders (VAEs) [69], Normalizing Flows [100], and Diffusion models [111], have demonstrated their effectiveness in producing high-quality data across various domains (cf. Fig. 1.1).

Formally, consider an unknown target distribution P defined in the sample space $\mathcal{X}$. A generative model is a distribution $\widehat{P}_G$ defined through the mapping G from a latent space $\mathcal{Z}$ to $\mathcal{X}$ and a distribution Q defined on $\mathcal{Z}$. The mapping function G is built, i.e. trained, such that $\widehat{P}_G$ approximates P. However, in practice, $\widehat{P}_G$ is never equal to P. For example, comparing the results of two promising models, like DALL-E 2 from OpenAI and Midjourney, reveals nuances. In the case of Midjourney's samples, they tend to appear more convincing to human observers. On the other hand, DALL-E 2's



(a) Midjourney

(b) DALL-E 2

Fig. 1.1.: Comparison of two generative models: DALL-E and Midjourney given the same prompt: *A dog playing with a child.* The Midjourney samples have a high quality but a low diversity. In every sample, the child is a young golden blond boy petting a clear seated dog outside. On the other hand, DALL-E 2 samples have a better diversity and a lower quality. The paws, hands, and faces of the subjects are less convincing, but the subjects are displayed in various situations and backgrounds with diverse ethnicity, gender and age.

samples might sometimes miss certain details, such as substituting a leg for a hand, which could influence how people perceive the model's performance. Nevertheless, DALL-E 2 manages to capture a larger variety of scenarios, backgrounds, subjects, and ethnicity, thus better encapsulating the underlying distribution, as illustrated in Fig. 1.1.

Why does this limitation occur? The first hypothesis is that it reflects the limited expressivity of existing generative models. Ideally, a model with unlimited expressiveness would perfectly match the target distribution P, capable of generating both diverse and high-quality samples. Conversely, a highly restricted model might only be capable of generating samples with either high fidelity but low diversity or a broader range but poorly generated. Although modern models have advanced significantly, they lie in a middle ground where their expressivity is still somewhat constrained. In parallel with classification tasks, performance limitations might be partly attributed to the regularization enforced on the mapping G [18, 19]. As deep learning models have grown exponentially in size and depth, certain regularization have become crucial in maintaining their stability in generative scenarios [8, 12, 16, 84, 91, 134]. Some studies suggest that, under specific assumptions regarding the disconnectedness of the support of P, performance limitations can be attributed to constraints enforce on the function G and in particular the Lipschitz constants [25, 56, 118]. However, it is crucial to note that these publications focus primarily on the on very specific metric on P [25] or metrics exclusively related to quality [56, 118].

These findings concentrate either on an unclear trade-off or just the quality, without considering diversity and quality separately.

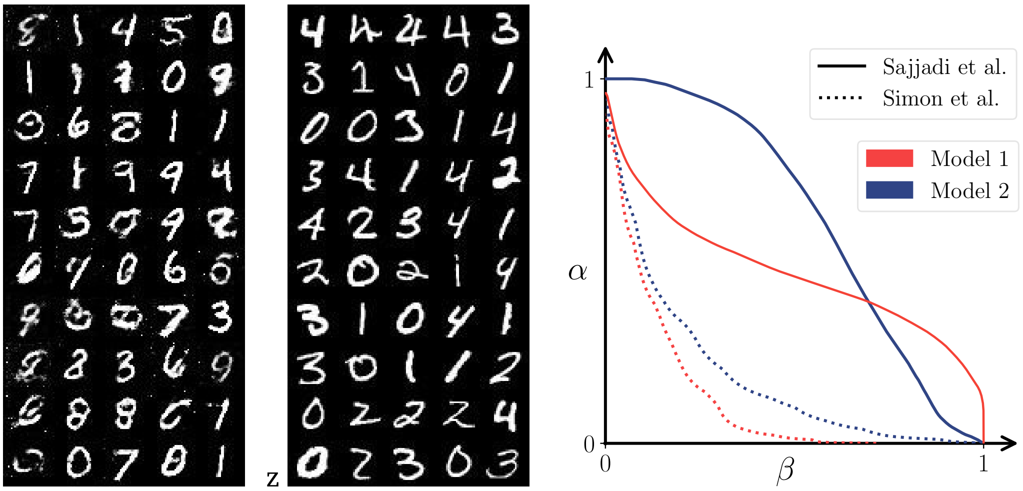
Observing these limitations, the community has predominantly directed its efforts toward models that can generate high-quality outputs. However, depending on the use-case, generative models might require high quality samples for high-resolution image and video generation, artistic synthesis or 3D model design. Alternatively, they might be required to generate high-diversity samples for applications like data augmentation, drug discovery, or anomaly detection. The divergent requirements and limitations reveal a crucial trade-off between *sample quality* and *diversity*. Traditional metrics that initially sufficed, such as the Inception Score [104] and the Fréchet Inception Distance [51] in computer vision or BLEU [94], MAUVE [96] and Perplexity [81] in Natural Language Processing, encapsulate both quality and diversity in an unclear way, highlighting the emerging requirement for metrics that can independently assess the quality and diversity of generative models.

Inspired by the classification task metrics, two notions have recently emerged to assess quality and diversity: Precision and Recall. There are strong similarities between the questions asked in the two domains.

	Precision	Recall
Classification	How many positively classified samples are positive ?	How many positive samples are positively classified ?
Generation	How many generated samples are coherent ?	How many coherent samples can be generated ?

The analogy appears to be straightforward. Precision refers to quality by capturing how much $\widehat{P}_G$ can generate samples of P. Recall refers to diversity by estimating how many samples of P can be generated by $\widehat{P}_G$. However, answering these questions in a generative task is more complex than in a classification task because the model produces diverse and novel outputs, making the definitions of true and false positives ambiguous. Unlike classifiers that produce distinct and discrete labels, generative models are defined by continuous densities, which complicates direct comparison. Furthermore, the desired qualities in generative models, like ensuring diversity without mere memorization or qualitative and novelty, do not appear in the classification metrics.

For that reason, many studies have emerged on the formal definition of precision and recall and on developing methods to compute these metrics. These metrics can be categorized into two distinct groups: those that produce a concise pair of interpretable values [21, 67, 73, 85], and those providing a more intricate and



(a) Model 1: Diverse
FID: 17.06, IS: 2.69

(b) Model 2: Precise
FID: 8.80, IS:2.57

(c) PR-Curves for Model 1 and 2.

Fig. 1.2.: Two different models are displayed with very different performances. Model 1 have a great diversity and display all different digits, but contours, backgrounds, and shapes are sometimes incoherent. Model 2 is generating coherent samples from only half the classes. Traditional metrics - FID (↓) and IS (↑) - are given for comparison.

precise evaluation through a parameterized curve [32, 103, 108]. Although the latter category, similar to the ROC curves in classification tasks, provides more comprehensive information, their complexity makes them less straightforward to interpret. For example, we computed different measures to evaluate two models that generate handwritten digits. The parameterized curves are shown in Fig. 1.2c evaluating two different models specifically tuned to be solely diverse or solely precise, respectively in Figures 1.2a and 1.2b. Concise measures are given in Tab. 1.1.

Regarding the methods used to evaluate these metrics, those that rely on k-nearest neighbors (k-NN) tend to be computationally expensive and struggle when dealing with data in higher dimensions [21, 73, 85, 103]. On the other hand, methods that involve classifiers or kernel density estimators vary greatly in results [67, 108].

	Model 1		Model 2	
	Quality	Diversity	Quality	Diversity
Sajjadi et al. [103]	0.90	0.64	0.78	0.91
Kynkäänniemi et al. [73]	0.54	0.91	0.84	0.70
Naeem et al. [85]	0.36	0.78	0.60	0.61
Simon et al. [108]	0.34	0.56	0.54	0.58

Tab. 1.1.: Different quality and diversity measures for the two models displayed in Figure 1.2a and Figure 1.2b. The higher the measure, the better the model performs. Most measures reflect that Model 1 is more diverse and Model 2 is more precise.

However, all of these methods share a common limitation, non-differentiability. This non-differentiability poses a substantial issue, making it infeasible to directly train a model for the explicit improvement of these metrics.

It is clear that a model cannot achieve both good precision and good recall. Whether due to the training procedure or the structure of the model [83, 121], it appears that a model naturally focuses on a specific but not explicit trade-off of precision and recall. Various methods have emerged in generative modeling to tune the trade-off. However, their efficacy varies. These methods only affect the training data using instance selection [28] or, are used *post hoc*, by changing the sampling procedure from Q or $\widehat{P}_G$ with a fixed mapping G [7, 9, 57, 117, 123]. Moreover, these methods usually exhibit a strong bias toward enhancing precision only, effectively pushing the model in one direction of the trade-off. Others, while promising, are resource intensive and may not be practical in real-world scenarios [47, 82]. No method currently focuses on tuning the precision-recall trade-off during training or with a limited resource cost.

1.2 Problem Statement

These motivations underscore the fundamental questions and challenges tackled in this thesis. Through an extensive investigation into generative models, we aim to answer the following question:

Question: *How can we characterize, tune, and improve precision and recall of Generative Models?*

To address this question, we divide the problem into two components: assessment and enhancement of the model. Initially, our attention is directed toward evaluating *Precision* (i.e. sample quality) and *Recall* (i.e. sample diversity). Subsequently, we explore strategies to improve the model's precision or recall.

1.2.1 Assessing Precision and Recall

To answer the question of characterizing Precision and Recall for generative models, we need a cohesive definition. Therefore, the first question we will answer is:

Question 1: *How can we unify the definitions of precision and recall for generative models?*

To do so, we will regroup the different definitions in the framework of f-divergences. We will show that the definition of PR-Curves can be written as a family of f-divergence divergences, and we will write every other definition within our framework. Once a unified evaluation system is established, we can analyze these metrics and its link to the regularization:

Question 2: *What Precision and Recall can be achieved with neural networks with bounded Lipschitz constants?*

While our work aims to be applied to generative models, we will focus on Generative Adversarial Networks and Normalizing Flows, for which the Lipschitz properties are an essential property for stability. Taking almost no assumption on the support of P, we will highlight pathological cases for which the model will fail to match the distribution.

1.2.2 Improving on Precision and Recall

Generative models face the challenge of improving both precision and recall. This is a complex task that can be achieved by multiple approaches. In our exploration, we focus on adjusting two main aspects: the loss function and the sampling method.

Adjusting the Loss Function: In this case, the only flexibility lies within the training procedure and, especially in the choice of the loss function. Under these constraints, no additional computational resources, we cannot anticipate simultaneous enhancements in both precision and recall. Nevertheless, we can adjust the balance: enabling the model to prioritize precision or recall, and particularly any explicit trade-off between those two. This leads to a fundamental question:

Question 3: *Can we train a generative model to optimize an explicit user-specified trade-off between Precision and Recall?*

We will thus leverage the theoretical analysis conducted to answer **Q1**, and develop a method to train the model to minimize an f-divergence representing a well-defined trade-off between precision and recall.

Modifying the Sampling Method: Following the loss function adjustment, we explore the possibilities within the sampling method, allowing for a slight elevation in the computational cost of generating samples. Thus, if we consider the distribution $\widehat{P}_G$ defined a by a fixed model G, and focusing on the sampling method, rejection sampling, we will answer the following question:

Question 4: *With rejection sampling under limited budget, how much can we increase Precision and Recall of a pre-trained model?*

We will demonstrate that there is a way to optimize the rejection of drawn samples in order to maximize both Precision and Recall, while being restricted by a limited budget.

1.3 Structure and Contribution

In this thesis, we aim to tackle to the four stated problems, linearly in five chapters:

- **Chapter 2** and **Chapter 3**

  In Chapter 2, we introduce various generative models in machine learning, including Generative Adversarial Networks, Diffusion models, and Normalizing Flows. The chapter provides readers with a comprehensive understanding of these models' principles and capabilities. Additionally, in Chapter 3, we present the different Precision Recall measures defined in the literature. By the end of Chapter 2 and Chapter 3, readers will have gained valuable insights into the landscape of generative models and the essential tools used to evaluate their performance in subsequent chapters.

- **Chapter 4**

  In Chapter 4, we address **Q1** and **Q2**, exploring one particular measure of "sample quality" and "sample diversity". Our key contribution is to show that a measure proposed by Simon et al. [108], can be elegantly expressed as an f-divergence, denoted the Precision-Recall divergence $\mathcal{D}_{\lambda\text{-PR}}$. This connection allows us to link $\mathcal{D}_{\lambda\text{-PR}}$ with other precision and recall concepts and establish a clear relationship between $\mathcal{D}_{\lambda\text{-PR}}$ and all other f-divergences, thus answering **Q1**. Moreover, we leverage the Lipschitz constant of Generative Adversarial Networks and Normalizing Flows. By analyzing these constants, we derive insightful lower bounds on the PR-Divergence, highlighting the limits. Throughout this chapter, to address **Q2**, we emphasize the existence of certain pathological cases that can significantly impact PR-divergence.

- **Chapter 5**

  Chapter 5, we address **Q3**, based on the insights from Chapter 3 and the PR-Divergence. While PR-Divergence demonstrates promise for evaluating generative models, we uncover the limitation that it cannot be directly optimized using existing methods. To overcome this challenge, we propose and develop a novel approach in this chapter. Our method allows models to be trained to minimize a specific $\mathcal{D}_{\lambda\text{-PR}}$, essentially allowing the optimization of a particular trade-off between precision and recall. In this chapter, we offer theoretical evidence of the convergence of our proposed method, providing reassurance of its effectiveness. Additionally, we present experimental results obtained from applying the method to both Generative Adversarial Networks and Normalizing Flows.

- **Chapter 6**

  In Chapter 6, we tackle **Q4**, focusing on a rejection sampling method with a restricted budget. We demonstrate that this approach is not only optimal but also highly efficient in practice. Using this method with a given budget, we achieve a minimal divergence after rejection. Moreover, we show that our proposed approach allows for direct minimization of the divergence between the original distribution P and the refined distribution $\widetilde{P}$.

  Through rigorous theoretical analysis and practical experimentation, we establish the effectiveness and efficiency of our proposed method, offering a robust solution to minimize divergence and refine generative models within resource constraints.

2 Background on Deep Learning Generative Models

> *I visualize a time when we will be to robots what dogs are to humans. And I am rooting for the machines.*
>
> — **Claude Shannon**
> (Father of information theory)

Contents

In this chapter, we provide a comprehensive background on Generative Models, a crucial foundation for our thesis. Our primary focus is on presenting general frameworks and algorithms, but it is essential to validate our contributions using real-world image datasets and existing models. Therefore, we introduce generative models both theoretically as mathematical tools and then practically by showing how they are implemented using deep neural networks.

To achieve this, it is advisable to begin with the Problem Statement in Section 2.1.1, followed by an presentation of f-divergences in Section 2.1.2, essential for the understanding of this thesis. Finally, we examine in Section 2.1.4 how the general framework can be extended. Then, for those inclined toward practical implementation, a detailed overview can be found in Section 2.1.3, with dedicated sections on Generative Adversarial Networks in Section 2.2.1, Normalizing Flows in Section 2.2.2 and Diffusion Models in Section 2.2.3. These "in-practice" sections are particularly relevant for understanding the experiments conducted in this thesis.

2.1 General Framework and Notations

In this section, we introduce essential notation, symbols, and fundamental definitions that establish the foundation for understanding generative models. Additional complementary notations, symbols, and abbreviations can be found in the Preface.

2.1.1 Generative Modeling: Basic Problem Statement

To define a generative model, several elementary concepts are necessary.

- Consider an input space $\mathcal{X} \subset \mathbb{R}^d$. In $\mathcal{P}$, the set of all probability distributions defined on $\mathcal{X}$, we consider a target distribution P, i.e., the data distribution. It is defined on its support $\mathrm{Supp}(P) \subseteq \mathcal{X}$, and we denote p its Radon-Nikodym density function with respect to a reference distribution μ. Typically, $\mathcal{X}$ can be a space of images of d pixels with values in $[0,1]$: $\mathcal{X} = [0,1]^d$, or it can be a set of sentences of d tokens within a vocabulary of size K, and thus $\mathcal{X} = [\![0, K]\!]^d$.
- Let us define a latent space $\mathcal{Z} \subset \mathbb{R}^m$ on which we define a latent distribution Q. The main priority set for Q is that the sampling procedure must be straightforward. For that reason, it is often chosen to be in the exponential family, or a mixture of exponential distributions. Typically, Q is chosen as a simple distribution such as a multivariate Gaussian distribution: $\mathcal{N}(\mathbf{0}_m, I_m)$. Note that the dimension of $\mathcal{Z}$ is not necessarily the same as the dimension of $\mathcal{X}$, m is usually lower or equal than d.
- Finally, consider $\mathcal{G}$ the set of measurable mapping functions G from $\mathcal{Z}$ to $\mathcal{X}$. Measurability is defined with respect to σ-algebra on $\mathcal{X}$ and $\mathcal{Z}$. For a given mapping G, we define the approximated distribution $\widehat{P}$ as the push-forward $G\#Q$, and thus we denote $\widehat{\mathcal{P}}(\mathcal{X}) = \left\{\widehat{P} = G\#Q | G \in \mathcal{G}\right\}$ the set of all distributions $\widehat{P}$ induced by the generator functions of $\mathcal{G}$ from a fixed latent distribution Q. The procedure for sampling from $\widehat{P}$ is as follows:
  1. Sample $\boldsymbol{z} \sim Q$,
  2. Compute the image $\boldsymbol{x} = G(\boldsymbol{z})$.

In general, a generative model consists of the push-forward distribution $\widehat{P}$, defined by the latent distribution Q and the mapping function G. We use the notation $\widehat{P}$ when it is clear from the context or $\widehat{P}_G$ when necessary to emphasize the dependency on G.

It is pertinent to note that we do not specify a particular function G in advance. Instead, we learn G to minimize the difference between P and $\widehat{P}$. This difference is quantified using a probability dissimilarity measure, denoted by $D(P\|\widehat{P})$, which

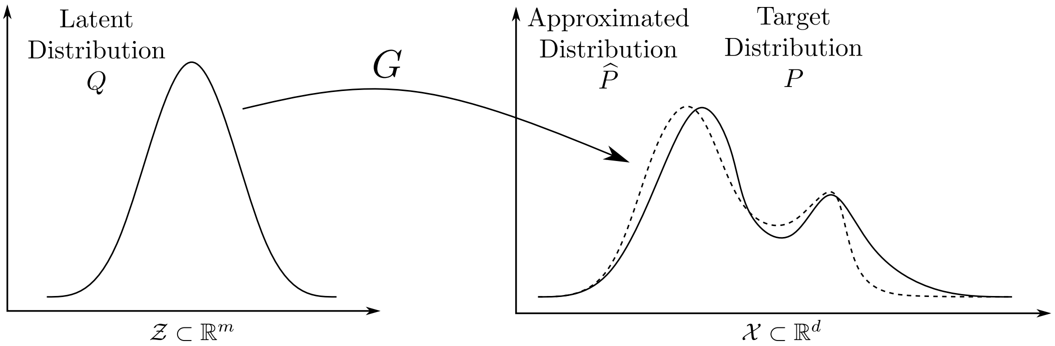

Latent Distribution Q
G
Approximated Distribution $\widehat{P}$
Target Distribution P
$\mathcal{Z} \subset \mathbb{R}^m$
$\mathcal{X} \subset \mathbb{R}^d$

Fig. 2.1.: One-dimensional illustration of a generative model. A target distribution P is approximated by a distribution $\widehat{P}$, defined by a latent distribution Q and a measurable mapping G.

evaluates how well the generated samples match the true data distribution. The objective for G is to solve:

$$\min_{G \in \mathcal{G}} D(P \| \widehat{P}). \tag{2.1}$$

In this work, our focus is on a popular class of dissimilarity measures for probability distributions known as f-divergences.

2.1.2 f-divergences to measure dissimilarity between distributions

f-divergences represent a fundamental concept in probability theory and statistics. They provide a framework for quantifying the divergence or dissimilarity between two probability distributions, which is a crucial aspect in comparing and contrasting different probability models. These divergences were introduced by Alfréd Rényi in the same paper in which he introduced the Rényi entropy [102]. Rényi's work laid the foundation for understanding these divergences, and subsequent research by Csiszár, Morimoto, and Ali & Silvey further developed the theory [4, 26]. As a result, f-divergences are sometimes known as Csiszár–Morimoto divergences, Ali–Silvey distances, or Csiszár f-divergences.

Definitions of f-divergences To define an f-divergence, we require a convex lower semi-continuous function $f : [0, +\infty] \rightarrow]-\infty, +\infty]$ that satisfies $f(1) = 0$. Typically, we consider two distributions P and $\widehat{P}$ in $\mathcal{P}(\mathcal{X})$, the set of probability measures defined on $\mathcal{X}$. We assume that P and $\widehat{P}$ are absolutely continuous with respect to a reference distribution μ on $\mathcal{X}$. Absolute continuity, also denoted $P \ll \mu$, means that for any measurable set $A \subseteq \mathcal{X}$, if $\mu(A) = 0$, then $P(A) = 0$. The f-divergence between P and $\widehat{P}$ is formally defined as follows:

Definition 2.1.1 (f-divergences).
For any two probability distributions P and $\widehat{P}$ in $\mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$. Let p and $\widehat{p}$ be the Radon-Nikodym densities of P and $\widehat{P}$ with respect to μ, respectively. Let f be any convex lower semi-continuous function $f : [0, \infty] \to] - \infty, +\infty]$ such that $f(1) = 0$, the f-divergence between P and $\widehat{P}$ is

$$\mathcal{D}_f(P \| \widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{2.2}$$

However, the definition of f-divergence can be extended to any distributions that are no longer absolutely continuous if the function $u \mapsto f(u)/(u-1)$ is non-decreasing. Note that in the literature, f-divergence is typically defined for $P \ll \widehat{P}$ as:

$$\mathcal{D}_f(P \| \widehat{P}) = \int_{\mathcal{X}} f\left(\frac{\mathrm{d}P}{\mathrm{d}\widehat{P}}\right) \mathrm{d}\widehat{P}, \tag{2.3}$$

but we will focus on the case where P and $\widehat{P}$ are absolutely continuous with respect to μ because, in this case, f-divergences can be represented using an expected value, which will prove valuable in this thesis:

$$\mathcal{D}_f(P \| \widehat{P}) = \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right]. \tag{2.4}$$

The function f is commonly referred to as the generator function, and many well-known divergences can be expressed in the form of f-divergence. This justifies the use of f-divergence as a general framework for dissimilarity measures between probability distributions, even if there exist other types of statistical divergences, such as Integral Probability Metrics or the Bregman Divergence. In Table 2.1, we provide a summary of typical divergences, their notation, definitions, and the corresponding generator function f.

f-divergences exhibit several properties, including:

- **Linearity with respect to f:** f-divergences are linear with respect to the generator function f. For any two functions f_1 and f_2 satisfying the conditions in Definition 2.1.1 and any real numbers a and b, the divergence

$$\mathcal{D}_{af_1+bf_2}(P \| \widehat{P}) = a\mathcal{D}_{f_1}(P \| \widehat{P}) + b\mathcal{D}_{f_2}(P \| \widehat{P}). \tag{2.5}$$

- **Non-Negativity:** For any two probability distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$, f-divergences satisfy the non-negativity property: $\mathcal{D}_f(P \| \widehat{P}) \geq 0$. The equality holds if and only if $P = \widehat{P}$.

Divergence	Notation	Definition	$f(u)$
Kullback-Leibler	$\mathcal{D}_{\mathrm{KL}}$	$\int_{\mathcal{X}} p(\boldsymbol{x}) \log\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x})$	$u \log u$
Reverse Kullback-Leibler	$\mathcal{D}_{\mathrm{rKL}}$	$\int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \log\left(\frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x})$	$-\log u$
Total Variation	$\mathcal{D}_{\mathrm{TV}}$	$\frac{1}{2} \int_{\mathcal{X}} \lvert p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}) \rvert \, \mathrm{d}\mu(\boldsymbol{x})$	$\frac{1}{2}\lvert u-1 \rvert$
Jeffrey	$\mathcal{D}_{\mathrm{Je}}$	$\mathcal{D}_{\mathrm{KL}}(P \Vert \widehat{P}) + \mathcal{D}_{\mathrm{KL}}(\widehat{P} \Vert P)$	$(u-1)\log(u)$
Jensen-Shannon	$\mathcal{D}_{\mathrm{JS}}$	$\frac{1}{2}\mathcal{D}_{\mathrm{KL}}(P \Vert R) + \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(\widehat{P} \Vert R)$ where $R = \frac{1}{2}(P + \widehat{P})$	$u \log u$ $+(1+u)\log(1+u)$
χ^2 Pearson	$\mathcal{D}_{\chi^2}$	$\int_{\mathcal{X}} \frac{(p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}))^2}{\widehat{p}(\boldsymbol{x})} \mathrm{d}\mu(\boldsymbol{x})$	$(u-1)^2$
Hellinger	$\mathcal{D}_{\mathrm{He}}$	$\int_{\mathcal{X}} \left(\sqrt{p(\boldsymbol{x})} - \sqrt{\widehat{p}(\boldsymbol{x})}\right)^2 \mathrm{d}\mu(\boldsymbol{x})$	$(\sqrt{u}-1)^2$
Amari α-Divergence	$\mathcal{D}_{\alpha}^{\mathrm{A}}$	$\frac{1}{\alpha(\alpha-1)} \int_{\mathcal{X}} p(\boldsymbol{x}) \left[\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)^{\alpha} - 1\right] \mathrm{d}\mu(\boldsymbol{x})$	$\frac{1}{\alpha(\alpha-1)}(u^{\alpha}-1)$

Tab. 2.1.: List of common f-divergences.

- **Joint Convexity with respect to P and $\widehat{P}$:** f-divergences are jointly convex, making them suitable for various optimization problems. For any $\lambda \in [0,1]$ and any suitable distributions $P_1, P_2, \widehat{P}_1, \widehat{P}_2 \in \mathcal{P}(\mathcal{X})$, we have the following.

$$\begin{aligned} &\mathcal{D}_f(\lambda P_1 + (1-\lambda)P_2 \Vert \lambda \widehat{P}_1 + (1-\lambda)\widehat{P}_2) \\ &\qquad\qquad \leq \lambda \mathcal{D}_f(P_1 \Vert \widehat{P}_1) + (1-\lambda)\mathcal{D}_f(\widehat{P}_1 \Vert \widehat{P}_2) \end{aligned} \tag{2.6}$$

- **Invariance with respect to f with Linear functions:** f-divergences remain invariant under affine transformations in f. For $f^{\dagger}(u) = f(u) + \gamma(u-1)$ for any constant $\gamma \in \mathbb{R}$:

$$\mathcal{D}_f(P \Vert \widehat{P}) = \mathcal{D}_{f^{\dagger}}(P \Vert \widehat{P}). \tag{2.7}$$

- **Reversal by Convex Inversion of f:** f-divergences can be reversed by applying a convex inversion. For any function f its convex inversion is defined as $g(u) := uf(1/u)$, then if f satisfies the properties to be a generator function, then g satisfies the same properties, and we have:

$$\mathcal{D}_f(P \Vert \widehat{P}) = \mathcal{D}_g(\widehat{P} \Vert P). \tag{2.8}$$

The Variational form of f-divergences One key property of f-divergences is their ability to be expressed in a dual variational form [87]. To explore this property, we first introduce the concept of the Fenchel conjugate, also called the convex conjugate.

Definition 2.1.2 (Fenchel Conjugate)**.**
For $f : \mathrm{dom}(f) \to [-\infty, +\infty]$, the Fenchel conjugate, denoted f^*, is defined as

$$f^*(t) = \sup_{u \in \mathrm{dom}(f)} \{ut - f(u)\}. \tag{2.9}$$

In this thesis, we focus on certain properties of the Fenchel conjugate that are particularly relevant for our purposes:

- **Biconjugate**: The biconjugate $(f^*)^*$, i.e. Fenchel conjugate of the Fenchel conjugate of f, is equal to f if and only if f is lower semi-continuous and convex. Thus, for f-divergences generator function:

  $$(f^*)^*(u) = f(u). \tag{2.10}$$

- **Maximizing Argument**: If the function f is differentiable, then the derivative of the Fenchel conjugates maximizes the argument in (2.9):

  $$\nabla f^*(t) = \arg \sup_{u \in \mathrm{dom}(f)} \{ut - f(u)\}, \tag{2.11}$$

  and therefore, using the biconjugate property on suitable f, the inverse function of the derivative of f is the derivative of the Fenchel conjugate and vice-versa:

  $$[\nabla f]^{-1}(t) = \nabla f^*(t) \quad \text{and} \quad [\nabla f^*]^{-1}(u) = \nabla f(u). \tag{2.12}$$

The Fenchel conjugate allows us to reformulate f-divergence $\mathcal{D}_f(P \| \widehat{P})$. Denoting $\mathcal{T}$ denoting the set of all measurable functions $\mathcal{X} \to \mathbb{R}$:

$$\begin{aligned} \mathcal{D}_f(P \| \widehat{P}) &= \int_{\mathcal{X}} \hat{p}(\boldsymbol{x}) \sup_{t \in \mathrm{dom}(f^*)} \left\{ t \frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} - f^*(t) \right\} \mathrm{d}\mu(\boldsymbol{x}) \\ &= \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} p(\boldsymbol{x}) T(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}) - \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f^*(T(\boldsymbol{x})) \mathrm{d}\mu(\boldsymbol{x}) \right) \end{aligned} \tag{2.13}$$

Thus, the f-divergence between two distributions can be written in terms of expectations, leading to its variational form:

Divergence	$f^*(t)$	$T^{\text{opt}}(\boldsymbol{x})$
$\mathcal{D}_{\text{KL}}$	$\exp(t-1)$	$1+\log\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)$
$\mathcal{D}_{\text{rKL}}$	$-1-\log(-t)$	$-\frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}$
$\mathcal{D}_{\text{TV}}$	t	$\frac{1}{2}\text{sign}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}-1\right)$
$\mathcal{D}_{\text{Je}}$	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$	$1+\log\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)-\frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}$
$\mathcal{D}_{\text{JS}}$	$-\log\left(2-\exp(t)\right)$	$\log\left(\frac{p(\boldsymbol{x})}{p(\boldsymbol{x})+\widehat{p}(\boldsymbol{x})}\right)+\log 2$
$\mathcal{D}_{\chi^2}$	$\frac{t^2}{4}+t$	$2\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}-1\right)$
$\mathcal{D}_{\text{He}}$	$\frac{t}{1-t}$	$\left(1-\sqrt{\widehat{p}(\boldsymbol{x})/p(\boldsymbol{x})}\right)$
$\mathcal{D}_{\alpha}^{\text{A}}$	$\frac{1}{\alpha}\left[(t(\alpha-1)+1)^{\frac{\alpha}{\alpha-1}}-1\right]$	$\frac{1}{\alpha-1}\left[\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)^{\alpha-1}-1\right]$

Tab. 2.2.: List of the Fenchel conjugates of the generator function of the common f-divergences. $W: t \to W(t)$ is the Lambert-W function.

Theorem 2.1.3 (Dual variational form of an f-divergence)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ two distributions such that P is absolutely continuous with respect to $\widehat{P}$ and f a suitable generator function. The f-divergence between P and $\widehat{P}$ admits a dual variational form:

$$\mathcal{D}_f(P\|\widehat{P}) = \sup_{T\in\mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[f^*(T(\boldsymbol{x}))\right]\right). \tag{2.14}$$

We use $T^{\text{opt}} \in \mathcal{T}$ to denote the function that achieves the supremum.

Using the maximizing argument property of Fenchel conjugates, we can relate the density ratio to the optimal function T^{opt}:

$$T^{\text{opt}}(\boldsymbol{x}) = \nabla f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \tag{2.15}$$

In Table 2.2, we provide a list of Fenchel Conjugates corresponding to f-divergences mentioned in Table 2.1 and the expressions of the optimal function T^{opt}.

A Note on α-divergence The (Amari) α-divergence $\mathcal{D}_{\alpha}^{\text{A}}$, introduced by Amari and Nagaoka [5], is a versatile divergence measure that generalizes many common f-divergences. In particular, when $\alpha \to 1$, it corresponds to the Kullback-Leibler divergence, and when $\alpha = 0$, it becomes the reverse KL divergence. For $\alpha = 1/2$, it is the Hellinger divergence, and for $\alpha = 2$, it is equivalent to the divergence χ^2.

Additionally, another very similar definition of α-divergence has been introduced by Tsallis [122]:

$$\mathcal{D}_{\alpha}^{\mathrm{T}}(P\|\widehat{P}) = \frac{1}{\alpha-1}\int_{\mathcal{X}} p(\boldsymbol{x})\left[\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)^{\alpha} - 1\right]\mathrm{d}\mu(\boldsymbol{x}). \tag{2.16}$$

Note that $\mathcal{D}_{\alpha}^{\mathrm{T}} = \alpha\mathcal{D}_{\alpha}^{\mathrm{A}}$. The Tsallis α-divergence, in addition to being prior to the Amari α-divergence, is related to another widely used divergence, the Rényi α-divergence denoted $\mathcal{D}_{\alpha}^{\mathrm{R}}$, which has many applications in information theory and statistics. The Rényi α-divergence is given by:

$$\mathcal{D}_{\alpha}^{\mathrm{R}}(P\|\widehat{P}) = \frac{1}{\alpha-1}\log\int_{\mathcal{X}} p(\boldsymbol{x})^{\alpha}\widehat{p}(\boldsymbol{x})^{1-\alpha}\mathrm{d}\boldsymbol{x}. \tag{2.17}$$

Even if the Rényi divergence is not an f-divergence, it can be related to the Tsallis α-divergence as $\mathcal{D}_{\alpha}^{\mathrm{R}} = \frac{\log(1+(\alpha-1)\mathcal{D}_{\alpha}^{\mathrm{T}})}{\alpha-1}$. The Rényi divergence has been extensively studied [38] and will prove to be useful in defining metrics for model evaluation.

2.1.3 Generative Models with Deep Neural Networks

As mentioned, the goal of a generative model is to minimize the difference between the target distribution P and the approximate distribution $\widehat{P}_G$ which depends on the mapping function G. In theory, if G is in the set of all measurable functions, one can easily find $\widehat{P} = P$. However, in practice, G is represented using a deep neural network, whose the architecture affects its expressivity.

Mathematically, we define a neural network by a parameter vector $\theta \in \Theta$ where Θ is the set of all possible parameter values determined by the architecture of the deep neural network, including factors such as its depth, width, structure, etc. For simplicity in notation, we will use $\widehat{P}$, $\widehat{P}_G$, or $\widehat{P}_\theta$ to refer to the learned distributions defined by the mapping function G_θ represented by θ.

In practice, the objective of training a generative model is to find the optimal parameter vector θ that minimizes the f-divergence between the target distribution P and the approximated distribution $\widehat{P}$. This objective is formalized as an optimization problem:

$$\theta^* = \operatorname*{argmin}_{\theta\in\Theta}\mathcal{D}_f(P\|\widehat{P}_\theta), \tag{2.18}$$

Since the f-divergence divergence cannot be directly calculated, it must be approximated using a set of samples drawn from P, i.e., the training dataset $\mathcal{D}$, and, if applicable, samples drawn from $\widehat{P}$. Generative models employ different methods to estimate the measure of dissimilarity. For each model, there exists an objective

Algorithm 1 Stochastic Gradient Descent (SGD) to train generative models

```
Initialize model parameters $\theta$
for $t = 1, 2, \ldots, T$ do
    Sample a mini-batch $\mathcal{B}_t^{\text{real}}$ from $\mathcal{D}$
    If necessary, sample a mini-batch $\mathcal{B}_t^{\text{fake}}$ from $\widehat{P}_\theta$
    Compute the loss $l$ using $\mathcal{B}_t^{\text{real}}$ and $\mathcal{B}_t^{\text{fake}}$
    Compute the gradient $\nabla_\theta l_t$ of the objective function w.r.t. $\theta$
    Update model parameters: $\theta \leftarrow \theta - \eta_t \cdot \nabla_\theta l_t$, where $\eta_t$ is the learning rate
end for
```

function l that is minimized to approximate or provide an upper bound for the dissimilarity measure. Detailed examples of loss functions used in state-of-the-art models are provided in Section 2.2.

To minimize the objective, one of the fundamental optimization algorithms used is the stochastic gradient descent (SGD) [14]. This algorithm iteratively adjusts the model parameters using gradients estimated from a subset of the training data. Although SGD forms the basis for many optimization techniques, variations such as mini-batch SGD and variants with adaptive learning rates have been developed to enhance convergence speed and stability [68].

In Algorithm 1 we present a simplified version of the stochastic gradient descent algorithm used to train generative models. $\mathcal{B}_t$ represents a mini-batch of data instances sampled from the training dataset in iteration t. The gradient estimate $\nabla_\theta l_t$ is calculated using the mini-batch and provides an approximation of the true gradient of the objective function with respect to the model parameters θ. The learning rate η_t determines the step size of each parameter update and is usually set using techniques such as learning rate schedules or adaptive learning rate methods [35]. Although Algorithm 1 captures the core, modern generative models may incorporate adaptations and novel optimization techniques for better convergence and stability. For example, one common approach is to regularize the parameter vector θ, enforcing on G_θ Lipschitz constraints defined in Section 4.4.

Throughout this thesis, we will focus on various aspects of this training procedure, exploring different f-divergences, different objective functions, and optimization techniques. This exploration aim to improve our understanding of the trade-offs and challenges in training generative models, and in particular the trade-off between quality and diversity.

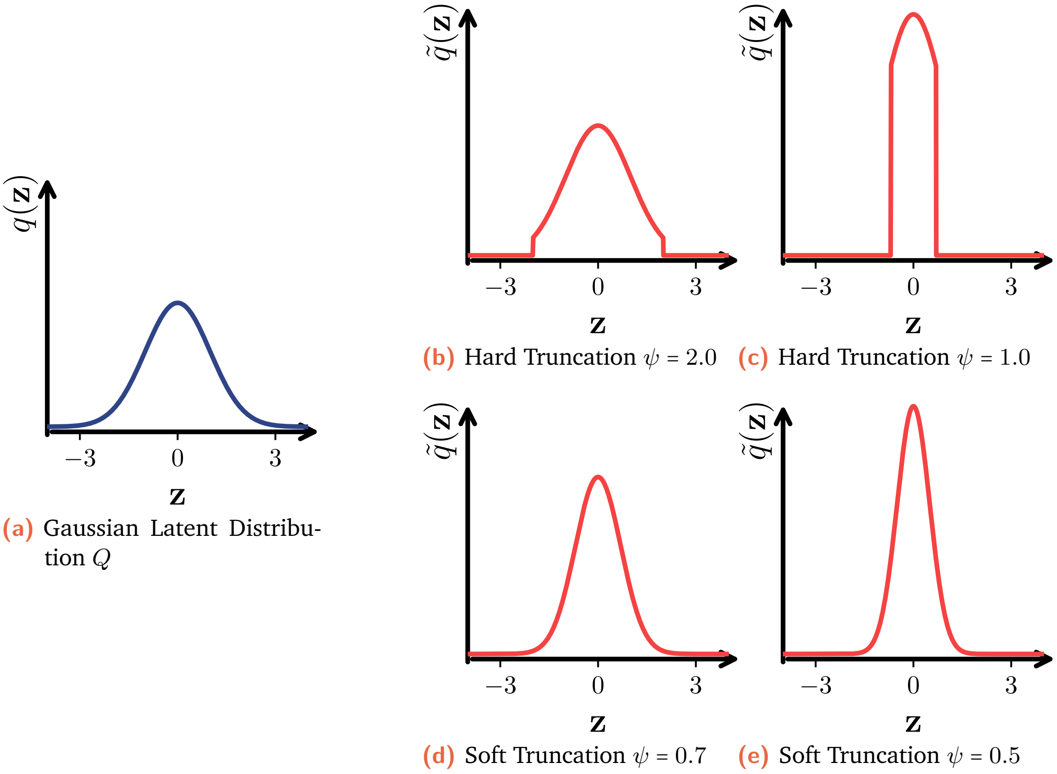
2.1.4 Extension of the General Framework

The general framework for generative models that we have discussed so far includes a latent distribution Q, a mapping function G_θ, and the associated push forward distribution $\widehat{P}_G = G_\theta \# Q$. However, the practical implementation can be more complex, leading to various extensions of the framework.

In practice, certain models are trained to minimize $\mathcal{D}_f(P \| \widehat{P}_G)$, but may not draw samples as expected. For example, one common trick to improve generative model quality involves the following steps:

- Sample $\boldsymbol{z} \sim \widetilde{Q}$, a modified version of Q.
- Compute the image $\boldsymbol{x} = G(\boldsymbol{z})$.

Depending on the community, $\widetilde{Q}$ is either a truncated version of Q, which is typically denoted as Hard Truncation and illustrated in Figure 2.2b [16, 105], or a rescaled version of Q, which is denoted as Soft Truncation [31, 70] and illustrated in Figure 2.2d. These modifications can improve the quality of the generated samples by restricting the latent space to a region where the model performs better.



(a) Gaussian Latent Distribution Q

(b) Hard Truncation $\psi = 2.0$ (c) Hard Truncation $\psi = 1.0$

(d) Soft Truncation $\psi = 0.7$ (e) Soft Truncation $\psi = 0.5$

Fig. 2.2.: Illustration of the hard and soft truncation methods.

Furthermore, there are various methods for sampling from $\widehat{P}$. For instance, in rejection sampling Azadi et al. [9] and Von Neuman [129], which will be detailed in Section 6.1.2, the sampling procedure can be more complex:

- Sample $\boldsymbol{z} \sim Q$.
- Compute the image $\boldsymbol{x} = G(\boldsymbol{z})$.
- Accept or reject the image $\boldsymbol{x}$ based on some function $a(\boldsymbol{x})$.

In summary, in practice, there exist multiple ways to sample from Q and, more importantly, from $\widehat{P}$. These variations and extensions illustrate the complexity of implementing generative models.

2.2 State-of-the-Art Models

Generative models differ in how they estimate the dissimilarity measures, leading to variations in the objective function l and the mapping function G. In recent years, three key architectures, Generative Adversarial Networks (GANs), Normalizing Flows (NFs), and Diffusion Models, have played a significant role in advancing deep generative models. These models have remained popular for several reasons. GANs have been at the forefront of various applications for an extended period, NFs possess a unique ability to model data density accurately, and Diffusion Models have undergone substantial development in recent years.

2.2.1 Generative Adversarial Networks

Introduced by Goodfellow et al. [44], GANs have continued to evolve, generating high-quality samples. At the core of this architecture is a min-max optimization problem involving two networks: the generator G and the discriminator T. The generator tries to create data samples that are indistinguishable from real ones, while the discriminator attempts to differentiate between the real and generated samples. This forms a two-player minimax game, mathematically represented by:

$$\min_{G} \max_{T} \left(\mathbb{E}_{\boldsymbol{x} \sim P} \left[\log T(\boldsymbol{x}) \right] + \mathbb{E}_{\boldsymbol{z} \sim Q} \left[\log(1 - T(G(\boldsymbol{z}))) \right] \right). \tag{2.19}$$

This equation represents the objective function of the GAN, where $T(\boldsymbol{x})$ is the discriminator's estimate of the probability that real data instance $\boldsymbol{x}$ is real, and $G(\boldsymbol{z})$ is the data instance generated by the generator. The generator and the discriminator are trained alternately, with the generator aiming to produce data that the discriminator cannot distinguish from real data, hence minimizing the log-probability of the discriminator being correct.

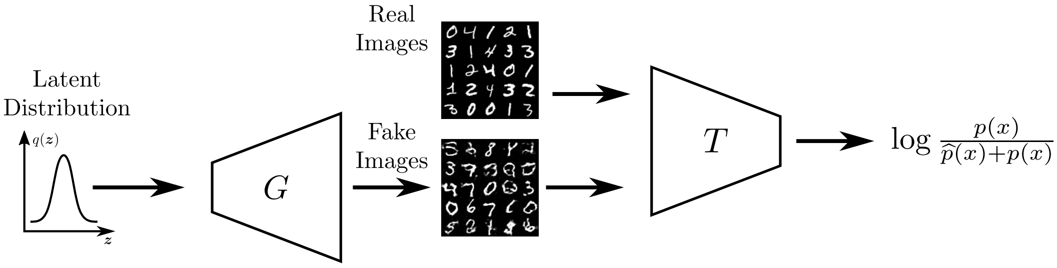


Fig. 2.3.: Original structure of a GAN introduced by Goodfellow et al. [44].

Originally, GANs were designed to minimize one specific divergence, that we will denote $\mathcal{D}_{\mathrm{GAN}}$. However, the model has since been extended to minimize any f-divergence. Nowozin et al. [90] introduced the f-GAN framework, based on the variational estimation of the f-divergence presented in Equation (2.14) and solves the following objective function:

$$\min_{G \in \mathcal{G}} \max_{T \in \mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x} \sim P} \left[T(\boldsymbol{x}) \right] - \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}_G} \left[f^*(T(\boldsymbol{x})) \right] \right). \tag{2.20}$$

T is trained to improve the estimation of any f-divergence between the target distribution P and the learned distribution $\widehat{P}$, while G is trained to minimize this estimation. As illustrated in Figure 2.4 and in Equation (2.20), T is trained using samples from both P and $\widehat{P}$, while G is trained solely using samples from P. We can show that if the discriminator in the original framework in Equation (2.19) as $\boldsymbol{x} \mapsto \log(T(\boldsymbol{x}))$ then we can show that the objective function of the original GAN is equivalent to minimizing the Jensen-Shannon divergence:

$$\mathcal{D}_{\mathrm{GAN}}(P \| \widehat{P}) = 2\mathcal{D}_{\mathrm{JS}}(P \| \widehat{P}) - \log(4). \tag{2.21}$$

$\mathcal{D}_{\mathrm{GAN}}$ is not an f-divergence as it does not, for instance, satisfy the positivity property. For simplicity, we will frequently refer to it as a divergence and use both terms interchangeably. In principle, GANs have the capacity to minimize various f-divergences. However, the neural network architectures that used to parametrize the functions G and T have witnessed substantial evolution in recent years:

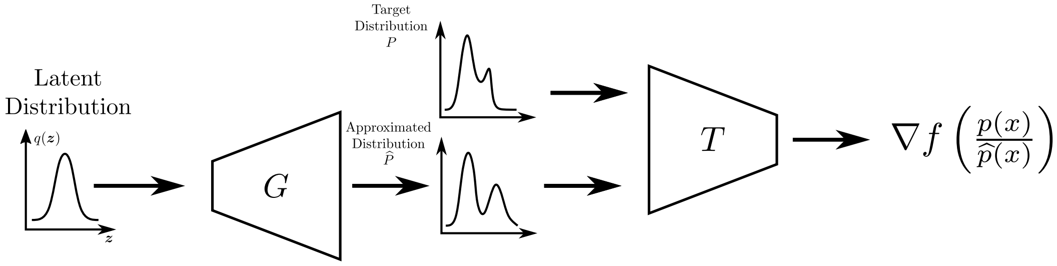

Target Distribution
P
Latent Distribution
$q(z)$
z
G
Approximated Distribution
$\widehat{P}$
T
$\nabla f \left(\frac{p(x)}{\widehat{p}(x)} \right)$

Fig. 2.4.: Structure of an f-GAN introduced by Nowozin et al. [90].

- **PGGAN (2017):** The Progressive Growing of Generative Adversarial Networks, abbreviated as PGGAN, marked a significant advancement in the quality of generated images. Introduced by Karras et al. [60], the PGGAN systematically upscales the resolution of generated images by adding new layers to both the generator and the discriminator progressively during the training process. Initially, training starts with low-resolution images, which significantly simplifies the network learning task. As training progresses, new layers are added to both the generator and the discriminator to increase the resolution. This progressive strategy not only improved the quality of the generated images, but also significantly accelerated the training process. It enabled the generation of images with resolutions up to 1024×1024 pixels, which was a groundbreaking achievement at the time of its introduction. Moreover, the PGGAN introduced several beneficial techniques for stabilizing training, such as mini-batch standard deviation layer, equalized learning rate, and pixel-wise normalization, which have since become commonplace in the training of advanced GAN models.
- **W-GAN (2017):** Introduced by Arjovsky et al. [8], Wasserstein GAN (W-GAN) marked a significant stride in the training stability of GANs. Central to W-GAN is the Wasserstein distance metric which replaces the Jensen-Shannon divergence used in the original GAN formulation. This change addresses the issue of mode collapse often witnessed in traditional GANs, fostering more stable and robust training dynamics. A vital component of W-GAN is the enforcement of a 1-Lipschitz constraint on the discrimintator. This constraint is critical to guarantee the validity of the Wasserstein distance as a meaningful loss metric in the training process. By adopting this Lipschitz condition, W-GAN encourages smoother and more meaningful gradients, facilitating a balanced growth between the generator and the discriminator and thus avoiding the dreaded mode collapse and fostering more diversified generative outcomes.
- **SAGAN (2018):** Self-Attention Generative Adversarial Networks (SAGAN), introduced in a paper by Zhang et al. [134], revolutionized the capabilities of Generative Adversarial Networks by incorporating self-attention mechanisms within the network structure. This key addition enables the model to focus on long-range spatial relationships, thereby capturing patterns and structures that were previously elusive. Moreover, SAGAN introduced Spectral Normalization, a technique used to stabilize the training of the generator; it controls the Lipschitz constant of the model by normalizing the spectral norm of the weight matrices.
- **BigGAN** (2018) [16]: BigGAN, which stands for "Big Generative Adversarial Networks", represents a remarkable milestone in the development of GANs, particularly in the generation of high-fidelity and high-resolution images. Introduced by Brock et al. [16], this model is characterized by its substantial

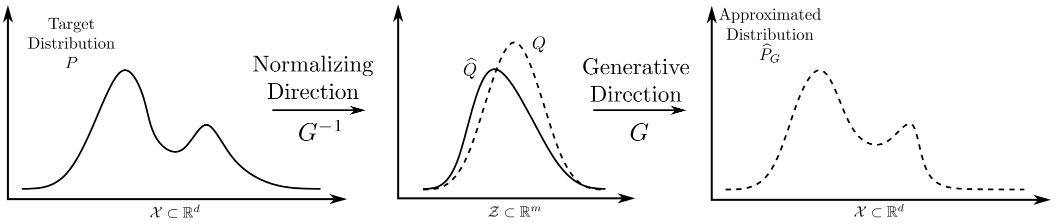


Fig. 2.5.: Visualization of the bidirectional mapping in Normalizing Flows. The figure illustrates how NFs perform a bidirectional transformation: pushing the target distribution $\mathcal{P}$ into the latent space $\mathcal{Z}$ using the generative direction and mapping the latent distribution $\mathcal{Q}$ to the approximated distribution $\widehat{\mathcal{P}}$ using the normalizing direction through the function G.

scaling in both the depth and width of the network, as well as in the batch size during training. The primary novelty of BigGAN lies in the utilization of a modified training objective, which incorporates a hinge loss function into the original GAN loss function. This alteration in the loss function can be mathematically represented as:

$$\min_{G\in\mathcal{G}}\max_{T\in\mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x}\sim P}\left[\min(0,-1+T(\boldsymbol{x}))\right] + \mathbb{E}_{\boldsymbol{z}\sim Q}\left[\min\left(0,-1-T\left(G\left(\boldsymbol{z}\right)\right)\right)\right]\right). \tag{2.22}$$

In addition, BigGAN introduced other vital techniques, such as class-conditional batch normalization, which allows for the incorporation of class labels into both the generator and discriminator, enabling the generation of more class-consistent images.

- **StyleGAN series** (2019-2022) [62, 63, 64, 105]: Introduced varied styles and scales with subsequent versions offering alias-free generators. Aliasing is a phenomenon where signals become indistinct when sampled. In the case of image generation, aliasing can result in patterns or textures that appear distorted, pixelated, or show Moiré patterns. An alias-free generator utilizes specific techniques to prevent these issues, often employing multi-scale architectures, inspired by PGGAN, and strategies for smooth sampling.

2.2.2 Normalizing Flows

Normalizing Flows (NFs) are a crucial type of generative model because they can track data density. This makes them useful in various applications such as physics simulations [59, 74, 100], anomaly detection [29, 106], noise modeling [1], sound generation [98] and Boltzmann samplers [75]. In theory, Normalizing Flows are defined as a bijection, an invertible mapping between the data space $\mathcal{X}$ and the latent space $\mathcal{Z}$. This transformation can work in two directions: the forward pass $G : \mathcal{X} \to \mathcal{Z}$, denoted as *generative direction*, and the inverse direction $G^{-1} : \mathcal{Z} \to \mathcal{X}$, denoted the *normalizing direction*. As illustrated in Figure 2.5, the bijectivity is crucial because it enables pushing the latent distribution into the image space, similar to GAN, but it also enables pushing the target distribution into the latent space, thus defining $\widehat{Q} = G^{-1} \# P$. In doing so, we can compute the divergence $\mathcal{D}_f(\widehat{Q} \| Q)$, which is equal to $\mathcal{D}_f(P \| \widehat{P})$. To do so, Normalizing Flows rely on the change of variable formula to track the density:

$$\widehat{p}(\boldsymbol{x}) = q(G^{-1}(\boldsymbol{x})) \,|\det J_{G^{-1}}(\boldsymbol{x})|, \tag{2.23}$$

The change of variable formula is composed of two terms: the term $q(G^{-1}(\boldsymbol{x}))$ accounting for where the point is mapped and the term $|\det J_{G^{-1}}(\boldsymbol{x})|$ accounting for how the mapping locally dilates or expand the space. Using this formula, a Normalizing Flow is usually trained by direct maximum likelihood estimation (MLE). More precisely, the model is training by maximizing the log-likelihood:

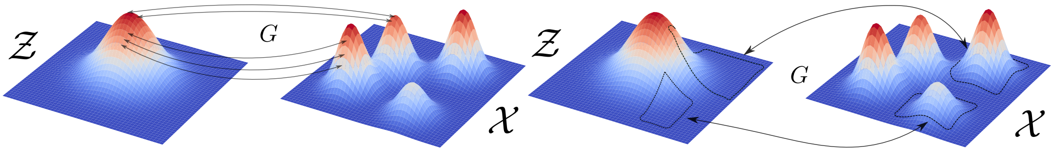
$$\begin{aligned} &\max_{G \in \mathcal{G}} \; \mathbb{E}_{\boldsymbol{x} \sim P} \left[\log \widehat{p}_G(\boldsymbol{x})\right] \\ &\qquad = \max_{G \in \mathcal{G}} \mathbb{E}_{\boldsymbol{x} \sim P} \left[\log\left(q\left(G^{-1}(\boldsymbol{x})\right)\right) + \log\left(|\det J_{G^{-1}}(\boldsymbol{x})|\right)\right]. \end{aligned} \tag{2.24}$$

Note that training by MLE is equivalent to training the model to minimize the Kullback-Leibler divergence $\mathcal{D}_{\mathrm{KL}}(P \| \widehat{P})$. With $H(P)$, the entropy of the target distribution P, we have the following.

$$\max_{G \in \mathcal{G}} \; \mathbb{E}_{\boldsymbol{x} \sim P} \left[\log \widehat{p}_G(\boldsymbol{x})\right] = H(P) - \min_{G \in \mathcal{G}} \; \mathcal{D}_{\mathrm{KL}}(P \| \widehat{P}_G). \tag{2.25}$$

However, Grover et al. [46] introduced the Flow-GAN framework to train NFs to minimize any f-divergence by using a discriminator T trained to estimate the $\mathcal{D}_f(P \| \widehat{P})$. Furthermore, since the GAN training procedure is known to be unstable, Grover et al. [46] showed that adding a log-likelihood to the min-max objective is very efficient in stabilizing the optimization:

$$\min_{G \in \mathcal{G}} \left(\max_{T \in \mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x} \sim P} \left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x} \sim \widehat{P}_G} \left[f^*(T(\boldsymbol{x}))\right] \right) + \gamma \mathbb{E}_{\boldsymbol{x} \sim P} \left[\log \widehat{p}_G(\boldsymbol{x})\right] \right). \tag{2.26}$$



(a) Illustration of $q(G^{-1}(\boldsymbol{x}))$.

(b) Illustration of $|\det J_{G^{-1}}(\boldsymbol{x})|$.

Fig. 2.6.: Components of the change of formula in Normalizing Flows. The first term, $q(G^{-1}(\boldsymbol{x}))$, maps data points in the latent space, while the second term, $|\det J_{G^{-1}}(\boldsymbol{x})|$, represents the local expansion or contraction of the space.

In practice, the set $\mathcal{G}$ is restricted to the bijection that can be represented by neural networks. There exist many ways to build an invertible neural network, but according to Kobyzev et al. [72], to be practical, Normalizing Flows should satisfy the following critical conditions:

- Be invertible, so that G is used for sampling and G^{-1} is used for computing the likelihood.
- Be expressive enough to accurately model the desired distribution.
- Offer computational efficiency, which encompasses both the calculations involving G and G^{-1}, and the calculation of the determinant of the Jacobian.

