

Imperial College London
Department of Electrical and Electronic Engineering

Universal Probabilistic Generative Models for Non-Gaussian Signals

Shengxi Li

Supervised by Professor Danilo P. Mandic

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy of Imperial College London

Communications & Signal Processing Group
Department of Electrical & Electronic Engineering
Imperial College London
2021

To my parents and girlfriend – My certainty in the uncertainty

Declaration of Originality

I herewith declare that the content of this thesis is the product of my own work under the guidance of my PhD supervisor Prof. Danilo P. Mandic. All material that is not my own work has been properly acknowledged with the standard referencing practices employed in this discipline.

Shengxi Li

Copyright Declaration

The copyright of this thesis resets with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

We live in a rapidly evolving era of Big-Data, with modern rich-content signals which exhibit large variance, complicated structures and large data volumes; this all calls for robust, explainable and expressive generative models. Probabilistic generative models are aimed at learning signal behaviours and providing a human-level understanding in an unsupervised way, pursuing the ultimate goal of artificial intelligence. Yet, the existing models are mostly based on the Gaussian distribution which, despite its mathematical tractability, is accompanied by limitations in terms of robustness and flexibility to accommodate the typically unbalanced and heavily tailed natures of modern signals. In this thesis, we address these issues through the class of elliptical distributions, and propose novel and efficient generative models with guaranteed robustness and enhanced expressibility. Deep neural networks are also addressed in this context to further improve the model expressive power, whereby we introduce a novel structure that seamlessly combines an auto-encoder and a generative adversarial net (GAN). This makes it possible to achieve semantically meaningful learning, thus ensuring model explainability.

To suit the needs of machine intelligence for real-world data, we first generalise the existing symmetric elliptical distribution to accommodate asymmetric signals, a commonplace in practice, by employing a directional von-Mises-Fisher (vMF) distribution and introducing a skewed elliptical distribution. In such a generative model, the generating process is clear, physically meaningful, and generic, allowing for explicit probabilistic density functions (pdfs) to be obtained. More importantly, the proposed generalisation is proved to share all desirable properties with the elliptical distribution, such as easy manipulation and practical implementations. We further provide a stable method to estimate the proposed model, and its superior performance is verified both theoretically and through experimental results.

Next, we address the paradigm of learning from multi-modal signals, by proposing the general elliptical mixture model (EMM), which is shown to be sufficiently expressive to fit arbitrary real-valued distributions. Another important property of the EMM is its robustness, which is quantified through the influence functions as theoretical bounds. However, the existing EMMs are limited to several specific types of elliptical distributions. We therefore propose a unifying framework for computable and identifiable EMMs. To

universally and efficiently solve EMMs, we establish a statistical manifold of elliptical distributions, and by finding the mismatch in their statistical manifolds, a reformulation trick and a redesigned cost are proposed, so as to allow for an equivalent statistical manifold and achieve extremely fast convergence speed.

To overcome the problem of spurious convergence to local optima when learning EMMs, we further propose a relaxed Wasserstein distance for significantly enhanced discrepancy measurement between probabilistic models; the so proposed distance is proved to be an upper bound of the Wasserstein distance, and more importantly, this allows for both explicit statistical manifold metrics and the corresponding operations. For the universal manifold solver, we propose an adaptive stochastic gradient descent method to accelerate the convergence. The flexible EMMs with so quantified robustness, estimated through efficient and universal solvers, are shown to almost approach the global optimum.

The focus next shifts to using deep neural nets in generative models, to further enhance model expressibility for complicated signals. In particular, we propose to use the characteristic functions (CFs), a powerful tool that contains all the information about any general distribution, to evaluate distributions for signals with high dimensions and complicated structures. In the analysis, we first prove the theoretical completeness of the evaluation, and that benefiting for the inherent complex-valued CF, its phase and amplitude parts possess physical meanings. More importantly, we introduce a reciprocal theory to make the CFs complete when comparing two transformed signals through deep neural networks. The proposed reciprocal CF generative adversarial net (RCF-GAN) is therefore seamlessly combines the auto-encoder and GAN, and learns a semantic low-dimensional manifold for both the generation and reconstruction with enhanced model explainability. We further propose an optimal sampling strategy under the umbrella of flexible elliptical distributions, and an anchor design to improve the convergence speed. Experimental results show that the proposed RCF-GAN achieves the state-of-the-art generation and reconstruction on images, whilst enjoying simple yet stable training procedure.

Finally, for data observed on irregular domains, the graph link prediction problem is considered as a probabilistic generative model, making it possible to learn graphs via statistics from graph signals. We show that most of the generative models can be applied in this task, and further propose a graph RCF-GAN (GRCF-GAN), which learns meaningful representations of graph nodes and achieves the state-of-the-art performances.

Acknowledgements

I would first give my greatest thanks to my supervisor, Prof. Danilo Mandic, to provide the opportunity of pursuing my PhD throughout 3.5 years with an unforgettable collaboration. He always provides helpful and valuable comments for my research, and is open-minded to give various interesting topics for me to investigate and play with. Pursuing the PhD by stepping into a new research field is filled with difficulties and uncertainties; he encourages me a lot, dragging me from the research void, and fully supports every decision I made, as well as shares the happy beers when we make through. He always unreservedly chooses to support me whenever I am in trouble. He also interprets my name SHengxi as “SuperHero”; of course I am not, this however comforts me a lot when facing difficult moments and keeps me to be self-motivated in my research. I would also wish to share my sincere thanks to my examiners, Prof. Imad M. Jaimoukha and Prof. Vasile Palade, who have provided constructive and valuable comments in improving this thesis by their thoughtful and profound knowledge in relevant fields, simultaneously resulting into many new and existing viewpoints, inspirations and motivation. All of those, shape me to be an independent researcher and come with the work presented in this thesis.

I would also like to thank my master supervisor, Prof. Mai Xu, who guided me to the amazing field of signal processing that now becomes my life-long enthusiasm. He showed his uncommon patience to provide valuable suggestions towards a freshman researcher, which I am truly grateful and benefit from.

My group members and collaborators also provided many exceptional collaborations, discussions and valuable comments to this thesis, including Prof. Ljubisa Stankovic, Zeyang Yu, Bruno Scalzo-Dees, Giuseppe Calvi, Ilia Kisil, Metin Yarici, Kriton Konstantinidis, Yao Xu, Min Xiang, Yili Xia, Zehua Chen, Ahmad Moniri, Thayne Thanthawaritthisai, and Alex Stott. Special thanks to my closest collaborator, Zeyang Yu, who helped me a lot both academically and personally.

It is my fortune as well to have so many friends in Imperial College London, where I gain creative ideas and insights. So I would like to extend my sincere thanks to Xin Deng, Jinyuan Xia, Shuaifeng Zhi, Junjie Huang, Tianrui Liu, Roxana Alexandru, Nitish Mital, Hamdi Joudeh, Conghui Li, Zhuqing Tang, to name but a few. It is also my great pleasure to have many interesting chats and experiences with my office close neighbours, Roxana

Alexandru, Jinyuan Xia and Metin Yarici. I shall also thank the Imperial Lee Family scholarship and funding, which support my overall PhD research, letting me to be able to focus on the challenging topics without worrying about other irrelevant affairs. Thousands of words cannot express my thanks to all the support from my friends, especially in the special period of Coronavirus-19 when I stayed at home a whole year in London.

Last but not least, I wish to thank my girlfriend, Yue Zhang, and my parents; they all firmly support my decisions, just like the always descending gradient when I am on the road of pursuing the global optima, that pushes me a bit forward to skip from saddle points and helps jump out from local optimum.

Abbreviations

Generative models:

GMM	Gaussian mixture model
EMM	Elliptical mixture model
vMF distribution	von-Mises-Fisher distribution
GMRF	Gaussian Markov random field
VAE	Variational auto-encoder
GAN	Generative adversarial net
VGAE	Variational graph auto-encoder
DCGAN	Deep convolutional GAN
W-GAN	Wasserstein GAN
W-GAN-GP	Wasserstein GAN with gradient penalty
MMD-GAN	Maximum mean discrepancy GAN
AGE	Adversarial generator encoder
RCF-GAN	Reciprocal characteristic function GAN
GRCF-GAN	Graph reciprocal characteristic function GAN
GCN	Graph convolutional net

Solvers:

MLL	Maximum log-likelihood
EM	Expectation maximisation
IRA	Iteratively reweighing algorithm

Metrics:

MSE	Mean squared error
KL divergence	Kullback–Leibler divergence
NLL	Negative log-likelihood
ELBO	Evidence lower bound
JS divergence	Jensen-Shannon divergence
EMD	Earth mover’s distance

IPM	Integral probability metric
MMD	Maximum mean discrepancy
CF	Characteristic function
ECF	Empirical characteristic function
FID	Fréchet inception distance
KID	Kernel inception distance
AUC	Average area under the ROC curve
AP	Average precision
RKHS	Reproducing kernel Hilbert space

Other abbreviations:

pdf	Probabilistic density function
cdf	Cumulative density function
mgf	Moment generating function
psd	Positive semi-definite
iid	Independent and identically distributed
IF	Influence function

Notations

Nomenclature

x, y, z	Light lowercase	Scalar parameters
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Bold lowercase	Vector parameters
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	Bold uppercase	Matrix parameters
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	Light calligraphic	Random scalar variables
$\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Z}}$	Bold calligraphic	Random vector variables
$\mathcal{X}_\theta$	Subscript θ	Random variable parametrised by a parameter θ

Domains and spaces

m	Dimension
n	Number of samples
k	Number of clusters
$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex numbers
$\mathbb{R}^m$	Real numbers of dimension m
$\mathcal{H}$	Reproducing kernel Hilbert space
$\mathcal{M}$	Riemannian manifold
ds^2	Riemannian metric
$\mathbb{P}$	Manifold of positive semi-definite matrices
$T_{\Sigma}\mathcal{M}$	Tangent space of Riemannian manifold $\mathcal{M}$ at point Σ
∇_E	Euclidean gradient
∇_R	Riemannian gradient

Special notations

L	Loss term (function)
$\mathbf{I}$	Identity matrix
$\boldsymbol{\mu}$	Mean (location) parameter
$\boldsymbol{\Sigma}$	Covariance matrix

Q	Precision matrix ($\mathbf{Q} = \mathbf{\Sigma}^{-1}$)
A	Adjacency matrix
L	Laplacian matrix
D	Degree matrix
Z	Latent random variables
S	Random variable uniformly distributed on a unit spherical surface
V	Random variable satisfying a von-Mises-Fisher distribution

Statistics

$p_{\mathcal{X}}(\mathbf{x})$	Probabilistic density function of random variable $\mathcal{X}$, indexed by $\mathbf{x}$ in the function; Equal to $p(\mathbf{x})$ if there is not ambiguity in context
$\Phi_{\mathcal{X}}(\mathbf{t})$	Characteristic function of random variable $\mathcal{X}$
$\mathbb{E}_{\mathcal{X}}[\cdot]$	Expectation operation over random variable $\mathcal{X}$ Equal to $\mathbb{E}_{p_{\mathcal{X}}(\mathbf{x})}[\cdot]$, $\mathbb{E}_{p(\mathbf{x})}[\cdot]$, or $\mathbb{E}[\cdot]$ if there is not ambiguity in context
$\text{Var}_{\mathcal{X}}[\cdot]$	Variance operation over random variable $\mathcal{X}$ Equal to $\text{Var}_{p_{\mathcal{X}}(\mathbf{x})}[\cdot]$, $\text{Var}_{p(\mathbf{x})}[\cdot]$, or $\text{Var}[\cdot]$ if there is not ambiguity in context
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$
$\mathcal{E}(\boldsymbol{\mu}, \mathbf{\Sigma}, g)$	Multivariate elliptical distribution with mean $\boldsymbol{\mu}$, covariance matrix $\mathbf{\Sigma}$ and function type g
$=^d$	Equivalence in probabilistic distributions, e.g., $\mathcal{X} =^d \mathcal{Y} + \mathcal{Z}$
$\sim$	Satisfying probabilistic density function, e.g., $\mathcal{X} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$

Operations

$\det(\mathbf{A})$	Determinant operation of a matrix $\mathbf{A}$
$\text{tr}(\mathbf{A})$	Trace operation of a matrix $\mathbf{A}$
$\mathbf{A}^T$	Transpose of a matrix $\mathbf{A}$
$\mathbf{A}^H$	Hermitian transpose of a matrix $\mathbf{A}$
$\mathbf{A}^{1/2}$	Square root of a positive semi-definite matrix $\mathbf{A}$
$\cdot$	Inner product

Contents

Declaration of Originality	i
Abstract	vii
Acknowledgements	ix
Abbreviations	x
Notations	xiii
1 Introduction	1
1.1 Motivation and objectives	1
1.2 Thesis outline and original contributions	6
1.3 Publication list	9
2 Background	13
2.1 Explicit Generative Models	13
2.1.1 Gaussian Models	16
2.1.2 Elliptical Distributions	21
2.1.3 Existing Generalised Elliptical Distributions	27
2.2 Implicit Generative Models	29

2.2.1	Probabilistic Flows	30
2.2.2	Variational Auto-Encoder (VAE)	31
2.2.3	Generative Adversarial Net (GAN)	35
2.3	Preliminaries on Statistical Metrics	38
2.3.1	Kullback–Leibler (KL) Divergence and Extensions	38
2.3.2	Wasserstein Distance	40
2.3.3	Integral Probability Metrics (IPMs)	41
2.4	Preliminaries on Statistical Manifolds	42
3	Generalised Elliptical Distributions	45
3.1	Introduction	46
3.2	Generalisation via Von Mises-Fisher Distribution	47
3.2.1	The vMF Elliptical Distribution	47
3.2.2	Properties	49
3.2.3	Optimisation	51
3.3	Numerical Results	52
3.4	Appendix	54
3.4.1	Proof of the Closed-Form pdf in (3.2)	54
3.4.2	Proof of the Optimum in (3.14)	55
3.5	Summary	56
4	General Framework for Learning the Elliptical Mixture Model	57
4.1	Introduction	58
4.2	Manifold Optimisation for the EMM	61

4.2.1	Statistical Metrics for Elliptical Distributions	61
4.2.2	Manifold Optimisation on the EMM	63
4.2.3	Regularisation	67
4.3	Influence Functions of the EMM	68
4.4	Experimental Results	71
4.4.1	Parameter Settings and Environments	71
4.4.2	Toy Examples	74
4.4.3	Evaluations over the Synthetic Datasets	78
4.4.4	BSDS500 Dataset	79
4.5	Appendix	82
4.5.1	Proof of Lemma 4.1	82
4.5.2	Proof of Theorem 4.1	82
4.5.3	Proof of Lemma 4.3	83
4.5.4	Proof of Lemma 4.4	86
4.6	Summary	87
5	Solving General EMMs through a Relaxed Wasserstein Manifold	88
5.1	Introduction	89
5.2	Computable and Identifiable EMMs	92
5.3	Statistical Manifold towards EMMs	93
5.3.1	Relaxed Wasserstein Distance between EMMs	95
5.3.2	Statistical Manifold for EMM Problems	96
5.4	Adaptively Accelerated Optimisation	98
5.4.1	Vanilla Gradient Descent on Statistical Manifold	98

5.4.2	Adaptively Accelerated Algorithm	100
5.5	Experimental Results	101
5.5.1	Assessment over the <i>Dadam</i> on Synthetic Data	104
5.5.2	Overall Comparisons on Synthetic Data	104
5.5.3	Overall Comparisons on the MNIST	105
5.5.4	Overall Comparisons on the BSDS500	106
5.6	Appendix	106
5.6.1	Proof for expressions in Table 5.1	106
5.6.2	Proof of Theorem 5.1	109
5.6.3	Proof of Lemma 5.1	110
5.6.4	Proof of Lemma 5.2	111
5.6.5	Calculations within Table 5.2	112
5.7	Summary	114
6	Reciprocal GAN through Characteristic Functions (RCF-GAN)	115
6.1	Introduction	116
6.2	Characteristic Function Loss and Efficient Sampling Strategy	120
6.2.1	Characteristic Function and Elliptical Distribution	120
6.2.2	Distance Measure via Characteristic Functions	121
6.2.3	Sampling the Characteristic Function Loss	123
6.3	Reciprocal Adversarial Learning	126
6.3.1	Characteristic Function Loss in RCF-GAN	126
6.3.2	Matching in the Embedded Space	127
6.3.3	Putting Everything Together	129

6.4	Experimental Results	130
6.4.1	Settings for Image Generation	130
6.4.2	Random Image Generation under DCGAN Structure	132
6.4.3	Phase and Amplitude in the CF Loss	133
6.4.4	Image Reconstruction and Interpolation under DCGAN Structure	135
6.4.5	Ablation Study	136
6.4.6	Advancements under ResNet Structure	137
6.5	Appendix	139
6.5.1	Proof of Lemma 6.1	139
6.5.2	Proof of Lemma 6.2	141
6.5.3	Proof of Lemma 6.3	142
6.5.4	Proof of Lemma 6.4	142
6.6	Summary	143
7	Application in Graph Link Prediction	145
7.1	Introduction	145
7.2	Probabilistic Generative Models in Graph Link Prediction	148
7.2.1	Basic Gaussian Model	150
7.2.2	Factor Model and Polynomial Models	151
7.2.3	Deep Graph Generative Models	153
7.3	Proposed Graph Link Prediction Method	154
7.4	Experimental Results	155
7.5	Summary	157

8 Conclusion	158
8.1 Summary of Thesis Achievements	158
8.2 Future Work	160
8.2.1 Generating Process: Advanced Elliptical Models	160
8.2.2 Metric: Statistical Manifold and Graphs	161
8.2.3 Metric: Characteristic Functions in Distribution Alignment	161
8.2.4 Solver: Auto-Encoding in an Adversarial Way	162
8.2.5 Application: Image Semantic Compression and Representation	162
Bibliography	162

List of Tables

4.1	Details of the 9 elliptical distributions used for assessments	72
4.2	Properties of the elliptical distributions used for evaluations	73
4.3	Overall results averaged across m and k for 9 types of EMMs optimised by Our, IRA and RMO methods.	77
4.4	Detailed comparisons among Our, IRA and RMO on optimising 3 EMMs in the case of $c = 10$ and $e = 10$	77
4.5	Comparisons averaged over 500 pictures among Our, the IRA and the RMO for estimating the 5 EMMs	80
4.6	Comparisons among Our (with different manifold solvers), IRA and RMO methods on the BSDS500 image patches	80
5.1	$\mathcal{R}^2 \leftrightarrow$ Computable elliptical distributions	94
5.2	Basic operations of the manifold in Lemma 5.2	98
5.3	Comparisons among our, WO/M and EM methods for GMMs by varying m and k	102
5.4	Comparison performance of our, WO/M and IRA algorithms over the BSDS500 dataset via five metrics	103

6.1	The FID and KID scores obtained from the DCGAN [1] structure. The results of the DCGAN and W-GAN-GP are from [2] and [3]. The corresponding publicly available codes were run to obtain the results of the W-GAN [4], MMD-GAN [5], OCF-GAN and OCF-GAN-GP [6]. The results of the AGE were tested from its pre-trained models [7].	130
6.2	The FID and KID scores of reconstructed images under the DCGAN structure. The AGE results were tested from its pre-trained models [7].	136
6.3	Ablation study on the CelebA dataset. The proposed RCF-GAN was evaluated and compared to the one without (w/o) reciprocal requirement ($\lambda = 0$) and without anchor design.	137
6.4	The FID and KID scores under the ResNet structure, for 64×64 and 128×128 image sizes. The result of Sphere GAN was obtained from the original article [8]. We ran the available code of the OCF-GAN-GP [6] with its implemented ResNet structure because it failed to converge in our structure given in Fig. 6.6.	138

List of Figures

1.1	Thesis structure, with the flow aiming to demystify generative models by tractable generating process, statistical metrics, and efficient solvers.	12
2.1	The relationships among the generative models reviewed in this chapter. . .	14
2.2	Illustration of the generating process of Gaussian and elliptical distributions.	25
2.3	Illustration of the comparison principles of the KL divergence and Wasserstein distance.	40
2.4	Illustrations of basic operations on a Riemannian manifold.	43
3.1	Illustration of generating the proposed vMF elliptical distribution via its stochastic representation of (3.1). The red dot denotes the point $(0, 0)$	48
3.2	Log-likelihood error ratios with respect to different dimensions, m , and degrees of concentration, τ , in estimating the proposed: (a) vMF Gaussian and (b) vMF Cauchy distribution. One example per distribution of randomly generated samples from the stochastic representations of (3.1) is also provided in the upper right panel. For the convenience of illustration, we show the main part of vMF Cauchy samples whilst ignoring some highly scattered samples.	54

- 4.1 For reproducibility, we follow the work of [9] to generate three one-dimensional clusters using the inverse of the Gaussian cumulative distribution function. These three clusters are centred respectively at $\mu_1 = 0$, $\mu_2 = -5$ and $\mu_3 = -10$, and IF curves for μ_1 are illustrated. The mixture distributions are the Gaussian, Cauchy, Laplace and GG1.5, as shown in Table 4.1. The theoretical bounds are plotted in solid lines and the actual IFs are plotted in dotted lines. The horizontal dotted lines represent the boundedness (upper bounds) where the mixtures exhibit robustness. The zoomed versions of the IFs of the mean are given in the black box of each figure. 69
- 4.2 Toy examples consisting of four Gaussian sets. (a) Data structure: The red circles represent the mean values at $(5, 5)$, $(5, -5)$, $(-5, 5)$ and $(-5, -5)$, and the black circles denote covariance ellipses including 95% data samples of each Gaussian distribution. (b) Test case adding 100% uniform noisy samples to the data. (c) Test case of data that consist of two Gaussian and two Cauchy sets. (d) Tails of the distributions which were utilised in this test. It needs to be pointed out that the Cauchy samples in (c) are spread over a wide range, so that we show (c) within $(\pm 15, \pm 15)$ for illustration convenience. 75
- 4.3 Optimisation results of the proposed method across the 5 EMMs. The top row shows the results for Case 1 and the bottom row shows the results for Case 2. The red circles denote the ground-truth as shown in Fig. 4.2-(a), whilst the black crosses and ellipses represent the estimated mean and covariance matrices. The colour of each sample is corresponding to that of Fig. 4.2-(a), and is classified by selecting the maximum posterior among the clusters. 75
- 4.4 Cost difference against the number of iterations of the 5 EMMs for Case 1. The top figures show the cost difference against the iterations optimised via Our, IRA and RMO methods. The bottom quantities are the final convergence speed in terms of the number of iterations and execution time (ites/sec). 76

4.5	Cost difference against the number of iterations of the 5 EMMs for Case 2. The top figures show the cost difference against the iterations optimised via Our, IRA and RMO methods. The bottom quantities are the final convergence speed in terms of the number of iterations and execution time (ites/sec).	76
4.6	Reconstructed images by Our method when optimising the 5 EMMs.	79
5.1	Illustration of a simple one-dimensional GMM with 3 clusters. (a): The ground-truth of clusters is plotted via the black dotted lines; Parametric models $\rho(\boldsymbol{\theta})$ are denoted as colourful lines; We set ALL the parameters $\boldsymbol{\theta}$ of the GMM to the ground-truth EXCEPT the mean values of only two clusters, i.e., μ_1 and μ_2 . (b): The KL divergence and the normalised gradient versus the variance of μ_1 and μ_2 . (c): The Wasserstein distance and the normalised gradient versus the variance of μ_1 and μ_2	89
5.2	The Wasserstein distance against the number of iterations for the four considered algorithms, averaged over the three types of datasets ($\{m, k\} = \{2, 3\}, \{8, 9\}, \{16, 27\}$). The best learning rate is shown in the legend. 1000 iterations are for illustration purpose. The computational time per iteration is discussed in Table 5.3.	102
5.3	The optimised GMMs via the EM and our ($W/M+Dadam$) methods on a flower-shaped synthetic data ($n = 10,000$) with random initialisations. Each cluster in the flower-shaped data is composed by Gaussian-distributed samples.	102
5.4	Wasserstein distance against the computational time for the <i>EM</i> and <i>Our</i> , on the downsampled 3×3 and 5×5 MNIST test data. The best learning rate is chosen as $\alpha = 0.1$. The cluster number is chosen as $k = 3$. The maximal iterations are set to 10,000 for both methods.	105

- 6.1 The overall structure of the proposed RCF-GAN. The generator serves to minimise the CF loss between the embedded real and fake distributions. The *critic* serves to minimise the CF loss between the embedded real and the input noise distributions, whilst maximising the CF loss between the embedded fake and the input noise distributions. Moreover, an MSE loss between the embedded fake and the input noise distributions is regularised as the auto-encoder loss, which has not been shown in the figure. An optional *t*-net can be employed to optimally sample the CF loss. 118
- 6.2 Two experiments on the MNIST dataset which illustrate the physical meaning of the phase and amplitude of the CF. (a) A multivariate Gaussian fit to the images of digits 1 and 2, by naively assuming that each pixel is independent from other pixels. The phase and amplitude information of the CFs between the two multivariate distributions were then swapped, followed by random sampling from the swapped distributions. (b)-(d) A generator was directly trained on the given images of each digit. To avoid the impact from the *critic*, we *DID NOT* employ the *critic* in this experiment but directly calculated the loss between images after the generator with different α . We performed training for amplitude for $\alpha = 0.999$ in (b), phase only for $\alpha = 0.001$ in (c) and equally training the amplitude and phase information for $\alpha = 0.5$ in (d). 124
- 6.3 Convergence curves and the images generated by the proposed RCF-GAN from Gaussian noise, under the DCGAN [1] structure. The curves represent an average over a moving window, with 500 iterations. 134
- 6.4 The FID and KID scores for different α , under the DCGAN [1] structure. Observe the embedded space (by interpolated images) of the proposed RCF-GAN, which was learnt with different α on CelebA dataset. The details of image interpolation are given in Section 6.4.4. 135

6.5	Image reconstruction (upper panel) and interpolation (lower panel) by the proposed RCF-GAN, AGE [7] and MMD-GAN [5] for the CelebA dataset, under the DCGAN [1] structure. The upper panel shows the reconstructed images (in even columns) corresponding to the original images (in odd columns). The lower panel displays the linear interpolation in the embedded domain.	136
6.6	The ResNet 128×128 structure adopted in this work. The numbers in red colour represent the channel setting of ResNet 64×64	137
6.7	Random generation, reconstruction and interpolation by the proposed RCF-GAN, using ResNet and images of size 128×128 pixels. The upper panel shows images of the CelebA dataset and the lower panel images from the LSUN_B dataset.	138
7.1	Three types of graph connections given the observed nodal sample $\mathbf{v} = [3, 2, 1, 0, -1, -2, -3]^T$	150
7.2	The spectrum values corresponding to the three types of graph connections in Figure 7.1. The energy is calculated via $\mathbf{v}^T \mathbf{L} \mathbf{v}$, where small values indicate a smooth graph.	150
7.3	Principle of the use of VGAE in link prediction. The training process of the VGAE is based on the provided training edges and node features, whilst the test edges are not connected. Different from the standard auto-encoder, the encoder of VGAE outputs the estimated mean and covariance values of Gaussian distributions ($\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_6$ and $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \dots, \boldsymbol{\sigma}_6$), while the input to the decoder is randomly drawn from the corresponding Gaussian distributions. Being a graph variance of the VAE, the loss consists of two parts: the term $\text{KL}(q(\mathbf{z} \mathbf{A}, \mathbf{v}) p(\mathbf{z}))$ of <i>Loss 1</i> which reflects how well the output Gaussian approaches the standard Gaussian distribution, $p(\mathbf{z})$ (zero mean and identity covariance), while the term $\mathbb{E}_{q(\mathbf{z} \mathbf{A}, \mathbf{v})}[\log p(\mathbf{A} \mathbf{z})]$ of <i>Loss 2</i> corresponds to the quality of reconstruction of the adjacency matrix, $\mathbf{A}$	154
7.4	The overall structure of the proposed GRCF-GAN.	155

7.5 The AUC and AP scores of our GRCF-GAN and standard VGAE, with m_{layer} ranging from 256 to 2048. The solid lines show the AUC and AP of our GRCF-GAN in testing, whereas the dashed lines designate for the VGAE. Since spectral clustering [10] is a typical baseline when not using GCNs, spectral clustering results from [11] are shown in dotted lines. . . . 156

Chapter 1

Introduction

1.1 Motivation and objectives

“The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man’s mind.”

– James Clerk Maxwell (1850)

From the beginning of human history, we have been observing signals surrounded by various types of noise and uncertainties, and by trial and error, have gradually learnt to use *probability* to “guess” the true inner relationship the world hides to us, and simultaneously explain how a phenomenon or signal of interest is *generated* from this inner relationship. Fast forward tens of thousands of years, the notion of generating from some basic probability distributions is the key idea behind *probabilistic generative models* in machine learning and statistical signal processing. In other words, given the observed data, generative models typically assume latent variables that depict the cause from the inner relationship, together with a generating process that is able to replicate similar signals given the observed data.

More importantly, the area of probabilistic generative models is one of the most promising

fields to achieve the ultimate goal of artificial intelligence (AI), i.e., models and algorithms with a human-level understanding of signals¹. It does not require the labels associated with data, thus avoiding the high cost of acquiring data labels in many applications, and learns to model data in an un-supervised way. Furthermore, it benefits from treating signals in a probabilistic way, which relieves the notorious over-fitting issue and also indicates the uncertainty when performing decision making. On the other hand, its ability to replicate similar signals allows to clearly understand both origins and evolutions of signals, in a similar logic to the quote by Richard Feynman:

What I cannot create, I do not understand.

Moreover, this typically enables the latent variables to possess high-level or semantic information that can be further employed in down-stream tasks such as predictions.

The generative models can be generally formulated as

$$\mathcal{X} =^d \mathcal{G}\{\mathcal{Z}\}, \quad (1.1)$$

where $\mathcal{X}$ is a random variable representing observed signals, $\mathcal{Z}$ is the latent (random) variable and $\mathcal{G}$ denotes the generating process, either deterministic or probabilistic. Its probability relationship has been widely used and is given by,

$$p_{\mathcal{X}}(\mathbf{x}) = \int_{\mathcal{Z}} p_{\{\mathcal{X}=\mathcal{G}\{\mathcal{Z}\}\}}(\mathbf{x}|\mathbf{z})p_{\mathcal{Z}}(\mathbf{z})d\mathbf{z}. \quad (1.2)$$

Different assumptions on $\mathcal{Z}$ and $\mathcal{G}$ thus define different generative models. Typically, $\mathcal{Z}$ and $\mathcal{G}$ are parametrised for the ease of optimisation, and the parameters can exist in various forms and can thus be estimated either deterministically or stochastically (Bayesian). For example, these parameters can be the mean, $\boldsymbol{\mu}$, or the covariance matrix, $\boldsymbol{\Sigma}$, of a standard Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of $\mathcal{Z}$; these can also be the parameters of deep neural networks given by $\mathcal{G}$. It needs to be pointed out that there are still several non-parametric generative models which operate by, for example, only assuming some smooth properties in kernel density estimation models [12] or consistency in Dirichlet process [13]; in practice, those models, however, typically cannot scale well in high dimensions, and are thus not

¹<https://openai.com/blog/generative-models/>

discussed in this thesis. Therefore, in this thesis, we focus on introducing and improving parametric probabilistic generative models.

Furthermore, the generating process $\mathcal{G}$ can be both linear and non-linear. More specifically, for a linear transform of $\mathcal{G}$, the generated distribution, $\mathcal{X}$, can be often obtained explicitly, and this type of models is called *explicit generative models* [14]. Obviously, explicit generative models can clearly indicate the generation procedure, as well as formulate closed-form probability density functions (pdfs), thus allowing for easily understanding/explaining the observed data. On the other hand, the linearity of those models severely limits the applications. For data which either reside in high dimensional spaces or exhibit complicated structures, a non-linear $\mathcal{G}$ is preferable. The non-linear operation in $\mathcal{G}$ is typically implemented using the neural networks, and the resulted generative distributions $\mathcal{X}$ not necessarily have explicit pdfs, and are thus referred to as *implicit generative models* [14], or nowadays *deep generative models*.

It is the fact that we basically cannot have² a “free lunch” for all types of signals without any explicit or implicit assumption on the underlying structure. In other words, an important basis is required through the domain or prior knowledge when dealing with different signal types. This means that both explicit and implicit generative models have their own pros and cons in different applications. Generally speaking, explicit models exhibit good *explainability*, that is, the resulting model can be well explained, whereas implicit models are more preferable for the *expressibility*, that is, the resulting model is powerful enough to express complicated distributions by replicating similar signals.

This thesis addresses both explicit and implicit generative models, with the aim of improving their respective generative capability for different types of signals. More specifically, it is well-known that the Gaussian distribution is the workhorse in generative models, with the popularity of the Gaussian distribution mainly arising from its mathematical convenience such as conjugate properties in Bayesian inference, as well as the support of the central limit theorem. However, the Gaussian assumption is most often inadequate in real-world applications. More importantly, models with Gaussian assumptions are extremely

²This refers to the *No Free Lunch* theorem, which states that when considering all possible applications, any two algorithms perform equivalently [15].

sensitive to small deviations from the Gaussian assumptions. In other words, when it comes to various real-world applications, a small violations of the Gaussian assumption can result into highly biased models and results.

To pave a path of balancing the computational (mathematical) ease and the requirements of real-world applications, advanced models built upon the Gaussian distribution have to consider three essential perspectives of generative models:

- **Meaningful generating process:** It is the fact that an arbitrary transform from the Gaussian distribution can result in a meaningless and even incomputable distribution. Therefore, we need to choose or develop generative models by reflecting our belief on signals, so as to make it possible to interpret and reason on the resulting estimated models.

Within explicit generative models, attempted improvements aim to enhance robustness of the models, which allows for some mild violations (outliers) of the Gaussian assumptions. The family of symmetric elliptical distributions [16] is one of such robust³ generalisations that inherits the location-scales feature of the Gaussian distribution but is much more robust to assumption violations, thus allowing for more flexibility in applications. However, this type of distributions still does not imply any asymmetric property, yet signals in many applications including finance, biometric and audio scenarios, are not necessarily symmetric. To this end, this thesis proposes a skewed version of the elliptical distribution, to accommodate asymmetric signals. Furthermore, to deal with multi-modal signals, the finite mixture model, especially Gaussian mixture model (GMM), can be employed, and in this thesis, we systematically propose a much more powerful model, i.e., elliptical mixture model (EMM) which exhibits many properties that are desirable in practical applications. Furthermore, to model highly complicated data, we can choose non-linear transforms of the Gaussian distribution, with the resulting distributions typically not having explicit pdfs, and thus belonging to implicit generative models, including variational auto-encoders (VAEs) [18] and generative adversarial nets (GANs) [19].

³Robustness in statistics means that an estimation method is able to avoid completely breakdown given the existence of outliers or model assumption violations, whilst minimising the decrease of estimation efficiency [17], which shall be introduced in detail in Chapter 2.

This thesis further proposes to seamlessly and naturally combine the GANs and the auto-encoders, under one unifying umbrella, thus benefiting from the advantages of both VAEs and GANs.

- **Comprehensive statistical metrics:** As generative models aim at mimicking observed signal behaviours and outputting similar signals, it is crucial to have means of measuring the discrepancy between the generated signals and observed signals, as a way of promoting similarity.

One of the most popular such metrics is the Kullback–Leibler (KL) divergence, a special case of the Bregman divergence [20]. The KL divergence enjoys the convenience of computation, which is equivalent to the widely applied maximum log-likelihood (MLL) framework. However, the KL divergence (even the Bregman divergence) operates through a “bin-to-bin” comparison between two pdfs, which yields meaningless values when two pdfs are well separated or are supported in manifolds of different dimensions [21]. This deficiency could lead to local optima of explicit generative models and the issue of gradient vanishing in implicit generative models. Recently, the Wasserstein distance, which employs a “cross-bin” comparison, has offered many practical benefits by virtue of the reflection of sample space [22]. It provides a comprehensive comparison between two distributions, but suffers from high computational complexity. This thesis further proposes a relaxed Wasserstein distance for the ease of computations. Furthermore, for a rigorous and comprehensive comparison of any two pdfs, on even distributions which do not have a valid pdf, this thesis proposes to employ the characteristic function (CF) of distributions, which is a powerful tool that contains all the information relevant to the distributions and is computationally affordable.

- **An efficient and stable solver:** The generative models are typically parametrised either explicitly or implicitly to allow for adjustments and optimisations. The solver is therefore crucial to optimise the generative model and perform the best fitting.

Unfortunately, solving most of the generative models does not necessarily yield a closed-form optimum, which depends on the format of generative models and the choice of statistical metrics. Typically, the solvers operate in an iterative way. For

example, a coordinate descent method called expectation-maximisation (EM) has been the *de facto* in solving the GMMs under the KL divergence, whilst other models may be solved by gradient descent methods. However, the gradient descent strategy does not ensure a closed boundary within probability spaces. In other words, during gradient descent, the updated parameters may not formulate a valid probabilistic generative model. For example, when updating a covariance matrix of a Gaussian distribution by gradient descent, it does not necessarily remain positive semi-definite (psd) even when its initialisation is psd. Thus, the updated result is ill-posed given the Gaussian generative model. An elegant way of solving this is to establish a statistical manifold and to implement gradient descent through manifold operations. The natural gradient, proposed by Amari [23], is a special case of this, and this thesis explicitly establishes statistical manifolds for various generative models, also achieving fast convergence.

1.2 Thesis outline and original contributions

This thesis sets out to improve generative models in terms of the above three key perspectives. The overall flow of this thesis is shown in Figure 1.1. Generally, the Gaussian distribution can be analysed to be generated from the distribution called χ^2 distribution, which shapes the tails of Gaussian pdf to be quite light, so that the Gaussian distribution does not give any chance or possibility to the data samples far away from its centre. In other words, it thus does not allow signals violating the assumption. To address this issue, the elliptical distribution extends the underlying $\sqrt{\chi^2}$ distribution to an arbitrary one-dimension positive random variable $\mathcal{R}$, which is much more robust to assumption violations and is also flexible, because it includes many existing important distributions such as Cauchy, Student- t , generalised Gaussian, logistic, and α -stable distributions. However, the scalar random variable, $\mathcal{R}$, imposes equal weights in all the dimensions. The resulting elliptical distribution is symmetric and thus cannot well accommodate skewed data.

Chapter 3 therefore focuses on the problem of generalising symmetric distributions, given the fact that a large class of modern probabilistic learning systems assumes symmet-

ric distributions but real-world data tend to obey skewed distributions and are thus not always adequately modelled through symmetric distributions. To address this issue, elliptical distributions are increasingly used to generalise symmetric distributions, and further improvements to skewed elliptical distributions have recently attracted much attention. However, existing approaches are either hard to estimate or have complicated and abstract representations. To this end, we propose to employ the von-Mises-Fisher (vMF) distribution to obtain an explicit and simple probability representation of the skewed elliptical distribution. This is shown not only to allow us to deal with non-symmetric learning systems, but also to provide a physically meaningful way of generalising skewed distributions. For rigour, our extension is proved to share important and desirable properties with its symmetric counterpart. We also demonstrate both theoretically and through examples, that the proposed vMF distribution is both easy to generate and stable to estimate.

Furthermore, the above distributions are uni-modal and thus exhibit limitations for multi-modal applications, e.g., signals with multiple clusters and peaks. Therefore, in Chapters 4 and 5, we propose to optimise the EMMs, in order to fit any real-valued distribution and also to promise enhanced robustness, flexibility and stability over the widely employed GMM.

Chapter 4 aims to solve the problem encountered with existing studies based on the EMM, which are restricted to several specific types of elliptical distributions, that are not supported by general solutions or systematic analysis frameworks; this significantly limits the rigour in the design and power of EMMs in applications. To this end, we propose a novel general framework for estimating and analysing the EMMs, achieved through Riemannian manifold optimisation. First, we investigate the relationships between Riemannian manifolds and elliptical distributions, and the so established connection between the original manifold and a reformulated one indicates a mismatch between these manifolds, a major cause of failure of the existing optimisation strategies for solving general EMMs. We next propose a universal solver which is based on the optimisation of a re-designed cost and prove the existence of the same optimum as in the original problem; this is achieved in a simple, fast and stable way. We further calculate the influence functions of the EMM as theoretical bounds to quantify robustness to outliers. Comprehensive numerical results

demonstrate the ability of the proposed framework to accommodate EMMs with different properties of individual functions, in a stable way and with fast convergence speed. Finally, the enhanced robustness and flexibility of the proposed framework over the standard GMM are demonstrated both analytically and through comprehensive simulations.

Chapter 5 proceeds to solve EMMs by the Wasserstein distance which provides a much more desirable optimisation space compared to the widely adopted KL divergence that is employed in Chapter 4. More specifically, we provide a stable solution to the EMMs that is both robust to initialisations and reaches a superior optimum by adaptively optimising along a manifold of a relaxed Wasserstein distance. To this end, we first provide a unifying account of computable and identifiable EMMs, which serves as a basis to rigorously address the underpinning optimisation problem. Due to a probability constraint, solving this problem is extremely cumbersome and unstable, especially under the Wasserstein distance. To relieve this issue, we introduce an efficient optimisation method which operates on a statistical manifold defined under a relaxed Wasserstein distance, which allows for explicit metrics and computable operations, thus significantly stabilising and improving the EMM estimation. We further propose an adaptive method to accelerate the convergence. Experimental results demonstrate the excellent performance of the proposed EMM solver.

The above proposed distributions all belong to explicit generative models. Although we can benefit from those to clearly understand the generating process, for example, the skewed direction of signals when using vMF elliptical distributions or the cluster information when using EMMs, these are still limited when fitting complicated and high-dimension signals.

Chapter 6 therefore employs deep neural nets to propose a new structure of combining GANs and auto-encoders, as an implicit generative model. More specifically, one called integral probability metric (IPM) equips GANs with the necessary theoretical support for comparing statistical moments in an embedded domain of the critic, while stabilising their training and mitigating model collapse issues. For enhanced intuition and physical insight, we introduce a generalisation of IPM-GANs which operates by directly comparing

probability distributions rather than their moments. This is achieved through CFs, a powerful tool that uniquely comprises all the information about any general distribution. For rigour, we first prove theoretically the ability of the CF loss to compare probability distributions, and establish physical meaning of the phase and amplitude of CFs. An optimal sampling strategy is then developed to calculate the CFs, and an equivalence between the embedded and data domains is proved when a reciprocal theory holds. We therefore seamlessly combine the IPM-GAN with an auto-encoder structure, equipped with an advanced anchor architecture, which adversarially learns a semantic low-dimensional manifold for both generation and reconstruction. This efficient structure, called reciprocal CF GAN (RCF-GAN), uses only two modules and a simple training strategy, to achieve the state-of-the-art bi-directional generation. Experiments demonstrate the superior performance of the RCF-GAN on image generation, reconstruction and interpolation.

Chapter 7 further applies the generative models over irregular domains, i.e., the graph domain, to address the widely concerned graph learning problem. In many situations where the graph topology cannot be directly observed or even when the data is partially observed, the inference of graph structure is a key first step. Unfortunately, given the observed graph data, graph learning is an ill-posed problem, of which a reasonable solution requires some priors. Thus, it is natural to connect and summarise those techniques under the umbrella of probabilistic generative models. We herein propose to an advanced graph RCF-GAN (GRCF-GAN) for graph topology inferencing, with the state-of-the-art performances achieved over a range of applications.

It is our hope that this thesis will serve to provide a new and unified perspective on combining explicit and implicit generative models, paving the way for future work to embark upon this intrinsic combination.

1.3 Publication list

The following publications are the outcomes of the material presented in the thesis:

Submitted

- **Shengxi Li**, Danilo Mandic. “Von Mises-Fisher Elliptical Distribution”, *submitted to IEEE Transactions on Neural Networks and Learning Systems*.
- **Shengxi Li**, Zeyang Yu, Min Xiang, Danilo Mandic. “Reciprocal GAN through Characteristic Functions (RCF-GAN)”, *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (under major revision)*.