These can be designed using various strategies, including Ordinary Differential Equations (ODEs) [23], autoregressive models [71, 93], or residual networks [11, 22] and coupling-based models, affine [30, 31], mixture of continuous distributions [52], splines [37]. In this thesis, we focus on a few architectures:

- **NICE:** Introduced by Dinh et al. [30], NICE (Non-linear Independent Components Estimation) introduced the concept of normalizing flows. This model utilizes additive coupling layers, which facilitates the computation of the Jacobian determinant (equal to 1). However, its expressiveness was somewhat limited, making it less adaptable compared to subsequent models. A typical transformation in NICE, represented using additive coupling layers, is given by:

$$\begin{cases} \boldsymbol{y}_{1:i} &= \boldsymbol{x}_{1:i}, \\ \boldsymbol{y}_{i+1:d} &= \boldsymbol{x}_{i+1:d} + t(\boldsymbol{x}_{1:i}), \end{cases} \tag{2.27}$$

  where x and y are the input and output vectors respectively, and t is a learned translation function implemented through a neural network. Invert mapping uses $\boldsymbol{y}_{1:i} = \boldsymbol{x}_{1:i}$ to reverse the applied translation $t(\boldsymbol{x}_{1:i})$.

- **RealNVP:** Further developed by Dinh et al. [31], RealNVP extends the work initiated by NICE by incorporating affine coupling layers, which can learn more complex data distributions, while maintaining the computational efficiency in the determination of exact likelihoods. The mathematical representation of the affine coupling layers in RealNVP is as follows:

$$\begin{cases} \boldsymbol{y}_{1:i} &= \boldsymbol{x}_{1:i}, \\ \boldsymbol{y}_{i+1:d} &= s(\boldsymbol{x}_{1:i}) \odot \boldsymbol{x}_{i+1:d} + t(\boldsymbol{x}_{1:i}), \end{cases} \tag{2.28}$$

  Similarly to NICE, the invert function depends on inverting a scaling and translating operation that only requires $\boldsymbol{y}_{1:i} = \boldsymbol{x}_{1:i}$. In this architecture, the determinant Jacobian matrix is computed as $\prod_j s(\boldsymbol{x}_{1:i})_j$.

- **GLOW:** The GLOW model, introduced by Kingma and Dhariwal [70], further enhanced the expressiveness of normalizing flows by integrating 1x1 convolutions and a more flexible coupling layer structure, facilitating the learning of complex distributions without significantly increasing the computational load. Its characteristic coupling layer is defined similarly to RealNVP but includes the 1x1 convolution operation, defined a matrix:

$$\mathbf{W} = \mathbf{PL}(\mathbf{U} + \mathrm{diag}(\boldsymbol{s})), \tag{2.29}$$

  where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ and $\mathbf{U}$ are, respectively, lower and upper triangular matrix with ones on the diagonal. With this structure, the determinant of the Jacobian matrix can be computed for the 1×1 convolution as $\prod_j s_j$.

- **ResFlow:** ResFlow, or Residual Flow, was proposed in works by Behrmann et al. [11] and Chen et al. [22]. Its structure, based on Deep Residual Networks [50] blocks, allows for rich expressiveness and deep architectures without suffering from the problems related to training deep networks. A residual connection in ResFlow can be mathematically represented as:

$$\boldsymbol{y} = \boldsymbol{x} + F(\boldsymbol{x}), \tag{2.30}$$

  where $F(\boldsymbol{x})$ is a function representing a series of transformations on the input $\boldsymbol{x}$. To ensure invertibility, the function F is constraint to be Lipschitz. If the Lipschitz constant is lower than 1, the inverse can be computed with an iterative algorithm. There is no close form for the determinant of the Jacobian in this architecture, but it is approximated using a Russian Roulette estimator.

Similarly to GANs models, the Lipschitz continuity plays a fundamental role in guaranteeing stability. As highlighted by Behrmann et al. [12], maintaining Lipschitz constraints prevents issues such as the concentration of measure phenomenon, which can cause numerical instability and hamper successful model training. This constraint, while necessary in ResFlow models, is also an important property of RealNVP or GLOW models, for instance.

2.2.3 Diffusion Models

Diffusion models have undergone numerous advancements in recent years. Initially conceptualized as denoising models [53, 112], the scope has expanded with the development of Score-Matching Diffusion Models [61, 114] which we detail in this section.

This model is based on diffusion processes $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ defined by a Îto SDE:

$$\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, \tag{2.31}$$

where $\boldsymbol{f}(.,t) \in \mathbb{R}^d \to \mathbb{R}^d$ is the drift, $g(t) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient, and $\boldsymbol{w} \in \mathbb{R}^d$ is a standard Wiener process. It defines a sequence of distributions $\{P_t\}_{t\in[0,T]}$, with densities p_t. With this definition, the target distribution is $P = P_0$ and $\boldsymbol{f}$, g and T are chosen such that P_T tends toward a tractable distribution Q. In practice, Q is a normal distribution in $\mathbb{R}^d$. The principle of the generative model is based on the reverse SDE introduced by Anderson [6]:

$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g(t)^2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}, \tag{2.32}$$

where $\mathrm{d}\bar{\boldsymbol{w}}$ denotes a different standard Wiener process. Therefore, if the score function $\boldsymbol{s}(t, \boldsymbol{x}_t) := \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ is known, the reverse process can be simulated to go from P_T to P_0. Thus, by applying the reverse process with a learned score function $\boldsymbol{s}_\theta$ to data points drawn from Q it defines the distributions $\widehat{P}$. As shown in the work of Song et al. [113], this principle is used to train the model to minimize the KL divergence between the target distribution P and the approximated distribution $\widehat{P}$:

$$\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) = \mathcal{D}_{\mathrm{KL}}(P_T\|Q) + \int_0^T g(t)\mathbb{E}_{\boldsymbol{x}_t\sim P_t}\left[\|\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) - \boldsymbol{s}_\theta(t, \boldsymbol{x}_t)\|^2\right]\mathrm{d}t. \tag{2.33}$$

In practice, the score function is estimated with a neural network that is typically based on U-Net architectures [101]. While they are valuable for comparing different generative modeling techniques due to their outstanding performances in image generation tasks, we will not dive into the details of their training in this thesis.

Thesis Scope and Approach

This thesis focuses on the training of diverse generative models, placing particular emphasis on the types of loss functions employed. We will both consider the type of f-divergence and also the arguments of f-divergence. We will be required to change not only the objective function, but also the training algorithm, when necessary.

Our work focuses mainly on models that can be trained to easily minimize any f-divergence. Therefore, we will consider GANs in the f-GAN framework and Normalizing Flows in the Flow-GAN framework. The diffusion models will be considered as a reference for the training of generative models.

It is important to clarify that this thesis does not dive deeply into the nuances of neural network architectures for these models. Rather, our focus is on a comprehensive exploration of generative models as a whole, with findings applicable to a wide range of architectures. When a model is being trained, retrained, or fine-tuned, we fix the neural architecture and evaluate the method within the set $\mathcal{G}$ defined by this architecture.

After exploring the theoretical and practical aspects of generative models, we are prepared to discuss model evaluation methods. Generative models have broad applications, but assessing their performance can be challenging. In the following sections, we will explore evaluation methodologies and metrics to better understand these models and their real-world applications.

3 A panorama of Precision Recall measures

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

— **Alan Turing**
(Father of Computer Science)

Contents

In the previous chapter, we discussed how generative models are trained to minimize a dissimilarity measure between the target and generated distributions, typically an f-divergence. The method to approximate this metric to minimize can vary depending on the type of model, and thus the objective function is generally model-specific. For a fair and consistent assessment of generative models, it is crucial that the metrics used are model-agnostic. The method and algorithm to compute the evaluation metrics must be identical for any generative model. Moreover, to be easily computable, they should depend solely on a set of samples drawn from both P and $\widehat{P}$, without the need for additional training.

As the performance of generative models increases, the need for more refined metrics has become more pronounced. Traditional metrics such as the Fréchet Inception Distance (FID) and Inception Score (IS) have become less effective for comparing state-of-the-art models. As we shall see in Section 3.1, FID and IS do not independently assess quality and diversity, highlighting the need for additional metrics specifically designed to address this limitation. Consequently, we will examine the concepts of Precision and Recall as they apply to generative models, focusing on two main interpretations: one based on the support of distributions in Section 3.2.2, and another based on density estimations in Section 3.2.1. In addition, we introduce the Coverage and Density metrics, which offer another perspective on quality and diversity, in Section 3.3.1, but also the Precision-Recall Divergence Frontier in Section 3.3.2 and the Precision-Recall Cover in Section 3.3.3, three more alternative methods. The final part of this chapter, Section 3.4 outlines the existing dependency between all these metrics.

This chapter is intended to provide an overview of the various metrics used to evaluate generative models and in particular the ones recently introduced to assess the quality and diversity of the generated samples. We will also discuss the limitations of these metrics and the challenges they present. The goal is to provide a comprehensive understanding of the current state-of-the-art in generative model evaluation and to highlight the need for further research in this area.

3.1 Inception Score and Fréchet Inception Distance

The Inception Score and the Fréchet Inception Distance emerged as the first widely accepted metrics to benchmark generative models. As direct evaluation of model-generated samples in pixel space presents significant complexity, both IS and FID utilize Inception-v3 [116], a model pre-trained on classification tasks, for their computations. This model has proven its efficacy on the ImageNet dataset. Although IS and FID both rely on Inception-v3, they employ this model distinctively to evaluate various attributes of the images produced by generative models.

Inception Score (IS): The Inception Score introduced by Salimans et al. [104], is a metric designed to assess both the quality and diversity of images produced by a generative model. The IS relies on the classification capability of the model. The generated samples should be easily classified by the Inception model with a label distribution similar to the one of the real samples. The IS can be defined as follows:

Definition 3.1.1 (Inception Score).
Let denote $\mathbb{P}(Y|x)$ be the conditional class distribution of an image x given by the

Inception-v3 model and $\mathbb{P}(Y)$ the class distribution in dataset sampled from P. The Inception Score is defined as:

$$\mathrm{IS}(\widehat{P}) = \exp\left(\mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[\mathcal{D}_{\mathrm{KL}}(\mathbb{P}(Y|\boldsymbol{x})\|\mathbb{P}(Y))\right]\right) \tag{3.1}$$

where $\mathcal{D}_{\mathrm{KL}}$ is the Kullback-Leibler divergence.

With H being the entropy function, Equation (3.1) can be reformulated as:

$$\log\left(\mathrm{IS}(\widehat{P})\right) = H\left(\mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[\mathbb{P}(Y|\boldsymbol{x})\right]\right) - \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[H\left(\mathbb{P}(Y|\boldsymbol{x})\right)\right]. \tag{3.2}$$

The score is optimized when it satisfies two principal conditions:

1. If the entropy of $\mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}[\mathbb{P}(Y|\boldsymbol{x})]$ is maximized. This term is a proxy measure of diversity. The label predictions of the generated samples must be uniformly distributed over all possible labels. This indicates that the generative model generates heterogeneous labels, thereby ensuring diversity.
2. If for every $\boldsymbol{x} \sim \widehat{P}$, $H_Y\left(\mathbb{P}(Y|\boldsymbol{x})\right)$ is minimized. In theory, this term is meant to assess the quality. In fact, the entropy of the label distribution for the generated images must be minimal. This suggests that the classification model is highly confident in predicting a singular label per image, which implies that the images are distinct.

However, in practical applications, the Inception Score has revealed several shortcomings [10, 13, 41, 103]:

- The IS is not sensitive to measuring intraclass diversity. In particular, if the model generates only one high-quality image per class, the IS will be high.
- The IS does not directly measure the realism of individual images. If the images are peripherally saturating, noisy, or distorted, and if the classification confidence is high, they are evaluated as realistic.
- The IS is biased toward the classes represented in ImageNet, since the Inception-v3 model is trained on this dataset. For instance, if the goal is to evaluate models that generate faces such as the CelebA dataset, IS will favor models generating faces with glasses, sunglasses, or cowboy hat, since these attributes are ImageNet classes.
- IS is not necessarily optimal when P and $\widehat{P}$ are identical. Since it does not directly compare the generated distribution with the true data distribution, Barratt and Sharma [10] shows that $\mathrm{IS}(P)$ is not always optimal.

For all the mentioned reasons, the IS has been progressively superseded as of 2020 by the Fréchet Inception Score, which is observed to be more correlated with human perception of quality.

Fréchet Inception Distance (FID) The Fréchet Inception Distance (FID) measures the distance between latent feature vectors calculated for real and generated images. The FID is based on the Fréchet distance, that is, the 2-Wasserstein distance in $\mathbb{R}^m$. The FID is based on two main assumptions:

1. The latent representation of samples of P and $\widehat{P}$ is more meaningful than the pixel representation of the image. In practice, using the output of the last pooling layer of Inception-v3, which is in dimension 2048, we denote it $\phi : \mathbb{R}^d \to \mathbb{R}^m$.
2. The latent representations of the samples of P and $\widehat{P}$ are multivariate normal distributions with respective latent mean vectors $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ and their respective latent covariance matrices $\boldsymbol{\Sigma}$ and $\widehat{\boldsymbol{\Sigma}}$.

Considering these assumptions, the FID calculation becomes feasible: the Gaussian assumption allows for a closed-form computation, while the assumption of lower dimensionality ensures that this computation can be performed within a reasonable time. In practice, we take two sets of samples $\{\boldsymbol{x}_1^{\text{real}}, \dots, \boldsymbol{x}_N^{\text{real}}\}$ drawn from both P and $\{\boldsymbol{x}_1^{\text{fake}}, \dots, \boldsymbol{x}_N^{\text{fake}}\}$ drawn from $\widehat{P}$. We consider the empirical mean and covariance of the latent representation of the samples:

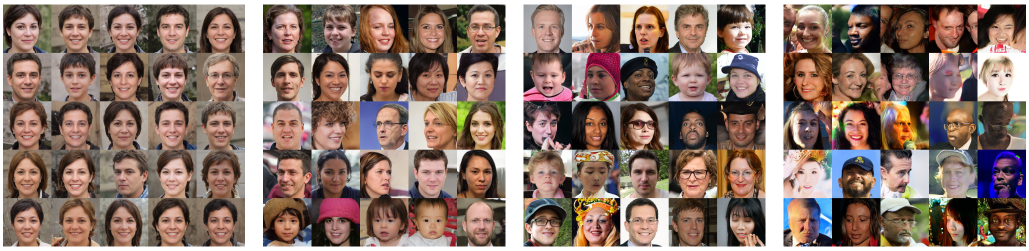
$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}_i^{\text{real}}) \quad \text{and} \quad \boldsymbol{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\phi(\boldsymbol{x}_i^{\text{real}}) - \boldsymbol{\mu}\right)\left(\phi(\boldsymbol{x}_i^{\text{real}}) - \boldsymbol{\mu}\right)^{\top}, \tag{3.3}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}_i^{\text{fake}}) \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\phi(\boldsymbol{x}_i^{\text{fake}}) - \hat{\boldsymbol{\mu}}\right)\left(\phi(\boldsymbol{x}_i^{\text{fake}}) - \hat{\boldsymbol{\mu}}\right)^{\top} \tag{3.4}$$

Definition 3.1.2 (Fréchet Inception Distance)**.**
Let $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\hat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ be the empirical mean and covariance of the latent representation of samples drawn from P and $\widehat{P}$. The Fréchet Inception Distance is defined as:

$$\text{FID}(\widehat{P}, P) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|^2 + \text{Tr}\left(\boldsymbol{\Sigma} + \widehat{\boldsymbol{\Sigma}} - 2(\boldsymbol{\Sigma}\widehat{\boldsymbol{\Sigma}})^{1/2}\right) \tag{3.5}$$

This approach enables us to go beyond the pixel space by comparing distributions in the latent space of an image-trained model. This allows for a comparison that is more in line with human visual perception. Although this method has opened new possibilities for evaluating generative models, it is not without limitations [13, 24, 51]:



(a) FID= 91.7 (b) FID= 16.9 (c) FID= 4.5 (d) FID= 16.7

Fig. 3.1.: Samples from StyleGAN model with different truncation set-up. FID heavily penalizes setup A with high quality and low diversity, and yet setups B and D are rated similarly even if the setup is mode diverse and setup A is more aligned with human perception of performance. Finally, setup D is ranked first despite visual artifacts. Source: Kynkäänniemi et al. [73]

- The FID compares statistical summaries (mean and covariance) of the latent distributions of Inception, a discriminative model. Therefore, it may not capture all aspects of image quality, such as texture and local structure, that are perceptible to humans.
- FID does not distinguish between different types of error in image generation. For example, it treats a noisy object the same as a completely wrong object being generated, which may not align with human judgment.
- Similarly to IS, FID accounts for both quality and diversity, but without a clear trade-off. For example, Kynkäänniemi et al. [73] highlights the limitations of FID with samples drawn from StyleGAN in Figure 3.1.

In summary, the Inception Score and Fréchet Inception Distance have historically served as valuable benchmarks for evaluating generative models. However, as generative models advance and produce increasingly high-quality outputs, the limitations of IS and FID become more pronounced. These metrics do not sufficiently capture quality and diversity as independent dimensions and may not reflect the nuanced performance of state-of-the-art models. Despite this, they are still widely used in certain contexts, such as monitoring model convergence and detecting mode collapse, albeit with a significant computational cost. In fact, for a robust evaluation of generative models, both IS and FID are estimated to require up to 50,000 samples to produce reliable and meaningful scores.

3.2 Precision and Recall for Generative Models

Several methods have been proposed to address the limitations of IS and FID. In this thesis, we focus on Precision and Recall, which have been adapted from the field of binary classification to the field of generative modeling in order to assess quality and diversity independently. Before introducing their adaptation for generative models, let us recall their classic definitions in classification tasks. They are based on the proportions of true positive defined in Table 3.1:

	Predicted Positive (PP)	**Predicted Negative (PN)**
Positive (P)	True Positive (TP)	False Negative (FN)
Negative (N)	False Positive (FP)	True Negative (TN)

Tab. 3.1.: Confusion matrix: Taxonomy of the classification outcomes.

Building on this, precision is the proportion of predicted positive data points correctly classified, and the recall is the proportion of positive data points correctly classified. Formally:

Definition 3.2.1 (Precision and Recall for binary classification)**.**
In a binary classification task, let P be the number of positive instances, PP the number of positive prediction instances, and TP the number of true positive instances. Precision and Recall for binary classification are then defined as:

$$\text{precision} \coloneqq \frac{\mathrm{TP}}{\mathrm{PP}} \quad \textit{and} \quad \text{recall} \coloneqq \frac{\mathrm{TP}}{\mathrm{P}}. \tag{3.6}$$

In generative models, the positive label refers to whether a point can be generated by P and the predicted positive data points are the ones that can be generated by $\widehat{P}$. With this transposition, we can grasp what Precision and Recall stand for in generative modeling:

- **Precision** measures the proportion of generated samples that could be generated by P. High precision in a generative model suggests that the generated samples are of high quality.
- **Recall** evaluates the proportion of samples drawn from P that can be generated by $\widehat{P}$. High recall suggests that most of the data points generated by $\widehat{P}$ are highly diverse.

However, this transposing to generative modeling is not straightforward. The challenge in applying Precision and Recall to generative models lies in defining whether a point can be generated by the distributions P or $\widehat{P}$. Unlike in classification tasks, where the ground-truth labels are known, in generative tasks, there is no explicit label to say whether a point is generated by a distribution. Thus, the different definitions differ in how the notion is transposed to the generative task. In this thesis, we will explore two particular definitions of Precision and Recall: one based only on the support of the distributions and the other based on the density of the distributions.

- **Support-Based Approach:** This mainstream definition considers the support of the distributions. With this definition, a point can be generated by a distribution if it is in its support, i.e. if the density is positive.
- **Density-Based Approach:** This more refined approach considers a range a different classifiers based on the density ratio with varying threshold. It produces Precision-Recall Curves (PR-Curves), similar to the ones in the classification task.

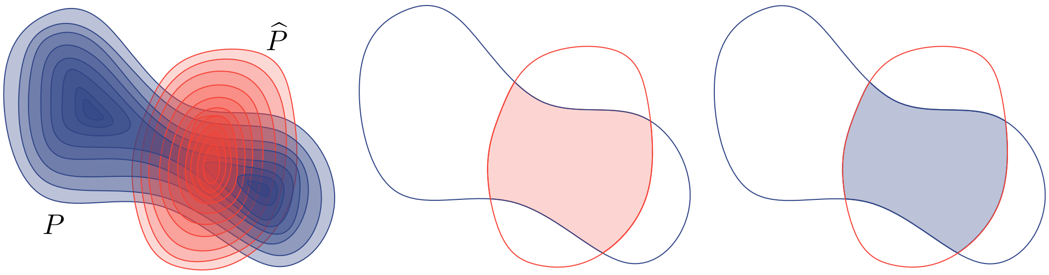
The next sections will review these concepts, describe how they are calculated, and explore their different variations.

3.2.1 The support-based approach: $(\bar{\alpha}, \bar{\beta})$.

In theory: One way to apply Precision and Recall to generative model, and more specifically to distribution comparison, is to consider a binary point of view on the samples: for any given sample $\boldsymbol{x}$, the true label is positive if $\boldsymbol{x} \in \mathrm{Supp}(P)$ and the predicted label is positive if $\boldsymbol{x} \in \mathrm{Supp}(\widehat{P})$. Kynkäänniemi et al. [73] present how quality and diversity can be assessed with the pair of values $(\bar{\alpha}, \bar{\beta})$:

Definition 3.2.2 (Support-Based Precision and Recall [73].).
For any distributions $P \in \mathcal{P}(\mathcal{X})$ and $\widehat{P} \in \mathcal{P}(\mathcal{X})$, we say that the distribution P has precision $\bar{\alpha}$ at recall $\bar{\beta}$ with respect to $\widehat{P}$ if

$$\bar{\alpha} \coloneqq \widehat{P}(\mathrm{Supp}(P)) \quad \textit{and} \quad \bar{\beta} \coloneqq P(\mathrm{Supp}(\widehat{P})). \tag{3.7}$$



(a) Distributions P and $\widehat{P}$. (b) Precision $\bar{\alpha} = \widehat{P}(\mathrm{Supp}(P))$ (c) Recall $\bar{\beta} = P(\mathrm{Supp}(\widehat{P}))$.

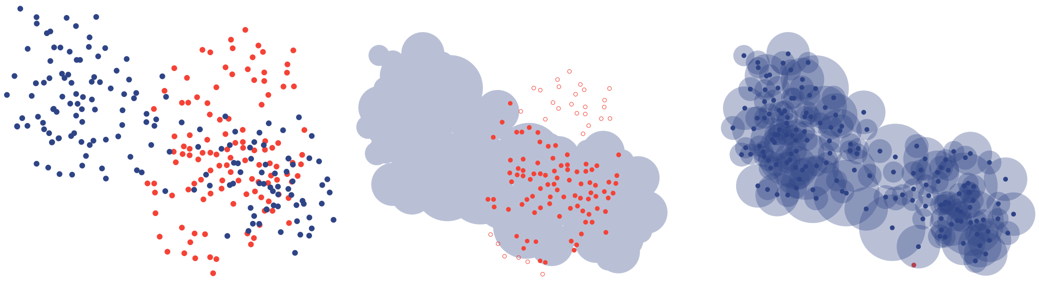
Fig. 3.2.: Example of 2D distributions P and $\widehat{P}$. Only the support of the distributions are considered to compute the Precision and the Recall.

Precision $\bar{\alpha}$ is the proportion of generated data that lies on the support of the real data. Recall $\bar{\beta}$ is the proportion of the support of the real data that is covered by the generated data.

This definition is often referred in the literature as the *Improved Precision and Recall*. The support-based approach simplifies the evaluation of generative models by condensing the complex nature of distribution comparison into two interpretable values. Consider the example given in Figure 3.2, theoretically, the Precision is the proportion of the support of $\widehat{P}$ in the support of P and the Recall is the proportion of the support of P in the support of $\widehat{P}$.

In practice: Computing the exact support of P and $\widehat{P}$ is not feasible in large dimension. Instead, Kynkäänniemi et al. [73] rely on an estimation using a k-nearest neighbors (k-NN) algorithm. Instead of working directly with pixel space, these metrics are derived from latent space representations obtained from an image classification model that captures content-related features. The method uses the VGG network [109], which is, similarly to Inception-v3, a deep convolutional neural network popular for its image recognition capabilities. According to the authors, using VGG, Precision and Recall correlate more than with Inception-v3 with the human perception of diversity and quality.

More specifically, assume that we have N real points $\boldsymbol{x}_1^{\text{real}}, \ldots, \boldsymbol{x}_N^{\text{real}}$ sampled from P and N fake points $\boldsymbol{x}_1^{\text{fake}}, \ldots, \boldsymbol{x}_N^{\text{fake}}$ sampled from $\widehat{P}$. Let ϕ be the embedding function based on VGG, which is also in dimension 2048. Let us define $B_{k,\phi}(\boldsymbol{x}, P)$ the ball centered on $\phi(\boldsymbol{x})$, whose radius is the distance to the k-th nearest neighbor of $\phi(\boldsymbol{x})$ in the set of projections $\phi(\boldsymbol{x}_i^{\text{real}}), \ldots, \phi(\boldsymbol{x}_N^{\text{real}})$. And vice versa with $B_{k,\phi}(\boldsymbol{x}, \widehat{P})$ and the set of projections $\phi(\boldsymbol{x}_i^{\text{fake}}), \ldots, \phi(\boldsymbol{x}_N^{\text{fake}})$. This ball has a variable size, and the



(a) Samples drawn from both distributions P and $\widehat{P}$.

(b) Estimation of the support of the distribution P.

(c) Estimation of the Precision

Fig. 3.3.: Computation of the support-based Precision given samples from P and $\widehat{P}$. Sample of P and used to estimate $\mathrm{Supp}(P)$ and the Precision is the ratio of the number of points in the support vs. the total number of points.

higher the theoretical density, the smaller the ball. The estimation of the support is the union of the ball centered on every point sample from a distribution:

$$\widehat{\text{Supp}}(P) \coloneqq \bigcup_{i=1}^{N} B_{k,\phi}(\boldsymbol{x}_i^{\text{real}}, P) \quad \text{and} \quad \widehat{\text{Supp}}(\widehat{P}) \coloneqq \bigcup_{i=1}^{N} B_{k,\phi}(\boldsymbol{x}_i^{\text{fake}}, \widehat{P}) \tag{3.8}$$

Finally, the Precision is the proportion of the points sampled from $\widehat{P}$ in the estimation support of P among all the points sampled from $\widehat{P}$ and the recall is the proportion of points sampled from P that are in the support of $\widehat{P}$:

$$\bar{\alpha} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{\phi(\boldsymbol{x}_i^{\text{fake}}) \in \widehat{\text{Supp}}(P)\}} \quad \text{and} \quad \bar{\beta} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{\phi(\boldsymbol{x}_i^{\text{real}}) \in \widehat{\text{Supp}}(\widehat{P})\}} \tag{3.9}$$

For the example distribution given in Figure 3.2 we show in Figure 3.3, how the support of P is estimated and how the Precision is computed. To ensure a robust estimate of the support-based Precision and Recall, a sample size ranging from 10,000 to 50,000 is recommended.

Drawbacks: Despite its practical utility, the support-based method has some limitations:

- It is unable to distinguish two distributions that share the same support but have different densities.
- It is sensitive to the quality of the support estimation. In particular, this method can be disproportionately affected by outliers.
- The k-NN algorithm used to estimate the support is not differentiable and computationally expensive, especially in high dimensions.

Although support-based metrics may not be ideal for theoretical analysis of distributions due to their inherent simplicity, they have proven to be practical for comparative evaluations of models. To address their sensitivity to outliers, various refinements have been proposed. For example, Kim et al. [67] introduced a technique to filter isolated data points when estimating support. In a separate development, [85] presented an alternative metric known as Density and Coverage, which is elaborated in Section 3.3.1.

3.2.2 The density-based approach: PR-Curves

Since the method presented in the previous section does not account for the difference between densities, it is not suitable for a theoretical analysis of probability distributions. For example, in Figure 3.4, we give an example of two distributions with very different densities that share the same support. In such cases, the support-based metrics would indicate perfect Precision and Recall, failing to capture the true dissimilarity between the distributions. Sajjadi et al. [103] introduced a more refined metric that incorporates the dissimilarity of densities into the evaluation of generative models. This approach, similar to the precision and recall curves used in classification tasks, allows for a more detailed assessment of how well the generated distribution $\widehat{P}$ matches the target distribution P.

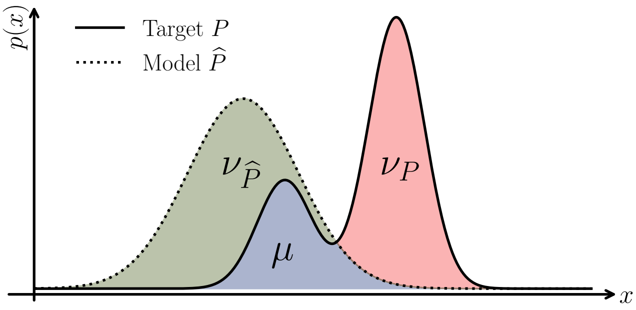


Fig. 3.4.: Representation of μ, ν_P and $\nu_{\widehat{P}}$. μ represent the proportion of P and $\widehat{P}$ that overlap. ν_P accounts for the loss in Recall and $\nu_{\widehat{P}}$ accounts for the loss in Precision.

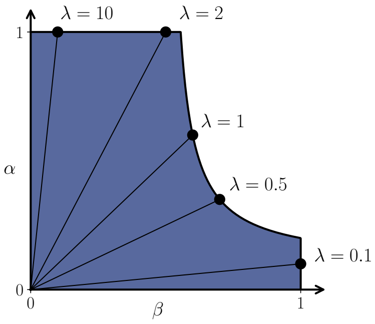


Fig. 3.5.: Representation of a set $\text{PR}(P, \widehat{P})$ and the corresponding PR-Curve $\text{PRD}(P, \widehat{P})$.

In theory: Sajjadi et al. [103] initially proposed a version of this approach for finite state-space distributions only. This was further extended to continuous distributions by Simon et al. [108], allowing a comprehensive analysis that applies to a wider range of generative modeling scenarios. The core idea is to measure how well the generated distribution $\widehat{P}$ can replicate the real distribution P and vice versa, considering every possible distribution μ that overlaps P and $\widehat{P}$, in other words, every possible distribution defined at the intersection of the supports of P and $\widehat{P}$. The Precision and Recall are defined as the proportion of the mass of P and $\widehat{P}$:

Definition 3.2.3 (Precision and Recall - Sajjadi et al. [103]).
Let P and $\widehat{P}$ be two distributions defined on a finite state space $\mathcal{X}$. For $\alpha, \beta \in [0,1]$, the probability distribution $\widehat{P}$ has a Precision α at Recall β w.r.t. P if there exist distributions $\mu \in \mathcal{P}(\text{Supp}(P) \cap \text{Supp}(\widehat{P}))$, $\nu_P \in \mathcal{P}(\text{Supp}(P))$ and $\nu_{\widehat{P}} \in \mathcal{P}(\text{Supp}(\widehat{P}))$ such that

$$P = \beta\mu + (1-\beta)\nu_P \quad \text{and} \quad \widehat{P} = \alpha\mu + (1-\alpha)\nu_{\widehat{P}}$$

The component ν_P denotes the part of P that is not covered by μ, and therefore cannot be generated by $\widehat{P}$. Similarly, $\nu_{\widehat{P}}$ denotes the part of $\widehat{P}$ not covered by μ, and therefore cannot generate P.

For any given distribution μ, the pair (α, β) represents a trade-off between precision and recall for the distributions P and $\widehat{P}$. This balance is influenced by the fact that μ

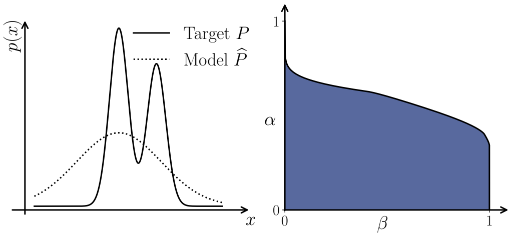


Fig. 3.6.: A model $\widehat{P}$ with high Recall and low Precision with the corresponding PR-Curve.

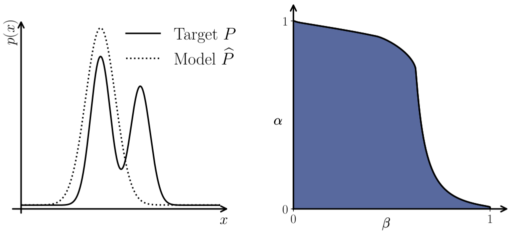


Fig. 3.7.: A model $\widehat{P}$ with high Precision and low Recall with the corresponding PR-Curve.

overlaps with P and $\widehat{P}$. If P and $\widehat{P}$ are not identical, μ will only partially encompass the weights of P and $\widehat{P}$. Consequently, μ might prioritize covering the overlap support of P to achieve higher precision α, or it might focus on encompassing $\widehat{P}$ to achieve greater recall β. Alternatively, if μ is largely different from P and $\widehat{P}$, both α and β will be low. We denote by $\mathrm{PR}(P, \widehat{P}) \subset [0,1]^2$ the set of all possible pairs of Precision Recall. Some sets $\mathrm{PR}(P, \widehat{P})$ are illustrated as the blue area in Figures 3.5, 3.6 and 3.7.

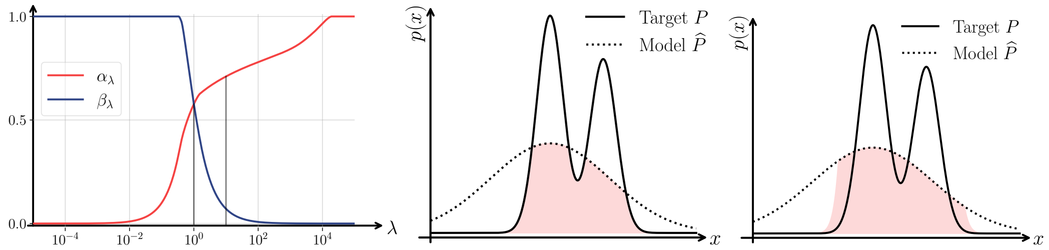
However, setting either α or β at a specific value, it is possible to determine a distribution μ that maximizes the coverage of $\widehat{P}$ or P as much as possible and therefore finds the best trade-off for a given value of α or β. This is the idea behind the PR-Curve, which is the boundary of the set $\mathrm{PR}(P, \widehat{P})$, and is illustrated in Figure 3.5. The PR-Curve denoted as $\mathrm{PRD}(P, \widehat{P}) \subset [0,1]^2$ is the set of all the best possible trade-offs. It has been defined first by Sajjadi et al. [103] for finite state-space distributions and then extended to continuous distributions by Simon et al. [108]. In this manuscript, we will use a reformulation of the PR-Curve that encompasses both the discrete and the continuous case, as proposed by Simon et al. [108]:

Theorem 3.2.4 (PR-Curve - Simon et al. [108]).
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions such that $P, \widehat{P} \ll \mu$. The PR-Curve is the set $\mathrm{PRD}(P, \widehat{P})$ *defined as :*

$$\mathrm{PRD}(P, \widehat{P}) = \{(\alpha_\lambda, \beta_\lambda) \,|\, \lambda \in [0, \infty]\} \tag{3.10}$$

with:

$$\alpha_\lambda = \mathbb{E}_{\widehat{P}}\left[\min\left(\lambda \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right)\right] \quad \textit{and} \quad \beta_\lambda = \mathbb{E}_P\left[\min\left(1, \frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})} \frac{1}{\lambda}\right)\right]. \tag{3.11}$$

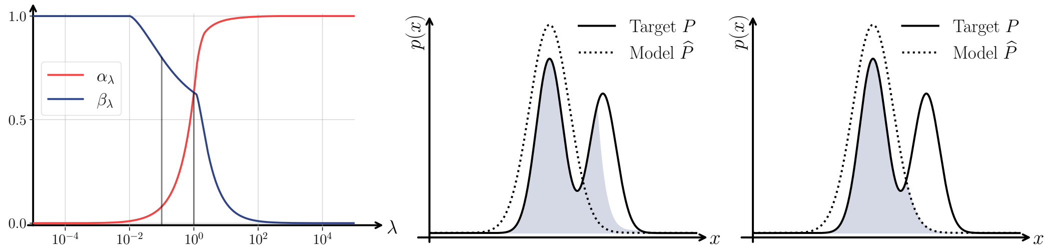


(a) Values of α_λ and β_λ for the distribution in Figures 3.8c and 3.8b

(b) Representation of α_1

(c) Representation of α_{10}

Fig. 3.8.: Example of the values of α_λ and β_λ for the distribution in Figures 3.8c and 3.8b. In Figures 3.8b and 3.8c, the values of α_1 and α_{10} are represented as the ratio of the red area and the area under the dotted curve.



(a) Values of α_λ and β_λ for the distribution in Figures 3.9b and 3.9c

(b) Representation of $\beta_{1/10}$

(c) Representation of β_1

Fig. 3.9.: Example of the values of α_λ and β_λ for the distribution in Figures 3.9b and 3.9c. In Figures 3.9b and 3.9c, the values of $\beta_{1/10}$ and β_1 are represented as the ratio of the blue area and the area under the dotted curve.

The parameter λ in Equation (3.11) is not only a way to parameterize the PR-Curve, but can also be interpreted as a threshold on the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$, similar to the notion of threshold met in classification tasks:

- Consider the Precision α_λ. If the distributions were identical, for every $\lambda \geq 1$, then $\alpha_\lambda = 1$ and for every $\lambda \leq 1$, then $\alpha_\lambda = \lambda$ decreases to 0. If we consider the high values of λ ($\lambda \geq 1$), then all values of $\boldsymbol{x}$ such that $\lambda p(\boldsymbol{x}) < \widehat{p}(\boldsymbol{x})$, contribute to decrease α_λ. Intuitively, λ serves as a threshold on the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$: the Precision α_λ is penalized when the density of $\widehat{P}$ is λ times higher than the density of P. In other words, the Precision for high values of λ is reduced when $\widehat{P}$ overestimates P.
- Consider the Recall β_λ. If the distributions were identical, for every $\lambda \leq 1$, then $\beta_\lambda = 1$ and for every $\lambda \geq 1$, then $\beta_\lambda = 1/\lambda$ decreases to 0. If we consider the low values of λ ($\lambda < 1$), then all values of $\boldsymbol{x}$ such that $\widehat{p}(\boldsymbol{x}) < \lambda p(\boldsymbol{x})$ contribute to decrease β_λ. In contrast to Precision, the Recall β_λ is penalized when the density of P is λ times higher than the density of $\widehat{P}$. In other words, the Recall for low values of λ is reduced whenever $\widehat{P}$ underestimates P.

For example, we can consider two cases: an example with low quality and high diversity in Figure 3.6 and an example with low diversity and high quality in Figure 3.7. We plot how the PR-Curves are build in Figures 3.8a and 3.9a. First, we can note how the Recall in the high diversity example and the Precision in the high quality examples are close to the optimal. Then in the low-quality example, we show how α_λ is computed for two different values of λ in Figures 3.8b and 3.8c. It is the ratio of the red area to the area under the dotted curve. For $\lambda = 10$, we consider that when $\widehat{p}$ is 10 times lower than p, the learned distribution is not penalized; however, if the density is higher than that, it will be considered as an overestimation. In other words, the further λ is from 1, the less picky the evaluation is about considering if the densities are similar. The same reasoning can then be applied to the low-diversity example in Figure 3.9.

It is important to note one point on which the PR-Curves in classification tasks differ from the PR-Curves introduced for generative model evaluation. The threshold in classification is a crucial part of the classifier and each value defines a different classifier, therefore each point on the PR-Curves corresponds to a different model. However, here, the threshold is a crucial part of the *evaluation* of the model and thus the PR-Curve assesses a single model, but the threshold will determine how (1) picky the evaluation is on the similarity of the densities and (2) the importance of quality or diversity. For example, for $\lambda = 1$, the density must be exactly the same for the Precision and Recall to be maximal. On the contrary, for large values of λ, the evaluation is less picky about the similarity of the densities. In particular, for $\lambda = 0$ and $\lambda = \infty$, the evaluation is only based on the support of the distributions and, therefore, does not take into account the densities:

Theorem 3.2.5 (Support-based and PR-Curves)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Then, the support-based Precision and Recall $(\bar{\alpha}, \bar{\beta})$ are related to the PR-Curve values $\mathrm{PRD}(P, \widehat{P})$ for $\lambda = 0$ and $\lambda = \infty$:

$$\bar{\alpha} = \max_{\lambda} \alpha_\lambda = \alpha_\infty \quad \textit{and} \quad \bar{\beta} = \max_{\lambda} \beta_\lambda = \beta_0. \tag{3.12}$$

This Theorem can be proven by using Equation (3.11) and using the traditional convention of measure theory that $0 \times \infty = 0$. It shows that the support-based approach is a particular case of the density-based approach and that we can retrieve the support-based approach metrics by looking and the maximum Precision α_λ and the maximum Recall β_λ.

We can verify on discrete examples in Figure 3.10 how the PR-Curves exhibit several fundamental properties that are desirable for Precision-Recall metrics.

- If the supports of P and $\widehat{P}$ are disjoint, both precision and recall should be minimal. For instance in Figure 3.10 A, the maximum Precision and Recall are both 0.
- If the supports do not fully overlap, either Recall or Precision should be compromised. As illustrated in Figures 3.10 B and E, either Precision or Recall is limited to a maximum of 0.5. In the first case, the maximum Precision α_λ is 0.5, since half of the points sampled from $\widehat{P}$ are in the support of P. In the second case, the maximum Recall β_λ is 0.5, since half of the points sampled from P are in the support of $\widehat{P}$.
- When the supports coincide, but the densities do not, the PR-Curves should reflect this dissimilarity. The support is matching; therefore, the maximum Precision and Recall are both 1. However, in Figure 3.10 C, $\widehat{P}$ generates an excess of points in the second bin, which is indicated by a decrease in Precision for high values of λ. And vice-versa, in Figure 3.10 D.
- Lastly, if both the supports and the densities are identical, the PR-Curves should indicate maximum Precision and Recall, as shown in Figure 3.10 F.

In practice: For the continuous and discrete example distributions, we have computed the PR-Curves by computing the expected values given in Equation (3.11) using the close form of the density ratio. However, in practice the difficulty lies in estimating the density ratio; a task particularly difficult in high dimensions [115]. Consequently, similarly to the FID and the support-based approach, the PR-Curves are not computed using the pixel representation of the images. Both works of Sajjadi et al. [103] and Simon et al. [108] have proposed not only distinct definitions but also different methodologies for computing PR-Curves. However, both are based on the embedding function ϕ built using Inception-v3 introduced for the FID score. The two methods are summarized as follows.

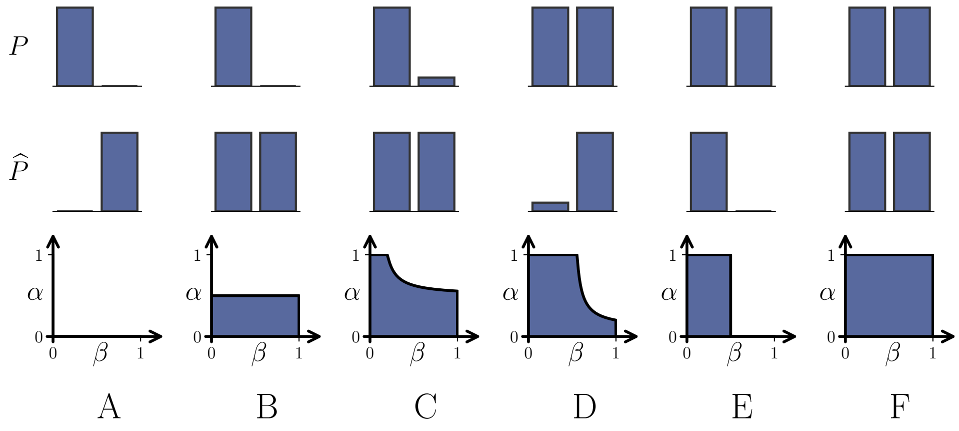
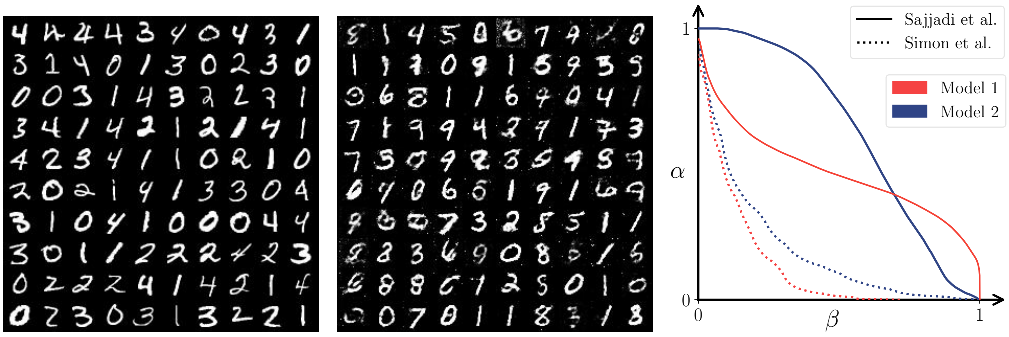


Fig. 3.10.: Example of distributions P and $\widehat{P}$ and their PR-Curve. Inspired from [103].



(a) Model 1: High Precision (b) Model 2: High Recall (c) PR-Curves

Fig. 3.11.: Example of PR-Curves computed with Sajjadi et al. [103] and Simon et al. [108] for two models generating samples of the MNIST dataset: one model generating precise samples and one model generating diverse samples.

- **Sajjadi et al. [103]'s method:** This approach uses a k-means-based algorithm to group the embedding of data points sampled from both P and $\widehat{P}$. Following this, the densities per cluster are the ratio of the number of points in each cluster to the total number of points per distribution. The PR-Curve is then constructed by varying the threshold λ and computing the Precision and Recall for each value of λ.
- **Simon et al. [108]'s method:** This technique involves the use of a discriminator-based model. Precision and Recall are estimated by training an ensemble model h to classify samples from P and samples from $\widehat{P}$ using the cross entropy loss. At optimality, Precision and Recall are calculated by re-weighting the sum of false positive and false negative rates with λ. If we (arbitrarily) assume that $h(\boldsymbol{x}) = 1$ corresponds to a sample from $\widehat{P}$ and $h(\boldsymbol{x}) = 0$ corresponds to a sample from P, then Precision and Recall are given by:

$$\alpha_\lambda = \mathbb{E}_P\left[\mathbb{1}_{\{h(\boldsymbol{x})=1\}}\right] + \lambda\mathbb{E}_{\widehat{P}}\left[\mathbb{1}_{\{h(\boldsymbol{x})=0\}}\right] \quad \text{and} \quad \beta_\lambda = \frac{\alpha_\lambda}{\lambda}. \tag{3.13}$$

  For this method also, the classifier is trained on the latent representation $\phi(\boldsymbol{x})$ of the images sampled from P and $\widehat{P}$.

We can compare both methods on the MNIST dataset in Figure 3.11. The PR-Curves are computed for two models generating samples of the MNIST dataset: one model generating precise samples and one model generating diverse samples.

Drawbacks: Despite, the theoretical advantages of the PR-Curves, these methods have limitations:

- The method relies heavily on density estimation, and every method has its own limitations. For example, the k-means approach fails to capture a large number of packed samples [79]. The classifier-based method drastically depends on

			Model 1		Model 2	
Approach	Authors	Method	P	R	P	R
Support-based	Kynkäänniemi et al. [73]	k-NN	0.54	0.91	0.84	0.70
Density-based	Sajjadi et al. [103]	k-means	0.90	0.64	0.78	0.91
Density-based	Simon et al. [108]	Classifier	0.34	0.56	0.54	0.58

Tab. 3.2.: Metrics α and β to evaluate quality and diversity proposed by Sajjadi et al. [103] and Simon et al. [108] for the models in Figure 3.11.

the classifier training procedure and tends to strongly underestimate the PR-Curves, as observed in Figure 3.11 and fails to distinguish between the two different behavior of the models.

- Although the approach is more refined compared to the support-based method, the interpretation of the results can be more complex and less intuitive. This complexity might make it harder for practitioners to draw clear conclusions about the performance of their models.

To address the complexity of interpreting the results, the authors introduce a method to summarize the PR-Curve into two scalar values, ranging from 0 to 1. This is achieved by generalizing the F_1 score for a given hyperparameter s. The generalized formula is given as:

$$F_s(\lambda) = (1+s^2)\frac{\alpha_\lambda \beta_\lambda}{s^2\alpha_\lambda + \beta_\lambda} \tag{3.14}$$

The proxies for Precision and Recall are then defined as:

$$\alpha = \max_\lambda F_s(\lambda) \quad \text{and} \quad \beta = \max_\lambda F_{\frac{1}{s}}(\lambda) \tag{3.15}$$

In their experiments, Sajjadi et al. [103] estimate based on visual inspection that setting $s = 8$ provides the best estimation for visual quality in terms of precision and recall. In Table 3.2 we detail the different pairs of metrics for the models in Figure 3.11.

Precision and Recall in this thesis

In practical applications, the support-based approach to Precision and Recall, as proposed by Kynkäänniemi et al. [73], is predominantly used within the image generation community. This preference is attributed to its stronger correlation with human visual perception and its straightforward interpretability. However, the density-based approach offers a more refined theoretical evaluation of generative models. In Section 3.4, we will elaborate on the positioning of this thesis toward the various approaches to define and compute the Precision and Recall.

3.3 Alternative Metrics in Generative Model Evaluation

In this section, we review various metrics introduced in recent years to evaluate generative models, moving beyond the early work by Sajjadi et al. [103], Kynkäänniemi et al. [73], and Simon et al. [108]. These foundational studies focused on assessing the quality and diversity of models, but recent developments have built on their ideas, aiming to address specific challenges, generalize methods, or offer new points of view. We start with the Density and Coverage metric of Naeem et al. [85], which aims to improve the support-based approach. Then, we discuss the PR-Curve based on Information Divergence Frontiers by Djolonga et al. [32], which aims to generalize the definition of PR-Curves. Finally, we look at the Precision Recall Cover by Cheema and Urner [21], which aims to bridge the gap between theoretical concepts and practical computation of Precision and Recall.

We have chosen to explore only a few alternative metrics in this section as they are the most relevant to the thesis. However, it is important to note that many other metrics have been proposed in the literature that were built for orthogonal purposes or for specific domains of applications. For example, we can mention the α-Precision, β-Recall and Authenticity introduced by Alaa et al. [2], the Vendi Score introduced by Friedman and Dieng [40], the MAUVE Score by Pillutla et al. [96].

3.3.1 Density and Coverage

The Density and Coverage metrics, introduced by Naeem et al. [85], offer an alternative to the Precision and Recall metrics of Kynkäänniemi et al. [73], the support-based approach, with a focus on addressing some of their limitations. Due to the k-NN estimation of the support-based Precision and Recall are sensitive to outliers, the authors aim to provide a more robust assessment of fidelity and diversity. The metrics do not estimate how the supports of the distributions overlap, but rather how well the generated samples populate the neighborhoods of the real data points. For that reason, they do not fall into the category of Precision/Recall metrics and are called Density and Coverage metrics.

These metrics rely on estimating the neighborhoods of the real data points and therefore also rely on a k-NN algorithm. Similarly to other metrics, the algorithm is applied in the embedding space of the Inception-v3 model defined by ϕ. This approach is similar to the support-based method: we assume that we have N projections $\phi(\boldsymbol{x}_1^{\text{real}}), \ldots, \phi(\boldsymbol{x}_N^{\text{real}})$ and $\phi(\boldsymbol{x}_1^{\text{fake}}), \ldots, \phi(\boldsymbol{x}_N^{\text{fake}})$ of samples drawn from P and $\widehat{P}$. For the k-NN, we use $B_{k,\phi}(\boldsymbol{x}, P)$ the ball centered on $\phi(\boldsymbol{x})$ whose radius is the distance to the k-th nearest neighbor of $\phi(\boldsymbol{x})$ in the set of projections $\phi(\boldsymbol{x}_i^{\mathrm{real}}), \ldots, \phi(\boldsymbol{x}_N^{\mathrm{real}})$. The Density and Coverage are then defined as:

Definition 3.3.1 (Density and Coverage)**.**
Let $\boldsymbol{x}_1^{\mathrm{real}}, \ldots, \boldsymbol{x}_N^{\mathrm{real}}$ and $\boldsymbol{x}_1^{\mathrm{fake}}, \ldots, \boldsymbol{x}_N^{\mathrm{fake}}$ be samples drawn from P and $\widehat{P}$. The Density is defined as:

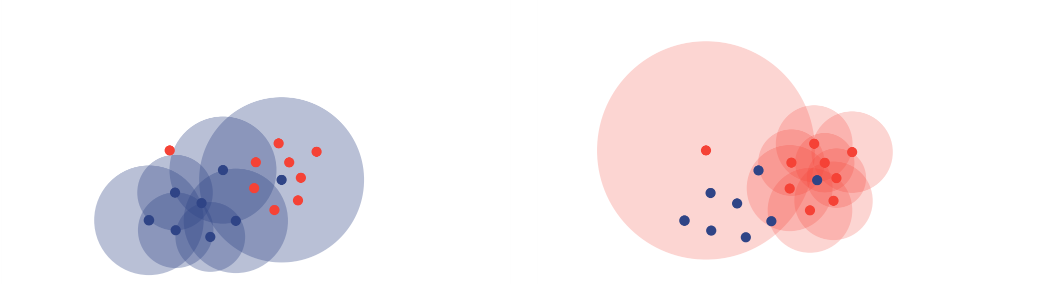
$$Density \coloneqq \frac{1}{kN} \sum_{i=1}^{N} \sum_{i=1}^{N} \mathbb{1}_{\left\{\boldsymbol{x}_j^{\mathrm{fake}} \in B_{k,\phi}(\boldsymbol{x}_i^{\mathrm{real}}, P)\right\}} \tag{3.16}$$

The Coverage is defined as:

$$Coverage \coloneqq \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\left\{\exists j \text{ s.t. } \boldsymbol{x}_j^{\mathrm{fake}} \in B_{k,\phi}(\boldsymbol{x}_i^{\mathrm{real}}, P)\right\}} \tag{3.17}$$

The Density and Coverage are build to independently assess quality and diversity:

- **Density:** This metric is assessing quality. It quantifies how densely the generated samples populate the regions around real data points. Unlike Precision, which only considers whether a generated sample is within the support of the real data, Density evaluates the concentration of generated samples in the neighborhood of real data points. The Density is not bounded to $[0, 1]$: Even if the optimal value is 1, it can be higher or lower than 1 if on average $\widehat{P}$ overestimates or underestimates P.
- **Coverage:** This metric is assessing diversity. It quantifies the ratio of neighborhood of the real data points in which there exists a generated point. As the number of samples evaluated increases, the optimal (and maximum) value of the Coverage is $1 - 1/2^k$.



(a) Neighborhoods of the samples P.

(b) Neighborhoods of the samples $\widehat{P}$.

Fig. 3.12.: Example of samples from P and $\widehat{P}$ with one outlier in each set of samples. The colored balls represent the balls B_k with $k = 3$, i.e. the neighborhoods. The Density is $4/9$ and the Coverage is $3/9$. The Precision is $8/9$ and the Recall is $7/9$.

The Precision and Recall of Kynkäänniemi et al. [73] are based on the pairwise distances of the real samples $\boldsymbol{x}_i^{\text{real}}$ and the pairwise distances of the generated samples $\boldsymbol{x}_i^{\text{fake}}$ for both support estimation. An observation made by Naeem et al. [85] is that datasets contain fewer outliers than the generated samples, and therefore build their metrics on the pairwise distance of the real samples only. By doing so, the Density and Coverage metrics are less sensitive to outliers. For example, consider the scenario depicted in Figure 3.12, where both distributions P and $\widehat{P}$ include an outlier. In Figure 3.12a, we analyze the relationship between Density and Precision. The estimated support of P is represented by the union of colored circles, resulting in a precision of $8/9$ since eight out of nine points of $\widehat{P}$ fall within this support. However, the Density is calculated as $(0+1+1+1+1+1+2+2+3)/(3*9) = 4/9$. This calculation is based on the count of neighborhoods within which each generated point falls. Similarly, in Figure 3.12b, we examine Coverage in comparison to Recall. While the Recall is $7/9$, the Coverage is only $3/9$, indicating that only three points of P have a corresponding point from $\widehat{P}$ within their neighborhood. In these examples, the presence of outliers leads to an overestimation of quality and diversity when using Precision and Recall, whereas the Density and Coverage metrics provide a more nuanced and potentially more accurate assessment.

Drawbacks:

- While Density and Coverage offer a more robust evaluation, their complexity can make them less interpretable compared to the straightforward nature of Precision and Recall.
- Similarly to the support-based approach of Precision and Recall, the effectiveness of these metrics is highly dependent on the choice of the hyperparameter k in the k-NN algorithm.

3.3.2 PR-Curves Using Information Divergence Frontiers

This section introduces the PR-Curves based on Information Divergence Frontiers (IDF), a generalization of the PR-Curve concept for continuous distributions introduced by Simon et al. [108].