Peer-reviewed Journals

- **Shengxi Li**, Zeyang Yu, Danilo Mandic. “A Universal Framework for Learning the Elliptical Mixture Model”, *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2020.
- Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, **Shengxi Li**, Anthony G. Constantinides. “Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications”, *Foundations and Trends in Machine Learning*, vol. 13: no. 4, pp 332-530, 2020.
- Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, **Shengxi Li**, Anthony G. Constantinides. “Data Analytics on Graphs Part II: Signals on Graphs”, *Foundations and Trends in Machine Learning*, vol. 13: no. 2-3, pp 158-331, 2020.
- Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, **Shengxi Li**, Anthony G. Constantinides. “Data Analytics on Graphs Part I: Graphs and Spectra on Graphs”, *Foundations and Trends in Machine Learning*, vol. 13: no. 1, pp 1-157, 2020.

Peer-reviewed Conferences

- **Shengxi Li**, Zeyang Yu, Min Xiang, Danilo Mandic. “Reciprocal Adversarial Learning via Characteristic Functions”, *in Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 217-228, 2020. (Note: Spotlight, top 4%)

- **Shengxi Li**, Zeyang Yu, Min Xiang, Danilo Mandic. “Solving General Elliptical Mixture Models through an Approximate Wasserstein Manifold”, in *Proceedings of Conference of Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 4658-4666, 2020. (Note: Oral, top 5%)
- Kriton Konstantinidis, **Shengxi Li**, Danilo Mandic. “Kernel Learning with Tensor Networks”, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2920-2924, 2021.
- Zeyang Yu, **Shengxi Li**, Danilo Mandic. “Widely Linear Complex-valued Autoencoder: Dealing with Noncircularity in Generative-discriminative Models”, in *Proceedings International Conference on Artificial Neural Networks (ICANN)*, pp. 339-350, Springer, Cham, 2019.
- Sayed Pouria Talebi, Stefan Werner, **Shengxi Li**, Danilo Mandic. “Tracking Dynamic Systems in α -stable Environments”, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4853-4857, 2019.

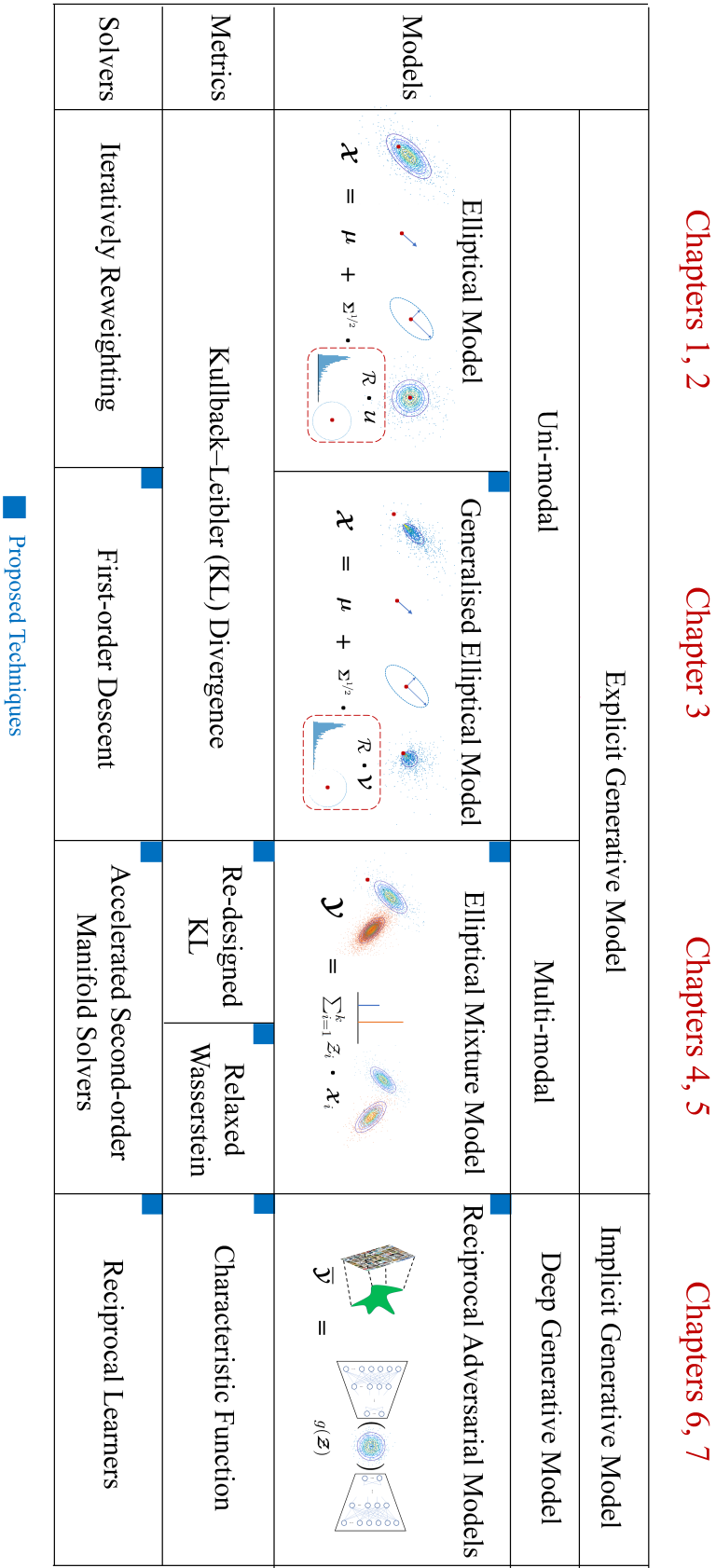


Figure 1.1: Thesis structure, with the flow aiming to demystify generative models by tractable generating process, statistical metrics, and efficient solvers.

Chapter 2

Background

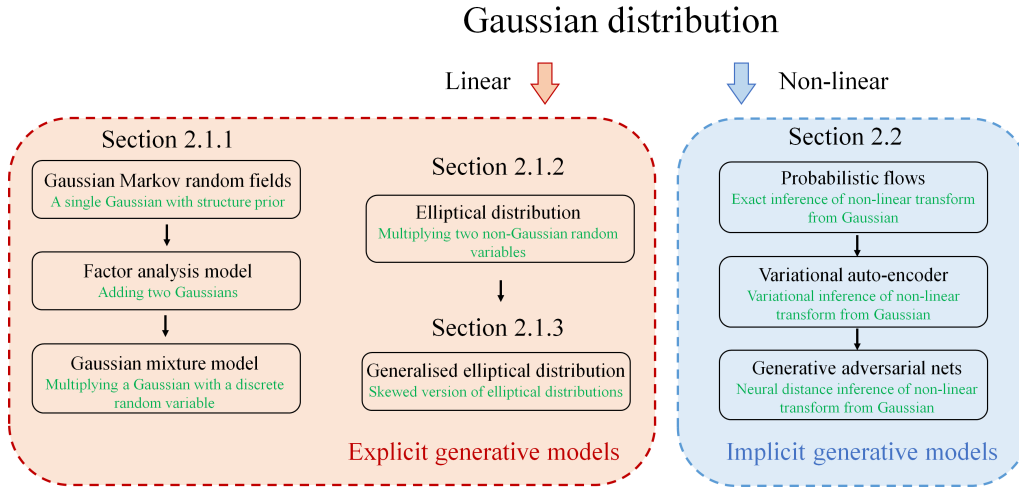
“Everyone believed in the normal distribution, the mathematicians because they thought it was an experimental fact, the experimenters because they thought it was a mathematical theorem ... Normality is a myth; there never has, and never will be, a normal distribution.”

– Robert Charles Geary (1947)

In this chapter, we introduce the basic background for the material covered in this thesis. We first introduce typical explicit/implicit generative models in Sections 2.1 and 2.2, with the overall principle summarised in Figure 2.1. We then explain statistical distances in Section 2.3 that have been widely applied to evaluate and optimise the models, as well as the basics of manifold optimisations in Section 2.4, which have been receiving increasing attention in modern constrained optimisation.

2.1 Explicit Generative Models

The popularity of the Gaussian distribution arises mainly from its simple formulation and mathematical convenience in manipulation, such as conjugate properties in Bayesian inference. With the support of the central limit theorem, Gaussian



Impulsive signals are not rare in practice, for example, they characterise financial crashes or flash crashes as a result of wrong manipulations, and are often present in physiological signals such as in electrocardiography. In such applications, models built upon the Gaussian distribution almost invariably fail to yield accurate predictions and inferences. It is therefore necessary to introduce a more general class of “robust” distributions which are capable of dealing with general random variables, a prerequisite for efficient machine learning. Another issue is the limited flexibility of Gaussian-based models, which is prohibitive to their application in rapidly emerging scenarios based on multi-faceted data with unbalanced channels; sources of such imbalance may be due to different natures of the data channels involved, different powers in the constitutive channels, or temporal misalignment, all commonplace in practice.

Despite the existence of robust techniques, developing a generalised distribution starting from the Gaussian distribution is not straightforward because the generalisation should be as tractable and meaningful as the Gaussian distribution, so that many applications based on the Gaussian building blocks can also be established based on the generalised distribution, whilst maintaining the robustness and flexibility. The elliptical distribution is one of the most significant such generalisations. When possessing an explicit probability density function, the elliptical distribution is formulated as tractably as the Gaussian distribution, and it also belongs to the location-scale family. In other words, it is characterised by the parameters of a location (mean) vector and a scale (scatter) matrix. The additional flexibility of the elliptical distribution arises from the generator density function, as its variations yield many important distributions, such as Laplacian, Cauchy, Weibull, Gamma, generalised Gaussian, Student- t , hyperbolic and α -stable distributions, to name but a few. More importantly, an appropriate density generator can also result in a robust M-estimator to further reinforce the robustness. The elliptical distribution also enjoys many desirable properties in the manipulation and also admits marginal and conditional elliptical distributions. We therefore review the existing elliptical

distributions in Sections 2.1.2 and 2.1.3.

2.1.1 Gaussian Models

Given the fact that the real world signals do not necessarily satisfy the single Gaussian distribution due to noise, skewness, complicated structures, to name but a few, a typical strategy is to employ the Gaussian distribution as a building block and develop Gaussian linear models through linear transforms. Our brief introduction of explicit generative models cannot by any means review all the generative models, but focus on a review of several representative and relevant Gaussian linear models first.

Gaussian Markov Random Fields (GMRFs)

The Gaussian Markov random field (GMRF) has been widely applied in spatial statistics and analysis [27] such as in image segmentation [28] and understanding [29]. Basically, GMRF extends the naive Gaussian model by imposing neighbouring structures as a prior, and it defines a random field $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ that satisfies

$$p(x_i | \{x_j\}_{j \neq i}) = p(x_i | \{x_j\}_{j \in \Omega_i}), \quad (2.1)$$

where i and j index two random variables within the field $\mathbf{X}$, and Ω_i denotes the neighbourhood of x_i . The definition of a neighbourhood is typically accompanied with (undirected) probabilistic graphical models, so that the random variables (nodes) are defined as neighbours when they are connected in graphs, i.e., satisfying the Markov property. Therefore, (2.1) assumes signals to be conditioned solely based on their neighbours, and consequently, GMRF allows for priors of structural neighbouring information that is typically sparse and smooth, and thus capable of extracting useful information and suppressing signal outliers.

The conditional distribution of a Gaussian still follows the Gaussian distribution,

and without loss of generality, we separate a random variable $\mathbf{x}$ as $[\mathbf{x}_1, \mathbf{x}_2]^T$ to formalise the conditional Gaussian distribution,

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}), \quad (2.2)$$

so that the conditional distribution can be denoted as (Chapter 2.3.1 of [30])

$$\mathbf{x}_1 | \mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}), \quad (2.3)$$

where

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \quad \boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \quad (2.4)$$

On the other hand, it is more convenient to use the precision matrix, $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$, than the covariance matrix to represent the conditional distribution. Upon rewriting in terms of the precision matrix, we obtain $\boldsymbol{\Sigma}_{1|2}$ as precisely the Schur complement with the following block-wise matrix property [31],

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} = \boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_{11}^{-1} & -\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} \\ -\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1} & \boldsymbol{\Sigma}_{22}^{-1} + \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} \end{bmatrix}. \quad (2.5)$$

Therefore, the block matrix $\mathbf{Q}_{11}$ depicts the precision matrix $\boldsymbol{\Sigma}_{1|2}^{-1}$ of the conditional Gaussian distribution. On the other hand, compared with the covariance matrix, $\boldsymbol{\Sigma}$, the elements in $\mathbf{Q}$, to be zero or not, truly indicate the independence when $\mathbf{x}_1$ conditioned on $\mathbf{x}_2$, avoiding the misleading influence on causation, as *correlation does not mean the causation*¹.

Besides its practical applications, GMRF also connects many generative models. For example, starting from the GMRF, we may also have an infinite dimensional version of the Gaussian process. On the other hand, when the neighbouring is defined based on temporal rather than spatial relationships, an auto-regressive model is

¹Two random variables can show a strong correlation when they are all highly influenced by a middle random variable, however, they are not the causation of one another.

obtained [32]. Gaussianity can also be generalised by other distributions to form the Markov random field (MRF), such as the Ising model or restricted Boltzmann machine [33]. For conciseness and clarity, we omit explorations of this direction in the thesis.

Factor Model

Another important model in probabilistic generative models is the factor analysis model, which forms the basis of many important tools, such as the probabilistic principal component analysis [34].

More specifically, the observed data is assumed to be generated via a factor model that can be represented as

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\epsilon}, \quad (2.6)$$

where $\mathbf{W}$ is a matrix, $\mathbf{z}$ is a vector of latent variables (or factor loadings) which is Gaussian distributed with zero mean and an identity covariance matrix ($\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$), while $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda})$ is also Gaussian distributed with a diagonal covariance, $\boldsymbol{\Lambda}$, but is independent of $\mathbf{z}$.

On the basis of this factor model, it is easy to obtain the distribution of $\mathbf{x}$ as

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^T + \boldsymbol{\Lambda}) \quad (2.7)$$

The factor model can be estimated through MLL of $\mathbf{x}$, which results into the probabilistic principle component analysis [34]. As shall be shown, the MLL estimation is equivalent of using the Kullback–Leibler (KL) divergence as distance metrics.

Furthermore, to find the latent variables given observed signals, we can choose to maximise the posterior distribution of $\mathbf{z}$ from the factor model, as follows

$$\begin{aligned} \arg \max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}) &\propto \arg \max_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) = \arg \max_{\mathbf{z}} (\log(p(\mathbf{x}|\mathbf{z})) + \log(p(\mathbf{z}))) \\ &= \arg \min_{\mathbf{z}} ((\mathbf{x} - \mathbf{W}\mathbf{z})^T \boldsymbol{\Lambda}^{-1} (\mathbf{x} - \mathbf{W}\mathbf{z}) + \mathbf{z}^T \mathbf{z}). \end{aligned} \quad (2.8)$$

Upon further assumption of an isotropic diagonal, $\mathbf{\Lambda} = \sigma^2 \mathbf{I}$, (2.8) can be simplified by denoting $\mathbf{y} = \mathbf{W}\mathbf{z}$ as

$$\arg \max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}) = \arg \min_{\mathbf{y}} (||\mathbf{x} - \mathbf{y}||^2 + \sigma^2 \mathbf{y}^T (\mathbf{W}\mathbf{W}^T)^{-1} \mathbf{y}). \quad (2.9)$$

This means that given the observed signals, $\mathbf{x}$, we aim at finding the corresponding $\mathbf{z}$ which simultaneously minimises the reconstruction error $||\mathbf{x} - \mathbf{y}||^2$ and satisfies some smoothness or sparsity constraint given by $(\mathbf{W}\mathbf{W}^T)^{-1}$.

For rigour, the posterior of $p(\mathbf{z}|\mathbf{x})$ can be explicitly obtained. We first rewrite (2.6) by stacking $\mathbf{x}$ and $\mathbf{z}$ together and obtain the augmented Gaussian distribution (please refer to Chapter 2.3.3 of [30]) as follows,

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{W}\mathbf{W}^T + \mathbf{\Lambda} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{I} \end{bmatrix}). \quad (2.10)$$

Using the results of conditional Gaussian distribution from (2.4), we can obtain the posterior as

$$\mathbf{z}|\mathbf{x} \sim \mathcal{N}(\mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \mathbf{\Lambda})^{-1}\mathbf{x}, \mathbf{I} - \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \mathbf{\Lambda})^{-1}\mathbf{W}) \quad (2.11)$$

It is interesting to note that even for zero-mean $\mathbf{z}$ and $\mathbf{x}$, the mean of the posterior $p(\mathbf{z}|\mathbf{x})$ is not zero and is related to the observed signals, $\mathbf{x}$. However, for the covariance matrix, the term $\mathbf{I} - \mathbf{W}^T(\mathbf{W}\mathbf{W}^T + \mathbf{\Lambda})^{-1}\mathbf{W}$ that indicates the uncertainty, is irrelevant to the observed signals $\mathbf{x}$.

The factor model is actually a special form of the linear Gaussian models [35], and it can be naturally extended by employing some non-linear operations such as polynomials $\mathbf{x} = \sum_{i=0}^k \mathbf{W}^i \mathbf{z}_i$ and even neural networks $\mathbf{x} = g(\mathbf{z})$. However, the cost of involving non-linear transforms is that the posterior $p(\mathbf{z}|\mathbf{x})$ is no longer explicit, and requires a lower bound to be employed in optimisation, as shown shortly.

Gaussian Mixture Model (GMM)

The above models fit signals to be within a single modal, and in order to fit multi-modal signals, a finite mixture model can be employed. One of the most widely applied finite mixture models is the GMM, which consists of several Gaussian distributions (clusters). The GMM can reflect on many important cues in the observed signals, such as locations, scales and portion of clusters, as well as the spread of clusters in signals.

For generality, we assume that there are k mixtures (Gaussian distributions) in the GMM model, latent variables $\mathcal{Z}_i \in \{0, 1\}$ are binary, and the probability of choosing the i -th mixture is denoted by $p(\mathcal{Z}_i = 1) = \pi_i$, so that $\sum_{i=1}^k \mathcal{Z}_i = 1$ and $\sum_{i=1}^k \pi_i = 1$. If we employ the random variable $\mathcal{Y}$ to denote the set of Gaussian distributions, that is, the GMM, and use the notation $\mathcal{X}$ to denote single Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, then

$$\mathcal{Y} = {}^d \sum_{i=1}^k \mathcal{Z}_i \mathcal{X}_i \sim \sum_{i=1}^k \pi_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \quad (2.12)$$

Similar to the factor model, we can also choose to use the MLL to estimate the GMM. Given a set of independent and identically distributed (iid) observed signals $\{\mathbf{x}_j\}_{j=1}^n$, we can have the log-likelihood of the GMM as follows,

$$L = \sum_{j=1}^n \log \left(\sum_{i=1}^k \frac{\pi_i}{\sqrt{(2\pi)^m \det(\boldsymbol{\Sigma}_i)}} \exp \left(-\frac{1}{2} (\mathbf{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i) \right) \right). \quad (2.13)$$

Obviously, by maximising L , we cannot obtain closed-form solutions $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \pi_i\}_{i=1}^k$ due to the log of sum operation. An alternative is to use the well-known expectation maximisation (EM) algorithm [36]. We use the notation in particular $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta})$ to denote that the posterior distribution is parametrised by $\boldsymbol{\theta}$, where here $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \pi_i\}_{i=1}^k$. Then, the EM algorithm iterates the following E step and M step

until convergence,

$$\begin{aligned}
\mathbf{E} \text{ step: } & \text{Calculate posterior : } p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}}) = \frac{\pi_i^{\text{old}} \mathcal{N}_j(\boldsymbol{\mu}_i^{\text{old}}, \boldsymbol{\Sigma}_i^{\text{old}})}{\sum_{l=1}^k \pi_l^{\text{old}} \mathcal{N}_j(\boldsymbol{\mu}_l^{\text{old}}, \boldsymbol{\Sigma}_l^{\text{old}})} \\
\mathbf{M} \text{ step: } & \text{Update parameters : } \pi_i^{\text{new}} := \frac{\sum_{j=1}^n p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}})}{n} \\
& \boldsymbol{\mu}_i^{\text{new}} := \frac{\sum_{j=1}^n p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}}) \mathbf{x}_j}{\sum_{j=1}^n p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}})} \\
& \boldsymbol{\Sigma}_i^{\text{new}} := \frac{\sum_{j=1}^n p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}}) (\mathbf{x}_j - \boldsymbol{\mu}_i) (\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^n p(z_i = 1 | \mathbf{x}_j; \boldsymbol{\theta}^{\text{old}})}
\end{aligned} \tag{2.14}$$

We will show later in Section 2.2.2 that the EM algorithm is a type of variational inference which maximises a lower bound, called the evidence lower bound (ELBO), whereby the original log-likelihood cannot be solved explicitly. Briefly speaking, in the GMM problem, ELBO turns the intractable log-sum operation into the solvable sum-log format, so that at the maximisation step (i.e., the M step), that is, into the update on parameters $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \pi_i\}_{i=1}^k$ can be obtained explicitly.

On the other hand, due to its exponential function, the GMM can be also regarded as mixtures of RBFs and therefore be generalised as deep RBF neural networks [25, 26]. As shall be shown in the next section, the elliptical distribution provides various choices of kernels rather than the exponential format in the RBF kernel.

2.1.2 Elliptical Distributions

Robustness

The above Gaussian related models inevitably suffer from the sensitivity to outliers or model violations that arise from the intrinsic nature of the Gaussian distribution. Thus, in many applications the Gaussian distribution cannot hold, and what makes things worse is that when slightly violating the Gaussian assumption, the resulting estimation can be severely affected and even yield completely wrong results. For example, although the central limit theorem supports the Gaussian distribution

resulted from sufficiently adding up iid random variables, its requirement on the sufficient number of random variables may not hold.

More importantly, the central limit theorem also requires that the random variable is with a finite variance, and in cases when random variables are with infinite variance (impulsive behaviours), the central limit theorem fails and the generalised central limit theorem is applied to eventually result in the distribution called α -stable distribution [16], instead of the Gaussian distribution. Impulsive signals are not rare in practice, for example, they exist in the financial crashes or flash crashes as a result of wrong manipulations, where models built upon the Gaussian distribution almost fail to make predictions and inferences. It is therefore necessary to employ some “robust” models to deal with the possible violations.

A rigorous account of the robustness of estimators may involve robust statistics [17], which is employed to evaluate *whether a model is capable of avoiding a complete breakdown given the existence of model violations, whilst minimising the decrease of estimation efficiency*. The *influence function* (IF) is one of the most important criteria and bounded IFs indicate the robustness [17]. Denoting the clean distribution as $\mathcal{X}_\theta$ and the contaminated distribution as $\mathcal{X}_{\theta,\epsilon}$ by a portion ϵ of outliers at $\mathbf{x}_0$, the IF is defined as follows,

$$F_{\hat{\theta}}(\mathbf{x}_0; \mathcal{X}_\theta) = \lim_{\epsilon \rightarrow 0} \frac{\hat{\theta}(\mathcal{X}_{\theta,\epsilon}) - \hat{\theta}(\mathcal{X}_\theta)}{\epsilon} = \left[\frac{\partial \hat{\theta}(\mathcal{X}_{\theta,\epsilon})}{\partial \epsilon} \right]_{\epsilon \rightarrow 0}, \quad (2.15)$$

where $\hat{\theta}(\mathcal{X}_{\theta,\epsilon})$ represents the parameter $\hat{\theta}$ estimated under the contaminated distribution $\mathcal{X}_{\theta,\epsilon}$, and $\hat{\theta}(\mathcal{X}_\theta)$ denotes the estimation without outliers. Generally speaking, the IF depicts the estimation deviation due to the existence of a small portion of outliers. We now provide an insight into the IF through a simple example. It is well-known that when using a single Gaussian distribution and estimating its mean, $\boldsymbol{\mu}$, via MLL, given observed iid signals $\{\mathbf{x}\}_{j=1}^n$, we have the closed-form solution of $\boldsymbol{\mu} = \frac{\sum_{j=1}^n \mathbf{x}_j}{n}$, which is the mean value of signals. However, when one sample $\mathbf{x}_1$ (i.e., a small portion of outliers) within the signals violates the Gaussian assumption

and is, for example, infinitely large ($\mathbf{x}_1 \rightarrow \infty$) in some cases, the estimation can be severely distorted due to $\boldsymbol{\mu} = \frac{\mathbf{x}_1 + \sum_{j=2}^n \mathbf{x}_j}{n} \rightarrow \infty$, which indicates an unbounded IF. Given this unbounded IF, the Gaussian distribution estimated by MLL is therefore not robust. In the following, we shall introduce a family of distributions named elliptical distributions, which include both robust and non-robust distributions, as well as an extensive number of important distributions such as the α -stable as mentioned above, Laplacian, t and logistic distributions [16].

Elliptical distribution

More specifically, a random variable, $\mathcal{X} \in \mathbb{R}^m$, is said to exhibit an elliptical distribution if and only if it admits the following stochastic representation [16],

$$\mathcal{X} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{R}\boldsymbol{\Lambda}\boldsymbol{S}, \quad (2.16)$$

where $\mathcal{R} \in \mathbb{R}^+$ is a non-negative real scalar random variable which models tail properties of the elliptical distribution, and $\boldsymbol{S} \in \mathbb{S}^{(m'-1)}$ is a random vector which is uniformly distributed on a unit spherical surface² with the pdf within the class of $2\pi^{-m'/2}\Gamma(m'/2)$. The product $\mathcal{R} \cdot \boldsymbol{S}$ thus represents a spherical distribution, of which the “spread” or distribution tail is dominated by $\mathcal{R}$, whilst the uniform distribution $\boldsymbol{S}$ supports the distribution to m' dimensions. The constant parameter $\boldsymbol{\Lambda}$ distorts the spherical distribution to an ellipse and the location vector $\boldsymbol{\mu}$ then translates the distribution. For a comprehensive review of elliptical distributions, we refer to [16].

When $m' = m$, that is, for a non-singular scatter matrix $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$, the pdf for elliptical distributions does exist and has the following form

$$p(\mathbf{x}) = \underbrace{2\pi^{-\frac{m}{2}}\Gamma(\frac{m}{2})}_{c_m} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} g(\underbrace{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}_t). \quad (2.17)$$

In (2.17), the term c_m serves as a normalisation term and relates solely to m . We

²The term $\mathbb{S}^{m'-1}$ is defined as $\mathbb{S}^{m'-1} := \{\mathbf{x} \in \mathbb{R}^{m'} : \mathbf{x}^T \mathbf{x} = 1\}$.

denote the Mahalanobis distance $(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ by t . The density generator, $g(t)$, can be explicitly expressed as $t^{-(m-1)/2} p_{\mathcal{R}}(\sqrt{t})$, where $t > 0$ and $p_{\mathcal{R}}(t)$ denotes the pdf of $\mathcal{R}$ [37]. Thus, $\mathcal{R}$, or equivalently³ $\mathcal{R}^2$, fully characterises $g(\cdot)$, i.e., the type of elliptical distributions. For example, when $\mathcal{R}^2 =^d \chi_m^2$ (χ_m^2 denotes the chi-squared distribution of dimension m), then in (2.17), $g(t) \propto \exp(-t/2)$, which formulates the multivariate Gaussian distribution. Therefore, the elliptical distribution can be fully characterised by $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\mathcal{R}$. For simplicity, the elliptical distribution in (2.17) will be denoted by $\mathcal{X} \sim \mathcal{E}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{R})$ or $\mathcal{X} \sim \mathcal{E}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$, where $\mathcal{E}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, g) = c_m \det(\boldsymbol{\Sigma})^{-1/2} g(t)$ of (2.17).

Furthermore, when estimating the elliptical distribution via MLL, given observed signals $\{\mathbf{x}_j\}_{j=1}^n$, we obtain the following log-likelihood

$$L = \sum_{j=1}^n \log c_m - \frac{1}{2} \log \det(\boldsymbol{\Sigma}) + \log g(t), \quad (2.18)$$

where the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given by

$$\boldsymbol{\mu} = \frac{\sum_{j=1}^n \psi(t_j) \mathbf{x}_j}{\sum_{j=1}^n \psi(t_j)}, \quad \boldsymbol{\Sigma} = -2 \frac{\sum_{j=1}^n \psi(t_j) (\mathbf{x}_j - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})^T}{n}, \quad (2.19)$$

where $\psi(t) = g'(t)/g(t)$ and $g'(t) = dg(t)/dt$, and $t_j = (\mathbf{x}_j - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_j - \boldsymbol{\mu})$. In many elliptical distributions, $\psi(t)$ shows a decreasing trend, i.e., $\psi(t) \rightarrow 0$ when $t \rightarrow \infty$, which effectively penalises the impact of outliers. On the other hand, we should not ignore the fact that because t also includes the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that are to be estimated, the solution of (2.19) is not in the closed-form format. A method called iteratively reweighing algorithm (IRA) has been proposed to iterate (2.19) until convergence [38]. However, the IRA is not guaranteed to converge for all elliptical distributions [38].

We next aim to demystify the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by its

³The term $\mathcal{R}^2$ is frequently used in practice because $\mathcal{R}^2 =^d (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$.

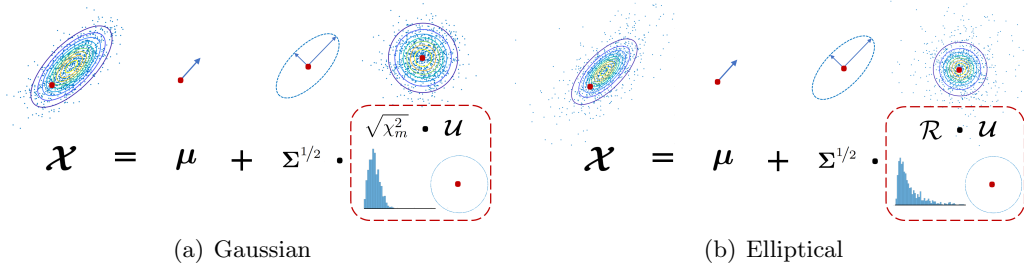


Figure 2.2: Illustration of the generating process of Gaussian and elliptical distributions.

stochastic representation as [37]

$$\mathcal{X} =^d \mu + \Sigma^{1/2} \sqrt{\chi_m^2} \cdot \mathcal{S}. \quad (2.20)$$

With this generating process, we can easily see that the term $(\mathcal{X} - \mu)^T \Sigma^{-1} (\mathcal{X} - \mu)$, also called the Mahalanobis distance, satisfies the χ_m^2 distribution when $\mathcal{X}$ is Gaussian distributed, because of

$$((\mathcal{X} - \mu)^T \Sigma^{-1/2}) (\Sigma^{-1/2} (\mathcal{X} - \mu)) =^d \sqrt{\chi_m^2} \mathcal{U}^T \mathcal{U} \sqrt{\chi_m^2} =^d \sqrt{\chi_m^2} \cdot 1 \cdot \sqrt{\chi_m^2} =^d \chi_m^2. \quad (2.21)$$

Note that regardless of μ and Σ , the Mahalanobis distance always follows the same χ_m^2 distribution; this property has been widely applied in hypothesis testing for Gaussianity, anomaly/outlier detection, etc.

We further provide illustrations when generating the Gaussian and the elliptical distributions in Figure 2.2. It is obvious that the elliptical distribution possesses more flexible choices on $\mathcal{R}$, while a widely spread $\mathcal{R}$ can lead to heavy-tailed distributions, and the corresponding robust models.

Remark 2.1. Before proceeding further, we shall emphasise the importance of the stochastic representation in analysing elliptical distributions:

1) Since $\mathcal{R}$ is independent of $\mathcal{S}$, the Mahalanobis distance $(\mathcal{X} - \mu)^T \Sigma^{-1} (\mathcal{X} - \mu)$ ($=^d \mathcal{R}^2$) is also independent of the normalised random variable $\Sigma^{-1/2} (\mathcal{X} - \mu) / \sqrt{(\mathcal{X} - \mu)^T \Sigma^{-1} (\mathcal{X} - \mu)}$ ($=^d \mathcal{S}$), an important property in many proofs in this thesis.

2) The stochastic representation provides an extremely simple way to generate sam-

ples, because only two random variables, i.e., the one-dimensional $\mathcal{R}$ and uniform $\mathcal{S}$, can be easily generated.

3) When $\mathcal{R}$ is composed from a scale mixture of normal distributions, the IRA method converges [39]. For a general elliptical distribution, however, the convergence is not ensured.

Elliptical mixture model (EMM)

Given the power of elliptical distributions, it is natural to establish the mixtures of elliptical distributions, i.e., EMMs, as a potential generative model. The generating process of EMMs is similar to that of GMMs, where $\mathbf{x}_i$ in (2.12) now satisfies a single elliptical distribution. In order to successfully establish EMMs, we first need to answer the three main questions:

- Are EMMs identifiable so that one EMM would not be modelled by two different sets of parameters?
- Are EMMs computable so that we can estimate it via numerical methods?
- Do EMMs converge with existing solvers?

Existing works that have answered these three questions are limited to several specific types of distributions. For example, the mixtures of the t -distributions have been thoroughly studied [40–42], on the basis of the IRA method. A more general mixture model has been proposed in [43] based on the Pearson type VII distribution (includes the t -distribution as a special case). Moreover, as the transformed coefficients in the wavelet domain tend to be Laplace distributed, a mixture of Laplace distributions has been proposed in [44] for image denoising. Its more general version, a mixture of hyperbolic distributions, has also been recently introduced in [45]. The above distributions belong to the *scale mixture of normals* class, which ensures the convergence of the IRA. Another recent work proposes a Fisher-Gaussian distribution as mixing components to better accommodate the curvature of data, with the

Markov chain Monte Carlo used to solve a Bayesian model [46]. This distribution has a closed-form representation and mainly consists of a von Mises-Fisher distribution convolved with Gaussian noise, which belongs to generalised skew normal distributions [47]. More importantly, when the concentration parameter $\tau \rightarrow 0$, this distribution then belongs to the mixture of symmetric elliptical distributions.

On the other hand, Wiesel proved the convergence of the IRA in [48] via the concept of geodesic convexity of Riemannian manifold, and Zhang *et al.* further relaxed the convergence conditions in [49]. The work in [50] proves similar results from another perspective of the Riemannian manifold, which also states that the IRA cannot ensure a universal convergence for all EMMs. In other words, for other elliptical distributions, the convergence is no longer guaranteed. Despite several attempts, current EMMs, including [51–53], are rather of an *ad hoc* nature.

Besides the IRA method for solving several EMMs, gradient-based numerical algorithms typically rest upon additional techniques that only work in particular situations (e.g., gradient reduction [54], re-parametrisation [55] and Cholesky decomposition [56, 57]). Recently, Hosseini and Sra directly adopted a Riemannian manifold method for estimating the GMM, which provides an alternative to the traditional EM algorithm in the GMM problem [58, 59]. However, their method fails to retain the optimum in the EMM problem. To this end, as shall be introduced in Chapter 4, we propose a universal scheme to consistently and stably achieve the optimum at a fast convergence speed, which acts as a “necessity” instead of an “alternative” in the EMM problem, as the IRA algorithm may not converge.

2.1.3 Existing Generalised Elliptical Distributions

The above introduced elliptical distributions are symmetric, and thus inevitably limited to asymmetric data. The skewness can be achieved by adding a weighting term $\pi(\mathbf{x} - \boldsymbol{\mu})$ to the existing symmetric pdfs [47, 60, 61], in a way similar to the skewed normal distribution [62]. However, this type of skewness does not necessarily start from a stochastic representation of (2.16), which impedes clear interpretations

of its inner relationships, generations and moments. A further successful variant employs conditional distributions of a symmetric elliptical distribution [63], to give

$$\mathbf{x}_{se} =^d \mathbf{y} \mid \mathcal{Y}_0 > 0, \text{ where } \begin{bmatrix} \mathbf{y} \\ \mathcal{Y}_0 \end{bmatrix} \sim \mathcal{E} \left(\begin{bmatrix} \boldsymbol{\mu} \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\beta}^T \\ \boldsymbol{\beta} & 1 \end{bmatrix}, g \right) \quad (2.22)$$

where the parameter $\boldsymbol{\beta}$ controls the skewness of the distribution. The form of (2.22) represents a typical skewed elliptical distribution, of which $\mathcal{Y}_0$ has also been extended to higher dimensions [64–66]. Importantly, the form of (2.22) is invariant under quadratic forms [47], and is closed under marginalisation and affine transforms; we refer to [67] for more detail. However, the estimation of the above skewed elliptical distributions can be ill-posed, especially regarding its shape (skewness) parameter. Although the singularity issue in the information matrix of shape parameter can be relieved by a centralised parametrisation trick [60], the estimation of the shape parameter in this skewed version could still diverge, which calls for other penalty techniques [68]. Note that the moment estimation method employed to estimate skewed normal distributions is also inadequate for skewed elliptical distributions [69].

A further generalisation that explicitly comes with the stochastic representations was proposed by Frahm [37], and is given by

$$\mathbf{x}_{ge} =^d \boldsymbol{\mu} + \hat{\mathcal{R}}\boldsymbol{\Lambda}\boldsymbol{u}, \quad (2.23)$$

with a form similar to that in (2.16). The difference lies in the scalar random variable, $\hat{\mathcal{R}}$, that no longer requires to be non-negative and can be even negative; although a uniform distribution on a unit sphere similar to $\boldsymbol{S}, \boldsymbol{u}$ is dependent with $\hat{\mathcal{R}}$, which skews the distribution. This generalisation includes the skewed elliptical distribution in (2.22) as a special case, and is closed under affine transformation, marginalisations and even conditioning [37].

The skewness of (2.23) arises from the dependency between $\hat{\mathcal{R}}$ and $\boldsymbol{u}$. In other words, although $\boldsymbol{u}$ is still distributed on a unit sphere surface, its dependency on

$\hat{\mathcal{R}}$ enables varying importance along directions. Given the fact that there are only a few distributions of a scalar random variable conditioned on a vector, this flexible dependency is usually abstract and under-determined in practical modelling. More importantly, despite generalising and skewing the elliptical distribution, this dependency also impedes the analysis of the generalised elliptical distribution, because the roles of $\hat{\mathcal{R}}$, $\mathbf{\Lambda}$ and $\mathbf{U}$ become unclear and sometimes interchangeable. For example, in the elliptical distribution, the dispersion is uniquely dominated by $\mathbf{\Sigma}$, while this no longer holds in generalised elliptical distributions [37], and could lead to multiple minima/maxima when modelling data in practice. We therefore propose a new generalisation towards symmetric elliptical distributions by addressing all of the above problems in Chapter 3.

2.2 Implicit Generative Models

Implicit generative models have intensively received attentions due to the success of deep neural nets, with many important structures proposed to model data and signal distributions. Basically, implicit generative models assume a deep neural net as $\mathcal{G}$ to transform from some simple distributions $\mathbf{Z}$, for example a spherical distribution, by $\mathcal{R}\mathbf{S}$ or $\sqrt{\chi_m^2}\mathbf{S}$, and then achieve a powerful implicit distribution by $\mathbf{X} =^d \mathcal{G}\{\mathbf{Z}\}$. In this way, a deep neural net can be trained so as the generated distribution approaches the true distribution of observed signals. Deep neural nets, obviously, are capable of modelling complicated distributions in real-world signals such as images/videos, natural language processing and bio-engineering tasks.

The variational auto-encoder (VAE) [18], probabilistic flow [70] and generative adversarial net (GAN) [19] are the best known representatives in literature, and we shall briefly introduce them as a background for the material in this thesis. Generally speaking, the VAE approaches $\mathbf{X}$ by optimising a proper posterior, i.e., by limiting suitable latent variable distributions that are most likely to generate the true distribution. The probabilistic flow based models are straightforward to use invertible transforms, so that the evaluation can be exactly inferenced. The GANs,

however, employ another neural net called a discriminator to distinguish between the generated distribution and the true distribution, so that the generator needs to compete against the discriminator to approach to the true distribution. The discriminator basically outputs a trained discrepancy, which is also called the neural distance [71].

2.2.1 Probabilistic Flows

Different from the linear models mentioned in Section 2.1.1, probabilistic flow based models employ a non-linear transform (implemented by neural nets) on the latent variable to fit the distributions of observed signals [70]. However, a non-linear transform dose not necessarily admit a closed-form pdf of the resulting distribution. To address this issue, one strategy is to formulate every transform in neural nets to be bijective, so that the generating process “flows” from the latent distribution to the signal distribution with invertible transforms. In this way, although still not in a closed-form pdf, we can have an explicit relationship between the latent and signal distributions, and are thus able to use gradient descent to update the transforms.

Specifically, when $\mathbf{Z}$ and $\mathbf{X}$ are of the same dimension, and the transform $\mathbf{X} =^d g(\mathbf{Z})$ is differential and invertible, we can establish the explicit relationship

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{Z}}(\mathbf{z}) \left| \det \left(\frac{d\mathbf{z}}{d\mathbf{x}} \right) \right| = p_{\mathbf{Z}}(g^{-1}(\mathbf{x})) |\det(\mathbf{J}_{g^{-1}}(\mathbf{x}))|, \quad (2.24)$$

where g^{-1} denotes an inverse of the function g , given by $\mathbf{Z} =^d g^{-1}(\mathbf{X})$, and $\mathbf{J}_{g^{-1}}(\mathbf{x}) \in \mathbb{R}^{m \times m}$ is the Jacobian matrix with regard to variable $\mathbf{x}$, given by

$$\mathbf{J}_{g^{-1}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1^{-1}}{\partial x_1} & \frac{\partial g_1^{-1}}{\partial x_2} & \cdots & \frac{\partial g_1^{-1}}{\partial x_d} \\ \frac{\partial g_2^{-1}}{\partial x_1} & \frac{\partial g_2^{-1}}{\partial x_2} & \cdots & \frac{\partial g_2^{-1}}{\partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_d^{-1}}{\partial x_1} & \frac{\partial g_d^{-1}}{\partial x_2} & \cdots & \frac{\partial g_d^{-1}}{\partial x_d} \end{bmatrix}. \quad (2.25)$$

More importantly, when the function g is a neural network that is composed of

consecutive layers, $g = g_1 \circ g_2 \circ g_3 \cdots \circ g_l$, the Jacobian matrix has the following favourable “separability” property,

$$\det(\mathbf{J}_{g^{-1}}(\mathbf{x})) = \det(\mathbf{J}_{g_1^{-1}}(\mathbf{x})) \cdot \det(\mathbf{J}_{g_2^{-1}}(\mathbf{x})) \cdots \det(\mathbf{J}_{g_l^{-1}}(\mathbf{x})) \quad (2.26)$$

Thus, given (2.24) and (2.26), we can use the KL divergence to maximise the log-likelihood of observed signals $\{\mathbf{x}_j\}_{j=1}^n$ as follows

$$L = \sum_{j=1}^n \log p_{\boldsymbol{\mathcal{X}}}(\mathbf{x}) = \sum_{j=1}^n \left(\log p_{\boldsymbol{\mathcal{Z}}}(g^{-1}(\mathbf{x}_j)) + \sum_{i=1}^l \log |\det(\mathbf{J}_{g_i^{-1}}(\mathbf{x}_j))| \right). \quad (2.27)$$

The calculation of the determinant Jacobians, however, typically regimes an $\mathcal{O}(m^3)$ computational complexity [70], which is intractable in high dimensions. Therefore, for practical purposes, the invertible transforms $\{g_i\}_{i=1}^l$ have to be chosen by some specific forms for the ease of computations, at a cost of at most $\mathcal{O}(m)$, whilst avoiding trivial transforms to increase the model expressive power. We refer the reader to some typical works in this direction such as NICE [72], RealNVP [73] and GLOW [74], as well as a well-documented review [70].

The probabilistic flows provide an elegant way of achieving exact inference whilst possessing an adequate and powerful expressibility. However, in order to obtain invertible transforms, the dimension of latent variables needs to be the same as the dimension of signals, that is, extremely high, and each layer in the network needs to keep the same dimension. More importantly, only limited types of transforms cater for the invertible requirement and computational feasibility which leads to excessive stacking of hidden layers to maintain the expressive power. Therefore, probabilistic flows still suffer from high computational burdens, as well as large size of network designs.

2.2.2 Variational Auto-Encoder (VAE)

It is important to have reduced dimensions in the latent space, because although signals such as images/videos are of high dimensions, their semantic representations

always lie on some low-dimensional manifolds. To this end, the VAE [18] employs a non-linear transform of a reduced dimension implemented by neural nets to fit the distributions of observed signals, in this way, extracting semantically meaningful information in the latent space. The cost to pay is that the transform is no longer invertible, so that the resulting distribution $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ cannot be obtained in a closed-form pdf, and the relationship is not explicit between the generated and signal distributions either. A remedy is to generate plenty of samples from the latent distribution $\mathcal{Z}$, as⁴ $\{\mathbf{z}_i\}_{i=1}^{n'}$, and obtain a proxy to the generated distribution by its samples $p(\mathbf{x}|\mathbf{z}_i)$. It is, however, a challenging to compare the generated samples and the observed signals, both of which are in the format of random samples instead of distributions.

Furthermore, VAEs basically use the KL divergence to compare the generated distribution and the true distribution, which results in the MLL method. As shall be mentioned in Section 2.3, the KL divergence has its intrinsic limitations when comparing two well-separated distributions, and is thus even worse when comparing the random samples from two distributions. Another way of understanding this is from the beginning of training, where the generated distribution is basically random noise which makes it intractable for the generated samples to exactly “hit” the true distribution because they both reside in high dimensions and exhibit complicated structures. Thus, most of likelihoods of the observed signals to be maximised, $p(\mathbf{x}|\mathbf{z}_i)$, are zero and therefore cannot provide any useful information or gradients to train the network, which severely degrades the generation process.

It is thus natural to ask: *can we “help” the generation by specifying a proper latent variable distribution that is most likely to generate the true distribution?* The posterior $p(\mathbf{z}|\mathbf{x})$ is the best choice to formulate this latent variable distribution. However, due to the intractability of $p(\mathbf{x})$, we cannot achieve explicit posterior as the Bayesian theorem gives $p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$. The VAE, therefore, as its name implied, employs variational inference to calculate the posterior.

⁴Please note that the number of generated samples n' is not necessarily equal to the number of observed signals n .

Because we cannot achieve an explicit posterior, $p(\mathbf{z}|\mathbf{x})$, variational inference basically employs another computable distribution $q(\mathbf{z}|\mathbf{x})$ and attempts to minimise the KL divergence between $p(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{x})$ as a way of approximating this intractable posterior. Therefore, we have

$$\begin{aligned}
d_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) &= \int q(\mathbf{z}|\mathbf{x}) \ln \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z} = - \int q(\mathbf{z}|\mathbf{x}) \ln \frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= - \int q(\mathbf{z}|\mathbf{x}) \ln \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{z}|\mathbf{x})p(\mathbf{x})} d\mathbf{z} \\
&= - \underbrace{\int q(\mathbf{z}|\mathbf{x}) \ln \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z}}_{\text{ELBO}} + \ln p(\mathbf{x}),
\end{aligned} \tag{2.28}$$

where the first terms is called *evidence lower bound* (ELBO). It can be further expanded as

$$\begin{aligned}
\text{ELBO} &= \int q(\mathbf{z}|\mathbf{x}) \ln \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q(\mathbf{z}|\mathbf{x}) \ln p(\mathbf{x}|\mathbf{z}) d\mathbf{z} + \int q(\mathbf{z}|\mathbf{x}) \ln p(\mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}|\mathbf{x}) \ln q(\mathbf{x}|\mathbf{z}) d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\ln p(\mathbf{x}|\mathbf{z})] - d_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))
\end{aligned} \tag{2.29}$$

Therefore, upon combining (2.28) and (2.29), we arrive at the following rearranged equation

$$\ln p(\mathbf{x}) - d_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) = \text{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\ln p(\mathbf{x}|\mathbf{z})] - d_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{2.30}$$

By inspecting (2.30), we observe that if we are maximising the ELBO, although we cannot obtain an explicit $p(\mathbf{x})$ needed by the MLL, we are basically maximising $\ln p(\mathbf{x})$, i.e., the log-likelihood, as well as simultaneously minimising the approximation error on the posteriors $d_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$. This is precisely what is desired in the above elaboration. Furthermore, the ELBO in (2.30) can be maximised through an auto-encoder structure, where $q(\mathbf{z}|\mathbf{x})$ is an encoder and $p(\mathbf{x}|\mathbf{z})$ decodes from the latent variable. More importantly, as for implementation considerations, we can assume both $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$ to be Gaussian distributions, whereby their mean

values and the (diagonal) covariance matrix are the output from neural networks; the first term of (2.30) therefore becomes the widely used MSE reconstruction loss, whereas the second term of (2.30) can be obtained explicitly given two Gaussian distributions [18]. Furthermore, since sampling from $q(\mathbf{z}|\mathbf{x})$ cannot ensure an efficient gradient update, a re-parametrisation trick has been applied to enable the back-propagation by sampling from the standard Gaussian distribution [18].

It is the fact that the variational inference is a powerful structure for solving Bayesian problems, and if we revisit the GMM problem in Section 2.1.1, we can still regard it as a variational inference problem where, different from the VAE, the posterior in the GMM $p(\mathbf{x}|\mathbf{z})$ can be explicitly obtained. We can thus rewrite (2.29) as

$$\begin{aligned} \text{ELBO} &= \int q(\mathbf{z}|\mathbf{x}) \ln \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= \int q(\mathbf{z}|\mathbf{x}) \ln p(\mathbf{x}|\mathbf{z}) d\mathbf{z} + \int q(\mathbf{z}|\mathbf{x}) \ln p(\mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}|\mathbf{x}) \ln q(\mathbf{x}|\mathbf{z}) d\mathbf{z} \quad (2.31) \\ &= \int q(\mathbf{z}|\mathbf{x}) \ln p(\mathbf{x}, \mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}|\mathbf{x}) \ln q(\mathbf{x}|\mathbf{z}) d\mathbf{z}. \end{aligned}$$

Given that the posterior of the GMM problem can be explicitly parametrised by $\boldsymbol{\theta} = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \pi_i\}_{i=1}^k$, for the current iteration during estimation, we can choose the variational posterior $q(\mathbf{z}|\mathbf{x})$ to be equal to $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}^{\text{old}})$, which is the E step in (2.14). Then, we can maximise the ELBO by maximising the first term $\int p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z}$ (whilst ignoring the second term $\int p(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{x}|\mathbf{z}; \boldsymbol{\theta}^{\text{old}}) d\mathbf{z}$ due to its irrelevance to the optimisation), which results into the M step in (2.14). In this way, variational inference provides an extremely powerful tool under the KL divergence, whereby the VAE elegantly builds upon the variational inference and achieves the generation on complicated signals such as images.

However, in practice, the VAE typically generates blurred images, and a common conjecture is that under the KL divergence, the VAE tends to diffuse the generated distribution across the whole data space, instead of focusing on specific parts of the data space [75]. This results in the MSE loss which is too conservative to generate sharp images. It is also the fact that naive auto-encoders also use MSE losses

and provide reconstructed blurred images. Also, VAEs perform poorly when the assumed Gaussian distributions of posterior, prior and generating distributions are insufficiently flexible [76].

2.2.3 Generative Adversarial Net (GAN)

Different from approximating the posterior in the VAE, the GAN [19] proposes a discriminator to distinguish between the generated and true data distributions. In this way, research focus now turns to developing efficient statistical metrics in generative models, as by using the deep neural nets, the resulting distributions do not necessarily have closed-form pdfs and thus need more powerful metrics for evaluations.

More specifically, the GAN consists of two neural net modules, that is, a generator and a discriminator. The goal of the generator, as usual, is to generate distributions that are close to the true data distribution, whereas the discriminator aims to distinguish between the generated distribution and the true distribution. These compete against each other and if properly trained, they arrive at a Nash equilibrium [19]. The overall objective function is formulated as follows,

$$\min_g \max_f L(g, f) = \mathbb{E}_{p_{\mathcal{X}}(\mathbf{x})}[\log f(\mathbf{x})] + \mathbb{E}_{p_{\mathcal{Z}}(\mathbf{z})}[\log (1 - f(g(\mathbf{z})))] \quad (2.32)$$

where g is the output of the generator, f is the output of the discriminator of range $[0, 1]$, and $\mathcal{X}$ denotes the true data distribution, whereas $\mathcal{Z}$ is the latent distribution. In (2.32), the maximal f^* is obtained when it accurately recognises the generated distribution by outputting $f(g(\mathbf{z})) = 0$, whilst it outputs $f(\mathbf{x}) = 1$ for the true distribution. Its optima are given by $f^*(\mathbf{x}) = \frac{p_{\mathcal{X}}(\mathbf{x})}{p_{\mathcal{X}}(\mathbf{x}) + p_{g(\mathcal{Z})}(\mathbf{x})}$ [19]. On the other hand, given a maximal f^* , g^* only reaches its minimum when it has exactly the same distribution as the true data distribution, i.e., $g(\mathcal{Z}) = \mathcal{X}$, which yields the global optima and also named as the Nash equilibrium in this zero-sum game. In this case, $L(g, f) = -\log 4$, and $f^*(\mathbf{x}) = 1/2$ [19], meaning that the discriminator cannot

distinguish between the generated and true distributions. Furthermore, due to the cross-entropy form of (2.32), it has been also proved that given an optimal f^* , the $L(g, f^*)$ is equivalent to comparing the JS divergence between the generated and true distributions, as follows [19]

$$\min_g L(g, f^*) = -\log 4 + 2 \cdot d_{\text{JS}}(\mathcal{X}, g(\mathcal{Z})), \quad (2.33)$$

The training of GANs iterates the optimisation on g and f until the convergence. However, the original format of GANs [19] requires extremely careful designs on both net modules and training steps, so that the generator and discriminator can play against each other properly. In practice, the GAN exhibits strong instability in training and suffers from mode collapse, which has been noticed and discussed in the literature [77]. A main conjecture regarding this issue stems from the fact that the true distribution basically resides in a low-dimensional semantic manifold where, under the JS divergence, a perfect discriminator does exist and therefore cannot provide any back-propagating gradient to train the generator [78].

It is the fact that existing statistical metrics are typically inferior in high dimensional spaces and for complicated distributions when directly used, and the key obstacle is a lack of an efficient and comprehensive metric to evaluate the discrepancy between the generated distribution and the true distribution. Main trends in stabilising and improving GANs tend to employ the IPMs to measure the generated and true distributions, with the notable ones using a dual form of Wasserstein distance to formulate the Wasserstein GAN (W-GAN) [4]. In the W-GAN, the discriminator, also called the *critic* in IPM-GANs, now formulates a part of the Wasserstein distance to ensure theoretical completeness, but needs to satisfy the requirement of 1-Lipschitz requirement [4].

A naive weight clipping scheme in the W-GAN [4] employs a gradient penalty in W-GAN (W-GAN-GP) to mitigate the heavily constrained *critic* by penalising the gradient norm [79], followed by a further elegant treatment by restricting the largest

singular value of the net weights [80]. It has been understood that although the *critic* cannot search within all functions satisfying the Lipschitz requirement [71, 81], the *critic* still serves as a way of transforming high dimensional but insufficiently supported data distributions into low dimensional yet broadly supported (simple) distributions in the embedded domain [3]. Comparing the embedded statistics, however, then becomes much easier. For example, Cramer GAN compares the mean with an advanced $\mathcal{F}$ from the Cramer distance to correct the biased gradient [82], whilst McGAN [83] explicitly compares the mean and the covariance in the embedded domain. Fisher GAN employs a scale-free Mahalanobis distance and thus a data dependent $\mathcal{F}$ [84], which is basically the Fisher-Rao distance in the embedded domain between two Gaussian distributions with the same covariance. The recent Sphere GAN further compares higher-order moments up to a specified order, and avoids the requirement of Lipschitz continuity condition by projecting onto a spherical surface [8]. Moreover, in a non-parametric way, BE-GAN directly employs an auto-encoder as the *critic*, whereby the auto-encoder loss is compared through embedded distributions [85]. The sliced Wasserstein distance has also been utilised to measure the discrepancy in the embedded domain [86]. Another non-parametric metric was obtained by the kernel trick of the MMD-GAN [3, 5], which treats $\mathcal{F}$ as the RKHS.

On the other hand, there are also works trying to combine auto-encoders in an adversarial way. To address the smoothing artefact of the variational auto-encoder [18], several works aim to incorporate the adversarial style in (variational) auto-encoders, in the hope of gaining clearer images whilst maintaining the ability of reconstruction. These mostly comprise at least three modules, an encoder, a decoder, and an adversarial module [87–92]. To the best of our knowledge, there is one exception, called the adversarial generator encoder (AGE) [7], which incorporates two modules in adversarially training an auto-encoder under a *max-min* problem. The AGE still assumes the Gaussianity in the embedded distributions and only compares the mean and the diagonal covariance matrix; this is however insufficient

to identify two distributions, and additionally requires the pixel domain loss to be utilised in implementations. As shall be introduced in Chapter 6, we propose a novel framework called RCF-GAN, playing a *min-max* problem, which is fundamentally different from the AGE, as the auto-encoder in our RCF-GAN is a necessity to achieve the theoretical guarantee of a reciprocal, with the proposed anchor design. In contrast, without the auto-encoder, the AGE could still work (see Theorems 1 and 2 in [7]). Furthermore, other than the first- and second-order moments, the proposed RCF-GAN fully compares the discrepancies between distributions in the embedded domain via CFs. Benefiting from the powerful non-parametric metric via the CFs, our RCF-GAN adversarially learns distributions only in the embedded domain, that is, on a semantically meaningful manifold, without the need of any operation on data domain.

2.3 Preliminaries on Statistical Metrics

2.3.1 Kullback–Leibler (KL) Divergence and Extensions

The KL divergence measures the difference between two distributions, $p_{\mathcal{X}}(\mathbf{x})$ and $p_{\mathcal{Y}}(\mathbf{x})$, as

$$d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) = \int p_{\mathcal{X}}(\mathbf{x}) \log \left(\frac{p_{\mathcal{X}}(\mathbf{x})}{p_{\mathcal{Y}}(\mathbf{x})} \right) d\mathbf{x}. \quad (2.34)$$

We also use interchangeably the notation $d_{\text{KL}}(p_{\mathcal{X}}(\mathbf{x}) || p_{\mathcal{Y}}(\mathbf{x}))$. Generally speaking, the KL divergence satisfies most of properties that a distance requires, except for the symmetry property ($d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) \neq d_{\text{KL}}(\mathcal{Y}, \mathcal{X})$). In other words, the KL divergence uniquely associates two distributions so that $\mathcal{X} =^d \mathcal{Y}$ if and only if $d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) = 0$ or $d_{\text{KL}}(\mathcal{Y}, \mathcal{X}) = 0$. Moreover, the KL divergence also satisfies the triangle property, i.e., $d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) \leq d_{\text{KL}}(\mathcal{X}, \mathcal{Z}) + d_{\text{KL}}(\mathcal{Z}, \mathcal{Y})$, which is also called convexity.

Taking two multivariate Gaussian distributions, $\mathcal{X} \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$,

as an instance, their KL divergence can be explicitly obtained as [93]

$$d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{2} \left(\log \frac{|\Sigma_2|}{|\Sigma_1|} + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma_2^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \text{tr}(\Sigma_2^{-1} \Sigma_1) - m \right) \quad (2.35)$$

Furthermore, in many applications we have observed signals $\{\mathbf{x}_j\}_{j=1}^n$ that are assumed to be sampled from $p_{\mathcal{X}}(\mathbf{x})$ and we then wish to optimise $p_{\mathcal{Y}}(\mathbf{x})$ that is parametrised by $\boldsymbol{\theta}$, i.e., $p_{\mathcal{Y}}(\mathbf{x}; \boldsymbol{\theta})$. The minimisation of the KL divergence then yields

$$\begin{aligned} \min_{\boldsymbol{\theta}} d_{\text{KL}}(\mathcal{X}, \mathcal{Y}) &= \int p_{\mathcal{X}}(\mathbf{x}) \log p_{\mathcal{X}}(\mathbf{x}) d\mathbf{x} - \int p_{\mathcal{X}}(\mathbf{x}) \log p_{\mathcal{Y}}(\mathbf{x}; \boldsymbol{\theta}) d\mathbf{x} \\ &\propto - \int p_{\mathcal{X}}(\mathbf{x}) \log p_{\mathcal{Y}}(\mathbf{x}; \boldsymbol{\theta}) d\mathbf{x} = - \sum_{j=1}^n \log p_{\mathcal{Y}}(\mathbf{x}_j; \boldsymbol{\theta}), \end{aligned} \quad (2.36)$$

which is exactly the MLL method, i.e., based on maximising the likelihood of the distribution $p_{\mathcal{Y}}(\mathbf{x}; \boldsymbol{\theta})$ to be optimised, so that $p_{\mathcal{Y}}(\mathbf{x}; \boldsymbol{\theta})$ is highly likely to generate the samples from $\mathcal{X}$.

Although MLL is appreciated for its ease in practice, the KL divergence suffers from its “bin-to-bin” way of comparing distributions. In other words, for every bin of $\mathbf{x}_j$ - $\mathbf{x}_{j+1}$, it compares the two distributions “vertically” by $\int_{\mathbf{x}_j}^{\mathbf{x}_{j+1}} p_{\mathcal{X}}(\mathbf{x}) \log (p_{\mathcal{X}}(\mathbf{x})/p_{\mathcal{Y}}(\mathbf{x})) d\mathbf{x}$, which is insufficient because this may ignore the significant difference between two disjoint distributions. As shown in Figure 2.3, when the two distributions are well separated, the KL divergence outputs consistently large values, which leads to the so called gradient vanishing.

A symmetric (and smooth) version of the KL divergence is the Jensen-Shannon (JS) divergence, which is defined as

$$d_{\text{JS}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{2} d_{\text{KL}}(\mathcal{X}, \mathcal{Z}) + \frac{1}{2} d_{\text{KL}}(\mathcal{Y}, \mathcal{Z}), \quad (2.37)$$

where $p_{\mathcal{Z}}(\mathbf{x}) = p_{\mathcal{X}}(\mathbf{x}) + p_{\mathcal{Y}}(\mathbf{x})/2$. The JS divergence also indicates the mutual entropy of $\mathcal{X}$ and $\mathcal{Y}$, switched by a binary indicator $\mathcal{Z}$. This scenario is similar to the original

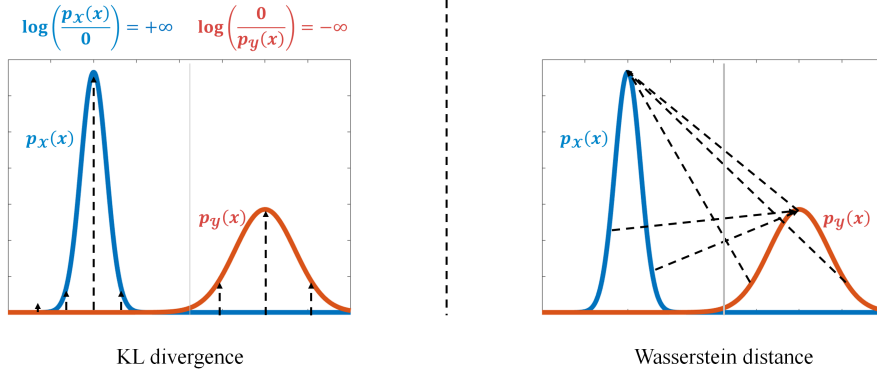


Figure 2.3: Illustration of the comparison principles of the KL divergence and Wasserstein distance.

GAN [19], where the JS divergence has also been reflected in the objective function. Further generalisations include the f-divergence [94] and the Bregman divergence [95], with the KL divergence and JS divergence as special cases. Although finding applications in few generative models [96], these are more abstract and meaningful in theoretical analysis, and thus be less popular compared with the KL divergence.

2.3.2 Wasserstein Distance

The popularity of Wasserstein distance arises from its ability to provide a powerful measurement between two distributions. The Wasserstein distance originates from Gaspard Monge [22], whereas its popular representation comes from an independent work of Kantorovich [97], which has the form,

$$d_W^\alpha(\mathcal{X}, \mathcal{Y}) = \inf_{\tau \in \Pi(\mathcal{X}, \mathcal{Y})} \mathbb{E}_\tau[|\mathbf{x} - \mathbf{y}|^\alpha], \quad \mathbf{x} \sim p_{\mathcal{X}}(\mathbf{x}), \mathbf{y} \sim p_{\mathcal{Y}}(\mathbf{y}), \quad (2.38)$$

where $\Pi(\mathcal{X}, \mathcal{Y})$ denotes the collection of joint distributions of $\mathcal{X}$ and $\mathcal{Y}$. The Wasserstein distance was initially used to solve the problem of optimal transport, which aims to find the minimal transport cost between two piles. When $\alpha = 1$, this is also called the Earth mover's distance (EMD). Moreover, $\Pi(\mathcal{X}, \mathcal{Y})$ is called accordingly the transport plans. By definition, the Wasserstein distance compares two distributions by finding the minimal transport plan within all possible transport plans, thus conducting a “cross-bin” comparison as shown in Figure 2.3.

Benefiting from its “cross-bin” comparison, the Wasserstein distance indicates the difference between two distributions better than the KL divergence, and has therefore been widely employed in evaluating complicated distributions generated by, for example, deep generative models. However, the problem of the Wasserstein distance is its computational difficulty, because we still have to find the minimum and basically do not have an explicit representation except for $m = 1$ [98], posing difficulties in further optimisations.

Many algorithms have been proposed to efficiently calculate the Wasserstein distance, where we refer to [98] for more detail. However, we need to point out to the dual form of the Wasserstein distance, which is the core of the Wasserstein related GANs. We consider a special case for $\alpha = 1$ because of its simplicity and popularity, and refer to Chapter 5 of [97] for other cases. Specifically, the term $d_W(\mathcal{X}, \mathcal{Y})$ has the dual form given by,

$$d_W^\alpha(\mathcal{X}, \mathcal{Y}) = \sup_{f \in \mathcal{F}_L} \mathbb{E}_{\mathcal{X}}[f(\mathbf{x})] - \mathbb{E}_{\mathcal{Y}}[f(\mathbf{y})], \quad (2.39)$$

where $\mathcal{F}_L$ represents the collection of all 1-Lipschitz continuous functions. This allows for the Wasserstein distance to be solved by a neural net that performs the operation of finding f , which is the core of Wasserstein GANs [4, 78], as well as providing advanced theoretical guarantees for the convergence of GANs.

2.3.3 Integral Probability Metrics (IPMs)

The dual form of the Wasserstein distance in (2.39) basically provides another family of distance metrics called the integral probability metrics (IPMs) [99], which all have the following representation

$$d_{\text{IPM}} = \sup_{f \in \mathcal{F}} |\mathbb{E}_{\mathcal{X}}[f(\mathbf{x})] - \mathbb{E}_{\mathcal{Y}}[f(\mathbf{y})]|, \quad (2.40)$$

where here the symbol $\mathcal{F}$ denotes a collection of bounded real functions. Different $\mathcal{F}$ basically define different IPMs [100], for example, when $\mathcal{F}$ is the collection $\mathcal{F}_L$, then it formulates the Wasserstein distance (dual form). Another important IPM is called the maximum mean discrepancy (MMD) [101], where $\mathcal{F}$ is considered to be the reproducing kernel Hilbert space (RKHS) [102]. Using kernel embeddings, the MMD distance can be efficiently calculated by the explicit relationship given by

$$d_{\text{MMD}}^2 = \mathbb{E}_{\mathbf{x}}[\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}] + \mathbb{E}_{\mathbf{y}}[\langle \phi(\mathbf{y}), \phi(\mathbf{y}') \rangle_{\mathcal{H}}] - 2\mathbb{E}_{\mathbf{x}, \mathbf{y}}[\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}], \quad (2.41)$$

where $\phi(\mathbf{x})$ is the eigenfunction and allows for $f(\mathbf{x}) = \langle f, \phi(\mathbf{x}) \rangle_{\mathcal{H}}$, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in the RKHS. Given samples from $\mathcal{X}$ and $\mathcal{Y}$, i.e., $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{y}_i\}_{i=1}^{n'}$, the (empirical) MMD can be obtained using the kernel trick as follows,

$$d_{\text{MMD}}^2 = \frac{1}{n(n-1)} \sum_{i \neq j}^n k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n'(n'-1)} \sum_{i \neq j}^{n'} k(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{nn'} \sum_{i,j}^{n,n'} k(\mathbf{x}_i, \mathbf{y}_j). \quad (2.42)$$

2.4 Preliminaries on Statistical Manifolds

A Riemannian manifold $(\mathcal{M}, \rho)$ is a smooth (differential) manifold $\mathcal{M}$ (i.e., locally homeomorphic to the Euclidean space) which is equipped with a smoothly varying inner product, ρ , on its tangent space. The inner product also defines a Riemannian metric on the tangent space, so that the length of a curve and the angle between two vectors can be correspondingly defined. Curves on the manifold with the shortest paths are called *geodesics*, which exhibit constant instantaneous speed and generalise straight lines in the Euclidean space. The distance between two points on $\mathcal{M}$ is defined as the minimum length of all geodesics connecting these two points.

We shall use the symbol $T_{\Sigma}\mathcal{M}$ to denote the *tangent space* at the point Σ , which is the first-order approximation of $\mathcal{M}$ at Σ . Consequently, the *Riemannian gradient* of a function f is defined with regard to the equivalence between its inner product with an arbitrary vector ξ on $T_{\Sigma}\mathcal{M}$ and the Fréchet derivative of f at ξ . Moreover,

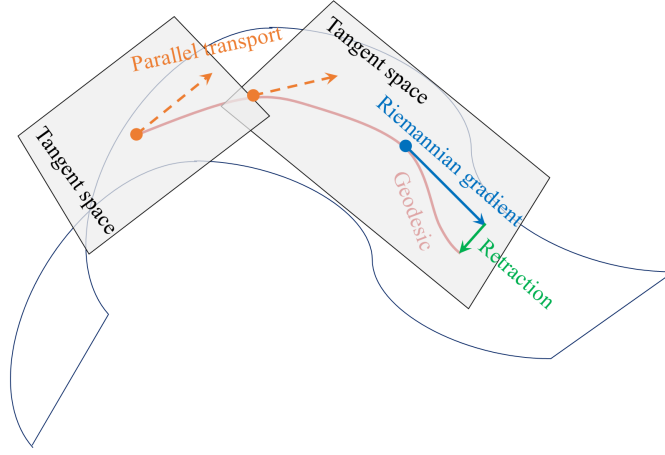


Figure 2.4: Illustrations of basic operations on a Riemannian manifold.

a smooth mapping from $T_{\Sigma}\mathcal{M}$ onto $\mathcal{M}$ is called the *retraction*, whereby an exponential mapping obtains the point on geodesics in the direction of the tangent space. Because the tangent spaces vary across different points on $\mathcal{M}$, *parallel transport* across different tangent spaces can be introduced on the basis of the Levi-Civita connection, which preserves the inner product and the norm. In this way, we can convert a complex optimisation problem on $\mathcal{M}$ into a more analysis friendly space, i.e., $T_{\Sigma}\mathcal{M}$. For a comprehensive text on the optimisation on the Riemannian manifold, we refer to [103] and illustrate some basic operations in Figure 2.4. Therefore, on the basis of the above basic operations, the manifold optimisation can be performed by the Riemannian gradient descent [104]. The retraction is then utilised to map a step descent from the tangent space to the manifold. To accelerate gradient descent optimisation, the parallel transport can also be utilised to accumulate the first-order moments [105–109].

The statistical manifold is therefore a Riemannian manifold built on probabilistic distributions, that is, each point on the manifold is a probability distribution. When the distribution is characterised by parameters, we can relate the manifold back to the parameter space. For example, when establishing a statistical manifold of multivariate Gaussian distributions that are parametrised by the covariance matrix, we can build each point of the manifold as the covariance matrix, and its Riemannian metric is related to the Hessian of the employed statistical distance. When the KL

divergence is employed, we then obtain the Fisher information tensor as the Riemannian metric. Pioneering in this direction is the work of Rao, which introduced the Rao distance to define the statistical difference between two multivariate Gaussian distributions [110]. This distance was later generalised and calculated in closed-form [111–113], to obtain an explicit metric (also called the Fisher-Rao metric). The metric $\rho_{\Sigma}(\eta, \xi) \triangleq \langle \eta, \xi \rangle \triangleq \text{tr}(\eta \Sigma^{-1} \xi \Sigma^{-1})$ was adopted by to measure the Rao distance through covariance matrices of two multivariate Gaussian distributions. It is also possible to obtain a closed-form solution for the geodesic between two positive definite matrices Σ_1 and Σ_2 , $\gamma(t) = \Sigma_1^{1/2} (\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2})^t \Sigma_1^{1/2}$, to yield its geodesic distance $d(\Sigma_1, \Sigma_2) = \|\ln(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2})\|_F$ [114]. A geodesic convex function f is defined as $f(\gamma(t)) \leq (1-t)f(\Sigma_1) + tf(\Sigma_2)$ with $t \in [0, 1]$. However, with regard to other elliptical distributions, the corresponding Fisher-Rao metric is not guaranteed to be well suited for optimisation [23].

On the other hand, there is another type of distributions, named the exponential family, that overlaps with elliptical distributions; its Fisher-Rao metric can be explicitly determined by a second-order derivative of the potential function [115]. However, the corresponding Riemannian gradient, Levi-Cevita connection, exponential mapping, parallel transport, etc., may not necessarily be obtained explicitly and in a general form for multivariate exponential families [112]. The existing literature mainly analyses the Gaussian distribution [116] and the dually-flat affine geometry [117] in terms of α -connections. More importantly, even though the optimisation can be formulated, a further obstacle is the lack of re-parametrisation property. As shown in the sequel, the absence of re-parametrisation could lead to extremely slow convergence when solving EMMs.

Chapter 3

Generalised Elliptical Distributions

A large class of modern probabilistic learning systems assumes symmetric probability distributions, however, real-world data tend to obey skewed distributions and are thus not always adequately modelled through symmetric distributions. To address this issue, elliptical distributions are increasingly used to generalise symmetric distributions, and further improvements to skewed elliptical distributions have recently attracted much attention. However, existing approaches are either hard to estimate or have complicated and abstract representations. To this end, we propose to employ the von-Mises-Fisher (vMF) distribution to obtain an explicit and simple probability representation of the skewed elliptical distribution. This is shown not only to allow for dealing with non-symmetric learning systems, but also to provide a physically meaningful way of generalising skewed distributions. For rigour, our extension is proved to share important and desirable properties with its symmetric counterpart. We also demonstrate both theoretically and through examples that the proposed vMF distribution is both easy to generate and stable to estimate.

3.1 Introduction

Probabilistic distributions are a common underpinning tool in the modelling, understanding and predicting a wide variety of real-world signals. The normal distribution has been a workhorse in probabilistic modelling, as it admits a simple representation and mathematical tractability, while its application is justified through the central limit theorem. However, issues such as the lack of robustness and flexibility when dealing with general signals remain a serious obstacle to real-world applications. The family of elliptical distributions generalises normal distributions, and possesses many desired properties such as simple generation, controllable robustness and flexibility. Elliptical distributions include the normal, Cauchy, t , logistic, and Weibull distributions [16]. The well-behaved nature of elliptical distributions underpins powerful modelling tools, such as unimodal [118, 119], mixture models [120, 121], Bayesian frameworks [122] and probabilistic graphical models [123].

Despite success, elliptical distributions inherit some of the limitations of symmetric distributions, which affects their modelling power, as in many cases such as financial, biometric and audio scenarios, the data are not symmetric due to an intrinsic coupling, systematic trend, outliers, or a small number of samples available. To address this issue, several skewed versions of the elliptical distributions have been proposed, with the majority [47, 60, 61, 63] following a similar way of generalising the normal distribution by adding a skewness weighting function [62]. Although attracting attentions in recent applications [124–126], this type of skewed elliptical distributions results in a different stochastic representation from the (symmetric) elliptical distribution, which is prohibitive to invariance analysis, sample generation and parameter estimation. A further extension employs a stochastic representation of elliptical distributions by assuming some inner dependency [37]. However, due to the added dependency, the relationships between parameters become unclear and sometimes interchangeable, thus impeding the estimation.

In this chapter, we start from the stochastic representation of elliptical distribu-

tions, and propose a novel generalisation by employing the von Mises-Fisher (vMF) distribution to explicitly specify the direction and skewness, whilst keeping the independence among the components in elliptical distributions. Such a generalisation is intuitive and maximally resembles the original (symmetric) elliptical distributions, which is beneficial in three aspects: i) it admits a simple and closed-form density function, so that all the elliptical distributions can be explicitly generalised as the proposed vMF elliptical distribution; ii) it shares many desirable properties with the original elliptical distribution, including the independence between the quadratic term (or the Mahalanobis distance) and the whitened variables, the invariance property, and explicit moments; iii) it inherits the robustness properties from the elliptical distributions, and can be estimated stably and efficiently, even by a naive numerical gradient descent method. This opens a new avenue for the design and implementation of robust probabilistic learning systems, such as generative models in unsupervised learning and discriminative models in supervised learning systems.

3.2 Generalisation via Von Mises-Fisher Distribution

3.2.1 The vMF Elliptical Distribution

Being distributed on a unit sphere surface, the vMF is a popular choice in directional statistics [127–130]. The vMF distribution is determined by two parameters: $\boldsymbol{\mu}_v$ for the main direction and τ for the concentration (denoted as $vMF(\boldsymbol{\mu}_v, \tau)$). Therefore, it is natural and beneficial to replace $\boldsymbol{S}$ in the elliptical distribution (introduced in Chapter 2.1.2) by the vMF distribution as a way of explicitly expressing the direction information. We thus propose a new type of generalisation of the elliptical distributions in the form

$$\boldsymbol{x} = {}^d \boldsymbol{\mu} + \mathcal{R}\boldsymbol{\Lambda}\boldsymbol{v}, \quad (3.1)$$

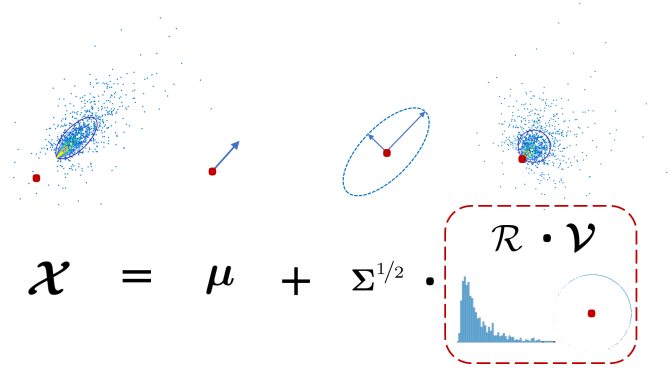


Figure 3.1: Illustration of generating the proposed vMF elliptical distribution via its stochastic representation of (3.1). The red dot denotes the point $(0, 0)$.

where $\mathcal{V}$ denotes a random variable satisfying the vMF distribution $vMF(\boldsymbol{\mu}_v, \tau)$. The stochastic representation is illustrated in Figure 3.1.

In our definition, $\mathcal{R}$ remains the same, i.e., non-negative and independent of $\mathcal{V}$. More importantly, when $\tau \rightarrow 0$, the vMF distribution approaches the uniform distribution on a unit sphere $\mathcal{S}$, and consequently (3.1) degenerates into a symmetric elliptical distribution. This generalisation maximally preserves the formats and desirable properties of the symmetric elliptical distribution, such as the independence and clear physical meaning of each part. In other words, in our proposed vMF elliptical distribution, $\boldsymbol{\mu}$ closely relates to the data location, $\boldsymbol{\Lambda}$ controls the dispersion, $\mathcal{R}$ governs the tails and $\mathcal{V}$ the directions (skewness). As shall be shown shortly, this is beneficial in both theoretical analysis and practical estimator settings.

The pdf of $\mathbf{X}$ in (3.1) can be obtained in a closed-form as

$$p_{\mathbf{X}}(\mathbf{x}) = \det(\boldsymbol{\Sigma})^{-1/2} \cdot p_{\mathcal{V}}\left(\frac{\boldsymbol{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu})}{\sqrt{t}}\right) \cdot g(t), \quad (3.2)$$

where t represents the Mahalanobis distance i.e., $t = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ and $p_{\mathcal{V}}(\cdot)$ is the pdf of vMF distribution $vMF(\boldsymbol{\mu}_v, \tau)$. We provide the proof of (3.2) in Appendix-3.4.1. An intuitive way of understanding our generalisation is through the fact that $vMF(\boldsymbol{\mu}_v, \tau)$ resembles a Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}_v, 1/\tau \mathbf{I})$, constrained on a unit circle (especially for an adequately large τ). Furthermore, the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_v, 1/\tau \mathbf{I})$, as a special case of the elliptical distribution, has the stochastic

representation $\mathbf{V} \approx^d \boldsymbol{\mu}_v + \frac{1}{\sqrt{\tau}} \sqrt{\chi_m^2} \mathbf{S}$. Please recall that $\mathbf{S}$ is a uniform distribution on a unit sphere surface as introduced in Chapter 2.1.2. Therefore, we can approximate (3.1) as

$$\begin{aligned} \mathbf{X} &\stackrel{d}{=} \boldsymbol{\mu} + \mathcal{R} \Lambda \mathbf{V} \approx^d \boldsymbol{\mu} + \mathcal{R} \Lambda (\boldsymbol{\mu}_v + \frac{1}{\sqrt{\tau}} \sqrt{\chi_m^2} \mathbf{S}) \\ &\approx^d \boldsymbol{\mu} + \Lambda \boldsymbol{\mu}_v \mathcal{R} + \frac{1}{\sqrt{\tau}} \Lambda (\mathcal{R} \sqrt{\chi_m^2}) \mathbf{S}. \end{aligned} \quad (3.3)$$

Given that random variables $\{\mathcal{R}, \sqrt{\chi_m^2}, \mathbf{S}\}$ are mutually independent, the term $(\boldsymbol{\mu} + \frac{1}{\sqrt{\tau}} \Lambda (\mathcal{R} \sqrt{\chi_m^2}) \mathbf{S})$ constitutes a symmetric elliptical distribution whilst the term $\Lambda \boldsymbol{\mu}_v \mathcal{R}$ adds to the asymmetry on the direction $\Lambda \boldsymbol{\mu}_v$.

3.2.2 Properties

We now proceed to introduce the properties of the proposed vMF elliptical distribution, with Proposition 3.1 addressing the invariance under the quadratic terms.

Proposition 3.1. *The skewness of the distribution is invariant to the quadratic term $(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu})$.*

Proof. From (3.1), we have

$$(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu}) \stackrel{d}{=} \mathcal{R}^2 \mathbf{V}^T \mathbf{V} \stackrel{d}{=} \mathcal{R}^2. \quad (3.4)$$

Because $\mathcal{R}$ is independent of $\mathbf{V}$, the quadratic term is irrelevant to the skewness part of our vMF elliptical distribution.

This completes the proof. □

The property in Proposition 3.1 is the basis for satisfying the distributional invariance [131], which provides convenience in applications such as hypothesis testing and dealing with sampling bias [47].

Given the fact that $\boldsymbol{\Sigma}^{-1/2} (\mathbf{X} - \boldsymbol{\mu}) / (\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu}) \stackrel{d}{=} \mathbf{V}$, from (3.1) we can easily obtain the following result based on the proof of the Proposition 3.1.

Proposition 3.2. *The quadratic term $(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu})$ is independent of the whitened random variable $\boldsymbol{\Sigma}^{-1/2}(\mathbf{X} - \boldsymbol{\mu})/(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu})$.*

This independence property plays a crucial role in the calculation of the robustness and other related proofs. Furthermore, due to the neat generalisation and clear physical meaning of our vMF elliptical distribution, we can write the moments in a closed-form, with the first two moments addressed in Proposition 3.3.

Proposition 3.3. *Given the generalisation of (3.1), we have*

$$\begin{aligned}\mathbb{E}[\mathbf{X}] &= \boldsymbol{\mu} + (\rho_m(\tau)\mathbb{E}[\mathcal{R}]\boldsymbol{\Lambda})\boldsymbol{\mu}_v \\ \text{Var}[\mathbf{X}] &= \mathbb{E}[\mathcal{R}]^2 \frac{\rho_m(\tau)}{\tau} \boldsymbol{\Sigma} - \mathbb{E}[\mathcal{R}]^2 \left(1 - \frac{m}{\tau} \rho_m(\tau) - \rho_m^2(\tau)\right) \boldsymbol{\Lambda} \boldsymbol{\mu}_v \boldsymbol{\mu}_v^T \boldsymbol{\Lambda}^T\end{aligned}\tag{3.5}$$

where $\rho_m(\tau) \in (0, 1)$ is the ratio of two modified Bessel functions of the first kind, given by

$$\rho_m(\tau) = I_{m/2}(\tau)/I_{m/2-1}(\tau).\tag{3.6}$$

Proof. Given the stochastic representation of (3.1), we have

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} + \mathbb{E}[\mathcal{R} \boldsymbol{\Lambda} \boldsymbol{\mathcal{V}}].\tag{3.7}$$

More importantly, due to the independence between $\mathcal{R}$ and $\boldsymbol{\mathcal{V}}$, we further have

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} + \mathbb{E}[\mathcal{R}] \boldsymbol{\Lambda} \mathbb{E}[\boldsymbol{\mathcal{V}}].\tag{3.8}$$

Recall that $\boldsymbol{\mathcal{V}}$ is the vMF parametrised as $vMF(\boldsymbol{\mu}_v, \tau)$. Its first-order moment is then calculated by $\mathbb{E}[\boldsymbol{\mathcal{V}}] = \rho_m(\tau) \boldsymbol{\mu}_v$ [132]. Thus, the first-order moment of the vMF elliptical distribution can be obtained as

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu} + (\rho_m(\tau)\mathbb{E}[\mathcal{R}]\boldsymbol{\Lambda})\boldsymbol{\mu}_v.\tag{3.9}$$

Similarly, the second-order moment of the vMF distribution is given by $\mathbb{E}[\boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{V}}^T] =$

$\rho_m(\tau)/\tau \mathbf{I} + (1 - m/\tau \rho_m(\tau)) \boldsymbol{\mu}_v \boldsymbol{\mu}_v^T$ [132], so that we arrive at

$$\begin{aligned}
\text{Var}[\boldsymbol{\mathcal{X}}] &= \mathbb{E}[\boldsymbol{\mathcal{X}} \boldsymbol{\mathcal{X}}^T] - \mathbb{E}[\boldsymbol{\mathcal{X}}] \mathbb{E}[\boldsymbol{\mathcal{X}}]^T \\
&= \mathbb{E}[\boldsymbol{\mu} \boldsymbol{\mu}^T + \boldsymbol{\mu} \mathcal{R} \boldsymbol{\mathcal{V}}^T \boldsymbol{\Lambda}^T + \mathcal{R} \boldsymbol{\Lambda} \boldsymbol{\mathcal{V}} \boldsymbol{\mu}^T + \mathcal{R}^2 \boldsymbol{\Lambda} \boldsymbol{\mathcal{V}} \boldsymbol{\mathcal{V}}^T \boldsymbol{\Lambda}^T] \\
&\quad - (\boldsymbol{\mu} \boldsymbol{\mu}^T + \mathbb{E}[\mathcal{R}] \boldsymbol{\mu} \mathbb{E}[\boldsymbol{\mathcal{V}}]^T \boldsymbol{\Lambda}^T + \mathbb{E}[\mathcal{R}] \boldsymbol{\Lambda} \mathbb{E}[\boldsymbol{\mathcal{V}}] \boldsymbol{\mu}^T + \mathbb{E}[\mathcal{R}]^2 \boldsymbol{\Lambda} \mathbb{E}[\boldsymbol{\mathcal{V}}] \mathbb{E}[\boldsymbol{\mathcal{V}}]^T \boldsymbol{\Lambda}^T) \\
&= \mathbb{E}[\mathcal{R}]^2 \boldsymbol{\Lambda} (\mathbb{E}[\boldsymbol{\mathcal{V}} \boldsymbol{\mathcal{V}}^T] - \mathbb{E}[\boldsymbol{\mathcal{V}}] \mathbb{E}[\boldsymbol{\mathcal{V}}]^T) \boldsymbol{\Lambda}^T \\
&= \mathbb{E}[\mathcal{R}]^2 \frac{\rho_m(\tau)}{\tau} \boldsymbol{\Sigma} - \mathbb{E}[\mathcal{R}]^2 (1 - \frac{m}{\tau} \rho_m(\tau) - \rho_m^2(\tau)) \boldsymbol{\Lambda} \boldsymbol{\mu}_v \boldsymbol{\mu}_v^T \boldsymbol{\Lambda}^T
\end{aligned} \tag{3.10}$$

This completes the proof. $\square$

Furthermore, when $\tau \rightarrow 0$, the following bounds hold, according to the Taylor series of the modified Bessel function of the first kind ($I_a(b) \approx \frac{1}{\Gamma(a+1)} (\frac{b}{2})^a$ for $b \rightarrow 0$):

$$\begin{aligned}
\lim_{\tau \rightarrow 0} \rho_m(\tau) &= \frac{I_{m/2}(\tau)}{I_{m/2-1}(\tau)} = \frac{\Gamma(m/2)}{2\Gamma(m/2+1)} \tau = 0, \\
\lim_{\tau \rightarrow 0} \frac{\rho_m(\tau)}{\tau} &= \frac{\Gamma(m/2)}{2\Gamma(m/2+1)} = \frac{\Gamma(m/2)}{2(m/2)\Gamma(m/2)} = \frac{1}{m}.
\end{aligned} \tag{3.11}$$

Therefore, from Proposition 3.3, when $\tau \rightarrow 0$, we have $\mathbb{E}[\boldsymbol{\mathcal{X}}] = \boldsymbol{\mu}$ and $\text{Var}[\boldsymbol{\mathcal{X}}] = \mathbb{E}[\mathcal{R}^2]/m \boldsymbol{\Sigma}$, which are exactly the moments of the symmetric elliptical distributions [37].