In theory: The core idea of this method is to model the intersection of distributions P and $\widehat{P}$ using auxiliary distributions $Q^{\cup}$ and $Q^{\cap}$. In Simon et al. [108], the trade-off is based on comparing the density ratio with a threshold λ. In this work, they generalize the notion of trade-off. The intersection distribution $Q_{\alpha,\pi}$ is defined for the trade-off

$\pi \in [0,1]$ based on the Rényi α-divergence $\mathcal{D}^{\mathrm{R}}_{\alpha}$ introduced in Section 2.1.2. With $\alpha \in [0,\infty]$, the intersection distribution is defined in a way that solves the problem:

$$Q_{\alpha,\pi} = \operatorname*{argmin}_{Q} \pi \hat{D}_{\alpha}(Q\|\widehat{P}) + (1-\pi)\hat{D}_{\alpha}(Q\|P) \tag{3.18}$$

where $\hat{D}_{\alpha} := \frac{1}{\alpha-1}\exp^{(\mathcal{D}^{\mathrm{R}}_{\alpha}/(\alpha-1))}$. The authors show, based on the work of Nielsen and Nock [88], that there exists an explicit form of the density of

$$\forall \boldsymbol{x}, \quad q_{\pi,\alpha}(\boldsymbol{x}) \propto (\pi\widehat{p}(\boldsymbol{x})^{\alpha} + (1-\pi)p(\boldsymbol{x})^{\alpha})^{\frac{1}{\alpha}} \tag{3.19}$$

Finally the generalization of the PR-Curve is defined as:

Definition 3.3.2 (PR-Curve IDF).
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Let $\alpha \in [0,\infty]$ and $\pi \in [0,1]$. The PR-Curve IDF is defined as:

$$\mathrm{PRD}^{\mathrm{IDF}}_{\alpha}(P,\widehat{P}) = \left\{\left(e^{-\mathcal{D}^{\mathrm{R}}_{\alpha}(Q_{\alpha,\pi}\|\widehat{P})}, e^{-\mathcal{D}^{\mathrm{R}}_{\alpha}(Q_{\alpha,\pi}\|P)}\right) \middle| \, \pi \in [0,1]\right\} \tag{3.20}$$

where $\mathcal{D}^{\mathrm{R}}_{\alpha}$ is the Rényi α-divergence, and $Q_{\alpha,\pi}$ is the intersection distribution for different values of α and λ defined in Equation (3.18).

This formulation of PR-Curves IDF aims to be a general form of expressing curves defined on $[0,1]^2$ to evaluate generative models. The curve itself is parameterized by α and the points on this curve are parameterized by π. The authors show that the PR-Curves IDF can express both the density-based approach and the support-based approach. First, by taking $\alpha \to \infty$, the Rényi α-divergence becomes a weak metric $\mathcal{D}^{\mathrm{R}}_{\infty}(P\|Q) = \log\sup_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})/q(\boldsymbol{x})$. Therefore, we have the following.

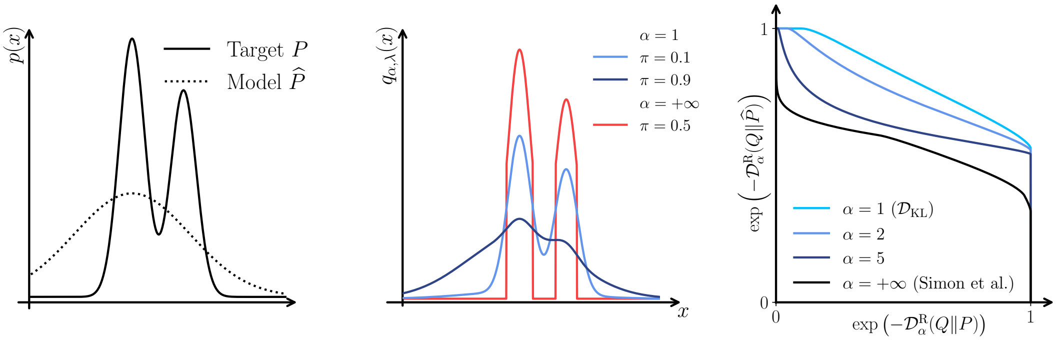


Fig. 3.13.: Illustration of the PR-Curves Using Information Divergence Frontiers for an example with high diversity. From left to right: The target distribution P and the model $\widehat{P}$, the intersection distribution for different values of α and λ, and finally the different PR-Curves. We illustrate the case where $\alpha \to \infty$, equivalent to the PR-Curve defined by Simon et al. [108].

Theorem 3.3.3 (PR-Curves and PR-Curves IDF)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Then, the density-based approach of Precision and Recall, i.e., the PR-Curve PRD *are related to the PR-Curve IDF values* $\mathrm{PRD}^{\mathrm{IDF}}_{\alpha}$ *for* $\alpha \to \infty$*:*

$$\mathrm{PRD}(P, \widehat{P}) = \mathrm{PRD}^{\mathrm{IDF}}_{\infty}. \tag{3.21}$$

The trade-off parameter π *is related to the threshold* λ *as* $\pi = \lambda/(1+\lambda)$.

Therefore, by taking $\alpha \to \infty$ and $\pi = 0$ and $\pi = 1$, one can retrieve the support-based Precision and Recall by using Theorem 3.2.5.

In Figure 3.13, we illustrate the metric for different parameters α. First, we can see that the intersection distribution is smoother with the definition introduced for any Rényi α-divergence than with the one introduced by Simon et al. [108].

Drawbacks: Even if this work has been presented as a theoretical contribution, this metrics has some limitations:

- The PR-Curves IDF is complex to compute and interpret. The framework also includes a union model $Q^{\cup}_{\alpha,\lambda}$ and, therefore, the evaluation process includes two curves. This complexity makes the method less accessible to practitioners.
- The PR-Curves IDF depend on an α-divergence. Minka [83] discuss how these divergences for different α are more biased toward quality or diversity. Therefore, both π and α trade in quality and diversity. The authors suggest using $\alpha = 1$, that is, the $\mathcal{D}_{\mathrm{KL}}$ as the default value, but this choice arbitrarily promotes diversity (as we will discuss in Section 4.1).

3.3.3 Precision-Recall Cover

Precision-Recall Cover , a recent metric introduced in Cheema and Urner [21]. This metric aims to bridge the gap between theoretical concepts and practical computation of Precision and Recall. The goal is to formulate a definition of Precision and Recall closer to practical computation.

In theory: The practical computation of Precision-Recall Cover will rely on a k-NN algorithm. One way to make a connection with k-NN algorithm is to build a metric relying on ball with fixed probability:

Definition 3.3.4 ((a, b) Precision-Recall Cover).
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Let $a, b \in [0, 1]$ such that $a \leq b$. The (a, b)-Precision Coverage of P by $\widehat{P}$ is given by:

$$\mathrm{PC}_{a,b}(P, \widehat{P}) = \mathbb{P}_{\boldsymbol{x} \sim \widehat{P}}[P(B_{\widehat{P}}(\boldsymbol{x}, b)) \geq a] \tag{3.22}$$

where $B_{\widehat{P}}(\boldsymbol{x}, b)$ denotes the ball of probability mass b around the point $\boldsymbol{x}$ with respect to the distribution $\widehat{P}$. Similarly, the (a, b)-Recall Coverage of P by $\widehat{P}$ is:

$$\mathrm{RC}_{a,b}(P, \widehat{P}) = \mathbb{P}_{\boldsymbol{x} \sim P}[\widehat{P}(B_P(\boldsymbol{x}, b)) \geq a] \tag{3.23}$$

where $B_P(\boldsymbol{x}, b)$ denotes the ball of probability mass b around the point $\boldsymbol{x}$ with respect to the distribution P.

In this definition, a and b act as tunable thresholds. The parameter a determines the minimum probability mass required for an area to be considered as "sufficiently covered" by the distribution, while b sets the threshold for an area to be regarded as "negligibly small". This flexibility allows users to adjust the sensitivity of the evaluation measure according to practical computation. Moreover, the Precision-Recall Cover can encompass the support-based approach:

Theorem 3.3.5 (Precision-Recall Cover and Support-Based Approach).
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions. Then for any $\epsilon > 0$, there exist sufficiently small values of a and b such that:

$$|\mathrm{PC}_{a,b}(P, \widehat{P}) - \bar{\alpha}| < \epsilon \quad \textit{and} \quad |\mathrm{RC}_{a,b}(P, \widehat{P}) - \bar{\beta}| < \epsilon. \tag{3.24}$$

Computation: First, similarly to the every other method presented so far, the algorithm is applied in the embedding space of the Inception-v3 model defined by ϕ. The authors propose a method to compute the Precision-Recall Cover using a k-NN algorithm. Although we have typically defined the ball $B_{k,\phi}$ as the ball centered on $\phi(\boldsymbol{x})$ whose radius is the distance to the k-th nearest neighbor of $\phi(\boldsymbol{x})$ in the set of projections $\phi(\boldsymbol{x}_i^{\mathrm{real}}), \ldots, \phi(\boldsymbol{x}_N^{\mathrm{real}})$, here we will denote this ball as $B_P(\boldsymbol{x}, k/N)$. This ball can also be seen as the ball centered on $\phi(\boldsymbol{x})$, which has an empirical probability mass k/N. Consequently, if we define the empirical probabilities of a set $B \subset \mathcal{X}$ as

$$P_N(B) = \frac{1}{N} \sum_i^N \mathbb{1}_{\{\phi(x_i^{\mathrm{real}}) \in B\}} \quad \text{and} \quad \widehat{P}_N(B) = \frac{1}{N} \sum_i^N \mathbb{1}_{\{\phi(x_i^{\mathrm{fake}}) \in B\}}, \tag{3.25}$$

then we can define the empirical Precision-Recall Cover as:

Definition 3.3.6 ((k,k') Empirical Precision-Recall Cover)**.**
The empirical Precision-Recall Cover of P by $\widehat{P}$. Let $k,k' \in \mathbb{N}$ such that $k \leq k'$. The (k,k')-Precision Coverage of P by $\widehat{P}$ is given by:

$$\mathrm{PC}_{k,k'}(P,\widehat{P}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{P_N(B_{\widehat{P}}(\boldsymbol{x}_i^{\mathrm{fake}}, k'/N)) \geq k/N\}}. \tag{3.26}$$

Similarly, the (k,k')-Recall Coverage of P by $\widehat{P}$ is:

$$\mathrm{RC}_{k,k'}(P,\widehat{P}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{\widehat{P}_N(B_P(\boldsymbol{x}_i^{\mathrm{real}}, k'/N)) \geq k/N\}}. \tag{3.27}$$

Empirical Precision-recall cover is the practical way to compute the Precision-Recall Cover. The authors provide, to some extent, a theoretical guarantee that the empirical Precision-Recall Cover is close to the theoretical Precision-Recall Cover.

Drawbacks: This metric being very recent, it has not yet been widely used and drawbacks are not well known.

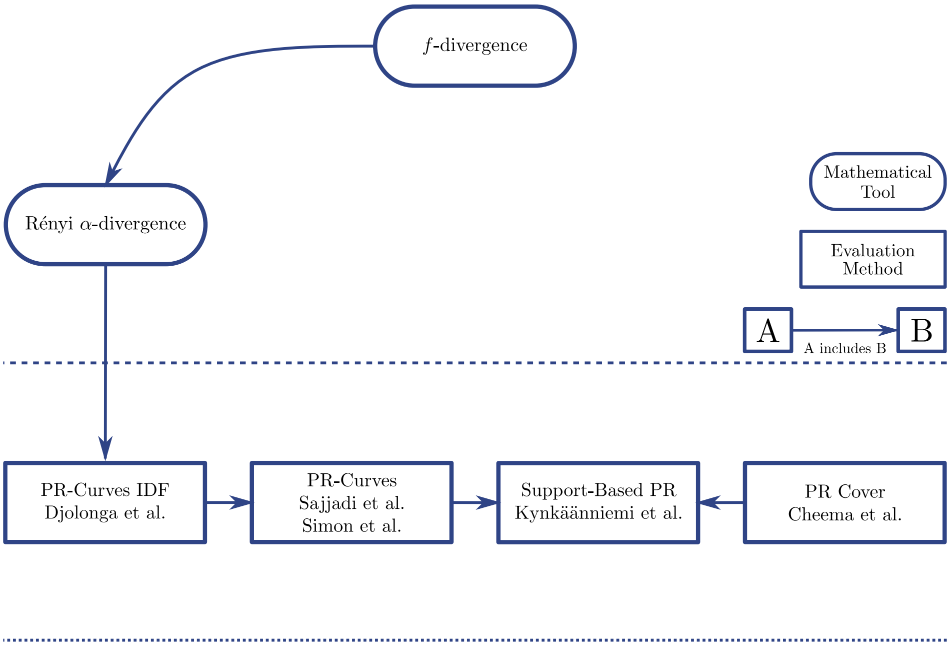
3.4 Linking the different measures

As we mentioned in the different sections, the different metrics are built on different theoretical concepts and have different practical computations. However, the different metrics are not independent and can be linked together. In this section, we recall the different links between the metrics presented in the previous sections.

All interactions are summarized in Figure 3.14. We differentiate the evaluation methods and the theoretical tool on which the metrics are based. We specify the relationship between the different metrics as a relation $A \to B$ if the metric B can be written in terms of the metric A. Therefore, it includes the case where the metric A is a generalization of the metric B, if the metric B is a limit case of A or if the metric B can be computed from the metric A.

We will recall all results depicted in the figure:

- **PR-Curves IDF and PR-Curves**: The PR-Curves IDF by Djolonga et al. [32] is a generalization of the PR-Curves by Sajjadi et al. [103] and Simon et al. [108]. The PR-Curves IDF computes the Rényi α-divergence. When $\alpha \to \infty$,



f-divergence
Rényi α-divergence
Mathematical Tool
Evaluation Method
A
B
A includes B
PR-Curves IDF Djolonga et al.
PR-Curves Sajjadi et al. Simon et al.
Support-Based PR Kynkäänniemi et al.
PR Cover Cheema et al.

Fig. 3.14.: Different Links between definition existing in the literature. $A \to B$ means that the metric B can be written in terms of the metric A.

the PR-Curves IDF is equivalent to the PR-Curves by Simon et al. [108] as expressed in Theorem 3.3.3:

$$\mathrm{PRD}(P, \widehat{P}) = \mathrm{PRD}^{\mathrm{IDF}}_{\infty}. \tag{3.28}$$

- **PR-Curves and Support Based PR:** The Theorem 3.2.5 shows that the Precision and Recall Kynkäänniemi et al. [73] are a special case of the PR-Curves by Sajjadi et al. [103] and Simon et al. [108] for $\lambda = 0$ and $\lambda = +\infty$. In other words, they represent the maximum values of the PR-Curves:

$$\bar{\alpha} = \max_{\lambda} \alpha_{\lambda} \quad \text{and} \quad \bar{\beta} = \max_{\lambda} \beta_{\lambda}. \tag{3.29}$$

- **Precision-Recall Cover and Support Based PR**: The Theorem 3.3.5 states that for sufficiently small thresholds a and b, the Precision-Recall Cover approximates by the support-based Precision and Recall:

$$\mathrm{PC}_{a,b}(P, \widehat{P}) \approx \widehat{P}(\mathrm{Supp}(P)) \quad \text{and} \quad \mathrm{RC}_{a,b}(P, \widehat{P}) \approx P(\mathrm{Supp}(\widehat{P})). \tag{3.30}$$

Note how the Density and Coverage metrics by Naeem et al. [85] are independent of the other metrics. This is because they are based on a different theoretical framework. In practice, they correlate well with the support-based approach, but they are not directly linked to it.

Our point of view on Prior Art

In our experimental analysis in Chapter 5 and Chapter 6, we will use metrics that are commonly accepted and used within the research community to evaluate our models. These include the FID and the support-based approach of Precision and Recall, but also incorporate the Density and Coverage metrics. These metrics are now required by the community in image generation for a comprehensive comparison of generative models.

However, in this thesis, we also add theoretical results to the existing spectrum of precision-recall metrics and leverage these metrics to tune and improve generative models.

PR-Curves and Precision-Recall Cover: We will bridge the gap between the PR-Curves and the Precision-Recall Cover by showing that the PR-Curves can be computed from the Precision-Recall Cover, extending the results of Cheema and Urner [21].

Integration of PR-Curves, f-divergences: Another contribution is the introduction of PR-divergence to directly include PR-Curves in the framework of f-divergences framework. We also show that any f-divergence can be written as a PR-Divergences. This connection provides a more unified understanding of model evaluation and model training.

Tractable Measures for Model Training Building on the theoretical foundations between f-divergences and PR-Curves, we have developed methods to make these measures more tractable in practical model training scenarios. Our approach focuses on optimizing precision-recall trade-offs, enabling the development of models that effectively balance these crucial aspects.

Applying PR-Curves to Rejection Sampling in Generative Models In this part of our work, we focus on applying the theoretical framework of PR-Curves to improve rejection sampling, a key technique to improve generative modeling. This practical application of our theoretical findings helps to refine the sampling process, leading to more effective generative models. Our approach demonstrates the value of theoretical insights in advancing the field of generative modeling.

4 Precision and Recall as an f-divergence

> *Science is what we understand well enough to explain to a computer. Art is everything else we do.*
>
> — **Donald Knuth**
> (1974 Turing Price Laureate
> and TEX Creator)

Contents

Question 1: *How can we unify the definitions of precision and recall for generative models?*

Question 2: *What Precision and Recall can be achieved with neural networks with bounded Lipschitz constants?*

Now that we have introduced both f-divergences to train models and Precision-Recall metrics to assess, we can now explore how these concepts interrelate. In

Section 4.1, we will elaborate on the relationships between f-divergences and the notions of quality and diversity. Consequently, in Section 4.2, we will write the PR-Curves as a family of f-divergences, termed the Precision-Recall divergence, denoted PR. In Section 4.3, we will show how the PR-Divergence connects with existing metrics such as f-divergences and Precision-Recall Cover. In doing so, we will demonstrate how PR-Divergence is a central tool that bridges the gap between Precision Recall metrics and f-divergences, addressing Question 1.

Furthermore, we will leverage the Precision-Recall divergence to quantify the fundamental limits of neural networks regarding quality and diversity. Section 4.4 is dedicated to addressing Question 2, exploring how the PR-Divergence is influenced by the Lipschitz constraints of the neural network.

Contributions: Several contributions are presented in this chapter:

- We introduce the Precision-Recall divergence, a family of f-divergences and show how it can be used to fully understand the connection between f-divergences and Precision/Recall metrics. This result has been published at the conference:
  - *Alexandre Verine et al. "Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows". en. In:* Advances in Neural Information Processing Systems *36 (Dec. 2023), pp. 32539–32573.*
- We show that there exists a relationship between PR-Curves and the PR-Cover. This work is still unpublished.
- We show how the Lipschitz constraint of the generator impacts the Precision and Recall. This is the generalization of results that were proven for the Total Variation only and published at the conference:
  - *Alexandre Verine et al. "On the expressivity of bi-Lipschitz normalizing flows". en. In:* Proceedings of The 14th Asian Conference on Machine Learning. *ISSN: 2640-3498. PMLR, Apr. 2023, pp. 1054–1069.*

4.1 f-Divergences: quality and diversity insights

First, remark that if the mapping function G had unlimited expressivity, then for any f-divergence, the optimal distribution $\widehat{P}$ minimizing $\mathcal{D}_f(P\|\widehat{P})$ would be identical to P. However, in practice, we observe that $\widehat{P}$ differs from P, and the resulting distribution $\widehat{P}$ at convergence is significantly influenced by the choice of the f-divergence. For example, Minka [83] noted that optimizing Kullback-Leibler divergence tends to favor *mass-covering* models and that optimizing the reverse KL and Jensen-Shannon tends to favor *mode-seeking* behaviors.

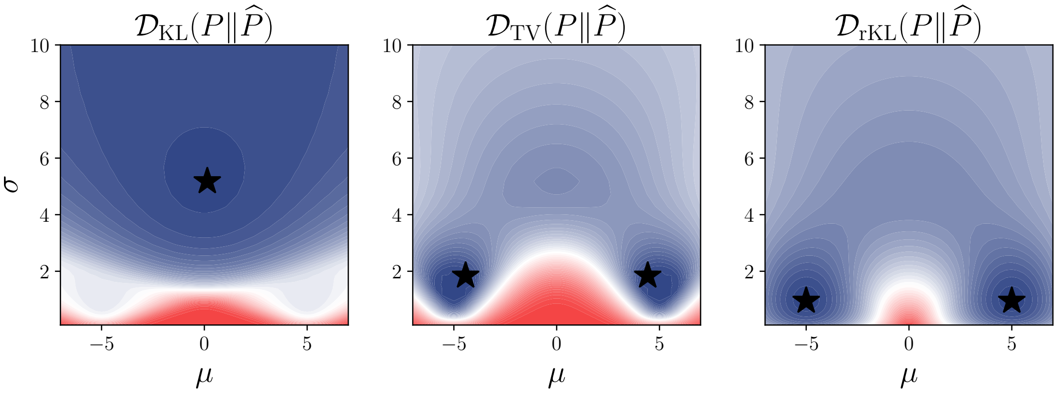


Fig. 4.1.: $\mathcal{D}_f$ between the target distribution P in Figure 4.2 and a distribution $\widehat{P} = \mathcal{N}(\mu, \sigma^2)$. The minimum is represented by ★.

To illustrate this, we present an example using a simple model in Figure 4.1. In this example, we fit a single Gaussian $\mathcal{N}(\mu, \sigma^2)$ to a Gaussian mixture using different f-divergences. The figure plots these divergences in the parameter space as functions of μ and σ. As illustrated in Figure 4.2, some divergences result in a mode-seeking distribution, with a low variance centered on a mode, promoting diversity of samples. Others exhibit a mass-covering effect with a large variance centered between the two modes.

The observed behaviors of different f-divergences can be better understood by examining the function f that defines these divergences. An f-divergence is an expected value of a function of the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$ with respect to $\widehat{P}$:

$$\mathcal{D}_f(P\|\widehat{P}) = \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[\widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right].$$

Consequently, as shown in Figure 4.3, f-divergences penalize likelihood ratios differently. For example, in the Kullback-Leibler divergence, the function f_{KL} imposes a significant penalty on likelihood ratios exceeding 1, that is, scenarios where $\widehat{P}$

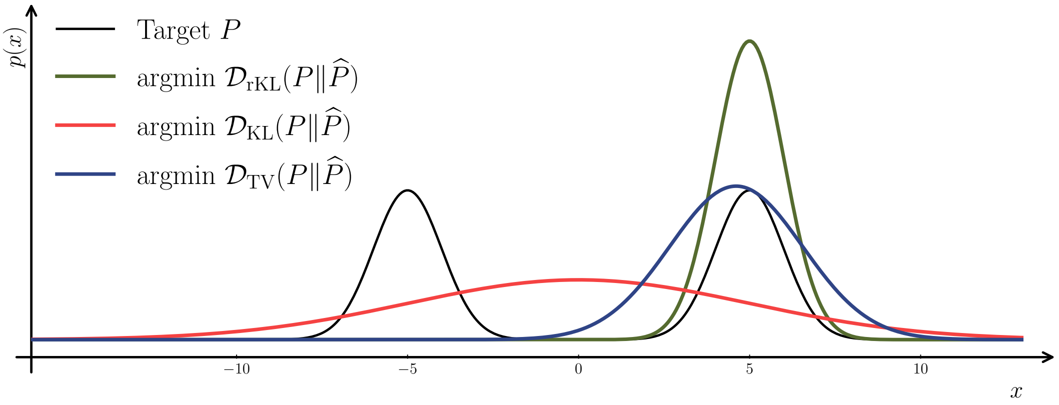


Fig. 4.2.: Mode seeking vs. mass covering effects in different f-divergences

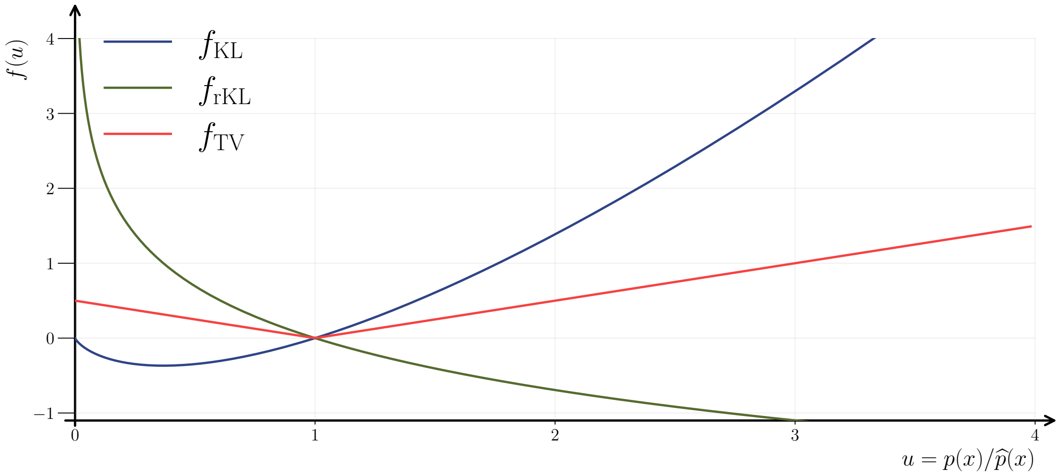


Fig. 4.3.: Analysis of the function f in f-divergences

underestimates P. This characteristic drives $\widehat{P}$ to prioritize covering all modes of P to minimize the instances where $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) > 1$ and $\widehat{p}(\boldsymbol{x}) > 0$. In contrast, the reverse Kullback-Leibler divergence penalizes lower likelihood ratios, leading to a minimization strategy where $\widehat{P}$ aims to reduce the number of points where $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) > 1$ and $\widehat{p}(\boldsymbol{x}) > 0$.

The Total Variation divergence, being symmetric divergence, usually induces a more balanced behavior. In this specific scenario, it leads to a mode-seeking distribution, but a slight adjustment in the gap between modes can shift it towards a mass-covering distribution. In conclusion, these observations motivates a more concrete dive into understand the exact connection between Precision and Recall, especially PR-Curves, with f-divergences.

4.2 The Precision-Recall Divergence

In this section, we introduce a novel f-divergence, called Precision-Recall (PR) Divergence and denoted $\mathcal{D}_{\lambda\text{-PR}}$. First, we will define the f-divergence and then, along its main properties, we will clarify its connection to PR-Curves defined in Section 3.2.1. In 4.3, using the PR-Divergence, we show the link between PR-Curves and traditional f-divergences and Precision-Recall Cover defined in Section 3.3.

4.2.1 Definition & Properties

f-divergences are fully characterized by their generator function f. Thus, we start by introducing the parameterized generator function f_λ as follows:

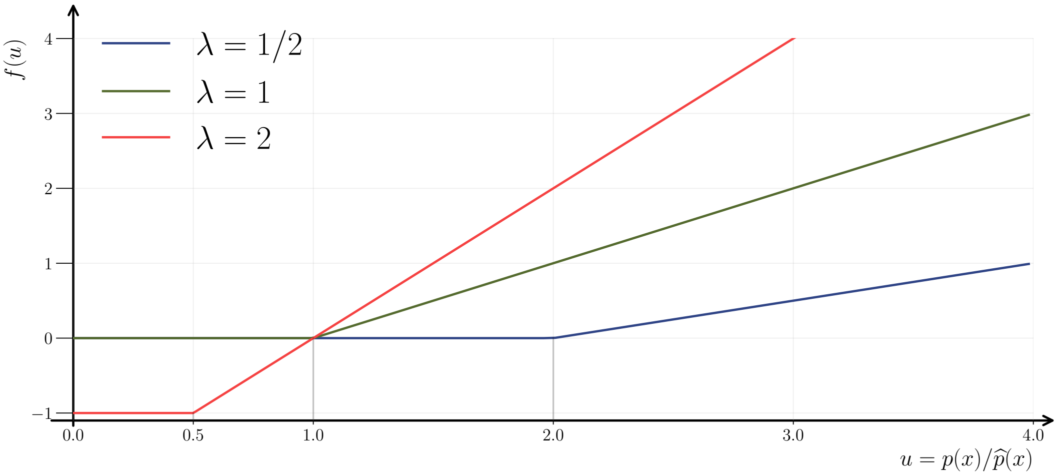


Fig. 4.4.: Graphical representation of the generator function f_λ for various values of the parameter λ.

Definition 4.2.1 (PR-Divergence generator function f_λ).
Given a trade-off parameter $\lambda \in [0, +\infty]$, we define the generator function $f_\lambda : [0, +\infty] \to]-\infty, +\infty]$ given by

$$f_\lambda(u) = \begin{cases} \max(\lambda u, 1) - \max(\lambda, 1) & \textit{for } \lambda \in [0, +\infty[, \\ \mathbb{1}_{\{u=0\}} & \textit{for } \lambda = +\infty. \end{cases} \tag{4.1}$$

In Figure 4.4, the function f_λ is illustrated for different values of λ. Based on this generator function, we can define a family of divergences: the Precision-Recall Divergences denoted *PR-Divergence* indexed by the trade-off parameter. For every $\lambda \in [0, +\infty]$, we can show that the induced PR-Divergence is an f-divergence:

Proposition 4.2.2 (PR-Divergence).
For any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$, then for any $\lambda \in [0, +\infty]$ the PR-Divergence defined as

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{4.2}$$

belongs to the class of f-divergences.

Proof. The proof is detailed in Appendix B.1.

The PR-Divergence as an f-divergence also enjoys the same properties as other f-divergences. In particular, it can also be expressed as a dual variational form with a specific optimal discriminator. The following proposition gives some properties of the PR-Divergence.

Proposition 4.2.3 (Properties of the PR-Divergence).
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, then the following assertions hold.

- *The Fenchel conjugate f_λ^* of f_λ is defined on $\operatorname{dom}\left(f_\lambda^*\right) = [0, \lambda]$ and given by:*

$$f_\lambda^*(t) = \begin{cases} t/\lambda & \textit{for } \lambda \leq 1, \\ t/\lambda + \lambda - 1 & \textit{otherwise.} \end{cases} \tag{4.3}$$

- *The optimal discriminator for the dual variational form is:*

$$T^{\text{opt}}(\boldsymbol{x}) = \lambda \text{sign}\left(\lambda \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - 1\right). \tag{4.4}$$

- *The reverse divergence is:*

$$\mathcal{D}_{\lambda\text{-PR}}(\widehat{P} \| P) = \lambda \mathcal{D}_{\frac{1}{\lambda}\text{-PR}}(P \| \widehat{P}). \tag{4.5}$$

- *For $\lambda = 1$, we have:*

$$\mathcal{D}_{1\text{-PR}}(P \| \widehat{P}) = \mathcal{D}_{\text{TV}}(P \| \widehat{P})/2. \tag{4.6}$$

Proof. The proof is detailed in Appendix B.1.

The PR-Divergence can be seen as a generalization of the Total Variation divergence. In particular, for $\lambda = 1$, the PR-Divergence is half the Total Variation divergence.

4.2.2 PR-Curves and PR-Divergence

We can now show how the PR-curves and the PR-Divergence are related. In fact, we can show that every point in the PR-Curve, parameterized by $\lambda \in [0, \infty]$ can be expressed as a function of the PR-Divergence for the exact same λ.

Theorem 4.2.4 (PR-Curves as a function of $\mathcal{D}_{\lambda\text{-PR}}$).
Given $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, the PR-Curve ∂PRD is related to the PR-Divergence $\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P})$ as follows.

$$\alpha_\lambda(P \| \widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}). \tag{4.7}$$

$$\beta_\lambda(P \| \widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(\widehat{P} \| P). \tag{4.8}$$

Proof. The proof is detailed in Appendix B.1.

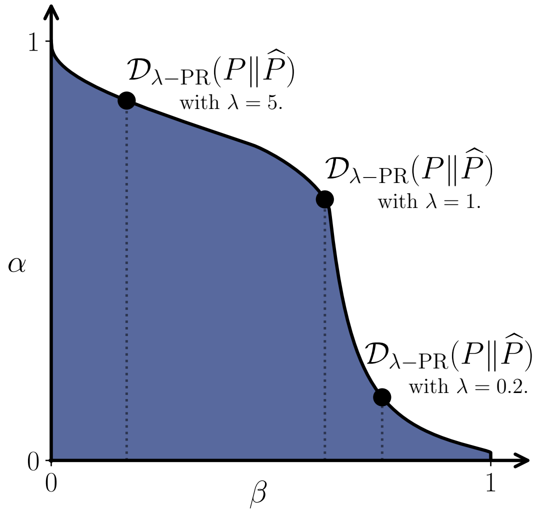


Fig. 4.5.: Illustration of Theorem 4.2.4. Every point of the PR-Curve correspond to a specific $\mathcal{D}_{\lambda\text{-PR}}$, and in particular, minimizing this f-divergence in equivalent to maximizing the abscissa α_λ.

A direct consequence of Theorem 4.2.4 is that, as illustrated in Figure 4.5, every point on the PR-Curve correspond to a specific PR-Divergence, and thus minimizing $\mathcal{D}_{\lambda\text{-PR}}$ is equivalent to maximizing α_λ:

$$\underset{\widehat{P}\in\mathcal{P}(\mathcal{X})}{\operatorname{argmin}}\, \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) = \underset{\widehat{P}\in\mathcal{P}(\mathcal{X})}{\operatorname{argmax}}\, \alpha_\lambda(P\|\widehat{P}). \tag{4.9}$$

This makes $\mathcal{D}_{\lambda\text{-PR}}$ a uniquely suitable candidate for training a generative model with a specific Precision and Recall trade-off.

For example, by taking the example models introduced in Section 4.1, we can see that the PR-Divergence can be used to optimize models to maximize any Precision Recall trade-off $(\alpha_\lambda, \beta_\lambda)$. We can observe that the model that minimizes the PR-Divergence $\lambda = 0.1$ in Figure 4.6 has large variance. As a matter of fact, any model

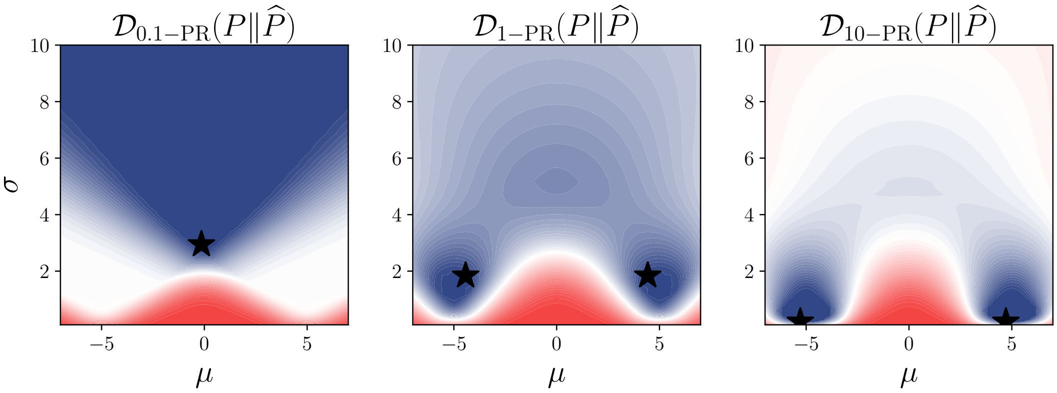


Fig. 4.6.: PR-Divergence between the target distribution P in Figure 4.7 and a distribution $\widehat{P} = \mathcal{N}(\mu, \sigma^2)$. The minimum is represented by ★.

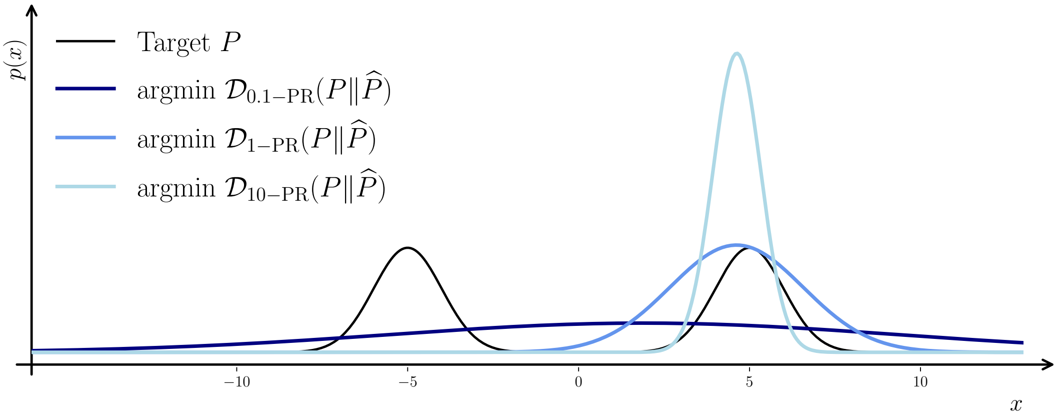
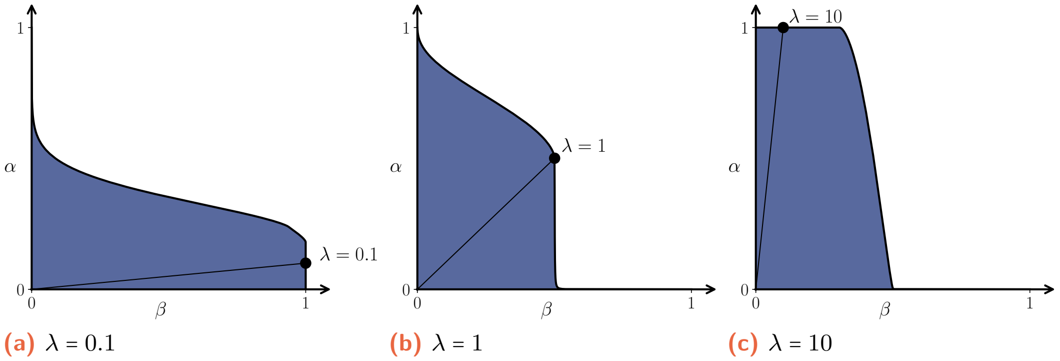


Fig. 4.7.: Models optimizing the PR-Divergence illustrated in Figure 4.6. The corresponding PR-Curves are represented in Figure 4.8.

in the triangular blue zone in Figure 4.7 minimizes the PR-Divergence for $\lambda = 0.1$. It leads to a mass covering behavior, and thus any distribution with high variance centered between the two modes is also an optimal solution. In the corresponding PR-Curve in Figure 4.8, we can see that this model maximizes the value of β_λ up to 1 (and therefore α_λ since $\alpha_\lambda = \lambda\beta_\lambda$). For $\lambda = 1$, the behavior is more complex: Given the (low) expressivity of the model $\widehat{P}$, this solution is highly dependent on the distribution P and here leads to a mode-seeking distribution. Note that this is the same distribution as the one minimizing the Total Variation. Finally, for $\lambda = 10$, the model minimizes the PR-divergence for $\lambda = 10$ and thus is in one of the two mode-seeking zones.

This chapter focuses on theoretically bridging the gap between f-divergences and the notions of Precision and Recall for generative models. However, we will see in Chapter 5 how these PR-Divergences can be used in practice to train complex generative models in higher dimensions.



(a) $\lambda = 0.1$

(b) $\lambda = 1$

(c) $\lambda = 10$

Fig. 4.8.: PR-Curves for different values of λ in the examples illustrated in Figure 4.7

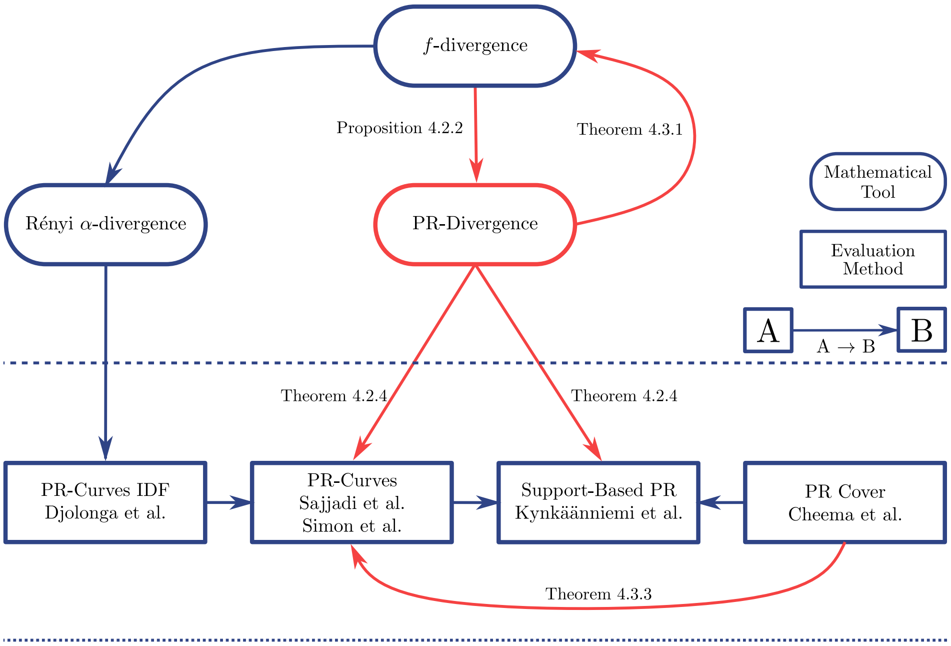


Fig. 4.9.: Contributions to the links between Quality/Diversity Metrics. $A \rightarrow B$ means that the metric B can be written in terms of the metric A.

4.3 Relation with other metrics

We have shown that we could write a PR-Curve as a set of f-divergences, a first step between bridging the gap between PR-Curves and f-divergences. In this section, we will (1) write *any* f-divergences as a weighted average of PR-Divergence and (2) connect the definition of PR-Curves and the Precision-Recall Cover introduced by Cheema and Urner [21] defined in Section 3.3. The different contributions are summarized in Figure 4.9.

4.3.1 Relation with f-divergences

We have seen in the Section 4.1, that f-divergences are observed to have different behavior pushing the optimal distribution to be mode-seeking or mass-covering at convergence. For the Total Variation, the trade-off is straightforward and balanced since minimizing $\mathcal{D}_{\mathrm{TV}}$ is equivalent to minimizing $\mathcal{D}_{1\text{-PR}}$. However, for all the other f-divergences the trade-off between Precision and Recall that is minimized remains unclear.

In this section, we show the trade-offs to which minimizing an f-divergence corresponds. In particular, we show that every f-divergence, under mild conditions, can be written as a weighted average of Precision-Recall Divergence. Note that an equivalent result has been simultaneously discovered in Siry et al. [110].

We first show the link with any f-divergence in Theorem 4.3.1 and then discuss the cases of the $\mathcal{D}_{\text{KL}}$, the $\mathcal{D}_{\text{rKL}}$ and the $\mathcal{D}_{\text{JS}}$ in Corollary 4.3.2.

Theorem 4.3.1 (f-divergence as weighted sums of PR-Divergences)**.**
For any $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$. If the generator function f is twice differentiable, then:

$$\mathcal{D}_f(P \| \widehat{P}) = \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) \mathrm{d}\lambda. \tag{4.10}$$

Proof. Let $c : [-\infty, +\infty] \mapsto \mathbb{R}$ be a $\mathcal{C}^2$ function. The goal is to express any $f(u)$ for all $u \in \mathbb{R}$ as a weighted average of $f_\lambda^{\text{PR}}(u)$ over $\lambda \in [0, +\infty[$. In other words, we need to write $f(u)$ for all $u \in \mathbb{R}$ as:

$$\int_0^{+\infty} c''(\lambda) f_\lambda(u) \mathrm{d}\lambda = \int_0^\infty c''(\lambda) \left[\max(\lambda u, 1) - \max(\lambda, 1)\right] \mathrm{d}\lambda \tag{4.11}$$

Decomposing the integral using integration by part, fully detailed in Section B.1, we can show that the function c must satisfy the following:

$$\forall u \in [0, +\infty[, \quad f(u) = uc\left(\frac{1}{u}\right) - c(1). \tag{4.12}$$

Differentiating with respect to u, we have:

$$f'(u) = c\left(\frac{1}{u}\right) - \frac{1}{u} c'\left(\frac{1}{u}\right) \quad \text{and} \quad f''(u) = \frac{1}{u^3} c''\left(\frac{1}{u}\right). \tag{4.13}$$

Consequently, with $\lambda = 1/u$, we have the following.

$$c''(\lambda) = \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right). \tag{4.14}$$

With such a result we can write any f-divergence as:

$$\begin{aligned}
\mathcal{D}_f(P \| \widehat{P}) &= \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}) \\
&= \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) f_\lambda^{\text{PR}}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\lambda \mathrm{d}\mu(\boldsymbol{x}) \\
&= \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \left[\int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda^{\text{PR}}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x})\right] \mathrm{d}\lambda \\
&= \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) \mathrm{d}\lambda,
\end{aligned}$$

which concludes the proof.

As a sanity check, observe that the weights $f''(1/\lambda)/\lambda^3$ remain invariant under an affine transformation in f much like $\mathcal{D}_f$ (see Section 2.1.2). Using this expression of f-divergences, we can estimate how much weight an f-divergence put on every trade-off between Precision and Recall. By combining Theorem 4.3.1 with Theorem 4.2.4, we find the implicit relationship that captures the Precision/Recall trade-offs made by minimizing any arbitrary f-divergences:

$$\underset{\widehat{P}\in\mathcal{P}(\mathcal{X})}{\operatorname{argmin}} \mathcal{D}_f(P\|\widehat{P}) = \underset{\widehat{P}\in\mathcal{P}(\mathcal{X})}{\operatorname{argmin}} \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \alpha_\lambda(P\|\widehat{P})\mathrm{d}\lambda \tag{4.15}$$

In particular, since Normalizing Flows and GANs are respectively minimizing the Kullback-Leibler Divergence and the Jensen-Shannon Divergence, we can explain their respective behavior in terms of quality and diversity by computing the weights attributed the different trade-offs. We will also compute the weights for the reverse Kullback-Leibler to better contextualize this result.

Corollary 4.3.2 ($\mathcal{D}_{\mathrm{KL}}$, $\mathcal{D}_{\mathrm{JS}}$ and $\mathcal{D}_{\mathrm{rKL}}$ as an average of $\mathcal{D}_{\lambda\text{-PR}}$).
The Kullback-Leibler, the Jensen-Shannon and the reverse Kullback-Leibler divergences can be written as a weighted average of PR-Divergence $\mathcal{D}_{\lambda\text{-PR}}$*:*

$$\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda^2}\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})\mathrm{d}\lambda, \tag{4.16}$$

$$\mathcal{D}_{\mathrm{JS}}(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda(\lambda+1)}\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})\mathrm{d}\lambda, \tag{4.17}$$

$$\textit{and} \quad \mathcal{D}_{\mathrm{rKL}}(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda}\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})\mathrm{d}\lambda. \tag{4.18}$$

Proof. In particular for $\mathcal{D}_{\mathrm{KL}}$, $f_{\mathrm{KL}}(u) = u\log u$, therefore $f''_{\mathrm{KL}}(u) = 1/u$ that gives:

$$\mathcal{D}_{\mathrm{KL}}(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda^2}\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})\mathrm{d}\lambda. \tag{4.19}$$

For $\mathcal{D}_{\mathrm{rKL}}$ we can use Equation (4.14) with $f_{\mathrm{rKL}}(u) = -\log u$:

$$\mathcal{D}_{\mathrm{rKL}}(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda}\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})\mathrm{d}\lambda. \tag{4.20}$$

Finally, for the Jensen-Shannon Divergence, we have $f_{\mathrm{JS}}(u) = u\log(u) - (u+1)\log(u+1)/2$, the second derivative is $f''_{\mathrm{JS}}(u) = \frac{1}{u} - \frac{1}{u+1}$. The weight coefficient is then

$$\frac{1}{\lambda^3}f''_{\mathrm{JS}}\left(\frac{1}{\lambda}\right) = \frac{1}{\lambda^3}\left[\frac{1}{1/\lambda} - \frac{1}{1/\lambda+1}\right] = \frac{1}{\lambda^3}\left[\frac{\lambda^2}{1+\lambda}\right] = \frac{1}{\lambda(\lambda+1)}. \tag{4.21}$$

Applying this result in Equation (4.14), we get the expression of the Jensen-Shannon Divergence.

As we can see in Corollary 4.3.2, both $\mathcal{D}_{\mathrm{KL}}$ $\mathcal{D}_{\mathrm{rKL}}$ and $\mathcal{D}_{\mathrm{JS}}$ can be decomposed into a sum of PR-Divergences terms $\mathcal{D}_{\lambda\text{-PR}}$, each weighted with $1/\lambda^2$, $1/\lambda$ and $1/\lambda(\lambda+1)$ which are illustrated in Figure 4.10. Note that the weights for $\mathcal{D}_{\mathrm{KL}}$ and $\mathcal{D}_{\mathrm{rKL}}$ are not inverse. This is due to the fact that $\mathcal{D}_{\lambda\text{-PR}}$ is defined with respect to α_λ.

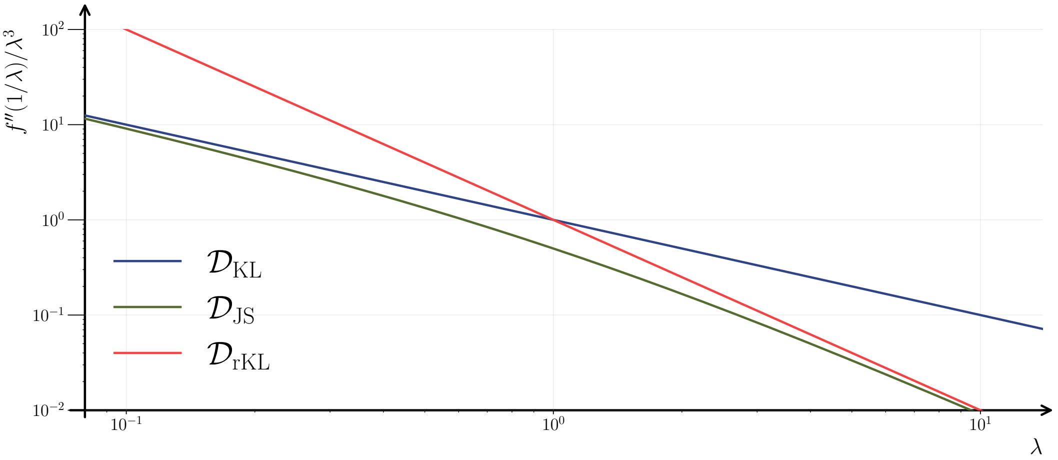


Fig. 4.10.: Weights associated by the Kullback-Leibler, the reverse Kullback-Leibler and the Jensen-Shannon divergences on the precision α_λ.

Nevertheless, we can see that for $\lambda < 1$, for trade-offs that assign more importance to Recall rather than Precision, the weights associated by the Kullback-Leibler are greater than the weights associated by the reverse Kullback-Leibler. This explains why $\mathcal{D}_{\mathrm{KL}}$ is more *mass-covering* than the reverse Kullback-Leibler. We can have the opposite analysis on trade-offs favoring Precision, i.e. $\lambda > 1$, explaining why $\mathcal{D}_{\mathrm{rKL}}$ is more *mode-seeking* than $\mathcal{D}_{\mathrm{KL}}$. In particular, this explains the mass-covering behavior observed in Normalizing Flows trained with log-likelihood maximization, i.e. $\mathcal{D}_{\mathrm{KL}}$. We have expressed these behaviors relatively one divergence with respect to the other by showing that one is more inclined to quality of diversity than the other, even if both divergences are both strongly inclined to diversity. We will experimentally show in Section 5, that both divergence are actually favoring low values of λ. The weights assigned by the Jensen-Shannon Divergence are more balanced, reflecting the symmetric property of $\mathcal{D}_{\mathrm{JS}}$.

In the preceding section, we demonstrated that the PR-Divergence can be expressed as an f-divergence. Subsequently, we illustrated how any f-divergence can be reformulated in terms of PR-Divergences, thus establishing the relationship on both sides.

4.3.2 Relation with Precision-Recall Cover

As explained in Section 3.4, the Precision and Recall Cover introduced by Cheema and Urner [21] is a generalization of the support-based approach of Precision and Recall. In other terms, the original work proves the connection between the Precision-Recall Cover and the PR-Curve only for $\lambda = 0$ and $\lambda = +\infty$. In this section, we show that the PR-Curve can be written as a limit of the Precision-Recall Cover for any $\lambda \in]0, +\infty[$, thus (1) extending the results of Cheema and Urner [21] and (2) providing a new generalization of PR-Curves.

Theorem 4.3.3 (PR-Curves in terms of Precision-Recall Cover)**.**
For any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$, for any $\lambda \in]0, +\infty[$, the PR-Curve can be written as a limit of the Precision-Recall Cover as:

$$\alpha_\lambda(P\|\widehat{P}) = \lim_{b\to 0} \mathrm{PC}_{b/\lambda,b}(P,\widehat{P}) + \lambda\mathrm{RC}_{b\lambda,b}(P,\widehat{P}) \tag{4.22}$$

$$\textit{and} \quad \beta_\lambda(P\|\widehat{P}) = \lim_{b\to 0} \frac{1}{\lambda}\mathrm{PC}_{b/\lambda,b}(P,\widehat{P}) + \mathrm{RC}_{b\lambda,b}(P,\widehat{P}). \tag{4.23}$$

Proof. First, we recall that $B_{\widehat{P}}(\boldsymbol{x}, b)$ denotes the ball centered on $\boldsymbol{x}$ of probability mass b with respect to $\widehat{P}$. By definition, the (a, b)−Precision Cover is:

$$\mathrm{PC}_{a,b}(P,\widehat{P}) = \mathbb{P}_{\boldsymbol{x}\sim\widehat{P}}\left[P(B_{\widehat{P}}(\boldsymbol{x}, b)) \geq a\right]. \tag{4.24}$$

Setting $a = b/\lambda$ and using the fact that $b = \widehat{P}(B_{\widehat{P}}(\boldsymbol{x}, b))$, we can write the Precision-Recall Cover as:

$$\mathrm{PC}_{a,b}(P,\widehat{P}) = \mathbb{P}_{\boldsymbol{x}\sim\widehat{P}}\left[P(B_{\widehat{P}}(\boldsymbol{x}, b)) \geq \widehat{P}(B_{\widehat{P}}(\boldsymbol{x}, b))/\lambda\right], \tag{4.25}$$

which is equivalent to:

$$\mathrm{PC}_{a,b}(P,\widehat{P}) = \mathbb{P}_{\boldsymbol{x}\sim\widehat{P}}\left[\lambda\frac{P(B_{\widehat{P}}(\boldsymbol{x}, b))}{\mu(B_{\widehat{P}}(\boldsymbol{x}, b))} - \frac{\widehat{P}(B_{\widehat{P}}(\boldsymbol{x}, b))}{\mu(B_{\widehat{P}}(\boldsymbol{x}, b))} \geq 0\right], \tag{4.26}$$

where $\mu(\cdot)$ is the Lebesgue measure. As a consequence of the Radon-Nikodym theorem [39], by taking the limit $b \to 0$, we have the following.

$$\lambda\frac{P(B_{\widehat{P}}(\boldsymbol{x}, b))}{\mu(B_{\widehat{P}}(\boldsymbol{x}, b))} - \frac{\widehat{P}(B_{\widehat{P}}(\boldsymbol{x}, b))}{\mu(B_{\widehat{P}}(\boldsymbol{x}, b))} \xrightarrow[b\to 0]{} \lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}). \tag{4.27}$$

We can rewrite the Precision Cover as:

$$\mathrm{PC}_{b/\lambda,b}(P,\widehat{P}) \xrightarrow[b\to 0]{} \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x})\mathbb{1}_{\{\lambda p(\boldsymbol{x}) \geq \widehat{p}(\boldsymbol{x})\}} \mathrm{d}\mu(\boldsymbol{x}) \tag{4.28}$$

Using the same reasoning on the Recall Cover by setting $a = b\lambda$, we can show that:

$$\mathrm{RC}_{b\lambda,b}(P,\widehat{P}) \xrightarrow[b\to 0]{} \int_{\mathcal{X}} p(\boldsymbol{x})\mathbb{1}_{\{\widehat{p}(\boldsymbol{x}) \geq \lambda p(\boldsymbol{x})\}} \mathrm{d}\mu(\boldsymbol{x}). \tag{4.29}$$

Finally, we can write the PR-Curve as a limit of the Precision-Recall Cover:

$$\alpha_\lambda(P\|\widehat{P}) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{4.30}$$

$$= \int_{\mathcal{X}} \lambda p(\boldsymbol{x}) \mathbb{1}_{\{\lambda p(\boldsymbol{x}) \leq \lambda \widehat{p}(\boldsymbol{x})\}} \mathrm{d}\mu(\boldsymbol{x}) + \int_{\mathcal{X}} \lambda p(\boldsymbol{x}) \mathbb{1}_{\{p(\boldsymbol{x}) \geq \lambda \widehat{p}(\boldsymbol{x})\}} \mathrm{d}\mu(\boldsymbol{x}). \tag{4.31}$$

And then, considering the limit $b \to 0$, we have that:

$$\alpha_\lambda(P\|\widehat{P}) = \lim_{b \to 0} \lambda \mathrm{PC}_{b/\lambda, b}(P, \widehat{P}) + \mathrm{RC}_{b\lambda, b}(P, \widehat{P}). \tag{4.32}$$

Finally, since $\beta_\lambda = \alpha_\lambda / \lambda$, we have the results.

4.4 Lower bounds on the PR-Divergence in Neural Networks

The learned distribution is defined as $\widehat{P} = G\#Q$, therefore the set of possible distributions highly depends on the set of functions G represented by neural networks. In particular the fundamental limits of neural networks should also translate to limitations on the distributions P. In this section, we show how the Lipschitz property of neural networks can be used to show that, depending on the target distribution P, we can not achieve any Precision and Recall trade-off. To do so, we reformulate the Precision-Recall Divergence from the integral expression to a probabilistic formulation. Then we introduce the reader to the Lipschitz continuity, and finally we show how lower bounds on the PR-Divergence can be used to understand the fundamental limits of Neural Networks for generative modeling by showcasing pathological cases.

4.4.1 Probabilistic Formulation of the PR-Divergence

We have shown in Section 4.2 that the Precision-Recall Divergence can be expressed in terms of the Total Variation Distance ($\mathcal{D}_{\mathrm{TV}}$) for $\lambda = 1$. This divergence has two popular formulations, the integral formulation:

$$\mathcal{D}_{\mathrm{TV}}(P\|\widehat{P}) = \frac{1}{2} \int_{\mathcal{X}} |p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x}, \tag{4.33}$$

and the probabilistic formulation:

$$\mathcal{D}_{\mathrm{TV}}(P\|\widehat{P}) = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| P(\mathcal{A}) - \widehat{P}(\mathcal{A}) \right|. \tag{4.34}$$

Therefore, the PR-Divergence can be trivially expressed as the probabilistic formulation for $\lambda = 1$. We show that there exists a similar reformulation for any $\lambda \in [0, +\infty]$:

Lemma 4.4.1 (Probabilistic Formulation of PR-Divergence).
For any $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ and $\lambda \in [0, +\infty]$, PR-Divergence can be expressed as:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A}) \right| - |\lambda - 1| . \tag{4.35}$$

Proof. First, we show that for any $\lambda \in [0, +\infty]$:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \frac{1}{2} \int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})| \, \mathrm{d}\mu(\boldsymbol{x}) - \frac{1}{2} |\lambda - 1| . \tag{4.36}$$

Then, we can prove that for any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ and $\lambda \in [0, +\infty]$, we have:

$$\frac{1}{2} \int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})| = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A}) \right| - \frac{1}{2} |\lambda - 1| . \tag{4.37}$$

The complete proof is detailed in Appendix B.1. Therefore, combining Equations (4.36) and (4.37), we have the result:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A}) \right| - |\lambda - 1| , \tag{4.38}$$

which concludes the proof.

This reformulation allows for a versatile manipulation of the PR-Divergence. In particular, to illustrate cases where the PR-Divergence is strictly positive, we look for specific sets $\mathcal{A}$ for which the difference $\lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A})$ is large. We will show that leveraging the Lipschitz continuity property of the generator function G can offer insights on the limits of Neural Networks, by lower bounding the PR-Divergence.

4.4.2 Lipschitz Properties of Neural Networks

Most of the neural network-based generative models benefit from Lipschitz constraints. Whether it is for training stability reasons, such as in GANs [16, 105, 134], to ensure numerical stability tracking the density ratio in Normalizing Flows [11, 12] or stable generation of samples with Diffusion Models [131], the Lipschitz property is a key feature of Neural Networks. We can define the Lipschitz continuity as follows:

Definition 4.4.2 (L_1-Lipschitz Continuity)**.**
Let $G : \mathcal{Z} \subset\!\!\mapsto \mathcal{X}$ be a function. We say that G is L_1-Lipschitz continuous if:

$$\forall \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{Z}, \quad \|G(\boldsymbol{z}_1) - G(\boldsymbol{z}_2)\| \leq L_1 \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|. \tag{4.39}$$

4.4.3 Pathological Cases to bound the PR-Divergence

In this section, we will show that the Lipschitz continuity of Neural Networks can limit the expressivity of the models. To do so, we assume that the latent distribution Q is a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, which is typically the case. Let Γ and γ be the Gamma and the lower incomplete Gamma functions respectively:

$$\forall s > 0,\ \forall r > 0 \quad \Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t} \mathrm{d}t \quad \text{and} \quad \gamma(s, r) = \int_0^{r} t^{s-1} e^{-t} \mathrm{d}t. \tag{4.40}$$

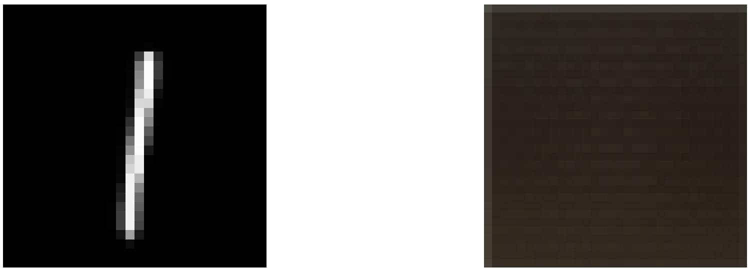
We can show that the PR-Divergence can be strictly positive for some target distributions P and some generator functions G.

Theorem 4.4.3 (L_1-Lipschitz forward G mapping fails to capture P)**.**
Let $P \in \mathcal{P}(\mathcal{X})$ be the target distribution defined on $\mathcal{X} \subset \mathbb{R}^d$, and let $\widehat{P} = G\#Q$ where $G : \mathcal{Z} \mapsto \mathcal{X}$ and Q be the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ defined on $\mathcal{Z} \subset \mathbb{R}^m$. Let $B_{R,G(\mathbf{0})}$ be the balls of radius R centered on $G(\mathbf{0})$. If G is L_1-Lipschitz, then we have the lower bound:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) \geq \sup_{R \geq 0} \left(\frac{\gamma\left(\frac{m}{2}, \frac{R^2}{2L_1^2}\right)}{\Gamma\left(\frac{m}{2}\right)} - \lambda P(B_{R,G(\mathbf{0})}) \right) - |\lambda - 1|\,. \tag{4.41}$$

Therefore, if there exists a ball for which the target distribution P satisfies $P(B_{R,G(\mathbf{0})}) < \frac{1}{\lambda}\gamma\left(\frac{m}{2}, \frac{R^2}{2L_1^2}\right)/\Gamma\left(\frac{m}{2}\right) - |1 - 1/\lambda|$, then the PR-Divergence is strictly positive.

Fig. 4.11.: Example of a target distribution for which Theorem 4.4.3 applies: the subset B_R concentrates little weight in P, but $\widehat{P}(B_R) = Q(G^{-1}(B_R))$ can only be as small as $Q(B_{R/L_1})$.


(a) MNIST

(b) CIFAR-10

Fig. 4.12.: Representation of $G(\mathbf{0})$ for a Residual Flow trained by Chen et al. [22] on the datasets MNIST and CIFAR-10.

Proof. First, since Q is the standard Gaussian distribution in $\mathbb{R}^m$, then for any $r \geq 0$ and $\boldsymbol{z} \sim Q$, the measure Q of the ball $B_{r,\mathbf{0}}$ can be computed using the cumulative distribution function of the χ^2 distribution:

$$Q(B_{r,\mathbf{0}}) = \mathbb{P}\left(\|\boldsymbol{z}\|^2 \leq r^2\right) = \frac{\gamma\left(\frac{m}{2}, \frac{r^2}{2}\right)}{\Gamma\left(\frac{m}{2}\right)}. \tag{4.42}$$

Then, since G is L_1-Lipschitz, thus $B_{R/L_1,0} \subseteq G^{-1}(B_{R,G(\mathbf{0})})$. Therefore, we have the following.

$$\widehat{P}(B_{R,G(\mathbf{0})}) \geq \widehat{P}(G(B_{R/L_1,G(\mathbf{0})})) = Q(B_{R/L_1,0}). \tag{4.43}$$

Using Lemma 4.4.1, we have:

$$\begin{aligned}
\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) &= \sup_{\mathcal{A}\subseteq\mathcal{X}} \left|\lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A})\right| - |\lambda - 1| &(4.44)\\
&\geq \sup_{R\geq 0} \widehat{P}(B_{R,G(\mathbf{0})}) - \lambda P(B_{R,G(\mathbf{0})}) - |\lambda - 1| &(4.45)\\
&\geq \sup_{R\geq 0} Q(B_{R/L_1,0}) - \lambda P(B_{R,G(\mathbf{0})}) - |\lambda - 1|\,. &(4.46)
\end{aligned}$$

Therefore, using the closed form of the measure of the ball $B_{r,\mathbf{0}}$ given in Equation (4.42), we have the result.

This theorem states that if the center of the latent Gaussian is mapped to a region with low density then the PR-Divergence can be bounded. This assumption is significant, but generally plausible. For example, it is not uncommon to see a multimodal density distribution with modes that are well distinct. Assuming that these modes are roughly of equal probability, a mapping is expected to evenly distribute these modes around the Gaussian distribution mean within the latent space. For instance, considering the Residual Flows trained on MNIST and CIFAR-10, we can observe in Figure 4.12 that $G(\mathbf{0})$ is either a plausible image and the low density region includes one of the modes of the distribution or a noisy image and the low density region is mostly empty.

Therefore, if we assume that the function G is L_1-Lipschitz, the mapping cannot expand arbitrarily. As a result, the mass represented by the low-density region, for example a ball $B_{R,G(\mathbf{0})}$, is mapped to a region larger than the ball $B_{R/L_1,\mathbf{0}}$, thus ensuring that the Gaussian measure associated with this broader zone is at least as great as $Q(B_{R/L_1})$. This concept is demonstrated through a one-dimensional example, as depicted in Figure 4.11.

To be more precise, the theorem says that if there exists a radius of the ball centered of $G(\mathbf{0})$ for which the target distribution P satisfies $P(B_{R,G(\mathbf{0})}) < \frac{1}{\lambda}\gamma\left(\frac{m}{2}, \frac{R^2}{2L_1^2}\right)/\Gamma\left(\frac{m}{2}\right) - |1 - 1/\lambda|$, then the PR-Divergence is strictly positive. There are three important parameters in this condition:

- The Lipschitz constant L_1 of the forward mapping G: $Q(B_{R/L_1,0})$ is a decreasing function of L_1, thus, illustrating the importance of the Lipschitz constant of large values of L_1 trading-off the stability of the model for the expressivity.
- The dimension m: As illustrated in Figure 4.13, the Gaussian measure of the ball $B_{R,\mathbf{0}}$ decreases with dimension m. In a high dimension, the condition is less likely to be satisfied. In low dimension, the Lipschitz constant L_1 has a greater impact.
- The trade-off parameter λ: The condition is less likely to be satisfied for extreme values of λ, high or low. In fact, the PR-divergence for extreme values of λ evaluates the overlap of the supports, and we do not make assumptions about the support of the target distribution P. However, there exists a sweat spot for λ where the condition is more likely to be satisfied. It illustrates that pathological will affect both the quality as weight is assigned between modes and the diversity as less weight is mapped to those modes.

This theorem is general. In the literature, results bounding the Lipschitz constant or the maximum precision (for $\lambda = +\infty$) exist for a disconnected manifold of P only [25,

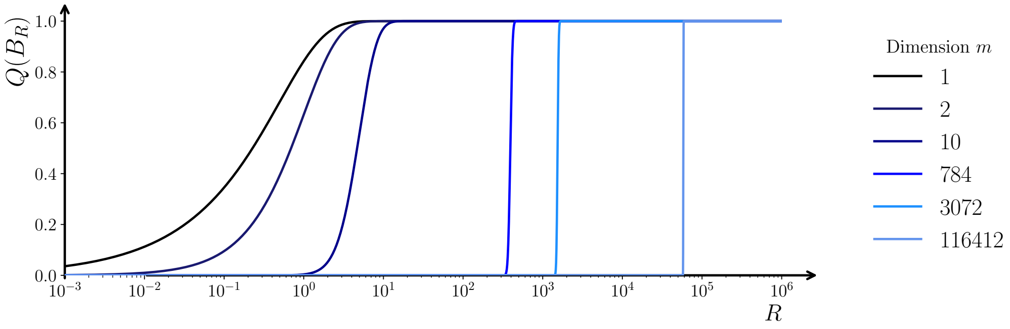


Fig. 4.13.: Representation of the Gaussian Measure of the ball $B_{R,\mathbf{0}}$ for different values of radius R in dimension m. The dimensions are corresponding the small dimensions but also to the dimension of the MNIST dataset, i.e. $m = 784$, CIFAR-10 dataset, i.e. $m = 3072$ and the ImageNet dataset, i.e. $m = 116412$.

118]. Theorem 4.4.3 is a generalization of these results, bounding the PR-Divergence for any support of the distribution P and for any trade-offs λ. Moreover, this result only concerns the Lipschitz constraint of the forward mapping. In Appendix A, we show that for Normalizing Flows, Lipschitz continuity of the inverse mapping can result in a lower bound on the PR-Divergence.

4.5 Concluding Remarks and Discussion

In this chapter, we addressed two questions regarding Precision/Recall for generative models:

- **Question Q1 :** *How can we unify the definitions of precision and recall for generative models?*
  We introduced Precision-Recall Divergence, a novel framework that encapsulates both precision and recall into a unified metric: the f-divergence, a widely used class of divergences in generative modeling. We also show how any f-divergence can be written in terms of Precision and Recall.
- **Question Q2:** *What Precision and Recall can be achieved with neural networks with bounded Lipschitz constants?*
  Building on the reformulation of the Precision-Recall Divergence, we demonstrated that the Lipschitz property (and bi-Lipschitz, when applicable) of the generator function G can be used to lower bound the PR-Divergence. In other words, we showed how the Lipschitz constraint of the generator function G can limit the expressivity of the models. We showed that PR-divergence can be strictly positive for some target distributions P and some generator functions G.

This chapter contributes to a more refined understanding of the quality-diversity metrics for generative models through the PR-Divergence. Additionally, it highlights the critical role of the Lipschitz constraint in limiting the overall expressivity of generative models. In other words, with limited expressivity, a model cannot achieve both high quality and high diversity. Based on this analysis, we propose in Chapter 5 an approach based on PR-Divergence to train models to be optimal for a given trade-off between Precision and Recall.

This theoretical analysis could be further improved, and we list the potential future works:

- **Symmetric PR-Divergence**: With such a definition, the PR-Divergence is not symmetric and is directly proportional to the Precision α_λ. It would be interesting to define a symmetric version of the PR-Divergence, such that

$\mathcal{D}_\lambda(P \| \widehat{P}) = \mathcal{D}_{1/\lambda}(\widehat{P} \| P)$. It would be a more intuitive metric to evaluate. The main problem lies in expressing any f-divergence in terms of a symmetric PR-Divergence. Although the interpretation of such a result would be easier to understand, the mathematical complexity of the problem is likely to be much higher.

- **Connecting expressivity with the AUC**: The area under the PR-Curve (AUC) plays a crucial role in the evaluation of generative models. It would be interesting to connect the expressivity of the model with the AUC, for instance, with the Lipschitz. This would provide a new interpretation of the AUC and a new way to understand the fundamental limits of generative models and the trade-off to be made in training generative models.

5 Tuning models to a user defined trade-off

> *It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong.*
>
> — **Richard P. Feynmann**
> (1965 Physics Nobel Laureate)

Contents

Question 3: *Can we train a generative model to directly focus on an explicit user-specified trade-off between Precision and Recall?*

In the previous chapter, we have shown that maximizing any point on the Precision-Recall Curve, which represents a specific trade-off between quality and diversity, corresponds to minimizing a specific f-divergence, the PR-Divergence. In this we will show how to train a generative model to tackle any trade-off between Precision and Recall. We will first recall the framework of f-GAN and show why it does not address this problem in Section 5.1. We will therefore propose a different method to tackle this problem in Section 5.2, and we will theoretically prove that this method converges in Section 5.2.2. Finally, we demonstrate the effectiveness of our method on both toy examples and real-world dataset in Section 5.3 with deep learning generative models and compare it with the state-of-the-art methods.

Contributions: One main contribution of this chapter is the following.

- We propose a method to train a generative model to focus on a specific trade-off between quality and diversity, and show that this method actually changes the trade-off of generative models.

This result has been published as:

- Alexandre Verine et al. "Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows". en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 32539–32573

5.1 Framework of f-GAN

As any trade-off Precision and Recall trade-off is represented by a specific f-divergence, it is natural to think that the f-GAN framework could be used to train a generative model to focus this f-divergence. First, we recall the f-GAN framework and show how it works in practice for GANs. We will show with some example that the f-GAN framework does not tackle this problem, and we will try to explain why. The f-GAN framework introduced in [90] is a generalization of the GAN framework. The goal is to train a neural network function G to minimize any divergence D_f between the data distribution $P \in \mathcal{P}(\mathcal{X})$ and the generated distribution $\widehat{P}_G = G\#Q$ where $Q \in \mathcal{P}(\mathcal{Z})$ is the latent distribution:

$$\min_{G \in \mathcal{G}} \mathcal{D}_f(P \| \widehat{P}_G). \tag{5.1}$$

However, to compute the f-divergence between P and $\widehat{P}_G$, we need to compute the density ratio $r(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$, which is intractable in practice. To address this problem, we leverage the dual approximation of the f-divergence defined as:

Definition 5.1.1 **($\mathcal{D}_{f,T}^{\text{dual}}(P\|\widehat{P})$ Dual approximation of an f-divergence $\mathcal{D}_f$).** *Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two probability distributions such that $P, \widehat{P} \ll \mu$, and f a suitable function for $\mathcal{D}_f$ to be an f-divergence. The dual approximation of the f-divergence is defined as:*

$$\mathcal{D}_{f,T}^{\text{dual}}(P\|\widehat{P}) = \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}}\left[f^*(T(\boldsymbol{x}))\right]. \tag{5.2}$$

Algorithm 2 Traditional f-GAN training procedure

```
repeat
    Update $T$ by ascending the gradient of
        $\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right].$
    Update $G$ by descending the gradient of
        $-\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f^*(T(\boldsymbol{x}))\right].$
until convergence.
```

Let $\mathcal{T}$ be the set of all measurable functions $T : \mathcal{X} \to \mathbb{R}$. The Theorem 2.1.3 introduced by [87] and in detail in Section 2.1.2, shows that f-divergence is an upper of the dual approximation on $\mathcal{T}$:

$$\mathcal{D}_f(P\|\widehat{P}) = \sup_{T\in\mathcal{T}} \mathcal{D}_{f,T}^{\text{dual}}(P\|\widehat{P}) \tag{5.3}$$

To estimate the f-divergence using the dual approximation, we introduce another neural network T trained as a discriminator to approximate the f-divergence. The optimization problem becomes a min-max problem:

$$\min_{G\in\mathcal{G}}\max_{T\in\mathcal{T}} \mathcal{D}_{f,T}^{\text{dual}}(P\|\widehat{P}) \;=\; \min_{G\in\mathcal{G}}\max_{T\in\mathcal{T}} \left(\mathbb{E}_{\boldsymbol{x}\sim P}[T(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}[f^*(T(\boldsymbol{x}))]\right). \tag{5.4}$$

A gradient descent algorithm can converge to a solution of this min-max problem by alternating between two steps: the discriminator step and the generator step. A simplified version of the training procedure is presented in Algorithm 2.

Theoretically training a model to minimize any f-divergence should be similar depending on the choice of the function f. However, the practical implementation of this training is more complex: for some f-divergences the training procedure is unstable training and, for others, does not converge.

The learning process involves two neural networks to represent the two functions. Neural networks typically consist of real-valued output functions with an output element-wise activation function. For the generator function G, it produces samples from the data space $\mathcal{X}$, which is commonly $\mathbb{R}^d$ or $\{0,1\}^d$, depending on the specific research community. This leads to the use of either no final activation function or a sigmoid activation function. On the other hand, the codomain of the discriminator function T must be a subset of the domain of f^*: $\mathrm{dom}(f^*)$. Therefore, the seminal work by Nowozin et al. [90] introduced specific activation functions for the discriminator for each f-divergence. Denote $W_\phi : \mathcal{X} \to \mathbb{R}$ as the neural network representing the discriminator and $a : \mathbb{R} \to \mathrm{dom}(f^*)$ as the final activation function. Therefore, the discriminator function is defined as $T_\phi(\boldsymbol{x}) = a(W_\phi(\boldsymbol{x}))$, where $\phi \in \Phi$

f-divergence	$f^*(t)$	$\mathrm{dom}(f^*)$	Output Activation $a(w)$
$\mathcal{D}_{\mathrm{KL}}$	$\exp(t-1)$	$\mathbb{R}$	w
$\mathcal{D}_{\mathrm{rKL}}$	$-1-\log(-t)$	$]-\infty,0[$	$-\exp(-w)$
$\mathcal{D}_{\mathrm{TV}}$	t	$[-\frac{1}{2},\frac{1}{2}]$	$\frac{1}{2}\tanh(w)$
$\mathcal{D}_{\chi^2}$	$\frac{1}{2}t^2+t$	$[-1,+\infty]$	$-1+\log(1+\exp(w))$
$\mathcal{D}_{\mathrm{GAN}}$	$-\log(1-\exp(-t))$	$]-\infty,0[$	$-\log(1+\exp(-w))$

Tab. 5.1.: Examples of activation functions for the discriminator in the f-GAN framework.

is the parameter vector of W_ϕ. Examples of the activation function a are listed in Table 5.1. The maximization step thus solves:

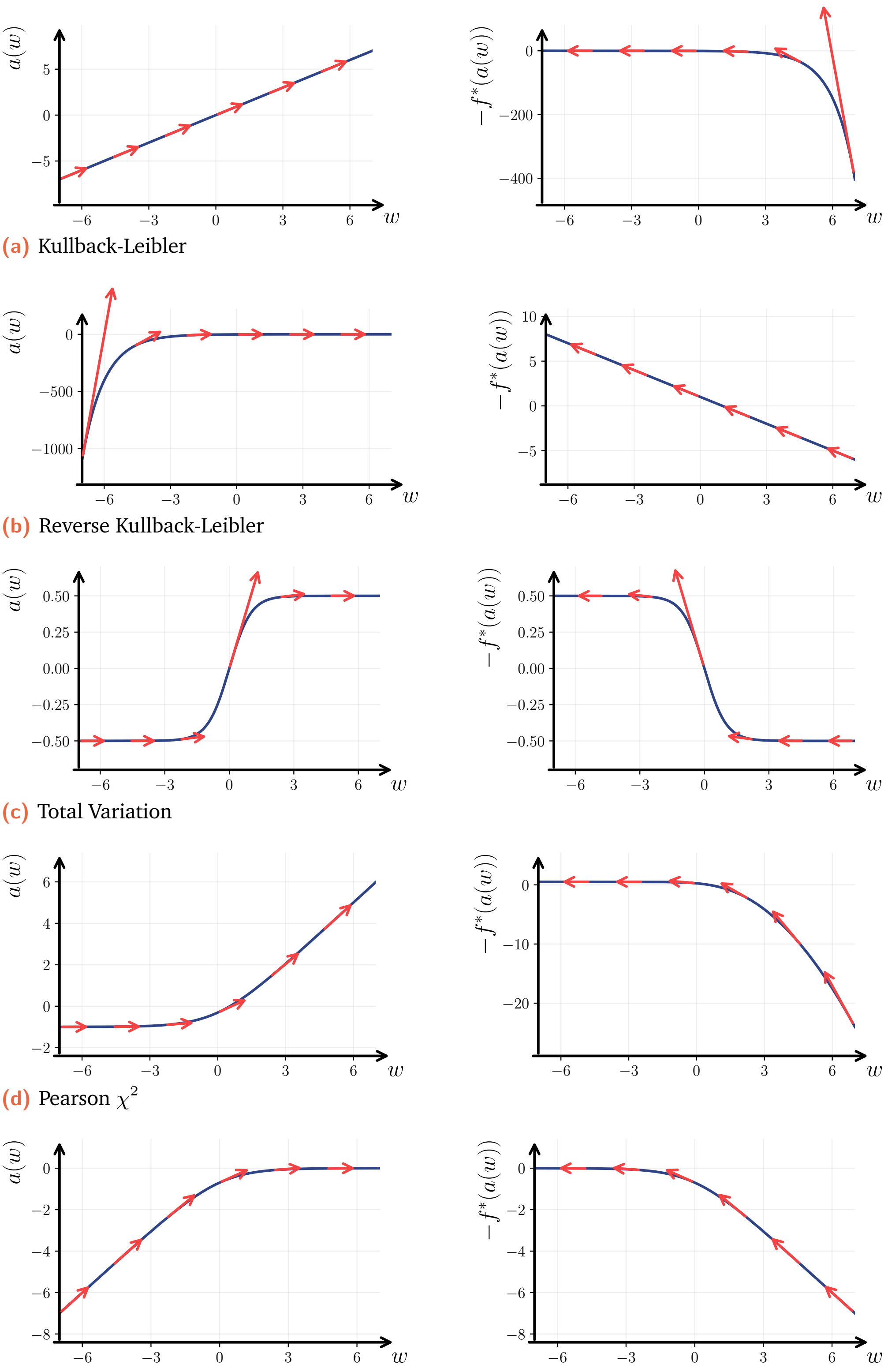
$$\max_{\phi\in\Phi} \left(\mathbb{E}_{\boldsymbol{x}\sim P}\left[a(W_\phi(\mathbf{x}))\right]-\mathbb{E}_{\mathbf{x}\sim\widehat{P}_G}\left[f^*(a(W_\phi(\mathbf{x})))\right]\right). \tag{5.5}$$

To update the discriminator, we need to compute the gradient of the objective with respect to the parameters of the discriminator. This is done by backpropagation through the discriminator and the activation function. The chain rule on $\nabla_\phi\mathcal{D}^{\mathrm{dual}}_{f,a(W_\phi)}$ gives the following:

$$\mathbb{E}_{\boldsymbol{x}\sim P}\left[\frac{\partial a(W_\phi(\mathbf{x}))}{\partial W_\phi(\mathbf{x})}\frac{\partial W_\phi(\mathbf{x})}{\partial\phi}\right]-\mathbb{E}_{\mathbf{x}\sim\widehat{P}_G}\left[\frac{\partial f^*(a(W_\phi(\mathbf{x})))}{\partial W_\phi(\mathbf{x})}\frac{\partial W_\phi(\mathbf{x})}{\partial\phi}\right]. \tag{5.6}$$

The update depends on both the gradient of $a(w)$ and $-f^*(a(w))$ with respect to w. We give some examples of functions $a(w)$ and $-f^*(a(w))$ in Figure 5.1. The left column corresponds to the loss of points $\boldsymbol{x}$ drawn from P, and the right column corresponds to the loss for points $\boldsymbol{x}$ drawn from $\widehat{P}_G$. On the different curves, we represented the direction and magnitude of the gradient in red to maximize the loss. For every divergence, it pushes $W_\phi(\boldsymbol{x})=w$ to be high for $\boldsymbol{x}\sim P$ and low for $\boldsymbol{w}\sim\widehat{P}$. By looking at these examples, we can grasp how the function f, and especially its convex conjugate f^* can affect the stability of the training. We can identify three main cases:

- **Instability:** For both the $\mathcal{D}_{\mathrm{KL}}$ and the $\mathcal{D}_{\mathrm{rKL}}$ divergences, the gradient can be exponentially large for some values of w. This can lead to unstable training, as the discriminator can drastically change from one iteration to the next.
- **Vanishing gradients:** The gradients for the divergence $\mathcal{D}_{\mathrm{TV}}$ are close to 0 in most of the domain for both real and generated data points. This means that for any random initialization of the discriminator, the gradient will most likely be close to 0 and the optimization will stall.
- **Balanced gradients:** The gradients of the $\mathcal{D}_{\chi^2}$ and $\mathcal{D}_{\mathrm{GAN}}$ divergences are balanced between the real and generated data points. The loss function being



(a) Kullback-Leibler

(b) Reverse Kullback-Leibler

(c) Total Variation

(d) Pearson χ^2

(e) GAN

Fig. 5.1.: Examples of the activation function $a(w)$ and $-f^*(a(w))$ for the Kullback-Leibler, Pearson χ^2 and GAN divergences.

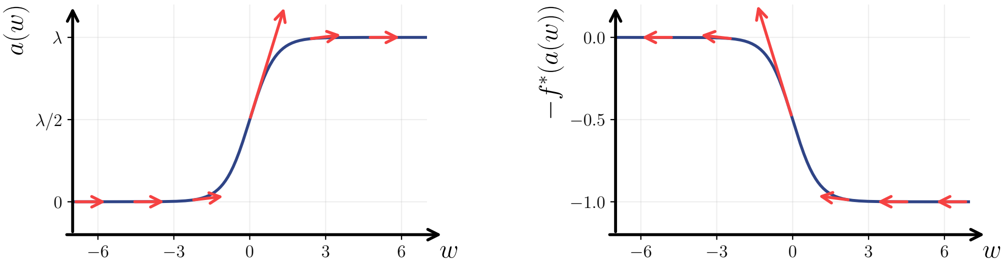


Fig. 5.2.: Examples of the activation function $a(w)$ and $-f_\lambda^*(a(w))$ for the PR-Divergence.

either for high or low values either has a linear or polynomial behavior and thus the gradients are not too large or too small.

Similar experimental observations have been observed in the works of Nowozin et al. [90], Grover et al. [46], [119], Um and Suh [125], Li and Farnia [76].

5.2 Minimizing the PR-Divergence

An intuitive approach to minimize the PR-Divergence would be to directly use the f-GAN framework with the PR-Divergence. However, the function f_λ is very similar to f_{TV} as

$$\text{dom}(f_\lambda) = [0, \lambda] \quad \text{and} \quad f_\lambda^*(t) = \lambda f_{\text{TV}}^*(t) + \max(\lambda - 1, 0). \tag{5.7}$$

Therefore, we might expect the same issues as for the Total Variation divergence. First, we can see in Figure 5.2 that the gradients of the PR-Divergence have the same behavior as the Total Variation. The gradients are close to 0 for most of the domain and the discriminator will not learn anything. We expect the gradients to vanish. This is confirmed by our experiments in Section 5.3 where we show that the PR-Divergence is not minimized by the f-GAN framework. For example, we will train a BigGAN model [16] on the CIFAR-10 dataset to minimize the PR-Divergence. We can compare the naive approach, i.e., the f-GAN framework, and our method, detailed in Section 5.2. We can see in Figure 5.3 that the naive approach does not manage to converge, as the discriminator T will not learn to differentiate between the real and generated data.

To overcome this issue, we propose a method to train a generative model to focus on a specific trade-off between quality and diversity. Our method is based on the idea of estimating the PR-Divergence by using the primal form of the f-divergence based on the density ratio. Training GAN using the density ratio has first been introduced in the work of Uehara et al. [124] and then Poole et al. [97] to improve GAN training for various f-divergences. In addition to this work, we compare different

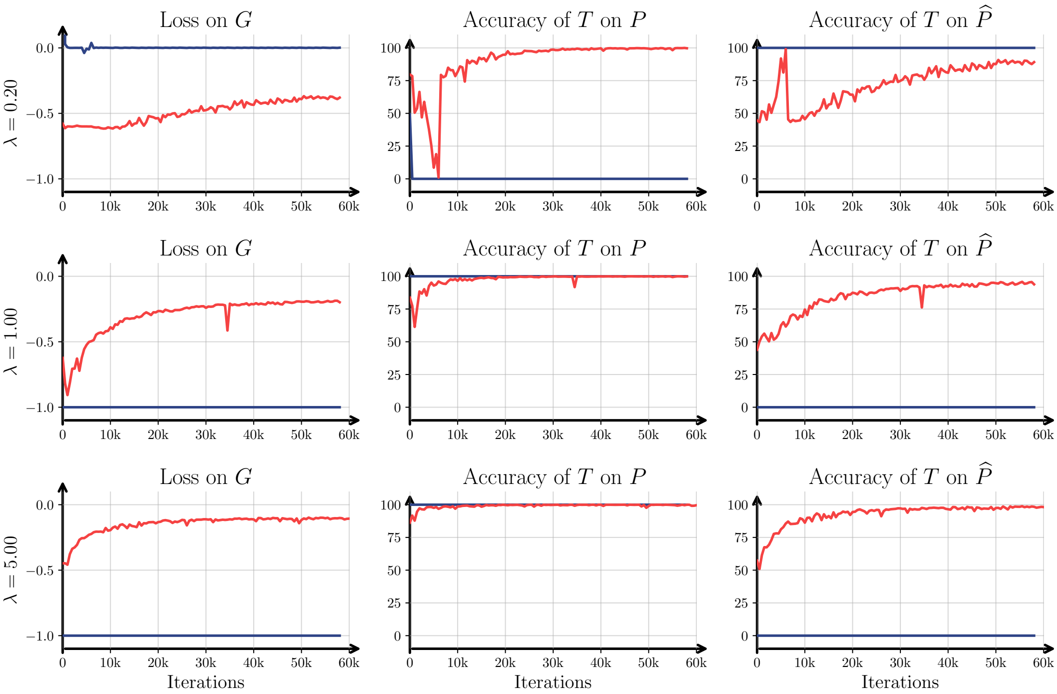


Fig. 5.3.: Illustration of the naive approach to minimize the PR-Divergence, compared to our method proposed in Section 5.2.

discriminator-based estimators for the density and we will prove that if the density ratio is well estimated, the PR-Divergence will be correctly estimated.

5.2.1 The two-objectives method

If the naive approach fails because the discriminator cannot converge due to f_λ^*, we can try to change the objective of the discriminator. To estimate the value of any f-divergence $\mathcal{D}_f$, we propose to train the discriminator using an auxiliary divergence based on a function $g \neq f$. The main idea is to choose a function g that is suitable for stable discriminator training. If T_g is trained to estimate the f-divergence $\mathcal{D}_g$ between P and $\widehat{P}_G$ by optimizing the following objective:

$$T_g \in \arg\max_{T \in \mathcal{T}} \mathcal{D}_g^{\text{dual}}(P \| \widehat{P}_G), \tag{5.8}$$

then at optimality, we have:

$$\nabla g^*(T_g^{\text{opt}}(\boldsymbol{x})) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, \tag{5.9}$$

and any f-divergence can be computed as follows using T_g:

$$\mathcal{D}_f(P \| \widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}) = \int_{\mathcal{X}} p(\boldsymbol{x}) f\left(\nabla g^*(T_g^{\text{opt}}(\boldsymbol{x}))\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{5.10}$$

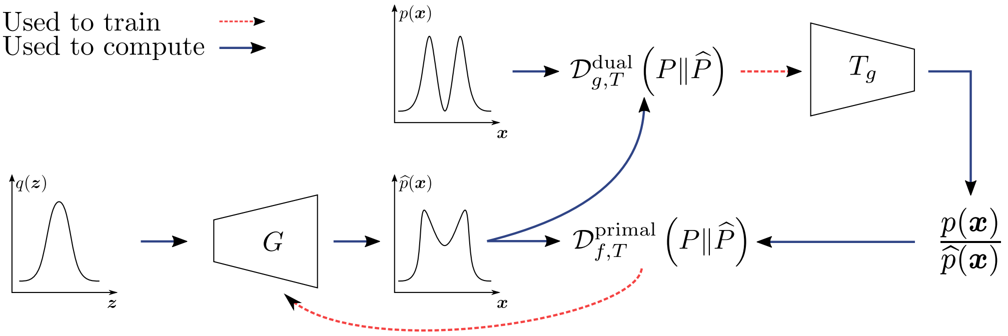

Used to train
Used to compute
$p(\boldsymbol{x})$
$\boldsymbol{x}$
$\mathcal{D}^{\text{dual}}_{g,T}\left(P\|\widehat{P}\right)$
T_g
$q(\boldsymbol{z})$
$\boldsymbol{z}$
G
$\widehat{p}(\boldsymbol{x})$
$\boldsymbol{x}$
$\mathcal{D}^{\text{primal}}_{f,T}\left(P\|\widehat{P}\right)$
$\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$

Fig. 5.4.: Illustration of the PR-GAN training procedure.

In practice, T_g is parameterized by a neural network trained to maximize $\mathcal{D}^{\text{dual}}_{g,T}$. We can still estimate the value the f-divergence by using the primal approximation defined as follows:

Definition 5.2.1 ($\mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P})$ **Primal approximation of an f-divergence $\mathcal{D}_f$).** *Let $P \in \mathcal{P}(\mathcal{X})$ and $\widehat{P} \in \mathcal{P}(\mathcal{X})$ be two probability distributions such that $P, \widehat{P} \ll \mu$. For any functions $T : \mathcal{X} \to \mathbb{R}$, $f : \mathbb{R}^+ \to \mathbb{R}$ and $g : \mathbb{R}^+ \to \mathbb{R}$ such that $\mathcal{D}_f$ and $\mathcal{D}_g$ are f-divergences*

$$\mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P}) = \int_{\mathcal{X}} p(\boldsymbol{x}) f\left(r(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}), \tag{5.11}$$

where $r : \mathcal{X} \to \mathbb{R}^+$, the estimation of the density ratio is given by $r(\boldsymbol{x}) = \nabla g^*(T(\boldsymbol{x}))$.

To implement this approach, we propose this simplified approach in Algorithm 3 with $f = f_\lambda$, which we illustrate in Figure 5.4. The training algorithm is very similar to the traditional f-GAN training procedure, and the computational complexity is exactly the same. Instead of solving a min-max problem on the same objective function:

$$\min_{G \in \mathcal{G}} \max_{T \in \mathcal{T}} \mathcal{D}^{\text{dual}}_{f,T}(P\|\widehat{P}_G), \tag{5.12}$$

we solve simultaneously a bi-level problem on two different objectives:

$$\max_{T \in \mathcal{T}} \mathcal{D}^{\text{dual}}_{g,T}(P\|\widehat{P}_G) \quad \text{and} \quad \min_{G \in \mathcal{G}} \mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P}_G). \tag{5.13}$$

Note that the training procedure no longer depends on f^*_λ.

The success of this approach depends on how well $r(\boldsymbol{x})$ approximates the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. In the next section, we will show that, depending on g, if the density ratio is well estimated, the PR-Divergence can be correctly estimated.

Algorithm 3 PR-GAN training procedure

```
repeat
    Update $T$ by ascending the gradient of
        $\mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[g^*(T(\boldsymbol{x}))\right].$
    Update $G$ by descending the gradient of
        $\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[f(\nabla g^*(T(\boldsymbol{x})))\right].$
until convergence.
```

5.2.2 Theoretical guaranty of convergence

To prove that the PR-Divergence can be correctly estimated by the primal approximation, we need to be able to estimate how well the density ratio is estimated. A popular tool for estimating the quality of function approximations is the Bregman divergence introduced by Bregman [15]. It is particularly relevant for matching density ratios and is used in the context of generative models [55, 115, 124]. The Bregman divergence is a general framework for measuring the difference between two points in a convex space. It is defined as follows:

Definition 5.2.2 (Bregman Divergence)**.**
Let $g : \mathbb{R}^d \to \mathbb{R}$ be a strictly convex differentiable function on a convex set Ω. The Bregman divergence associated to g between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as:

$$\mathrm{Breg}(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x}) - g(\boldsymbol{y}) - \nabla g(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}). \tag{5.14}$$

The Bregman divergence is the different between $g(\boldsymbol{x})$ and the Taylor expansion of g on $\boldsymbol{y}$ evaluated in $\boldsymbol{x}$. The Bregman divergence is always positive and is equal to 0 if and only if $\boldsymbol{x} = \boldsymbol{y}$. By choosing the right function f, the Bregman divergence can be equal to the squared Euclidean distance, the KL-Divergence, the squared Hellinger distance or the squared Mahalanobis distance. In Figure 5.5, we illustrate

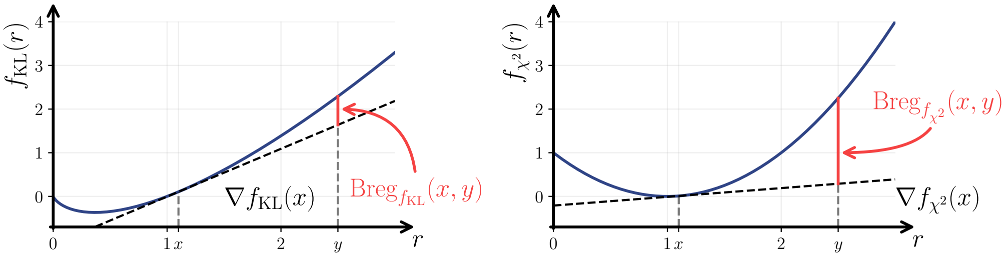


Fig. 5.5.: Illustration of the Bregman divergence for $g = f_{\mathrm{KL}}$ and $g = f_{\chi^2}$ for two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}$.

the Bregman divergence two functions $g = f_{\mathrm{KL}}$ and $g = f_{\chi^2}$, for two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}$, the divergences can greatly differ.

Minimizing the Bregman divergence to estimate an f-divergence is a known result introduced by Nguyen et al. [86] and Sugiyama et al. [115]. However, we can prove that the Bregman divergence between the density ratio and the estimated density ratio is *exactly* the approximation error of the f-divergence $\mathcal{D}_g$ by the dual approximation $\mathcal{D}_{g,T}^{\mathrm{dual}}$. Therefore, minimizing the former will also minimize the latter. This dual approximation reformulation is the following Theorem:

Theorem 5.2.3 (Error of the estimation of an f-divergence under the dual form.)**.** *For any discriminator $T : \mathcal{X} \to \mathbb{R}$ and $r(\boldsymbol{x}) = \nabla f^*(T(\boldsymbol{x}))$,*

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}_{g,T}^{\mathrm{dual}}(P\|\widehat{P}) = \mathbb{E}_{\widehat{P}}\left[\mathrm{Breg}_g\left(r(\boldsymbol{x}), \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right]. \tag{5.15}$$

Proof. For any $T : \mathcal{X} \to \mathbb{R}$,

$$\mathcal{D}_{g,T}^{\mathrm{dual}} = \mathbb{E}_{\boldsymbol{x}\sim P}\left[T(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[g^*\left(T(\boldsymbol{x})\right)\right] \tag{5.16}$$

$$= \mathbb{E}_{\boldsymbol{x}\sim\widehat{P}}\left[\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}T(x) - g^*\left(T(\boldsymbol{x})\right)\right] \tag{5.17}$$

Using the optimal discriminator T^{opt} we have that $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) = \nabla g^*(T^{\mathrm{opt}}(\boldsymbol{x}))$:

$$\mathcal{D}_g(P\|\widehat{P}) - D_{g,T}^{\mathrm{dual}}(P\|\widehat{P}) = \mathcal{D}_{g,T^{\mathrm{opt}}}^{\mathrm{dual}}(P\|\widehat{P}) - \mathcal{D}_{g,T}^{\mathrm{dual}}(P\|\widehat{P}) \tag{5.18}$$

$$= \mathbb{E}_{\widehat{P}}\left[\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\left(T^{\mathrm{opt}}(\boldsymbol{x}) - T(\boldsymbol{x})\right) - g^*\left(T^{\mathrm{opt}}(\boldsymbol{x})\right) + g^*\left(T(\boldsymbol{x})\right)\right] \tag{5.19}$$

$$= \mathbb{E}_{\widehat{P}}\left[\nabla g^*(T^{\mathrm{opt}}(\boldsymbol{x}))\left(T^{\mathrm{opt}}(\boldsymbol{x}) - T(\boldsymbol{x})\right) - g^*\left(T^{\mathrm{opt}}(\boldsymbol{x})\right) + g^*\left(T(\boldsymbol{x})\right)\right] \tag{5.20}$$

Recall that for any continuously differentiable strictly convex function g, the Bregman divergence of g is $\mathrm{Breg}_g(a,b) = g(a) - g(b) - \langle\nabla g(b), a-b\rangle$. So we have:

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}_{g,T}^{\mathrm{dual}}(P\|\widehat{P}) = \mathbb{E}_{\widehat{P}}\left[\mathrm{Breg}_{g^*}\left(T(\boldsymbol{x}), T^{\mathrm{opt}}(\boldsymbol{x})\right)\right] \tag{5.21}$$

Let us now use the following property: $\mathrm{Breg}_g(a,b) = \mathrm{Breg}_{g^*}(a^*,b^*)$ where $a^* = \nabla g(a)$ and $b^* = \nabla g(b)$.

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{dual}}_{g,T}(P\|\widehat{P}) = \mathbb{E}_{\widehat{P}}\left[\text{Breg}_g\left(\nabla g^*(T(\boldsymbol{x})), \nabla g^*(T^{\text{opt}}(\boldsymbol{x}))\right)\right] \tag{5.22}$$

$$= \mathbb{E}_{\widehat{P}}\left[\text{Breg}_g\left(\nabla g^*(T(\boldsymbol{x})), \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right] \tag{5.23}$$

With $r(\boldsymbol{x}) = \nabla f^* T(\boldsymbol{x})$ as our estimator of $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. We have

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{dual}}_{g,T}(P\|\widehat{P}) = \mathbb{E}_{\widehat{P}}\left[\text{Breg}_g\left(r(\boldsymbol{x}), \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right], \tag{5.24}$$

which concludes the proof.

Based on this observation, the quality of the estimation of f-divergence is not only related to g but also to ∇g. In fact, we can show that if g is strongly convex, then the error on the estimation error of $\mathcal{D}^{\text{primal}}_{f,T}$ is bounded:

Theorem 5.2.4 (Bound on the estimation of an f-divergence using an auxiliary g-divergence)**.**
Let $f, g : \mathbb{R}^+ \to \mathbb{R}$ be such that g is μ-strongly convex, f is σ-Lipschitz, and $\mathcal{D}_f$, $\mathcal{D}_g$ be f-divergences. For any discriminator $T : \mathcal{X} \to \text{dom}(g^*)$, let $r(\boldsymbol{x}) = \nabla g^*(T(\boldsymbol{x}))$. Then:

$$\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{dual}}_{g,T} \leq \epsilon \implies \left|\mathcal{D}_f(P\|\widehat{P}) - \mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P})\right| \leq \sigma\sqrt{\frac{2\epsilon}{\mu}}. \tag{5.25}$$

Proof. Assume that g is μ-strongly convex, then:

$$\text{Breg}_g(a, b) \geq \frac{\mu}{2}\|a - b\|^2. \tag{5.26}$$

If $\mathbb{E}_{\widehat{P}}\left[\text{Breg}_g\left(r(\boldsymbol{x}), \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right] \leq \epsilon$ and if g is μ-strongly convex, then

$$\mathbb{E}_{\widehat{P}}\left[\left(r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)^2\right] \leq \frac{2\epsilon}{\mu}. \tag{5.27}$$

Consider an arbitrary f-divergence $\mathcal{D}_f(P\|\widehat{P})$ and its primal approximation $\mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P})$:

$$\left|\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{primal}}_{g,T}(P\|\widehat{P})\right| = \left|\mathbb{E}_{\widehat{P}}\left[g\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) - g\left(r(\boldsymbol{x})\right)\right]\right| \tag{5.28}$$

$$\leq \mathbb{E}_{\widehat{P}}\left[\left|g\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) - g\left(r(\boldsymbol{x})\right)\right|\right]. \tag{5.29}$$

If we assume that f is σ-Lipschitz, then:

$$\begin{aligned}\left|\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{primal}}_{g,T}(P\|\widehat{P})\right| &\leq \mathbb{E}_{\widehat{P}}\left[\sigma\left|r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right|\right] && (5.30)\\ &\leq \sigma\mathbb{E}_{\widehat{P}}\left[\left|r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right|\right]. && (5.31)\end{aligned}$$

Using Jensen's inequality, we have the following.

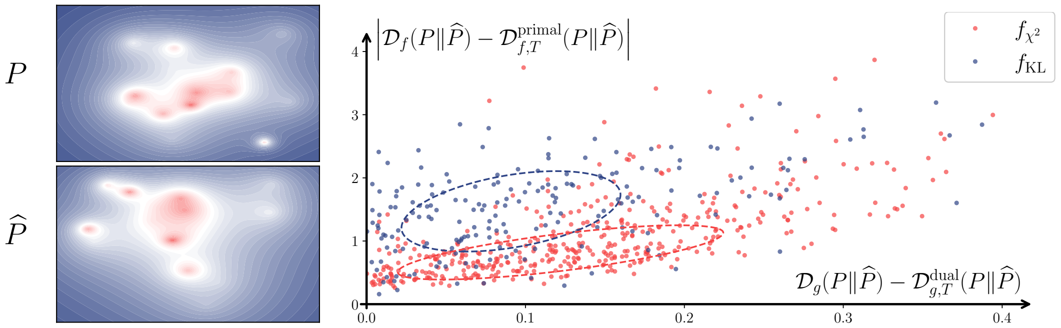
$$\left|\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{primal}}_{g,T}(P\|\widehat{P})\right| \leq \sigma\sqrt{\mathbb{E}_{\widehat{P}}\left[\left(r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)^2\right]}. \tag{5.32}$$

Finally, using equation (5.27), we have:

$$\left|\mathcal{D}_g(P\|\widehat{P}) - \mathcal{D}^{\text{primal}}_{g,T}(P\|\widehat{P})\right| \leq \sigma\sqrt{\frac{2\epsilon}{\mu}}, \tag{5.33}$$

which concludes the proof.

This theorem shows that if the discriminator T is sufficiently well trained to maximize $\mathcal{D}^{\text{dual}}_{g,T}(P\|\widehat{P})$ then the primal approximation $\mathcal{D}^{\text{primal}}_{f,T}(P\|\widehat{P})$ converges to the true value of $\mathcal{D}_f(P\|\widehat{P})$, under the conditions that g is strongly convex and f is Lipschitz. In our case, the PR divergence is a piecewise linear function with a Lipschitz constant of λ. For the auxiliary function g we can take $g = f_{\chi^2}$ which is strongly convex, since it also has a stable training behavior. We can then expect that the PR-Divergence can be correctly estimated by the primal approximation. Experimentally, we compare the estimation of the PR-Divergence using f_{χ^2} and f_{KL} as auxiliary divergences. For randomly generated Gaussian mixtures in 2D, we show in Figure 5.6 that the estimation of the PR-Divergence is better using f_{χ^2} than f_{KL}. The average error of the estimation, the Mahalanobis distance, is lower for f_{χ^2}.



(a) Example of distributions.

(b) Error of estimation of the PR-Divergence.

Fig. 5.6.: Primal approximation using $g = f_{\chi^2}$ and $g = f_{KL}$ for 2D random Gaussian mixtures. The Mahalanobis distance for each set in represented by the ellipses.

5.3 Experiments

In this section, we use our proposed approach to train various models to minimize the PR-Divergence. More specifically, we train models on 2D synthetic data in order to visualize how the PR-Curves behave when the models are trained on PR with various λ. Then, we increase the dimensionality of the data and train Normalizing Flows on the MNIST [132] and Fashion-MNIST [130] datasets and GANs on the CIFAR-10 [3] and CelebA [78] datasets. Finally, to show that our method scales to large datasets, we show that our method can be used to fine-tune models on larger datasets such as the ImageNet dataset [27] and the FFHQ [63].

In two dimensions, we will evaluate the quality and diversity using PR-Curves introduced by Sajjadi et al. [103]. In high dimension, as PR-Curves are less reliable, we use the support-based approach of Precision and Recall introduced by Kynkäänniemi et al. [73]. We will also use the Fréchet Inception Distance [51] and the Inception Score [104]. For GANs, we will add Density and Coverage [85]. For Precision, Recall, Density, and Coverage, we used 10k samples and $k = 3$ for MNIST and Fashion-MNIST, $k = 5$ for CIFAR-10, CelebA, ImageNet, and FFHQ. We used 50k samples for the Inception Score and the Fréchet Inception Distance. In Appendix C, we provide more details on the experimental setup, for instance, the exact architecture of the models, the hyperparameters, the optimizers, etc.

5.3.1 Training on 2D synthetic data

First, we train a model on a synthetic dataset to visualize the PR-Curves for different values of λ. If we choose a model with high expressivity, the output distribution $\widehat{P}$ will be able to fit the data distribution P perfectly. Therefore, we choose a model with poor expressivity: a RealNVP [31] with only 3 coupling layers. Even, if this

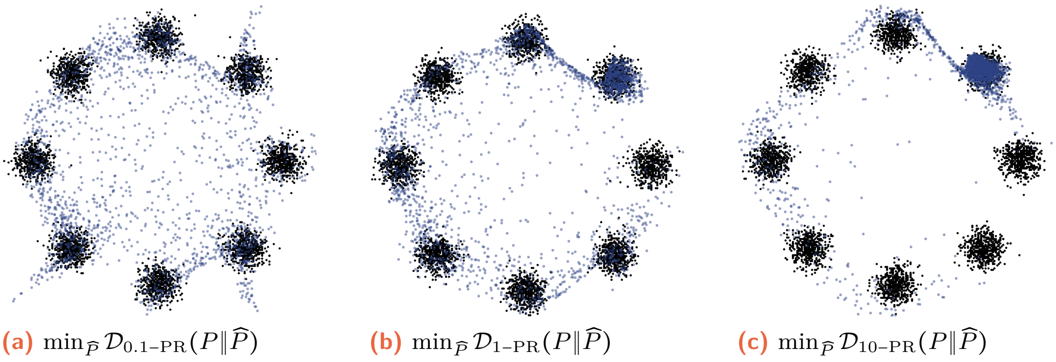

(a) $\min_{\widehat{P}} \mathcal{D}_{0.1\text{-PR}}(P\|\widehat{P})$ (b) $\min_{\widehat{P}} \mathcal{D}_{1\text{-PR}}(P\|\widehat{P})$ (c) $\min_{\widehat{P}} \mathcal{D}_{10\text{-PR}}(P\|\widehat{P})$

Fig. 5.7.: $\widehat{P}$ minimizing the PR-Divergences for different values of λ on the 8-Gaussians dataset. Samples from P are black and samples from $\widehat{P}$ are in blue.

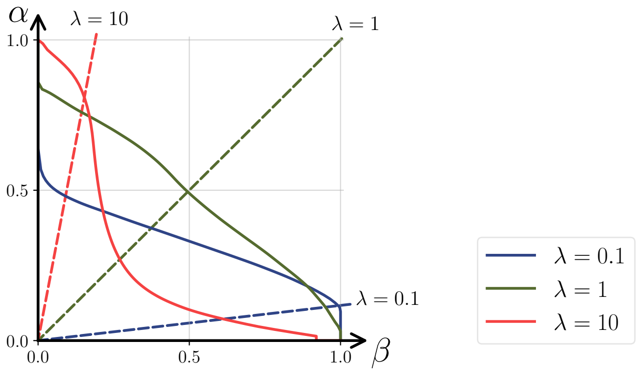


Fig. 5.8.: PR-Curves for the 8-Gaussians dataset.

model is a Normalizing Flow, it can be trained to minimize any f-divergence with a discriminator with the Flow-GAN framework detailed in Section 2.2.2. We train the models using our approach on the 8-Gaussians dataset with $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$. The learned distributions $\widehat{P}$ are shown in Figure 5.7. We can observe that the model with $\lambda = 0.1$ is mass covering as it covers the 8 modes of the data distribution. The model with $\lambda = 1$ is more balanced between quality and diversity. The model with $\lambda = 10$ is focused on a single mode that illustrates a mode-seeking behavior.

We show the PR-Curves in Figure 5.8. We can see that the PR-Curves are very different for each value of λ. The model train to maximize $\alpha_{0.1}$ performs best in $\alpha_{0.1}$, but performs poorly in α_1 and α_{10}. The same observation can be made for the other models. This experiment clearly shows that using our approach, we can train models to focus on a specific trade-off between quality and diversity.

5.3.2 Training Normalizing Flows

Normalizing Flows is a good example for testing our approach on low-complexity datasets such as MNIST and Fashion-MNIST. Normalizing Flows are traditionally trained with maximum likelihood estimation (MLE), in other words, by minimizing the Kullback-Leibler divergence, a divergence that usually promotes diversity. For that reason, they typically produce lower-quality samples. However, it is still a widely used model for generative tasks due to its ability to track the density. In this experiment, we train a GLOW model [70] on the MNIST and Fashion-MNIST datasets to minimize the PR-Divergence. We observe that training the Normalizing Flow for a few steps with the MLE objective before switching to any f-divergence is beneficial for the discriminator and increases the speed on convergence. Therefore, in the experiments (in this section only) we first train the model with the MLE objective for 10 epochs before switching to the PR-Divergence. We also compare with models trained with the Flow-GAN procedure but using the Kullback-Leibler or reverse Kullback-Leibler.

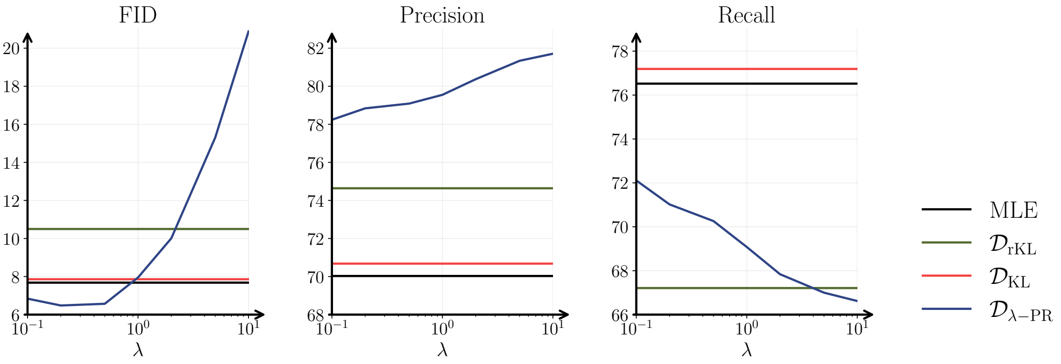


Fig. 5.9.: Quantitative metrics for the GLOW model trained on the MNIST dataset. FID (↓), P (↑) and R (↑).