3.2.3 Optimisation

Given the i.i.d. nature of the samples $\{\mathbf{x}_i\}_{i=1}^n$, our vMF elliptical distribution can be estimated by maximising the log-likelihood (MLL), which is equivalent to minimising the Kullback-Leibler divergence between the parametric and empirical distributions. It also needs to be pointed out that other distance measures can be employed in estimation, such as the Wasserstein distance, a subject of future work.

For notational simplicity, we shall denote the whitened term as $\mathbf{z}_i = \frac{\boldsymbol{\Sigma}^{-1/2}(\mathbf{x}_i - \boldsymbol{\mu})}{\sqrt{t_i}}$ and the quadratic term as $t_i = (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})$. Then, due to the simple form of

the pdf of our generalised distribution, we can omit the constant term and write the log-likelihood from (3.2) as

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_v, \tau) &\propto -\frac{n}{2} \ln \det(\boldsymbol{\Sigma}) + n\left(\frac{m}{2} - 1\right) \ln \tau \\ &\quad - n \ln I_{m/2-1}(\tau) + \sum_{i=1}^n \tau \boldsymbol{\mu}_v^T \mathbf{z}_i + \sum_{i=1}^n \ln g(t_i) \end{aligned} \quad (3.12)$$

Furthermore, as $\|\boldsymbol{\mu}_v\| = 1$, this leads to constrained optimisation, whereby we reparametrise (3.12) by $\mathbf{v} = \tau \boldsymbol{\mu}_v$ (where $\tau = \|\mathbf{v}\|_2$ and $\boldsymbol{\mu}_v = \mathbf{v}/\|\mathbf{v}\|_2$) as

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{v}) &\propto -\frac{n}{2} \ln \det(\boldsymbol{\Sigma}) + n\left(\frac{m}{2} - 1\right) \ln \|\mathbf{v}\|_2 \\ &\quad - n \ln I_{m/2-1}(\|\mathbf{v}\|_2) + \sum_{i=1}^n \mathbf{v}^T \mathbf{z}_i + \sum_{i=1}^n \ln g(t_i). \end{aligned} \quad (3.13)$$

By taking the derivatives with respect to $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\mathbf{v}$, and setting them to $\mathbf{0}$, we obtain the following optimum, with the detailed proof given in Appendix 3.4.2.

$$\begin{aligned} \boldsymbol{\mu}^* &= \frac{\sum_{i=1}^n \left(\left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right) \mathbf{x}_i - \frac{1}{\sqrt{t_i}} \boldsymbol{\Sigma}^{\frac{1}{2}} \mathbf{v} \right)}{\sum_{i=1}^n \left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right)} \\ \boldsymbol{\Sigma}^* &= \frac{\sum_{i=1}^n \left(\left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right) (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T - \boldsymbol{\Sigma} \mathbf{z}_i \mathbf{v}^T \right)}{n} \\ \mathbf{v}^* &= \rho_m^{-1} \left(\left\| \frac{1}{n} \sum_{i=1}^n \mathbf{z}_i \right\|_2 \right) \frac{\sum_{i=1}^n \mathbf{z}_i}{\left\| \sum_{i=1}^n \mathbf{z}_i \right\|_2} \end{aligned} \quad (3.14)$$

In (3.14), $\rho_m^{-1}(\cdot)$ is an inverse function of $\rho_m(\cdot)$ and $\psi(t) = \frac{\partial g(t)/\partial t}{g(t)}$. Moreover, when $\tau \rightarrow 0$, $\mathbf{v}^* \rightarrow \mathbf{0}$, and the optimal $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ also represent the optimum of the symmetric elliptical distribution.

3.3 Numerical Results

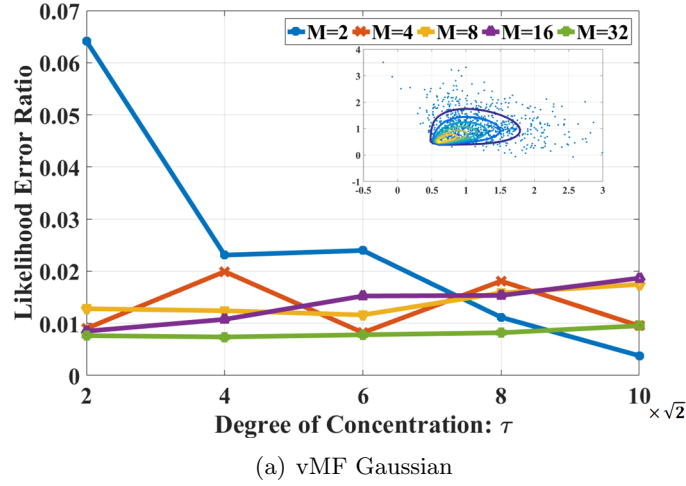
This section supports the analysis through experimental assessment of properties of the proposed vMF elliptical distribution, together with its estimation method. In our experiments, we tested 5 degrees of concentration, i.e., $\tau = \{2\sqrt{2}, 4\sqrt{2}, 6\sqrt{2}, 8\sqrt{2}, 10\sqrt{2}\}$,

together with 5 different dimensions ($m = \{2, 4, 8, 16, 32\}$), under two types of distributions, i.e., vMF Gaussian and vMF Cauchy distributions, thus having $5 \times 5 \times 2 = 50$ test cases. For each test case, we generated 1,000 synthetic samples from distributions of randomly chosen parameters $(\boldsymbol{\mu}, \boldsymbol{\mu}_v, \boldsymbol{\Sigma})$. The only controlled parameter was the eccentricity of the distribution, which is the ratio of the maximal eigenvalue over the minimal eigenvalue of $\boldsymbol{\Sigma}$; we set it smaller than 4 to avoid the majority of samples being aligned on low-dimensional spaces. For the synthetic data, in each test case, we performed estimation over 10 trails, with random initialisations. Standard steepest gradient descent was used to update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, with the learning rate of 0.01, while $\boldsymbol{\mu}_v$ and τ were optimised by a truncated Newton method proposed in [130]. The mean values of likelihood error ratios over the 10 trails are plotted in Figure 3.2, with the likelihood error ratio defined as

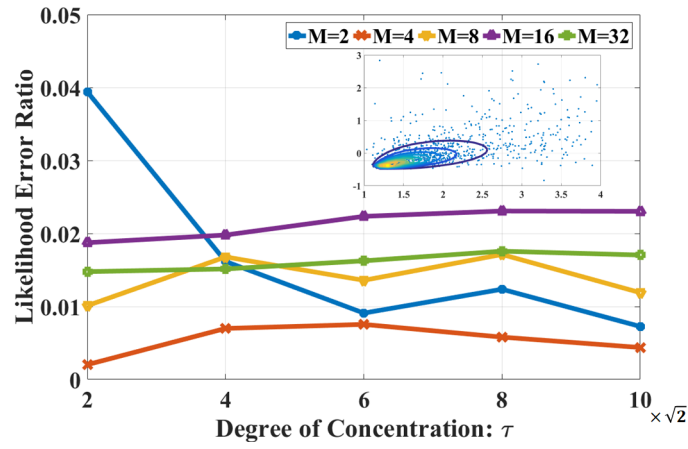
$$r = \frac{|l_{est} - l_{true}|}{l_{true}}, \quad (3.15)$$

while l_{est} denotes the log-likelihood given by the estimated parameters and l_{true} for the true log-likelihood.

From Figure 3.2, we observe that different from other generalised elliptical distributions, the generation of our vMF elliptical distributions is straightforward to implement and easy to control. More importantly, by virtue of its simple estimation method, the error ratios of estimating both vMF Gaussian and Cauchy distributions were both lower than 0.07, and even lower than 0.03 except for the case $m = 2$. The obtained small error ratios also verify that the proposed vMF elliptical distribution is not only easy to generate but is also stable. Moreover, Figure 3.2 also shows that increasing either the dimensions, m , or the degrees of concentration, τ , did not enlarge the estimated error, thus validating the consistency of our simple yet robust and effective estimation. Employing advanced numerical solvers such as conjugate gradient descent and trust region method could further improve the estimation, a subject of future work.



(a) vMF Gaussian



(b) vMF Cauchy

Figure 3.2: Log-likelihood error ratios with respect to different dimensions, m , and degrees of concentration, τ , in estimating the proposed: (a) vMF Gaussian and (b) vMF Cauchy distribution. One example per distribution of randomly generated samples from the stochastic representations of (3.1) is also provided in the upper right panel. For the convenience of illustration, we show the main part of vMF Cauchy samples whilst ignoring some highly scattered samples.

3.4 Appendix

3.4.1 Proof of the Closed-Form pdf in (3.2)

Notice that $\mathcal{R}$ is independent of $\mathcal{V}$, so that the following holds

$$p_{\{\mathcal{R}, \mathcal{V}\}}(r, \mathbf{v}) = p_{\mathcal{R}}(r) \cdot p_{\mathcal{V}}(\mathbf{v}) \quad (3.16)$$

We next define a random variable by a multiplication $\mathcal{Y} = f(\mathcal{R}, \mathcal{V}) = \mathcal{R} \cdot \mathcal{V}$.

It needs to be pointed out that $\mathbf{v}$ is a unit random vector, which means that the multiplication function f is an injection due to $\{\mathcal{R}, \mathbf{v}\} = f^{-1}(\mathcal{Y}) = \{\|\mathbf{y}\|_2, \mathbf{y}/\|\mathbf{y}\|_2\}$. Given the change of variables formula [133], the matrix volume is $\|\mathbf{y}\|_2^{-(m-1)}$. Thus, the pdf of $\mathcal{Y}$ can be obtained as

$$p_{\mathcal{Y}}(\mathbf{y}) = p_{\{\mathcal{R}, \mathbf{v}\}}(\|\mathbf{y}\|_2, \frac{\mathbf{y}}{\|\mathbf{y}\|_2}) \cdot \|\mathbf{y}\|_2^{-(m-1)} = p_{\mathcal{R}}(\|\mathbf{y}\|_2) \cdot p_{\mathbf{v}}(\frac{\mathbf{y}}{\|\mathbf{y}\|_2}) \cdot \|\mathbf{y}\|_2^{-(m-1)} \quad (3.17)$$

By employing the linear transform, $\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\mathbf{y}$, the pdf of our vMF elliptical distribution can be obtained by substituting $t = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \|\mathbf{y}\|_2^2$, as

$$p_{\mathbf{x}}(\mathbf{x}) = \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \cdot p_{\mathbf{v}}(\frac{\boldsymbol{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu})}{\sqrt{t}}) \cdot \underbrace{t^{-\frac{m-1}{2}} p_{\mathcal{R}}(\sqrt{t})}_{g(t)} \quad (3.18)$$

3.4.2 Proof of the Optimum in (3.14)

By using the relationship of the modified Bessel function of the first kind $\partial I_a(b)/\partial b = I_{a+1}(b) + a/b I_a(b)$, we can simplify the derivative of $\mathbf{v}$ as

$$\begin{aligned} \frac{1}{n} \frac{\partial L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{v})}{\partial \mathbf{v}} &= (\frac{m}{2} - 1) \frac{\mathbf{v}}{\|\mathbf{v}\|_2^2} + \frac{1}{n} \sum_{i=1}^n \mathbf{z}_i \\ &\quad - \frac{1}{I_{m/2-1}(\|\mathbf{v}\|_2)} \frac{\mathbf{v}}{\|\mathbf{v}\|_2} (I_{m/2}(\|\mathbf{v}\|_2) + \frac{\frac{m}{2} - 1}{\|\mathbf{v}\|_2} I_{m/2-1}(\|\mathbf{v}\|_2)) \\ &= -\frac{\mathbf{v}}{\|\mathbf{v}\|_2} \frac{I_{m/2}(\|\mathbf{v}\|_2)}{I_{m/2-1}(\|\mathbf{v}\|_2)} + \frac{1}{n} \sum_{i=1}^n \mathbf{z}_i \end{aligned} \quad (3.19)$$

By setting the derivative in (3.19) to $\mathbf{0}$, we arrive at the optimal $\mathbf{v}$ in the form

$$\mathbf{v}^* = \frac{\|\mathbf{v}\|_2}{\rho_m(\|\mathbf{v}\|_2)} \frac{1}{n} \sum_{i=1}^n \mathbf{z}_i = \rho_m^{-1}(\|\frac{1}{n} \sum_{i=1}^n \mathbf{z}_i\|_2) \frac{\sum_{i=1}^n \mathbf{z}_i}{\|\sum_{i=1}^n \mathbf{z}_i\|_2} \quad (3.20)$$

The optimal $\boldsymbol{\mu}$ can be calculated in a similar way

$$0 = \frac{\partial L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{v})}{\partial \boldsymbol{\mu}} = \sum_{i=1}^n \left(\frac{-\boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{v}}{t_i^{0.5}} + \frac{\mathbf{v}^T \mathbf{z}_i \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})}{t_i} \right) - 2\psi(t_i) \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}), \quad (3.21)$$

with the optimum at

$$\boldsymbol{\mu}^* = \frac{\sum_{i=1}^n \left(\left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right) \mathbf{x}_i - \frac{1}{\sqrt{t_i}} \boldsymbol{\Sigma}^{\frac{1}{2}} \mathbf{v} \right)}{\sum_{i=1}^n \left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right)} \quad (3.22)$$

To find the optimum of $\boldsymbol{\Sigma}$, it is convenient to decompose $\boldsymbol{\Sigma}$ into $\boldsymbol{\Lambda} \boldsymbol{\Lambda}^T$ by the Cholesky decomposition because the square root of the positive definite matrix $\boldsymbol{\Sigma}$ is guaranteed. Then, by taking the derivatives with regard to $\boldsymbol{\Lambda}$, we have

$$\begin{aligned} 0 &= \frac{\partial L(\boldsymbol{\mu}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda} \boldsymbol{\Lambda}^T, \mathbf{v})}{\partial \boldsymbol{\Lambda}} = -n \boldsymbol{\Lambda}^{-T} \\ &+ \sum_{i=1}^n \left(\frac{1}{t_i} \mathbf{v}^T \mathbf{z}_i \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Lambda}^{-T} - \mathbf{z}_i \mathbf{v}^T \boldsymbol{\Lambda}^{-T} \right) \\ &- \sum_{i=1}^n \left(2\psi(t_i) \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Lambda}^{-T} \right) \end{aligned} \quad (3.23)$$

After some simplifications, the optimal $\boldsymbol{\Sigma}$ can be easily obtained in a closed-form as

$$\boldsymbol{\Sigma}^* = \frac{\sum_{i=1}^n \left(\left(\frac{\mathbf{v}^T \mathbf{z}_i}{t_i} - 2\psi(t_i) \right) (\mathbf{x}_i - \boldsymbol{\mu}) (\mathbf{x}_i - \boldsymbol{\mu})^T - \boldsymbol{\Sigma} \mathbf{z}_i \mathbf{v}^T \right)}{n} \quad (3.24)$$

3.5 Summary

We have proposed a novel generalised elliptical distribution, termed the vMF elliptical distribution, in order to generalise symmetric elliptical distributions and equip them with physical meaning in the analysis, simple representations and the ability to represent general skewed distributions. This has been achieved in an intuitive manner, starting from the stochastic representation, which has enabled the proposed vMF elliptical distribution to exhibit many desirable properties, together with an explicit pdf. We have also introduced optimal parameter estimation into the vMF elliptical distributions, and have experimentally validated that even a basic steepest gradient descent method can achieve consistent and stable low estimation errors. This opens new avenues for both future applications and research on skewed distributions, especially in the context of probabilistic learning machines.

Chapter 4

General Framework for Learning the Elliptical Mixture Model

Mixture modelling using elliptical distributions promises enhanced robustness, flexibility and stability over the widely employed Gaussian mixture model (GMM). However, existing studies based on the elliptical mixture model (EMM) are restricted to several specific types of elliptical probability density functions, which are not supported by general solutions or systematic analysis frameworks; this significantly limits the rigour in the design and power of EMMs in applications. To this end, we propose a novel general framework for estimating and analysing the EMMs, achieved through Riemannian manifold optimisation. First, we investigate the relationships between Riemannian manifolds and elliptical distributions, and the so established connection between the original manifold and a reformulated one indicates a mismatch between these manifolds, a major cause of failure of the existing optimisation for solving general EMMs. We next propose a universal solver which is based on the optimisation of a re-designed cost and prove the existence of the same optimum as in the original problem; this is achieved in a simple, fast and stable way. We further calculate the influence functions of the EMM as theoretical bounds to quantify robustness to outliers. Comprehensive numerical results demonstrate the ability of the proposed framework to accommodate EMMs with different properties of individual

functions in a stable way and with fast convergence speed. Finally, the enhanced robustness and flexibility of the proposed framework over the standard GMM are demonstrated both analytically and through comprehensive simulations.

4.1 Introduction

Finite mixture models have a prominent role in statistical machine learning, owing to their ability to enhance probabilistic awareness in many learning paradigms, including clustering, feature extraction and density estimation [134]. Among such models, the GMM is the most widely used [135], with its popularity stemming from a simple formulation and the conjugate property of Gaussian distribution. Despite mathematical elegance, a standard GMM estimator is subject to robustness issues, as even a slight deviation from the Gaussian assumption or a single outlier in data can significantly degrade the performance or even break down the estimator [136]. Another issue with GMMs is their limited flexibility, which is prohibitive to their application in rapidly emerging scenarios based on multi-faceted data which are almost invariably unbalanced; sources of such imbalance may be due to different natures of the data channels involved, different powers in the constitutive channels, or temporal misalignment [137].

An important class of flexible multivariate analysis techniques are elliptical distributions (introduced in Chapter 2.1.2), which are quite general and include as special cases a range of standard distributions, such as the Gaussian distribution, the logistic distribution and the t -distribution [16]. The desired robustness to unbalanced multichannel data is naturally catered for in elliptical distributions; indeed estimating certain elliptical distribution types results in robust M-estimators [17], thus making them a natural candidate for robust and flexible mixture modelling. In this work, we therefore consider mixtures of elliptical distributions, or elliptical mixture model (EMM), in probabilistic modelling. By virtue of the inherent flexibility of EMMs, it is possible to model a wide range of standard distributions under one um-

brella, as EMM components may exhibit different properties, which makes EMMs both more suitable for capturing intrinsic data structures and more meaningful in interpreting data, as compared to the GMM. Another appealing property of EMMs is their identifiability in mixture problems, which has been proved by Holzmman *et al.* [138]. In addition, it has also been reported that several members of the EMM family can effectively mitigate the singular covariance problem experienced in the GMM [40].

Existing mixture models related to elliptical distributions are most frequently based on the t -distribution [40–42], the Laplace distribution [44], and the hyperbolic distribution [45]; these are optimised by a specific generalised expectation-maximisation process, called the iteratively reweighting algorithm (IRA) [38]. These elliptical distributions belong to the class of *scale mixture of normals* [139], where the IRA actually operates as an expectation maximisation (EM) algorithm, and such an EMM model is guaranteed to converge. However, for other types of elliptical distributions, the convergence of the IRA requires constraints on both the type of elliptical distributions and the data structure [38, 49, 50]. Therefore, although beneficial and promising, the development of a universal method for estimating the EMM is non-trivial, owing to both theoretical and practical difficulties.

To this end, we set out to rigorously establish a whole new framework for estimating and analysing the identifiable EMMs, thus opening an avenue for practical approaches based on general EMMs. More specifically, we first analyse the second-order statistical differential tensors to obtain the Riemannian metrics on the mean and the covariance of elliptical distributions. A reformulation trick is typically used to convert the mean-covariance-estimation problem into a covariance-estimation-only problem [38]. We further investigate the relationship between the manifolds with and without the reformulation, and find that an equivalence of the two manifolds only holds in the Gaussian case. Since for general elliptical distributions, the equivalence is not guaranteed, this means that a direct optimisation on the reformulated manifold cannot yield the optimum for all EMMs. To overcome this issue, we

propose a novel method with a modified cost of EMMs and optimise it on a matched Riemannian manifold via manifold gradient descent, where the same optimum as in the original problem is achieved in a fast and stable manner. The corresponding development of a gradient-based solver, rather than the EM-type solver (i.e., the IRA), is shown to be beneficial, as it offers more flexibility in model design, through various components and regularisations. We should point out that even for the EMMs where the IRA converges, our proposed method still outperforms the widely employed IRA. We finally systematically verify the robustness of EMMs by proving the influence functions (IFs) in closed-form, which serves as the theoretical bound.

The recent related work in [58, 59] adopts manifold optimisation for GMM problems by simply optimising on the intrinsic manifold. However, this strategy is inadequate in EMM problems due to a mismatch in manifolds for optimisation, which leads to a different optimum after reformulation. More importantly, as the flexibility of EMMs allows for inclusion of a wide range of distributions, this in turn requires the statistics of mixture modelling to be considered in the optimisation, whilst the work [58, 59] only starts from a manifold optimisation perspective. The key contributions of this work are summarised as follows:

- 1) We justify the usage of Riemannian metrics from the statistics of elliptical distributions, and in this way connect the original manifold with the reformulated one, where the convergence can be highly accelerated.
- 2) A novel method for accurately solving general EMMs in a fast and stable manner is proposed, thus making the flexible EMM truly practically applicable.
- 3) We rigorously prove the IFs in closed-form as theoretical bounds to qualify the robustness of EMMs, thus providing a systematic framework for treating the flexibility of EMMs.

4.2 Manifold Optimisation for the EMM

In this section, we shall first justify the Riemannian metrics of elliptical distributions in Section 4.2.1, followed by a layout of the EMM problem, and the introduction of the proposed method in Section 4.2.2. Finally, a novel type of regularisation on the EMMs is introduced in Section 4.2.3, which includes the *mean-shift* algorithm as a special case. Please note that we use the same notations to represent an elliptical distribution $\mathcal{X} \sim \mathcal{E}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$ given by Chapter 2.1.2.

4.2.1 Statistical Metrics for Elliptical Distributions

Although there are various metrics designed for measuring the distance between matrices [140–143], not all of them arise from the smooth varying inner product (i.e., Riemannian metrics), which would consequently give a “true” geodesic distance. One of the widely employed Riemannian metrics is the intrinsic metric $\text{tr}(d\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}d\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1})$, which can be obtained via the “entropy differential metric” [144] of two multivariate Gaussian distributions. The entropy related metric was later used by Hiai and Petz to define the Riemannian metric for positive definite matrices [145]. In this chapter, we follow the work of [145] to calculate the corresponding Riemannian metrics for the elliptical distributions.

Lemma 4.1. *Consider the class of elliptical distributions $\mathcal{E}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$. Then, given local coordinate vectors $d\boldsymbol{\Sigma}$, the Riemannian metric for the covariance is given by*

$$ds^2 = \frac{1}{2} \text{tr}(d\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}d\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}). \quad (4.1)$$

Proof. Please see Appendix-4.5.1. □

More importantly, as the metric is related to $\boldsymbol{\Sigma}$, the Levi-Civita connection is given by $\nabla_{\mathbf{X}}\mathbf{Y} = -\frac{1}{2}(\mathbf{X}\boldsymbol{\Sigma}^{-1}\mathbf{Y} + \mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{X})$ [146], where $\mathbf{X}, \mathbf{Y}$ are vector fields on the manifold of $\boldsymbol{\Sigma}$. The corresponding exponential mapping, which moves along with

the geodesics given the direction from a tangent vector, can be explicitly obtained as $\text{Exp}_{\Sigma}(\mathbf{U}) = \Sigma^{\frac{1}{2}} \exp(\Sigma^{-\frac{1}{2}} \mathbf{U} \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}}$, where $\mathbf{U} \in T_{\Sigma} \mathcal{M}$ [146].

When estimating parameters of elliptical distributions, the mean vector and the covariance matrix need to be estimated simultaneously. An elegant strategy would be to incorporate the mean and the covariance into an augmented matrix with one extra dimension [38]. Such a strategy has also been successfully employed in the work of [58, 59], which is called the “reformulation trick”. Thus, based on the metrics of Lemma 4.1, we can introduce the following relationship related to the reformulation.

Lemma 4.2. *Consider the class of elliptical distributions, $\mathcal{E}(\mathbf{y}|\mathbf{0}, \tilde{\Sigma}, g)$. Then, upon reformulating $\mathbf{y}$ and $\tilde{\Sigma}$ as*

$$\mathbf{y} = [\mathbf{x}^T, 1]^T, \quad \tilde{\Sigma} = \begin{pmatrix} \Sigma + \lambda \boldsymbol{\mu} \boldsymbol{\mu}^T & \lambda \boldsymbol{\mu} \\ \lambda \boldsymbol{\mu}^T & \lambda \end{pmatrix} \quad (4.2)$$

the subsequent Riemannian metric follows

$$\begin{aligned} ds^2 &= \text{tr}(d\tilde{\Sigma} \tilde{\Sigma}^{-1} d\tilde{\Sigma} \tilde{\Sigma}^{-1}) \\ &= \lambda d\boldsymbol{\mu}^T \Sigma^{-1} d\boldsymbol{\mu} + \frac{1}{2} \text{tr}(d\Sigma \Sigma^{-1} d\Sigma \Sigma^{-1}) + \frac{1}{2} (\lambda^{-1} d\lambda)^2. \end{aligned} \quad (4.3)$$

Proof. The proof is a direct extension of that in [147], where only the Gaussian case is proved, and will thus be omitted. $\square$

As $d\boldsymbol{\mu}^T \Sigma^{-1} d\boldsymbol{\mu}$ and $\frac{1}{2} \text{tr}(d\Sigma \Sigma^{-1} d\Sigma \Sigma^{-1})$ exactly formulate the two manifolds of EMMs without reformulation, we can inspect the relationship between the manifolds of EMMs with and without the reformulation from Lemma 4.2, which provides another perspective in understanding the reformulation.

Remark 4.1. *In Lemma 4.2, there is a mismatch between the two manifolds, due to the term $\frac{1}{2}(\lambda^{-1} d\lambda)^2$. When restricted to the Gaussian case, we show in the sequel that the gradient of λ vanishes when optimising $\tilde{\Sigma}$, i.e., $d\lambda = 0$. In this case, manifold optimisation on $\tilde{\Sigma}$ is performed under the same metric as a simultaneous*

optimisation on a product manifold of the mean and the covariance, which leads to the success of [58, 59] in solving GMMs. However, this property does not hold for general EMMs.

4.2.2 Manifold Optimisation on the EMM

Generally, we assume that the EMM consists of k mixing components, each elliptically distributed. To make the proposed EMM flexible enough to capture inherent structures in data, in our framework it is not necessary for every elliptical distribution within the mixture to have the same density generator (denoted by $\mathcal{E}_i(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, g_i)$). In finite mixture models, the probability of choosing the i -th mixing component is denoted by π_i , so that $\sum_{i=1}^k \pi_i = 1$. For a set of i.i.d samples $\mathbf{x}_j$, $j = 1, 2, 3, \dots, n$, when using the MLL method (aka. under the Kullback–Leibler divergence), the negative log-likelihood can be obtained as

$$L = - \sum_{j=1}^n \ln \sum_{i=1}^k \pi_i c_m \det(\boldsymbol{\Sigma}_i)^{-\frac{1}{2}} g_i((\mathbf{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i)). \quad (4.4)$$

The estimation of π_i , $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ therefore requires the minimisation of L in (4.4). By setting the derivatives of L to 0, we arrive at the following equations,

$$\begin{aligned} \pi_i &= \frac{\sum_{j=1}^n \xi_{ij}}{n}, \quad \boldsymbol{\mu}_i = \frac{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij}) \mathbf{x}_j}{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij})}, \\ \boldsymbol{\Sigma}_i &= -2 \frac{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij}) (\mathbf{x}_j - \boldsymbol{\mu}_i) (\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^n \xi_{ij}}, \end{aligned} \quad (4.5)$$

where $\xi_{ij} = \frac{\mathcal{E}_i(\mathbf{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, g_i) \pi_i}{\sum_{i=1}^k \mathcal{E}_i(\mathbf{x}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, g_i) \pi_i}$ is the posterior distribution of latent variables; $t_{ij} = (\mathbf{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i)$ is the Mahalanobis distance; $\psi_i(t_{ij}) = g'_i(t_{ij})/g_i(t_{ij})$ acts almost as an M-estimator for most heavily-tailed elliptical distributions, which decreases to 0 when the Mahalanobis distance t_{ij} increases to infinity. It is obvious that the solutions π_i , $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are intertwined with ξ_{ij} and t_{ij} , which prevents a closed-form solution¹ of (4.4). By iterating (4.5), this results to an EM-type solver, which is

¹It should be pointed out that there are multiple solutions to (4.5) and the goal here is to find a local

exactly the IRA algorithm. However, the convergence of the IRA is not guaranteed for general EMMs [38].

On the other hand, when directly estimating the reformulated EMM, i.e., $\mathcal{E}_i(\mathbf{y}|\mathbf{0}, \tilde{\Sigma}_i, g_i)$, similarly, we arrive at

$$\tilde{\pi}_i = \frac{\sum_{j=1}^n \tilde{\xi}_{ij}}{n}, \tilde{\Sigma}_i = -2 \frac{\sum_{j=1}^n \tilde{\xi}_{ij} \psi_i(\tilde{t}_{ij}) \mathbf{y}_j \mathbf{y}_j^T}{\sum_{j=1}^n \tilde{\xi}_{ij}}, \quad (4.6)$$

where $\tilde{t}_{ij} = \mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j$ and $\tilde{\xi}_{ij} = \frac{\mathcal{E}_i(\mathbf{y}_j|\mathbf{0}, \tilde{\Sigma}_i, g_i) \pi_i}{\sum_{i=1}^k \mathcal{E}_i(\mathbf{y}_j|\mathbf{0}, \tilde{\Sigma}_i, g_i) \pi_i}$. It needs to be pointed out that the directly reformulated EMM optimises on the augmented space $\mathbf{y}_j = [\mathbf{x}_j^T, 1]^T$ of $\mathbb{R}^{m+1}$, which is typically a mismatch to the original problem within the dimension m . This intrinsic difference becomes clear after decomposing $\tilde{\Sigma}_i$ to obtain the corresponding solutions, $\boldsymbol{\mu}_i$ and Σ_i , as well as λ_i . This is achieved in the form

$$\begin{pmatrix} \Sigma_i + \lambda_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T & \lambda_i \boldsymbol{\mu}_i \\ \lambda_i \boldsymbol{\mu}_i^T & \lambda_i \end{pmatrix} = -\frac{2}{\sum_{j=1}^n \tilde{\xi}_{ij}} \sum_{j=1}^n \tilde{\xi}_{ij} \psi_i(\tilde{t}_{ij}) \begin{pmatrix} \mathbf{x}_j \mathbf{x}_j^T & \mathbf{x}_j^T \\ \mathbf{x}_j & 1 \end{pmatrix}. \quad (4.7)$$

Because $(\tilde{t}_{ij} = \mathbf{y}_j^T \tilde{\Sigma}_i \mathbf{y}_j) \neq (t_{ij} = (\mathbf{x}_j - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i))$, $\psi_i(\tilde{t}_{ij})$ in (4.7) does not equal $\psi_i(t_{ij})$ in (4.5). The only exception is when $\psi_i(\cdot)$ is a constant, e.g., $\psi_i(\cdot) \equiv -\frac{1}{2}$ for the Gaussian distribution. In this case, $\lambda_i \equiv 1$, and the manifold with the reformulation is same as the original one.

To retain the same optimum as the original problem, we introduce a new parameter c_i , which aims to mitigate the mismatch of the reformulated manifold brought by λ_i . The same optimum is ensured in the following theorem.

Theorem 4.1. *The optimisation of π_i , $\tilde{\Sigma}_i$ and c_i based on the following re-designed cost*

$$\tilde{L} = -\sum_{j=1}^n \ln \sum_{i=1}^k \pi_i \cdot c_m \cdot (c_i \det(\tilde{\Sigma}_i))^{-1/2} g_i \left(\mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j - c_i \right) \quad (4.8)$$

stationary point. Finding the global optima is difficult in mixture problems [148] and beyond the scope of this chapter.

has the same optimum as those in (4.5):

$$\begin{aligned}\pi_i &= \frac{\sum_{j=1}^n \xi_{ij}}{n}, c_i = \frac{1}{\lambda_i} = -\frac{\sum_{j=1}^n \xi_{ij}}{2 \sum_{j=1}^n \xi_{ij} \psi_i(t_{ij})} \\ \tilde{\Sigma}_i &= -2 \frac{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij}) \mathbf{y}_j \mathbf{y}_j^T}{\sum_{j=1}^n \xi_{ij}}\end{aligned}\quad (4.9)$$

Proof. Please see Appendix-4.5.2. $\square$

We optimise (4.8) on a product manifold of $\tilde{\Sigma}_i$, π_i and c_i . For optimising $\tilde{\Sigma}_i$, on the basis of the metric in Section 4.2.1, we calculate Riemannian gradient as $\nabla_R \tilde{L} = \tilde{\Sigma}_i (\nabla_E \tilde{L}) \tilde{\Sigma}_i$, where $\nabla_E \tilde{L}$ is the Euclidean gradient of cost $\tilde{L}$ via $\partial \tilde{L} / \partial \tilde{\Sigma}_i$. Furthermore, although explicitly formulated, it is important to mention that the exponential mapping provided after Lemma 4.1 operates on a matrix, which comes with an extremely high computational complexity (typically $\mathcal{O}(m^4)$) and even needs a certain degree of approximation [149]. A common way to approximate the exponential mapping is via the retraction, of which the accuracy is up to the first order to the exponential mapping [150]. Thus, we employ the Taylor series expansion of $\exp(\Sigma^{-\frac{1}{2}} \mathbf{U} \Sigma^{-\frac{1}{2}})$ as a way of the approximation, via

$$\begin{aligned}\text{Exp}_{\Sigma}(\mathbf{U}) &= \Sigma^{\frac{1}{2}} \exp(\Sigma^{-\frac{1}{2}} \mathbf{U} \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}} \\ &\approx \Sigma^{\frac{1}{2}} (\mathbf{0} + \Sigma^{-\frac{1}{2}} \mathbf{U} \Sigma^{-\frac{1}{2}} + \frac{1}{2} \Sigma^{-\frac{1}{2}} \mathbf{U} \Sigma^{-1} \mathbf{U} \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}} \\ &= \Sigma + \mathbf{U} + \frac{1}{2} \mathbf{U} \Sigma^{-1} \mathbf{U} = R_{\Sigma}(\mathbf{U}),\end{aligned}\quad (4.10)$$

where the approximation is performed up to the cubic (third-order) term. It can be easily verified that $R_{\Sigma}(\mathbf{U})$ is a retraction (Chapter 4.1 of [103]), which significantly reduces the computational complexity from a matrix exponential to simple linear operations on matrices. Finally, we employ the conjugate gradient descent [151] as a manifold solver, with a pseudo-code for our method given in Algorithm 1.

The advantages of our algorithm, by virtue of its inherent reformulation, can be understood from two aspects. First, through the reformulation, our method is capable of providing a relatively global descent in terms of the re-parametrised $\tilde{\Sigma}_i$, whereas

Algorithm 1: The proposed method for optimising the EMM.

input: n observed samples: $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$;

initialisation: $\{\boldsymbol{\mu}_i^0\}_{i=1}^k$, $\{\boldsymbol{\Sigma}_i^0\}_{i=1}^k$, $\{\pi_i^0\}_{i=1}^k$, $\{c_i^0\}_{i=1}^k$, and $\{\lambda_i^{ini}\}_{i=1}^k$
for $i = 1$ **to** k **do**

 for $j = 1$ **to** n **do**

 $\mathbf{y}_j, \tilde{\boldsymbol{\Sigma}}_i^0 \leftarrow \text{Reparamet}(\mathbf{x}_j, \boldsymbol{\mu}_i^0, \boldsymbol{\Sigma}_i^0, \lambda_i^{ini})$
while *not converged (at the t -th iteration)*: **do**

 for $i = 1$ **to** k **do**

 Calculate Euclidean gradients $\nabla_E(\tilde{\boldsymbol{\Sigma}}_i^t)$, $\nabla_E(\pi_i^t)$ and $\nabla_E(c_i^t)$, by differentiating $\tilde{L}$ of (4.1)

 $\pi_i^{t+1} \leftarrow$ Step descent based on $\nabla_E(\pi_i^t)$

 $c_i^{t+1} \leftarrow$ Step descent based on $\nabla_E(c_i^t)$

 Update $\tilde{\boldsymbol{\Sigma}}_i^t$:

 $\nabla_R(\tilde{\boldsymbol{\Sigma}}_i^t) \leftarrow \text{Rgradient}(\tilde{\boldsymbol{\Sigma}}_i^t, \nabla_E(\tilde{\boldsymbol{\Sigma}}_i^t))$

 $\mathbf{U}_i^t \leftarrow$ Step descent based on $\nabla_R(\tilde{\boldsymbol{\Sigma}}_i^t)$

 $\tilde{\boldsymbol{\Sigma}}_i^{t+1} \leftarrow \text{Retraction}(\tilde{\boldsymbol{\Sigma}}_i^t, \mathbf{U}_i^t)$
for $i = 1$ **to** k **do**

 $\boldsymbol{\mu}_i^*, \boldsymbol{\Sigma}_i^* \leftarrow \text{Decomposition}(\tilde{\boldsymbol{\Sigma}}_i^*)$
output: $\{\boldsymbol{\mu}_i^*\}_{i=1}^k$, $\{\boldsymbol{\Sigma}_i^*\}_{i=1}^k$ and $\{\pi_i^*\}_{i=1}^k$
Function $\text{Reparamet}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)$:

Re-parametrisation via (4.2)

return $\mathbf{y}, \tilde{\boldsymbol{\Sigma}}$
Function $\text{Rgradient}(\boldsymbol{\Sigma}, \nabla_E)$:

 return $\nabla_R = (\boldsymbol{\Sigma} \nabla_E \boldsymbol{\Sigma})$
Function $\text{Retraction}(\boldsymbol{\Sigma}, \mathbf{U})$:

 return $\boldsymbol{\Sigma}_{new} = (\boldsymbol{\Sigma} + \mathbf{U} + \frac{1}{2} \mathbf{U} \boldsymbol{\Sigma}^{-1} \mathbf{U})$
Function $\text{Decomposition}(\tilde{\boldsymbol{\Sigma}}, c)$:

 Decompose via inverting (4.2): $\begin{pmatrix} \boldsymbol{\Sigma} + \frac{1}{c} \boldsymbol{\mu} \boldsymbol{\mu}^T & \frac{1}{c} \boldsymbol{\mu} \\ \frac{1}{c} \boldsymbol{\mu}^T & \frac{1}{c} \end{pmatrix} = \tilde{\boldsymbol{\Sigma}}$

 return $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

optimisation without the reformulation requires a sophisticated incorporated step descent on both $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, to ensure a well-behaved convergence. On the other hand, one typical singularity case is when certain $\boldsymbol{\mu}_i$ moves to the boundary of the data during optimisation, in which the cluster is likely to model a small set of data samples (e.g., one or two samples). In contrast, the proposed reformulated EMMs can be regarded as zero-mean mixtures, which to some extent relieves this singularity issue.

4.2.3 Regularisation

We impose the inverse-Wishart prior distribution to regularise the EMM (i.e., $p_{\Sigma_i}(\Sigma_i) \propto \frac{1}{\det(\Sigma_i)^{v/2}} \exp(-\frac{v\text{tr}(\Sigma_i^{-1}\mathbf{S})}{2})$), where v controls the freedom and $\mathbf{S}$ is the prior matrix. The advantages of using a form of $\text{tr}(\Sigma_i^{-1}\mathbf{S})$ are two-fold: i) it is strictly geodesic convex in Σ_i and ii) the solutions are ensured to exist for any data configuration [152]. By utilising maximising a posterior on covariance matrices, we obtain the same solutions of π_i and μ_i as those of (4.5), whereas Σ_i now becomes

$$\Sigma_i = \frac{-2 \cdot \sum_{j=1}^n \xi_{ij} \psi_i(t_{ij}) (\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^T + v\mathbf{S}}{\sum_{j=1}^n \xi_{ij} + v}. \quad (4.11)$$

Similar to Theorem 4.1, the following proposition can be obtained for the reformulation with regularisations.