Results on MNIST: With the MNIST dataset (and with Fashion-MNIST) the notion of Precision and Recall can be easily inspected visually: Precision is high if the digits are well formed, and Recall is high if the digits are diverse. We present in Figure 5.10 the samples from the models trained with MLE, $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$. We can see that the model trained with MLE and the model trained with $\lambda = 0.1$ produce various samples but of poor quality. The model with $\lambda = 10$ produces samples of better quality, but only samples of classes 0, 1, 7, 6, and 9. The model with $\lambda = 1$ is a trade-off between the two different models. In Figure 5.9, we show how the different metrics evolve with respect to the loss function. Note how MLE and $\mathcal{D}_{\mathrm{KL}}$ are close in terms of FID, Precision, and Recall. This result is expected since MLE is a discriminator-free method to minimize $\mathcal{D}_{\mathrm{KL}}$. Finally, we can see how the Precision is increasing with λ and the Recall is decreasing. The model trained with the reverse Kullback-Leibler divergence has a better Precision than the model trained with the Kullback-Leibler divergence but a lower Recall, however, its Precision is lower than that of the model trained with the PR-Divergence. As observed in Corollary 4.3.2,

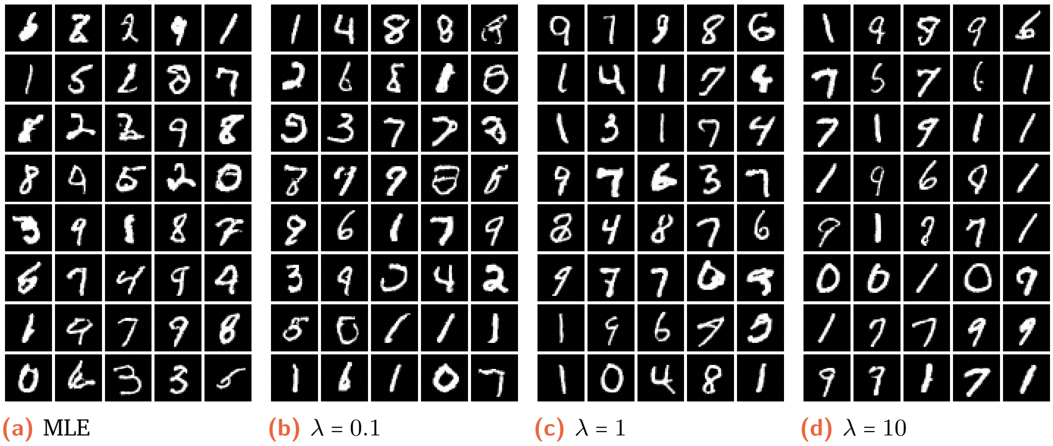


Fig. 5.10.: Samples from a GLOW Normalizing Flow trained with MLE and the PR-Divergence.

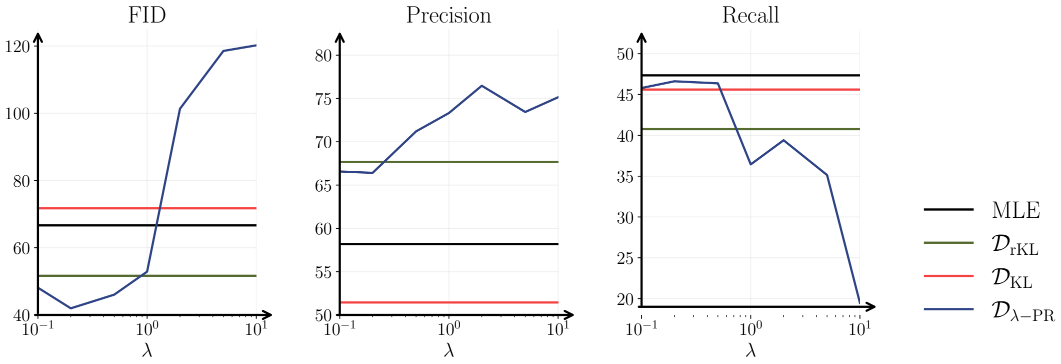
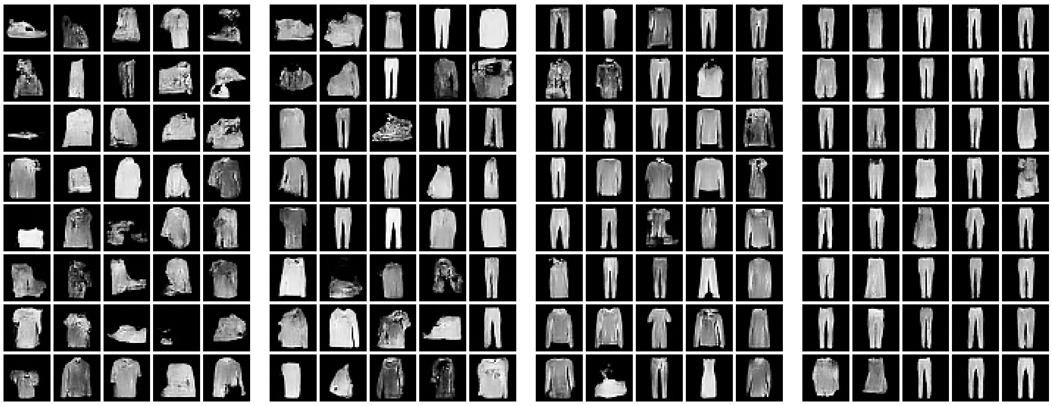


Fig. 5.11.: Quantitative metrics for the GLOW model trained on the Fashion-MNIST dataset. FID (↓), P (↑) and R (↑).

the reverse Kullback-Leibler, while more mode-seeking than the Kullback-Leibler, is still focus on lower values of λ.

Results on Fashion-MNIST The results on the Fashion-MNIST dataset are similar to the results on the MNIST dataset. We show in Figure 5.11 the evolution of the different metrics with respect to the loss function. We observe similar results: (1) models trained with the Kullback-Leibler and the MLE have similar performance, (2) the model trained with the reverse Kullback-Leibler is more focused on Precision and less on Recall than the ones trained with MLE and the Kullback-Leibler, (3) the Precision is increasing with λ and the Recall is decreasing. In Figure 5.12, we show samples from models trained with MLE, $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$. We can see that the MLE-trained model and the $\lambda = 0.1$-trained model produce various samples but of poor quality. However, we can note that the quality of the samples is better for the model trained with $\lambda = 0.1$ than for the model trained with MLE. The model with $\lambda = 10$ produces samples of better quality, but has collapsed to the class of trousers and skirts only.



(a) MLE (b) $\lambda = 0.1$ (c) $\lambda = 1$ (d) $\lambda = 10$

Fig. 5.12.: Samples from a GLOW Normalizing Flow trained with MLE and the PR-Divergence.

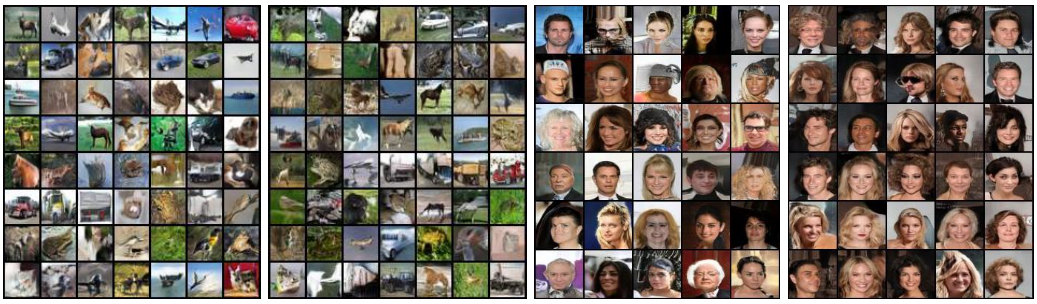
5.3.3 Training and fine-tuning GANs

In this section we train a BigGAN introduced by Brock et al. [16], detailed in Section 2.2.1. First, we show that our approach can be used to train BigGAN models from scratch on dataset such as CIFAR-10 and CelebA. In this experiment, we will focus on understanding the behavior of the model during training and on how λ affects the quality and diversity of the samples. We will also show that our approach can be used to fine-tune BigGAN models on larger datasets such as ImageNet and FFHQ. The models are pre-trained, and we will show that our approach can be used to tune the quality and diversity of the model by fine-tuning the model on the PR-Divergence. We will compare our approach to the traditional truncation methods to tune quality and diversity in generative models.

Training BigGAN: We have fully trained models on CIFAR-10 and on CelebA64 to show that our training algorithm can help to tune models to a specific trade-off between quality and diversity. The final values of each metric mentioned above are given in in Table 5.2. We can see that the Precision and the Density are increasing with λ and the Recall and the Covering are decreasing with λ. We have computed the different metrics for some state-of-the-art models, when available. Furthermore, we can observe some samples in Figure 5.13 generated by the model trained on CIFAR-10 with $\lambda = 0.1$ and $\lambda = 10$ and on CelebA64 with $\lambda = 0.5$ and $\lambda = 5$. We can observe the change in diversity through the difference in the colors. The more diverse models have wider range of main object colors (in CIFAR-10) or background colors

Model	CIFAR-10 32×32					CelebA 64×64				
	FID	P	R	D	C	FID	P	R	D	C
Baseline BigGAN	13.37	86.51	65.66	0.76	0.81	9.16	78.41	**51.42**	0.89	0.48
$\lambda = 0.05$	13.29	81.10	70.63	0.61	0.80	-	-	-	-	-
$\lambda = 0.1$	**11.62**	81.78	**74.58**	0.66	**0.83**	-	-	-	-	-
$\lambda = 0.2$	13.36	84.85	65.13	0.74	0.82	8.79	83.37	44.07	1.09	**0.54**
$\lambda = 0.5$	14.50	83.27	68.23	0.70	0.81	**6.03**	77.60	**55.98**	0.88	0.50
$\lambda = 1.0$	14.03	83.04	69.35	0.68	0.79	13.07	81.70	36.85	1.00	0.47
$\lambda = 2.0$	16.94	84.93	59.79	0.75	0.78	14.23	82.98	32.87	1.16	0.49
$\lambda = 5.0$	32.54	83.39	56.94	0.68	0.73	22.45	**83.96**	25.81	**1.21**	0.43
$\lambda = 10.0$	39.69	84.11	39.29	0.75	0.67	-	-	-	-	-
$\lambda = 20.0$	67.03	**90.03**	21.81	**0.98**	0.56	-	-	-	-	-
DenseFlow [45]	–	88.90	60.81	0.86	0.71	–	85.83	38.22	1.17	0.82
ADM-IP [89]	3.25	80.67	83.65	0.65	0.87	1.53*	23.42	64.48	0.09	0.24
EDM G++ [66]	1.77*	78.48	85.83	0.60	0.87	-	-	-	-	-
StyleGAN-xl [105]	1.85	85.11	70.04	0.75	0.85	-	-	-	-	-

Tab. 5.2.: BigGAN trained with the vanilla approach [16] and with a variety of λ using our approach on CIFAR-10 and CelebA64. We compare our approach with hard truncation on the baseline model. FID (↓), Precision (↑), Recall (↑), Density (↑) and Coverage (↑) are reported. In **bold**, our best model is highlighted and the state-of-the-art FID is marked with an exponent *.



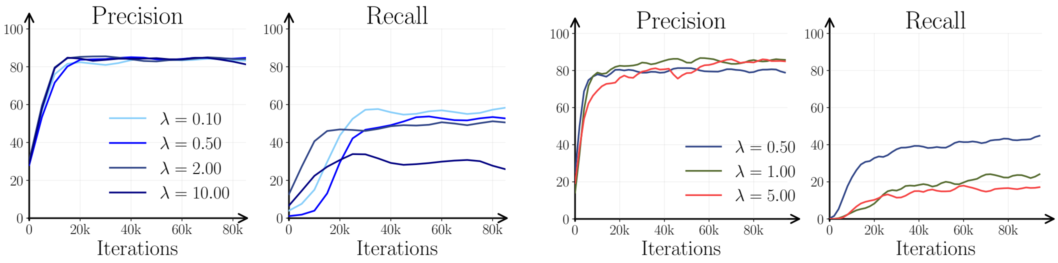
(a) CIFAR-10: $\lambda = 0.1$ (b) CIFAR-10: $\lambda = 10$ (c) CelebA64: $\lambda = 0.5$ (d) CelebA64: $\lambda = 5$

Fig. 5.13.: Samples from BigGAN trained with the PR-Divergence on CIFAR-10 and CelebA64.

(in CelebA64). The more quality-focused models have fewer artifacts rendering the objects or the faces.

In Figure 5.14, we can observe how Precision and Recall evolve during training. We can see how the Precision converges to its final value in approximately 10k iterations. Even if the final value of Precision across different λ is different, the variance is low compared to the difference of Recall. For both CIFAR-10 and CelebA64 we can see that the Recall is increasing slower than Precision and reaches a plateau between 15k and 30k iterations for the model trained on CIFAR-10. On CelebA the model trained with $\lambda = 0.5$ for instance, reaches a Precision plateau at 15k iterations, but the Recall is still slowly increasing until 80k iterations. This experiment shows that GANs will first focus on Precision and then on Recall.

Fine-tuning BigGAN: Considering that training GANs models on high complexity and high resolution datasets such as ImageNet and FFHQ is computationally expensive, we show that our approach can be used to fine-tune pre-trained models on these datasets. We use the BigGAN model pre-trained on the ImageNet dataset and the FFHQ dataset. We fine-tune the models on the PR-Divergence with various λ. Our results are shown in Table 5.3. We can see that our approach can be used



(a) CIFAR-10 (b) CelebA64

Fig. 5.14.: Evolution of the Precision and Recall during training on CIFAR-10 and CelebA64.

Model	ImageNet 128 × 128					FFHQ 256 × 256				
	FID	P	R	D	C	FID	P	R	D	C
Baseline BigGAN	9.83	28.04	41.21	0.14	0.17	41.41	65.57	**10.17**	0.52	0.47
Soft $\psi = 0.7$	11.39	23.04	31.13	0.11	0.15	56.43	76.59	4.87	0.70	0.41
Soft $\psi = 0.5$	15.49	20.20	19.83	0.10	0.14	82.05	84.48	1.58	0.89	0.32
Hard $\psi = 2.0$	**9.69**	25.83	39.89	0.13	**0.18**	43.32	68.84	8.66	0.58	0.47
Hard $\psi = 1.0$	12.12	21.86	35.42	0.11	0.15	56.19	76.44	4.76	0.75	0.44
Hard $\psi = 0.5$	15.21	21.13	29.55	0.10	0.13	71.32	80.99	4.84	0.84	0.36
$\lambda = 0.2$	9.92	26.69	42.04	0.13	0.17	35.66	78.70	9.45	0.88	0.60
$\lambda = 0.5$	10.82	26.83	**42.38**	0.13	0.16	**35.24**	78.41	9.66	0.89	0.60
$\lambda = 1.0$	20.42	29.72	28.21	**0.15**	0.15	35.91	78.95	8.32	0.90	0.57
$\lambda = 2.0$	20.21	30.27	30.49	0.14	0.14	36.33	81.10	8.69	1.05	**0.64**
$\lambda = 5.0$	20.76	**30.87**	28.38	**0.15**	0.15	38.16	**84.31**	8.52	**1.15**	0.63
ADM [53]	2.97	26.63	68.54	0.14	0.16	-	-	-	-	-
StyleGAN-xl [105]	1.81^*	11.35	68.04	0.04	0.09	2.19^*	79.91	38.79	0.86	0.73

Tab. 5.3.: BigGAN fine-tune with the vanilla approach [16] and with a variety of λ using our approach on ImageNet128 and FFHQ256. We compare our approach with hard truncation on the baseline model. FID (↓), Precision (↑), Recall (↑), Density (↑) and Coverage (↑) are reported. In **bold**, our best model is highlighted and the state-of-the-art FID is marked with an exponent *.

to tune Precision and Recall. Furthermore, we compare our results with another method traditionally used to tune quality and diversity of pre-trained models: Hard Truncation or Soft Truncation detailed in Section 2.1.4. We compare our results with truncation because it is widely used in the literature, but our method can be used together with truncation to further improve the quality and diversity of the samples. We can see that our approach outperforms the traditional truncation methods in terms of Precision and Recall.

5.4 Concluding Remarks and Discussion

In this chapter, we addressed one question concerning Precision and Recall in generative models :

- **Question Q3**: *Can we train a generative model to directly focus on an explicit user-specified trade-off between Precision and Recall?* We have shown, that the PR-Divergence, which corresponds to the trade-off between Precision and Recall defined by Sajjadi et al. [103], can be used to train generative models to focus on a specific trade-off between quality and diversity. We have shown that our approach can be used to train models on synthetic data, low-complexity datasets, and high-complexity datasets. Furthermore, we have also shown that our approach can be used to fine-tune pre-trained models on large datasets.

This theoretical analysis could be further improved, and we list several potential future works:

- **Optimizing the AUC:** In this section, we have investigated a method to optimize a model for any point on the PR-Curve. However, we can wonder if it is possible to optimize the area under of the PR-Curve (AUC). The AUC of the PR-Curve is computed as follows:

$$\mathrm{AUC} = \int_0^{+\infty} \frac{\alpha_\lambda(P \| \widehat{P})^2}{\lambda^2} \mathrm{d}\lambda. \tag{5.34}$$

  Therefore, one possible approach is to optimize the model to minimize a weighted sum of PR-Divergence to optimize an approximation of the AUC.

- **Trading-off Precision and Recall in Diffusion Models:** Under specific conditions, score-matching diffusion models can be trained to minimize any f-divergences using a density ratio estimator. It would be interesting to investigate if PR-divergence can be used to train diffusion models to focus on a specific trade-off between quality and diversity, as it is as of today the state-of-the-art method for generative models.

- **Other methods to train the PR-Divergence:** In this chapter, we have shown that the PR-Divergence can be trained using a discriminator learned to estimate the χ^2 Divergence. It would be interesting to investigate other methods to train the PR-Divergence: for instance, using the naive approach with decreasing regularization to train the discriminator, or using another tractable divergence estimation method to train the generator.

6 Optimal Budgeted Rejection Sampling to improve Precision and Recall

> *I have not failed. I've just found 10,000 ways that won't work.*
>
> — **Thomas Edison**
> (Founder of the first industrial laboratory.)

Contents

Question 4: *With rejection sampling under limited budget, how much can we increase Precision and Recall of a pre-trained model?*

In the previous chapters, we have seen that we could change the f-divergence minimized by a generative model during training to tune Precision and Recall. However, we have considered only the loss function for now. And, to train a generative model to minimize any f-divergence, we need to train a discriminator to estimate the value of the divergence. At the end of the training, the discriminator is typically discarded and the generator is used to generate new samples. However, the discriminator can be used to estimate the density ratio between the data distribution and the learned distribution. This density ratio can be used to improve the quality of the generated samples. A naive approach to use the density ratio to perform rejection sampling to generate new samples. The issue with this approach is that it can be computationally expensive, especially in high dimensions: it can require thousands (or millions) of inferences on the mapping function G to sample a single point. In this chapter, we will focus on the sampling method used to generate new samples, and see how it can improve Precision and Recall even with a limited computation budget.

We will first review existing sampling methods that extend the traditional framework of generative models. In particular, we will recall Rejection Sampling in Section 6.1.1 and explain how it can be used to improve generative models in Section 6.1.2. In addition, we briefly introduce comparable methods in Section 6.1.3. Then, to address Question **Q4**, we will introduce a new sampling method, the Optimal Budgeted Rejection Sampling (OBRS) in Section 6.2.1. We will show that this method is optimal to minimize any f-divergence with a limited budget and especially that it can improve Precision *and* Recall both theoretically and experimentally in Sections 6.2.2 and 6.2.3. Finally, we will show that training generative models to account for rejection sampling can further improve the quality and the diversity of the generated samples in Section 6.3.

Contributions: The contributions of this chapter are the following:

- We propose a new sampling method, the Optimal Budgeted Rejection Sampling, that is optimal to minimize any f-divergence under a fixed budget.
- We train generative models to account for the rejection sampling, and show that it improves (1) the convergence of the training, and (2) the f-divergence between the data distribution and the learned distribution after rejection.

These results have been published as:

- Alexandre Verine et al. "Optimal Budgeted Rejection Sampling for Generative Models". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics* (Mar. 2024). arXiv:2311.00460 [cs]

6.1 Rejection for Generative Models

The aim of this chapter is to understand how we can improve rejection sampling to improve generative models and especially how we can do it with limited computational resources. We will first introduce the Rejection Sampling method and explain how it is used in practice. The seminal work of Azadi et al. [9] has led to several other sampling methods, which we will briefly introduce.

6.1.1 Rejection Sampling

Rejection Sampling is a classical method to sample from a target distribution using a proposal distribution and the ratio of density between both distributions. The idea was introduced by Von Neuman [129] and has several names in the literature such as the Acceptance-Rejection Sampling or Screening Sampling.

In the context of the thesis, we consider a target distribution P with density p and the proposal is the learned distribution $\widehat{P}$ with density $\widehat{p}$. The idea behind rejection sampling is to accept or reject samples from $\widehat{P}$ using an acceptance function $a : \mathcal{X} \to [0,1]$ such that the probability of accepting a sample $\boldsymbol{x}$ from $\widehat{P}$ is $a(\boldsymbol{x})$. We detail the sampling procedure in Algorithm 4. The distribution induced by the rejection procedure based on a is a new distribution in $\mathcal{P}(\mathcal{X})$ denoted $\widetilde{P}_a$. The density $\widetilde{p}_a(\boldsymbol{x})$ of $\widetilde{P}_a$ has the following form:

$$\widetilde{p}_a(\boldsymbol{x}) = \frac{\widehat{p}(\boldsymbol{x})a(\boldsymbol{x})}{Z}, \tag{6.1}$$

where $Z > 0$ is a normalizing constant that ensures that $\int_{\mathcal{X}} \widetilde{p}_a(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}) = 1$. The normalizing constant plays a crucial role in Rejection Sampling as it is the overall acceptance rate:

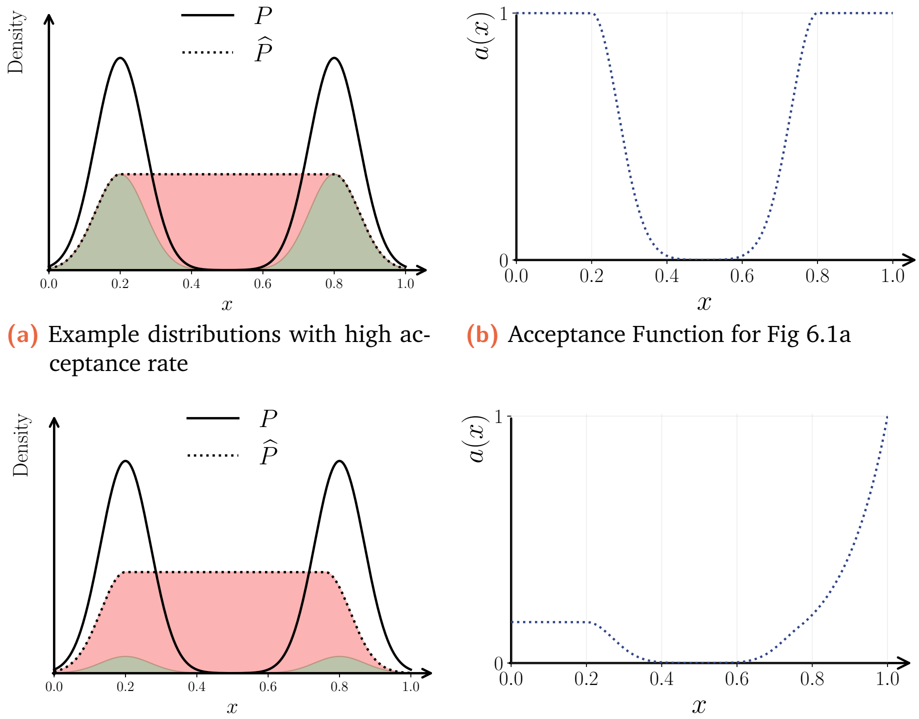
$$\mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = Z. \tag{6.2}$$

In other terms, one needs to draw $1/Z$ samples from $\widehat{P}$ to have on average one sample accepted. We present a general formulation of rejection sampling, however in the literature, this method is generally defined for a fixed acceptance function:

$$a_{\mathrm{RS}}(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})M}, \tag{6.3}$$

Algorithm 4 Rejection Sampling

```
repeat
    Sample $\boldsymbol{x}$ from $\widehat{P}$
    Sample $u \sim \mathcal{U}([0,1])$
    if $u \leq a(\boldsymbol{x})$ then
        Accept $\boldsymbol{x}$.
    end if
until $N$ samples are accepted.
```



(a) Example distributions with high acceptance rate

(b) Acceptance Function for Fig 6.1a

(c) Example distributions with low acceptance rate

(d) Acceptance Function for Fig 6.1c

Fig. 6.1.: Example of Rejection Sampling in 1D. The target distribution P is a Gaussian mixture, and the proposal distribution in dashed line in a distribution covering both modes. In Figure 6.1a, $\widehat{P}$ covers both modes of P while in Figure 6.1c, $\widehat{P}$ is slightly narrower on the right. The maximum ratio density M is higher in Figure 6.1c leading to a lower acceptance rate: from 45% to 8%. The acceptance rate is represented as the ratio of the green area on the area under $\widehat{p}$.

where $M = \sup_{x \in \mathcal{X}} \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$ is the maximum density ratio. In this case, the acceptance rate is $Z = 1/M$ and the distribution $\widetilde{P}_{a_{\mathrm{RS}}}$ perfectly matches the target distribution P since:

$$\widetilde{p}_{a_{\mathrm{RS}}}(\boldsymbol{x}) = \widehat{p}(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x}) Z M} = p(\boldsymbol{x}). \tag{6.4}$$

However, for high-dimensional $\mathcal{X}$, M can take high values and set a very low acceptance rate as stated by MacKay [80]. We give an example in Figure 6.1 to illustrate this point. In this example, we consider a target distribution P that is a Gaussian mixture and the proposal distribution $\widehat{P}$ is a distribution that covers both modes of P. In Figure 6.1a, the acceptance rate is approximately 45% while in Figure 6.1c, the acceptance rate is lower than 8%, even if the distribution $\widehat{P}$ are very similar. In high dimension and especially with GANs, it is not uncommon that the learned distribution $\widehat{P}$ misses modes from the target distribution P, leading to a low acceptance rate. In this case, rejection sampling can be very inefficient as it requires drawing many samples from $\widehat{P}$ to have one accepted, as we will see in the next section.

6.1.2 Discriminator Rejection Sampling

In the seminal work of Azadi et al. [9], the authors propose to use the discriminator of a GAN to further refine the generation process. As we have shown in the previous chapters, at convergence the discriminator T can be used to estimate the density ratio between the data distribution and the learned distribution:

$$r(\boldsymbol{x}) = \nabla f^*(T(\boldsymbol{x})). \tag{6.5}$$

In fact, if the discriminator is optimal, then $r^{\text{opt}}(\boldsymbol{x}) = p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. The idea of the Discriminator Rejection Sampling (DRS) is to use the discriminator to estimate the acceptance function a in Equation (6.3) and to refine the learned distribution $\widehat{P}$ to match the target distribution P. The acceptance function is defined as:

$$a(\boldsymbol{x}) = \frac{r(\boldsymbol{x})}{M}, \tag{6.6}$$

where $M = \sup_{x \in \mathcal{X}} r(\boldsymbol{x})$ is the maximum estimated density ratio. For a BigGAN model trained on CelebA, the acceptance rate can be as low as 10^{-6}. For that reason, the authors propose a trick to increase the acceptance rate by scaling the acceptance function by a factor $\gamma \in \mathbb{R}$:

$$a_{\text{DRS}}(\boldsymbol{x}) = \frac{r(\boldsymbol{x})}{r(\boldsymbol{x})\,(1 - e^{\gamma}) + Me^{\gamma}}. \tag{6.7}$$

With this acceptance function, the values $\gamma < 0$ increase the acceptance rate. However, the authors do not provide any theoretical guarantees on the choice of γ, the effect it has on how $\widetilde{P}_{a_{\text{DRS}}}$ differs from P. We illustrate the effect of γ in Figure 6.2 by manually setting γ to -1.7 and -3.6 to enforce an acceptance rate of 25% and 50%.

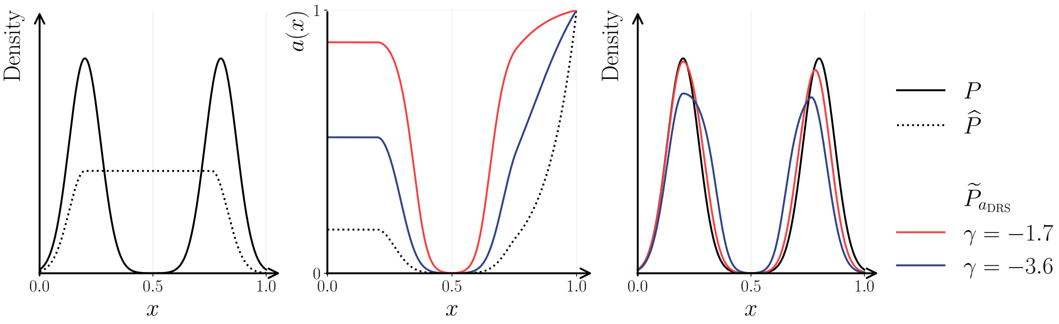


Fig. 6.2.: Example of Discriminator Rejection Sampling in 1D. The target distribution P is a Gaussian mixture, and the proposal distribution in dashed line in a distribution covering both modes. $\gamma = -1.7$ enforces an acceptance rate of 25% while $\gamma = -3.6$ enforces an acceptance rate of 50%.

6.1.3 Other sampling methods

The work Azadi et al. [9] has led to several other sampling methods that we will briefly introduce in this section. Since this method is based on rejection sampling, which is a standard but also simplistic method, other more complex methods have been proposed to use the density ratio to improve sampling:

- MH-GAN by Turner et al. [123] uses a specific version of the Metropolis-Hastings algorithm to sample from the learned distribution: independent Metropolis-Hastings. The idea is to draw a first sample $\boldsymbol{x}_0$ from the proposal distribution $\widehat{P}$. Then, we successively draw K new samples $\boldsymbol{x}_k \sim \widehat{P}$ and accept it with probability:

  $$\min\left(1, \frac{r(\boldsymbol{x}_k)}{r(\boldsymbol{x}_{k-1})}\right). \tag{6.8}$$

  This method was shown to improve the quality of the generated samples. In fact, contrary to DRS, MH-GAN compares the density ratio of K samples to keep only the best. Moreover, the method directly sets the computational budget by setting K, and thus using NK inferences on the generator G and NK inferences on the discriminator T to generate N samples.

- Discriminator Optimal Transport (DOT) by Tanaka [117] consists of using the discriminator to estimate the optimal transport between the learned distribution and the target distribution. In practice, they can draw N samples from the learned distribution and perform a gradient descent using $\nabla_{\boldsymbol{x}} T(\boldsymbol{x})$ to move the samples to the target distribution. This method was shown to improve the quality of the generated samples. This method only requires N inferences on the generator G to generate N samples, but it requires to compute the gradient of the discriminator T numerous times.

- Discriminator Gradient Flow (DG$f$low) by Ansari et al. [7] is also a method that uses the discriminator to perform a gradient flow in the latent space. To do this, they draw N latent vectors $\boldsymbol{z} \sim Q$ and progressively move them using $\nabla_{\boldsymbol{z}}(T(G(\boldsymbol{z})))$ to improve the f-divergence between the target distribution and the one generated by mapping the N latent vectors. This method is computationally expensive as it requires the computation of the gradient of $T \circ G$ numerous times (n_{ite}) for the N samples.

These methods are more complex than DRS and have been shown to improve the quality of the generated samples. However, a larger number of inferences are required in the generator G and the discriminator T to generate N samples. Similarly to DRS, these methods can set the computational budget by setting the number of iterations. Note that in DRS, the acceptance rate is the expected budget per sample, therefore some samples will have a higher computational budget than others, while MH-GAN, DOT and DG$f$low set the same budget for all samples. On a two-dimensional experiment, we will later show that theses different methods outperform DRS in terms of quality of the generated samples but at the cost of a higher computational budget. For limited computational resources, DRS is drastically better than the other methods, as we will show in Section 6.2.3.

In the next section, we will introduce a new method, based on Rejection Sampling, that is optimal to minimize any f-divergence under a acceptance rate, i.e. a fixed budget on average.

6.2 The Optimal Budgeted Rejection Sampling (OBRS)

In this section, we study the problem of Rejection Sampling with a limited budget $K \in [1, +\infty[$. K is the expected number of samples to draw from the proposal distribution $\widehat{P}$ to draw a single sample from $\widetilde{P}_a$. We will start by introducing a method to find the optimal acceptance function to minimize any f-divergence under a fixed budget. Then, we will show that this method can the f-divergence between P and the refined distribution $\widetilde{P}_a$. Finally, we will characterize the improvements on the Precision and the Recall of a generative model.

6.2.1 Optimal acceptance function

Given a fixed distribution $\widehat{P}$, set by a function G, and a target distribution P, we aim to find the optimal acceptance function a to minimize any f-divergence under a fixed budget K, as follows:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(P \| \widetilde{P}_a) \\ \text{s.t.} \quad & \begin{cases} \mathbb{E}_{\widehat{P}}[a(\boldsymbol{x})] \geq 1/K, \\ \forall \boldsymbol{x} \in \mathcal{X}, \, 0 \leq a(\boldsymbol{x}) \leq 1. \end{cases} \end{aligned} \tag{6.9}$$

Here, the constraint $\mathbb{E}_{\widehat{P}}[a(\boldsymbol{x})] \geq 1/K$ is used to bound the expected acceptance rate. For $K = 1$, the only a that satisfies the constraints in (6.9) is the unit function $a(\boldsymbol{x}) = 1 \ \forall \ \boldsymbol{x} \in \mathcal{X}$. This case corresponds to no rejection (or accept with probability 1), and we have $\widetilde{P}_a = \widehat{P}$ almost everywhere. On the other hand, if $K \geq M$ where $M = \sup_{x \in \mathcal{X}} \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$, the optimal acceptance function is the one defined in (6.3), and we have $\widetilde{P}_a = P$. In this case, the expected budget is M.

While the problem (6.9) is a convex optimization problem, the optimal acceptance function a_{OBRS} is not straightforward to compute. We can however focus on the discrete case. In the discrete case, we search for the optimal acceptance vector $\boldsymbol{a} \in \mathbb{R}^N$ for a set of N samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ drawn from the proposal distribution $\widehat{P}$. In this case, the problem (6.9) can be written as:

$$\begin{aligned} \min_{\boldsymbol{a} \in \mathbb{R}^N} \quad & \sum_i^N K a_i \widehat{p}_i f\left(\frac{p_i}{a_i \widehat{p}_i K}\right) \\ \text{s.t.} \quad & \begin{cases} \sum_i^N \widehat{p}_i a_i = 1/K \\ \forall i,\ 0 \leq a_i \leq 1 \end{cases} \end{aligned} \tag{6.10}$$

In that case, the objective function is continuous with respect to $\boldsymbol{a}$ and the constraint set for $\boldsymbol{a}$ is closed and bounded, therefore according to the Weierstrass theorem, there exist a solution to this problem. In the following theorem, we give an explicit form for the optimal solution a_{OBRS} for finite $\mathcal{X}$ using Lagrangian duality:

Theorem 6.2.1 (Optimal Acceptance Function)**.**
For a sampling budget $1 \leq K \leq M$ and finite $\mathcal{X}$, the solution to the problem (6.9) is,

$$a_{\mathrm{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \frac{c_K}{M}, 1\right), \tag{6.11}$$

where $c_K \geq 1$ is such that $\mathbb{E}_{\boldsymbol{x} \sim \widehat{p}}[a_{\mathrm{OBRS}}(\boldsymbol{x})] = 1/K$.

Proof. The full proof is given in Appendix B. However, we can give an intuition of the proof. First, without loss of generality, we can consider $\mathcal{D}_f(\widetilde{P}_a \| P)$ instead of $\mathcal{D}_f(P \| \widetilde{P}_a)$ by considering $f'(u) = u f(1/u)$. We consider that K is typically lower than M, so we simplify the problem to $\mathbb{E}_{\widehat{p}}[a(\boldsymbol{x})] = 1/K$. Using the definition of the refined density $\widetilde{p}_a$ in (6.1), we can rewrite the objective in the discrete case as:

$$\begin{aligned} \min_{\boldsymbol{a} \in \mathbb{R}^N} \quad & \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) \\ \text{s.t.} \quad & \begin{cases} \sum_i^N \widehat{p}_i a_i = 1/K \\ \forall i,\ 0 \leq a_i \leq 1 \end{cases} \end{aligned} \tag{6.12}$$

The Lagrangian function associated with the problem 6.12 is:

$$\mathcal{L}(\boldsymbol{a}, \mu, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) + \mu\left[\boldsymbol{a}^T \widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T \boldsymbol{\lambda}_1 - \boldsymbol{a}^T \boldsymbol{\lambda}_2 \tag{6.13}$$

Using KKT conditions, we can show that the optimal solution $a_i^\star$ is:

$$a_i^\star = \frac{p_i}{\widehat{p}_i K} \nabla f^* \left(\frac{\lambda_{2i}^\star - \lambda_{1i}^\star}{\widehat{p}_i K} - \mu^\star / K\right) \tag{6.14}$$

For every f-divergence the Fenchel conjugate f^* is strongly increasing on $\mathrm{dom}(f^*)$ therefore the optimal solution is $\lambda^\star_{1i} = 0$ for every i. By using strong duality and the Fenchel conjugate definition, we can show that the solution of problem 6.12 satisfy:

$$a_i^\star = \frac{p_i}{\hat{p}_i K}\nabla f^* \left(\min\left(-\mu^\star/K, \nabla f\left(\frac{\hat{p}_i K}{p_i}\right)\right)\right). \tag{6.15}$$

Using again that f^* is strictly increasing and that $[\nabla f]^{-1} = \nabla f^*$, we have the following.

$$a_i^\star = \min\left(\frac{p_i}{\hat{p}_i K}\nabla f^* \left(-\mu^\star/K\right), 1\right). \tag{6.16}$$

Note that $\nabla f^* \left(-\mu^\star/K\right)/K = c_K$ is a constant solely determined by K, $\boldsymbol{p}$ and $\widehat{\boldsymbol{p}}$ since:

$$\sum_i \widehat{p}_i \min\left(\frac{p_i}{\widehat{p}_i K}\nabla f^* \left(-\mu^\star/K\right), 1\right) = 1/K. \tag{6.17}$$

Therefore, by scaling the constant for easier intuition, the optimal solution is:

$$a_i^\star = \min\left(\frac{p_i}{\hat{p}_i}\frac{c_K}{M}, 1\right), \tag{6.18}$$

which concludes the proof.

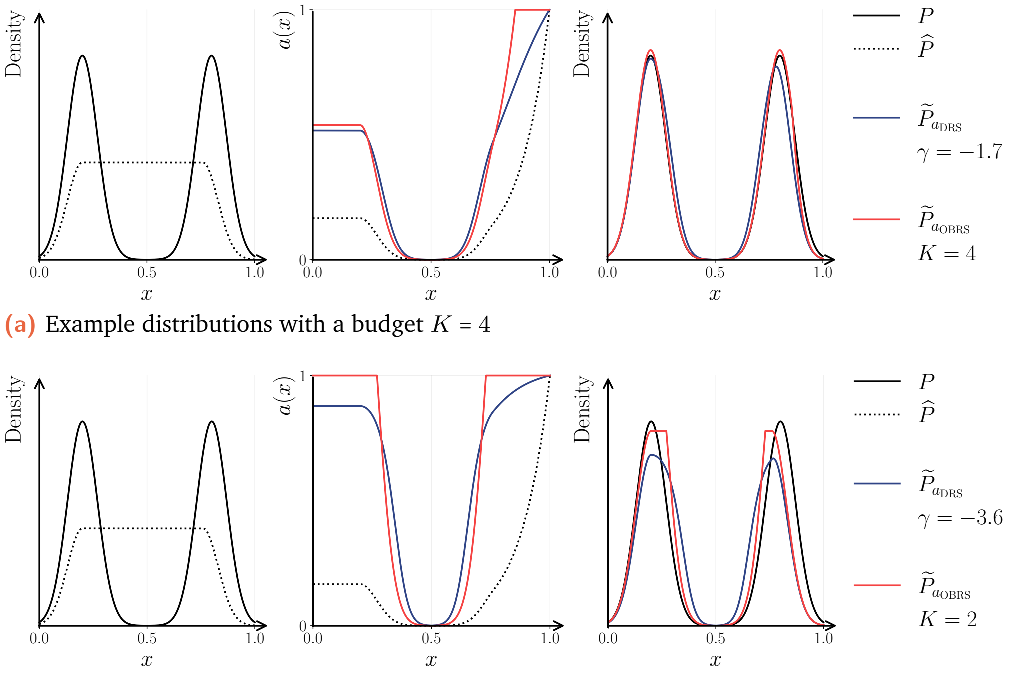
This theorem gives an explicit form for the optimal acceptance function a_{OBRS} to minimize the f-divergence under a fixed budget K. Therefore, by performing Rejection Sampling using the optimal acceptance function a_{OBRS}, we can generate a distribution $\widetilde{P}_{a_{\mathrm{OBRS}}}$ that is optimal in terms of f-divergence under the budget K. We call this method the Optimal Budgeted Rejection Sampling (OBRS). We can, however, make a few remarks on the optimal acceptance function:

- The acceptance function

  $$a_{\mathrm{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c_K}{M}, 1\right),$$

  The form of the optimal acceptance is a clipped version of the optimal acceptance function in Equation (6.3). A very similar acceptance function was previously introduced by Grover et al. [48], with the sole argument that it is a "natural" approximation of the optimal acceptance function, but no theoretical argument was provided.

- We show that this acceptance function is indeed the optimal solution to problem (6.9) for a specific constant c_K. However, there is no explicit form for the constant in the theorem. This constant is solely determined by the budget K and the distributions P and $\widehat{P}$. In practice, we can estimate c_K by using a dichotomy method (also known as a bisection algorithm) to find the value



(a) Example distributions with a budget $K = 4$

(b) Example distributions with a budget $K = 2$

Fig. 6.3.: Example of Discriminator Rejection Sampling and Optimal Budgeted Rejection Sampling in 1D. The target distribution P is a Gaussian mixture, and the proposal distribution in dashed line in a distribution covering both modes. The acceptance function a_{OBRS} is computed using Algorithm 5 and the acceptance function a_{DRS} is computed using $\gamma = -1.7$ and $\gamma = -3.6$ to enforce an acceptance rate of 25% and 50%.

as detailed in Algorithm 5. However, a budget greater than $M = \sup_{x \in \mathcal{X}} \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}$ (unbudgeted sampling) implies that $c_K = 1$, and thus

$$a_{\mathrm{OBRS}}(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{M\widehat{p}(\boldsymbol{x})}. \tag{6.19}$$

- As we have said, c_K is solely determined by the budget K, and therefore the optimal acceptance function a_{OBRS} is independent of the function f. The OBRS method is optimal for *any* f-divergence. In other terms, whether we use a mass-covering divergence such as the Kullback-Leibler divergence or a mode-seeking divergence such as the reverse Kullback-Leibler or any PR-Divergence, the optimal acceptance function is the same. We will see in the next section how the OBRS method can improve the Precision and the Recall.

We show in Figure 6.3 how the optimal acceptance function a_{OBRS} compares to the acceptance function a_{DRS} for the examples given in Figure 6.2, therefore for budgets of $K = 4$ and $K = 2$. We can see that the acceptance function a_{OBRS} is a clipped version of a_{RS} that leads to a less smooth acceptance function than a_{DRS}. However, the refined distribution $\widetilde{P}_{a_{\mathrm{OBRS}}}$ appears to be closer to the target distribution P than

Algorithm 5 Dichotomy to compute c_K.

```
Input: N generated samples $\boldsymbol{x}_1^{\text{fake}}, \dots, \boldsymbol{x}_N^{\text{fake}} \sim \widehat{P}$, the density ratio function $r$ and $M = \sup_{\boldsymbol{x} \in \mathcal{X}} (r(\boldsymbol{x}))$.
Parameter: Budget $K$, Threshold $\epsilon$
Output: Constant $c_K$
  Let $c_{\min} = 1e^{-10}$ and $c_{\max} = 1e^{10}$.
  Let $c_K = (c_{\max} + c_{\min})/2$
  Define the loss $\mathcal{L}(c_K) = \sum_{i=1}^{N} \min\left(r\left(\boldsymbol{x}_i^{\text{fake}}\right) c_K / M, 1\right) - \frac{1}{K}$.
  while $|\mathcal{L}(c_K)| \geq \epsilon$ do
      if $\mathcal{L}(c_K) > \epsilon$ then
          Update: $c_{\max} = c_K$
      else if $\mathcal{L}(c_K) < -\epsilon$ then
          Update: $c_{\min} = c_K$
      end if
      Update: $c_K = (c_{\max} + c_{\min})/2$
      Update: $\mathcal{L}(c_K)$
  end while
```

$\widetilde{P}_{a_{\text{DRS}}}$. In Figure 6.4 we compare some f-divergences between the target distribution P and the refined distribution $\widetilde{P}_{a_{\text{OBRS}}}$ and $\widetilde{P}_{a_{\text{DRS}}}$ for different budgets. We can see that the f-divergence between P and $\widetilde{P}_{a_{\text{OBRS}}}$ is lower than the f-divergence between P and $\widetilde{P}_{a_{\text{DRS}}}$ for all budgets. This result is consistent with the fact that the OBRS method is optimal to minimize any f-divergence under a fixed budget. We will now show how the OBRS method can improve the Precision and the Recall of a generative model.

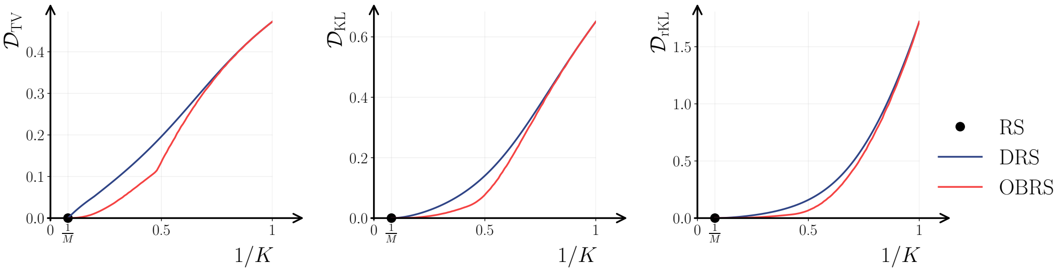


Fig. 6.4.: $\mathcal{D}_{\text{TV}}$, $\mathcal{D}_{\text{KL}}$ and $\mathcal{D}_{\text{rKL}}$ between the target distribution P and the refined distribution $\widetilde{P}_{a_{\text{OBRS}}}$ and $\widetilde{P}_{a_{\text{DRS}}}$ for different budgets. The f-divergence is systematically lower for $\widetilde{P}_{a_{\text{OBRS}}}$ than for $\widetilde{P}_{a_{\text{DRS}}}$.

6.2.2 Improving the Precision and the Recall

As we have seen, the OBRS method is optimal to minimize any f-divergence under a fixed budget. In this section, we first characterize how the budget affects the f-divergence, by giving a bound on the improvement of the f-divergence between the target distribution P and the refined distribution $\widetilde{P}_{a_{\text{OBRS}}}$ compared to the f-divergence between P and the learned distribution $\widehat{P}$. The first bound being general, it is not very tight. However, we will show that we can compute the exact improvement on the Precision and the Recall of a generative model.

We first give a bound on the improvement of the f-divergence between the target distribution P and the refined distribution $\widetilde{P}_{a_{\text{OBRS}}}$ compared to the f-divergence between P and the learned distribution $\widehat{P}$:

Theorem 6.2.2 (f-divergence Improvement)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions such that $P, \widehat{P} \ll \mu$ and a_{OBRS} be the optimal acceptance function for a budget K defined in Theorem 6.2.1. For any f-divergence, we have

$$\mathcal{D}_f\left(P \| \widetilde{P}_{a_{\text{OBRS}}}\right) \leq \mathcal{D}_f\left(P \| \widehat{P}\right) - \min\left(1, \frac{K-1}{M}\right) \mathcal{D}_f\left(P \| \widehat{P}\right). \tag{6.20}$$

Proof. We note $\widetilde{P} = \widetilde{P}_{a_{\text{OBRS}}}$ for simplicity. For any density p_γ such that $p_\gamma \leq K\widehat{p}$:

$$\mathcal{D}_f(P \| \widetilde{P}) \leq \mathcal{D}_f(P \| P_\gamma). \tag{6.21}$$

For bounding general f-divergences, we will choose $p_\gamma = \widehat{p} + \gamma(p - \widehat{p})$ with $\gamma = \min\left(1, (K-1)\inf_{\boldsymbol{x}' \in \mathcal{X}} \frac{\widehat{p}(\boldsymbol{x}')}{p(\boldsymbol{x}')}\right)$. Let us first show that $p_\gamma \leq K\widehat{p}$:

$$p_\gamma(\boldsymbol{x}) \leq \widehat{p}(\boldsymbol{x}) + (K-1)\inf_{\boldsymbol{x}'} \frac{\widehat{p}(\boldsymbol{x}')}{p(\boldsymbol{x}')}\left(p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})\right). \tag{6.22}$$

Note that for any $\boldsymbol{x} \in \mathcal{X}$, $\inf_{\boldsymbol{x}'} \frac{\widehat{p}(\boldsymbol{x}')}{p(\boldsymbol{x}')}\left(p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})\right) \leq \widehat{p}(\boldsymbol{x})$. Thus, we have the following.

$$p_\gamma(\boldsymbol{x}) \leq \widehat{p}(\boldsymbol{x}) + (K-1)\widehat{p} \leq K\widehat{p}(\boldsymbol{x}) \tag{6.23}$$

Next, let us show the lower bound. Recall that f-divergences are jointly convex, therefore $\mathcal{D}_f(p, \cdot)$ is convex. Thus, convexity implies:

$$\mathcal{D}_f(P \| P_\gamma) \leq (1-\gamma)\mathcal{D}_f(P \| \widehat{P}) + \gamma\mathcal{D}_f(P \| P) \leq (1-\gamma)\mathcal{D}_f(P \| \widehat{P}). \tag{6.24}$$

Using (6.21) and (6.24), we have the following.

$$\mathcal{D}_f(P \| \widetilde{P}) \leq \mathcal{D}_f(P \| P_\gamma) \leq (1-\gamma)\mathcal{D}_f(P \| \widehat{P}) \leq \mathcal{D}_f(P \| \widehat{P}) - \gamma\mathcal{D}_f(P \| \widehat{P}) \tag{6.25}$$

$$\leq \mathcal{D}_f(P \| \widehat{P}) - \min\left(1, \frac{K-1}{M}\right)\mathcal{D}_f(P \| \widehat{P}). \tag{6.26}$$

Theorem 6.2.2 gives a general bound on the improvement of the f-divergence between the target distribution P and the refined distribution $\widetilde{P}_{a_{\text{OBRS}}}$ compared to

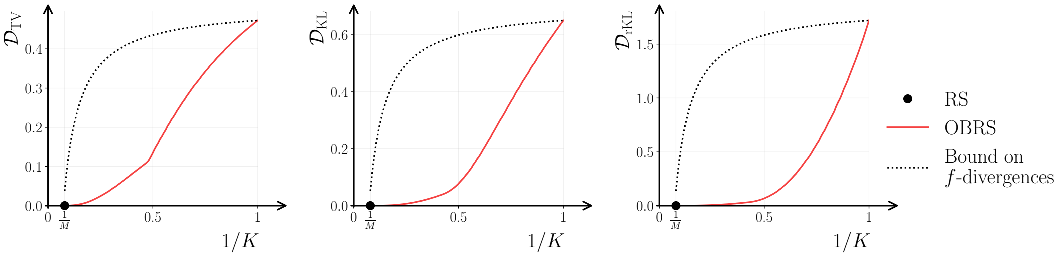


Fig. 6.5.: Bounds on the Total Variation, the Kullback-Leibler and the Reverse Kullback-Leibler divergence between the target distribution P and the refined distribution $\widetilde{P}_{a_{\mathrm{OBRS}}}$ for different budgets for the example given in Figure 6.3b.

the f-divergence between P and the learned distribution $\widehat{P}$. By being general, this bound can be far from the actual improvement. In Figure 6.5, we show the bounds on the f-divergence between the target distribution P and the refined distribution $\widetilde{P}_{a_{\mathrm{OBRS}}}$ for different budgets. We can see that for most budget, the bound on the improvement is not tight and the actual improvement is higher. We will now show that we can compute the improvement on the Precision and the Recall of a generative model:

Theorem 6.2.3 (Precision and Recall Improvement)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions such that $P, \widehat{P} \ll \mu$ and a_{OBRS} be the optimal acceptance function for a budget K defined in Theorem 6.2.1. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$, we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.

Proof. First, with $a(\boldsymbol{x}) = \min\left(1, \frac{c_k}{M}\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)$, thus:

$$\alpha_\lambda(P\|\widetilde{P}_a) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widetilde{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{6.27}$$

$$= \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M} p(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}). \tag{6.28}$$

We can show that we have two regimes:

- For $\lambda \geq \frac{Kc_K}{M}$:

$$\alpha_\lambda\left(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}\right) = 1 \quad \text{and} \quad \beta_\lambda\left(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}\right) = 1/\lambda$$

- For $\lambda \leq \frac{Kc_K}{M}$:

$$\alpha_\lambda(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}) = K\alpha_{\lambda/K}(P\|\widehat{P}) \quad \text{and} \quad \beta_\lambda(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}) = \beta_{\lambda/K}(P\|\widehat{P})$$

This can be seen as a vertical scaling of the PR-Curve. For a given point (α, β) in $\mathrm{PRD}(P\|\widehat{P})$, then the point with the same β in $\mathrm{PRD}(P\|\widetilde{P})$ has a Precision $K\alpha$, up to a certain saturating level $(\alpha < 1)$.

This theorem shows that for any fixed Recall, OBRS consistently improves Precision. More precisely, the improved PR-Curve is a K-fold vertical scaling of the initial

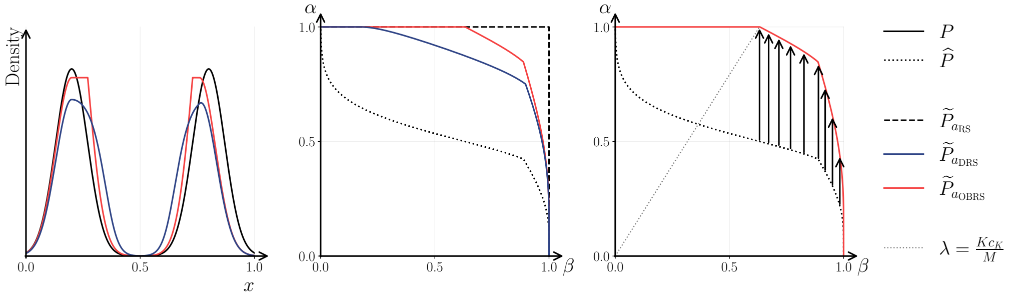


Fig. 6.6.: PR-Curves for the example given in Figure 6.3b. The PR-Curves are given for $\widehat{P}$, $\widetilde{P}_{a_{\mathrm{RS}}} = P$, $\widetilde{P}_{a_{\mathrm{DRS}}}$ and $\widetilde{P}_{a_{\mathrm{OBRS}}}$ for $K = 2$. On the right, we illustrate the improvement of the Precision and the Recall for the OBRS.

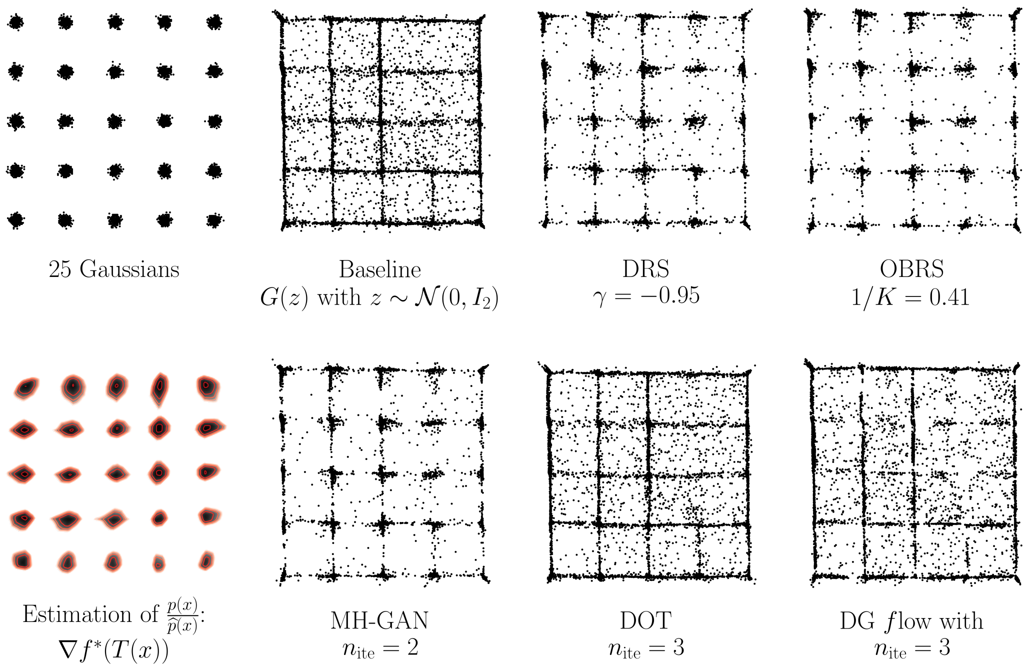
PR-Curve capped at 1. In other terms, if the proposal distribution $\widehat{P}$ has a high Recall but a low Precision, the OBRS will be very efficient. However, if the proposal distribution has a low Recall, the OBRS will require a high budget to improve the Recall.

In Figure 6.6, we show the PR-Curves for the example given in Figure 6.3b. First, we observe that the PR-Curve for $\widetilde{P}_{a_{\mathrm{OBRS}}}$ is always above the PR-Curve for $\widehat{P}$ and $\widetilde{P}_{a_{\mathrm{DRS}}}$. While the Recall is very similar for the two budgeted Rejection Sampling methods, the Precision is significantly improved for the OBRS. We can see that the PR-Curve for $\widetilde{P}_{a_{\mathrm{OBRS}}}$ is a vertical scaling of the PR-Curve for $\widehat{P}$ for $\lambda < \frac{Kc_K}{M}$ and is capped to 1 for $\lambda \geq \frac{Kc_K}{M}$. We will now show how OBRS performs on more complex experiments.

6.2.3 Experiments

In this section, we will show how the OBRS method can improve the Precision and the Recall of a generative model. We will first show the results on a simple 2D example that is traditionally used to compare different sampling methods, then, to demonstrate the versatility of the OBRS method, we will show how it can be used to improve the Precision and the Recall of a Diffusion Model trained on CIFAR-10. Finally, we will compare the OBRS method to the DRS method on a BigGAN trained on CelebA64.

25 Gaussians: A traditional distribution to compare different sampling methods is the 2D 25 Gaussians distribution. The target distribution P is a mixture of 25 Gaussians with equal weights and unit variance. We train a GAN to learn the distribution P by minimizing $\mathcal{D}_{\mathrm{GAN}}$. Using this model as the proposal distribution, we can compare the different sampling methods detailed in Section 6.1.3. For the baseline GAN model $\widehat{P}$, we plot the samples for the different sampling methods in Figure 6.7. We evaluated the different methods using a metric of quality and



25 Gaussians

Baseline $G(z)$ with $z \sim \mathcal{N}(0, I_2)$

DRS $\gamma = -0.95$

OBRS $1/K = 0.41$

Estimation of $\frac{p(x)}{\hat{p}(x)}$: $\nabla f^*(T(x))$

MH-GAN $n_{\text{ite}} = 2$

DOT $n_{\text{ite}} = 3$

DG $f$low with $n_{\text{ite}} = 3$

Fig. 6.7.: Example of DRS, OBRS, MH-GAN, DOT and DG$f$low on the 2D 25 Gaussians. The target distribution P is a mixture of 25 Gaussians and the proposal distribution is a GAN. The samples are generated using the different sampling methods with similar budget. The Precision and Recall for the different methods are given in Table 6.1.

diversity typically used in this experiment introduced by Dumoulin et al. [36]. The Precision is measured by the ratio of the number of points falling within 3 standard deviations of the closest mean of the 25 Gaussians. The Recall is measured by the ratio of the modes that have at least one sample falling within 3 standard deviations of the mean. Although this definition of quality and diversity is very simple, it is a good indicator of the quality of the generated samples and is used by various authors in the literature [7, 9, 20, 117, 123]. On the other hand, the recall is very dependent on the number of samples generated as a single sample which falls within 3 standard deviations of the mean can increase the recall significantly.

The computational budget of MH-GAN is hardly tunable, thus we start by setting the number of iterations of MH-GAN. Then we fix by binary search the budget of the other methods to have a similar computational cost. We show the results in Table 6.1. For such a mass covering proposal distribution, all the methods have a high Recall. However, we can see that OBRS outperforms the other methods in terms of not only Precision but also the time required to generate the samples. The DRS method has a similar time to the OBRS method, but a lower Precision. In terms of inferences on the generator and the discriminator, the OBRS method requires a similar number of calls on the generator and the discriminator as the DRS method. The MH-GAN method requires a significantly higher number of calls to the discriminator, leading

Model	Recall (%)	Precision (%)	Call of G	Call of D	Time (s)
Baseline G	100.0 ± 0.0	55.80 ± 0.99	2500 ± 0	0 ± 0	0.03 ± 0.01
OBRS (ours)	100.0 ± 0.0	**92.54 ± 0.54**	6262 ± 92	**6262 ± 92**	**0.45 ± 0.01**
DRS	100.0 ± 0.0	89.87 ± 0.59	6411 ± 93	6411 ± 93	**0.46 ± 0.01**
MH-GAN	100.0 ± 0.0	89.98 ± 0.61	6415 ± 45	19292 ± 23	6.38 ± 0.09
DOT	100.0 ± 0.0	58.47 ± 1.00	**2500 ± 0**	7500 ± 0	0.94 ± 0.14
DG$f$low	94.81 ± 2.83	56.00 ± 1.02	7500 ± 0	7500 ± 0	0.67 ± 0.13

Tab. 6.1.: Mixture of 25 Gaussians in 2D. Metrics for the different sampling methods: Recall (↑) and Precision (↑) as defined in [36]; Calls (↓) of G and D are the number of times the models are called to generate 2500 samples; Time (↓) is the time required to generate 2500 samples. For all metrics, we give the average and standard deviation for 1000 generations of 2500 samples. The best results are emphasized in **bold**.

to a higher computational cost. The DOT and DG$f$low methods have a fixed number based on the number of samples generated but perform a gradient descent on the models, thus explaining the longer time.

This experiment shows that the OBRS method outperforms the other methods in terms of Precision and Recall for a similar computational cost. We can also compare different methods for various budgets. In Figure 6.8, we show that for a given precision the OBRS method requires less time than the DRS method, approximately 5 times less than MH-GAN, 10 times less than DOT and 100 times less than DG$f$low. However, since DOT and DG$f$low allows by gradient descent to move the points, rather than accepting or rejecting them, the refinement of the distribution can be more efficient. We can see that for large budget, the two methods achieve a perfect Precision to the cost of a loss of Recall. For example, in Figure 6.10 that for DOT the refined distribution collapsed to a few single points. Finally, we can also compare the OBRS method with the DRS method with different proposal distributions. In Figure 6.9, we show the results for different acceptance rates. We can see that the OBRS method systematically outperforms the DRS method for all acceptance rates.

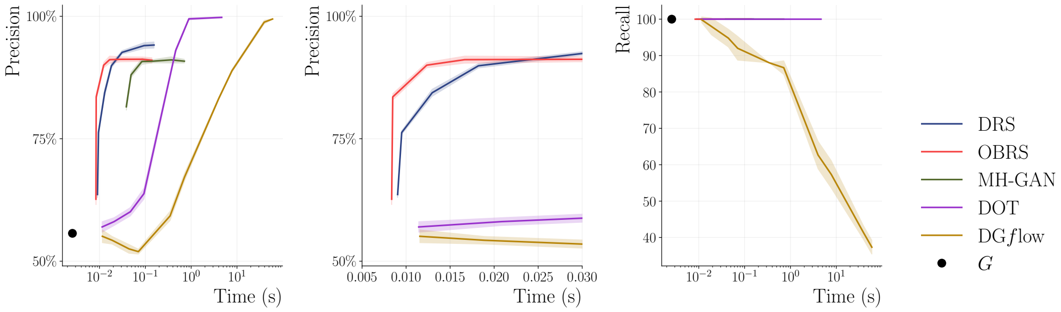


Fig. 6.8.: Precision and Recall for the 2D 25 Gaussians for computation times. For a given Precision, the OBRS method requires less time than the DRS method.

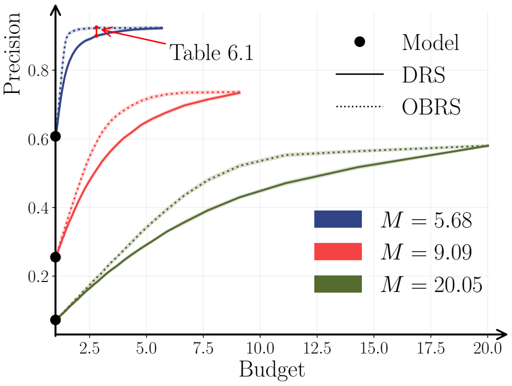


Fig. 6.9.: Comparison of OBRS and DRS for proposal distributions with different acceptance rates.

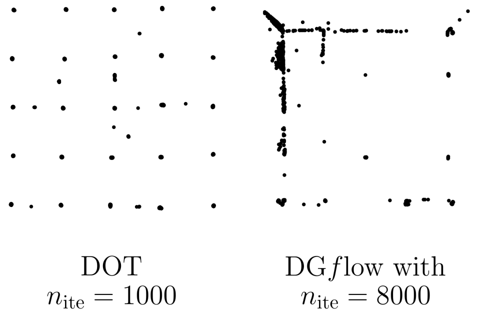


Fig. 6.10.: DOT and DG$flow$ for a large computational budget.

This experiment shows that the OBRS method can improve the Precision of generative models and, for low budgets, outperforms the different sampling methods in generative models. We will now show how the OBRS performs in a higher dimension.

Comparison of OBRS and DRS on CelebA64: To evaluate OBRS and DRS, we can use a BigGAN [16] model trained on CelebA in dimension 64×64. The model is trained using a hinge loss and therefore is poorly suited for density estimation. Therefore, after training the generator, as recommended by Azadi et al. [9] the discriminator is fine-tuned to estimate the density ratio by estimating $\mathcal{D}_{\text{GAN}}$. The discriminator is also calibrated as recommended by Turner et al. [123] and Che et al. [20] so that $\mathbb{E}_{\widehat{P}}[r(\boldsymbol{x})] = 1$. Then, we can use the discriminator to compute the acceptance function a_{OBRS} and a_{DRS} for different budgets. In this experiment, M, the maximum value of the density ratio is approximately 10^5, therefore accepting

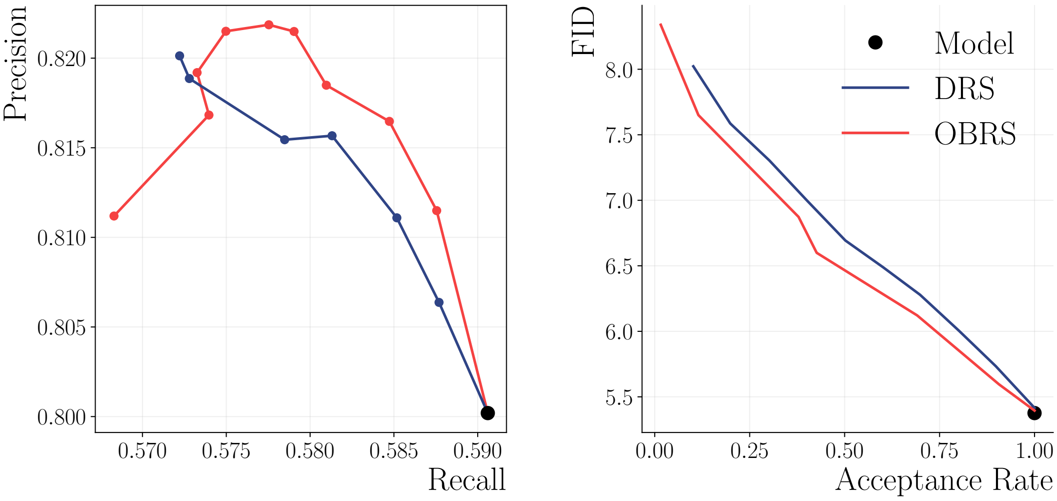


Fig. 6.11.: Comparison of OBRS and DRS for a BigGAN trained on CelebA64. The Precision (↑), the Recall (↑) and the FID (↓) are given for different acceptance rates.

one sample every 10^5 generations. In the following experiments, we did not evaluate traditional rejection sampling as it required more than 3 weeks to generate the 50k samples required for the evaluation metrics using 2 A100 80GB GPUs. Note that without rejection sampling, 50k samples can be generated in less than 20 seconds.

We can compare the two methods with the Precision, the Recall, and the FID for different acceptance rates. We show the results in Figure 6.11. Furthermore, we can see that the OBRS method systematically outperforms the DRS method for all acceptance rates in terms of FID. In terms of Precision and Recall, the OBRS method performs better than DRS only for acceptance rate larger than 30%, i.e., for lower budget. We believe that the discrepancy between the theoretical optimality and the empirical results can be explained by the fact that the discriminator is not perfectly estimating the density ratio, especially in the low density regions.

Versatility of OBRS: Rejection Sampling and in particular Optimal Budgeted Rejection Sampling does not only apply to GANs but also to any generative model. To demonstrate the versatility of our approach, we have used a discriminator trained by Kim et al. [66] on a diffusion model trained on CIFAR-10 by Karras et al. [61]. We observe in Table 6.2 that the OBRS method improves the FID of the generative model. However, Precision is slightly improved, and the Recall is not significantly improved. This experiment shows that the OBRS method can be applied to any generative model and improve the quality of the generated samples.

Every experiment shows that the OBRS method outperforms not only the DRS method but also the other sampling methods in terms of improving quality and preserving diversity. We have shown that the OBRS method can be applied to any generative model and improve the quality of the generated samples. However, the OBRS is particularly efficient when the proposal distribution has a high Recall, and in this section we have only used OBRS on fixed (pre-trained) models. Therefore, a different model, for instance, more mass-covering, might have lower performance before the rejection, but could be further improved with a lower budget. In the next section, we will show how we can train the model to take into account the rejection mechanism and further improve the quality and diversity.

1/K	FID	P	R
0.25	1.57	78.48	86.73
0.50	1.58	78.23	86.05
0.75	1.77	77.94	86.54
1	1.97	77.91	86.62

Tab. 6.2.: OBRS applied on a Diffusion Model EDM [61] with a classifier trained by Kim et al. [66]. The Precision (↑), the Recall (↑) and the FID (↓) are given for different acceptance rates.

6.3 Training the refined distribution

In traditional generative modeling, the generator G is optimized without considering any *a priori* knowledge regarding the rejection sampling that occurs post-training, which can lead to suboptimal generative models. This section advocates training with OBRS (Tw/OBRS) for generative models. First, we introduce the theoretical improvements and the observed effects on the loss function. Then, we introduce an algorithm to incorporate OBRS in the training procedure, and finally, we will show experimental results on high-dimensional generative models.

6.3.1 Principle of training with OBRS

In order to formalize the difference between training traditional procedure and directly training the refined distribution, let us reformulate rejection sampling, and in particular OBRS, in the domain of probability measures.

Let us define

$$B_K(\widehat{P}) = \left\{ \widetilde{P} \in \mathcal{P}(\mathcal{X}) \;\middle|\; \mathcal{D}^{\mathrm{R}}_{\infty}(\widetilde{P} \| \widehat{P}) \leq \log K \right\}, \tag{6.29}$$

where $\mathcal{D}^{\mathrm{R}}_{\infty}(\widetilde{P}\|\widehat{P}) = \sup_{\mathcal{X}} \log \widetilde{p}(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$ denotes the max-divergence (a limiting case of the α-Rényi Divergence $\mathcal{D}^{R}_{\alpha}$ with $\alpha \to \infty$). Note that $B_K(\widehat{P})$ is a convex set. Moreover, the following inclusion holds for any $K_2 \geq K_1 \geq 1$.

$$B_{K_1}(\widehat{P}) \subseteq B_{K_2}(\widehat{P}). \tag{6.30}$$

This set, a ball defined with respect to the weak metric $\mathcal{D}^{\mathrm{R}}_{\infty}$, characterizes the set of distributions allowed by a budgeted rejection sampling procedure. The following lemma shows that $B_K(\widehat{P})$ characterizes the set of distributions allowed by a rejection sampling procedure with a budget K and a proposal distribution $\widehat{P}$:

Lemma 6.3.1.
Let $\widehat{P} \in \mathcal{P}(\mathcal{X})$ be a distribution and $K \geq 1$ be a budget. $\widetilde{P} \in B_K(\widehat{P})$ if and only if there exist an acceptance function $a : \mathcal{X} \to [0,1]$, and a normalization constant Z such that $\widetilde{p}(\boldsymbol{x}) = \widehat{p}(\boldsymbol{x})a(\boldsymbol{x})/Z$ and the acceptance rate is greater than $1/K$.

Proof. This comes from the definition of the set $B_K(\widehat{P})$ as the set of distributions $\widetilde{P}$ that can be written as $\widetilde{p}(\boldsymbol{x}) = \widehat{p}(\boldsymbol{x})a(\boldsymbol{x})/Z$ with $\sup_{\mathcal{X}} \log \widetilde{p}(\boldsymbol{x})/\widehat{p}(\boldsymbol{x}) = \sup_{\mathcal{X}} \log a(\boldsymbol{x})/Z = \log 1/Z \leq \log K$.

Until now, we have mostly considered $\mathcal{P}(\mathcal{X})$, the set of all probability measures defined on $\mathcal{X}$. However, in generative modeling, the distributions we can learn are restricted to the set $\widehat{\mathcal{P}} = \left\{\widehat{P} = G\#Q \;\middle|\; G \in \mathcal{G}\right\}$, the set of all distributions $\widehat{P}$ induced by the generator functions $G \in \mathcal{G}$ from a fixed latent distribution Q. Traditionally, training the proposal distribution and defining the rejection scheme are sequential. By separating the training process from the rejection sampling process, we are, in effect, solving a two-step minimization problem, given below.

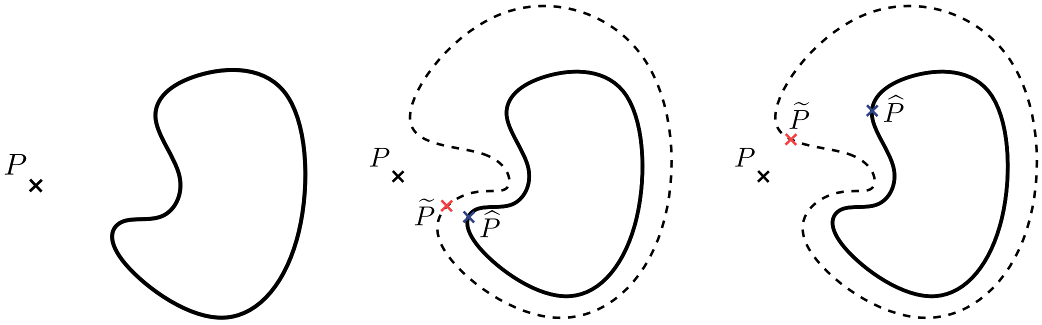
$$\text{First solve } \widehat{P}^{\text{opt}} \in \operatorname*{argmin}_{\widehat{P} \in \widehat{\mathcal{P}}} \mathcal{D}_f(P \| \widehat{P}); \tag{6.31}$$

$$\text{Next solve } \widetilde{P}^{\text{opt}} \in \operatorname*{argmin}_{\widetilde{P} \in B_K(\widehat{P}^{\text{opt}})} \mathcal{D}_f(P \| \widetilde{P}). \tag{6.32}$$

Crucially, $\widehat{P}^{\text{opt}}$ is chosen by the training procedure to optimize (6.31) whereas the final refined distribution $\widetilde{P}^{\text{opt}}$ is assessed via (6.32), resulting in a mismatched objective since $\widetilde{P}^{\text{opt}}$ is the final distribution. By incorporating the rejection scheme into the training objective, we get the following.

$$\min_{\widehat{P} \in \widehat{\mathcal{P}}} \min_{\widetilde{P} \in B_K(\widehat{P})} \mathcal{D}_f(P \| \widetilde{P}). \tag{6.33}$$

In other words, we propose to directly minimize any f-divergence between the target distribution and the refined distribution. We give an illustration of the training procedure in Figure 6.12. In addition to the (naive) improvement of the final divergence, we can show with example two side effects of training with OBRS that we will illustrate with simple examples but that we will also observe in high-dimensional generative models in Section 6.3.3.



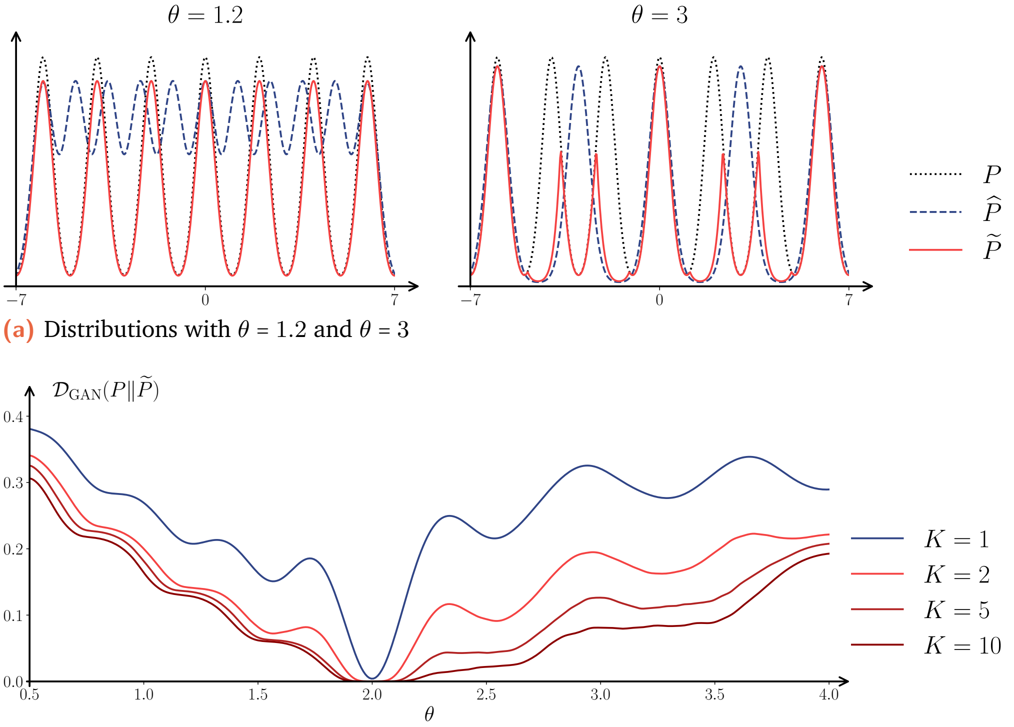
(a) Set $\widehat{\mathcal{P}}$ (b) Training without OBRS (c) Training with OBRS

Fig. 6.12.: Illustration of the training of the refined distribution. The set $\widehat{\mathcal{P}}$ is the set of all distributions induced by the set of generator functions $\mathcal{G}$. When training without OBRS, the generator is trained to minimize the divergence between the target distribution P and the learned distribution $\widehat{P}$, and then the refined distribution $\widehat{P}$ is the optimal distribution in the set $B_K(\widehat{P})$. When training with OBRS, the generator is trained such that the refined distribution $\widehat{P}$ is the optimal distribution within the union of all sets $B_K(\widehat{P})$.

Flattening effect on the parameter landscape: Note that the objective in (6.33) can be written as,

$$\min_{\widetilde{P} \in \bigcup_{\widehat{P} \in \widehat{\mathcal{P}}} B_K(\widehat{P})} \mathcal{D}_f(P \| \widetilde{P}). \tag{6.34}$$

Observe that the domain of $\widetilde{P}$ is the dilatation of $\widehat{\mathcal{P}}$ by the convex set B_K, resulting in a larger set $\bigcup_{\widehat{P} \in \widehat{\mathcal{P}}} B_K(\widehat{P})$. In practice, this results in a flattened loss landscape for optimizing over $\widehat{P}$ as in (6.33), thus preventing the model from getting stuck in suboptimal local minima. This concept is demonstrated with two examples that showcase its ability to flatten the parameter landscape. Firstly, Figure 6.13 shows a one-dimensional example where the loss is flattened by OBRS. We consider P to be a mixture of 10 Gaussians evenly distributed. The proposal distribution $\widehat{P}$ is also a mixture of 10 Gaussians with the same variances. However, the Gaussians are separated by a parameterized distance θ. In Figure 6.13a, we give two examples of proposal distributions with $\theta = 1.2$ and $\theta = 3$. In Figure 6.13b, we show the loss landscape without OBRS ($K = 1$) and with OBRS ($K > 1$). We can see that the loss landscape is flattened by OBRS as the budget K increases. With a high budget, $K = 10$, for instance, all local minima of loss have disappeared.



(a) Distributions with $\theta = 1.2$ and $\theta = 3$

(b) Loss landscape

Fig. 6.13.: Illustration of the flattening effect of the loss landscape. (a) The target distribution P is a mixture of 10 Gaussians. The proposal distribution $\widehat{P}$ is also a mixture of 10 Gaussians with the same variances separated by θ. (b) The loss landscape for the target distribution P and the refined distribution $\widetilde{P}$ without OBRS ($K = 1$) and with OBRS ($K > 1$). The loss landscape is flattened by OBRS.

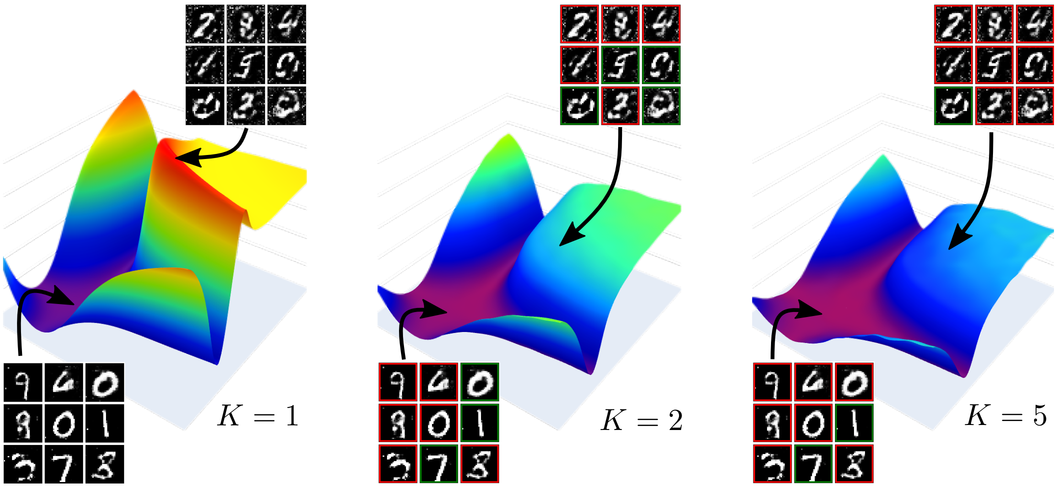
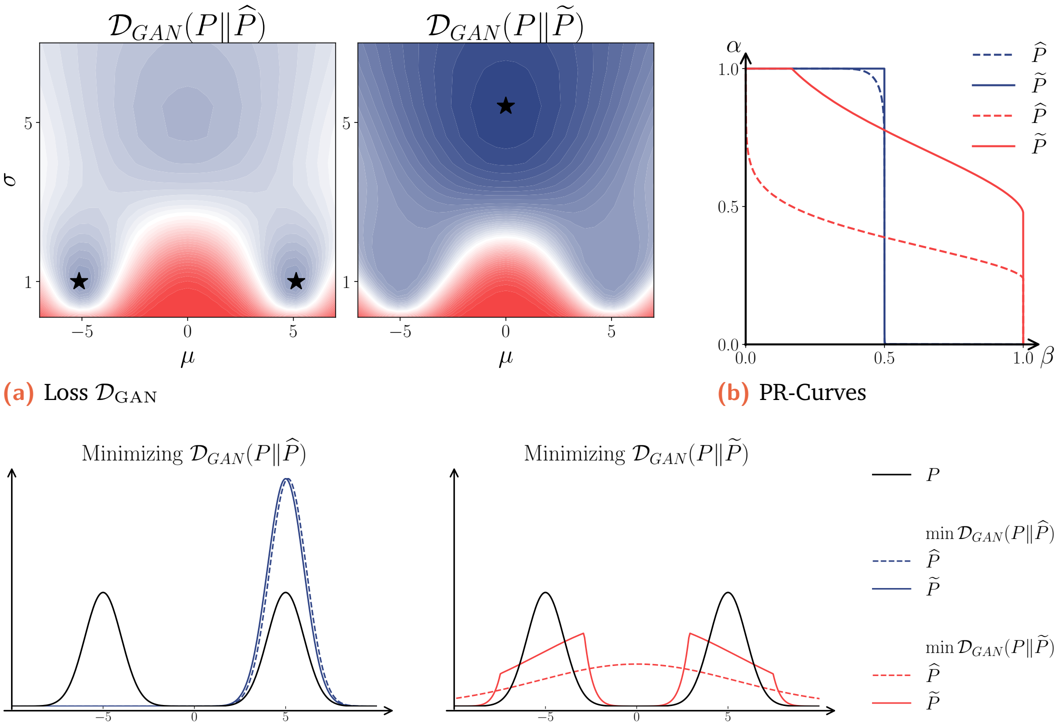


Fig. 6.14.: The loss landscape in the parameter domain of a GAN trained on MNIST. The x-axis and y-axis are random directions in the parameter space. The loss is between the target distribution P and the post-rejection distribution. There are three cases: no rejection ($K = 1$), 50% acceptance rate ($K = 2$) and 20% acceptance rate ($K = 5$). We give samples of the generator whose parameters are in the lowest and highest loss. In red, we show the rejected samples and in green the accepted samples. OBRS not only reduces loss, but also flattens out the loss landscape and helps avoid local minima.

We can also illustrate the flattening effect of the loss landscape in high-dimensional generative models similarly to Li et al. [77]. To do so, we can observe the loss in a two-dimensional projection of the parameters' domain of a neural network. Therefore, we can train a GAN on MNIST using the traditional f-GAN procedure using a generator G and a discriminator T. Let us define θ_0, the parameter vector of the generator G_{θ_0}. We randomly draw two directions θ_1 and θ_2 in the parameter domain: defining a hyperspace of generators defined as $G_{\theta_0+x\theta_1+y\theta_2}$ with $(x, y) \in \mathbb{R}^2$. For any given set of parameters, we can fine-tune a discriminator to estimate the density ratio and apply OBRS for different budgets. In Figure 6.14, we plot the loss in a parameter domain and show a batch of samples drawn from G_{θ_0} (lower left) and from G_θ for the highest loss (upper right). When OBRS is applied, we show in red the rejected samples and in green the accepted samples.

We observe that similarly to the one-dimensional example, the loss landscape is flattened by OBRS. We will see in Section 6.3.3 that this flattening effect is beneficial for training generative models. Furthermore, we will see how training with OBRS affects the proposal distribution.

A mass-covering $\widehat{P}$: The optimal $\widehat{P}$ might be different between (6.31) and (6.33). Theorem 6.2.3 explicitly states that OBRS is more efficient on mass-covering models than on mode-seeking ones. Taking rejection sampling into account in the training procedure pushes the distribution $\widehat{P}$ to be more *suitable* for rejection, and thus: more mode coverage. For example, consider a target distribution P as the Gaussian



(a) Loss $\mathcal{D}_{\mathrm{GAN}}$

(b) PR-Curves

(c) Distributions

Fig. 6.15.: Example of training without and with OBRS. The target distribution P is the Gaussian mixture, and the proposal distribution is a single Gaussian. (a) The loss of the generator G with respect to the target distribution P and the refined distribution $\widetilde{P}$ without OBRS ($K = 1$) and with OBRS ($K = 2$). (b) PR-Curves for the different distributions. (c) Distributions P, $\widehat{P}$, and $\widetilde{P}$ trained without and with OBRS.

mixture presented in Figure 6.15. Assume that the expressivity of $\widehat{P}$ is limited to a single Gaussian $\mathcal{N}(\mu, \sigma)$. If the goal is to naively minimize $\mathcal{D}_{\mathrm{GAN}}$, then, because of the mode-covering property of the divergence, $\widehat{P}$ covers only one mode. In that case, Theorem 6.2.3 shows that only precision can be improved, and thus a limited-budget rejection sampling scheme will not reshape $\widehat{P}$, leading to poor coverage, as the PR-Curve in Figure 6.15b shows. However, if μ and σ are set to directly minimize $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$, then the distribution $\widehat{P}$ changes drastically into a mass covering distribution, allowing the rejection process to match more closely (in terms of $\mathcal{D}_{\mathrm{GAN}}$), and the PR-Curve to improve as shown in Figure 6.15b.

Incorporating the OBRS in the training procedure is beneficial not only for the final quality of the generative model but also for the training procedure itself. We will now show how we can train the refined distribution with OBRS.

6.3.2 Our method

In this section, we propose a method to train discriminator-based generative models similar to the f-GAN framework. In other words, our method can be applied to either GANs or Normalizing Flows. We will first present the algorithm and then show the results on BigGAN models in high dimension in Section 6.3.3.

A direct approach to train the generator with OBRS is to train a discriminator $\widetilde{T}$ based on samples of both P and $\widetilde{P}$ to estimate $\mathcal{D}_f(P \| \widetilde{P})$. By estimating the f-divergence via the dual approximation, the generator can by train to minimize the f-divergence via importance sampling:

$$-\mathbb{E}_{\widetilde{P}}\left[f^*(\widetilde{T}(\boldsymbol{x}))\right] = -\mathbb{E}_{\widehat{P}}\left[\frac{\widetilde{p}(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} f^*(\widetilde{T}(\boldsymbol{x}))\right] \tag{6.35}$$

$$= -\mathbb{E}_{\widehat{P}}\left[K a_{\mathrm{OBRS}}(\boldsymbol{x}) f^*(\widetilde{T}(\boldsymbol{x}))\right]. \tag{6.36}$$

This method is very similar to the f-GAN framework recalled in Algorithm 6. However, the loss function for the generator depends on the acceptance probability a_{OBRS}. This function depends on the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. Therefore, it requires two different discriminators:

1. T used to estimate $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$ and trained with samples from P and $\widehat{P}$.
2. $\widetilde{T}$ used to estimate $p(\boldsymbol{x})/\widetilde{p}(\boldsymbol{x})$ and trained with samples from P and $\widetilde{P}$.

Needing two discriminators can be cumbersome, and we propose a simpler method with only one discriminator using the primal estimation of the final f-divergence. To do so, we train the discriminator T to estimate $\mathcal{D}_f(P \| \widehat{P})$ in order to approximate

Algorithm 6 f-GAN Tw/oOBRS

```
repeat
    Update $T$ by maximizing
        $\mathbb{E}_{\boldsymbol{x}\sim P}[T(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}[f^*(T(\boldsymbol{x}))]$.
    Update $G$ by minimizing
        $-\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}[f^*(T(\boldsymbol{x}))]$.
until convergence.
```

Algorithm 7 f-GAN Tw/OBRS

```
repeat
    Update $T$ by maximizing
        $\mathbb{E}_{\boldsymbol{x}\sim P}[T(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}[f^*(T(\boldsymbol{x}))]$.
    Update $c_K$ such that
        $\mathbb{E}_{\widehat{P}_G}[a_{\mathrm{OBRS}}(\boldsymbol{x})] \leq 1/K$.
    Update $G$ by minimizing
        $\mathbb{E}_{\boldsymbol{x}\sim \widehat{P}_G}\left[K a_{\mathrm{OBRS}}(\boldsymbol{x}) f\left(\frac{r(\boldsymbol{x})}{K a_{\mathrm{OBRS}}(\boldsymbol{x})}\right)\right]$.
until convergence.
```

the density ratio $p(\boldsymbol{x})/\widetilde{p}(\boldsymbol{x})$. Using this density ratio, we can compute the final f-divergence:

$$\mathcal{D}_f(P\|\widetilde{P}) = \mathbb{E}_{\widetilde{P}}\left[f\left(\frac{p(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \tag{6.37}$$

$$= \mathbb{E}_{\widehat{P}}\left[\frac{\widetilde{p}(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{\widehat{p}(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \tag{6.38}$$

$$= \mathbb{E}_{\widehat{P}}\left[K a_{\mathrm{OBRS}}(\boldsymbol{x}) f\left(\frac{1}{K a_{\mathrm{OBRS}}(\boldsymbol{x})}\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)\right]. \tag{6.39}$$

Consequently, we propose the Algorithm 7 to train the generator with OBRS. The discriminator is trained the same way as in the f-GAN framework, but the generator is trained to minimize:

$$\mathbb{E}_{\widehat{P}}\left[K a_{\mathrm{OBRS}}(\boldsymbol{x}) f\left(\frac{\nabla f^*(T(\boldsymbol{x}))}{K a_{\mathrm{OBRS}}(\boldsymbol{x})}\right)\right]. \tag{6.40}$$

Note that a_{OBRS} depends on the density ratio estimated by $\nabla f^*(T(\boldsymbol{x}))$ but also on the constant c_K. This value depends on the budget K but also on the proposal distribution $\widehat{P}$ and thus needs to be updated at each iteration. To do so, we use the Algorithm 5 presented in Section 6.2.1. In the next section, we will see how the frequency of the update of c_K can affect the quality of the generative model. The update in c_K is linear in the number of samples generated, and thus the computational cost is similar to the traditional f-GAN framework. We will now show the results of the training with OBRS on high-dimensional generative models.

6.3.3 Experiments

In this section, we implement our method to train a BigGAN model to minimize $\mathcal{D}_{\mathrm{GAN}}(P\|\widetilde{P})$ directly with a budget of $K = 2$. We will show that (1) the training with OBRS improves the speed of convergences and that (2) it improves the diversity and the FID of the model.

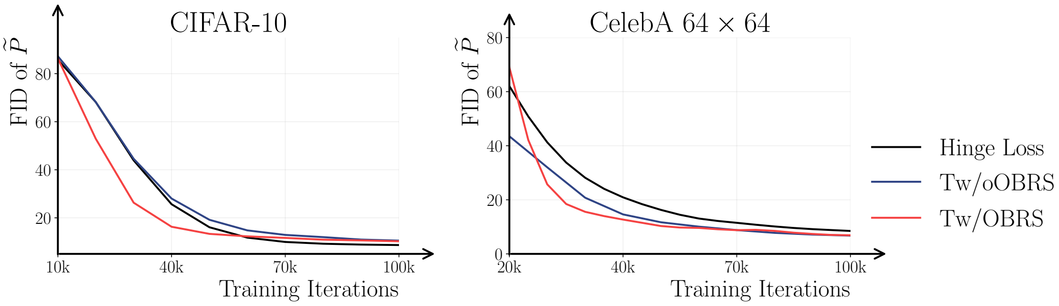


Fig. 6.16.: Training of a BigGAN on CelebA64 with the Baseline (Hinge Loss) and minimizing $\mathcal{D}_{\mathrm{GAN}}$ without OBRS (Tw/oOBRS) and with OBRS (Tw/OBRS). The FID (↓) is given for a budget $K = 2$.

Improving the speed of convergence: We have discussed in Section 6.3 that training with OBRS can flatten the loss landscape and avoid local minima. We show that its effect translates into faster convergence in practice. To evaluate this effect, we train BigGAN models with baseline loss (hinge loss), with traditional f -GAN loss (Tw/oOBRS), or using our approach (Tw/OBRS). We can observe in Figure 6.16 the FID during the training procedure for models trained on CIFAR-10 and CelebA. We can see that models trained with OBRS converges faster than both the baseline and the traditional loss f -GAN. Furthermore, we can also test this effect with several frequencies of the update of c_K. Several models are trained with the parameters c_K being updated every N_c iterations. In Figure 6.17, we show the FID for different frequencies of the update of c_K along with the time required to train the model. We observe that the frequency of updates does not affect the speed of convergence. Furthermore, we observe that updating c_K every 10 operation takes on average 19% longer to train than $\mathcal{D}_{\text{GAN}}$ without OBRS, while updating every 100 and 1000 iterations are only 1.69% and 0.03% longer.

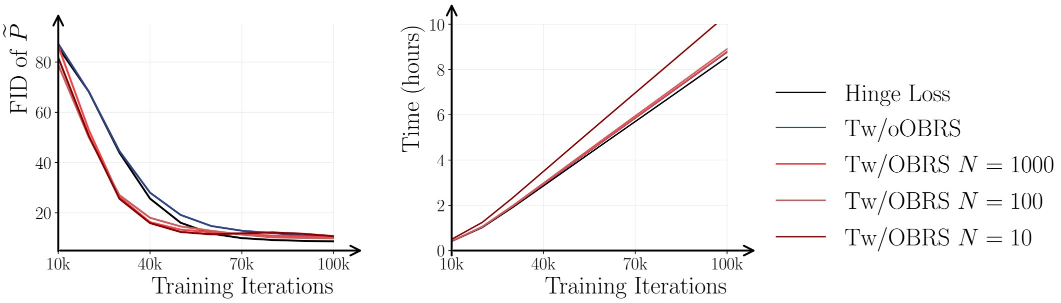


Fig. 6.17.: Effect on the frequency of the update of c_K on the FID of a BigGAN trained on CIFAR-10. The FID (↓) is given for different frequency of the update of c_K during training.

Improving the performances of the generative model: We have shown in Section 6.3 that training can lead to more mass-covering proposal distribution and that it can improve the overall performance of the model for a given budget. To test this, we train BigGAN models on CIFAR-10 and CelebA with the baseline loss, the traditional f-GAN loss, and our approach. To test our approach in higher dimension, we also fine-tune pre-trained models on CelebA and ImageNet. Every model is trained with $K = 2$ and, consequently, the model is evaluated with OBRS with $K = 2$. We show the results in Table 6.3 and Table 6.4. The FID for models trained with OBRS is generally better than for models trained with the traditional loss f -GAN and the baseline. However, a slight trade-off is observed for the Precision. The Recall is improved for the models trained with OBRS. For the fine-tuned models, the improvement in Recall is even more significant.

We have shown that training with OBRS can improve the performance of the generative model and that it can improve the speed of convergence. We have also shown

Dataset	Method	FID	P	R
CIFAR-10	Hinge Loss	**8.43**	**84.50**	65.39
32 × 32	Tw/oOBRS	11.18	83.24	68.44
	Tw/OBRS	8.98	80.09	**69.63**
CelebA	Hinge Loss	9.33	**80.23**	57.78
64 × 64	Tw/oOBRS	6.33	78.28	**61.02**
	Tw/OBRS	**5.42**	78.01	60.29

Tab. 6.3.: Results of training a BigGAN on different datasets. The FID (↓), the Precision (↑) and the Recall (↑) are given for the different methods.

Dataset	Method	FID	P	R
CelebA	Hinge Loss	9.33	**80.23**	57.78
64 × 64	Tw/OBRS	**3.74**	74.40	**65.15**
ImageNet	Hinge Loss	12.18	**27.75**	34.33
128 × 128	Tw/OBRS	**11.65**	26.84	**46.16**

Tab. 6.4.: Results of fine-tuning a BigGAN on different datasets. The FID (↓), the Precision (↑) and the Recall (↑) are given for the different methods.

that the frequency of update of c_K does not affect the performance of the model. Furthermore, we will now conclude this chapter and discuss the limitations and the perspectives of the OBRS method.

6.4 Concluding Remarks and Discussion

In this chapter, we addressed the question on improving Precision and Recall:

- **Question Q4:** *With rejection sampling under limited budget, how much can we increase Precision and Recall of a pre-trained model?*
  We propose an optimal rejection scheme with limited budget. We showed that this new sampling algorithm improves Precision but can hardly improve Recall. Therefore, to tackle this limitation, we advocate training the model with the rejection scheme. We have proposed a method to achieve this, and we showed that training with the rejection scheme can improve the Precision and the Recall of the model.

This work is this chapter contributes to help to improve generative models with limited additional resources. However, this approach could be extended or improved in several ways.

- **Improving density ratio estimation:** First, we are using the density ratio estimated by the discriminator trained with the traditional f-GAN loss. However, the density ratio is not perfectly estimated, and this leads to a suboptimal acceptance function. Similarly to Chapter 5, we could use the density ratio estimated with a discriminator trained with an auxiliary loss to improve the quality of the estimation of the acceptance function.

- **Extending the OBRS to diffusion models:** Also, our method only applies to rejection in the image space. In diffusion models, the image is generated by a series of transformations. We could apply the rejection in the latent space during the denoising process. This implies that the budget by sample can be further quantized and, therefore, the rejection can be refined. The work of Kim et al. [66] estimate the density ratio during the denoising process and could be used to apply the rejection in the latent space.

7 Conclusion

Contents

7.1 Summary of the Thesis

In this thesis, we have studied the trade-off between quality and diversity in generative models through the lens of the f-divergences in order to answer:

Question: *How can we characterize, tune, and improve precision and recall of Generative Models?*

We have tackled these problems from a theoretical point of view first on then we also addressed more practical aspects of training and refining neural network-based models. We can summarize the main contributions of this thesis as follows.

Summary of the contributions:

1. *We have unified several existing metrics of Precision and Recall in the f-divergence framework by introducing the Precision-Recall Divergence and by showing that most Precision-Recall based metrics can be expressed using this divergence. Furthermore, we have showed how any f-divergence can be written as trade-offs between Precision and Recall.*

2. *Equipped with the PR-Divergence, we have shown how popular generative models can sometimes demonstrate Precision or Recall. Our analysis establishes the link between the Lipschitz constraints of several neural networks with the occurrence of pathological cases where the PR-Divergence is bounded.*

3. *Building the connection between Precision-Recall metrics and f-divergence, we have shown that Precision-Recall Divergence can be used to fine-tune generative models. We have proposed a new training algorithm that uses the PR-Divergence to*

balance the quality and diversity of discriminator-based models such as Generative Adversarial Networks and Normalizing Flows.

4. *Finally, we have used a more elaborate sampling algorithm to improve the quality and diversity of the generated samples. We have shown that there exists a budget-constraint rejection algorithm that can be optimal in terms of f-divergences. We have also proposed a new training algorithm that uses this rejection mechanism, the OBRS, to further improve the performance of generative models by reducing the importance of the quality and diversity trade-off.*

7.2 Open Question and future works

Our work consists of contributions both theoretical and experimental. One of the key contribution of this thesis is the unification of Precision and Recall metrics in the f-divergence framework. We have shown that the Precision-Recall Divergence can be used to fine-tune between quality and diversity in generative models. However, for the practical aspects, we have focused our work on discriminator-based models such as Generative Adversarial Networks and Normalizing Flows. Therefore, there are still many open questions that need to be addressed in future work to extend our results to other types of generative models. Finally, we can also use the PR-Divergence for uses. We present some of the most important questions that we believe are worth investigating in the future.

7.2.1 Training fair generative models with PR-Divergence

Fairness is a crucial aspect of machine learning models, especially in generative models where the generated samples can be used to make decisions. However, the evaluation of fairness in generative models is still an open question. In particular, the current fairness metrics are based on the equality of the representation of sensitive attributes in the data distribution and the generated distribution [120, 133]. Typically, if there exists a sensitive attribute $a(\boldsymbol{x}) \in \{0,1\}$ in the data distribution such that $P = \pi_0 P_0 + \pi_1 P_1$, the fairness criterion is simply based on the equality of the representation of a in $\widehat{P} = \hat{\pi}_0 \widehat{P}_0 + \hat{\pi}_1 \widehat{P}_1$. For example,one traditional fairness criterion can be expressed as

$$\mathrm{Criterion}(\widehat{P}) = |\hat{\pi}_0 - \hat{\pi}_1| \,.$$

However, traditional metrics do not account for the differences of Precision Recall trade-offs between attributes. For instance in Figure 7.1, we illustrate a distribution $\widehat{P}$ for which the partition of the sensitive attribute are evenly distributed but the

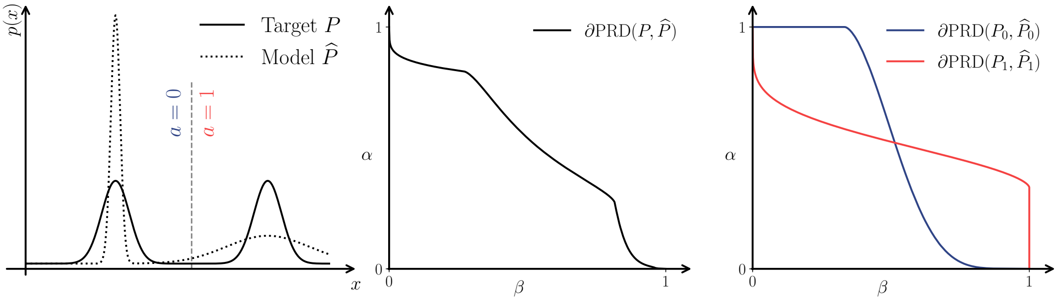


Fig. 7.1.: Pathological case for the usefulness of Precision-Recall Curves for fairness evaluation.

trade-offs between quality and diversity per class are very different. We believe that such a distribution can involve unfairness issues. Therefore, it raises the question of evaluating and training fair generative models based on the Precision-Recall Curve.

Question: *How can we build a fairness criterion based on the Precision-Recall Curve to train fair generative models?*

The first step to answer this question is to define a fairness criterion based on the PR-Curve. One possible way to do so is to a criterion based the dissimilarities between the PR-Curves for each attribute. Then, we could use this criterion to regularize f-divergences minimization during model training.

7.2.2 Quality and Diversity in Large Language Models

In this work we have focused on Precision and Recall to compare distributions over the data. In other words, we are evaluating f-divergences between the distributions P and $\widehat{P}$. This work can also apply Large Language Models (LLMs). In fact, some existing quality diversity metrics have been already adapted to assess LLMs, either with PR-Curves IDF by Pillutla et al. [96] or support-based metrics of Precision and Recall adapted by us in Bronnec et al. [17]. These works, among other study on model collapse [33, 34, 54, 99] highlights that the trade-off between quality and diversity also exists in LLMs. Therefore, it raises the question of how to adapt the PR-Divergence to evaluate the quality and diversity of LLMs.

Question: *How can we adapt the Precision-Recall Divergence to train LLMs on a specific trade-off between quality and diversity?*

The training of LLMs is traditionally autoregressive for the pretrained model and based on Reinforcement Learning (RL) for fine-tuning. For the former one, the challenge is to adapt the PR-Divergence to train on conditional distributions. One possible way to do so is to use is to upper bound the PR-Divergence by the conditional PR-Divergence similarly to the Total Variation in the work of Ji et al. [58]. For the latter, the challenge is to adapt the PR-Divergence to the reinforcement learning framework. One possible way to do so is to use f-divergences minimization RL framework [43, 65]. A potential future work would be to apply these methods to fine-tune LLMs on different trade-offs between quality and diversity and evaluate how it could mitigate model collapse.

7.2.3 Tuning the Precision-Recall Trade-off in Diffusion Models

The training method we have proposed in this thesis can be applied to Generative Adversarial Networks and Normalizing Flows. However, for a few years now, the state-of-the-art for image generation is diffusion models. In particular, Score-Matching Diffusion Models achieve stunning results in image generation [114]. These models are trained by minimizing $\mathcal{D}_{\mathrm{KL}}$. However, we can show that a score matching model s_θ can be trained to minimize any f-divergence between the data distribution and the model distribution. Under mild conditions:

$$\mathcal{D}_f(\widehat{P}\|P) = \mathcal{D}_f(\widehat{P}_T\|P_T) + \\ + \int \lambda(t)\mathbb{E}_{\boldsymbol{x}_t\sim P_t}\left[\frac{p_t(\boldsymbol{x}_t)}{\widehat{p}_t(\boldsymbol{x}_t)} f''\left(\frac{p_t(\boldsymbol{x}_t)}{\widehat{p}_t(\boldsymbol{x}_t)}\right)\|\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t) - s_\theta(\boldsymbol{x}_t)\|^2\right]\mathrm{d}t.$$

This equation shows that f-divergence can be minimized by score-matching models if the density ratio can be estimated. Therefore, it raises the following question.

Question: *How can we use tune the quality and diversity of Diffusion Models with f-divergences minimization?*

As f'' is mostly 0 for the PR-Divergence we could use other f-divergences, mode-seeking or mass-covering, to tune the quality and diversity of diffusion models. To do so, we can use the work of Kim et al. [66] to estimate the density ratio at each time step t. Thus, by reweighing the score matching loss with the estimated density ratio, we can tune the quality and diversity of diffusion models.

A Extension to Bi-Lipschitz Neural Networks

In this appendix, we discuss the extension of Section 4.4 to bi-Lipschitz neural networks. First, we will define the bi-Lipschitz property of a function and then discuss how the Precision-Recall Divergence can be bounded for models that are also bi-Lipschitz.

A.1 Bi-Lipschitz Continuity

Normalizing Flows, as a bijective mapping, are typically considered to be *bi-Lipschitz*, i.e. both the forward and the inverse mappings are Lipschitz continuous with different Lipschitz constants:

Definition A.1.1 ((L_1-L_2)-bi-Lipschitz Continuity)**.**
A function bijective function $G : \mathcal{Z} \mapsto \mathcal{X}$ is $(L_1$-$L_2)$-bi-Lipschitz continuous if both G is L_1-Lipschitz and G^{-1} is L_2-Lipschitz, i.e. if

$$\forall \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{Z}, \quad \|G(\boldsymbol{z}_1) - G(\boldsymbol{z}_2)\| \leq L_1 \|\boldsymbol{z}_1 - \boldsymbol{z}_2\| \tag{A.1}$$

and

$$\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}, \quad \|G^{-1}(\boldsymbol{x}_1) - G^{-1}(\boldsymbol{x}_2)\| \leq L_2 \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|. \tag{A.2}$$

Although the bi-Lipschitz property is sometimes defined in the literature with $L_1 = 1/L_2$, we choose to differentiate the Lipschitz constants to allow for more general results since in general the values differ. In particular, we can consider that $G^{-1} = F_1 \circ \ldots F_K$ is a composition of K residual functions L-Lipschitz $F_k(\boldsymbol{x}) = \boldsymbol{x} + f_k(\boldsymbol{x})$ with $L < 1$ which is the case in ResFlow models [22] . In that case, $L_2 \leq (1 + L)^K$ and $L_1 \geq 1/(1 - L)^K$.

A.2 Pathological case for bi-Lipschitz neural networks

In Section 4.4, we showed how the L_1-Lipschitz continuity of the generator function G can be used to compute a lower bound the PR-Divergence. We assumed the L_1-Lipschitz property of the forward mapping G, and thus applied to both Generative Adversarial Networks, Normalizing Flows and Diffusion models. Here we also assume the L_2-Lipschitz property of the inverse mapping G^{-1}, and thus applied to Normalizing Flows only. Several works insist on the importance of enforcing the Lipschitz continuity of Normalizing Flows and thus enforcing bi-Lipschitz continuity [12, 22]. In this section, we highlight a different pathological case for models with a L_2-Lipschitz inverse mapping G^{-1}.

We can also show that PR-Divergence can be strictly positive for some target distributions P and some generator functions G if the inverse mapping G^{-1} is L_2-Lipschitz. If the inverse mapping is L_2-Lipschitz, it means that the forward mapping cannot contract the mass indefinitely. Therefore, if we assume that there exists a ball $B_{R,\boldsymbol{x}}$ for which the target distribution P concentrates most of its weight, then G cannot map this ball to a region smaller than $B_{L_2 R, G^{-1}(\boldsymbol{x})}$. This concept is demonstrated through a one-dimensional example as depicted in Figure A.2.1. Note that contrary to the previous theorem, we can search for a high concentration ball with any center $\boldsymbol{x}$:

Theorem A.2.1 (Models with L_2-Lipschitz inverse mapping G^{-1} fails to capture high density balls)**.**
Let $P \in \mathcal{P}(\mathcal{X})$ be the target distribution defined on $\mathcal{X} \subset \mathbb{R}^d$, and let $\widehat{P} = G\#Q$ where $G : \mathcal{Z} \mapsto \mathcal{X}$ and Q be the Gaussian distribution defined on $\mathcal{Z} \subset \mathbb{R}^m$. Let $B_{R,\boldsymbol{x}}$ be the balls of radius R centered on $\boldsymbol{x}$. If G^{-1} is L_2-Lipschitz, then we have the upper bound:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) \geq \sup_{R \geq 0, \boldsymbol{x} \in \mathcal{X}} \left(\lambda P(B_{R,\boldsymbol{x}}) - \frac{\gamma\left(\frac{m}{2}, \frac{L_2^2 R^2}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \right) - |\lambda - 1|. \tag{A.3}$$

Therefore, if there exists a ball for which the target distribution P satisfies $P(B_{R,\boldsymbol{x}}) > \frac{1}{\lambda}\gamma\left(\frac{m}{2}, \frac{L_2^2 R^2}{2}\right) / \Gamma\left(\frac{m}{2}\right) + |1 - 1/\lambda|$, then the PR-Divergence is strictly positive.

Proof. The function G^{-1} is L_2-Lipschitz, thus, for every radius $R \geq 0$ and $\boldsymbol{x} \in \mathcal{X}$, we have $G^{-1}(B_{R,\boldsymbol{x}}) \subseteq B_{L_2 R, G^{-1}(\boldsymbol{x})}$. Therefore, we have the following.

$$\widehat{P}(G^{-1}(B_{R,\boldsymbol{x}})) = Q(G^{-1}(B_{R,\boldsymbol{x}})) \leq Q(B_{L_2 R, G^{-1}(\boldsymbol{x})}). \tag{A.4}$$

Fig. A.2.1.: Example of a target distribution for which Theorem A.2.1 applies: the subset B_R concentrates most weight in P, but $\widehat{P}(B_R) = Q(G^{-1}(B_R))$ can only be as large as $Q(B_{RL_2})$.

Using Lemma 4.4.1, we have that:

$$\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) = \sup_{\mathcal{A}\subseteq\mathcal{X}} \left|\lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A})\right| - |\lambda - 1| \tag{A.5}$$

$$\geq \sup_{R\geq 0, \boldsymbol{x}\in\mathcal{X}} \lambda P(B_{R,\boldsymbol{x}}) - \widehat{P}(B_{R,\boldsymbol{x}}) - |\lambda - 1| \tag{A.6}$$

$$\geq \sup_{R\geq 0, \boldsymbol{x}\in\mathcal{X}} \lambda P(B_{R,\boldsymbol{x}}) - Q(B_{L_2 R, G^{-1}(\boldsymbol{x})}) - |\lambda - 1| \tag{A.7}$$

$$\geq \sup_{R\geq 0, \boldsymbol{x}\in\mathcal{X}} \lambda P(B_{R,\boldsymbol{x}}) - Q((B_{L_2 R, 0})) - |\lambda - 1|\,. \tag{A.8}$$

Therefore, using the close form of the measure of the ball $B_{r,\mathbf{0}}$ given in Equation (4.42), we have the result.