Proposition 4.1. *The optimisation of π_i , $\tilde{\Sigma}_i$ and c_i based on the following cost function*

$$\tilde{L}_r = \tilde{L} + \sum_{i=1}^k (c_i \det(\tilde{\Sigma}_i))^{-v/2} \exp\left(-\frac{v\text{tr}(\tilde{\Sigma}_i^{-1}\tilde{\mathbf{S}})}{2}\right), \quad (4.12)$$

achieves the same optimal Σ_i as in (4.11) and the same π_i and μ_i as in (4.5), where $\tilde{\mathbf{S}} = \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$. The optimal c_i and λ_i are

$$c_i = \frac{1}{\lambda_i} = -\frac{\sum_{j=1}^n \xi_{ij} + v}{2 \sum_{j=1}^n \xi_{ij} \psi_i(t_{ij})}. \quad (4.13)$$

Proof. The proof is analogous to that of Theorem 4.1 and is therefore omitted. $\square$

Remark 4.2. *In (4.11), it can be seen that when $v \rightarrow \infty$, $\Sigma_i \rightarrow \mathbf{S}$. Furthermore, when $\mathbf{S} = \mathbf{I}_m$, $\Sigma_i = \sigma^2 \mathbf{I}_m$, the estimation of μ_i in (4.5) becomes $\frac{\sum_{j=1}^n \xi_{ij} \psi_i(\sigma^{-2} \|\mathbf{x}_j - \mu_i\|^2) \mathbf{x}_j}{\sum_{j=1}^n \xi_{ij} \psi_i(\sigma^{-2} \|\mathbf{x}_j - \mu_i\|^2)}$, which is the basic mean-shift algorithm with soft thresholds. Furthermore, when $\mathbf{S} = \mathbf{I}_m$, $\Sigma_i = \mathbf{I}_m$ and $\psi_i(t_{ij}) = -1/2$ (the GMM), it then turns to a soft version of the basic k -means algorithm. This all demonstrates that the EMM is a flexible framework in our regularisation settings and that we can choose v and $\mathbf{S}$ to achieve different models.*

It needs to be pointed out that although the inverse-Wishart prior is one of the popular priors (typically $\mathbf{S} = \mathbf{I}$), there are also other priors which suit different requirements. For example, there is also work using the Wishart prior, which is less informative but requires a particular setting of the parameters [153]. Instead of controlling the degrees of freedom by the scalar v , a generalised inverse-Wishart distribution has been applied to flexibly control the degrees of freedom [154]. Another pragmatic solution would be to decompose the covariance matrix into its standard deviation and correlation matrix components (inverse-Wishart distribution) so that the standard deviation can be treated in a flexible way [155]. Moreover, probabilistic graphical models can be used as a prior to explicitly control the sparsity of the matrices [156], where e.g., graphical LASSO can be applied. Furthermore, a robust distribution for positive definite matrices, named F -distribution, has become a popular choice for priors, which generalises the half-Cauchy and half- t distributions [157]. Recently, a Riemannian Gaussian distribution for the positive definite matrix has been proposed by replacing the Mahalanobis term with the Fisher-Rao metric of positive definite matrices [158]. A similar strategy can be extended to the Laplacian [159] and even to the elliptical distributions, which has a significant potential to generate a rich class of priors on positive definite matrices. This chapter investigates the inverse-Wishart prior as an example of regularisation, because it can further emphasise the flexibility of EMMs and also the compatibility of our re-parametrisation technique. The investigation on other priors is part of our future work.

4.3 Influence Functions of the EMM

Robustness properties of a single elliptical distribution (or more generally, an M-estimator) have been extensively studied [136, 160–162], typically from the perspective of influence functions (IFs) [163]. The IF is an important metric for quantifying the impact of an infinitesimal fraction of outliers on the estimations, which captures

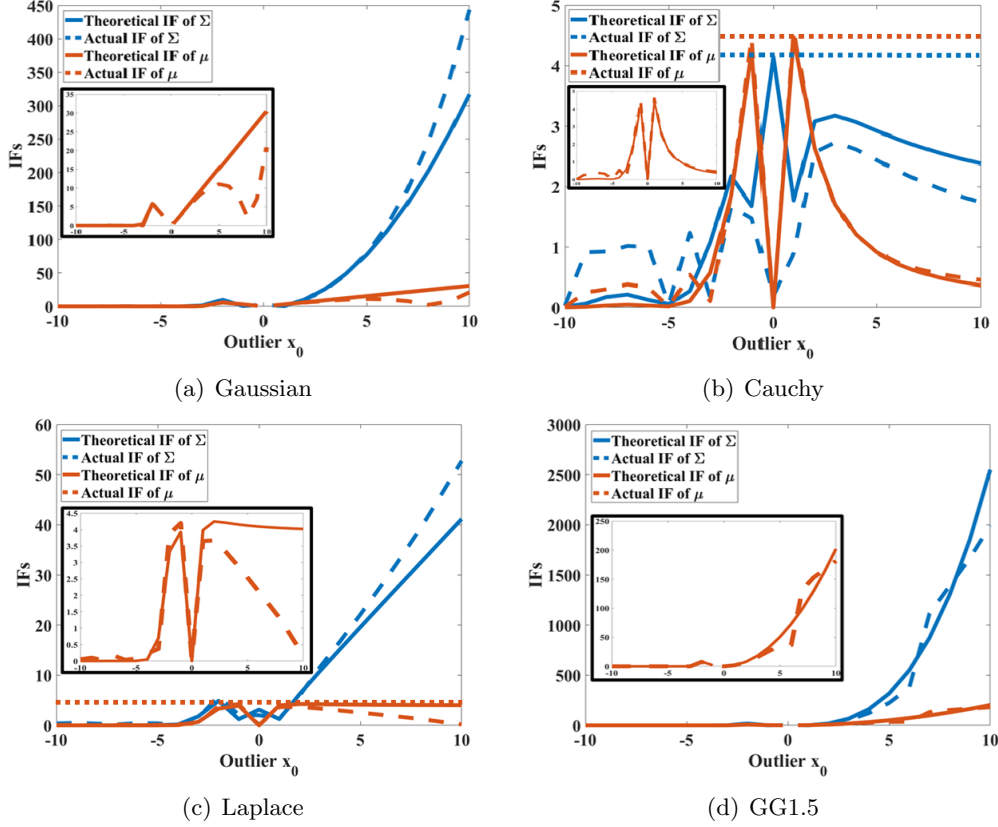


Figure 4.1: For reproducibility, we follow the work of [9] to generate three one-dimensional clusters using the inverse of the Gaussian cumulative distribution function. These three clusters are centred respectively at $\mu_1 = 0$, $\mu_2 = -5$ and $\mu_3 = -10$, and IF curves for μ_1 are illustrated. The mixture distributions are the Gaussian, Cauchy, Laplace and GG1.5, as shown in Table 4.1. The theoretical bounds are plotted in solid lines and the actual IFs are plotted in dotted lines. The horizontal dotted lines represent the boundedness (upper bounds) where the mixtures exhibit robustness. The zoomed versions of the IFs of the mean are given in the black box of each figure.

the local robustness. However, to the best of our knowledge, there exists no work on the IF of mixture models, especially for the EMMs. To calculate the IFs, we utilise $\mathbf{x}_0$ to denote point-mass outliers, which means that these outliers are point-mass distributed at $\mathbf{x}_0$ [136]. We also explicitly write the posterior distribution of latent variables as a function of $\mathbf{x}$ ($\xi_i(\mathbf{x}) = \frac{\mathcal{E}_i(\mathbf{x}|\mu_i, \Sigma_i, g_i)\pi_i}{\sum_{i=1}^k \mathcal{E}_i(\mathbf{x}|\mu_i, \Sigma_i, g_i)\pi_i}$), because in robustness analysis, we need to quantify it with respect to outliers. For simplicity, t_i is also defined as the Mahalanobis distance $(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)$ and $\mathbb{E}[\cdot]$ is the expectation over the true distribution of $\mathbf{x}$. Then, our analysis on the IFs is based on the following two lemmas.

Lemma 4.3. Consider the mixture of elliptical distributions, $\mathcal{E}_i(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, g_i)$. When data are well separated, upon denoting $(\mathbf{x}_0 - \boldsymbol{\mu}_i)$ by $\bar{\mathbf{x}}_0$ for the i -th cluster, its IF is given by,

$$F_{\boldsymbol{\Sigma}_i}(\mathbf{x}_0) = -\frac{\xi_i(\mathbf{x}_0)\psi_i(\bar{\mathbf{x}}_0^T \boldsymbol{\Sigma}_i^{-1} \bar{\mathbf{x}}_0)}{w_2} \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T + \boldsymbol{\Sigma}_i^{\frac{1}{2}} \left[\frac{2w_1 \cdot \xi_i(\mathbf{x}_0)\psi_i(\bar{\mathbf{x}}_0^T \boldsymbol{\Sigma}_i^{-1} \bar{\mathbf{x}}_0) \bar{\mathbf{x}}_0^T \boldsymbol{\Sigma}_i^{-1} \bar{\mathbf{x}}_0 + w_2 \cdot \xi_i(\mathbf{x}_0) \mathbf{I}}{2(mw_1 - w_2)w_2} \right] \boldsymbol{\Sigma}_i^{\frac{1}{2}}, \quad (4.14)$$

where w_1 and w_2 are constants (irrelevant to the outlier $\mathbf{x}_0$) given by

$$w_1 = \frac{\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t^2] + \mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t^2]}{m(m+1)} + \frac{\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i(t)t]}{m} + \frac{\mathbb{E}[\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x})]}{4}, \quad (4.15)$$

$$w_2 = \frac{\pi_i}{2} - \frac{\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t^2] + \mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t^2]}{m(m+1)}.$$

Lemma 4.4. Consider the mixture of elliptical distributions, $\mathcal{E}_i(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, g_i)$. When data are well separated, for the i -th cluster, its IF on the mean is given by

$$F_{\boldsymbol{\mu}_i}(\mathbf{x}_0) = \frac{1}{w_3} \xi_i(\mathbf{x}_0)\psi_i(\mathbf{x}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_i), \quad (4.16)$$

where w_3 is a constant (irrelevant to the outlier $\mathbf{x}_0$) given by

$$w_3 = \frac{2\mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t]}{m} + \mathbb{E}[\xi_i(\mathbf{x})\psi_i(t)] + \frac{2\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t]}{m}. \quad (4.17)$$

Proofs of the two lemmas are provided in the Appendices-4.5.3 and 4.5.4. The actual² and theoretical IF curves of the four EMMs are plotted in Fig. 4.1, showing that in practice the robustness of the EMMs can be well captured by our theoretical bounds.

More importantly, the robustness of the EMM can also be analysed from Lemmas

²The actual IF is obtained via numerical tests on the actual difference between the estimated parameter and the ground truth when increasing the absolute value of a single outlier, to establish whether an outlier could totally break down the estimation; this is cumbersome and requires extensive repeated estimations to obtain the curve.

4.3 and 4.4, and is determined by $\psi_i(\cdot)$ (or $g_i(\cdot)$) of each cluster. Specifically, when $\mathbf{x}_0 \rightarrow \infty$, $F_{\Sigma_i}(\mathbf{x}_0)$ is bounded (defined as *covariance robust*) only when $\psi_i(t)t$ is bounded for $t \rightarrow \infty$, which leads to bounded $\psi_i(\bar{\mathbf{x}}_0^T \Sigma_i^{-1} \bar{\mathbf{x}}_0) \bar{\mathbf{x}}_0^T \Sigma_i^{-1} \bar{\mathbf{x}}_0$ in (4.14). Likewise, bounded $F_{\mu_i}(\mathbf{x}_0)$ (defined as *mean robust*) requires bounded $\psi_i(t)\sqrt{t}$, which is slightly more relaxed than the requirement of *covariance robust*. For example, in Fig. 4.1, by inspecting the boundedness of the curves, we find that the Gaussian and GG1.5 mixtures are neither *covariance robust* and *mean robust*, while the Cauchy mixtures are both *covariance robust* and *mean robust*. For the Laplace mixtures, they are not *covariance robust* but are *mean robust*, which shows that the *covariance robust* is more stringent than the *mean robust*. Thus, the developed bounds provide an extremely feasible and convenient treatment for qualifying or designing the robustness within EMMs.

4.4 Experimental Results

Our experimental settings are first detailed in Section 4.4.1. We then employ in Section 4.4.2 two toy examples to illustrate the flexibility of EMMs in capturing different types of data. This also highlights the virtues of our method in universally solving EMMs. In Section 4.4.3, we systematically compare our EMM solver with other baselines on the synthetic dataset, followed by a further evaluation on the image data of BSDS500 in Section 4.4.4.

4.4.1 Parameter Settings and Environments

Baselines: We compared the proposed method (**Our**) with the regular manifold optimisation (**RMO**) method without reformulation (i.e., updating μ_i and Σ_i separately) and the **IRA** method, by optimising different EMMs over various data structures. It should be pointed out that the IRA includes a range of existing works on solving certain EMMs, e.g., the standard EM algorithm for the Gaussian distribution, [40] for the t -distribution and [45] for the hyperbolic distribution. Besides, the

convergence criterion in all the experiments was set by the cost decrease of adjacent iterations of less than 10^{-10} . For our method and the RMO, that involved manifold optimisation, we have utilised the default conjugate gradient solver in the Manopt toolbox [164]. We should also point out that we evaluated all the methods on original EMM problems and due to the fact that priors are highly data-dependent, we leave the reasonable and comprehensive evaluations on regularised EMMs as part of our future work.

Performance objectives: We compared our method with the RMO and IRA methods by comprehensively employing 9 different elliptical distributions³ as components within EMMs. These are listed in Table 4.1, with their properties provided in Table 4.2, where the Cauchy distribution is a special case of the student- t distribution with $v = 1$. We should also point out that the non-geodesic elliptical distributions cannot be solved by the IRA method [49, 50]. In contrast, as shown below, our method can provide a stable and fast solution even for the non-geodesic elliptical distributions.

Table 4.1: Details of the 9 elliptical distributions used for assessments

	Gaussian	Student- t ($v = 1$ and $v = 10$)	GG1.5	Logistic
$g(t)$	$g(t) \propto \exp(-0.5t)$	$g(t) \propto (1 + t/v)^{-(m+v)/2}$	$g(t) \propto \exp(-0.5t^{1.5})$	$g(t) \propto \frac{\exp(-t)}{(1 + \exp(-t))^2}$
	Laplace	Weib0.9	Weib1.1	Gamma1.1
$g(t)$	$g(t) \propto \frac{\mathcal{K}_{(1-0.5m)(\sqrt{2t})}}{\sqrt{0.5t}^{0.5m-1}}$	$g(t) \propto t^{-0.1} \exp(-0.5t^{0.9})$	$g(t) \propto t^{0.1} \exp(-0.5t^{1.1})$	$g(t) \propto t^{0.1} \exp(-0.5t)$
Note:	$\mathcal{K}_x(y)$ is the modified Bessel function of the second kind. Student- t with $v = 1$ is the Cauchy distribution.			

Synthetic datasets: We generated the synthetic dataset via randomly choosing the mean and the covariance, except for the *separation* c and *eccentricity* e [135, 165], which were controlled to comprehensively evaluate the proposed method under various types of data structures. The separation, c , of two clusters k_1 and k_2 is defined as $\|\boldsymbol{\mu}_{k_1} - \boldsymbol{\mu}_{k_2}\|^2 \geq c \cdot \max\{\text{tr}(\boldsymbol{\Sigma}_{k_1}), \text{tr}(\boldsymbol{\Sigma}_{k_2})\}$, and the eccentricity, e , is defined as a ratio of the largest and the smallest eigenvalue of the covariance matrix within one cluster. The smaller value of c indicates the larger overlaps between clusters; the

³Please note that in the next chapter (Chapter 5), we will provide a “ready-to-use” summary on other computable and identifiable EMMs.

Table 4.2: Properties of the elliptical distributions used for evaluations

	Gaussian	Student- <i>t</i>	Laplace	GG1.5
Covariance Robust	No	Yes	No	No
Mean Robust	No	Yes	Yes	No
Heavily Tailed	No	Yes	Yes	No
Geodesic Convex	Yes	Yes	Yes	Yes
	Logistic	Weib0.9	Weib1.1	Gamma1.1
Covariance Robust	No	No	No	No
Mean Robust	No	No	No	No
Heavily Tailed	No	Yes	No	No
Geodesic Convex	Yes	Yes	No	No

smaller value of e means more spherically distributed clusters. In total, we generated $3 \times 2 = 6$ types of synthetic datasets, whereby m and k were set in pairs to $\{8, 8\}$, $\{16, 16\}$, and $\{64, 64\}$; c and e were set in pairs to $\{10, 10\}$ and $\{0.1, 1\}$ to represent the two extreme cases. Each synthetic dataset contained 10,000 samples in total ($n = 10,000$) drawn from different mixtures of Gaussian distributions. For each test case (i.e., for each method and for each EMM), we repeatedly ran the optimisation over 50 trials, with random initialisations. Finally, we recorded average values of the iterations, the computational time and the final cost. When the optimisation failed, i.e., converging to singular covariance matrices or infinite negative likelihood, we also recorded and calculated the optimisation fail ratio within the 50 initialisations for each test case, to evaluate the stability in optimisation.

BSDS500 dataset: Finally, we evaluated our method on the image data, over two typical tasks. The first was related to image segmentation, where all the 500 pictures in the Berkeley segmentation dataset BSDS500 [166] were tested and reported in our results. We set $k = 2$ in this task in order to clearly show the effects of different EMMs in segmentation (as shown in Fig. 4.6). Evaluation over multiple parameters k was included in the second task. Moreover, each optimisation was initialised by the *k-means++* using the vl-feat toolbox [167], which is a typical initialisation method in clustering tasks such as the *k-means* clustering. The cost, iterations and computational time were recorded for all the 500 pictures. In the second task, our evaluation was implemented on another challenging task, by modelling and clustering 3×3 and 5×5 image patches from the image dataset. It needs to be pointed

out that this task is a core part of many applications, such as image denoising, image super-resolution and image/video compression, where similarities of image patches play an important role. Specifically, we randomly extracted 100 patches (3×3 and 5×5) from each image in the BSDS500, and vectorised those patches as data samples. Thus, we finally obtained the test data with sizes $50,000 \times 27$ and $50,000 \times 75$, where k was set to 3 and 9. Also, we ran each optimisation with 50 times random initialisations, and recorded the average final cost and the standard deviation.

4.4.2 Toy Examples

Before comprehensively evaluating our method, we first provide some intuition behind its performance based on flower-shaped data with 4 clusters ($n = 10,000$) (shown in Fig. 4.2-(a)). The flexibility of the EMMs via our method is illustrated by: (i) adding 100% uniform noise (i.e., 10,000 noisy samples), as shown in Fig. 4.2-(b); (ii) replacing two clusters by the Cauchy samples with the same mean and covariance matrices, as shown in Fig. 4.2-(c). The five distributions that were chosen as components in EMMs are shown in Fig. 4.2-(d).

The optimised EMMs are shown in Fig. 4.3. From this figure, we find that the GMM is inferior in modelling noisy or unbalanced data. This is mainly due to its lack of robustness, and thus similar results can be found in another non-robust EMM, i.e., the GG1.5. In contrast, for a robust EMM, such as the Cauchy and the Laplace, the desirable level of estimation is ensured in both cases. Therefore, a universal solver is crucial as it enables flexible EMMs can be well optimised for different types of data.

We further plot the iteration numbers against the average cost difference of Our, IRA and RMO methods when optimising the two cases in Fig. 4.4 and 4.5. Note that for the purpose of illustrations [58], the cost difference is defined by the absolute difference between the cost of each iteration and the relatively “optimal” cost,

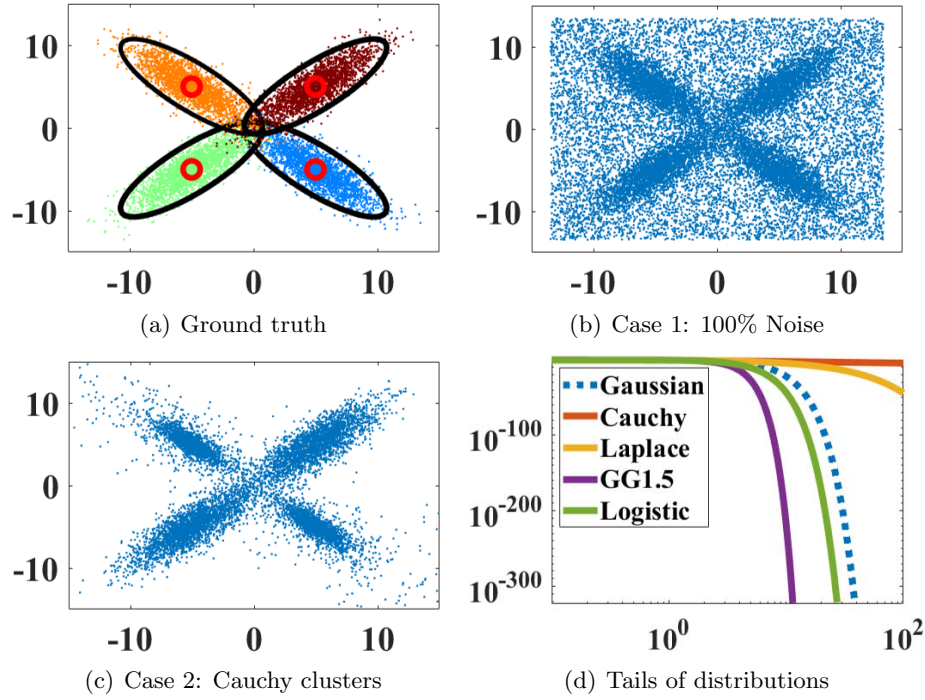


Figure 4.2: Toy examples consisting of four Gaussian sets. (a) Data structure: The red circles represent the mean values at $(5, 5)$, $(5, -5)$, $(-5, 5)$ and $(-5, -5)$, and the black circles denote covariance ellipses including 95% data samples of each Gaussian distribution. (b) Test case adding 100% uniform noisy samples to the data. (c) Test case of data that consist of two Gaussian and two Cauchy sets. (d) Tails of the distributions which were utilised in this test. It needs to be pointed out that the Cauchy samples in (c) are spread over a wide range, so that we show (c) within $(\pm 15, \pm 15)$ for illustration convenience.

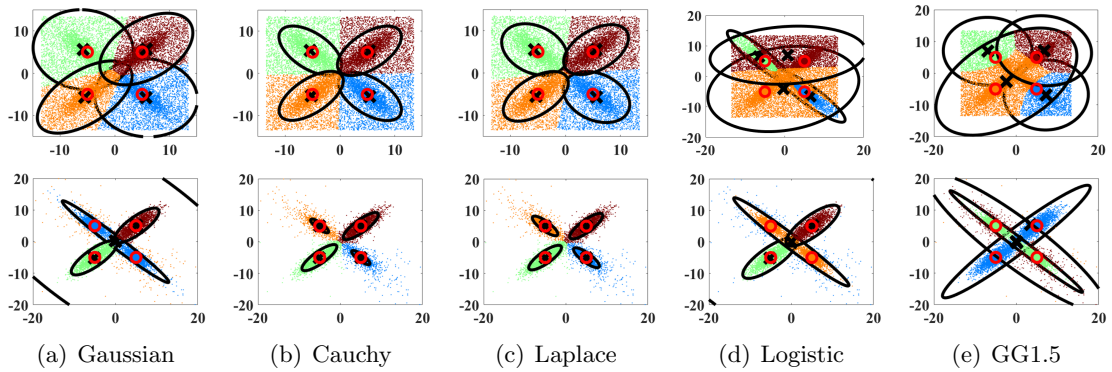


Figure 4.3: Optimisation results of the proposed method across the 5 EMMs. The top row shows the results for Case 1 and the bottom row shows the results for Case 2. The red circles denote the ground-truth as shown in Fig. 4.2-(a), whilst the black crosses and ellipses represent the estimated mean and covariance matrices. The colour of each sample is corresponding to that of Fig. 4.2-(a), and is classified by selecting the maximum posterior among the clusters.

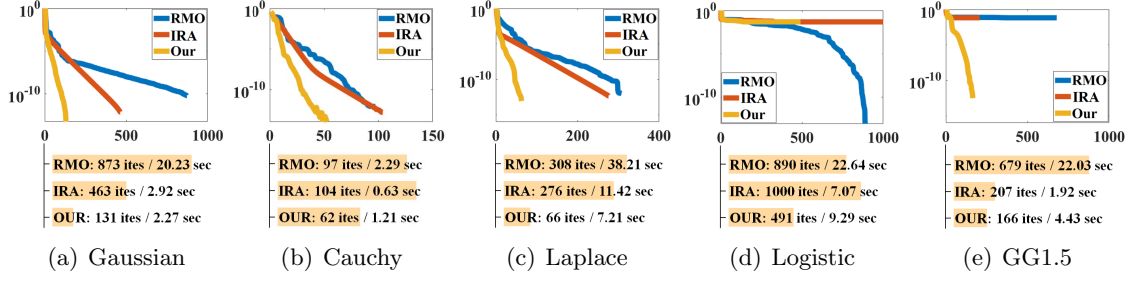


Figure 4.4: Cost difference against the number of iterations of the 5 EMMs for Case 1. The top figures show the cost difference against the iterations optimised via Our, IRA and RMO methods. The bottom quantities are the final convergence speed in terms of the number of iterations and execution time (ites/sec).

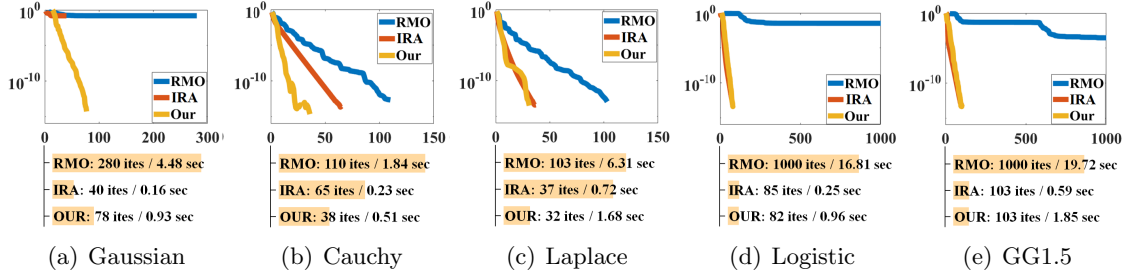


Figure 4.5: Cost difference against the number of iterations of the 5 EMMs for Case 2. The top figures show the cost difference against the iterations optimised via Our, IRA and RMO methods. The bottom quantities are the final convergence speed in terms of the number of iterations and execution time (ites/sec).

which was obtained by choosing the lowest value among the final costs of the three methods. From the two figures, the IRA achieved a monotonic convergence, because it consistently increases the lower bound of the log-likelihood; our method, although occasionally fluctuating at the late stage of convergence, such as for the Cauchy distribution in Fig. 4.5, consistently achieved the lowest cost among all EMMs and converged with the least number of iterations. A further cautious choice of optimisers as well as line search methods can probably achieve a monotonic convergence. Moreover, although one iteration of our method takes longer time than that of the IRA method due to the line search, the overall computational time of our method is comparable to that of the IRA method. As shown in the next section, our method even performs much faster in terms of computational time than the IRA for higher dimensions and larger numbers of clusters ($m > 2$ and $k > 4$).

Table 4.3: Overall results averaged across m and k for 9 types of ENMs optimised by Our, IRA and RMO methods.

$c = 10, e = 10$										
	Gaussian	Cauchy	Student- t ($v = 10$)	GG1.5	Logistic	Weib0.9	Weib1.1	Gamma1.1	Mix	
Our	Ite. / T. (s)	108 / 40.3	135 / 34.5	136 / 32.7	219 / 160	113 / 43.2	158 / 37.8	139 / 56.5	138 / 38.3	132 / 51.8
	Cost	64.5	65.6	64.8	69.8	47.8	64.4	64.4	64.4	67.5
	Ite. / T. (s)	367 / 90.0	—	—	—	428 / 249	310 / 46.8	—	—	—
IRA	Cost	66.2	—	—	—	49.0	64.5	—	—	—
	Ite. / T. (s)	658 / 1738	848 / 1590	776 / 1681	796 / 1435	744 / 1651	825 / 1806	773 / 1625	760 / 1550	711 / 1547
	Cost	64.3	65.8	64.7	71.6	47.5	64.4	64.5	64.5	66.8
$c = 0.1, e = 1$										
Our	Ite. / T. (s)	693 / 201	424 / 99.9	621 / 106	581 / 100	734 / 242	686 / 119	655 / 111	664 / 94.6	470 / 65.7
	Cost	40.0	41.0	40.2	39.8	23.5	39.1	40.1	40.1	39.2
	Ite. / T. (s)	—	—	—	—	—	711 / 101	—	—	—
IRA	Cost	—	—	—	—	—	40.6	—	—	—
	Ite. / T. (s)	1000 / 1707	956 / 1661	951 / 1789	898 / 1206	963 / 1508	969 / 1632	975 / 1744	958 / 1561	917 / 1505
	Cost	40.4	41.5	40.8	42.1	23.8	40.3	40.5	40.4	40.4
RMO										
Note:	T. (s): Time (seconds); Ite.: Iteration numbers; —: Singularity or infinity in either $m = 8$, $m = 16$ or $m = 64$.									

Table 4.4: Detailed comparisons among Our, IRA and RMO on optimising 3 ENMs in the case of $c = 10$ and $e = 10$

(m, k)	Gaussian						Cauchy						Logistic							
	Our		IRA		RMO		Our		IRA		RMO		Our		IRA		RMO			
(8, 8)	T. (s)	3.70 ± 5.44	5.48 ± 4.13	4.23 ± 10.76	2.23 ± 1.67	3.04 ± 2.45	20.4 ± 10.6	4.28 ± 4.60	4.71 ± 4.08	18.9 ± 17.3	Ite.	165 ± 240	538 ± 403	122 ± 308	107 ± 81.3	340 ± 275	640 ± 334	177 ± 191	463 ± 398	537 ± 489
	Co./Fa.	19.4 / 0%	19.6 / 0%	19.3 / 0%	20.3 / 0%	20.4 / 0%	20.3 / 0%	14.9 / 0%	15.0 / 0%	14.9 / 0%										
	T. (s)	15.8 ± 13.4	28.8 ± 15.2	221 ± 78.3	38.1 ± 34.5	40.9 ± 18.2	260 ± 1.84	15.7 ± 7.64	23.6 ± 14.6	193 ± 115										
(16, 16)	Ite.	115 ± 94.0	400 ± 207	853 ± 304	272 ± 243	570 ± 254	1000 ± 0.00	115 ± 54.0	325 ± 201	734 ± 433	Co./Fa.	37.7 / 0%	38.0 / 0%	37.8 / 0%	38.7 / 0%	38.8 / 0%	39.0 / 0%	28.6 / 0%	28.8 / 0%	28.6 / 0%
	T. (s)	101 ± 18.0	236 ± 0.00	4988 ± 152	63.2 ± 11.7	—	4491 ± 1549	110 ± 19.3	720 ± 343	4741 ± 475										
	Ite.	45.8 ± 5.67	163 ± 0.00	1000 ± 0.00	26.9 ± 3.70	—	902 ± 309	49.3 ± 5.98	497 ± 235	960 ± 121										
(64, 64)	Co./Fa.	136 / 0%	141 / 90%	136 / 10%	138 / 0%	— / 100%	138 / 0%	99.7 / 0%	103 / 70%	99.1 / 10%										
Note:	T. (s): Time (seconds); Ite.: Iteration numbers; Co.: Final cost; Fa.: Optimisation fail ratio; —: Singularity or infinity																			

4.4.3 Evaluations over the Synthetic Datasets

We next systematically evaluated our method based on the synthetic dataset described in Section 4.4.1. For each dataset, we had $9 \times 3 = 27$ test cases, i.e., 9 types of EMMs for the 3 methods. The 9 types of EMMs include 8 types of elliptical distributions and the other one (denoted as Mix) is composed by half the number of Cauchy distributions and the other half of Gaussian distributions. Table 4.3 shows the overall result averaged across dimensions m and k for the 9 EMMs. As can be seen from Table 4.3, our method exhibits the fastest convergence speed in terms of both the number of iterations and computation time, and it also obtains the minimum cost. It can also be found that datasets with more overlaps (i.e., $\{c = 0.1, e = 1\}$) take a longer time to optimise, whereby iterations and computational time increase for all the 3 methods. On the other hand, by comparing the results of our method and those of the RMO method, we can clearly see a significant improvement in both convergence speed and final minimum, which verifies the effectiveness of our reformulation technique.

We provide further details of the comparisons of different m and k in Table 4.4, where 3 EMMs and $\{c = 10, e = 10\}$ were reported due to the space limitation and the fact that similar results can be found for other EMMs and settings of Table 4.3. We can see from this table that the superior performance of our method is consistent over different dimensions m and cluster numbers k . Table 4.4 shows that our method requires the minimum number of iterations as well as least computational time. More importantly, the standard deviations for the iterations and computational time of our method are almost the lowest, which means that our method is able to stably optimise the EMMs. With an increase in m and k , our method consistently achieves the best performance of the average costs with 0% fail ratio. In contrast, the IRA can become extremely unstable. One reason is due to the fact that, the IRA cannot converge for the non-geodesic convex distributions such as the *Weib1.1* and *Gamma1.1* in Table 4.3 [48]. Another perspective is that it even failed on geodesic convex distributions in Table 4.4 (e.g., 90% fail ratio for the Gaussian

and 100% fail ratio for the Cauchy when $m = 64, k = 64$). This may be due to the separate updating scheme on μ_i and Σ_i , which has been mentioned in Section 4.2.2. Although we see an enhanced stability when using manifold optimisation, this separate updating scheme, we believe, is also the reason that the RMO requires extremely large computational complexity to converge (> 900 iterations and > 4000 seconds when $m = 64, k = 64$) and in several cases it even failed to converge altogether. Our method, on the one hand, re-parametrises the parameters to perform a simultaneous update on μ_i and Σ_i via $\tilde{\Sigma}_i$; the re-parametrised EMMs also have a fixed zero mean that prevents potential clusters to move to the data boundary. These two aspects enable our method to consistently and stably optimise EMMs with the fastest convergence speed and lowest cost.

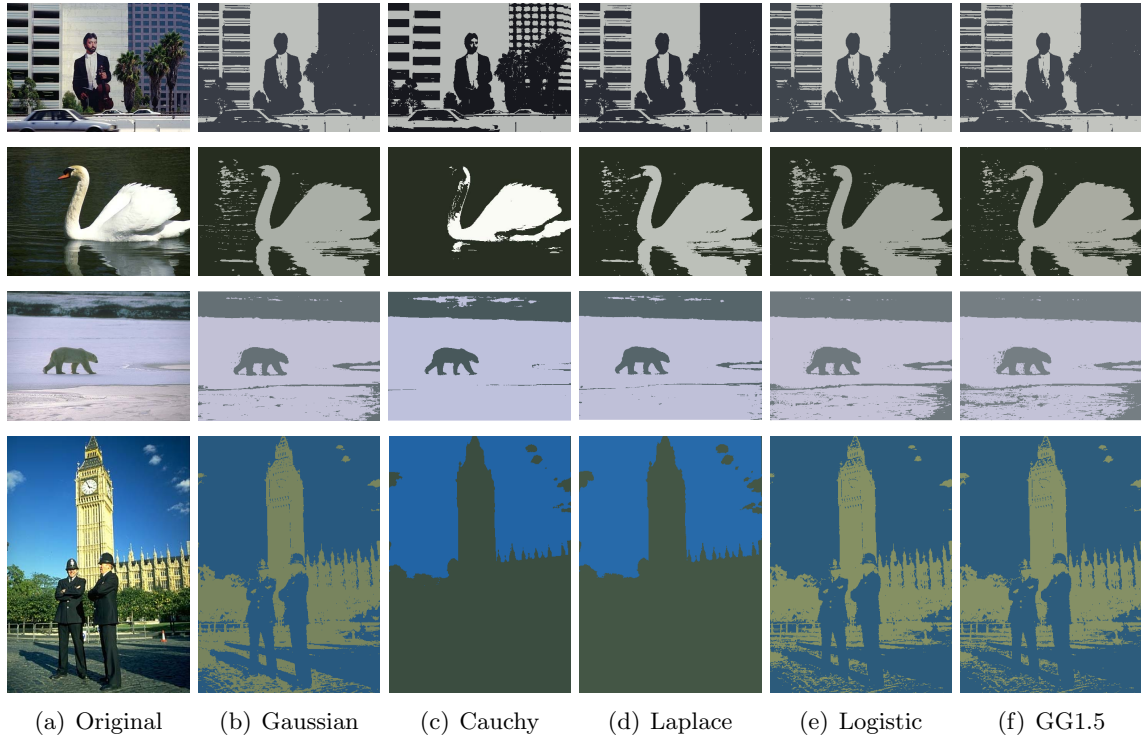


Figure 4.6: Reconstructed images by Our method when optimising the 5 EMMs.

4.4.4 BSDS500 Dataset

Convergence speed between Our, the IRA, and the RMO methods was compared over averaged results across the whole 500 pictures in BSDS500, as shown in Table

Table 4.5: Comparisons averaged over 500 pictures among Our, the IRA and the RMO for estimating the 5 EMNs

	Gaussian			Cauchy			Laplace			GG1.5			Logistic		
	Our	IRA	RMO	Our	IRA	RMO	Our	IRA	RMO	Our	IRA	RMO	Our	IRA	RMO
Iterations	56	234	954	87	390	939	101	288	962	61	224	982	58	239	965
Time (s)	5.55	11.8	165	7.12	16.8	159	20.4	42.2	486	8.32	14.4	205	6.73	13.3	177
Cost	12.3	12.3	12.3	12.1	12.4	12.3	12.2	12.2	12.3	12.4	12.4	12.4	11.1	11.1	11.2
SSIM [168]		0.5930			0.6052			0.6112			0.5813			0.5792	

Table 4.6: Comparisons among Our (with different manifold solvers), IRA and RMO methods on the BSDS500 image patches

	Gaussian						Cauchy					
	Our	Our (STP)	Our (LBF)	IRA	RMO	Our	Our (STP)	Our (LBF)	IRA	RMO	Our	Our (STP)
$k, m = 3, 27$												
Time (s)	28.3±7.62	210±109	102±8.67	155±37.0	486±197	118±72.9	224±106	302±139	—	556±26.2		
Iterations	109±27.5	749±390	318±37.3	930±221	844±342	416±254	817±387	670±253	—	983±54.4		
Cost	72.8±1.30	73.6±6.30	72.0±1.84	73.9±2.01	75.5±6.72	70.7±1.53	69.5±4.63	70.5±1.47	—	70.7±1.82		
$k, m = 9, 75$												
Time (s)	170±23.3	125±34.1	1609±74.7	—	4568±904	595±473	2025±38.6	898±157	—	4389±1229		
Iterations	90.0±8.80	63.0±25.5	998±7.07	—	921±177	295±230	1e3±0	394±93.0	—	918±259		
Cost	160±3.33	178±21.1	160±2.89	—	172±0.31	170±0.25	171±3.32	171±2.37	—	173±4.31		

4.5. Again, our method converged with the fastest speed and achieved the minimum cost error among the RMO and the IRA methods. We further show in Fig. 4.6 four reconstructed images via our method when optimising the five EMMs. Observe that the images reconstructed from the non-robust distributions (e.g., the Gaussian and the GG1.5 distributions) were more “noisy” than those from the robust distributions (e.g., the Cauchy and Laplace distributions); however, these may capture more details in images. Thus, by our method, different EMMs can flexibly model data for different requirements or applications.

Finally, we evaluated our method on modelling the BSDS500 image patches, which is a challenging but important task in practice. More comprehensively, besides the manifold conjugate gradient solver adopted as a default of our method, we further evaluated the manifold steepest descent (denoted as Our (STP)) and the manifold LBFGS (denoted as Our (LBF)) as alternatives to solve the step descent in Algorithm 1. The results are reported in Table 4.6. From this table, we can see again that our method consistently achieved the lowest costs across different initialisations. For example, for the Gaussian mixtures, it obtained 72.8 ± 1.30 in the 50 random initialisations, compared to 73.9 ± 2.01 of the IRA and 75.5 ± 6.72 of the RMO. Also, our method converged with the least computational time and was able, compared to the existing methods, to consistently provide fast, stable and superior optimisation. Furthermore, by employing different solvers, the results achieved by our method are comparable to one another; among the three solvers, the steepest descent solver took slightly more computational time due to its naive update rule, whilst the conjugate gradient solver performed slightly better than the other two solvers in balancing the computational time and cost. Overall, all the three solvers of our method achieved much better results than the IRA and the RMO methods, which also verifies the effectiveness of our re-designed cost. Therefore, we can conclude that for various scenarios and applications, our method was able to consistently yield a superior model with the lowest computational time.

4.5 Appendix

4.5.1 Proof of Lemma 4.1

To calculate the Riemannian metric for the covariance matrix, we follow the work of [145] to calculate the Hessian of the Boltzman entropy of elliptical distributions. The Boltzman entropy is first obtained as follows,

$$\begin{aligned} H(\mathbf{x}|\Sigma) &= \int_{\mathbf{x}} p_{\mathbf{x}}(\mathbf{x}) \ln p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbf{x}} p_{\mathbf{x}}(\mathbf{x}) \left[-\frac{1}{2} \ln |\Sigma| + \ln c_m + \ln g(t) \right] d\mathbf{x} \\ &= -\frac{1}{2} \ln |\Sigma| + \ln c_m + \int_{\mathbb{R}^m} p_{\mathcal{R}}(t) \ln g(t) dt. \end{aligned} \quad (4.18)$$

Because $(\ln c_m + \int_{\mathbb{R}^m} p_{\mathcal{R}}(t) \ln g(t) dt)$ is irrelevant to Σ , the Hessian of $H(\mathbf{x}|\Sigma)$ can be calculated as

$$\frac{\partial H(\mathbf{x}|\Sigma + s\Sigma_0 + h\Sigma_1)}{\partial s \partial h} \Big|_{s=0, h=0} = \frac{1}{2} \text{tr}(\Sigma_0 \Sigma^{-1} \Sigma_1 \Sigma^{-1}). \quad (4.19)$$

The Riemannian metric can thus be obtained as $ds^2 = \frac{1}{2} \text{tr}(d\Sigma \Sigma^{-1} d\Sigma \Sigma^{-1})$, which is the same as the case for multivariate normal distributions and is the mostly widely used metric.

This completes the proof.

4.5.2 Proof of Theorem 4.1

The proof of this property rests upon an expansion $\mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j$ to become $(\mathbf{x}_j - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i) + \frac{1}{\lambda_i}$ within derivatives of $\tilde{L}$. By setting $\partial \tilde{L} / \partial \lambda_i = 0$ and $\partial \tilde{L} / \partial c_i = 0$, we then arrive at

$$\begin{aligned} \lambda_i &= -2 \frac{\sum_{j=1}^n \tilde{\xi}_{ij} \psi_i(t_{ij} + \frac{1}{\lambda_i} - c_i)}{\sum_{j=1}^n \tilde{\xi}_{ij}}, \\ c_i &= -\frac{1}{2} \frac{\sum_{j=1}^n \tilde{\xi}_{ij}}{\sum_{j=1}^n \tilde{\xi}_{ij} \psi_i(t_{ij} + \frac{1}{\lambda_i} - c_i)}, \end{aligned} \quad (4.20)$$

where

$$\tilde{\xi}_{ij} = \frac{\pi_i \cdot c_m \cdot \sqrt{c_i \cdot \det(\tilde{\Sigma}_i^{-1})} \cdot g_i \left(\mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j - c_i \right)}{\sum_{i=1}^k \pi_i \cdot c_m \cdot \sqrt{c_i \cdot \det(\tilde{\Sigma}_i^{-1})} \cdot g_i \left(\mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j - c_i \right)}. \quad (4.21)$$

By inspecting $\lambda_i = 1/c_i$, $\det(\tilde{\Sigma}_i) = \lambda_i \cdot \det(\Sigma_i)$ and $\mathbf{y}_j^T \tilde{\Sigma}_i^{-1} \mathbf{y}_j = (\mathbf{x}_j - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i) + \frac{1}{\lambda_i}$, we obtain $\tilde{\xi}_{ij} = \xi_{ij}$ and $\psi_i(t_{ij} + \frac{1}{\lambda_i} - c_i) = \psi_i(t_{ij})$.

To prove the equivalence of the optimum of $\tilde{\Sigma}_i$ in (4.8) and optima of $\boldsymbol{\mu}_i$ and Σ_i , we directly calculate $\partial \tilde{L} / \partial \tilde{\Sigma}_i$ in (4.8) and set it to 0, to yield

$$\tilde{\Sigma}_i = -2 \frac{\sum_{j=1}^n \tilde{\xi}_{ij} \psi_i(t_{ij} + \frac{1}{\lambda_i} - c_i) \mathbf{y}_j \mathbf{y}_j^T}{\sum_{j=1}^n \tilde{\xi}_{ij}}. \quad (4.22)$$

Again, as $\lambda_i = 1/c_i$ and $\tilde{\xi}_{ij} = \xi_{ij}$, we arrive at

$$\tilde{\Sigma}_i = -2 \frac{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij}) \mathbf{y}_j \mathbf{y}_j^T}{\sum_{j=1}^n \xi_{ij}} = -2 \frac{\sum_{j=1}^n \xi_{ij} \psi_i(t_{ij})}{\sum_{j=1}^n \xi_{ij}} \begin{bmatrix} \mathbf{x}_j \mathbf{x}_j^T & \mathbf{x}_j \\ \mathbf{x}_j^T & 1 \end{bmatrix}. \quad (4.23)$$

By substituting $\boldsymbol{\mu}_i$ and Σ_i of (4.5) and λ_i in (4.20), we have

$$\tilde{\Sigma}_i = \begin{bmatrix} \Sigma_i + \lambda_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T & \lambda_i \boldsymbol{\mu}_i \\ \lambda_i \boldsymbol{\mu}_i^T & \lambda_i \end{bmatrix}, \quad (4.24)$$

which means that the optimum value $\tilde{\Sigma}_i$ is exactly the reformulated form by the optimum values of (4.5).