Similar conclusions can be drawn as with Theorem referred to in Theorem 4.4.3. The Lipschitz constant L_2, associated with the inverse mapping G^{-1}, plays a crucial role: a higher value of L_2 results in a less stringent bound. The dimension m is another critical factor. However, unlike the case with the previously mentioned theorem, the bound tends to become more limiting as the dimension increases.

Mathematical Supplementary

B

Contents

In this section, we provide full demonstrations of the theorems and propositions presented in Chapter 4 that were too long to be included in the main text. In particular, we provide the proofs of the following results.

- Proposition 4.2.2 (PR-Divergence) in page 59.
- Proposition 4.2.3 (Properties of the PR-Divergence) in page 59.
- Theorem 4.2.4 (PR-Curves as a function of $\mathcal{D}_{\lambda\text{-PR}}$) in page 60.
- Theorem 4.3.1 (Weighted Precision-Recall Divergence) in page 64.
- Lemma 4.4.1 (Probabilistic Formulation of Precision and Recall) in page 69.
- Theorem 4.4.3 (L_1-Lipschitz mapping function G fails to capture the low-density balls) in page 70.
- Theorem A.2.1 (L_2-Lipschitz inverse mapping function G^{-1} fails to capture the high-density balls) in page 128.
- Theorem Convergence PR Div estimation
- Theorem 6.2.3 (Precision and Recall Improvement) in page 107.

B.1 Proofs of Chapter 4

B.1.1 Proof of Proposition 4.2.2

Proposition (PR-Divergence)**.**
For any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$, then for any $\lambda \in [0, +\infty]$ the PR-Divergence defined as

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda \left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.1}$$

belongs to the class of f-divergences.

Proof. An f-divergence is defined with a generator function that satisfies three properties: lower semi-continuous, convexity and $f(1) = 0$. For $\lambda < +\infty$, f_λ can be written as the $\max$ of two linear functions $u \mapsto \lambda u$ and $u \mapsto 1$, it is continuous and convex. Moreover, for $u = 1$ we have $f_\lambda(1) = \max(\lambda, 1) - \max(\lambda, 1) = 0$. For $\lambda = +\infty$, $f_{+\infty}$ the function is lower semicontinuous as the function is continuous on the set $]0, +\infty[$ the epigraph of the function is a convex set, therefore the function is convex. Finally, $f_{+\infty}(1) = 0$. Consequently, the PR-Divergence is an f-divergence.

B.1.2 Proof of Proposition 4.2.3

Proposition (Properties of the PR-Divergence)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, then the following assertions hold.

- *The Fenchel conjugate f_λ^* of f_λ is defined on $\operatorname{dom}\left(f_\lambda^*\right) = [0, \lambda]$ and given by:*

$$f_\lambda^*(t) = \begin{cases} t/\lambda & \text{for } \lambda \leq 1, \\ t/\lambda + \lambda - 1 & \text{otherwise.} \end{cases} \tag{B.2}$$

- *The optimal discriminator for the dual variational form is:*

$$T^{\text{opt}}(\boldsymbol{x}) = \lambda \text{sign}\left(\lambda \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - 1 \right). \tag{B.3}$$

- *The reverse divergence is:*

$$\mathcal{D}_{\lambda\text{-PR}}(\widehat{P} \| P) = \lambda \mathcal{D}_{\frac{1}{\lambda}\text{-PR}}(P \| \widehat{P}). \tag{B.4}$$

- *For $\lambda = 1$, we have:*

$$\mathcal{D}_{1\text{-PR}}(P \| \widehat{P}) = \mathcal{D}_{\text{TV}}(P \| \widehat{P})/2. \tag{B.5}$$

Proof.

- The generator function f_λ of the Precision-Recall Divergence is $f(u) = \max(\lambda u, 1) - \max(\lambda, 1)$ then its Fenchel conjugate function is:

$$f^*(t) = \sup_{u \in \mathrm{dom}(f)} \{ut - f(u)\} \tag{B.6}$$

$$= \max(\lambda, 1) + \sup_{u \in \mathbb{R}^+} \{ut - \max(\lambda u, 1)\}. \tag{B.7}$$

If $t > \lambda$ or $\lambda < 0$, then the $\sup_{u \in \mathbb{R}^+} \{tu - \max(\lambda u, 1)\} = \infty$ for respectively $u \to \infty$ and $u \to -\infty$. The domain of f^* is thus restricted to $[0, \lambda]$. Thus, for $0 \le t \le \lambda$, the supremum is obtained for $u = 1/\lambda$ since 0 is in the sub-differential of the function in $1/\lambda$ as illustrated in Figure B.1.1 Consequently the Fenchel conjugate of f is:

$$\forall t \in [0, \lambda], \quad f^*(t) = \max(\lambda, 1) + t\lambda - 1 = \begin{cases} t/\lambda & \text{if } \lambda \le 1, \\ t/\lambda - 1 + \lambda & \text{otherwise.} \end{cases} \tag{B.8}$$

- We have that $\forall \boldsymbol{x} \in \mathcal{X}$, the optimal discriminator function T^{opt} satisfies:

$$T^{\text{opt}}(\boldsymbol{x}) = \nabla f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right). \tag{B.9}$$

And, given the expression of f_λ in Equation (4.1), the function is constant on $[0, 1/\lambda]$ and linear on $]1/\lambda, +\infty[$ with a slope λ. Therefore, the derivative is:

$$\nabla f_\lambda(u) = \begin{cases} 0 & \text{if } u \le 1/\lambda, \\ \lambda & \text{if } u > 1/\lambda, \end{cases} = \lambda \mathrm{sign}\,(u - 1/\lambda). \tag{B.10}$$

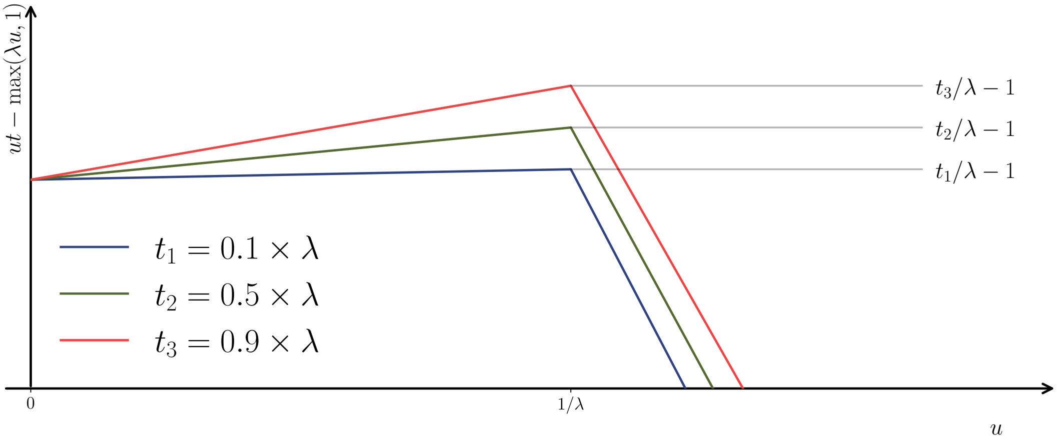


Fig. B.1.1.: Illustration of the computation of f_λ^*.

and finally, the optimal discriminator is:

$$T^{\text{opt}}(\boldsymbol{x}) = \lambda \text{sign}\left(\lambda \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} - 1\right). \tag{B.11}$$

- Then we can compute the reverse $\mathcal{D}_{\lambda\text{-PR}}$:

$$\begin{aligned}
\mathcal{D}_{\lambda\text{-PR}}(\widehat{P}\|P) &= \int_{\mathcal{X}} p(\boldsymbol{x}) f_\lambda\left(\frac{\widehat{p}(\boldsymbol{x})}{p(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x} && \text{(B.12)}\\
&= \int_{\mathcal{X}} \max(\lambda \widehat{p}(\boldsymbol{x}), p(\boldsymbol{x})) - p(\boldsymbol{x}) \max(\lambda, 1)\, \mathrm{d}\boldsymbol{x} && \text{(B.13)}\\
&= \lambda \left(\int_{\mathcal{X}} \max(\widehat{p}(\boldsymbol{x}), p(\boldsymbol{x})/\lambda) \mathrm{d}\boldsymbol{x} - \max(1, 1/\lambda) \right) && \text{(B.14)}\\
&= \lambda \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \max(1, \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}/\lambda) - \widehat{p}(\boldsymbol{x}) \max(1, 1/\lambda)\, \mathrm{d}\boldsymbol{x} && \text{(B.15)}\\
&= \lambda \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_{1/\lambda}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x} && \text{(B.16)}\\
&= \lambda \mathcal{D}_{\frac{1}{\lambda}\text{-PR}}(P\|\widehat{P}). && \text{(B.17)}
\end{aligned}$$

- Building on these results, we can show that :

$$\begin{aligned}
\mathcal{D}_{\text{TV}}(P\|\widehat{P}) &= \int_{\mathcal{X}} |p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})|\, \mathrm{d}\mu(\boldsymbol{x}) && \text{(B.18)}\\
&= \int_{\mathcal{X}} \max(p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}), 0) + \max(\widehat{p}(\boldsymbol{x}) - p(\boldsymbol{x}), 0) \mathrm{d}\mu(\boldsymbol{x}) && \text{(B.19)}
\end{aligned}$$

Then since $\mathcal{D}_{1\text{-PR}}(P\|\widehat{P}) = \int_{\mathcal{X}} \max(\widehat{p}(\boldsymbol{x}), p(\boldsymbol{x})) - p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{\mathcal{X}} \max(\widehat{p}(\boldsymbol{x}) - p(\boldsymbol{x}), 0) \mathrm{d}\boldsymbol{x}$ and $\mathcal{D}_{1\text{-PR}}(P\|\widehat{P}) = \mathcal{D}_{1\text{-PR}}(\widehat{P}\|P)$, we have:

$$\begin{aligned}
\mathcal{D}_{\text{TV}}(P\|\widehat{P}) &= \mathcal{D}_{1\text{-PR}}(P\|\widehat{P}) + \mathcal{D}_{1\text{-PR}}(\widehat{P}\|P) && \text{(B.20)}\\
&= 2\mathcal{D}_{1\text{-PR}}(P\|\widehat{P}). && \text{(B.21)}
\end{aligned}$$

B.1.3 Proof of Theorem 4.2.4

Theorem (PR-Curves as a function of $\mathcal{D}_{\lambda\text{-PR}}$)**.**
Given $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, the PR-Curve ∂PRD is related to the PR-Divergence $\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})$ as follows.

$$\alpha_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}). \tag{B.22}$$

$$\beta_\lambda(P\|\widehat{P}) = \min(1, \lambda) - \mathcal{D}_{\lambda\text{-PR}}(\widehat{P}\|P). \tag{B.23}$$

Conversely, suppose that there exists a strictly decreasing linear function $h : [0,1] \to \mathbb{R}^+$ and an f-divergence $\mathcal{D}_f$ such that $h(\alpha_\lambda(P\| \widehat{P})) = \mathcal{D}_f(P\|\widehat{P})$ for all $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$, then $f(u) = c_1 f_\lambda(u) + c_2(u-1)$.

Proof. Every point of the PR-Curves $(\alpha_\lambda, \beta_\lambda)$ can solely be determined by $\alpha_\lambda(P\|\widehat{P})$ since $\beta_\lambda(P\|\widehat{P}) = \lambda\alpha_\lambda(P\|\widehat{P})$. We have to prove that α_λ can be written as a function of an f-divergence for any $\lambda \in [0, +\infty]$. First for $\lambda = +\infty$, we have the following:

$$\mathcal{D}_{\infty-\mathrm{PR}}(P\|\widehat{P}) = \mathbb{E}_{\widehat{P}}\left[\mathbb{1}_{\left\{\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}=0\right\}}\right] \tag{B.24}$$

$$= \mathbb{E}_{\widehat{P}}\left[\mathbb{1}_{\{p(\boldsymbol{x})=0\}}\right] \tag{B.25}$$

$$= \widehat{P}(\overline{\mathrm{Supp}(P)}), \tag{B.26}$$

where $\overline{\mathrm{Supp}(P)}$ is the complement of the support of P. Consequently, $1 - \widehat{P}(\overline{\mathrm{Supp}(P)}) = \widehat{P}(\mathrm{Supp}(P)) = \alpha_{+\infty}(P\|\widehat{P})$. Then for $\lambda \in [0, +\infty[$, we can develop the expression of α_λ:

$$\alpha_\lambda = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.27}$$

$$= \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \min\left(\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.28}$$

For this integral to be considered as an f-divergence, we need f to be first convex lower semi-continuous and then to satisfy $f(1) = 0$. However, for every $a, b \in \mathbb{R}$, the min satisfies $\min(a,b) = a + b - \max(a,b)$. Therefore,

$$\alpha_\lambda = \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \left[\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} + 1 - \max\left(\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right)\right] \mathrm{d}\mu(\boldsymbol{x}) \tag{B.29}$$

$$= \lambda \int_{\mathcal{X}} p(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}) + 1 - \int_{\mathcal{X}} \max\left(\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.30}$$

$$= \lambda + 1 - \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \max\left(\lambda\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}, 1\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.31}$$

Thus, we take $f_\lambda(u) = \max(\lambda u, 1) - \max(\lambda, 1)$ defined Definition 4.2.1. The precision becomes:

$$\alpha_\lambda = \lambda + 1 - \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) - \max(\lambda, 1)\int_{\mathcal{X}} \widehat{p}(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}) \tag{B.32}$$

$$= \min(\lambda, 1) - \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}) = \min(\lambda, 1) - \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}). \tag{B.33}$$

Consequently, α_λ can be written as a function of an f-divergence $\mathcal{D}_{\lambda\text{-PR}}$ with $f(u) = \max(\lambda u, 1) - \max(\lambda, 1)$. Now we prove the converse. Suppose there exists a strictly decreasing linear function $h : [0,1] \to \mathbb{R}^+$ and an f-divergence $\mathcal{D}_f$ such that $h(\alpha_\lambda(P\|\widehat{P})) = \mathcal{D}_f(P\|\widehat{P})$ for all $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$.

For $P = \widehat{P}$, we get from the definition of α_λ that $\alpha_\lambda(P\|P) = \min(\lambda, 1)$. Hence,

$$0 = \mathcal{D}_f(P\|P) = h(\alpha_\lambda(P\|P)) = h(\min(\lambda, 1)). \tag{B.34}$$

Combining the above with the fact that h is a strictly decreasing linear function, we see that for any fixed λ, h must be of the form, $h(u) = c_\lambda(\min(\lambda, 1) - u)$, where $c_\lambda > 0$ is a constant. Now,

$$\mathcal{D}_f(P\|\widehat{P}) = h(\alpha_\lambda(P\|\widehat{P})) = c_\lambda\left[\min(\lambda, 1) - \alpha_\lambda(P\|\widehat{P})\right] = c_\lambda \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}), \tag{B.35}$$

where the last equality follows from the first part of the theorem, which shows that $\alpha_\lambda(P\|\widehat{P}) = \min(\lambda, 1) - \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P})$. Rewriting the above inequality, we get the following.

$$\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) = \frac{1}{c_\lambda}\mathcal{D}_f(P\|\widehat{P}) = \mathcal{D}_{\frac{1}{c_\lambda}f}(P\|\widehat{P}). \tag{B.36}$$

By the uniqueness theorem of f-divergence $f(u) = \frac{c_1}{c_\lambda} f_\lambda(u) + c_2(u-1)$ for some constants $c_1, c_2 \in \mathbb{R}$.

B.1.4 Proof of Theorem 4.3.1

Theorem (f-divergence as weighted sums of PR-divergences).
For any $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$. If the generator function f is twice differentiable, then:

$$\mathcal{D}_f(P\|\widehat{P}) = \int_0^\infty \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) \mathrm{d}\lambda, \tag{B.37}$$

Proof. Let $c : [0, +\infty[\mapsto]-\infty, +\infty]$ be a $\mathcal{C}^2$ function . The goal is to express any $f(u)$ for all $u \in [0, +\infty[$ as a weighted average of $f_\lambda^{\mathrm{PR}}(u)$ over $\lambda \in [0, 1/u_{\min}]$:

$$\forall u \in [0, +\infty[, \int_0^\infty c''(\lambda) f_\lambda(u) \mathrm{d}\lambda = \int_0^\infty c''(\lambda)\left[\max(\lambda u, 1) - \max(\lambda, 1)\right] \mathrm{d}\lambda \tag{B.38}$$

We can split the integrals to evaluate $\max(\lambda u, 1)$ and $\max(\lambda, 1)$:

$$\begin{aligned}\int_0^\infty c''(\lambda) f_\lambda(u) \mathrm{d}\lambda = &\int_0^\infty c''(\lambda) \max(\lambda u, 1) \mathrm{d}\lambda \\ &- \int_0^\infty c''(\lambda) \max(\lambda, 1) \mathrm{d}\lambda\end{aligned} \tag{B.39}$$

$$
\begin{aligned}
&= \int_0^{1/u} c''(\lambda) \max(\lambda u, 1) \mathrm{d}\lambda \\
&\quad + \int_{1/u}^{\infty} c''(\lambda) \max(\lambda u, 1) \mathrm{d}\lambda \\
&\quad - \int_0^{1} c''(\lambda) \max\left(\lambda, 1\right) \mathrm{d}\lambda \\
&\quad - \int_1^{\infty} c''(\lambda) \max\left(\lambda, 1\right) \mathrm{d}\lambda
\end{aligned}
\tag{B.40}
$$

$$
\begin{aligned}
&= \int_0^{1/u} c''(\lambda) \mathrm{d}\lambda \\
&\quad + u \int_{1/u}^{\infty} c''(\lambda)\lambda \mathrm{d}\lambda \\
&\quad - \int_0^{1} c''(\lambda) \mathrm{d}\lambda \\
&\quad - \int_1^{\infty} c''(\lambda)\lambda \mathrm{d}\lambda.
\end{aligned}
\tag{B.41}
$$

Integrating by part for any $a, b \in \mathbb{R}$, we have: $\int_a^b c''(\lambda)\lambda \mathrm{d}\lambda = [c'(\lambda)\lambda]_a^b - \int_a^b c'(\lambda)\mathrm{d}\lambda$. Thus, it satisfies:

$$
\begin{aligned}
\int_0^{\infty} c''(\lambda) f_\lambda^{\mathrm{PR}}(u) \mathrm{d}\lambda &= \int_0^{1/u} c''(\lambda) \mathrm{d}\lambda \\
&\quad + u \left[c'(\lambda)\lambda\right]_{1/u}^{\infty} - u \int_{1/u}^{\infty} c'(\lambda) \mathrm{d}\lambda \\
&\quad - \int_0^{1} c''(\lambda) \mathrm{d}\lambda \\
&\quad - \left[c'(\lambda)\lambda\right]_1^{\infty} + \int_1^{\infty} c'(\lambda) \mathrm{d}\lambda
\end{aligned}
\tag{B.42}
$$

$$
\begin{aligned}
&= \left[c'(\lambda)\right]_0^{1/u} \\
&\quad + u \left[c'(\lambda)\lambda\right]_{1/u}^{\infty} - u \left[c(\lambda)\right]_{1/u}^{\infty} \\
&\quad - \left[c'(\lambda)\right]_0^{1} \\
&\quad - \left[c'(\lambda)\lambda\right]_1^{\infty} + \left[c(\lambda)\right]_1^{\infty}
\end{aligned}
\tag{B.43}
$$

$$
\begin{aligned}
&= c'\left(\frac{1}{u}\right) - c'(0) \\
&\quad + u \lim_{v \to \infty} c'\left(v\right) v - uc'\left(\frac{1}{u}\right) \frac{1}{u} - u \lim_{v \to \infty} c\left(v\right) + uc\left(\frac{1}{u}\right) \\
&\quad - c'(1) + c'\left(0\right) \\
&\quad - \lim_{v \to \infty} c'\left(v\right) v + c'(1) \times 1 + \lim_{v \to \infty} c\left(v\right) - c(1)
\end{aligned}
\tag{B.44}
$$

$$
\begin{aligned}
&= \left[\lim_{v \to \infty} \left(c'\left(v\right) v - c\left(v\right)\right)\right] (u - 1) \\
&\quad + uc\left(\frac{1}{u}\right) - c(1).
\end{aligned}
\tag{B.45}
$$

We would like $\int_0^{\infty} c''(\lambda) f_\lambda^{\mathrm{PR}}(u) \mathrm{d}\lambda$ to be equal to f on $[0, +\infty[$. Since the two f-divergences generated by f and g are equal if there exists a $\gamma \in \mathbb{R}$ such that $f(u) = g(u) + \gamma(u - 1)$, the divergence generated by $u \mapsto \int_0^{\infty} c''(\lambda) f_\lambda^{\mathrm{PR}}(u) \mathrm{d}\lambda$ is equal to the divergence generated by $u \mapsto uc\left(\frac{1}{u}\right) - c(1)$. Therefore, we require the function c to satisfy:

$$\forall u \in [0, +\infty], \quad f(u) = uc\left(\frac{1}{u}\right) - c(1).$$

Differentiating with respect to u, we have:

$$f'(u) = c\left(\frac{1}{u}\right) - \frac{1}{u}c'\left(\frac{1}{u}\right). \tag{B.46}$$

And finally:

$$\begin{aligned} f''(u) &= -\frac{1}{u^2}c\left(\frac{1}{u}\right) + \frac{1}{u^2}c'\left(\frac{1}{u}\right) + \frac{1}{u^3}c''\left(\frac{1}{u}\right) && \text{(B.47)} \\ &= \frac{1}{u^3}c''\left(\frac{1}{u}\right). && \text{(B.48)} \end{aligned}$$

Consequently, with $\lambda = 1/u$, we have that:

$$\forall \lambda \in [0, +\infty], \quad c''(\lambda) = \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right). \tag{B.49}$$

With such a results we can write any f-divergence as:

$$\begin{aligned} \mathcal{D}_f(P\|\widehat{P}) &= \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x}) \\ &= \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) \int_0^M \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) f_\lambda^{\mathrm{PR}}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\lambda \mathrm{d}\mu(\boldsymbol{x}) \\ &= \int_0^M \int_{\mathcal{X}} \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \widehat{p}(\boldsymbol{x}) f_\lambda^{\mathrm{PR}}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\lambda \mathrm{d}\mu(\boldsymbol{x}) \\ &= \int_0^M \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \left[\int_{\mathcal{X}} \widehat{p}(\boldsymbol{x}) f_\lambda^{\mathrm{PR}}\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right) \mathrm{d}\mu(\boldsymbol{x})\right] \mathrm{d}\lambda \\ &= \int_0^M \frac{1}{\lambda^3} f''\left(\frac{1}{\lambda}\right) \mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) \mathrm{d}\lambda. \end{aligned}$$

B.1.5 Proof of Lemma 4.4.1

Lemma B.1.1 (Probabilistic Formulation of PR-Divergence)**.**
For any $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ such that $P, \widehat{P} \ll \mu$ and $\lambda \in [0, +\infty]$, the PR-Divergence can be expressed as:

$$\mathcal{D}_{\lambda\text{-PR}}(P\|\widehat{P}) = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left|\lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A})\right| - |\lambda - 1|. \tag{B.50}$$

Proof. First, we show that for any $\lambda \in [0, +\infty]$:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \frac{1}{2}\int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})| \,\mathrm{d}\mu(\boldsymbol{x}) - \frac{1}{2}|\lambda - 1|. \tag{B.51}$$

We have that:

$$\begin{aligned}
\frac{1}{2}\int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})| \,\mathrm{d}\mu(\boldsymbol{x}) &= \frac{1}{2}\int_{\mathcal{X}} \max(\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}), 0) \qquad \text{(B.52)}\\
&\qquad + \max(\widehat{p}(\boldsymbol{x}) - \lambda p(\boldsymbol{x}), 0)\mathrm{d}\mu(\boldsymbol{x})\\
&= \int_{\mathcal{X}} \max(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x}))\mathrm{d}\mu(\boldsymbol{x}) \qquad \text{(B.53)}\\
&\qquad - \frac{1}{2}\left(\lambda \int_{\mathcal{X}} p(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x}) + \int_{\mathcal{X}} \widehat{p}(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x})\right)\\
&= \int_{\mathcal{X}} \max(\lambda p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x}))\mathrm{d}\mu(\boldsymbol{x}) - \max(\lambda, 1)\mathrm{d}\mu(\boldsymbol{x})\\
&\qquad + \frac{1}{2}\left(2\max(\lambda, 1) - \lambda - 1\right) \qquad \text{(B.54)}\\
&= \mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) + \frac{1}{2}|\lambda - 1|. \qquad \text{(B.55)}
\end{aligned}$$

Then, we can prove that for any distributions $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ and $\lambda \in [0, +\infty]$, we have:

$$\frac{1}{2}\int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{x})| = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left|\lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A})\right| - \frac{1}{2}|\lambda - 1|. \tag{B.56}$$

As a matter of fact, let $\mathcal{B} = \{\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x}) \geq 0\}$. Then we have:

$$\int_{\mathcal{B}} \lambda p - \widehat{p}\mathrm{d}\mu = \int_{\mathcal{X}\backslash\mathcal{B}} \widehat{p} - \lambda p \mathrm{d}\mu + \lambda - 1. \tag{B.57}$$

Therefore, on one side, we can write that:

$$\begin{aligned}
\int_{\mathcal{X}} |\lambda p(\boldsymbol{x}) - \widehat{p}(\boldsymbol{x})|\mathrm{d}\mu &= 2\int_{\mathcal{B}} \lambda p - \widehat{p}\mathrm{d}\mu + -1\lambda \qquad \text{(B.58)}\\
&\leq 2\sup_{\mathcal{A} \subseteq \mathcal{X}} \left|\int_{\mathcal{A}} \lambda p - \widehat{p}\mathrm{d}\mu\right| - |\lambda - 1|, \qquad \text{(B.59)}
\end{aligned}$$

since the supremum is reached for $\mathcal{A} = \mathcal{B}$ or $\mathcal{A} = \mathcal{X}\backslash\mathcal{B}$. Then, on the other side, we have for any $\mathcal{A} \subseteq \mathcal{X}$:

$$\begin{aligned}
\left|\int_{\mathcal{A}} \lambda p - \widehat{p}\mathrm{d}\mu\right| &= \max\left(\int_{\mathcal{A}} \lambda p - \widehat{p}\mathrm{d}\mu, \int_{\mathcal{A}} \widehat{p} - \lambda p\mathrm{d}\mu\right) \qquad \text{(B.60)}\\
&\leq \max\left(\int_{\mathcal{B}} \lambda p - \widehat{p}\mathrm{d}\mu, \int_{\mathcal{X}\backslash\mathcal{B}} \widehat{p} - \lambda p\mathrm{d}\mu\right) \qquad \text{(B.61)}\\
&\leq \max\left(\int_{\mathcal{B}} \lambda p - \widehat{p}\mathrm{d}\mu, \int_{\mathcal{B}} \lambda p - \widehat{p}\mathrm{d}\mu + 1 - \lambda\right) \qquad \text{(B.62)}\\
&\leq \max\left(\frac{1}{2}\int_{\mathcal{X}} |\lambda p - \widehat{p}| \,\mathrm{d}\mu + \frac{1}{2}|\lambda - 1|, \frac{1}{2}\int_{\mathcal{X}} |\lambda p - \widehat{p}| \,\mathrm{d}\mu - \frac{1}{2}|\lambda - 1|\right) \qquad \text{(B.63)}
\end{aligned}$$

$$= \frac{1}{2} \int_{\mathcal{X}} |\lambda p - \widehat{p}| \, \mathrm{d}\mu + \frac{1}{2} |\lambda - 1|. \tag{B.64}$$

Therefore, we have the result:

$$\sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \int_{\mathcal{A}} \lambda p - \widehat{p} \mathrm{d}\mu \right| - \frac{|\lambda - 1|}{2} \leq \frac{1}{2} \int_{\mathcal{X}} |\lambda p - \widehat{p}| \, \mathrm{d}\mu \leq \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \int_{\mathcal{A}} \lambda p - \widehat{p} \mathrm{d}\mu \right| - \frac{|\lambda - 1|}{2}. \tag{B.65}$$

Consequently, we can show Equation (B.56). Finally, combining Equations (B.51) and (4.37), we have the result:

$$\mathcal{D}_{\lambda\text{-PR}}(P \| \widehat{P}) = \sup_{\mathcal{A} \subseteq \mathcal{X}} \left| \lambda P(\mathcal{A}) - \widehat{P}(\mathcal{A}) \right| - |\lambda - 1| \, . \tag{B.66}$$

B.2 Proofs of Chapter 6

B.2.1 Proof of Theorem 6.2.1

Theorem (Optimal Acceptance Function)**.**
For a sampling budget $K \geq 1$ and finite $\mathcal{X}$, the solution to the problem (6.9) is,

$$a_{\mathrm{OBRS}}(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} \frac{c_K}{M}, 1 \right), \tag{B.67}$$

where $c_K \geq 1$ is such that $\mathbb{E}_{\boldsymbol{x} \sim \widehat{p}}[a_{\mathrm{OBRS}}(\boldsymbol{x})] = 1/K$.

Proof. The goal is to find an acceptance function $a(\boldsymbol{x})$ that first minimizes the f-divergence between the target distribution P and the distribution after the rejection process $\widetilde{P}_a$. With a budget of K, the average acceptance rate is $1/K$. The function a is the solution of the problem:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(P \| \widetilde{P}_a) \\ \text{s.t.} \quad & \begin{cases} \mathbb{P}\,(\text{acceptance}) \geq 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{B.68}$$

First, we can consider $\mathcal{D}_f(\widetilde{P}_a \| P)$ instead of $\mathcal{D}_f(P \| \widetilde{P}_a)$ without loss of generality: This is because $\mathcal{D}_f(P \| \widetilde{P}_a) = \mathcal{D}_{f'}(\widetilde{P}_a \| P)$ for $f' : x \mapsto x f(1/x)$. Further, the solution to the optimal $a(\boldsymbol{x})$ turns out to be independent of f.

Moreover, we can assume that the budget is always lower that the unlimited budget. In other terms, instead of forcing the acceptance rate to be greater to $1/K$ we can force is to be exactly equal to $1/K$. Then, the probability of acceptance being $\mathbb{P}\,(\text{acceptance}) = \mathbb{E}_{\widehat{P}}\,[a(\boldsymbol{x})]$, we can write an equivalent problem as:

$$\begin{aligned} \min_{a} \quad & \mathcal{D}_f(\widetilde{P}_a \| P) \\ \text{s.t.} \quad & \begin{cases} \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{B.69}$$

Using the definition of the densities in the rejection sampling context, $\widetilde{p}_a(\boldsymbol{x}) = K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x})$, the problem is equivalent to:

$$\begin{aligned} \min_{a} \quad & \mathbb{E}_P\left[f\left(\frac{K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x})}{p(\boldsymbol{x})}\right)\right] \\ \text{s.t.} \quad & \begin{cases} \mathbb{E}_{\widehat{P}}\left[a(\boldsymbol{x})\right] = 1/K \\ \forall \boldsymbol{x},\, 0 \leq a(\boldsymbol{x}) \leq 1 \end{cases} \end{aligned} \tag{B.70}$$

Switching to the discrete case, the problem becomes :

$$\begin{aligned} \min_{\boldsymbol{a} \in \mathbb{R}^N} \quad & \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) \\ \text{s.t.} \quad & \begin{cases} \sum_i^N \widehat{p}_i a_i = 1/K \\ \forall i,\, 0 \leq a_i \leq 1 \end{cases} \end{aligned} \tag{B.71}$$

The Lagrangian function associated with the problem B.71 is:

$$\mathcal{L}(\boldsymbol{a}, \mu, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = \sum_i^N p_i f\left(a_i \frac{\widehat{p}_i K}{p_i}\right) + \mu\left[\boldsymbol{a}^T \widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T \boldsymbol{\lambda}_1 - \boldsymbol{a}^T \boldsymbol{\lambda}_2 \tag{B.72}$$

All constraints are affine and the objective function is a convex function, therefore the optimal vector $\boldsymbol{a}^\star$ satisfies the KKT conditions:

$$\begin{cases} \nabla_{a_i} \mathcal{L}(\boldsymbol{a}^\star, \mu^\star \boldsymbol{\lambda}_1^\star, \boldsymbol{\lambda}_2^\star) = K\widehat{p}_i \nabla f\left(a_i^\star \frac{\widehat{p}_i K}{p_i}\right) + \mu^\star \widehat{p}_i + (\lambda_{1i}^\star - \lambda_{2i}^\star) = 0, \quad \forall i \\ \sum_i a_i^* \widehat{p}_i = 1/K \\ \lambda_{1i}^\star (a_i^\star - 1) = 0, \quad \forall i \\ \lambda_{2i}^\star a_i^\star = 0, \quad \forall i \\ \lambda_{1i}^\star, \lambda_{2i}^\star \geq 0, \forall i \end{cases} \tag{B.73}$$

Using the 1st condition:

$$a_i^\star = \frac{p_i}{\widehat{p}_i K} \left[\nabla f\right]^{-1} \left(\frac{\lambda_{2i}^\star - \lambda_{1i}^\star}{K\widehat{p}_i} - \mu/K\right) \tag{B.74}$$

Since $[\nabla f]^{-1} = \nabla f^*$:

$$a_i^\star = \frac{p_i}{\widehat{p}_i K}\left[\nabla f^*\right]\left(\frac{\lambda_{2i}^\star - \lambda_{1i}^\star}{\widehat{p}_i K} - \mu/K\right) \tag{B.75}$$

All the usual f^* are strictly increasing functions. Therefore, according to Eq B.75, all $a_i > 0$. Thus all $\lambda_{2i}^\star = 0$. The KKT conditions B.73 become :

$$\begin{cases} K\widehat{p}_i\nabla f\left(a_i^\star\frac{\widehat{p}_iK}{p_i}\right) + \mu^\star\widehat{p}_i + \lambda_{1i}^\star = 0, \quad \forall i \\ \sum_i a_i^\star\widehat{p}_i = 1/K \\ \lambda_{1i}^\star(a_i^\star - 1) = 0, \forall i \\ \lambda_{1i}^\star \geq 0, \forall i \end{cases} \tag{B.76}$$

And thus :

$$a_i^\star = \frac{p_i}{\widehat{p}_i K}\left[\nabla f^*\right]\left(-\frac{\lambda_{1i}^\star}{\widehat{p}_i K} - \mu/K\right) \tag{B.77}$$

To get the full formula for $a_i^\star$, we need to compute the λ_{1i}s. For this purpose, let us use strong duality to reformulate our problem:

$$\min_{\boldsymbol{a}}\max_{\boldsymbol{\lambda}\geq\boldsymbol{0},\boldsymbol{\mu}}\sum_i^N p_i f\left(a_i\frac{\widehat{p}_iK}{p_i}\right) + \mu\left[\boldsymbol{a}^T\widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T\boldsymbol{\lambda}_1 \tag{B.78}$$

$$= \max_{\boldsymbol{\lambda}\geq\boldsymbol{0},\boldsymbol{\mu}}\min_{\boldsymbol{a}}\sum_i^N p_i f\left(a_i\frac{\widehat{p}_iK}{p_i}\right) + \mu\left[\boldsymbol{a}^T\widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T\boldsymbol{\lambda}_1 \tag{B.79}$$

Then, we can use the Fenchel Conjugate:

$$\begin{aligned} \min_{\boldsymbol{a}}\sum_i^N p_i^* f\left(a_i\frac{\widehat{p}_iK}{p_i^*}\right) &+ \mu\left[\boldsymbol{a}^T\widehat{\boldsymbol{p}} - 1/K\right] + (\boldsymbol{a} - \mathbb{1})^T\boldsymbol{\lambda}_1 \\ &= \min_{\boldsymbol{a}}\sum_i^N p_i^*\left[f\left(a_i\frac{\widehat{p}_iK}{p_i^*}\right) - a_i\left(\frac{-\mu\widehat{p}_i - \lambda_{1i}}{p_i}\right)\right] \\ &\qquad - \mu/K - \mathbb{1}^T\boldsymbol{\lambda}_1 \\ &= -\sup_{\boldsymbol{a}}\left\{\sum_i^N p_i^*\left[a_i\left(\frac{-\mu\widehat{p}_i - \lambda_{1i}}{p_i}\right) - f\left(a_i\frac{\widehat{p}_iK}{p_i^*}\right)\right]\right\} \\ &\qquad - \mu/K - \mathbb{1}^T\boldsymbol{\lambda}_1 \\ &= -\sum_i^N\left[p_i^* f^*\left(-\frac{p_i^*}{\widehat{p}_iK}\frac{\mu\widehat{p}_i + \lambda_{1i}}{p_i}\right)\right] - \mu/K - \mathbb{1}^T\boldsymbol{\lambda}_1 \\ &= -\sum_i^N\left[p_i^* f^*\left(-\mu/K - \frac{\lambda_{1i}}{\widehat{p}_iK}\right)\right] - \mu/K - \mathbb{1}^T\boldsymbol{\lambda}_1 \end{aligned} \tag{B.80}$$

Define $u_i = \frac{\lambda_{i1}}{\hat{p}_i}$, assuming $\hat{p}_i > 0$ everywhere. Note that the constraints $\lambda_{i1} \geq 0$ and $u_i \geq 0$ are equivalent. The above equation becomes

$$\sup_{\lambda_1 \geq 0} \mathcal{L}\left(\boldsymbol{a}^\star, \mu^\star, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2^\star\right) = \sup_{\boldsymbol{u} \geq 0} -\sum_i^N p_i^* f^\star\left(-\left(\mu^\star + u_i\right)/K\right) - \sum_i^N \hat{p}_i u_i - \mu^\star/K \tag{B.81}$$

Let us make another change of variable to make a conjugate form appear. Define $v_i = -\left(\mu^\star + v_i\right)$. So $u_i = -\mu^\star - v_i$ and the constraint $u_i \geq 0$ becomes $v_i \leq -\mu^\star$. Also, define $g(t) = f(Kt)$. Then $g^*(t) = f^*(\frac{t}{K})$. Above equation becomes

$$\sup_{\boldsymbol{\lambda}_1 \geq 0} \mathcal{L}\left(\boldsymbol{a}^\star, \mu^\star, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2^\star\right) = \sup_{\boldsymbol{v} \leq -\mu^\star} \sum_i^N \hat{p}_i v_i - \sum_i^N p_i g^*\left(v_i\right) - \mu^\star\left(K - 1\right) \tag{B.82}$$

Recall that $\arg\sup_t \langle a, t\rangle - f(t) = \nabla f^*(a)$ and $\arg\sup_t \langle a, t\rangle - f^*(t) = \nabla f(a)$. Thus, given $\mu^\star$ we can compute the optimal values of v_i one by one as follows:

$$\begin{aligned}
v_i^\star &= \arg \sup_{v_i \leq -\mu^\star} \hat{p}_i v_i - p_i g^*\left(v_i\right) \\
&= \arg \sup_{v_i \leq -\mu^\star} \frac{\hat{p}_i}{p_i} v_i - g^*\left(v_i\right) \\
&= \min\left(-\mu^\star, \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right)
\end{aligned}$$

So $u_i^\star = \max\left(0, -\mu^\star - \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right)$. This gives us the optimal values of $\lambda_{i1}^\star$. Note that $\nabla g(t) = K \nabla f\left(Kt\right)$. Replacing $\frac{\lambda_{1i}^\star}{\hat{p}_i}$ by $u_i^\star$ in the formula of $a_i^\star$ gives us:

$$\begin{aligned}
a_i^\star &= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(-\mu^\star/K - \max\left(0, -\mu^\star - \nabla g\left(\frac{\hat{p}_i}{p_i}\right)/K\right)\right) \\
&= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(-\mu^\star/K + \min\left(0, \mu^\star + \nabla g\left(\frac{\hat{p}_i}{p_i}\right)/K\right)\right) \\
&= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(\min\left(-\mu^\star, \nabla g\left(\frac{\hat{p}_i}{p_i}\right)\right)/K\right) \\
&= \frac{p_i}{\hat{p}_i K} \nabla f^*\left(\min\left(-\mu^\star/K, \nabla f\left(\frac{\hat{p}_i K}{p_i}\right)\right)\right).
\end{aligned}$$

Note that ∇f^* is strictly increasing, thus:

$$\begin{aligned}
a_i^\star &= \frac{p_i}{\hat{p}_i K} \min\left(\nabla f^*\left(-\frac{\mu^\star}{K}\right), \frac{\hat{p}_i K}{p_i}\right) \\
&= \min\left(\frac{p_i}{\hat{p}_i K} \nabla f^*\left(-K\mu^\star\right), 1\right).
\end{aligned}$$

Note that $\nabla f^*\left(-\mu^\star/K\right)$ is a constant. So the optimal acceptance function under budget looks like $a(\boldsymbol{x}) = \min\left(1, c\frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})}\right)$ for some constant c defined by K only as:

$$\int_{\mathcal{X}} \min\left(\widehat{p}(\boldsymbol{x}), cp(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) = 1/K. \tag{B.83}$$

To facilitate the understanding of c, we can set this constant to be equal to c/M instead. Thus,

$$a(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{c}{M}, 1\right). \tag{B.84}$$

With that notation, $c \geq 1$ and if the optimal unlimited acceptance function is obtained with $c = 1$:

$$a(\boldsymbol{x}) = \min\left(\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\frac{1}{M}, 1\right) = \frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})M}, \tag{B.85}$$

which concludes the proof.

B.2.2 Proof of Theorem 6.2.3

Theorem (Precision and Recall Improvement)**.**
Let $P, \widehat{P} \in \mathcal{P}(\mathcal{X})$ be two distributions such that $P, \widehat{P} \ll \mu$ and a_{OBRS} be the optimal acceptance function for a budget K defined in Theorem 6.2.1. For any $(\alpha, \beta) \in \mathrm{PRD}(P, \widehat{P})$ we have $(\alpha', \beta) \in \mathrm{PRD}(P, \widetilde{P}_{a_{\mathrm{OBRS}}})$ with $\alpha' = \min\{1, K\alpha\}$.

Proof. First, with $a(\boldsymbol{x}) = \min\left(1, \frac{c_k}{M}\frac{p(\boldsymbol{x})}{\widehat{p}(\boldsymbol{x})}\right)$, let us recall that

$$\begin{aligned}
\widetilde{p}_a(\boldsymbol{x}) &= K\widehat{p}(\boldsymbol{x})a(\boldsymbol{x}) && \text{(B.86)}\\
&= \min\left(K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right). && \text{(B.87)}
\end{aligned}$$

Thus:

$$\begin{aligned}
\alpha_\lambda(P\|\widetilde{P}_a) &= \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), \widetilde{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) && \text{(B.88)}\\
&= \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}), K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}). && \text{(B.89)}
\end{aligned}$$

For $\lambda \geq Kc_K/M$:

$$\begin{aligned}
\alpha_\lambda(P\|\widetilde{P}_a) &= \int_{\mathcal{X}} \min\left(K\widehat{p}(\boldsymbol{x}), \frac{Kc_K}{M}p(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) && \text{(B.90)}\\
&= K\int_{\mathcal{X}} \min\left(\frac{c_K}{M}p(\boldsymbol{x}), \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) && \text{(B.91)}\\
&= K\mathbb{E}_{\widehat{P}}\left[\min\left(\frac{c_K}{M}\frac{p(\boldsymbol{x})}{p(\boldsymbol{x})}, 1\right)\right] && \text{(B.92)}\\
&= K\frac{1}{K} \quad \text{by definition of } c_K, && \text{(B.93)}\\
&= 1. && \text{(B.94)}
\end{aligned}$$

Thus, under a given threshold Kc_K/M, the precision is constant and equal to 1. For $\lambda \leq Kc_K/M$:

$$\alpha_\lambda(P\|\widetilde{P}_a) = \int_{\mathcal{X}} \min\left(\lambda p(\boldsymbol{x}),\, K\widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.95}$$

$$= K \int_{\mathcal{X}} \min\left(\frac{\lambda}{K} p(\boldsymbol{x}),\, \widehat{p}(\boldsymbol{x})\right) \mathrm{d}\mu(\boldsymbol{x}) \tag{B.96}$$

$$= K\alpha_{\lambda/K}(P\|\widehat{P}). \tag{B.97}$$

Finally, with $\alpha_\lambda = \lambda\beta_\lambda$,

$$\beta_\lambda(P\|\widetilde{P}_a) = \frac{K}{\lambda}\alpha_{\lambda/K}(P\|\widehat{P}) = \frac{K}{(\lambda)}\frac{\lambda}{K}\beta_{\lambda/K}(P\|\widehat{P}) = \beta_{\lambda/K}(P\|\widehat{P}), \tag{B.98}$$

Therefore we have two regimes:

- For $\lambda \geq \frac{Kc_K}{M}$:

$$\alpha_\lambda\left(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}\right) = 1 \quad \text{and} \quad \beta_\lambda\left(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}\right) = 1/\lambda$$

- For $\lambda \leq \frac{Kc_K}{M}$:

$$\begin{cases} \alpha_\lambda(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}) = K\alpha_{\lambda/K}(P\|\widehat{P}) \\ \beta_\lambda(P\|\widetilde{P}_{a_{\mathrm{OBRS}}}) = \beta_{\lambda/K}(P\|\widehat{P}) \end{cases}$$

This can be seen as a vertical scaling of the PR-Curve. For a given point (α, β) in $\mathrm{PRD}(P\|\widehat{P})$, then the point with the same β in $\mathrm{PRD}(P\|\widetilde{P})$ has a Precision $K\alpha$, up to a certain saturating level ($\alpha < 1$).

Experiment Details

C

Contents

In this appendix, we provide additional details on the experiments conducted in Chapter 5 and Chapter 6. We focus on the implementation details such the hyperparameters used for training and the architectures. We will discuss the experiments in the order they are presented in the main text:

- In Appendix C.1.1, we present the naive approach to tune the quality and diversity using the PR-Divergence discussed in Section 5.1 (page 76).
- In Appendix C.1.2, we present the experiments conducted on 2D Gaussians using RealNVP discussed in Section 5.3.1 (page 87).
- In Appendix C.1.3, we present the experiments conducted on MNIST and FashionMNIST using GLOW discussed in Section 5.3.2 (page 88).
- In Appendix C.1.4 and Appendix C.1.5, we present the experiments conducted with BigGANs discussed in Section 5.3.3 (page 91).
- In Appendix C.2.1 and Appendix C.2.2, we present the rejection sampling on 2D Gaussians, CelebaA and CIFAR-10 discussed in Section 6.2.3 (page 108).
- In Appendix C.2.3, we present the parameter landscape of a GAN trained on MNIST discussed in Section 6.3.1 (page 113).
- In Appendix C.2.4, we present the experiments conducted on BigGAN using the OBRS discussed in Section 6.3.3 (page 119).

The code for the BigGAN experiments is available at `https://github.com/AlexVerine/PrecisionRecallGan` and at `https://github.com/AlexVerine/RejectBigGan`.

C.1 Experiments in Chapter 5

C.1.1 Naive Approach with BigGAN

The goal of this experiment is to show that using the original framework of f-GAN applied to the Precision-Recall Divergence does not work as expected. To do so, we use the BigGAN architecture as it is a well-known model, among the SOTA models and easy to train. We have used the official Github repository[1] of Brock et al. [16] to train BigGAN with PyTorch [95], and we have changed the original hinge loss in order to fit the f-GAN framework. We have run our experiment training the same BigGAN with the same hyperparameters as the original paper: *Adam* optimizer, learning rates of 5.10^{-5} and 2.10^{-4} for the generator and the discriminator, batch size of 128. To make sure that the model is failing to train because of the PR Divergence and not any f-divergence, we have trained several models with different f-divergence (χ^2, Kullback-Leibler, Jensen-Shannon and Total Variation) with learning rates from 10^{-5} to 10^{-3} and several seeds, and we have observed the same behavior: models trained with the hinge loss, χ^2, Kullback-Leibler and Jensen-Shannon divergences are able to train even if some training instability is observed (early mode collapse), and models trained with the PR Divergence (and the TV) are not able to train at all.

C.1.2 RealNVP on 2D Gaussians

In Section 5.3.1, we have show how a model can be tuned by minimizing the PR-Divergence using our approach. To do so, we choose a model with low expressivity. Moreover, to avoid the mode collapse that can occurs easily in smaller dimension, we chose to use a Normalizing Flows trained for a few iterations with the MLE objective and then use our approach to specifically minimize any PR-Divergence. To do so, we use a RealNVP [31] for the generator G. We use an 8-coupling step composed of each of 2 linear layers 2-256-2 with LeakyRelu activation in between. For the discriminator, we used a 4 linear layers 2-1024-512-256-1 neural network with LeakyRelu activation between layers. For both, we use Adam optimizer with a learning rate of 2.10^{-5} for G and 1.10^{-4} for T. G has 540k parameters and 660k for T.

[1] https://github.com/ajbrock/BigGAN-PyTorch

C.1.3 GLOW on MNIST and FashionMNIST

In Section 5.3.2, we have shown how a model can be tuned by minimizing the PR-Divergence using our approach on small dimensional real-world datasets such a MNIST and FashionMNIST. The training procedure is relatively close. For both datasets, we use a multiscale GLOW [70]. The model has three levels of processing: images of size $4 \times 16 \times 16$, $16 \times 8 \times 8$ and $64 \times 8 \times 8$. Each level has 16 blocks of affine coupling with 3 layers of 512 channels of convolutional operations, leading to a total of 85.2M parameters. For the discriminator, we use a 1024-1024-512-256-1 linear layers neural network with LeakyRelu activation between layers, with 1.7M parameters. Both are trained with Adam using a learning rate of 1.10^{-5} for T and 1.10^{-6} for G with a batch size of 64. For both dataset, we train a model for 250 epochs using maximum likelihood estimation (MLE) with 4 GPUs V100 (~ 200 hours). The models are then fine-tuned with their different losses on 12 V100 GPUs for 30 epochs (~ 2 hours). For two epochs, we train the discriminator only, and then we train both models alternatively following our approach.

C.1.4 BigGAN on CIFAR-10 and CelebA64

While training large Normalizing Flows using discriminator is not popular in the community, training GANs is a common practice. Therefore, we stick our experiments to popular settings and datasets. We chose the BigGAN as its performance are close to SOTA with a much lower computational cost of training. To do this, we modify the official implementation of PyTorch of BigGAN by Brock et al. [16] to incorporate our method. We use the exact same hyperparameters as the original framework (Adam optimizer, learning rates of 5.10^{-5} and 2.10^{-4} for the generator and the discriminator, batch size of 128). Doing so, G and T respectively count 4.3M and 4.2M parameters for CIFAR-10 and 32.0M and 19.5M for CelebA64. CIFAR-10's models are trained on 4 V100 16 GB GPUs with a batch size of 128 for approximately 100k iterations (~ 7 hours), while CelebA64's models have been trained on 4 V100 32 GB GPUs with a batch size of 128 for 95k iteration (~ 20 hours).

C.1.5 BigGAN on ImageNet128 and FFHQ256

We have also fine-tuned BigGAN on ImageNet128 and FFHQ256. For the pre-train weights of the models, we have used the weights provided by Brock et al. [16] for ImageNet128 and a model trained by us for FFHQ256. For ImageNet128, the generator has 80.0M parameters and the discriminator 90M parameters. The models are trained on 8 A100 80 GB for 10k iterations (~ 2 days) using lower learning rates than the pre-training: 1.10^{-5} for both networks. Similarly for FFHQ256, the generator has 37M parameters and the discriminator 47M parameters. The models are trained on 8 A100 80 GB for 10k iterations (~ 1.5 days) using lower learning rates than the pre-training: 1.10^{-5} for both networks.

C.2 Experiments in Chapter 6

C.2.1 Rejection Algorithms on 2D Gaussians

Similarly to Section 5.3.1, we aim to train a generative model on a synthetic 2D dataset. However, in this section we compare our result to alternative methods and therefore we use the same model as in Che et al. [20] (one of the alternative methods). We use a generator composed of 4 linear layers 2-256-512-1024-2 with LeakyRelu activations for a total of 659k parameters. The discriminator is composed of 4 linear layers 2-1024-512-256-1 with LeakyRelu activations for a total of 659k parameters. We use Adam optimizer with a learning rate of 2.10^{-5} for G and for T. We train the model for 100k iterations on a single GPU with a batch size of 4096 for 4000 epochs (~ 1 hour).

C.2.2 OBRS on BigGAN and EDM

In Section 6.2.3, we have shown how the OBRS can be used to train a generative model. We have used the same BigGAN architecture as in Appendix C.1.4 and Appendix C.1.5 and the same hyperparameters. We have used the pretrained model we have used in baseline in Appendix C.1.4. However, the original BigGAN is trained with the hinge loss which offers no guaranty of the density ratio estimation [9]. Therefore, we fine-tune the discriminator using the Jensen-Shannon divergence with the exact same hyperparameters as the original training except the learning rate of 1.10^{-6}. We have trained the model for 10k iterations on 4 V100 16 GB GPUs (~ 1 hour). For the EDM we have use the discriminator pre-trained by Kim et al. [66] on the diffusion models EDM of Karras et al. [61]. In the latter experiment, no training or fine-tuning is needed.

C.2.3 Parameter Landscape of OBRS

In Section 6.3.1, we have shown the parameter landscape of a GAN trained on MNIST. We have used a shallow generator composed of 4 linear layers 100-256-512-1024-784 with LeakyRelu activations for a total of 1.5M parameters. The discriminator is composed of 4 linear layers 784-1024-512-256-1 with LeakyRelu activations for a total of 1.5M parameters. We use Adam optimizer with a learning rate of 2.10^{-4} for G and for T. We train the model for 35k iterations on 2 A100 80 GB GPUs with a batch size of 512 (~ 3 hours). We freeze the training at 35k iterations to define θ_0 when the training as already converged. Furthermore, we add a random noise to the parameters vector and restart training twice for 5k iterations to defined θ_1 and θ_2. We have used the same hyperparameters as the original training. To compute the loss landscape we use a generator using the same architecture with a parameter vector θ build with θ_0, θ_1 and θ_2. We fine-tune the discriminator with the Jensen-Shannon divergence with a learning rate of 1.10^{-6} for 1k iterations on 2 A100 80 GB GPUs with a batch size of 512 (~ 5 minutes) in order to have a better estimation of the density ratio $p(\boldsymbol{x})/\widehat{p}(\boldsymbol{x})$. Using this discriminator and the generator to generate samples, we train a second generator with one linear layer 784-1 to estimate $p(\boldsymbol{x})/\widetilde{p}(\boldsymbol{x})$ with a learning rate of 2.10^{-6} for 500 iterations on 2 A100 80 GB GPUs with a batch size of 512 (~ 1 minutes). This second discriminator is then used to compute the loss landscape by computing the primal approximation of $\mathcal{D}_{\text{GAN}}(P\|\widetilde{P})$.

C.2.4 Training with OBRS on BigGAN

In Section 6.3.3, we have shown how model can be trained to directly minimize the divergence between the target distribution and the refined distribution. To do we train BigGAN models with the same hyperparameters as in Appendix C.1.4 and Appendix C.1.5. The only difference is that we trained every model with three different losses: the Jensen-Shannon divergence, the hinge loss and the Jensen-Shannon divergence with the OBRS. We plot the FID during training to compare the different methods, but the FID is computed for the refined distribution only.

Our perspective on the experiments

In this appendix, we have presented the details of the experiments conducted in Chapter 5 and Chapter 6. We have shown that the Precision-Recall Divergence can be used to tune generative models between quality and diversity. We have also shown how Rejection can be used to improve the quality and the diversity of generative models. Our approaches aim to be as general as possible as long as the density ratio is tractable. For that reason we are using discriminator-based models, and thus we have applied our methods to GANs and especially BigGAN in higher dimension. We showed that Normalizing Flows could also be trained, and Diffusion models could also be improved. Our methods are not limited to a specific architecture or dataset, and we believe that they can be applied to tune any GANs and to any Normalizing Flows and to improve any generative model.

Résumé détaillé

D

D.1 Introduction

D.1.1 Contexte et Motivation

L'intelligence artificielle (IA) et l'apprentissage automatique (ML) se sont répandus dans divers secteurs, provoquant des changements révolutionnaires dans de nombreuses industries. Les systèmes d'IA ont la capacité d'apprendre des modèles à partir de données et de prendre des décisions intelligentes, ce qui a permis des avancées dans des domaines tels que l'analyse d'images, le traitement du langage naturel et la conduite autonome. Un aspect crucial du ML, la modélisation générative, est devenu un axe central, capable de créer de nouvelles instances de données ressemblant à des exemples du monde réel.

Les modèles génératifs cherchent à reproduire la distribution sous-jacente d'un ensemble de données pour générer de nouveaux échantillons cohérents. Cette tâche a suscité un intérêt croissant pour diverses applications créatives et pratiques, y compris la synthèse d'images pour les graphiques informatiques, le transfert de style en art, l'augmentation de données pour l'apprentissage automatique, la conception de molécules de médicaments en pharmacologie et la synthèse vocale dans le traitement du langage naturel. Dans le domaine du traitement d'images, des modèles tels que les Generative Adversarial Networks (GANs), les Variational Autoencoders (VAEs), les Normalizing Flows et les Diffusion models ont démontré leur efficacité à produire des données de haute qualité dans divers domaines.