This completes the proof.

4.5.3 Proof of Lemma 4.3

We here denote the contaminated distribution $\mathcal{F} \stackrel{d}{=} (1 - \epsilon) \mathcal{F}_{\mathbf{x}} + \epsilon \mathcal{F}_{\mathbf{x}_0}$, where ϵ is the proportion of outliers; $\mathcal{F}_{\mathbf{x}}$ is the true distribution of $\mathbf{x}$ and $\mathcal{F}_{\mathbf{x}_0}$ is the point-mass distribution at $\mathbf{x}_0$. For simplicity, we employ t to denote $(\mathbf{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)$ and t_0 to denote $(\mathbf{x}_0 - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_0 - \boldsymbol{\mu}_i)$. Then, the maximum log-likelihood estimation

on the Σ_i of the EMM becomes

$$\begin{aligned} (1 - \epsilon)\mathbb{E}[\xi_i(\mathbf{x})\psi_i(t)(\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T + \frac{1}{2}\xi_i(\mathbf{x})\Sigma_i] \\ + \epsilon\xi_i(\mathbf{x}_0)\psi_i(t_0)(\mathbf{x}_0 - \boldsymbol{\mu}_i)(\mathbf{x}_0 - \boldsymbol{\mu}_i)^T + \epsilon\frac{\xi_i(\mathbf{x}_0)}{2}\Sigma_i = 0. \end{aligned} \quad (4.25)$$

We first calculate the IF (denoted by F in the proof) when $\Sigma_i = \mathbf{I}$ and $\boldsymbol{\mu}_i = \mathbf{0}$. Thus, we have $t = \mathbf{x}^T \mathbf{x}$ and $t_0 = \mathbf{x}_0^T \mathbf{x}_0$ in the following proof. Then, according to the definition of IF, we differentiate (4.25) with respect to ϵ and when it approaches 0, we arrive at

$$\begin{aligned} \mathbb{E}\left[\frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\Big|_{\epsilon=0}\psi_i(t)\mathbf{x}\mathbf{x}^T + \xi_i(\mathbf{x})\frac{\partial \psi_i(\mathbf{x}^T \Sigma_i^{-1} \mathbf{x})}{\partial \epsilon}\Big|_{\epsilon=0}\mathbf{x}\mathbf{x}^T + \frac{1}{2}\frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\Big|_{\epsilon=0}\mathbf{I} + \frac{1}{2}\xi_i(\mathbf{x})F\right] \\ + \xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T + \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I} = 0. \end{aligned} \quad (4.26)$$

In addition, we obtain

$$\begin{aligned} \frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\Big|_{\epsilon=0} &= (\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x})) \cdot \left(-\frac{1}{2}\text{tr}(F) - \psi_i(t)\mathbf{x}^T F \mathbf{x}\right), \\ \frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\Big|_{\epsilon=0} &= -\psi_i'(t)\mathbf{x}^T F \mathbf{x}. \end{aligned} \quad (4.27)$$

By combining (4.26) and (4.27), we arrive at

$$\begin{aligned} \mathbb{E}\left[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x})) \cdot \left(-\frac{1}{2}\text{tr}(F) - \psi_i(t)\mathbf{x}^T F \mathbf{x}\right) \cdot (\psi_i(t)\mathbf{x}\mathbf{x}^T + \frac{1}{2}\mathbf{I})\right. \\ \left. - \xi_i(\mathbf{x})\psi_i'(t)(\mathbf{x}^T F \mathbf{x})\mathbf{x}\mathbf{x}^T + \frac{1}{2}\xi_i(\mathbf{x})F\right] \\ + \xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T + \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I} = 0. \end{aligned} \quad (4.28)$$

It should be pointed out that $(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})$ has the same distribution as $\mathcal{R}^2$. It is thus independent of $\frac{\Sigma^{-1/2}(\mathbf{x} - \boldsymbol{\mu})}{\sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma (\mathbf{x} - \boldsymbol{\mu})}}$ (denoted by $\mathbf{u}$), which has the same distribution as $\mathcal{S}$ (i.e., uniform distribution). For mixing components, when data are well-separated, those $\mathbf{x}$ that do not belong to the i -th cluster have extremely low $\xi_i(\mathbf{x})$. In other words, the expectation in (4.28) is dominated by the expectation of the data which belong to the i -th cluster. Therefore, to calculate the expectation,

we can treat the quadratic term $(\mathbf{x} - \mathbf{0})^T \mathbf{I}(\mathbf{x} - \mathbf{0}) = \mathbf{x}^T \mathbf{x}$ (i.e., t) as independent of the normalised term $\frac{\mathbf{I}^{-1/2}(\mathbf{x}-\mathbf{0})}{\sqrt{t}} = \frac{\mathbf{x}}{\sqrt{t}}$ (i.e., $\mathbf{u}$) of the i -th cluster.

Based on this approximation, we can rewrite (4.28) as

$$\begin{aligned} \mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x})) \cdot (-\frac{1}{2}\text{tr}(F) - \psi_i(t)t \cdot \mathbf{u}^T F \mathbf{u}) \cdot (\psi_i(t)t \cdot \mathbf{u} \mathbf{u}^T + \frac{1}{2}\mathbf{I}) \\ - \xi_i(\mathbf{x})\psi_i'(t)t^2(\mathbf{u}^T F \mathbf{u})\mathbf{u} \mathbf{u}^T + \frac{1}{2}\xi_i(\mathbf{x})F] \\ + \xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T + \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I} = 0. \end{aligned} \quad (4.29)$$

Moreover, as $\mathbf{u}$ is uniformly distributed, we have $\mathbb{E}[\mathbf{u} \mathbf{u}^T] = \frac{1}{m}\mathbf{I}$, $\mathbb{E}[\mathbf{u}^T F \mathbf{u}] = \frac{1}{m}\text{tr}(F)$ and $\mathbb{E}[(\mathbf{u}^T F \mathbf{u})\mathbf{u} \mathbf{u}^T] = \frac{1}{m(m+1)}(F + \text{tr}(F)\mathbf{I})$. Thus, we arrive at

$$\begin{aligned} - \frac{(\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t^2] + \mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t^2])(F + \text{tr}(F)\mathbf{I})}{m(m+1)} \\ - \frac{\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i(t)t]\text{tr}(F)}{2m}\mathbf{I} - \frac{\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))]\text{tr}(F)}{4}\mathbf{I} \\ - \frac{1}{2m}\mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i(t)t]\text{tr}(F)\mathbf{I} + \frac{\pi_i}{2}F \\ + \xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T + \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I} = 0. \end{aligned} \quad (4.30)$$

With w_1 and w_2 in (4.15), we can re-write (4.30) as

$$w_2 F = w_1 \text{tr}(F)\mathbf{I} - \xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T - \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I}. \quad (4.31)$$

Then, by taking the trace on both sides of (4.31), we have

$$\text{tr}(F) = \frac{\xi_i(\mathbf{x}_0)\psi_i(t_0)\mathbf{x}_0\mathbf{x}_0^T + \frac{1}{2}\xi_i(\mathbf{x}_0)\mathbf{I}}{mw_1 - w_2} \quad (4.32)$$

Thus, the IF at point $\mathbf{x}_0$ of the i -th cluster, when $\Sigma_i = \mathbf{I}$, can be obtained as

$$F(\mathbf{x}_0) = \left[\frac{2w_1 \cdot \xi_i(\mathbf{x}_0)\psi_i(\mathbf{x}_0^T \mathbf{x}_0)\mathbf{x}_0^T \mathbf{x}_0 + w_2 \cdot \xi_i(\mathbf{x}_0)}{2(mw_1 - w_2)w_2} \right] \cdot \mathbf{I} - \frac{\xi_i(\mathbf{x}_0)\psi_i(\mathbf{x}_0^T \mathbf{x}_0)}{w_2} \mathbf{x}_0 \mathbf{x}_0^T. \quad (4.33)$$

The IF is then obtained at point $\mathbf{x}_0$ of the i -th cluster for general Σ_i and μ_i according to its affine equivalence (i.e., $F_{\Sigma_i}(\mathbf{x}_0) = \Sigma_i^{1/2} F(\Sigma_i^{-1/2}(\mathbf{x}_0 - \mu_i)) \Sigma_i^{1/2}$).

This completes the proof.

4.5.4 Proof of Lemma 4.4

Similar to the proof of Lemma 3, we have the following equation for estimating μ_i with contaminated distribution $\mathcal{F} =^d (1 - \epsilon)\mathcal{F}_{\mathbf{x}} + \epsilon\mathcal{F}_{\mathbf{x}_0}$:

$$(1 - \epsilon)\mathbb{E}[\xi_i(\mathbf{x})\psi_i(t)\Sigma_i^{-1}(\mathbf{x} - \mu_i)] + \epsilon\xi_i(\mathbf{x}_0)\psi_i(t_0)\Sigma_i^{-1}(\mathbf{x}_0 - \mu_i) = 0 \quad (4.34)$$

Note that in (4.34), for simplicity we also use t to denote $(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i)$ and t_0 to denote $(\mathbf{x}_0 - \mu_i)^T \Sigma_i^{-1}(\mathbf{x}_0 - \mu_i)$. We here also utilise F to denote the IF in this proof.

In addition, by differentiating with ϵ and $\epsilon \rightarrow 0$, we arrive at

$$\begin{aligned} & \mathbb{E}\left[\frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\bigg|_{\epsilon=0}\psi_i(t)\Sigma_i^{-1}(\mathbf{x} - \mu_i)\right. \\ & \quad \left.+ \xi_i(\mathbf{x})\frac{\partial \psi_i(t)}{\partial \epsilon}\bigg|_{\epsilon=0}\Sigma_i^{-1}(\mathbf{x} - \mu_i) - \xi_i(\mathbf{x})\psi_i(t)\Sigma_i^{-1}F\right] \\ & \quad + \xi_i(\mathbf{x}_0)\psi_i(t_0)\Sigma_i^{-1}(\mathbf{x}_0 - \mu_i) = 0. \end{aligned} \quad (4.35)$$

Besides, we also calculate

$$\begin{aligned} \frac{\partial \xi_i(\mathbf{x})}{\partial \epsilon}\bigg|_{\epsilon=0} &= (\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x})) \cdot (-2\psi_i(t)(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}F) \\ \frac{\partial \psi_i(\mathbf{x})}{\partial \epsilon}\bigg|_{\epsilon=0} &= -2\psi_i'(t)(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}F. \end{aligned} \quad (4.36)$$

When data are well separated, we can assume that $t = (\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i)$ is

independent of $\mathbf{u} = \frac{\boldsymbol{\Sigma}_i^{-1/2}(\mathbf{x} - \boldsymbol{\mu}_i)}{\sqrt{t}}$. Therefore, (4.36) becomes

$$\begin{aligned} & 2 \cdot \mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t] \cdot \mathbb{E}[\mathbf{u}^T \boldsymbol{\Sigma}_i^{-\frac{1}{2}} F \mathbf{u}] \\ & + 2 \cdot \mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t] \cdot \mathbb{E}[\mathbf{u}^T \boldsymbol{\Sigma}_i^{-\frac{1}{2}} F \mathbf{u}] + \mathbb{E}[\xi_i(\mathbf{x})\psi_i(t)] \boldsymbol{\Sigma}_i^{-\frac{1}{2}} F \\ & = \xi_i(\mathbf{x}_0)\psi_i(t_0)\boldsymbol{\Sigma}_i^{-\frac{1}{2}}(\mathbf{x}_0 - \boldsymbol{\mu}_i), \end{aligned} \quad (4.37)$$

which yields $\mathbb{E}[\mathbf{u}^T \boldsymbol{\Sigma}_i^{-\frac{1}{2}} F \mathbf{u}] = \frac{1}{m} \boldsymbol{\Sigma}_i^{-\frac{1}{2}} F$. Thus, we arrive at

$$F = \frac{\xi_i(\mathbf{x}_0)\psi_i(t_0)(\mathbf{x}_0 - \boldsymbol{\mu}_i)}{\frac{2}{m} \mathbb{E}[(\xi_i(\mathbf{x}) - \xi_i^2(\mathbf{x}))\psi_i^2(t)t] + \frac{2}{m} \mathbb{E}[\xi_i(\mathbf{x})\psi_i'(t)t] + \mathbb{E}[\xi_i(\mathbf{x})\psi_i(t)]}. \quad (4.38)$$

This completes the proof.

4.6 Summary

We have proposed a general framework for a systematic analysis and universal optimisation of EMMs, and have conclusively demonstrated that this equips EMMs with significantly enhanced flexibility and ease of use in practice. In addition to the general nature and the power of the proposed universal framework for EMMs, we have also verified both analytically and through simulations, that this provides a reliable and robust statistical tool for analysing the EMMs. Furthermore, we have proposed a general solver which consistently attains the optimum for general EMMs. Comprehensive simulations over both synthetic and real-world datasets validate the proposed framework, which is fast, stable and flexible.

Chapter 5

Solving General EMMs through a Relaxed Wasserstein Manifold

Compared to the widely adopted Kullback–Leibler (KL) divergence in estimating Gaussian mixture models (GMMs) and even the elliptical mixture models (EMMs) in Chapter 4, we show in this chapter that the Wasserstein distance provides a more desirable optimisation space, which provides a stable solution to the EMMs that is both robust to initialisations and reaches a superior optimum by adaptively optimising along a manifold of a relaxed Wasserstein distance. To this end, we first provide a unifying account of computable and identifiable EMMs, which serves as a basis to rigorously address the underpinning optimisation problem. Due to a probability constraint, solving this problem is extremely cumbersome and unstable, especially under the Wasserstein distance. To relieve this issue, we introduce an efficient optimisation method on a statistical manifold defined under the relaxed Wasserstein distance, which allows for explicit metrics and computable operations, thus significantly stabilising and improving the EMM estimation. An adaptive method is further proposed to accelerate the convergence. Experimental results demonstrate the excellent performance of the proposed EMM solver.

5.1 Introduction

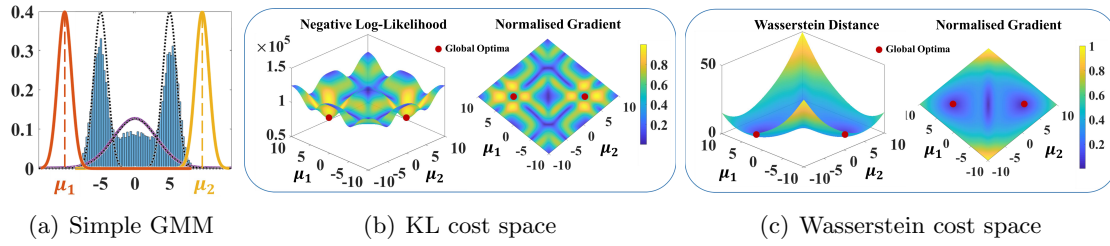


Figure 5.1: Illustration of a simple one-dimensional GMM with 3 clusters. (a): The ground-truth of clusters is plotted via the black dotted lines; Parametric models $\rho(\theta)$ are denoted as colourful lines; We set ALL the parameters θ of the GMM to the ground-truth EXCEPT the mean values of only two clusters, i.e., μ_1 and μ_2 . (b): The KL divergence and the normalised gradient versus the variance of μ_1 and μ_2 . (c): The Wasserstein distance and the normalised gradient versus the variance of μ_1 and μ_2 .

This chapter establishes a general solution to the finite mixture model problem, which has been attracting extensive research effort for decades, due to both its simple representation and potential for universal approximation on arbitrary distributions in $\mathbb{R}^m$. The finite mixture model also provides interpretable and statistical descriptions of data, which makes it a popular choice in a wide range of statistical learning paradigms, such as semi-supervised learning, capsule networks, and various image processing paradigms (e.g., de-noising, matching and registration).

The estimation on a finite mixture model boils down to a minimisation problem which considers the mixture of distributions as a parametric model $\rho(\theta)$, which is then optimised through a minimisation of a certain discrepancy measure between $\rho(\theta)$ and empirical distributions of observed data ρ^* , namely, $\min_{\theta} d(\rho(\theta), \rho^*)$. This minimisation, although not explicitly stated, is a constrained problem because $\rho(\theta)$ must maintain the property of a probability density throughout, to ensure that $d(\cdot, \cdot)$ is tractable.

Due to this probability constraint, various advanced numerical algorithms (solvers) have been typically restricted by either the requirement of an increasingly flexible $\rho(\theta)$ or a powerful $d(\cdot, \cdot)$. Such restrictions, for example, are one of the main rationales for using the expectation-maximisation (EM) algorithm to minimise the KL

divergence in Gaussian mixture model (GMM) problems [36]. On the other hand, gradient-based numerical algorithms typically rest upon additional techniques that only work in particular situations (e.g., gradient reduction [54], positive definite projection [58], re-parametrisation [55] and Cholesky decomposition [56]). Besides the GMM, there exist other solutions that allow for flexible choices of $\rho(\boldsymbol{\theta})$, which still belong to the EM-type methods (e.g., mixtures of t-distributions [40], Laplace distributions [44] and hyperbolic distributions [45]). Unfortunately, given other suitable candidates of distributions, those EM-type methods cannot ensure universal convergence [38, 49, 50], which dramatically limits the power of finite mixture models.

Another issue that has been highlighted in the literature is the sensitivity to initialisations when solving GMMs [148, 169]. One of the main reasons is due to the use of KL divergence, which operates based on a “bin-to-bin” comparison between two density histograms. This means that mixtures which fall into a spurious local minimum cannot be corrected via the points outside. Indeed, with random initialisations for the GMM, Jin has proved that the EM algorithm or any other first-order method which minimises the KL divergence are highly likely to result in arbitrary bad local minima [148]. This can also be easily verified from the non-smooth optimization space with various local optimum of the KL divergence as illustrated in Fig. 5.1-(b), even for estimating a simple GMM (Fig. 5.1-(a)). The gradients on the space are highly concentrated as well, which also leads to ill-posed gradient descent.

On the other hand, by virtue of the reflection of sample space [98] within the Wasserstein distance¹ [22] which employs a “cross-bin” comparison, many practical benefits may be achieved in learning tasks [4]. This property is particularly appealing in mixture model problems, where it ideally provides a comprehensive distance measure over all possible transport plans. The optimization space of the Wasserstein distance is also shown in Fig. 5.1-(c), of which smoothness is witnessed. Basically, there only exist global optimum in this case. The gradients on this space are also well behaved, which promises to achieve superior optimisation. Most recently,

¹Throughout this chapter, the term Wasserstein distance refers to the square-Wasserstein distance.

Kolouri *et al.* [170] have adopted the Wasserstein distance for solving GMM problems. However, the aforementioned probability constraint enforces an extremely small stepsize (learning rate) during optimisation. Given the random projections of the sliced Wasserstein distance, this setup leads to extremely slow convergence or even non-convergence (especially in high dimensions as shown in our experiments). At the end of their optimisation, the EM algorithm is still needed to stabilise the algorithm.

Motivations and Contributions: Despite extensive benefits, optimising the GMM under the Wasserstein distance is still not feasible due to the probability constraint. To address this from a different perspective, we resort to the statistical manifold. We should further point out that our work is different from information geometry [23], as directly establishing a manifold in the whole density space of mixture models is absolutely intractable and cumbersome in optimisation [171]. Another problem within the Wasserstein space is that the geodesic between two mixture models may not lie in mixture models of the same type [172], which leads to non-convergence.

We propose to resolve this problem by introducing an approximation to the Wasserstein distance, followed by establishing a statistical manifold via the so induced distance, which exhibits the desirable property of being complete within mixture models. The subsequent optimisation along this manifold intrinsically satisfies the probability constraint and ensures that the solution resides in the same mixture models. More importantly, minimising the induced distance is shown to be truly reducing the discrepancy between two mixture models, unlike most existing solutions which are based on the minimisation of the Euclidean distance from the optimal parameters. This ensures fast and stable convergence in optimisation. By realising that the existing Riemannian adaptive algorithms only make sense in updating vector parameters, we further develop a novel accelerated stochastic gradient descent method for updating the positive definite matrices.

In this way, our proposed framework makes it possible to incorporate a broad family of distributions, $\rho(\boldsymbol{\theta})$, including an important class of multivariate analysis tech-

niques called elliptical distributions [16] and to further investigate the mixture family termed the elliptical mixture model (EMM). We therefore provide computable and identifiable EMMs in a unified way, which demonstrates that EMMs are quite general and flexible and include the GMMs as special cases [16].

Overall, this chapter proposes a complete and efficient framework for solving general EMM problems, by establishing a statistical manifold under an relaxed Wasserstein distance which promotes stability and efficiency, together with an adaptive stochastic gradient algorithm to further accelerate the optimisation². Compared to the existing literature on mixture problems, the proposed solution achieves consistently superior performance not only in the GMM problems but also for general EMM problems. Our contributions can be summarised as follows:

- A unified framework for dealing with computable and identifiable EMMs, which introduces a rich choice of candidates for flexible finite mixture models.
- Establishment of the statistical manifold through the proposed relaxed Wasserstein distance, which provides explicit and complete operations within the manifold.
- An adaptive accelerated Riemannian gradient descent algorithm on the established manifold, to improve the optimisation and accelerate convergence.

5.2 Computable and Identifiable EMMs

Elliptical distributions include a wide range of standard distributions, and it therefore comes as no surprise that a unified summary of computable candidates as components in the EMMs is a prerequisite to problem definition and subsequent solutions. A classical summary can be found in Chapter 3 in [16]; however, despite progress this framework is not general enough as various elliptical distributions are still missing, and more importantly, it involves complicated representations for each

²The code of this chapter is available at https://github.com/ShengxiLi/wass_emm

type of elliptical distributions. The existing literature also employs different notations and formulations of particular distributions, which may lead to confusion. To this end, we provide a simple and unified framework for summarising the existing computable elliptical distributions via the stochastic representation, in a unifying notation that is provided in Chapter 2.1.2 of this thesis, which can then be used to constitute flexible and identifiable EMMs.

Due to the fact that the $\mathcal{R}^2$ decides the type of elliptical distributions, we here provide a unified summary of elliptical distributions in Table 5.1; this is achieved through stochastic representations in (2.16). This makes it possible to avoid complicated formulations, and to instead classify different categories simply through several typical distributions of $\mathcal{R}^2$, which also allows for simple and intuitive sample generations for elliptical distributions. The proof for expressions in this table is provided in Appendix-5.6.1. Uniquely, this further clarifies the commonalities between the members of the elliptical family of distributions. More importantly, constructing an EMM with the candidates in Table 5.1 can be easily proved to be identifiable based on Theorem 2 in [138]. It is thus convenient and safe to establish a well-defined EMM by the candidates summarised in Table 5.1.

5.3 Statistical Manifold towards EMMs

As the Wasserstein distance possesses a Riemannian structure [98], it is then natural to treat each EMM as a “point” in the manifold, whereby the metric is defined by the Hessian of the Wasserstein distance. However, as pointed out by Chen [172], in the Wasserstein space, the geodesic between two points, i.e., two GMMs, does not necessarily belong to the GMM of the same type. A way to solve this problem is to pull back from the whole density space to the parametric space [172]. This treatment, nevertheless, is still a toy solution due to two main considerations. The first deficiency is that the metric does not have a closed-form representation, and needs to be numerically obtained for every possible value in advance before the

Table 5.1: $\mathcal{R}^2 \leftrightarrow$ Computable elliptical distributions

Types	$\mathcal{R}^2$	$\leftrightarrow$	$C_m \cdot g(t)$	Typical Multivariate Dist.
Kotz Type [16]	$\mathcal{R}^2 \stackrel{=}{=}^d \mathcal{G}^{1/s},$ $\mathcal{G} \sim \text{Ga}(\frac{2a+m-2}{2s}, b)$ $a > 1 - \frac{m}{2}, b, s > 0$	$= \left(\frac{\Gamma(m/2) s b^{2a+m-2}/(2s)}{\Gamma((2a+m-2)/2s) \pi^{m/2}} \right) t^{a-1} \exp(-bt^s)$	Gamma: $s = 1$ Weibull: $a = s$ Generalised Gaussian: $a = 1$ Gaussian: $a = 1, s = 1, b = \frac{1}{2}$	
Scale Mixture of Normals	Pearson Type VII [16] $\mathcal{K}^{-1} \sim \text{Ga}(s - \frac{m}{2}, \frac{v}{2}),$ $v > 0, s > m/2$	$= \left(\frac{(\pi v)^{-m/2} \Gamma(s)}{\Gamma(s - m/2)} \right) (1 + t/v)^{-s}$	T -dist.: $s = \frac{m+v}{2}$ Cauchy: $v = 1, s = \frac{m+1}{2}$	
$\mathcal{R}^2 \stackrel{=}{=}^d \mathcal{G} \cdot \mathcal{K},$ $\mathcal{G} \sim \text{Ga}(\frac{m}{2}, \frac{1}{2}),$ $\mathcal{K}$ has different dist.	Hyperbolic Type [45] $\mathcal{K} \sim \text{GIG}(v, a, \lambda)$ $v, a > 0, \lambda \in \mathbb{R}$ Other Types [139, 173] $\sqrt{\mathcal{K}} \sim \partial \text{Kov}(\frac{\mathcal{K}}{2})/\partial \mathcal{K}$	$= \left(\frac{(v/a)^{\lambda/2}}{(2\pi)^{m/2} \text{BeK}_{\lambda}(\sqrt{av})} \right) \frac{\text{BeK}_{\lambda}(\lambda - m/2) (\sqrt{av + vt})}{(\sqrt{a/v + t/v})^{m/2 - \lambda}}$ $= \exp(-t)/(1 + \exp(-t))^2$	Inverse-Gaussian: $\lambda = -1/2$ K -dist. [162]: $a \rightarrow 0, \lambda > 0$ Laplace: $a \rightarrow 0, \lambda = 1, v = 2$	
Pearson Type II	$\mathcal{K} \sim \text{SaS}(\frac{a}{2}), a \in (0, 2)$ $\mathcal{R}^2 \sim \text{Beta}(m/2, s), s > 1$	$\propto \text{SaS}(a)$ $= \left(\frac{\Gamma(m/2 + s)}{\pi^{m/2} \Gamma(s)} \right) (1 - t)^{s-1}, t \in [0, 1]$	Logistic α -stable [16]	
Notations:	$a, b, s, v, \lambda, \alpha$ are adjustable parameters for different types of dist.; $\mathcal{G}$ and $\mathcal{K}$ are random variables related to $\mathcal{R}^2$; m is the dimension. Gamma dist.: $\text{Ga}(x, y) = y^x t^{x-1} \exp(-yt)/\Gamma(x)$; Inverse Gaussian dist.: $\text{GIG}(x, y, z) = \frac{(x/y)^{z/2}}{2 \text{BeK}_z(\sqrt{xy})} t^{z-1} \exp(-\frac{xt^2 + y}{2t})$; Kolmogorov-Smirnov dist.: $\text{Kov}(x) = 1 - 2 \sum_{n=1}^{\infty} (-1)^{n+1} \exp(-2n^2 x^2)$; Beta dist.: $\frac{\Gamma(x+y)}{\Gamma(x)\Gamma(y)} t^{x-1} (1 - t)^{y-1}$; $\text{SaS}(a)$: the symmetric α -stable dist. with index a ; $\text{BeK}_x(y)$: the Bessel function of the third kind; $\Gamma(x)$: the Gamma function.			

optimisation. This is highly prohibitive, especially for multivariate cases due to the curse of dimensionality. Besides computational intractability, operations such as the exponential mapping and the vector transport in the manifold cannot be well defined as there is typically a second-order differential equation involved to obtain those operations.

We thus propose a relaxed Wasserstein distance between two EMMs as a means to define a well-behaved manifold for the EMM problems; this is achieved by the property that the Wasserstein distance of two elliptical distributions is completely and explicitly defined. We then provide the Riemannian metric for the EMM problems according to the defined distance.

5.3.1 Relaxed Wasserstein Distance between EMMs

We now focus on the distance between two EMMs $\mathcal{Y}_1$ and $\mathcal{Y}_2$, and propose a relaxed Wasserstein distance by treating each distribution within an EMM as a “super-point” and defining a transport-like distance between those “super-points”. A rigorous definition is given as follows.

Definition 5.1. *Given two EMMs $\mathcal{Y}_1$ and $\mathcal{Y}_2$, a discrepancy measure is defined as*

$$d_U(\mathcal{Y}_1, \mathcal{Y}_2) = \min_{\gamma(i,j)} \left(\sum_{i,j} \frac{\gamma(i,j)}{k} d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) + \arccos \left(\sum_{i,j} \gamma(i,j) \sqrt{\pi_{i,1} \pi_{j,2}} \right) \right), \quad (5.1)$$

where $d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ is the Wasserstein distance between the elliptical distributions $\mathbf{x}_{i,1}$ and $\mathbf{x}_{j,2}$. $\gamma(i, j)$ is binary $\in \{0, 1\}$; for each i and j , $\gamma(i, j)$ satisfies $\sum_{i=1}^k \gamma(i, j) = 1$ and $\sum_{j=1}^k \gamma(i, j) = 1$.

Theorem 5.1. *Given two EMMs $\mathcal{Y}_1$ and $\mathcal{Y}_2$, the discrepancy measure $d_U(\mathcal{Y}_1, \mathcal{Y}_2)$ defines a distance.*

Proof. Please see Appendix-5.6.2. □

We provide our intuitions of defining the distance of (5.1). The second term $\arccos(\sum_{i,j} \gamma(i,j) \sqrt{\pi_{i,1} \pi_{j,2}})$ intrinsically relates to the probability constraint given by $\sum_{i=1}^k \pi_{i,1} = \sum_{j=1}^k \pi_{j,2} = 1$, while $\gamma(i,j)$ operates as a bijection between mixture components in $\mathbf{y}_1$ and $\mathbf{y}_2$. $\sum_{i,j} \frac{\gamma(i,j)}{k} d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ can thus be regarded as a discrete transport between k uniformly distributed “super-points” $\mathbf{x}_{i,1}$ and $\mathbf{x}_{j,2}$, for which their cost is defined as $d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$. We also noticed a computational distance proposed by Chen *et. al* [174]. Our metric can be further seen as a restriction (or upper bound) of the Chen’s metric because we operate via a one-to-one transport whilst the Chen’s metric admits arbitrary transport plans. However, the most advantageous property of our metric is that it defines an explicit manifold over EMMs, which will be introduced shortly.

More importantly, $d_U(\mathbf{y}_1, \mathbf{y}_2)$ comprehensively reflects the discrepancy between $\mathbf{y}_1$ and $\mathbf{y}_2$ via a summation operation; this means $d_U(\mathbf{y}_1, \mathbf{y}_2) = 0$ if and only if both the difference between mixture components $\mathbf{x}_{i,1}$ and $\mathbf{x}_{j,2}$ and the difference between latent variables $\pi_{i,1}$ and $\pi_{j,2}$ equal to 0. The following lemma further proves that for balanced EMMs, $d_U(\mathbf{y}_1, \mathbf{y}_2)$ is an upper bound of the Wasserstein distance $d_W^2(\mathbf{y}_1, \mathbf{y}_2)$.

Lemma 5.1. *Given two balanced EMMs $\mathbf{y}_1$ and $\mathbf{y}_2$ (i.e., $\pi_{i,1} = \pi_{j,2} = 1/k$ for all i, j), $d_U(\mathbf{y}_1, \mathbf{y}_2)$ is an upper bound of the Wasserstein distance:*

$$d_W^2(\mathbf{y}_1, \mathbf{y}_2) \leq d_U(\mathbf{y}_1, \mathbf{y}_2). \quad (5.2)$$

The equality holds when $k = 1$.

Proof. Please see Appendix-5.6.3. □

5.3.2 Statistical Manifold for EMM Problems

Before introducing the statistical manifold for EMMs, we would like to give credit to several most recent works on Gaussian distributions and elliptical distributions,

which lay a basis of our metric proposed in this section. We would like to mention [175] of the Wasserstein distance of Gaussian measures and [176] for the elliptical distributions. Most recently, the Riemannian manifold for Gaussian distributions has been established in an explicit form [177, 178]. Then, on the basis of the relaxed Wasserstein distance, we provide the Riemannian metric of the EMM problems by calculating the Hessian of the proposed distance in Definition 5.1 as follows.

Lemma 5.2. *The relaxed Wasserstein distance $d_U(\mathcal{Y}_1, \mathcal{Y}_2)$ represents an explicit Riemannian metric in the parametric space, and the corresponding Riemannian manifold is a product manifold of $\mathbb{R}^k \times \prod_{i=1}^k (\mathbb{R}^m \times \mathbb{P})$, where $\mathbb{P}$ is the $m \times m$ symmetric positive semi-definite manifold:*

- 1) *The manifold for the square root of π_i , i.e., $[\sqrt{\pi_1}, \sqrt{\pi_2}, \dots, \sqrt{\pi_k}]^T$ is a sphere manifold of $\mathbb{R}^k$.*
- 2) *The manifold for μ_i is the Euclidean space of $\mathbb{R}^m$.*
- 3) *The manifold for Σ_i , i.e., $\mathbb{P}$, is defined by*

$$ds^2 = \frac{\mathbb{E}[\mathcal{R}^2]}{m} (\mathbb{L}_{\Sigma_i}[d\Sigma]) \Sigma_i (\mathbb{L}_{\Sigma_i}[d\Sigma]). \quad (5.3)$$

In (5.3), $\mathbb{L}_{\mathbf{A}}[\mathbf{C}] = \mathbf{B}$ is a Lyapunov operator: $\mathbf{A}^T \mathbf{B} + \mathbf{B} \mathbf{A} = \mathbf{C}$, where $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{P}$ [179]. More importantly, the sectional curvature is non-negative ($= m/\mathbb{E}[\mathcal{R}^2] k_G$ where $k_G \geq 0$ is the sectional curvature for Gaussian cases [177]). Recall that $\mathcal{R}$ is defined in (2.16).

Proof. Please see Appendix-5.6.4. □

Remark 5.1. *The metric in (5.3) provides a manifold for positive definite matrices. Compared to the best known manifold belonging to the Hadamard manifold (non-positive sectional curvature) [180], the newly developed manifold provides an example of non-negative manifolds. It is actually further stated that an Alexandrov space has the non-negative curvature iff it is a Wasserstein space [181]. The established metric*

Table 5.2: Basic operations of the manifold in Lemma 5.2

For the h -th iteration		
	$\nabla_U(\cdot)$	$\text{Exp}(-\alpha \nabla_U(\cdot))$
$\sqrt{\pi}^h$	$\nabla_E(\sqrt{\pi}^h) - (\sqrt{\pi}^h)^T \nabla_E(\sqrt{\pi}^h) \cdot \sqrt{\pi}^h$	$\cos(\ \alpha \nabla_U(\sqrt{\pi}^h)\ _2) \sqrt{\pi}^h - \frac{\sin(\ \alpha \nabla_U(\sqrt{\pi}^h)\ _2)}{\ \nabla_U(\sqrt{\pi}^h)\ _2} \nabla_U(\sqrt{\pi}^h)$
μ_i^h	$\nabla_E(\mu_i^h)$	$\mu_i^h - \alpha \nabla_U(\mu_i^h)$
Σ_i^h	$\nabla_E(\Sigma_i^h) \Sigma_i^h + \Sigma_i^h \nabla_E(\Sigma_i^h)$	$(\mathbb{I}_{\Sigma_i^h} [-\alpha \nabla_U(\Sigma_i^h)] + I) \Sigma_i^h (\mathbb{I}_{\Sigma_i^h} [-\alpha \nabla_U(\Sigma_i^h)] + I)$
For the Radon transform $\mathbf{p} \in \mathbb{S}^{m-1}$, specified in the EMM problems:		
$\nabla_E(\sqrt{\pi}^h)$	The i -th dimension of $\nabla_E(\sqrt{\pi}^h)$ is $2 \int_{\mathbb{R}} c_m \phi(y) \sqrt{\pi_i^h} (\mathbf{p}^T \Sigma_i^h \mathbf{p})^{-1/2} g(\frac{(y - \mathbf{p}^T \mu_i^h)^2}{\mathbf{p}^T \Sigma_i^h \mathbf{p}}) dy$	
$\nabla_E(\mu_i^h)$	$= \left(-2 \int_{\mathbb{R}} c_m \phi(y) \pi_i^h (\mathbf{p}^T \Sigma_i^h \mathbf{p})^{-3/2} g'(\frac{(y - \mathbf{p}^T \mu_i^h)^2}{\mathbf{p}^T \Sigma_i^h \mathbf{p}}) (y - \mathbf{p}^T \mu_i^h) dy \right) \mathbf{p}$	
$\nabla_E(\Sigma_i^h)$	$= \left(- \int_{\mathbb{R}} c_m \phi(y) \pi_i^h (\mathbf{p}^T \Sigma_i^h \mathbf{p})^{-3/2} \left(\frac{1}{2} g(\frac{(y - \mathbf{p}^T \mu_i^h)^2}{\mathbf{p}^T \Sigma_i^h \mathbf{p}}) + g'(\frac{(y - \mathbf{p}^T \mu_i^h)^2}{\mathbf{p}^T \Sigma_i^h \mathbf{p}}) \frac{(y - \mathbf{p}^T \mu_i^h)^2}{\mathbf{p}^T \Sigma_i^h \mathbf{p}} \right) dy \right) \mathbf{p} \mathbf{p}^T$	
$\phi(y)$ is the Kantorovich potential [172]. $\nabla_E(\cdot)$ denotes the Euclidean gradient with regard to $\sqrt{\pi}^h$, μ_i^h and Σ_i^h .		

thus provides a desirable reflection on the curvature when dealing with Wasserstein related EMM problems.

5.4 Adaptively Accelerated Optimisation

By virtue of optimising on a statistical manifold, the probability constraint can be satisfied automatically; this allows us to incorporate various numerical algorithms when solving the EMM problems. More specifically, we first show that the constrained minimisation problem can be transformed to an unconstrained one when restricted on the statistical manifold, which results in the “vanilla” gradient descent on the manifold. We then propose an adaptively accelerated solver.

5.4.1 Vanilla Gradient Descent on Statistical Manifold

Similar to [182], for gradient descent methods, the way for gradient descent in our work is given by,

$$\begin{aligned}
 \mathcal{Y}_{\theta+\Delta\theta}^* &= \arg \min_{\mathcal{Y}_{\theta+\Delta\theta}} d_{SW}(\mathcal{Y}_{\theta+\Delta\theta}, \mathcal{Y}^*) + \frac{d_U(\mathcal{Y}_{\theta}, \mathcal{Y}_{\theta+\Delta\theta})}{c^2}, \\
 &\rightarrow \Delta\theta^* = \text{Exp}(-\alpha \nabla_U(\theta))
 \end{aligned} \tag{5.4}$$

where $\Delta\boldsymbol{\theta}$ is the step size for the next iteration and c^2 denotes a sphere of all realisable distributions, which ensures searching for optimal $\Delta\boldsymbol{\theta}^*$ without being slowed down by the curvature; $d_{SW}(\mathcal{Y}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}, \mathcal{Y}^*)$ denotes the sliced Wasserstein distance [183] between the parametric mixture model $\mathcal{Y}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}$ and the observed samples $\mathcal{Y}^*$. By relaxing $d_U(\mathcal{Y}_{\boldsymbol{\theta}}, \mathcal{Y}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}})$ with its Hessian term (i.e., the inner product on the tangent space), we naturally obtain an unconstrained Riemannian gradient descent of the second row of (5.4), where $\nabla_U(\boldsymbol{\theta})$ is the Riemannian gradient on the statistical manifold defined by $d_U(\cdot, \cdot)$; α is the stepsize of each iteration; $\text{Exp}(-\alpha\nabla_U(\boldsymbol{\theta}))$, called exponential mapping, projects the step movement $-\alpha\nabla_U(\boldsymbol{\theta})$ from the tangent space to the statistical manifold along geodesics [103], which ensures the probability constraint. On the other hand, when the proposed distance $d_U(\mathcal{Y}_{\boldsymbol{\theta}}, \mathcal{Y}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}})$ is changed to the Euclidean distance, (5.4) is the trivial gradient descent [170]. However, as mentioned above, this gradient descent does not satisfy the probability constraint nor does it reflect the probability space curvature, which leads to inefficient and unstable optimisation.

In (5.4), the sliced Wasserstein distance [183] provides a feasible solution in solving the semi-discrete Wasserstein problem, i.e., $\mathcal{Y}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}$ is continuous and $\mathcal{Y}^*$ is the sum of Dirac masses [184]. It uses unit random projections to turn the original problem to a one-dimensional Wasserstein problem via the Radon transform [183], so that a closed-form solution can be obtained³. More importantly, as the sliced Wasserstein distance is composed of a set of random projections [183], it is then natural to implement a stochastic gradient descent on the Riemannian manifold for each random projection (denoted as $\mathbf{p} \in \mathbb{S}^{m-1}$). The random projection process also allows for parallel gradient descent. We provide the basic operations of problem in (5.4) in Table 5.2, while the details are provided in Appendix-5.6.5.

³It needs to be pointed out that other relaxed distances (e.g., entropy relaxed Wasserstein distances) for the semi-discrete problem can also be seamlessly adopted in (5.4) of our work.

Algorithm 2: Riemannian adaptively accelerated manifold optimisation

input: n observed samples $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$; stepsize $\{\alpha^h\}_{h=1}^H$; hyper-parameters $\{\beta_1^h\}_{h=1}^H$ and β_2 ;

initialisation: 1st-order moment $\{\mathbf{u}_i^0\}_{i=1}^k$; 2nd-order moment $\{\mathbf{v}_i^0\}_{i=1}^k$;

for $h = 1$ **to** H **do**

Random projection: $\mathbf{p} \in \mathbb{S}^{(m-1)}$

Update $\sqrt{\boldsymbol{\pi}}^{h+1} = \text{Exp}(-\alpha \nabla_U(\sqrt{\boldsymbol{\pi}}^h))$

for $i = 1$ **to** k **do**

Update $\boldsymbol{\mu}_i^{h+1} = \text{Exp}(-\alpha \nabla_U(\boldsymbol{\mu}_i^h))$

Update $\boldsymbol{\Sigma}_i^h$ by the *Dadam*:

$\mathbf{u}_i^h = \beta_1^h \varphi_{\boldsymbol{\Sigma}_i^{h-1} \rightarrow \boldsymbol{\Sigma}_i^h}(\mathbf{u}_i^{h-1}) + (1 - \beta_1^h) \nabla_U(\boldsymbol{\Sigma}_i^h)$

$\mathbf{v}_i^h = \beta_2 \mathbf{v}_i^{h-1} + (1 - \beta_2) \nabla_E(\boldsymbol{\Sigma}_i^h) \nabla_E(\boldsymbol{\Sigma}_i^h)^T$

$\text{adp}_i^h = \max\{\mathbf{p}^T \mathbf{v}_i^h \mathbf{p}, \text{adp}_i^{h-1}\}$

$\boldsymbol{\Sigma}_i^{h+1} = \text{Exp}(-\alpha^h \mathbf{u}_i^h / \sqrt{\text{adp}_i^h})$

output: $\boldsymbol{\pi}^H, \{\boldsymbol{\mu}_i^H\}_{i=1}^k, \{\boldsymbol{\Sigma}_i^H\}_{i=1}^k$

5.4.2 Adaptively Accelerated Algorithm

On the basis of several accelerated Riemannian stochastic gradient descent methods which adopt the first-order moment information [107], recently Becigneu and Ganea [185] further proposed a Riemannian adaptive method by employing the second-order moments. Although this adaptive stochastic gradient descent method [185] may be incorporated to our work to further improve the convergence, we argue that it only makes sense for updating vector parameters, i.e., $\boldsymbol{\pi}^h$ and $\boldsymbol{\mu}_i^h$, because in this case the second-order moment can be calculated element-wise by a decomposition into product manifolds [185]. This is similar to the *Adam* algorithm in the Euclidean space [186]. When updating the matrix parameter, $\boldsymbol{\Sigma}_i^h$, however, it does not capture the second-order information properly due to the direct accumulation of the second-order moments over the whole matrix manifold; this slows down convergence in the optimisation.

The key difficulty for the matrix case is that it is meaningless to accumulate second-order moments in an element-wise manner due to the structure within the matrix. We here view the second-order moments of matrices from another perspective, by realising that a positive definite matrix can be decomposed into a set of eigenvectors and the corresponding eigenvalues. The eigenvectors can be regarded as a set

of projection directions and the eigenvalues are scalars that can be connected with the accumulation in those directions. Furthermore, we can see from Table 5.2 that the Euclidean gradient $\nabla_E(\Sigma_i^h)$ consists of a scalar weight multiplied by a rank-1 matrix characterised by the direction $\mathbf{p}\mathbf{p}^T$. Therefore, instead of the element-wise accumulation, we propose a direction-wise accumulation of second-order moments to adaptively adjust the stepsize when updating the matrix. The details of our algorithm in Algorithm 2, with our directional adaptive accelerated method (*Dadam*) of updating Σ_i^h being achieved by $\mathbf{v}_i^h$ and adp_i^h .

Moreover, as there is no explicit parallel transport in the Wasserstein space, we propose a new means of vector transport $\varphi_{\Sigma_i^{h-1} \rightarrow \Sigma_i^h}(\mathbf{u}_i^{h-1}) = \mathbb{L}_{\Sigma_i^{h-1}}[\mathbf{u}_i^{h-1}]\Sigma_i^h + \Sigma_i^h \mathbb{L}_{\Sigma_i^{h-1}}[\mathbf{u}_i^{h-1}]$ to accumulate the first-order moments.

As for the convergence analysis, because our goal is to minimise the Wasserstein distance, qualitatively, our manifold, defined by the Hessian of the relaxed Wasserstein distance as in Lemma 5.2, implicitly involves the second-order information and accelerates the convergence. Furthermore, adp_i^h denotes the accumulation in the current projection $\mathbf{p}$, which is a scalar (for computational ease) and does not need any eigen-decomposition operations. This, on the one hand, avoids the high computational complexity of calculating any inverse of matrices, making our method the same computational complexity as that of [185] at each iteration. On the other hand, it also makes our convergence analysis similar to that of [185], upon realising that the sectional curvature is automatically bounded from below; we omit the analysis here.

5.5 Experimental Results

We evaluated the effectiveness of our manifold and the proposed *Dadam* on both synthetic data and image data, by employing 4 EMMs, i.e., mixtures of *Gaussian*, *Logistic*, *Cauchy* and *Gamma* ($s = 1, a = 2, b = 0.5$ in Table 5.1).

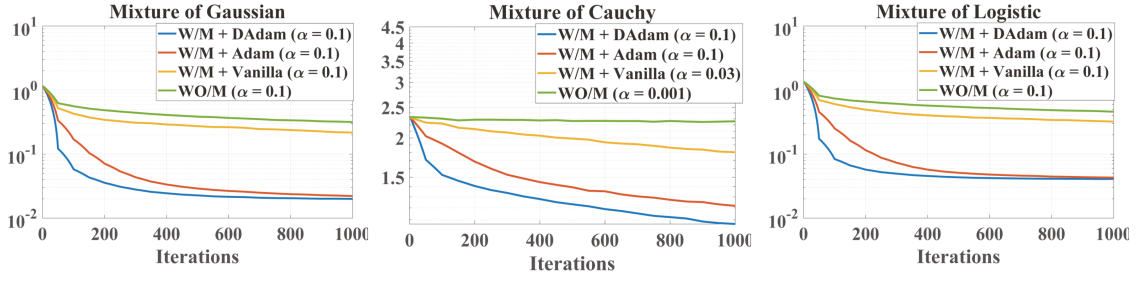


Figure 5.2: The Wasserstein distance against the number of iterations for the four considered algorithms, averaged over the three types of datasets ($\{m, k\} = \{2, 3\}, \{8, 9\}, \{16, 27\}$). The best learning rate is shown in the legend. 1000 iterations are for illustration purpose. The computational time per iteration is discussed in Table 5.3.

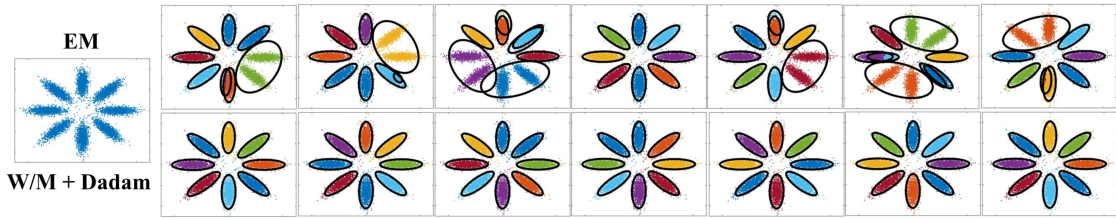


Figure 5.3: The optimised GMMs via the EM and our ($W/M+Dadam$) methods on a flower-shaped synthetic data ($n = 10,000$) with random initialisations. Each cluster in the flower-shaped data is composed by Gaussian-distributed samples.

Table 5.3: Comparisons among our, WO/M and EM methods for GMMs by varying m and k .

	$m = 2, k = 3$			$m = 8, k = 9$			$m = 16, k = 27$		
	Wass	NLL	Time / Ite.	Wass	NLL	Time / Ite.	Wass	NLL	Time / Ite.
W/M + Dadam	0.01 ± 0.00	5.10 ± 0.00	4.35ms	0.03 ± 0.03	19.49 ± 0.03	9.83ms	2.39 ± 0.33	44.92 ± 1.04	32.51ms
WO/M	0.03 ± 0.00	5.12 ± 0.00	4.23ms	4.26 ± 3.61	22.27 ± 1.69	9.64ms	5e3 ± 589	Inf	29.77ms
EM	0.65 ± 1.06	5.30 ± 0.35	3.10ms	0.70 ± 0.46	19.83 ± 0.29	13.92ms	2.51 ± 1.27	48.84 ± 1.24	215.00ms

Synthetic Data: Each synthetic dataset contains different mixtures of Gaussian distributions, with 10,000 samples in total. Specifically, we employed three types of synthetic datasets, i.e., $\{m = 2, k = 3\}$, $\{m = 8, k = 9\}$ and $\{m = 16, k = 27\}$. Each type contains 10 randomly generated mixture datasets (the eccentricity ε and the separation c of [165] were equal to 10). For every algorithm, we tested on each dataset over 10 random initialisations and recorded the mean and standard deviation. Then, we averaged over all datasets to obtain the performance of each

Table 5.4: Comparison performance of our, WO/M and IRA algorithms over the BSDS500 dataset via five metrics

	Gaussian Mixture			Logistic Mixture			Cauchy Mixture			Gamma Mixture		
	Our	WO/M	IRA	Our	WO/M	IRA	Our	WO/M	IRA	Our	WO/M	IRA
Wass	35.0 ± 4.68	89.5 ± 10.0	53.6 ± 21.9	45.8 ± 12.7	154 ± 10.9	68.0 ± 24.2	179 ± 15.3	211 ± 7.77	253 ± 65.4	35.5 ± 1.82	510 ± 71.2	–
NLL	11.8 ± 0.01	12.2 ± 0.11	11.8 ± 0.05	10.6 ± 0.02	12.5 ± 0.25	10.6 ± 0.06	12.1 ± 0.01	12.3 ± 0.01	12.0 ± 0.05	13.3 ± 0.07	19.1 ± 0.86	–
PSNR (dB)	18.9 ± 0.07	18.6 ± 0.18	18.3 ± 0.66	18.8 ± 0.11	18.2 ± 0.18	17.8 ± 0.67	19.3 ± 0.05	19.3 ± 0.06	18.4 ± 0.56	21.0 ± 0.08	18.0 ± 0.58	–
SSIM	0.68 ± 0.00	0.65 ± 0.01	0.66 ± 0.03	0.67 ± 0.01	0.62 ± 0.01	0.63 ± 0.03	0.70 ± 0.00	0.70 ± 0.00	0.69 ± 0.02	0.73 ± 0.00	0.59 ± 0.03	–
FailR	0.20%	53.1%	2.98%	0.46%	75.5%	2.68%	1.16%	0.82%	17.16%	0%	2.02%	100%

algorithm.

Image Data: We adopted the MNIST [187] dataset as well as the BSDS500 [166] benchmark dataset in our evaluation. For the MNIST dataset, each image in both the training and testing sets was downsampled to 3×3 and 5×5 ; by vectorising each we obtained two test data with $(n \times m)$ being $(70,000 \times 9)$ and $(70,000 \times 25)$, respectively. The evaluation on the BSDS500 dataset is slightly different from that on the MNIST dataset, which enables to verify our method on some basic tasks such as the image reconstruction. Specifically, instead of modelling over the whole dataset, each image in the dataset was treated as one test data; this means we had 500 test data in the BSDS500 dataset. Furthermore, each test data in both the MNIST and BSDS500 datasets was tested for every algorithm with 10 random initialisations, where the mean and standard deviation were also recorded.

Parameter Settings and Metrics: We found the best learning rates α by searching from $\{0.001, 0.003, 0.01, 0.03, 0.1, 0.3\}$. Similar to [185, 186], $\beta_{11} = 0.9$ and $\beta_2 = 0.999$. The maximum number of iterations for testing the synthesis data was 2,000 and that for the image data was 10,000. The Wasserstein distance (simplified as *Wass*) and averaged negative log-likelihood (*NLL*) were mainly employed for comparison. The optimisation fail ratio (*FailR*) was also reported to show the stability of optimisation under various initialisations. For image data, we adopted two well-known metrics, namely, the peak signal-to-noise ratio (*PSNR*) and structural similarity index (*SSIM*) [168], to evaluate the quality of reconstructed images

via maximising the posterior of optimised EMMs.

Baselines: Each main part of our method was assessed. The vanilla Riemannian gradient descent method on our manifold is denoted by $W/M + \text{Vanilla}$. The Riemannian *Adam* of [185] with our manifold is $W/M + \text{Adam}$. Our proposed *Dadam* is denoted by $W/M + \text{Dadam}$ or *Our* interchangeably. We also performed comparisons with the trivial gradient descent over the sliced Wasserstein distance without the manifold (denoted by WO/M), *which is the basics of [170]*. The EM-type methods were also compared, which are denoted as *EM* for GMMs and *IRA* for other EMMs [38]. All the methods were run on the Matlab 2017a under Intel Core(TM) i7-6700 CPU, where the time was recorded.

5.5.1 Assessment over the *Dadam* on Synthetic Data

We first evaluated the effectiveness of our *Dadam* in adaptively accelerating the convergence. In this part, we only optimised the Σ_i whilst setting the π and μ_i as the ground truth of the synthesis data. Fig. 5.2 illustrates the convergence speed; observe that the optimisation on our established manifold exhibits much faster convergence. Furthermore, both the adaptive methods of [185] and the *Dadam* were shown to boost the speed. The proposed *Dadam* achieved the fastest convergence, which verifies the effectiveness of our *Dadam*.

5.5.2 Overall Comparisons on Synthetic Data

We next compared our algorithm (i.e., $W/M + \text{Dadam}$) with the WO/M and the *EM* methods under GMM problems. Similar results can be found for other EMMs; the result for GMMs is given in Table 5.3. It should be pointed out that all parameters (π , μ_i and Σ_i) were optimised in this test and each result was reported via the corresponding best learning rates. From this table, observe that our algorithm not only achieves an extremely stable estimation (the lowest deviations) but also the lowest values of both the Wasserstein distance and the NLL. An illustrative example

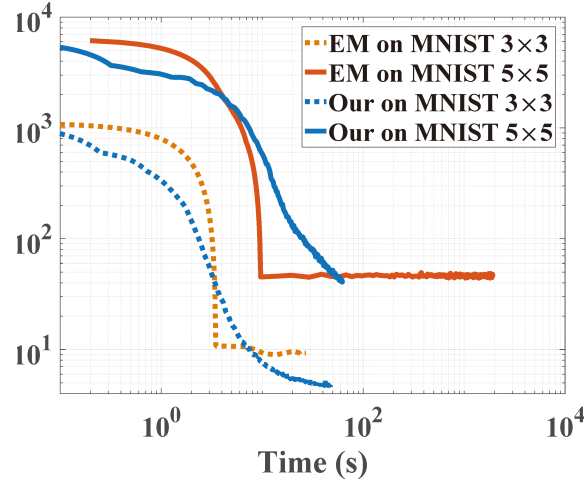


Figure 5.4: Wasserstein distance against the computational time for the *EM* and *Our*, on the downsampled 3×3 and 5×5 MNIST test data. The best learning rate is chosen as $\alpha = 0.1$. The cluster number is chosen as $k = 3$. The maximal iterations are set to 10,000 for both methods.

can be found in Fig. 5.3, in which our algorithm consistently achieves the optimal clustering but the optimised GMMs via the *EM* are highly unstable. From Table 5.3, we should also point out that although our algorithm implements a set of manifold operations, the computational burden over that without manifold optimisation (i.e., *WO/M*) is in average around 5%. However, our algorithm enjoys much faster speed compared to the *EM* algorithm per iteration.

5.5.3 Overall Comparisons on the MNIST

We further compared the computational complexity of our algorithm (*W/M+Dadam*) with the *EM* method, besides the time per iteration reported in Table 5.3. Fig. 5.4 plots the decrease of Wasserstein loss with regard to the computational time when optimising GMMs and the trends for other EMMs are the same. We can see clearly from this figure that our algorithm consistently achieves the lowest Wasserstein cost. The computational time of our algorithm is comparable to that of the *EM* algorithm when $m = 3 \times 3 = 9$; our algorithm, however, requires much less time when in higher dimensions such as $m = 5 \times 5 = 25$. We shall also need to point out that the random projections in our algorithm can be implemented parallelly, which can further

accelerate the convergence speed of our algorithm.

5.5.4 Overall Comparisons on the BSDS500

Finally, we evaluated our algorithm on the BSDS500 dataset, with the results shown in Table 5.4 shows the results. From this table, we see some improvements of using the Wasserstein distance instead of the KL divergence, by comparing the *WO/M* with the *IRA*. However, due to the probability constraint, the *WO/M* is highly unstable and its fail ratio reaches $> 50\%$ for the Gaussian mixture and Logistic mixture. More importantly, by optimising on our established manifold together with our *Dadam* method (denoted as *Our*), we can consistently achieve best performances over the five metrics. Again, our algorithm also achieve highly stable estimation, taking advantages from the Wasserstein distance, compared to the *IRA* which operates under the KL. Another advantage of our algorithm is its universal convergence, while the *IRA* does not converge for the mixture of *Gamma*, which limits the flexibility of EMMs.

5.6 Appendix

5.6.1 Proof for expressions in Table 5.1

Consider the term $p_{\mathcal{R}}(t) = g(t^2) \cdot t^{m-1}$. The pdf of $\mathcal{R}^2$ is then obtained accordingly as,

$$p_{\mathcal{R}^2}(t) = \frac{1}{2} \cdot g(t) \cdot t^{m/2-1}. \quad (5.5)$$

The term $g(t)$ can be further decomposed as $c \cdot g_c(t)$, where $g_c(t)$ is a nuclear term that only relates to t , and c is a normalisation term of $p_{\mathcal{R}}(t)$. This allows us to prove the results in Table 1 related to computable elliptical probability distribution functions (pdf) as follows.

- **Kotz type distributions:** The nuclear term of density generator is $g_c(t) = t^{a-1}\exp(-bt^s)$ and c can be calculated from the condition $\int p_{\mathcal{R}}(t)dt = \int g(t^2) \cdot t^{m-1}dt = 1$ as,

$$c = \left(\int_0^\infty t^{2a+m-3}\exp(-bt^{2s})dt \right)^{-1} = \frac{2sb^{\frac{2a+m-2}{2s}}}{\Gamma(\frac{2a+m-2}{2s})}. \quad (5.6)$$

We now arrive at the pdf of $\mathcal{R}^2$ for the Kotz type distributions, in the form

$$p_{\mathcal{R}^2}(t) = \frac{sb^{\frac{2a+m-2}{2s}}}{\Gamma(\frac{2a+m-2}{2s})} t^{\frac{m}{2}+a-2}\exp(-bt^s). \quad (5.7)$$

However, in practice, it is intractable to generate samples of $\mathcal{R}^2$ as in (5.7). On the other hand, it can be found that $\mathcal{G}^{1/s}$ has the same distribution as $\mathcal{R}^2$ when $\mathcal{G}$ is Gamma distributed. Specifically, when $\mathcal{G} \sim \text{Ga}(\frac{2a+m-2}{2s}, b)$, we obtain the pdf of $\mathcal{G}^{1/s}$ as,

$$p_{\mathcal{G}^{1/s}}(t) = st^{s-1} \frac{b^{\frac{2a+m-2}{2s}} t^{\frac{2a+m-2}{2}-s} \exp(-bt^s)}{\Gamma(\frac{2a+m-2}{2s})} = \frac{sb^{\frac{2a+m-2}{2s}}}{\Gamma(\frac{2a+m-2}{2s})} t^{\frac{m}{2}+a-2} \exp(-bt^s). \quad (5.8)$$

This proves that $\mathcal{R}^2 \stackrel{d}{=} \mathcal{G}^{1/s}$ for the Kotz type distributions.

- **Scale mixture of normals distributions:** The scale mixture of normal distributions consists of a mixture of zero-mean normal distributions (denoted as $\mathcal{X} = \sqrt{\mathcal{K}}\mathcal{N}$, where $\mathcal{N}$ is zero-mean normal distribution and $\mathcal{K}$ is called the mixing distribution with pdf $p_{\mathcal{K}}(t)$). Correspondingly, we can write the pdf of $\mathcal{X}$ as

$$p_{\mathcal{X}}(\mathbf{x}) = \int_t p_{\mathcal{X}}(\mathbf{x}|t) p_{\mathcal{K}}(t) dt \propto \int_t t^{-\frac{m}{2}} \exp\left(-\frac{\mathbf{x}^T \Sigma^{-1} \mathbf{x}}{t}\right) p_{\mathcal{K}}(t) dt. \quad (5.9)$$

Furthermore, to represent a scale mixture of normal distributions in the form of stochastic representations, by inspection we see that a normal distribution, $\mathcal{N}$, can be represented by a multiplication of $\sqrt{\chi_m^2}$ and $\mathcal{S}$ (from the stochastic representation), so that the following holds:

$$\mathcal{X} \stackrel{d}{=} \sqrt{\mathcal{K}}\mathcal{N} \stackrel{d}{=} \sqrt{\mathcal{K} \cdot \chi_m^2} \cdot \mathcal{S} \stackrel{d}{=} \sqrt{\mathcal{K} \cdot \mathcal{G}} \mathcal{S}, \quad (5.10)$$

where $\mathcal{G} \sim \text{Ga}(m/2, 2)$. In this case, $\mathcal{R}^2 \stackrel{d}{=} \mathcal{K} \cdot \mathcal{G}$ by the stochastic representation. Therefore, although (5.9) and (5.10) are equivalent, (5.10) provides a unified stochastic representation within the elliptical family. This allows us to discuss different $\mathcal{K}$ for different sub-classes of scale mixture normals in the following.

For the Pearson type VII distributions, where the nuclear density generator is given by $g_c(t) = (1 + t/v)^{-s}$, we can obtain the corresponding normalisation term c as follows,

$$c = \left(\int_0^\infty t^{m-1} (1 + t^2/v)^{-s} dt \right)^{-1} = \frac{2\Gamma(s)}{(v)^{m/2} \Gamma(s - m/2) \Gamma(m/2)}. \quad (5.11)$$

Consequently, the pdf of $\mathcal{R}^2$ becomes:

$$p_{\mathcal{R}^2}(t) = \frac{\Gamma(s)}{\Gamma(s - \frac{m}{2}) \Gamma(\frac{m}{2}) v^{\frac{m}{2}}} \left(1 + \frac{t}{v}\right)^{-s} t^{\frac{m}{2}-1}. \quad (5.12)$$

Again, the rather complicated form of (5.12) makes it impossible to generate samples. It can be found that when $p_{\mathcal{K}}(t) = \frac{(\frac{v}{2})^{s-m/2}}{\Gamma(s-m/2)} t^{m/2-s-1} \exp(-v/2t)$ and $\mathcal{G} \sim \text{Ga}(m/2, 2)$, $\mathcal{R}^2$ in (5.12) has the same pdf as $\mathcal{K} \cdot \mathcal{G}$. This can be verified as follows,

$$\begin{aligned} p_{\mathcal{R}^2}(t) &= \int_0^\infty p_{\mathcal{G}}(t|\tau) p_{\mathcal{K}}(\tau) d\tau = \int_0^\infty \frac{(\frac{2}{\tau})^{\frac{m}{2}} t^{\frac{m}{2}-1} \exp(-\frac{t}{2\tau}) (\frac{v}{2})^{s-\frac{m}{2}}}{\Gamma(\frac{m}{2}) \Gamma(s - \frac{m}{2})} \tau^{\frac{m}{2}-s-1} \exp(-\frac{v}{2\tau}) d\tau \\ &= \frac{\Gamma(s) t^{\frac{m}{2}-1}}{\Gamma(s - \frac{m}{2}) \Gamma(\frac{m}{2}) v^{\frac{m}{2}} (1 + \frac{t}{v})^s}, \end{aligned} \quad (5.13)$$

where $\mathcal{K}^{-1} \sim \text{Ga}(s - \frac{m}{2}, \frac{v}{2})$ and $\mathcal{R}^2 \stackrel{d}{=} \mathcal{K} \cdot \mathcal{G}$.

Moreover, for other types within the scale mixture normals, Barndorff *et al.* [188] have proved that the generalised hyperbolic distributions can be formulated in the form of (5.9) when $\mathcal{K}$ satisfies the inverse-Gaussian distribution. The elliptical logistic distributions are also mixtures of normals, where $\sqrt{\mathcal{K}}$ in (5.9) relates to the Kolmogorov-Smirnov distribution [173]. Besides, the relationship between the scale mixture normals and the α -stable distribution

is given in [139], which satisfies our equivalent form of (5.10). We omit the tedious proofs here.

- **Pearson type II distributions:** The nuclear density generator is $g_c(t) = (1 - t)^{s-1}$ with the constraint $t \in [0, 1]$. We omit the proof here because the calculation of c and the pdf of $\mathcal{R}^2$ are well documented in [16].

This completes the proof of Table 5.1.

5.6.2 Proof of Theorem 5.1

Symmetry: It is easy to verify the symmetry of $d_U(\mathbf{y}_1, \mathbf{y}_2)$, i.e., $d_U(\mathbf{y}_1, \mathbf{y}_2) = d_U(\mathbf{y}_2, \mathbf{y}_1)$.

Non-negativity: Due to $\arccos(\cdot) \geq 0$, $d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) \geq 0$ and $\gamma(i, j) \geq 0$ for all i, j , $d_U(\mathbf{y}_1, \mathbf{y}_2) \geq 0$. More importantly, the equality holds if and only if $\mathbf{y}_1 =^d \mathbf{y}_2$. In this case, we have $\min_{\gamma(i,j)} \sum_{i,j} \frac{\gamma(i,j)}{k} d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) = 0$ and $\min_{\gamma(i,j)} \arccos(\sum_{i,j} \gamma(i,j) \sqrt{\pi_{i,1} \pi_{j,2}}) = \arccos(\sum_i \pi_{i,1}) = \arccos(1) = 0$, which results in $d_U(\mathbf{y}_1, \mathbf{y}_2) = 0$.

Triangle inequality: Because $\gamma(i, j)$ is binary $\in \{0, 1\}$ for each pair $\{i, j\}$, and it satisfies $\sum_{i=1}^k \gamma(i, j) = 1$ and $\sum_{j=1}^k \gamma(i, j) = 1$, the $\gamma(i, j)$ operates as a bijection between the elliptical distributions within the first and second EMMs.

Then, in order to prove the triangle property, we denote the third EMM as $\mathbf{y}_3 =^d \sum_{h=1}^k z_{h,3} \mathbf{x}_{h,3}$, and investigate the relationship between $d_U(\mathbf{y}_1, \mathbf{y}_2)$ and $d_U(\mathbf{y}_1, \mathbf{y}_3) + d_U(\mathbf{y}_2, \mathbf{y}_3)$. We use $\gamma^*(\cdot, \cdot)$ to denote the optimal $\gamma(\cdot, \cdot)$ in the defined distance $d_U(\cdot, \cdot)$, and also define the following function

$$f(\mathbf{y}_1, \mathbf{y}_2, \gamma(i, j)) = \sum_{i,j} \gamma(i, j) d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) + \arccos\left(\sum_{i,j} \gamma(i, j) \sqrt{\pi_{i,1} \pi_{j,2}}\right). \quad (5.14)$$

By Definition 5.1, $\min_{\gamma(i,j)} f(\mathbf{y}_1, \mathbf{y}_2, \gamma(i, j)) = f(\mathbf{y}_1, \mathbf{y}_2, \gamma^*(i, j)) = d_U(\mathbf{y}_1, \mathbf{y}_2)$.

More importantly, for two arbitrary $\gamma(i, h)$ and $\gamma(h, j)$, their combination $\gamma(i, h) \cap \gamma(h, j) = \sum_{h=1}^k \gamma(i, h) \cdot \gamma(h, j)$ also formulates a transport plan $\gamma(i, j)$ because $\gamma(i, h) \cap \gamma(h, j)$ is still binary and $\sum_{i=1}^k \gamma(i, h) \cap \gamma(h, j) = \sum_{j=1}^k \gamma(i, h) \cap \gamma(h, j) = 1$.

We therefore now arrive at

$$\begin{aligned}
& d_U(\mathbf{y}_1, \mathbf{y}_3) + d_U(\mathbf{y}_2, \mathbf{y}_3) \\
&= f(\mathbf{y}_1, \mathbf{y}_3, \gamma^*(i, h)) + f(\mathbf{y}_2, \mathbf{y}_3, \gamma^*(j, h)) \\
&= \frac{1}{k} \left(\sum_{i,h} \gamma^*(i, h) d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{h,3}) + \sum_{h,j} \gamma^*(h, j) d_W^2(\mathbf{x}_{h,3}, \mathbf{x}_{j,2}) \right) \\
&\quad + \arccos \left(\sum_{i,h} \gamma^*(i, h) \sqrt{\pi_{i,1} \pi_{h,3}} \right) + \arccos \left(\sum_{h,j} \gamma^*(h, j) \sqrt{\pi_{h,3} \pi_{j,2}} \right) \\
&\geq \frac{1}{k} \sum_{i,j} (\gamma^*(i, h) \cap \gamma^*(h, j)) d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) \\
&\quad + \arccos \left(\sum_{i,j} (\gamma^*(i, h) \cap \gamma^*(h, j)) \sqrt{\pi_{i,1} \pi_{j,2}} \right) \\
&= f(\mathbf{y}_1, \mathbf{y}_2, \gamma^*(i, h) \cap \gamma^*(h, j)) \\
&\geq f(\mathbf{y}_1, \mathbf{y}_2, \gamma^*(i, j)) = d_U(\mathbf{y}_1, \mathbf{y}_2).
\end{aligned} \tag{5.15}$$

In (5.15), the first inequality holds due to the fact that both $d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ and $\arccos(\sum_{i,j} \gamma(i, j) \sqrt{\pi_{i,1} \pi_{j,2}})$ satisfy the triangle property. Moreover, the second inequality is due to the fact that the combined plan $\gamma(i, h) \cap \gamma(h, j)$ is not necessarily the optimal plan between $\mathbf{y}_1$ and $\mathbf{y}_2$, as the optimal plan $\gamma^*(i, j)$ achieves the minimum and defines the distance of $d_U(\mathbf{y}_1, \mathbf{y}_2)$.

This completes the proof of Theorem 5.1.

5.6.3 Proof of Lemma 5.1

Recall that $\mathbf{y} \sim \sum_{i=1}^k \pi_i \mathcal{E}(\mathbf{x} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathcal{R})$. We can now write the definition of the Wasserstein distance in the form

$$d_W^2(\mathbf{y}_1, \mathbf{y}_2) = \inf_{\eta(\mathbf{y}_1, \mathbf{y}_2)} \int_{m \times m} \eta(\mathbf{y}_1, \mathbf{y}_2) \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 d\mathbf{x}_1 d\mathbf{x}_2, \tag{5.16}$$

where $\eta(\mathbf{y}_1, \mathbf{y}_2)$ denotes the joint distribution between $\mathbf{y}_1$ and $\mathbf{y}_2$, and satisfies $\int_m \eta(\mathbf{y}_1, \mathbf{y}_2) d\mathbf{x}_1 = p_{\mathbf{y}_2}(\mathbf{x}_2)$ and $\int_m \eta(\mathbf{y}_1, \mathbf{y}_2) d\mathbf{x}_2 = p_{\mathbf{y}_1}(\mathbf{x}_1)$.

More importantly,

$$\begin{aligned} \sum_{i,j} \frac{\gamma^*(i,j)}{k} d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) &= \\ \sum_{i,j} \frac{\gamma^*(i,j)}{k} \inf_{\eta(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})} \int_{m \times m} \eta(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 d\mathbf{x}_1 d\mathbf{x}_2 & \quad (5.17) \\ = \int_{m \times m} \sum_{i,j} \frac{\gamma^*(i,j)}{k} \eta^*(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 d\mathbf{x}_1 d\mathbf{x}_2, \end{aligned}$$

where $\eta^*(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ is the optimal plan that achieves the Wasserstein distance between $\mathbf{x}_{i,1}$ and $\mathbf{x}_{j,2}$. Therefore, by comparing (5.16) and (5.17), we can easily observe that $\sum_{i,j} \frac{\gamma^*(i,j)}{k} \eta^*(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ consists a subset of joint distribution of $\eta(\mathbf{y}_1, \mathbf{y}_2)$, due to $\pi_{i,1} = \pi_{j,2} = 1/k$. In other words, because of the factorisation from $\mathbf{y}$ to $\mathbf{x}$, the $\sum_{i,j} \frac{\gamma^*(i,j)}{k} \eta^*(\mathbf{x}_{i,1}, \mathbf{x}_{j,2})$ does not necessarily achieve the optimal transport plan between $\mathbf{y}_1$ and $\mathbf{y}_2$.

Moreover, for balanced EMMs, the following holds for arbitrary $\gamma(i, j)$,

$$\arccos\left(\sum_{i,j} \gamma(i, j) \sqrt{\pi_{i,1} \pi_{j,2}}\right) = 0. \quad (5.18)$$

Thus, we have

$$d_W^2(\mathbf{y}_1, \mathbf{y}_2) \leq \sum_{i,j} \frac{\gamma^*(i,j)}{k} d_W^2(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}) = d_U(\mathbf{y}_1, \mathbf{y}_2). \quad (5.19)$$

This completes the proof of Lemma 5.1.

5.6.4 Proof of Lemma 5.2

Due to the fact that $\gamma(i, j)$ in Definition 1 is a bijection between the mixture components in $\mathbf{y}_1$ and $\mathbf{y}_2$, for each $\boldsymbol{\pi}_1$, $\boldsymbol{\mu}_{i,1}$ and $\boldsymbol{\Sigma}_{i,1}$ in $\mathbf{y}_1$, there exist only one corre-

sponding $\boldsymbol{\pi}_2$, $\boldsymbol{\mu}_{j,2}$ and $\boldsymbol{\Sigma}_{j,2}$, respectively, in $\mathcal{Y}_2$. In other words, when casting the problem onto the parameter space, the Hessian of $d_U(\mathcal{Y}_\theta, \mathcal{Y}_{\theta+s\Delta\theta})$ can be calculated as follows,

$$\begin{aligned} & \frac{\partial^2 d_U(\mathcal{Y}_\theta, \mathcal{Y}_{\theta+s\Delta\theta})}{\partial s^2} \Big|_{s \rightarrow 0} \\ &= \frac{1}{k} \sum_{i=1}^K \frac{\partial^2 d_W^2(\mathcal{X}_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}, \mathcal{X}_{\boldsymbol{\mu}_i+s\Delta\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i+s\Delta\boldsymbol{\Sigma}_i})}{\partial s^2} \Big|_{s \rightarrow 0} \\ & \quad + \frac{\partial^2 \arccos(\sqrt{\boldsymbol{\pi}^T}(\sqrt{\boldsymbol{\pi} + s\Delta\boldsymbol{\pi}}))}{\partial s^2} \Big|_{s \rightarrow 0}, \end{aligned} \quad (5.20)$$

where the square root operation $\sqrt{\cdot}$ is performed element-wise.

In (5.20), we can see that the manifold defined by $d_U(\mathcal{Y}_\theta, \mathcal{Y}_{\theta+\Delta\theta})$ is a product manifold defined by $d_W^2(\mathcal{X}_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}, \mathcal{X}_{\boldsymbol{\mu}_i+\Delta\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i+\Delta\boldsymbol{\Sigma}_i})$ and $\arccos(\sqrt{\boldsymbol{\pi}^T}(\sqrt{\boldsymbol{\pi} + \Delta\boldsymbol{\pi}}))$.

Furthermore, $\arccos(\sqrt{\boldsymbol{\pi}^T}(\sqrt{\boldsymbol{\pi} + \Delta\boldsymbol{\pi}}))$ defines a sphere manifold [103] (Examples 3.5.1 and 3.6.1), for which basic operations are provided in Examples 4.1.1, 5.4.1 and 8.1.7 of [103]. On the other hand, the Wasserstein distance between two elliptical distributions has an explicit representation as follows [176, 189],

$$\begin{aligned} d_W^2(\mathcal{X}_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}, \mathcal{X}_{\boldsymbol{\mu}_i+\Delta\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i+\Delta\boldsymbol{\Sigma}_i}) &= \|\Delta\boldsymbol{\mu}_i\|_2^2 \\ &+ \frac{\mathbb{E}[\mathcal{R}^2]}{m} \text{tr}(\boldsymbol{\Sigma}_i + (\boldsymbol{\Sigma}_i + \Delta\boldsymbol{\Sigma}_i) - 2(\boldsymbol{\Sigma}_i^{1/2}(\boldsymbol{\Sigma}_i + \Delta\boldsymbol{\Sigma}_i)\boldsymbol{\Sigma}_i^{1/2})^{1/2}). \end{aligned} \quad (5.21)$$

We can thus conclude that the manifold for $\boldsymbol{\mu}_i$ is the conventional Euclidean space within $\mathbb{R}^m$. Moreover, the manifold of $\boldsymbol{\Sigma}_i$ can also be correspondingly obtained on the basis of [178].

This completes the proof of Lemma 5.2.

5.6.5 Calculations within Table 5.2

The basic operations on a sphere manifold were described in [103] (Examples 3.5.1, 3.6.1, 4.1.1, 5.4.1 and 8.1.7). As the manifold for $\boldsymbol{\mu}_i$ is the conventional Euclidean space, the trivial gradient descent can be employed. Operations for $\boldsymbol{\Sigma}_i$ under the Wasserstein manifold can be found in [178]. Furthermore, we omit the scale weight

$\frac{\mathbb{E}[\mathcal{R}^2]}{m}$ in our work as it can be incorporated into the stepsize α during the gradient descent.

To calculate the Euclidean gradients, we first explicitly express our cost function, i.e., the sliced Wasserstein distance $d_{SW}(\mathbf{Y}_\theta, \mathbf{Y}^*)$, as follows,

$$d_{SW}(\mathbf{Y}_\theta, \mathbf{Y}^*) = \int_m \inf_{\eta(\text{Ra}(\mathbf{Y}_\theta, \mathbf{p}), \text{Ra}(\mathbf{Y}^*, \mathbf{p}))} \int_{\mathbb{R} \times \mathbb{R}} \eta_R \cdot (y_1 - y_2)^2 dy_1 dy_2 d\mathbf{p}, \quad (5.22)$$

and

$$\eta_R = \eta(\text{Ra}(\mathbf{Y}_\theta, \mathbf{p}), \text{Ra}(\mathbf{Y}^*, \mathbf{p})), \quad (5.23)$$

where y_1 denotes the Radon transform [183] $\text{Ra}(\mathbf{Y}_\theta, \mathbf{p})$, which projects $\mathbf{Y}_\theta$ onto a one-dimensional random variable along the direction $\mathbf{p}$. Similarly, y_2 denotes the Radon transform of $\mathbf{Y}^*$ with the direction $\mathbf{p}$. Recall that $\eta(\cdot, \cdot)$ is a joint distribution. Due to the fact that elliptical distributions belong to the location-scale family, as described by the stochastic representation, the Radon transform of each $\mathcal{X}_{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}$ has a simple representation in the form $\mathcal{X}_{\mathbf{p}^T \boldsymbol{\mu}_i, \mathbf{p}^T \boldsymbol{\Sigma}_i \mathbf{p}} \sim \mathcal{E}(x; \mathbf{p}^T \boldsymbol{\mu}_i, \mathbf{p}^T \boldsymbol{\Sigma}_i \mathbf{p}, \mathcal{R})$, where $\mathcal{X}$ is a one-dimensional elliptical distribution.

Fortunately, the one-dimensional Wasserstein distance has the closed-form solution [98], so that we can re-write (5.22) as

$$\begin{aligned} \min_{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} d_{SW}(\mathbf{Y}_\theta, \mathbf{Y}^*) = \\ \min_{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} \int_m \int_{\mathbb{R}} |y - T(y)|^2 \sum_{i=1}^k \frac{c_m \pi_i}{(\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p})^{\frac{1}{2}}} g\left(\frac{(y - \mathbf{p}^T \boldsymbol{\mu}_i)^2}{\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p}}\right) dy d\mathbf{p}, \end{aligned} \quad (5.24)$$

where $T(y)$ is the optimal transport plan between $\text{Ra}(\mathbf{Y}_\theta, \mathbf{p})$ and $\text{Ra}(\mathbf{Y}^*, \mathbf{p})$, which can be explicitly obtained via their cumulative distribution functions [172]. When optimising (5.24) using stochastic gradient descent, for each $\mathbf{p}$, we therefore minimise

$$\begin{aligned} \min_{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} L(\boldsymbol{\theta}, \mathbf{p}) = \\ \min_{\boldsymbol{\pi}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} \int_{\mathbb{R}} |y - T(y)|^2 \sum_{i=1}^k \frac{c_m \pi_i}{(\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p})^{\frac{1}{2}}} g\left(\frac{(y - \mathbf{p}^T \boldsymbol{\mu}_i)^2}{\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p}}\right) dy. \end{aligned} \quad (5.25)$$

Then, by calculating the derivatives with regard to $\sqrt{\pi_i}$, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, and upon introducing the Kantorovich potential $\phi(y)$ [172], we obtain the Euclidean gradients as follows,

$$\begin{aligned} \nabla_{\sqrt{\pi_i}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} L(\boldsymbol{\theta}, \mathbf{p}) = \\ \int_{\mathbb{R}} \phi(y) \nabla_{\sqrt{\pi_i}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i} \left(\sum_{i=1}^k \frac{c_m \sqrt{\pi_i}^2}{(\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p})^{\frac{1}{2}}} g\left(\frac{(y - \mathbf{p}^T \boldsymbol{\mu}_i)^2}{\mathbf{p}^T \boldsymbol{\Sigma}_i^h \mathbf{p}}\right) \right) dy, \end{aligned} \quad (5.26)$$

where $\nabla_{\sqrt{\pi_i}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}(\cdot)$ denotes the derivatives with regard to $\{\sqrt{\pi_i}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$, respectively. The Euclidean gradients in the Table 5.2 can then be easily calculated.

5.7 Summary

We have introduced a new and complete framework for solving general EMMs. To this end, we have first provided a unified and easy-to-use summary of the candidates for identifiable EMMs, which is achieved via a stochastic representation. Then, a relaxed Wasserstein distance for EMMs has been proposed, which unlike the existing Wasserstein distances allows the corresponding metrics to be explicitly calculated. The so established manifold has been shown to consistently improve performances in terms of Wasserstein cost and even the NLL cost, and also significantly stabilise the optimisation on EMMs, making them robust to initialisation. We have further proposed a directional adaptively accelerated algorithm to enhance and stabilise the convergence, the performance of which has been validated through comprehensive experimental results.

Chapter 6

Reciprocal GAN through Characteristic Functions (RCF-GAN)

Generative adversarial nets (GANs) involve deep neural nets in the generating process, and the integral probability metric (IPM) equips GANs with the necessary theoretical support for comparing statistical moments in an embedded domain of the critic, while stabilising their training and mitigating model collapse issues. For enhanced intuition and physical insight, we introduce a generalisation of IPM-GANs which operates by directly comparing probability distributions rather than their moments. This is achieved through characteristic functions (CFs), a powerful tool that uniquely comprises all the information about any general distribution. For rigour, we first prove theoretically the ability of the CF loss to compare probability distributions, and establish the physical meaning of the phase and amplitude of CFs. An optimal sampling strategy is then developed to calculate the CFs, and an equivalence between the embedded and data domains is proved when a reciprocal theory holds. We therefore seamlessly combine the IPM-GAN with an auto-encoder structure, equipped by an advanced anchor architecture, which adversarially learns a semantic low-dimensional manifold for both generation and reconstruction. This ef-

efficient structure, called reciprocal CF GAN (RCF-GAN), uses only two modules and a simple training strategy, to achieve the state-of-the-art bi-directional generation. Experiments demonstrate the superior performance of the RCF-GAN on generation, reconstruction and interpolations of images.

6.1 Introduction

GANs owe their success to their ability to capture complicated data distributions [19]. In practical applications, however, their significant potential still remains under-explored, as GANs typically suffer from unstable training and mode collapse issues [77]. An effective and elegant way to address these issues is to replace the Jensen-Shannon (JS) divergence in measuring the discrepancy in the original form of GANs [21] by another class of metrics called the IPM [99], given by

$$d(\mathcal{P}_d, \mathcal{P}_g) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim \mathcal{P}_d}[f(x)] - \mathbb{E}_{x \sim \mathcal{P}_g}[f(x)]|, \quad (6.1)$$

where the symbol $\mathcal{F}$ in IPMs represents a collection of (typically real) bounded functions, $\mathcal{P}_g$ denotes the generated distribution, and $\mathcal{P}_d$ is the real data distribution. Using IPMs to improve GANs has been justified by the fact that data distributions in the real-world are typically embedded on low-dimensional manifolds. This is intuitive because real-world data tend to preserve semantic information instead of just representing a collection of rather random pixels. Thus, while the divergence measure (“bin-to-bin” comparison) of the original GAN can easily max out, the IPMs such as the Wasserstein distance (“cross-bin” comparison) are able to consistently yield a meaningful measure between the generated and real data distributions [21].

The variation in the collections of $\mathcal{F}$ in (6.1), therefore, defines different IPM-GANs, while the supremum $\sup_{f \in \mathcal{F}}$ is typically achieved by the discriminator net, or more formally, the *critic* in the IPM-GANs. The original IPM-GAN was motivated by the Wasserstein GAN (W-GAN) [4], where $\mathcal{F}$ denotes the set of all 1-Lipschitz functions. However, it has been widely argued that the *critic* is not powerful enough

to search within all the 1-Lipschitz function spaces, which leads to limited diversity of the generator due to an ill-posed equivalence measurement of $\mathcal{P}_d$ and $\mathcal{P}_g$ [71, 81]. Follow-up works have been proposed to improve the W-GAN by either its enhancement to satisfy the 1-Lipschitz condition (e.g., by gradient penalty [79] or spectral normalization [80]) or by employing an *easy-to-implement* $\mathcal{F}$ for the *critic*. The latter, by virtue of relaxing the *critic*, typically leads to a tedious comparison on the embedded feature domain, i.e., by matching higher-order moments instead of the mean matching in the W-GAN. This path includes numerous recent GANs which additionally consider the second-order moment (e.g., Fisher-GAN [84] and McGAN [83]), together with explicitly (e.g., Sphere GAN [8]) or implicitly (e.g., MMD-GAN [3, 5]) comparing higher-order moments.