Formellement, considérons une distribution cible inconnue P définie dans l'espace d'échantillons $\mathcal{X}$. Un modèle génératif est une distribution $\widehat{P}_G$ définie par la fonction de mapping G d'un espace latent $\mathcal{Z}$ vers $\mathcal{X}$ et une distribution Q définie sur $\mathcal{Z}$. La fonction de mapping G est construite, c'est-à-dire entraînée, de manière à ce que $\widehat{P}_G$ approche P. Cependant, en pratique, $\widehat{P}_G$ n'est jamais égal à P.

Comparant les résultats de deux modèles prometteurs, comme DALL-E 2 d'OpenAI et Midjourney, on observe des nuances. Les échantillons de Midjourney semblent plus convaincants pour les observateurs humains, tandis que ceux de DALL-E 2 peuvent parfois manquer de certains détails, influençant ainsi la perception des performances du modèle. Toutefois, DALL-E 2 parvient à capturer une plus grande variété de scénarios, de contextes, de sujets et d'ethnicités, encapsulant ainsi mieux la distribution sous-jacente.

D.1.2 Problématique

Pourquoi cette limitation survient-elle ? La première hypothèse est qu'elle reflète l'expressivité limitée des modèles génératifs existants. Idéalement, un modèle avec une expressivité illimitée correspondrait parfaitement à la distribution cible P, capable de générer des échantillons à la fois diversifiés et de haute qualité. Inversement, un modèle fortement restreint ne pourrait générer que des échantillons avec une haute fidélité mais une faible diversité ou une gamme plus large mais mal générée. Bien que les modèles modernes aient considérablement progressé, leur expressivité reste quelque peu limitée.

En parallèle avec les tâches de classification, les limitations de performance peuvent être en partie attribuées à la régularisation imposée sur la fonction de mapping G. À mesure que les modèles de deep learning ont grandi exponentiellement en taille et en profondeur, certaines régularisations sont devenues cruciales pour maintenir leur stabilité dans des scénarios génératifs. Certaines études suggèrent que, sous des hypothèses spécifiques concernant la déconnexion du support de P, les limitations de performance peuvent être attribuées à des contraintes imposées sur la fonction G et en particulier aux constantes de Lipschitz. Cependant, il est crucial de noter que ces publications se concentrent principalement sur des métriques très spécifiques sur P ou sur des métriques exclusivement liées à la qualité, sans considérer la diversité et la qualité séparément.

Observant ces limitations, la communauté a principalement dirigé ses efforts vers des modèles capables de générer des sorties de haute qualité. Cependant, en fonction des cas d'utilisation, les modèles génératifs peuvent nécessiter des échantillons de haute qualité pour la génération d'images et de vidéos haute résolution, la synthèse artistique ou la conception de modèles 3D. Alternativement, ils peuvent être requis pour générer des échantillons très diversifiés pour des applications comme l'augmentation de données, la découverte de médicaments ou la détection d'anomalies. Les exigences et limitations divergentes révèlent un compromis crucial entre la qualité des échantillons et leur diversité.

D.1.3 Problématique de la Thèse

Ces motivations soulignent les questions et les défis fondamentaux abordés dans cette thèse. À travers une investigation approfondie des modèles génératifs, nous visons à répondre à la question suivante :

Question: *Comment caractériser, ajuster et améliorer la précision et le rappel des modèles génératifs ?*

Pour aborder cette question, nous divisons le problème en deux composants : l'évaluation et l'amélioration du modèle. Initialement, notre attention est dirigée vers l'évaluation de la *Précision* (c'est-à-dire la qualité des échantillons) et du *Rappel* (c'est-à-dire la diversité des échantillons). Ensuite, nous explorons des stratégies pour améliorer la précision ou le rappel du modèle.

Pour répondre à la question de la caractérisation de la précision et du rappel pour les modèles génératifs, nous avons besoin d'une définition cohérente. La première question que nous aborderons est :

Question 1: *Comment unifier les définitions de la précision et du rappel pour les modèles génératifs ?*

Pour ce faire, nous regrouperons les différentes définitions dans le cadre des f-divergences. Nous montrerons que la définition des courbes PR peut être écrite comme une famille de f-divergences, et nous écrirons chaque autre définition dans notre cadre. Une fois un système d'évaluation unifié établi, nous pourrons analyser ces métriques et leur lien avec la régularisation.

Les modèles génératifs font face au défi d'améliorer à la fois la précision et le rappel. Cette tâche complexe peut être réalisée par plusieurs approches. Dans notre exploration, nous nous concentrons sur l'ajustement de deux aspects principaux : la fonction de perte et la méthode d'échantillonnage.

Ajustement de la Fonction de Perte La flexibilité réside uniquement dans la procédure d'entraînement et, surtout, dans le choix de la fonction de perte. Sous ces contraintes, sans ressources computationnelles supplémentaires, nous ne pouvons pas anticiper des améliorations simultanées de la précision et du rappel. Néanmoins, nous pouvons ajuster l'équilibre : permettre au modèle de prioriser la précision ou le rappel, et en particulier tout compromis explicite entre les deux. Cela nous mène à une question fondamentale :

Question 2: *Pouvons-nous entraîner un modèle génératif pour optimiser un compromis explicite spécifié par l'utilisateur entre la précision et le rappel ?*

Nous tirerons ainsi parti de l'analyse théorique menée pour répondre à la question 1, et développerons une méthode pour entraîner le modèle à minimiser une f-divergence représentant un compromis bien défini entre précision et rappel.

Modification de la Méthode d'Échantillonnage Après l'ajustement de la fonction de perte, nous explorons les possibilités au sein de la méthode d'échantillonnage, permettant une légère augmentation du coût computationnel de génération des échantillons. Ainsi, si nous considérons la distribution $\widehat{P}_G$ définie par un modèle fixe G, et en nous concentrant sur la méthode d'échantillonnage, l'échantillonnage par rejet, nous répondrons à la question suivante :

Question 3: *Avec un échantillonnage par rejet sous un budget limité, jusqu'à quel point pouvons-nous augmenter la précision et le rappel d'un modèle pré-entraîné ?*

Nous démontrerons qu'il existe un moyen d'optimiser le rejet des échantillons tirés afin de maximiser à la fois la précision et le rappel, tout en étant restreint par un budget limité.

D.1.4 Structure et Contribution

Dans cette thèse, nous visons à aborder les quatre problèmes énoncés, de manière linéaire en cinq chapitres :

- **Chapitre 2** et **Chapitre 3**

  Dans le Chapitre 2, nous introduisons divers modèles génératifs en apprentissage automatique, y compris les Generative Adversarial Networks, les Diffusion models, et les Normalizing Flows. Ce chapitre fournit aux lecteurs une compréhension complète des principes et des capacités de ces modèles. De plus, dans le Chapitre 3, nous présentons les différentes mesures de précision et de rappel définies dans la littérature. À la fin du Chapitre 2 et du Chapitre 3, les lecteurs auront acquis des informations précieuses sur le paysage des modèles génératifs et les outils essentiels pour évaluer leurs performances dans les chapitres suivants.

- **Chapitre 4**

  Dans le Chapitre 4, nous abordons les questions **??** et **??**, explorant une mesure particulière de "qualité des échantillons" et de "diversité des échantillons". Notre contribution clé est de montrer qu'une mesure proposée par Simon et al. [108] peut être élégamment exprimée comme une f-divergence, dénommée la divergence Précision-Rappel $\mathcal{D}_{\lambda\text{-PR}}$. Cette connexion nous permet de lier $\mathcal{D}_{\lambda\text{-PR}}$ à d'autres concepts de précision et de rappel et d'établir une relation claire entre $\mathcal{D}_{\lambda\text{-PR}}$ et toutes les autres f-divergences, répondant ainsi à la question **??**. De plus, nous tirons parti de la constante de Lipschitz des Generative Adversarial Networks et des Normalizing Flows. En analysant ces constantes, nous dérivons des bornes inférieures perspicaces sur la divergence PR, mettant en évidence les limites. Tout au long de ce chapitre, pour répondre à la question **??**, nous soulignons l'existence de certains cas pathologiques qui peuvent avoir un impact significatif sur la divergence PR.

- **Chapitre 5**

  Dans le Chapitre 5, nous abordons la question **??**, en nous basant sur les insights des Chapitres 3 et 4. Bien que la divergence PR montre une promesse pour l'évaluation des modèles génératifs, nous découvrons la limitation qu'elle ne peut pas être directement optimisée en utilisant les méthodes existantes. Pour surmonter ce défi, nous proposons et développons une nouvelle approche dans ce chapitre. Notre méthode permet aux modèles d'être entraînés à minimiser une divergence PR spécifique, permettant essentiellement l'optimisation d'un compromis particulier entre la précision et le rappel. Dans ce chapitre, nous offrons des preuves théoriques de la convergence de notre méthode proposée, fournissant des garanties de son efficacité. De plus, nous présentons des résultats expérimentaux obtenus en appliquant la méthode aux Generative Adversarial Networks et aux Normalizing Flows.

- **Chapitre 6**

  Dans le Chapitre 6, nous abordons la question **??**, en nous concentrant sur une méthode d'échantillonnage par rejet avec un budget restreint. Nous démontrons que cette approche est non seulement optimale mais aussi hautement efficace en pratique. En utilisant cette méthode avec un budget donné, nous atteignons une divergence minimale après rejet. De plus, nous montrons que notre approche proposée permet la minimisation directe de la divergence entre la distribution originale P et la distribution raffinée $\widetilde{P}$.

  À travers une analyse théorique rigoureuse et une expérimentation pratique, nous établissons l'efficacité et l'efficience de notre méthode proposée, offrant une solution robuste pour minimiser la divergence et affiner les modèles génératifs dans les contraintes de ressources.

D.2 Contexte des Modèles Génératifs en Apprentissage Profond

Dans ce chapitre, nous fournissons un contexte complet sur les modèles génératifs, une base cruciale pour notre thèse. Nous présentons des cadres et des algorithmes généraux et validons nos contributions en utilisant des ensembles de données d'images réelles et des modèles existants. Nous introduisons les modèles génératifs à la fois théoriquement et pratiquement, montrant leur implémentation avec des réseaux de neurones profonds.

Pour ce faire, nous commençons par l'énoncé du problème dans la Section 2.1.1, suivi d'une présentation des divergences f dans la Section 2.1.2. Ensuite, nous examinons comment le cadre général peut être étendu dans la Section 2.1.4. Pour la mise en œuvre pratique, une vue d'ensemble détaillée est disponible dans la Section 2.1.3, avec des sections dédiées aux Generative Adversarial Networks dans la Section 2.2.1, aux Normalizing Flows dans la Section 2.2.2 et aux Diffusion Models dans la Section 2.2.3.

Modèle génératif Pour définir un modèle génératif, nous avons besoin de quelques concepts de base :

- Considérons un espace d'entrée $\mathcal{X} \subset \mathbb{R}^d$. Nous définissons une distribution cible P dans cet espace, qui représente la distribution des données réelles.
- Nous introduisons un espace latent $\mathcal{Z} \subset \mathbb{R}^m$ avec une distribution latente Q, souvent une distribution simple comme une gaussienne multivariée.
- Une fonction de mapping G transforme les variables latentes en échantillons dans l'espace des données, définissant ainsi une distribution approximée $\widehat{P}$. L'objectif est de rendre $\widehat{P}$ aussi proche que possible de P.

Le but est de minimiser la différence entre P et $\widehat{P}$ en apprenant G. Cette différence est mesurée à l'aide de diverses métriques, notamment les divergences f, que nous détaillons dans la section suivante.

f-divergences pour mesurer la dissimilarité entre distributions Les f-divergences sont des mesures essentielles pour quantifier la dissimilarité entre deux distributions de probabilité. Elles permettent de comparer la distribution générée $\widehat{P}$ avec la distribution réelle P de manière cohérente et mathématiquement rigoureuse. Ces divergences ont été développées pour offrir une large gamme de comparaisons en fonction de la fonction génératrice f utilisée. Les f-divergences possèdent des propriétés importantes telles que la non-négativité et la convexité, ce qui les rend utiles pour l'optimisation des modèles génératifs. Pour une présentation détaillée et des exemples spécifiques de f-divergences, reportez-vous à la Section 2.1.2.

Generative Adversarial Networks Les Generative Adversarial Networks (GANs), introduits par Goodfellow et al. [44], utilisent un problème d'optimisation min-max impliquant deux réseaux : le générateur G et le discriminateur T. Le générateur crée des échantillons de données, tandis que le discriminateur tente de différencier les échantillons réels des échantillons générés. Cela forme un jeu à deux joueurs où le générateur essaie de tromper le discriminateur. Les GANs ont été étendus pour minimiser diverses divergences f avec le cadre f-GAN introduit par Nowozin et al. [90]. Cela permet aux GANs d'adopter une approche plus flexible et robuste pour la génération de données, en ajustant le modèle pour minimiser des divergences spécifiques.

Normalizing Flows Les Normalizing Flows (NFs) sont des modèles génératifs qui permettent de suivre la densité des données. Ils fonctionnent comme une bijection entre l'espace des données $\mathcal{X}$ et l'espace latent $\mathcal{Z}$. Cette transformation bidirectionnelle permet de calculer la densité des données générées et de maximiser directement la vraisemblance des données. Les NFs sont particulièrement utiles pour des applications nécessitant une estimation précise de la densité, comme la détection d'anomalies et les simulations physiques. Pour plus de détails sur les NFs, reportez-vous à la Section 2.2.2.

Diffusion Models Les Diffusion Models utilisent des processus de diffusion pour générer des échantillons. Ils partent d'une distribution simple et appliquent un processus de diffusion inversé pour générer des échantillons à partir de la distribution cible. Ces modèles ont récemment gagné en popularité en raison de leurs excellentes performances dans la génération d'images. Pour plus de détails sur les Diffusion Models et leurs applications, consultez la Section 2.2.3.

Portée et Approche de la Thèse

Cette thèse se concentre sur l'entraînement de divers modèles génératifs, en mettant l'accent sur les types de fonctions de perte utilisées. Nous nous intéressons principalement aux modèles pouvant minimiser facilement toute f-divergence, en particulier les GANs dans le cadre f-GAN et les Normalizing Flows dans le cadre Flow-GAN. Les Diffusion Models seront considérés comme une référence.

L'objectif est d'explorer les modèles génératifs pour comprendre les compromis et les défis liés à l'entraînement, notamment entre la qualité et la diversité des échantillons générés.

D.3 Un panorama des mesures de Précision-Rappel

Dans le chapitre précédent, nous avons discuté de l'entraînement des modèles génératifs pour minimiser une mesure de dissimilarité entre les distributions cibles et générées, généralement une divergence-f. La méthode pour approximer cette métrique varie selon le type de modèle, et la fonction objective est donc généralement spécifique au modèle. Pour une évaluation juste et cohérente des modèles génératifs, il est crucial que les métriques utilisées soient indépendantes du modèle. La méthode et l'algorithme pour calculer les métriques d'évaluation doivent être identiques pour tout modèle génératif. De plus, pour être facilement calculables, elles doivent dépendre uniquement d'un ensemble d'échantillons tirés à la fois de P et $\widehat{P}$, sans nécessiter d'entraînement supplémentaire.

D.3.1 Inception Score et Fréchet Inception Distance

L'Inception Score (IS) et le Fréchet Inception Distance (FID) sont des métriques populaires pour évaluer les modèles génératifs. L'IS, introduit par Salimans et al. (2016), utilise la capacité de classification du modèle Inception-v3. Il mesure la qualité et la diversité des images générées, mais a plusieurs limitations, telles que la non-sensibilité à la diversité intraclasse et le biais envers les classes d'ImageNet.

Le FID, quant à lui, mesure la distance entre les vecteurs de caractéristiques latentes calculés pour les images réelles et générées. Il utilise la distance de Fréchet et est plus corrélé avec la perception humaine de la qualité. Cependant, le FID ne distingue pas entre différents types d'erreurs de génération et peut être biaisé par les classes représentées dans ImageNet.

D.3.2 Précision et Rappel pour les Modèles Génératifs

Les mesures de précision et de rappel ont été adaptées des tâches de classification binaire pour évaluer indépendamment la qualité et la diversité des modèles génératifs.

Approche basée sur le support Kynkäänniemi et al. (2019) ont introduit une méthode pour évaluer la qualité et la diversité en se basant sur le support des distributions. Dans cette approche, la précision ($\bar{\alpha}$) mesure la proportion des échantillons générés se trouvant dans le support des données réelles, et le rappel ($\bar{\beta}$) mesure la proportion des données réelles couvertes par les échantillons générés. Cette méthode utilise un algorithme de k-plus proches voisins (k-NN) pour estimer le support des distributions dans l'espace latent des représentations d'un réseau de classification d'images (comme VGG).

Courbes de Précision-Rappel Sajjadi et al. (2018) ont proposé une approche plus raffinée, les courbes de précision-rappel (PR-Curves), qui intègrent la dissimilarité des densités pour une évaluation plus détaillée. Les courbes PR sont construites en variant un seuil λ sur le rapport de densité entre les distributions réelles et générées, permettant de capturer le compromis entre précision α_λ et rappel β_λ pour chaque valeur de seuil. En pratique, les PR-Curves sont calculées en utilisant des méthodes basées sur k-means ou des classificateurs pour estimer les densités dans l'espace latent des représentations d'Inception-v3. Bien que cette approche soit plus complexe, elle offre une évaluation plus nuancée de la correspondance entre les distributions générées et cibles.

Les relations entre les différentes métriques de précision-rappel montrent que les méthodes basées sur le support sont des cas particuliers des courbes PR pour des seuils spécifiques. Cette compréhension unifiée permet de mieux évaluer les modèles génératifs et de développer des métriques plus robustes et interprétables.

Notre point de vue sur ces métriques

Nous reconnaissons la valeur des différentes approches de mesure de précision et rappel pour évaluer les modèles génératifs. Les méthodes basées sur le support sont largement utilisées en raison de leur simplicité et de leur interprétabilité. Cependant, nous voyons un grand potentiel dans les courbes de précision-rappel (PR-Curves) et les approches plus récentes comme le Precision-Recall Cover pour offrir une évaluation plus fine et théoriquement fondée. Nous pensons que combiner ces approches peut fournir une évaluation plus complète et nuancée des modèles génératifs. Par conséquent, nous encourageons l'adoption de métriques intégrant à la fois les évaluations basées sur le support et les courbes de précision-rappel pour une meilleure compréhension des performances des modèles génératifs.

D.4 Précision et Rappel comme une f-divergence

Maintenant que nous avons introduit les f-divergences pour entraîner des modèles et les métriques de Précision-Rappel pour les évaluer, nous pouvons explorer comment ces concepts s'interrelient. Dans la Section 4.1, nous élaborerons sur les relations entre les f-divergences et les notions de qualité et de diversité. En conséquence, dans la Section 4.2, nous exprimerons les courbes PR comme une famille de f-divergences, appelée la divergence Précision-Rappel, dénotée PR-Divergence. Dans la Section 4.3, nous montrerons comment la PR-Divergence se connecte avec les métriques existantes comme les f-divergences et la Précision-Rappel Cover. Ce faisant, nous démontrerons comment la PR-Divergence est un outil central qui comble le fossé entre les métriques de Précision-Rappel et les f-divergences, répondant ainsi à la première question.

De plus, nous utiliserons la PR-Divergence pour quantifier les limites fondamentales des réseaux neuronaux en termes de qualité et de diversité. La Section 4.4 est dédiée à la deuxième question, explorant comment la PR-Divergence est influencée par les contraintes de Lipschitz du réseau neuronal.

Contributions : Plusieurs contributions sont présentées dans ce chapitre :

- Nous introduisons la divergence Précision-Rappel, une famille de f-divergences et montrons comment elle peut être utilisée pour comprendre pleinement la connexion entre les f-divergences et les métriques de Précision/Rappel. Ce résultat a été publié à la conférence : *Alexandre Verine et al. "Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows". en. In:* Advances in Neural Information Processing Systems *36 (Dec. 2023), pp. 32539–32573.*
- Nous montrons qu'il existe une relation entre les courbes PR et la PR-Cover. Ce travail est encore inédit.
- Nous montrons comment la contrainte de Lipschitz du générateur impacte la Précision et le Rappel. C'est la généralisation de résultats prouvés uniquement pour la Variation Totale et publiés à la conférence : *Alexandre Verine et al. "On the expressivity of bi-Lipschitz normalizing flows". en. In:* Proceedings of The 14th Asian Conference on Machine Learning. *ISSN: 2640-3498. PMLR, Apr. 2023, pp. 1054–1069.*

D.4.1 f-Divergences : insights sur la qualité et la diversité

Nous observons que la distribution $\widehat{P}$ obtenue à convergence est significativement influencée par le choix de la f-divergence. Optimiser la divergence de Kullback-Leibler tend à favoriser des modèles couvrant la masse, tandis qu'optimiser la KL inverse et la Jensen-Shannon tend à favoriser des comportements de recherche de mode.

Pour illustrer cela, nous présentons un exemple en ajustant une seule gaussienne à un mélange gaussien en utilisant différentes f-divergences. Les divergences résultantes montrent des comportements de recherche de mode ou de couverture de masse selon la f-divergence choisie. Ces observations motivent une exploration plus concrète de la connexion entre Précision et Rappel, en particulier les courbes PR, et les f-divergences.

D.4.2 La Divergence Précision-Rappel

Dans cette section, nous introduisons une nouvelle f-divergence, appelée la Divergence Précision-Rappel, notée $\mathcal{D}_{\lambda\text{-PR}}$. Nous définissons cette f-divergence et clarifions sa connexion avec les courbes PR. En utilisant la PR-Divergence, nous montrons le lien entre les courbes PR et les f-divergences traditionnelles ainsi que la PR-Cover.

Selon le Théorème 4.2.4, chaque point de la courbe PR correspond à une PR-Divergence spécifique, et donc minimiser $\mathcal{D}_{\lambda\text{-PR}}$ équivaut à maximiser α_λ. Cela fait de la $\mathcal{D}_{\lambda\text{-PR}}$ une candidate particulièrement appropriée pour l'entraînement d'un modèle génératif avec un compromis spécifique entre la Précision et le Rappel.

D.4.3 Relation avec d'autres métriques

Nous avons montré que nous pouvions écrire une courbe PR comme un ensemble de f-divergences, une première étape pour combler le fossé entre les courbes PR et les f-divergences. Dans cette section, nous montrons (1) que toute f-divergence peut être écrite comme une moyenne pondérée de la PR-Divergence et (2) la connexion entre les définitions des courbes PR et des métriques de Précision-Rappel.

Selon le Théorème 4.3.1, chaque f-divergence, sous des conditions légères, peut être écrite comme une moyenne pondérée de PR-Divergences. En particulier, les divergences de Kullback-Leibler, de Jensen-Shannon et de Kullback-Leibler inverse peuvent être écrites comme une moyenne pondérée de PR-Divergence, comme indiqué dans le Corollaire 4.3.2.

D.4.4 Bornes inférieures de la PR-Divergence dans les Réseaux Neuronaux

La distribution apprise est définie par $\widehat{P} = G\#Q$, donc l'ensemble des distributions possibles dépend fortement de l'ensemble des fonctions G représentées par les réseaux neuronaux. En particulier, les limites fondamentales des réseaux neuronaux devraient également se traduire par des limitations sur les distributions P.

Selon le Lemme 4.4.1, la PR-Divergence peut être exprimée en termes probabilistes, ce qui permet une manipulation plus versatile de la PR-Divergence. Nous montrons ainsi que la continuité de Lipschitz des réseaux neuronaux peut limiter l'expressivité des modèles. En particulier, le Théorème 4.4.3 indique que si le centre de la gaussienne latente est mappé à une région de faible densité, alors la PR-Divergence peut être bornée. Cette hypothèse est significative mais généralement plausible, en particulier pour des distributions de densité multimodales.

Remarques et Discussions

Dans ce chapitre, nous avons répondu à deux questions concernant la Précision/Rappel pour les modèles génératifs :

- **Comment unifier les définitions de la précision et du rappel pour les modèles génératifs ?**
  Nous avons introduit la divergence Précision-Rappel, un nouveau cadre qui encapsule à la fois la précision et le rappel en une métrique unifiée : la f-divergence, une classe de divergences largement utilisée dans la modélisation générative. Nous avons également montré comment toute f-divergence peut être écrite en termes de Précision et Rappel.
- **Quelle Précision et Rappel peuvent être atteints avec des réseaux neuronaux ayant des constantes de Lipschitz bornées ?**
  En nous appuyant sur la reformulation de la divergence Précision-Rappel, nous avons démontré que la propriété de Lipschitz (et bi-Lipschitz, lorsque applicable) de la fonction génératrice G peut être utilisée pour borner inférieurement la PR-Divergence. En d'autres termes, nous avons montré comment la contrainte de Lipschitz de la fonction génératrice G peut limiter l'expressivité des modèles. Nous avons montré que la PR-Divergence peut être strictement positive pour certaines distributions cibles P et certaines fonctions génératrices G.

Ce chapitre contribue à une compréhension plus raffinée des métriques de qualité-diversité pour les modèles génératifs à travers la PR-Divergence. De plus, il souligne le rôle critique de la contrainte de Lipschitz dans la limitation de l'expressivité globale des modèles génératifs. En d'autres termes, avec une expressivité limitée, un modèle ne peut pas atteindre à la fois une haute qualité et une grande diversité. Basé sur cette analyse, nous proposons dans la Section 5 une approche basée sur la PR-Divergence pour entraîner des modèles de manière optimale pour un compromis donné entre Précision et Rappel.

D.5 Ajuster les modèles selon un compromis défini par l'utilisateur

Dans le chapitre précédent, nous avons montré que maximiser un point sur la courbe Précision-Rappel correspond à minimiser une divergence PR spécifique. Dans ce chapitre, nous montrons comment entraîner un modèle génératif pour traiter tout compromis entre Précision et Rappel. Nous rappelons d'abord le cadre des f-GAN et expliquons pourquoi il ne répond pas à ce problème. Ensuite, nous proposons une méthode différente pour aborder ce problème et nous prouvons théoriquement que cette méthode converge. Enfin, nous démontrons l'efficacité de notre méthode sur des exemples jouets et des ensembles de données réels avec des modèles génératifs d'apprentissage profond, en les comparant avec les méthodes de pointe.

Contributions : La principale contribution de ce chapitre est la suivante :

- Nous proposons une méthode pour entraîner un modèle génératif à se concentrer sur un compromis spécifique entre qualité et diversité, et montrons que cette méthode modifie effectivement le compromis des modèles génératifs.

Ce résultat a été publié comme suit :

- Alexandre Verine et al. “Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows”. en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 32539–32573

D.5.1 Cadre des f-GAN

Le cadre des f-GAN généralise celui des GAN. L'objectif est de former une fonction réseau neuronal G pour minimiser toute divergence D_f entre la distribution des données P et la distribution générée $\widehat{P}_G$. Toutefois, la mise en œuvre pratique de cette formation est complexe et certaines divergences entraînent une instabilité ou une non-convergence de l'entraînement.

D.5.2 Minimisation de la divergence PR

Nous proposons une méthode pour entraîner un modèle génératif à se concentrer sur un compromis spécifique entre qualité et diversité en utilisant la forme primale de la divergence basée sur le ratio de densité. Nous démontrons théoriquement que cette méthode converge.

Expériences Nous utilisons notre approche pour entraîner divers modèles afin de minimiser la divergence PR sur des ensembles de données synthétiques et réels. Nous montrons que notre méthode permet de former des modèles pour des compromis spécifiques entre qualité et diversité.

Entraînement sur des données synthétiques 2D Nous avons entraîné un modèle sur un ensemble de données synthétiques pour visualiser les courbes PR pour différentes valeurs de λ. Les résultats montrent que le modèle formé avec $\lambda = 0.1$ couvre les 8 modes de la distribution des données, tandis que le modèle avec $\lambda = 10$ se concentre sur un seul mode. Les courbes PR sont différentes pour chaque valeur de λ, illustrant notre capacité à entraîner des modèles pour se concentrer sur un compromis spécifique entre qualité et diversité.

Entraînement de Normalizing Flows Nous avons entraîné un modèle GLOW sur les ensembles de données MNIST et Fashion-MNIST. Les résultats montrent que l'entraînement avec $\lambda = 0.1$ produit des échantillons variés mais de qualité inférieure, tandis que l'entraînement avec $\lambda = 10$ produit des échantillons de meilleure qualité mais moins diversifiés.

Entraînement et ajustement de GANs Nous avons entraîné un BigGAN sur CIFAR-10 et CelebA64, puis nous avons affiné des modèles pré-entraînés sur ImageNet et FFHQ. Les résultats montrent que notre approche permet d'ajuster la Précision et le Rappel de manière efficace, surpassant les méthodes de troncature traditionnelles.

Remarques et Discussions

Dans ce chapitre, nous avons répondu à une question concernant la Précision et le Rappel dans les modèles génératifs :

- **Pouvons-nous entraîner un modèle génératif pour se concentrer directement sur un compromis explicite entre Précision et Rappel défini par l'utilisateur ?**

Nous avons montré que la divergence PR peut être utilisée pour former des modèles génératifs à se concentrer sur un compromis spécifique entre qualité et diversité. Nous avons démontré que notre approche peut être utilisée pour entraîner des modèles sur des données synthétiques, des ensembles de données de faible complexité et de haute complexité, ainsi que pour affiner des modèles pré-entraînés sur de grands ensembles de données.

D.6 Échantillonnage de rejet budgétisé optimal pour améliorer la précision et le rappel

Dans les chapitres précédents, nous avons vu que nous pouvions ajuster la divergence minimisée par un modèle génératif pour optimiser la précision et le rappel. Cependant, nous n'avons considéré que la fonction de perte. L'échantillonnage de rejet utilise le rapport de densité entre la distribution des données et la distribution apprise pour améliorer la qualité des échantillons générés. Mais cette approche peut être coûteuse en calculs, notamment en haute dimension. Nous nous concentrerons sur la méthode d'échantillonnage pour générer de nouveaux échantillons et voir comment elle peut améliorer la précision et le rappel avec un budget de calcul limité.

Nous passerons en revue les méthodes d'échantillonnage existantes, en particulier l'échantillonnage de rejet, et introduirons une nouvelle méthode : l'échantillonnage de rejet budgétisé optimal (OBRS). Nous montrerons théoriquement et expérimentalement que cette méthode peut améliorer la précision et le rappel sous un budget limité.

Contributions : Les contributions de ce chapitre sont les suivantes :

- Nous proposons une nouvelle méthode d'échantillonnage, l'Optimal Budgeted Rejection Sampling (OBRS), qui est optimale pour minimiser toute f-divergence sous un budget fixe.
- Nous entraînons des modèles génératifs en tenant compte de l'échantillonnage de rejet et montrons que cela améliore (1) la convergence de l'entraînement et (2) la divergence entre la distribution des données et la distribution apprise après rejet.

Ces résultats ont été publiés comme suit :

- Alexandre Verine et al. “Optimal Budgeted Rejection Sampling for Generative Models”. In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics* (Mar. 2024). arXiv:2311.00460 [cs]

D.6.1 Échantillonnage de rejet pour les modèles génératifs

L’échantillonnage de rejet est une méthode classique pour échantillonner à partir d’une distribution cible en utilisant une distribution proposée et le rapport de densité entre les deux distributions. Cette méthode est utile pour améliorer les modèles génératifs mais peut être inefficace en haute dimension.

L’échantillonnage de rejet du discriminateur (DRS) Le DRS utilise le discriminateur d’un GAN pour affiner le processus de génération. En ajustant la fonction d’acceptation, nous pouvons améliorer la distribution apprise pour qu’elle corresponde mieux à la distribution cible. Cependant, cette méthode n’a pas de garanties théoriques sur le choix du paramètre de réglage et peut nécessiter des ajustements manuels.

Autres méthodes d’échantillonnage D’autres méthodes d’échantillonnage incluent le MH-GAN, le transport optimal par discriminateur (DOT), et le flux de gradient du discriminateur (DG$f$low). Ces méthodes peuvent améliorer la qualité des échantillons générés mais à un coût de calcul plus élevé.

D.7 L’échantillonnage de rejet budgétisé optimal (OBRS)

Nous introduisons l’OBRS, qui optimise la fonction d’acceptation pour minimiser toute f-divergence sous un budget fixe. Théoriquement, cette méthode est optimale pour améliorer la précision et le rappel tout en respectant le budget de calcul.

D.8 Amélioration de la précision et du rappel

L’OBRS améliore systématiquement la précision pour un rappel donné. La courbe PR de la distribution raffinée est une version verticalement étirée de la courbe PR initiale, ce qui montre une amélioration de la précision tout en maintenant le rappel.

D.9 Expériences

Nous montrons que l'OBRS surpasse d'autres méthodes d'échantillonnage en termes de précision et de rappel avec un coût de calcul similaire. Les expériences incluent des tests sur des ensembles de données synthétiques et réels, ainsi que sur des modèles GANs et des modèles de diffusion.

En intégrant l'OBRS dans le processus d'entraînement, nous pouvons éviter les minima locaux et accélérer la convergence. Nous proposons une méthode pour entraîner des modèles génératifs en utilisant l'OBRS, améliorant ainsi la qualité et la diversité des échantillons générés.

Remarques et discussions

Nous avons montré que l'OBRS peut améliorer la précision et le rappel des modèles génératifs sous un budget limité. Nous avons également proposé une méthode pour entraîner les modèles en tenant compte de l'échantillonnage de rejet. Cependant, des améliorations sont possibles, notamment en optimisant l'estimation du rapport de densité et en étendant l'OBRS aux modèles de diffusion.

Bibliography

[1] Abdelrahman Abdelhamed, Marcus Brubaker, and Michael Brown. "Noise Flow: Noise Modeling With Conditional Normalizing Flows". en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 3165–3173 (cit. on p. 22).

[2] Ahmed M. Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela van der Schaar. *How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models*. arXiv:2102.08921 [cs, stat]. July 2022 (cit. on p. 45).

[3] Alex Krizhevsky. "Learning multiple layers of features from tiny images". In: 2009 (cit. on p. 87).

[4] S. M. Ali and S. D. Silvey. "A General Class of Coefficients of Divergence of One Distribution from Another". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 28.1 (1966). Publisher: [Royal Statistical Society, Wiley], pp. 131–142 (cit. on p. 11).

[5] Shun-ichi Amari and Hiroshi Nagaoka. "Methods of Information Geometry". en. In: trans. by Daishi Harada. Vol. 191. Translations of Mathematical Monographs. Providence, Rhode Island: American Mathematical Society, Apr. 2007 (cit. on p. 15).

[6] Brian D. O. Anderson. "Reverse-time diffusion equation models". In: *Stochastic Processes and their Applications* 12.3 (May 1982), pp. 313–326 (cit. on p. 26).

[7] Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. *Refining Deep Generative Models via Discriminator Gradient Flow*. arXiv:2012.00780 [cs, stat]. June 2021 (cit. on pp. 5, 100, 109).

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". In: *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia,* arXiv: 1701.07875. Dec. 2017 (cit. on pp. 2, 21).

[9] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. *Discriminator Rejection Sampling*. arXiv:1810.06758 [cs, stat]. Feb. 2019 (cit. on pp. 5, 19, 97, 99, 100, 109, 111, 150).

[10] Shane Barratt and Rishi Sharma. *A Note on the Inception Score*. arXiv:1801.01973 [cs, stat]. June 2018 (cit. on p. 31).

[11] Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. "Invertible Residual Networks". In: *Proceedings of the 36 th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019*. arXiv: 1811.00995. May 2019 (cit. on pp. 24, 25, 69).

[12]Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger Grosse, and Joern-Henrik Jacobsen. “Understanding and Mitigating Exploding Inverses in Invertible Neural Networks”. en. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, Mar. 2021, pp. 1792–1800 (cit. on pp. 2, 25, 69, 128).

[13]Ali Borji. “Pros and cons of GAN evaluation measures: New developments”. en. In: *Computer Vision and Image Understanding* 215 (Jan. 2022), p. 103329 (cit. on pp. 31, 32).

[14]Léon Bottou, Frank E. Curtis, and Jorge Nocedal. *Optimization Methods for Large-Scale Machine Learning*. arXiv:1606.04838 [cs, math, stat]. Feb. 2018 (cit. on p. 17).

[15]L. M. Bregman. “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming”. en. In: *USSR Computational Mathematics and Mathematical Physics* 7.3 (Jan. 1967), pp. 200–217 (cit. on p. 83).

[16]Andrew Brock, Jeff Donahue, and Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. arXiv:1809.11096 [cs, stat]. Feb. 2019 (cit. on pp. 1, 2, 18, 21, 69, 80, 91, 93, 111, 148, 149).

[17]Florian Le Bronnec, Alexandre Verine, Benjamin Negrevergne, Yann Chevaleyre, and Alexandre Allauzen. “Exploring Precision and Recall to assess the quality and diversity of LLMs”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* (Feb. 2024). arXiv:2402.10693 [cs] (cit. on pp. xii, 125).

[18]Louis Béthune, Thibaut Boissin, Mathieu Serrurier, et al. *Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks*. arXiv:2104.05097 [cs, stat]. Oct. 2022 (cit. on p. 2).

[19]Louis Béthune, Alberto González-Sanz, Franck Mamalet, and Mathieu Serrurier. “The Many Faces of 1-Lipschitz Neural Networks”. In: *arXiv:2104.05097 [cs, stat]* (May 2021). arXiv: 2104.05097 (cit. on p. 2).

[20]Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, et al. “Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 12275–12287 (cit. on pp. 109, 111, 150).

[21]Fasil Cheema and Ruth Urner. “Precision Recall Cover: A Method For Assessing Generative Models”. en. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2023, pp. 6571–6594 (cit. on pp. 3, 4, 45, 49, 53, 63, 66, 67).

[22]Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. “Residual Flows for Invertible Generative Modeling”. In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.* arXiv: 1906.02735. July 2020 (cit. on pp. 24, 25, 71, 127, 128, 183).

[23]Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. *Neural Ordinary Differential Equations*. arXiv:1806.07366 [cs, stat]. Dec. 2019 (cit. on p. 24).

[24]Min Jin Chong and David Forsyth. *Effectively Unbiased FID and Inception Score and where to find them*. arXiv:1911.07023 [cs]. June 2020 (cit. on p. 32).

[25] Rob Cornish, Anthony L. Caterini, George Deligiannidis, and Arnaud Doucet. "Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows". In: *Proceedings of the 37th International Conference on Machine Learning*. arXiv: 1909.13833. Apr. 2021 (cit. on pp. 2, 72).

[26] Imre Csiszár. "Information-type measures of difference of probability distributions and indirect observation". en. In: *Studia Scientiarum Mathematicarum Hungarica* (Jan. 1967) (cit. on p. 11).

[27] Jia Deng, Wei Dong, Richard Socher, et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. June 2009, pp. 248–255 (cit. on p. 87).

[28] Terrance DeVries, Michal Drozdzal, and Graham W. Taylor. *Instance Selection for GANs*. arXiv:2007.15255 [cs, stat]. Oct. 2020 (cit. on p. 5).

[29] Madson L. D. Dias, César Lincoln C. Mattos, Ticiana L. C. da Silva, José Antônio F. de Macedo, and Wellington C. P. Silva. "Anomaly Detection in Trajectory Data with Normalizing Flows". In: *arXiv:2004.05958 [cs, stat]* (Apr. 2020). arXiv: 2004.05958 (cit. on p. 22).

[30] Laurent Dinh, David Krueger, and Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. arXiv:1410.8516 [cs]. Apr. 2015 (cit. on p. 24).

[31] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. arXiv:1605.08803 [cs, stat]. Feb. 2017 (cit. on pp. 18, 24, 87, 148).

[32] Josip Djolonga, Mario Lucic, Marco Cuturi, et al. *Precision-Recall Curves Using Information Divergence Frontiers*. arXiv:1905.10768 [cs, stat]. June 2020 (cit. on pp. 4, 45, 51).

[33] Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. *Model Collapse Demystified: The Case of Regression*. en. arXiv:2402.07712 [cs, stat]. Apr. 2024 (cit. on p. 125).

[34] Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. *A Tale of Tails: Model Collapse as a Change of Scaling Laws*. en. arXiv:2402.07043 [cs]. Feb. 2024 (cit. on p. 125).

[35] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159 (cit. on p. 17).

[36] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, et al. *Adversarially Learned Inference*. arXiv:1606.00704 [cs, stat]. Feb. 2017 (cit. on pp. 109, 110).

[37] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. *Neural Spline Flows*. arXiv:1906.04032 [cs, stat]. Dec. 2019 (cit. on p. 24).

[38] Tim van Erven and Peter Harremoës. "R\'enyi Divergence and Kullback-Leibler Divergence". In: *IEEE Transactions on Information Theory* 60.7 (July 2014). arXiv:1206.2459 [cs, math, stat], pp. 3797–3820 (cit. on p. 16).

[39] G. B. Folland. "A guide to advanced real analysis". en. In: *A guide to advanced real analysis*. The Dolciani mathematical expositions no. 37. OCLC: ocn428026440. Washington, D.C.: Mathematical Association of America, 2009, p. 95 (cit. on p. 67).

[40] Dan Friedman and Adji Bousso Dieng. *The Vendi Score: A Diversity Evaluation Metric for Machine Learning*. arXiv:2210.02410 [cond-mat, stat]. July 2023 (cit. on p. 45).

[41] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. “Adversarial text-to-image synthesis: A review”. In: *Neural Networks* 144 (Dec. 2021), pp. 187–209 (cit. on p. 31).

[42] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “Image Style Transfer Using Convolutional Neural Networks”. en. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 2414–2423 (cit. on p. 1).

[43] Dongyoung Go, Tomasz Korbak, Germán Kruszewski, et al. *Aligning Language Models with Preferences through f-divergence Minimization*. arXiv:2302.08215 [cs, stat]. June 2023 (cit. on p. 126).

[44] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. “Generative Adversarial Networks”. In: *27th Conference on Neural Information Processing Systems (NeurIPS 2014)*. arXiv: 1406.2661. June 2014 (cit. on pp. 1, 19, 20, 159, 181).

[45] Matej Grcić, Ivan Grubišić, and Siniša Šegvić. *Densely connected normalizing flows*. arXiv:2106.04627 [cs]. Nov. 2021 (cit. on p. 91).

[46] Aditya Grover, Manik Dhar, and Stefano Ermon. *Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models*. arXiv:1705.08868 [cs, stat]. Jan. 2018 (cit. on pp. 23, 80).

[47] Aditya Grover and Stefano Ermon. *Boosted Generative Models*. arXiv:1702.08484 [cs, stat]. Dec. 2017 (cit. on p. 5).

[48] Aditya Grover, Ramki Gummadi, Miguel Lazaro-Gredilla, Dale Schuurmans, and Stefano Ermon. *Variational Rejection Sampling*. arXiv:1804.01712 [cs, stat]. Apr. 2018 (cit. on p. 103).

[49] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, et al. “Automatic chemical design using a data-driven continuous representation of molecules”. In: *ACS Central Science* 4.2 (Feb. 2018). arXiv:1610.02415 [physics], pp. 268–276 (cit. on p. 1).

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015 (cit. on p. 25).

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017 (cit. on pp. 3, 32, 87).

[52] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. *Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design*. arXiv:1902.00275 [cs, stat]. May 2019 (cit. on p. 24).

[53] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. arXiv:2006.11239 [cs, stat]. Dec. 2020 (cit. on pp. 25, 93).

[54] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. *The Curious Case of Neural Text Degeneration*. en. arXiv:1904.09751 [cs]. Feb. 2020 (cit. on p. 125).

[55] Hisham Husain and Richard Nock. “Generalization for Discriminator-Guided Diffusion Models via Strong Duality”. en. In: (Oct. 2023) (cit. on p. 83).

[56]Thibaut Issenhuth, Ugo Tanielian, Jérémie Mary, and David Picard. *Unveiling the Latent Space Geometry of Push-Forward Generative Models*. arXiv:2207.10541 [cs, stat]. May 2023 (cit. on p. 2).

[57]Thibaut Issenhuth, Ugo Tanielian, David Picard, and Jeremie Mary. "Latent reweighting, an almost free improvement for GANs". en. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 3574–3583 (cit. on p. 5).

[58]Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. *Tailoring Language Generation Models under Total Variation Distance*. en. arXiv:2302.13344 [cs]. Feb. 2023 (cit. on p. 126).

[59]Gurtej Kanwar, Michael S. Albergo, Denis Boyda, et al. "Equivariant flow-based sampling for lattice gauge theory". In: *Physical Review Letters* 125.12 (Sept. 2020). arXiv:2003.06413 [cond-mat, physics:hep-lat], p. 121601 (cit. on p. 22).

[60]Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. arXiv:1710.10196 [cs, stat]. Feb. 2018 (cit. on p. 21).

[61]Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. *Elucidating the Design Space of Diffusion-Based Generative Models*. arXiv:2206.00364 [cs, stat]. Oct. 2022 (cit. on pp. 25, 112, 150, 188).

[62]Tero Karras, Miika Aittala, Samuli Laine, et al. "Alias-Free Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 852–863 (cit. on p. 22).

[63]Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. arXiv:1812.04948 [cs, stat]. Mar. 2019 (cit. on pp. 22, 87).

[64]Tero Karras, Samuli Laine, Miika Aittala, et al. *Analyzing and Improving the Image Quality of StyleGAN*. arXiv:1912.04958 [cs, eess, stat]. Mar. 2020 (cit. on p. 22).

[65]Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. *A Distributional Approach to Controlled Text Generation*. arXiv:2012.11635 [cs]. May 2021 (cit. on p. 126).

[66]Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. "Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models". In: *Proceedings of the 40 th International Conference on Machine Learning.* Vol. 202. arXiv:2211.17091 [cs] version: 3. Honolulu, Hawaii, USA: JMLR, Apr. 2023 (cit. on pp. 91, 112, 122, 126, 150, 188).

[67]Pum Jun Kim, Yoojin Jang, Jisu Kim, and Jaejun Yoo. *TopP&R: Robust Support Estimation Approach for Evaluating Fidelity and Diversity in Generative Models*. arXiv:2306.08013 [cs]. June 2023 (cit. on pp. 3, 4, 37).

[68]Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980 [cs]. Jan. 2017 (cit. on p. 17).

[69]Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. arXiv:1312.6114 [cs, stat]. Dec. 2022 (cit. on p. 1).

[70] Durk P. Kingma and Prafulla Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions”. en. In: *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.* Vol. 31. 2018 (cit. on pp. 18, 25, 88, 149).

[71] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, et al. “Improved Variational Inference with Inverse Autoregressive Flow”. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016 (cit. on p. 24).

[72] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). arXiv: 1908.09257, pp. 1–1 (cit. on p. 24).

[73] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. “Improved Precision and Recall Metric for Assessing Generative Models”. In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.* arXiv: 1904.06991. Oct. 2019 (cit. on pp. 3, 4, 33, 35, 36, 44, 45, 47, 52, 87, 182).

[74] Jonas Köhler, Leon Klein, and Frank Noé. *Equivariant Flows: sampling configurations for multi-body systems with symmetric energies*. arXiv:1910.00753 [physics, stat]. Oct. 2019 (cit. on p. 22).

[75] Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein. “GENERALIZING HAMILTONIAN MONTE CARLO WITH NEURAL NETWORKS”. en. In: (2018) (cit. on p. 22).

[76] Cheuk Ting Li and Farzan Farnia. “Mode-Seeking Divergences: Theory and Applications to GANs”. en. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2023, pp. 8321–8350 (cit. on p. 80).

[77] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. *Visualizing the Loss Landscape of Neural Nets*. arXiv:1712.09913 [cs, stat]. Nov. 2018 (cit. on p. 116).

[78] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. *Deep Learning Face Attributes in the Wild*. arXiv:1411.7766 [cs]. Sept. 2015 (cit. on p. 87).

[79] David Lopez-Paz and Maxime Oquab. *Revisiting Classifier Two-Sample Tests*. arXiv:1610.06545 [stat]. Mar. 2018 (cit. on p. 43).

[80] David J C MacKay. “Information Theory, Inference, and Learning Algorithms”. en. In: (May 2005) (cit. on p. 98).

[81] Alessio Miaschi, Dominique Brunato, Felice Dell’Orletta, and Giulia Venturi. “What Makes My Model Perplexed? A Linguistic Investigation on Neural Language Models Perplexity”. In: *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Online: Association for Computational Linguistics, June 2021, pp. 40–47 (cit. on p. 3).

[82] Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, and José Miguel Hernández-Lobato. *Bootstrap Your Flow*. arXiv:2111.11510 [cs, stat]. Mar. 2022 (cit. on p. 5).

[83] Thomas Minka. “Divergence measures and message passing”. en. In: (2005), p. 17 (cit. on pp. 5, 49, 56).

[84] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. *Spectral Normalization for Generative Adversarial Networks*. arXiv:1802.05957 [cs, stat]. Feb. 2018 (cit. on p. 2).

[85] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. *Reliable Fidelity and Diversity Metrics for Generative Models*. arXiv:2002.09797 [cs, stat]. June 2020 (cit. on pp. 3, 4, 37, 45, 47, 52, 87).

[86] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11 (Nov. 2010). arXiv:0809.0853 [cs, math, stat], pp. 5847–5861 (cit. on p. 84).

[87] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. "On surrogate loss functions and f-divergences". In: *The Annals of Statistics* 37.2 (Apr. 2009). arXiv:math/0510521 (cit. on pp. 14, 77).

[88] Frank Nielsen and Richard Nock. "The Dual Voronoi Diagrams with Respect to Representational Bregman Divergences". en. In: *2009 Sixth International Symposium on Voronoi Diagrams*. Copenhagen, Denmark: IEEE, June 2009, pp. 71–78 (cit. on p. 48).

[89] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. *Input Perturbation Reduces Exposure Bias in Diffusion Models*. arXiv:2301.11706 [cs]. Feb. 2023 (cit. on p. 91).

[90] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. *f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization*. arXiv:1606.00709 [cs, stat]. June 2016 (cit. on pp. 20, 76, 77, 80, 159, 181).

[91] Augustus Odena, Jacob Buckman, Catherine Olsson, et al. *Is Generator Conditioning Causally Related to GAN Performance?* arXiv:1802.08768 [cs, stat]. June 2018 (cit. on p. 2).

[92] Aaron van den Oord, Sander Dieleman, Heiga Zen, et al. *WaveNet: A Generative Model for Raw Audio*. arXiv:1609.03499 [cs]. Sept. 2016 (cit. on p. 1).

[93] George Papamakarios, Theo Pavlakou, and Iain Murray. *Masked Autoregressive Flow for Density Estimation*. arXiv:1705.07057 [cs, stat]. June 2018 (cit. on p. 24).

[94] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation". en. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 311 (cit. on p. 3).

[95] Adam Paszke, Sam Gross, Francisco Massa, et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. arXiv:1912.01703 [cs, stat]. Dec. 2019 (cit. on p. 148).

[96] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, et al. "MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 4816–4828 (cit. on pp. 3, 45, 125).

[97] Ben Poole, Alexander A. Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. *Improved generator objectives for GANs*. arXiv:1612.02780 [cs, stat]. Dec. 2016 (cit. on p. 80).

[98] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. *WaveGlow: A Flow-based Generative Network for Speech Synthesis*. arXiv:1811.00002 [cs, eess, stat]. Oct. 2018 (cit. on p. 22).

[99] Rafael Rafailov, Archit Sharma, Eric Mitchell, et al. *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. arXiv:2305.18290 [cs]. Dec. 2023 (cit. on p. 125).

[100] Danilo Jimenez Rezende and Shakir Mohamed. “Variational Inference with Normalizing Flows”. In: *arXiv:1505.05770 [cs, stat]*. June 2016 (cit. on pp. 1, 22).

[101] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv:1505.04597 [cs]. May 2015 (cit. on p. 26).

[102] Alfréd Rényi. “On Measures of Entropy and Information”. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Vol. 4.1. University of California Press, Jan. 1961, pp. 547–562 (cit. on p. 11).

[103] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. “Assessing Generative Models via Precision and Recall”. In: *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada*. arXiv: 1806.00035. Oct. 2018 (cit. on pp. 4, 31, 38, 39, 42–45, 51, 52, 87, 93, 182, 187).

[104] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, et al. *Improved Techniques for Training GANs*. arXiv:1606.03498 [cs]. June 2016 (cit. on pp. 3, 30, 87).

[105] Axel Sauer, Katja Schwarz, and Andreas Geiger. “StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets”. en. In: *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*. Vancouver BC Canada: ACM, Aug. 2022, pp. 1–10 (cit. on pp. 18, 22, 69, 91, 93).

[106] Maximilian Schmidt and Marko Simic. *Normalizing flows for novelty detection in industrial time series data*. arXiv:1906.06904 [cs, stat]. June 2019 (cit. on p. 22).

[107] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6.1 (July 2019), p. 60 (cit. on p. 1).

[108] Loic Simon, Ryan Webster, and Julien Rabin. “Revisiting precision recall definition for generative modeling”. en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, May 2019, pp. 5799–5808 (cit. on pp. 4, 7, 38, 39, 42–45, 47–49, 51, 52, 157, 182, 183, 187).

[109] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv:1409.1556 [cs] version: 6. Apr. 2015 (cit. on p. 36).

[110] Rodrigue Siry, Ryan Webster, Loic Simon, and Julien Rabin. “On the Theoretical Equivalence of Several Trade-Off Curves Assessing Statistical Proximity”. en. In: *Journal of Machine Learning Research* 24 (2023) (cit. on p. 63).

[111] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. arXiv:1503.03585 [cond-mat, q-bio, stat]. Nov. 2015 (cit. on p. 1).

[112] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. arXiv:2010.02502 [cs]. Oct. 2022 (cit. on p. 25).

[113] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. *Maximum Likelihood Training of Score-Based Diffusion Models*. arXiv:2101.09258 [cs, stat]. Oct. 2021 (cit. on p. 26).

[114] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, et al. "SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS". en. In: (2021) (cit. on pp. 25, 126).

[115] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. "Density Ratio Estimation: A Comprehensive Review". en. In: () (cit. on pp. 42, 83, 84).

[116] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*. en. arXiv:1512.00567 [cs]. Dec. 2015 (cit. on p. 30).

[117] Akinori Tanaka. "Discriminator optimal transport". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019 (cit. on pp. 5, 100, 109).

[118] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. "Learning disconnected manifolds: a no GANs land". In: *Proceedings of the 37 th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020*. arXiv: 2006.04596. Dec. 2020 (cit. on pp. 2, 73).

[119] Chenyang Tao, Liqun Chen, Ricardo Henao, Jianfeng Feng, and Lawrence Carin Duke. "Chi-square Generative Adversarial Network". en. In: *Proceedings of the 35th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2018, pp. 4887–4896 (cit. on p. 80).

[120] Christopher TH Teo, Milad Abdollahzadeh, and Ngai-Man Cheung. *Fair Generative Models via Transfer Learning*. arXiv:2212.00926 [cs]. Dec. 2022 (cit. on p. 124).

[121] Hoang Thanh-Tung and Truyen Tran. *On Catastrophic Forgetting and Mode Collapse in Generative Adversarial Networks*. arXiv:1807.04015 [cs, stat]. Mar. 2020 (cit. on p. 5).

[122] Constantino Tsallis. "Possible generalization of Boltzmann-Gibbs statistics". In: *Journal of Statistical Physics* 52 (July 1988), pp. 479–487 (cit. on p. 16).

[123] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. "Metropolis-Hastings Generative Adversarial Networks". en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, May 2019, pp. 6345–6353 (cit. on pp. 5, 100, 109, 111).

[124] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. *Generative Adversarial Nets from a Density Ratio Estimation Perspective*. arXiv:1610.02920 [stat]. Nov. 2016 (cit. on pp. 80, 83).

[125] Soobin Um and Changho Suh. "A Fair Generative Model Using Total Variation Distance". en. In: (Oct. 2021) (cit. on p. 80).

[126] Alexandre Verine, Benjamin Negrevergne, Yann Chevaleyre, and Fabrice Rossi. "On the expressivity of bi-Lipschitz normalizing flows". en. In: *Proceedings of The 14th Asian Conference on Machine Learning*. ISSN: 2640-3498. PMLR, Apr. 2023, pp. 1054–1069 (cit. on pp. xi, 56, 162).

[127] Alexandre Verine, Benjamin Negrevergne, Muni Sreenivas Pydi, and Yann Chevaleyre. "Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows". en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 32539–32573 (cit. on pp. xi, 56, 76, 162, 165).

[128]Alexandre Verine, Muni Sreenivas Pydi, Benjamin Negrevergne, and Yann Chevaleyre. "Optimal Budgeted Rejection Sampling for Generative Models". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics* (Mar. 2024). arXiv:2311.00460 [cs] (cit. on pp. xi, 96, 168).

[129]Jon Von Neuman. *Various techniques used in connection with random digits. Monte Carlo methods*. Nat. Bureau Standards. 1951 (cit. on pp. 19, 97).

[130]Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:1708.07747 [cs, stat]. Sept. 2017 (cit. on p. 87).

[131]Zhantao Yang, Ruili Feng, Han Zhang, et al. *Eliminating Lipschitz Singularities in Diffusion Models*. arXiv:2306.11251 [cs]. June 2023 (cit. on p. 69).

[132]Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs* 2 (2010) (cit. on p. 87).

[133]Mariia Zameshina, Olivier Teytaud, Fabien Teytaud, et al. "Fairness in generative modeling". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. arXiv:2210.03517 [cs]. July 2022, pp. 320–323 (cit. on p. 124).

[134]Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. *Self-Attention Generative Adversarial Networks*. arXiv:1805.08318 [cs, stat]. June 2019 (cit. on pp. 2, 21, 69).

List of Figures

List of Tables