Generalising (6.1) as a moment matching problem has been justified as a natural and beneficial way to understand IPM-GANs [190–192]. This also compensates for the known deficiency whereby the *critic* may not transform the data distributions into unimodal distributions, as for example, in the case of Gaussian distribution that is solely determined by the first- and second-order moments.

Moreover, it is both more robust and elegant to compare the distributions directly, because the equivalence in distributions ensures the equivalence in the moments; the inverse, however, does not necessarily hold. As a powerful tool which contains all the information relevant to a distribution, the *characteristic function* (CF) provides a universal way of comparing distributions, even when their probability density functions (pdfs) do not exist. The CF also has a one-to-one correspondence with the cumulative density function (cdf), which has also been demonstrated to benefit the design of GANs [193]. Compared to the moment generating function (mgf) that has been reflected in the MMD-GAN [5], the CF is unique and universally existent, and is more sensitive to distribution alignments [194, 195]. More importantly, the CF is automatically aligned at $\mathbf{0}$; this means that even a simple “bin-to-bin” comparison between CFs can consistently provide a meaningful measure and thus avoid the issue of gradient vanishing associated with the original GAN [4]. Also, the weak conver-

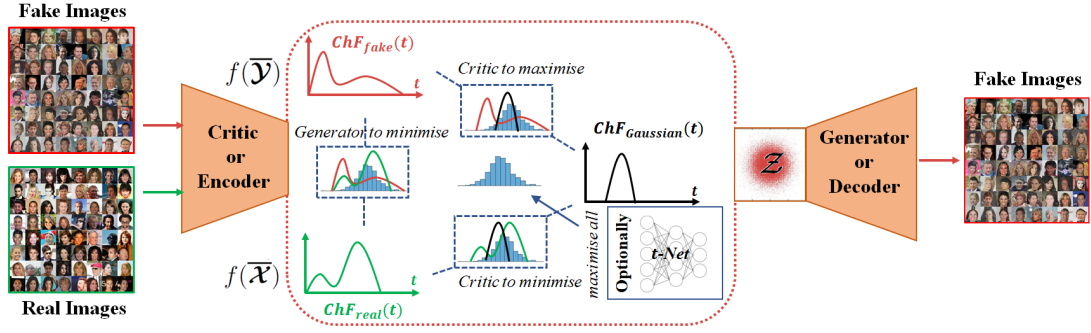


Figure 6.1: The overall structure of the proposed RCF-GAN. The generator serves to minimise the CF loss between the embedded real and fake distributions. The *critic* serves to minimise the CF loss between the embedded real and the input noise distributions, whilst maximising the CF loss between the embedded fake and the input noise distributions. Moreover, an MSE loss between the embedded fake and the input noise distributions is regularised as the auto-encoder loss, which has not been shown in the figure. An optional *t*-net can be employed to optimally sample the CF loss.

gence property of CFs ensures that the convergence in CFs implies the convergence in the distributions.

In this chapter, we propose a reciprocal CF GAN (RCF-GAN) as a natural generalisation of the existing IPM-GANs, with the overall structure shown in Fig. 6.1. It needs to be pointed out that incorporating the CF in a GAN is non-trivial because the CF is basically complex-valued and in addition the comparison has to be performed on functions. To address these difficulties, we first illuminate the role of CFs by showing that its phase is closely related to the distribution centre, whereas the amplitude dominates the distribution scale. This provides a feasible way of balancing the accuracy and diversity of the generation. Then, for the comparison over functions, we prove that other than in the whole space of CFs, sampling within a small ball around 0 of CFs is sufficient to compare two distributions. This also enables the proposed CF loss to be bounded and differentiable almost everywhere. We further optimise the sampling strategy by automatically adjusting sampling distributions under the umbrella of the *scale mixture of normals* [196].

Benefiting from the power of CFs in comparing distributions, we propose to perform distribution comparison solely in the embedded domain, and prove the equivalence to its data domain counterpart, when a reciprocal condition between the generator

and the *critic* holds. This motivates us to incorporate an auto-encoder structure to satisfy the theoretical requirement of a reciprocal. In this way, the *critic* in our RCF-GAN is further relaxed and only focuses on learning a meaningful embedding. Furthermore, different from many existing adversarial works with auto-encoders incorporating at least three modules¹ [3, 5, 87–92], our RCF-GAN only requires two modules that already exist in a GAN; the *critic* is an encoder and the generator also serves as a decoder. This is neat and reasonable and comes without an increase in computational complexity and complicated (unstable) training strategies, as well as without other stringent requirements such as the Lipschitz continuity. More importantly, the framework of comparing the full distributions in the embedded domain enables our CF-GAN to learn a semantic and meaningful latent space.

The effectiveness of our RCF-GAN is comprehensively experimentally verified through the tasks of image generation, interpolation and reconstruction, where our CF-GAN learns a meaningful embedded space and also avoids the smoothing artefact that arises from the use of point-wise mean square error (MSE) employed in the data domain. Thus, RCF-GAN benefits from both the auto-encoder and the GANs, i.e., *bi-directionally* generating *clear* images. Our experimental results show that the proposed RCF-GAN achieves remarkable improvements in image generation, together with an additional capability in the reconstruction and interpolation.

A very recent independent work [6] named OCF-GAN also employs the CF by using the same structure of MMD-GANs. Our RCF-GAN is substantially different from that in [6], in that:

- Our *critic* operates as semantic embeddings and learns a meaningful embedded space, instead of being a component to build complete metrics, as with the existing GANs (e.g., OCF-GAN, MMD-GANs and W-GANs) do.
- Our CF design, is novel in its triangle anchor design with the l_1 -norm (to stabilise convergence), meaningful analysis of amplitude and phase (to favour

¹To our best knowledge, the only exception is the AGE [7], which adopts two modules in an auto-encoder, under a max-min problem and different losses. Please see the Chapter 2.2 for the details.

other distribution alignment tasks), the use of t -net of outputting scales (to automatically optimise $\mathcal{T}$ distribution types), and useful supporting theory (to correctly and efficiently use CFs in practice).

- Our RCF-GAN seamlessly combines the auto-encoder and GANs by using only two neat modules while achieving the state-of-the-art performances, whereas the majority of adversarial learning structures use at least three modules with auto-encoding separated from GANs.

Consequently, while the results in [6] were reported with all images rescaled to the size of 32×32 , our RCF-GAN consistently outperforms the work in [6] in various ways including high resolutions, net structures and functionalities.

6.2 Characteristic Function Loss and Efficient Sampling Strategy

6.2.1 Characteristic Function and Elliptical Distribution

The CF of a random variable, $\mathbf{x} \in \mathbb{R}^m$, represents the expectation of its complex unitary transform, given by

$$\Phi_{\mathbf{x}}(\mathbf{t}) = \mathbb{E}_{\mathbf{x}}[e^{j\mathbf{t}^T \mathbf{x}}] = \int_{\mathbf{x}} e^{j\mathbf{t}^T \mathbf{x}} dF_{\mathbf{x}}(\mathbf{x}), \quad (6.2)$$

where $F_{\mathbf{x}}(x)$ is the cdf of $\mathbf{x}$. We thus have $\Phi_{\mathbf{x}}(\mathbf{0}) = 1$ and $|\Phi_{\mathbf{x}}(\mathbf{t})| \leq 1$ for all $\mathbf{t}$. This property ensures that CFs can be straightforwardly compared in a “bin-to-bin” manner, because all CFs are automatically aligned at $\mathbf{t} = \mathbf{0}$. Moreover, when the pdf of $\mathbf{x}$ exists, the expression in (6.2) is equal to its inverse Fourier transform; this ensures that $\Phi_{\mathbf{x}}(\mathbf{t})$ is uniformly continuous. Another important property of the CF is that it uniquely and universally retains all the information regarding a random variable. In other words, a random variable does not necessarily need to possess a pdf (e.g., when it is an α -stable distribution), but its CF always exists.

As the cdf, $F_{\mathcal{X}}(\mathbf{x})$, is unknown but has to be compared, we employ the empirical characteristic function (ECF) as an asymptotic approximation in the form of $\hat{\Phi}_{\mathcal{X}_n}(\mathbf{t}) = \sum_{i=1}^n e^{j\mathbf{t}^T \mathbf{x}_i}$, where $\{\mathbf{x}_i\}_{i=1}^n$ are n i.i.d. samples drawn from $\mathcal{X}$. As a result of the *Levy continuity theorem* [197], the ECF converges weakly to the population CF [198]. More importantly, the *uniqueness theorem* guarantees that two random variables have the same distribution if and only if their CFs are identical [199]. Therefore, together with the weak convergence, the ECF provides a feasible and good proxy to the distribution, which has also been preliminarily applied in two sample test [200, 201]. Before proceeding further, we introduce an important class of distributions that will be used in this work.

Example 6.1. *Within unimodal distributions, one broad class of distributions is called the elliptical distribution, which is general enough to include various important distributions such as the Gaussian, Laplace, Cauchy, Student-t, α -stable and logistic distributions. The elliptical distributions do not necessarily have pdfs, and we refer to [16] for more detail. The CF of an elliptical distribution, $\mathcal{X}$, however, always exists and has the following form*

$$\Phi_{\mathcal{X}}(\mathbf{t}) = e^{j\mathbf{t}^T \boldsymbol{\mu}} \psi(\mathbf{t}^T \boldsymbol{\Sigma} \mathbf{t}), \quad (6.3)$$

where $\boldsymbol{\mu}$ denotes the distribution centre, $\boldsymbol{\Sigma}$ is the distribution scale, and $\psi(\cdot)$ is a real-valued function $\mathbb{R} \rightarrow \mathbb{R}$, for example, $\psi(s) = e^{(-s^2/2)}$ for the Gaussian distribution. By inspecting (6.3) we can see that the phase of the CF is solely related to the location of data centre and the amplitude is only governed by the distribution scale (diversity).

6.2.2 Distance Measure via Characteristic Functions

The auto-alignment property of the CFs allows us to incorporate a simple “bin-to-bin” comparison over two complex-valued CFs (corresponding to two random

variables $\mathcal{X}$ and $\mathcal{Y}$), in the form

$$\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = \int_{\mathbf{t}} \underbrace{((\Phi_{\mathcal{X}}(\mathbf{t}) - \Phi_{\mathcal{Y}}(\mathbf{t}))(\Phi_{\mathcal{X}}^*(\mathbf{t}) - \Phi_{\mathcal{Y}}^*(\mathbf{t})))}_{c(\mathbf{t})}^{\frac{1}{2}} dF_{\mathcal{T}}(\mathbf{t}), \quad (6.4)$$

where Φ^* denotes the complex conjugate of Φ and $F_{\mathcal{T}}(\mathbf{t})$ is the cdf of a sampling distribution on $\mathbf{t}$. For the convenience of subsequent analysis, we represent the quadratic term for each $\mathbf{t}$ as $c(\mathbf{t}) = (\Phi_{\mathcal{X}}(\mathbf{t}) - \Phi_{\mathcal{Y}}(\mathbf{t}))(\Phi_{\mathcal{X}}^*(\mathbf{t}) - \Phi_{\mathcal{Y}}^*(\mathbf{t}))$. Importantly, $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ is a valid distance that measures the difference between two random variables via CFs, which is provided in Lemma 6.1; this means $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = 0$ if and only if $\mathcal{X} =^d \mathcal{Y}$. A specific type of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ in (6.4) is when the pdf of $\mathbf{t}$ is proportional to $\|\mathbf{t}\|^{-1}$, and its relationship to other metrics, including the Wasserstein and Kolmogorov distances, has been analysed in detail [202].

Lemma 6.1. *The discrepancy between two random variables $\mathcal{X}$ and $\mathcal{Y}$, given by $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ in (6.4), is a distance metric when the support of $\mathcal{T}$ resides in $\mathbb{R}^m$.*

Proof. Please refer to Appendix 6.5.1. □

Furthermore, given that the phase and amplitude of a CF indicate respectively the data centre and diversity as shown in Example 6.1, upon inspecting $c(\mathbf{t})$ we can rewrite it in a physically meaningful way, i.e., through the differences in the corresponding phase and amplitude terms as,

$$\begin{aligned} c(\mathbf{t}) &= |\Phi_{\mathcal{X}}(\mathbf{t})|^2 + |\Phi_{\mathcal{Y}}(\mathbf{t})|^2 - \Phi_{\mathcal{X}}(\mathbf{t})\Phi_{\mathcal{Y}}^*(\mathbf{t}) - \Phi_{\mathcal{Y}}(\mathbf{t})\Phi_{\mathcal{X}}^*(\mathbf{t}) \\ &= |\Phi_{\mathcal{X}}(\mathbf{t})|^2 + |\Phi_{\mathcal{Y}}(\mathbf{t})|^2 - |\Phi_{\mathcal{X}}(\mathbf{t})||\Phi_{\mathcal{Y}}(\mathbf{t})|(2 \cos(\mathbf{a}_{\mathcal{X}}(\mathbf{t}) - \mathbf{a}_{\mathcal{Y}}(\mathbf{t}))) \\ &= |\Phi_{\mathcal{X}}(\mathbf{t})|^2 + |\Phi_{\mathcal{Y}}(\mathbf{t})|^2 - 2|\Phi_{\mathcal{X}}(\mathbf{t})||\Phi_{\mathcal{Y}}(\mathbf{t})| \\ &\quad + 2|\Phi_{\mathcal{X}}(\mathbf{t})||\Phi_{\mathcal{Y}}(\mathbf{t})|(1 - \cos(\mathbf{a}_{\mathcal{X}}(\mathbf{t}) - \mathbf{a}_{\mathcal{Y}}(\mathbf{t}))) \\ &= \underbrace{(|\Phi_{\mathcal{X}}(\mathbf{t})| - |\Phi_{\mathcal{Y}}(\mathbf{t})|)^2}_{\text{amplitude difference}} + 2|\Phi_{\mathcal{X}}(\mathbf{t})||\Phi_{\mathcal{Y}}(\mathbf{t})| \underbrace{(1 - \cos(\mathbf{a}_{\mathcal{X}}(\mathbf{t}) - \mathbf{a}_{\mathcal{Y}}(\mathbf{t})))}_{\text{phase difference}}, \end{aligned} \quad (6.5)$$

where $\mathbf{a}_{\mathcal{X}}(\mathbf{t})$ and $\mathbf{a}_{\mathcal{Y}}(\mathbf{t})$ represent respectively the angles (phases) of $\Phi_{\mathcal{X}}(\mathbf{t})$ and $\Phi_{\mathcal{Y}}(\mathbf{t})$. This clearly shows that $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ effectively measures the amplitude dif-

ference between random variables, together with their phase difference weighted by the amplitudes. We can further consider a convex combination of the two terms via $0 \leq \alpha \leq 1$, to yield²

$$c_\alpha(\mathbf{t}) = \alpha(|\Phi_{\mathcal{X}}(\mathbf{t})| - |\Phi_{\mathcal{Y}}(\mathbf{t})|)^2 + (1 - \alpha)(2|\Phi_{\mathcal{X}}(\mathbf{t})||\Phi_{\mathcal{Y}}(\mathbf{t})|(1 - \cos(\mathbf{a}_{\mathcal{X}}(\mathbf{t}) - \mathbf{a}_{\mathcal{Y}}(\mathbf{t}))). \quad (6.6)$$

Recall that for the elliptical distributions in Example 6.1, the phase represents the distribution centre while the amplitude represents the scale; $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ thus measures the both discrepancy of the centres and diversity of two distributions. We show in Figure 6.2-(a) that by swapping the phase and amplitude parts, the saliency information follows the phase part of the CF, which captures the centres of the distribution³. We further illustrate in Figure 6.2-(b) that this property still holds for real data distributions, even though they are much more complicated and can even be non-unimodal. From Figure 6.2-(b)-(d), it shows that mainly training the phase (shown in Figure 6.2-(c)) results in generating images similar to an average of the real data, as a result of minimising the difference of the data centres. On the other hand, when mainly training the amplitude (shown in Figure 6.2-(b)), we can obtain diversified but inaccurate images (“wrong” numbers such as “1” for digit 7 and “6” for digit 5, uneven characters, disconnected artefacts, etc.). To this end, by using different weights in $c_\alpha(\mathbf{t})$, we can flexibly capture the main content via minimising the phase difference, whilst enriching the diversity of generated images by increasing the amplitude loss. This provides a meaningful and feasible way of understanding the GAN loss in controlling the generation process.

6.2.3 Sampling the Characteristic Function Loss

In practice, to calculate $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ efficiently, as mentioned in Section 6.2.1, $\Phi_{\mathcal{X}}(\mathbf{t})$ and $\Phi_{\mathcal{Y}}(\mathbf{t})$ can be evaluated by the ECFs of $\mathcal{X}$ and $\mathcal{Y}$, which are weakly convergent

²Such a split into the amplitude-only and phase-only parts within the mean squared loss is similar to the Least Mean Magnitude Phase (LMMP) algorithm [203, 204].

³This phenomenon has been discovered in the Fourier representation of signals [205, 206]. We validate that this also holds in probabilistic distributions.

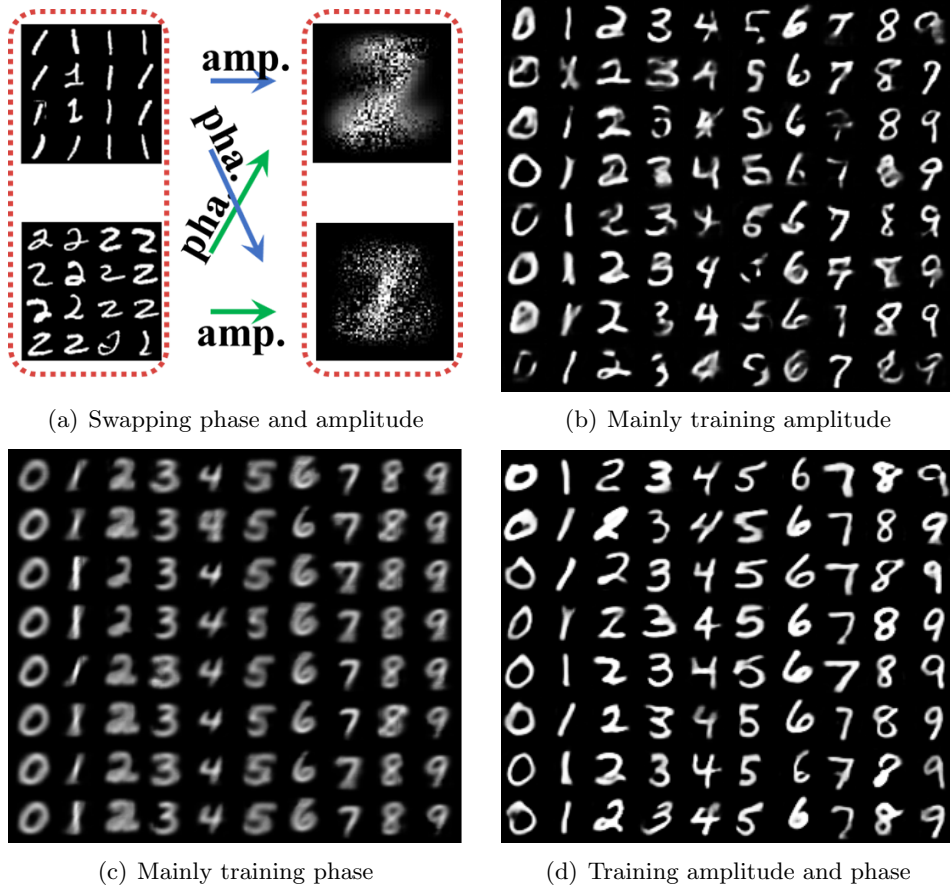


Figure 6.2: Two experiments on the MNIST dataset which illustrate the physical meaning of the phase and amplitude of the CF. (a) A multivariate Gaussian fit to the images of digits 1 and 2, by naively assuming that each pixel is independent from other pixels. The phase and amplitude information of the CFs between the two multivariate distributions were then swapped, followed by random sampling from the swapped distributions. (b)-(d) A generator was directly trained on the given images of each digit. To avoid the impact from the *critic*, we *DID NOT* employ the *critic* in this experiment but directly calculated the loss between images after the generator with different α . We performed training for amplitude for $\alpha = 0.999$ in (b), phase only for $\alpha = 0.001$ in (c) and equally training the amplitude and phase information for $\alpha = 0.5$ in (d).

to the corresponding population CFs. The remaining task is to sample from $F_{\mathcal{T}}(\mathbf{t})$. A direct approach would be to use the neural net where the input is Gaussian noise and the output are the samples of $F_{\mathcal{T}}(\mathbf{t})$. However, Proposition 6.1 indicates that this can lead to ill-posed optima whereby $F_{\mathcal{T}}(\mathbf{t})$ converges to some point mass distributions and is thus no longer supported in $\mathbb{R}^m$, as required in Lemma 6.1. In other words, for the degenerated $F_{\mathcal{T}}(\mathbf{t})$, we may have $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ but $\mathcal{X} \neq^d \mathcal{Y}$. In our experiment, we also found that directly optimising $F_{\mathcal{T}}(\mathbf{t})$ can cause instability.

Proposition 6.1. *The maximum of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ is reached when $F_{\mathcal{T}}(\mathbf{t})$ attains a mass point at $\mathbf{t}^*$, where $\mathbf{t}^* = \arg \max_{\mathbf{t}} c(\mathbf{t})$. The minimum of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ is reached when $F_{\mathcal{T}}(\mathbf{t})$ attains a mass point at $\mathbf{0}$.*

In the way of addressing this ill-posed optimisation on $F_{\mathcal{T}}(\mathbf{t})$, we can impose some constraints on $F_{\mathcal{T}}(\mathbf{t})$, for example, by assuming some parametric distributions. On the other hand, we may also be concerned that the constraints on $F_{\mathcal{T}}(\mathbf{t})$ can impede the ability of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ as a metric to distinguish $\mathcal{X}$ from $\mathcal{Y}$. Lemma 6.2 provides an efficient and feasible way of choosing $F_{\mathcal{T}}(\mathbf{t})$.

Lemma 6.2. *If random variables $\mathcal{X}$ and $\mathcal{Y}$ are supported on a finite interval $[-1, 1]^m$, $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ in (6.4) is still a distance metric for distinguishing $\mathcal{X}$ from $\mathcal{Y}$ for any $F_{\mathcal{T}}(\mathbf{t})$ that samples $\mathbf{t}$ within a small ball about $\mathbf{0}$.*

Proof. Please refer to Appendix 6.5.2. □

Owing to these desirable properties, as shown in the next section, we employ $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ as the loss to compare two distributions from the *critic*. By employing bounded activation functions (tanh, sigmoid, etc.), the requirement of Lemma 6.2 is automatically satisfied. Therefore, instead of searching within all the real distribution spaces, the choices of $F_{\mathcal{T}}(\mathbf{t})$ can be safely restricted to some zero-mean distributions, e.g., the Gaussian distribution. Furthermore, compared to the fixed Gaussian distribution, it is preferable, whilst avoiding the ill-posed optimum, that $F_{\mathcal{T}}(\mathbf{t})$ could be optimised to better accommodate the difference between two distributions.

In this work, we choose $F_{\mathcal{T}}(\mathbf{t})$ as the cdf of a broad class of distributions called the *scale mixture of normals*, in the form of

$$p_{\mathcal{T}}(\mathbf{t}) = \int_{\Sigma} p_{\mathcal{N}}(\mathbf{t}|\mathbf{0}, \Sigma) p_{\Sigma}(\Sigma) d\Sigma, \quad (6.7)$$

where $p_{\mathcal{T}}(\mathbf{t})$ is the pdf of $F_{\mathcal{T}}(\mathbf{t})$, while $p_{\mathcal{N}}(\mathbf{t}|\mathbf{0}, \Sigma)$ denotes the zero-mean Gaussian distribution with the covariance given by Σ , and $p_{\Sigma}(\Sigma)$ denotes distributions of

Σ . It needs to be pointed out that the *scale mixture of normals* comprises a large portion of the elliptical distributions and includes many important distributions (e.g., the Gaussian, Cauchy, Student- t , hyperbolic distributions [139]) by choosing different $p_{\Sigma}(\Sigma)$. Therefore, instead of directly optimising $F_{\mathcal{T}}(\mathbf{t})$, which leads to ill-posed solutions, we alternatively optimise the neural net to output the samples of $p_{\Sigma}(\Sigma)$. The use of the affine transformation (or the re-parametrisation trick) allows us to propagate back the gradients.

We should point out that the term $\int_{\mathbf{t}} c(\mathbf{t}) dF_{\mathcal{T}}(\mathbf{t})$ contained in our CF loss can also be interpreted as certain well behaved kernels in the MMD metric. This is due to the fact that the shift invariant and characteristic kernels in the MMD metric have to satisfy $k(\mathbf{x}, \mathbf{y}) = \int_{\mathbf{t}} e^{-j\mathbf{t}^T(\mathbf{x}-\mathbf{y})} dF_{\mathcal{T}}(\mathbf{t})$ for some compactly supported $F_{\mathcal{T}}(\mathbf{t})$ [100]. In contrast to the predefined and fixed kernels in the MMD-GANs, the proposed optimisation on the types of $F_{\mathcal{T}}(\mathbf{t})$ is thus able to learn this important hyperparameter, i.e., the type of kernels. On the other hand, the elliptical distributions in Example 6.1 potentially provide a set of well-defined characteristic kernels, by choosing $F_{\mathcal{T}}(\mathbf{t})$ as a normalised version of the CFs in (6.3). Then, the corresponding real-valued kernels are the density generators in [196].

6.3 Reciprocal Adversarial Learning

6.3.1 Characteristic Function Loss in RCF-GAN

Although the CF loss is a complete metric for measuring any forms of data distributions (e.g., Fig. 6.2-(b)-(d)), the CF loss in (6.4) works more efficiently and effectively in the embedded domain, with higher likelihood of learning fruitful representations of data. To this end, we first express our RCF-GAN in the IPM-GAN format as

$$d(\mathcal{P}_d, \mathcal{P}_g) = \sup_{\mathcal{T}, f \in \mathcal{F}} \mathcal{C}_{\mathcal{T}}(f(\overline{\mathcal{X}}), f(\overline{\mathcal{Y}})), \quad \overline{\mathcal{X}} \sim \mathcal{P}_d \text{ and } \overline{\mathcal{Y}} \sim \mathcal{P}_g, \quad (6.8)$$

Algorithm 3: RCF-GAN. In all the experiments in this work, the generator and the *critic* are trained once at each iteration. The optional *t*-net with the parameter θ_t is designated by $h_{\theta_t}(\cdot)$.

input: Real data distribution $\mathcal{P}_d$; Gaussian noise $\mathcal{P}_N$; batch sizes b_d , b_g , b_t and b_σ for the data, generator input noise, $\mathcal{T}$ and *t*-net input noise, respectively; learning rate l_r ; reciprocal regularisation in the embedded domain λ

output: Net parameters θ_c and θ_g for the *critic* and generator, respectively

while θ_c and θ_g not converge **do**

/ train the critic */*

 Sample from distributions: $\{\bar{\mathbf{x}}_i\}_{i=1}^{b_d} \sim \mathcal{P}_d$; $\{\mathbf{z}_i\}_{i=1}^{b_g} \sim \mathcal{P}_N$; $\{\mathbf{t}_i\}_{i=1}^{b_t} \sim \mathcal{P}_N$;
 $\{\sigma_i\}_{i=1}^{b_\sigma} \sim \mathcal{P}_N$

 Affine transform: $\{\mathbf{t}_i\}_{i=1}^{b_t} \leftarrow (\{\mathbf{t}_i\}_{i=1}^{b_t}, h_{\theta_t}(\{\sigma_i\}_{i=1}^{b_\sigma}))$ // optional

 Calculate adversarial loss: // empirical version of

$$-(\mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{Y}}), \mathbf{Z}) - \mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{X}}), \mathbf{Z}))$$

$$\mathcal{L} = -(\mathcal{C}_{\{\mathbf{t}_i\}_{i=1}^{b_t}}(f_{\theta_c}(g_{\theta_g}(\{\mathbf{z}_i\}_{i=1}^{b_g})), \{\mathbf{z}_i\}_{i=1}^{b_g}) - \mathcal{C}_{\{\mathbf{t}_i\}_{i=1}^{b_t}}(f_{\theta_c}(\{\bar{\mathbf{x}}_i\}_{i=1}^{b_d}), \{\mathbf{z}_i\}_{i=1}^{b_g}))$$

 Update: $\theta_t \leftarrow \theta_t + l_r \cdot \text{Adam}(\theta_t, \nabla_{\theta_t}[\mathcal{L}])$

$$\theta_c \leftarrow \theta_c + l_r \cdot \text{Adam}(\theta_c, \nabla_{\theta_c}[\mathcal{L} + \lambda \sum_{i=1}^{b_g} \|\mathbf{z}_i - f_{\theta_c}(g_{\theta_g}(\mathbf{z}_i))\|_2^2])$$

/ train the generator */*

 Sample from distributions: $\{\bar{\mathbf{x}}_i\}_{i=1}^{b_d} \sim \mathcal{P}_d$; $\{\mathbf{z}_i\}_{i=1}^{b_g} \sim \mathcal{P}_N$; $\{\mathbf{t}_i\}_{i=1}^{b_t} \sim \mathcal{P}_N$;
 $\{\sigma_i\}_{i=1}^{b_\sigma} \sim \mathcal{P}_N$

 Affine transform: $\{\mathbf{t}_i\}_{i=1}^{b_t} \leftarrow (\{\mathbf{t}_i\}_{i=1}^{b_t}, h_{\theta_t}(\{\sigma_i\}_{i=1}^{b_\sigma}))$ // optional

 Calculate adversarial loss: // empirical version of $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{Y}}), f(\bar{\mathbf{X}}))$

$$\mathcal{L} = \mathcal{C}_{\{\mathbf{t}_i\}_{i=1}^{b_t}}(f_{\theta_c}(g_{\theta_g}(\{\mathbf{z}_i\}_{i=1}^{b_g})), f_{\theta_c}(\{\bar{\mathbf{x}}_i\}_{i=1}^{b_d}))$$

 Update: $\theta_g \leftarrow \theta_g + l_r \cdot \text{Adam}(\theta_g, \nabla_{\theta_g}[\mathcal{L}])$

where a distinction is made between the random variables ($\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$) in the data domain and those ($\mathbf{X}$ and $\mathbf{Y}$) in the embedded domain, i.e., $\mathbf{X} =^d f(\bar{\mathbf{X}})$ and $\mathbf{Y} =^d f(\bar{\mathbf{Y}})$. Lemma 6.3 below shows that this metric is well-defined for neural net training.

Lemma 6.3. *The metric $\mathcal{C}_{\mathcal{T}}(\mathbf{X}, \mathbf{Y})$ is bounded and differentiable almost everywhere.*

Proof. Please refer to Appendix 6.5.3. □

Because $\mathcal{C}_{\mathcal{T}}(\cdot, \cdot)$ is bounded by construction, it relaxes the requirements on the *critic* $f \in \mathcal{F}$. Otherwise, we need to bound $\mathcal{F}$ to ensure the existence of the supremum [84].

6.3.2 Matching in the Embedded Space

Having proved that $\mathcal{C}_{\mathcal{T}}(\mathbf{X}, \mathbf{Y}) = 0 \Leftrightarrow \mathbf{X} =^d \mathbf{Y}$, we also need to prove the equivalence between $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{X}}), f(\bar{\mathbf{Y}})) = 0$ and $\bar{\mathbf{X}} =^d \bar{\mathbf{Y}}$, to ensure that our RCF-GAN correctly

learns the real distribution in the data domain. This result is provided in Lemma 6.4.

Lemma 6.4. *Denote the distribution mapping by $\bar{\mathcal{Y}} =^d g(\mathcal{Z})$. Given two functions $f(\cdot)$ and $g(\cdot)$ that map between the supports of $\bar{\mathcal{Y}}$ and $\mathcal{Z}$, if $\mathbb{E}_{\mathcal{Z}}[\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2] = 0$, we also have the reciprocal property $\mathbb{E}_{\bar{\mathcal{Y}}}[\|\bar{\mathbf{y}} - g(f(\bar{\mathbf{y}}))\|_2^2] = 0$, and vice versa. More importantly, this yields the following equivalences: $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{X}}), f(\bar{\mathcal{Y}})) = 0 \Leftrightarrow \mathcal{C}_{\mathcal{T}}(\bar{\mathcal{X}}, \bar{\mathcal{Y}}) = 0 \Leftrightarrow \mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{Y}}), \mathcal{Z}) = 0$ and $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{X}}), \mathcal{Z}) = 0$.*

Proof. Please refer to Appendix 6.5.4. □

As a prerequisite for Lemma 6.4, the co-domains between $f(\cdot)$ and $g(\cdot)$ need to reside on the supports of $\bar{\mathcal{Y}}$ and $\mathcal{Z}$. Otherwise, the reciprocal may not hold. In our RCF-GAN, we propose an anchor design for our *critic*, by rewriting the *critic* loss (by minimising) as $-(\mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{Y}}), \mathcal{Z}) - \mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{X}}), \mathcal{Z}))$. Thus, $\mathcal{Z}$ operates as the static anchor (or pivot) in the dynamic training process. Besides stabilising and improving the convergence in training, this further enables the *critic* to quickly map real data, $\bar{\mathcal{X}}$, to the support of $\mathcal{Z}$, whilst the generator tries to map the generated distribution, $\bar{\mathcal{Y}}$, to the real data, $\bar{\mathcal{X}}$. The adversarial part to maximise $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{Y}}), \mathcal{Z})$ aims to improve the generation quality against the generator loss, i.e., $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathcal{X}}), f(\bar{\mathcal{Y}}))$. Fig. 6.1 illustrates the triangle relationship in our anchor design.

Furthermore, Lemma 6.4 indicates that instead of being regarded as components of some IPMs (e.g., the W-GAN) to be optimised with strict restrictions, the *critic* can be basically regarded as a feature mapping because in the embedded domain the CF loss is a valid distance metric of distributions. The *critic* can then be relaxed to satisfy the reciprocal property. Therefore, we incorporate the auto-encoder in only two modules by interchangeably treating the *critic* as the encoder and the generator as the decoder. More importantly, Lemma 6.4 ensures that matching in the embedded space is sufficient due to $\mathbb{E}_{\mathcal{Z}}[\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2] = 0 \rightarrow \mathbb{E}_{\bar{\mathcal{Y}}}[\|\bar{\mathbf{y}} - g(f(\bar{\mathbf{y}}))\|_2^2] = 0$. This is beneficial in various applications such as the image generation (and

reconstruction), where in the data domain, the MSE loss typically leads to smooth artefacts.

6.3.3 Putting Everything Together

In practice, in Lemma 6.4, we regard $f(\cdot)$ as the *critic* and $g(\cdot)$ as the generator. The t -net is denoted by $h(\cdot)$ and the covariance matrix of its output is assumed to be diagonal (we thus represent it as σ), which is reasonable as in the embedded domain the multiple dimensions tend to be uncorrelated [18]. We also need to clarify that because the t -net is optional and in our RCF-GAN, fixed Gaussian can be directly sampled for $\mathbf{t}$, we separate the t -net from $f(\cdot)$. However, if the t -net is employed, they (the t -net and *critic*) are optimised simultaneously and share the same *critic* loss, i.e., $-(C_{\mathcal{T}}(f(\overline{\mathcal{Y}}), \mathcal{Z}) - C_{\mathcal{T}}(f(\overline{\mathcal{X}}), \mathcal{Z}))$. Moreover, the *critic* additionally minimises an MSE loss to ensure the reciprocal property. On the other hand, the generator is trained by minimising (6.8), as per usual. The pseudo-code for the proposed RCF-GAN is provided in Algorithm 3.

It needs to be pointed out that here we choose $\mathcal{Z}$ as the Gaussian distribution for a fair comparison to other GANs; other complex distributions can be seamlessly adopted in our framework according to different tasks at hand, for example, finite mixture models for un-supervised and semi-supervised classifications, and learnt distributions for sequential data processing.

Remark 6.1. *In addition to the ease of computation, the structure of the proposed RCF-GAN benefits from its interpretation as both a GAN and an auto-encoder, as a way of unifying them. As an auto-encoder, the RCF-GAN enables us to compare reconstructions solely on a meaningful embedded manifold, instead of the standard data domain comparison. When regarded as a GAN, the auto-encoder part theoretically and practically indicates the convergence; it also stabilises the training by pushing the embedded distributions to the static anchor $\mathcal{Z}$.*

6.4 Experimental Results

In this section, the proposed RCF-GAN is evaluated in terms of both image generation (including reconstruction and interpolation), with our code available at https://github.com/ShengxiLi/rcf_gan. We first provide the experimental settings and technical details, followed by performance comparisons with other models on image generation. To evaluate the learnt embedded space within RCF-GAN, the assessments on image reconstruction and interpolation are provided, by using solely the embedded features. We also need to point out that in this work we mainly focus on generating, reconstructing and interpolating images as those are the main criterion for evaluating GANs, and we will provide advanced results on link prediction of graph embedding to show the semantic meaning of our embedded space in Chapter 7. As shown later, the proposed RCF-GAN does not optimise the adjacency matrix which is the ultimate goal of graph learning, but instead achieves superior performances through basic graph convolutional nets.

6.4.1 Settings for Image Generation

Table 6.1: The FID and KID scores obtained from the DCGAN [1] structure. The results of the DCGAN and W-GAN-GP are from [2] and [3]. The corresponding publicly available codes were run to obtain the results of the W-GAN [4], MMD-GAN [5], OCF-GAN and OCF-GAN-GP [6]. The results of the AGE were tested from its pre-trained models [7].

Methods	FID			KID		
	CIFAR-10	Celeba	LSUN_B	CIFAR-10	Celeba	LSUN_B
DCGAN	37.7 [2]	21.4 [2]	70.4 [2]	—	—	—
W-GAN	42.64±0.26	31.85±0.28	57.05±0.37	0.025±0.001	0.023±0.001	0.048±0.002
W-GAN-GP	37.52±0.19 [3]	—	41.39±0.25 [3]	0.026±0.001 [3]	—	0.039±0.002 [3]
MMD-GAN	42.8±0.27	32.5±0.16	56.52±0.34	0.025±0.001	0.024±0.001	0.047±0.002
OCF-GAN	40.99±0.15	32.66±0.16	61.48±0.23	0.024±0.001	0.024±0.001	0.052±0.002
OCF-GAN-GP	33.68±0.21	16.09±0.25	65.18±0.317	0.021±0.001	0.011±0.001	0.060±0.002
AGE	32.54±0.24	23.19±0.14	—	0.020±0.001	0.017±0.001	—
RCF-GAN ^{ln} _(t_norm)	31.55±0.20	19.34±0.22	38.16±0.286	0.019±0.001	0.012±0.001	0.032±0.001
RCF-GAN ^{ln} _(t_net)	31.21±0.21	15.86±0.08	40.15±0.40	0.018±0.001	0.011±0.001	0.034±0.001
RCF-GAN ^{sn} _(t_net)	32.65±0.27	13.71±0.16	29.15±0.21	0.019±0.001	0.009±0.001	0.024±0.001

Note: Symbol t_norm corresponds to use the fixed Gaussian samples and t_net to the t_net , ln denotes the layer normalisation and sn is the spectral normalisation [80].

Datasets: Three widely applied benchmark datasets were employed in the evaluation: CelebA (faces of celebrities) [207], CIFAR-10 [208] and LSUN Bedroom

(LSUN_B) [209]. The images of the CelebA and LSUN_B were cropped to the size 64×64 and 128×128 , whilst the image size of the CIFAR10 was 32×32 . When evaluating the reconstruction, the test sets of the CIFAR10 and LSUN_B were employed, of which the samples were not used in the training.

Baselines: As our work is mainly related to the IPM-GANs, we compared our RCF-GAN with the W-GAN [4], W-GAN with gradient penalty (W-GAN-GP) [79] and MMD-GAN [3,5]. As an advancement of the MMD-GAN, the most recent work, OCF-GAN [6], together with its gradient penalty version (OCF-GAN-GP) was also compared. We need to point out that all the results reported in [6] were evaluated for the image size of 32×32 . We thus ran the experiments for the CelebA and LSUN_B for image sizes 64×64 and 128×128 by using its provided code. For image reconstruction, we compared our RCF-GAN with the recent adversarial generator-encoder (AGE) work [7], which empirically performs better than the adversarially learned inference (ALI) [92].

Metrics: The Fréchet inception distance (FID) [2] was employed as a performance metric, which is basically the Wasserstein distance between two Gaussian distributions, together with the kernel inception distance (KID) that arises from the MMD metric [3]. In evaluating the FID and KID scores, we randomly generated 25,000 samples for both generation and true images, and obtained these metrics, in terms of the mean and standard deviation, by 10 times repeated random selections and calculations.

Net structure and technical details: For a fair comparison, all the reported results were compared under the batch sizes of 64 (i.e., $b_d = b_g = b_t = b_\sigma = 64$). Moreover, all variances of Gaussian noise were set to 1, except for the input noise of the generator that was 0.3, because the reciprocal loss had to be minimised given the fact that the output of the *critic* is restricted to $[-1, 1]$. Furthermore, we do not require the Lipschitz constraint, which allows for a relatively larger learning rate ($l_r = 0.0002$ for both nets). Moreover, for the CIFAR10 and LSUN_B datasets, the dimension of the embedded domain was set to 128 and for the CelebA dataset

the dimension was 64. The optional *t*-net, if used, was a small three layer fully connected net, with the dimension of each layer being the same as the embedded dimension. It needs to be pointed out that except for several specific experiments (e.g., ablation study) aiming at in-depth analysis on our RCF-GAN, we set λ to linearly space from 0.6 to 1, which is reasonable as the role of *critic* changes from a discriminator to an encoder along the training procedure. Our default RCF-GAN used *t*-net and layer normalisation, and was trained with the vanilla CF loss (i.e., $\alpha = 0.5$ in (6.6)).

Next, we first provide in Section 6.4.2 a comprehensive assessment under the DCGAN structure [1] for a fair comparison with the existing GANs. To reveal the physical meaning of our CF loss, we also investigated effects of unequally weighting amplitude and phase in training our RCF-GAN in Section 6.4.3. Then, the reconstruction and interpolation of the proposed RCF-GAN is evaluated in Section 6.4.4, followed by the ablation study on key components adopted in the RCF-GAN in Section 6.4.5. Furthermore, Section 6.4.6 shows that the proposed RCF-GAN is scalable to various net structures and it achieved the state-of-the-art performances under the ResNet structure [210]. The performance of RCF-GAN could be further improved by e.g., progressive training [211], using individual learning rates [2], or adding noise to discriminators [21]. All experiments were run on a single Nvidia-GeForce-RTX2080 graphics card, except for the one of ResNet under image size 128×128 , which required two Nvidia-GeForce-RTX2080 graphics cards.

6.4.2 Random Image Generation under DCGAN Structure

The images generated from random Gaussian noise are shown in Fig. 6.3. Observe that by virtue of the proposed CF loss in the RCF-GAN, the generated images are clear and close to the real images; the FID and KID scores are further provided in Table 6.1. This table shows that the proposed RCF-GAN consistently achieved the best performances across the three datasets. The OCF-GAN-GP achieved comparable generation performance on the CelebA dataset, but had relatively inferior perfor-

manances compared to our RCF-GAN on the CIFAR-10 and LSUN_B datasets. Thus, although the most recent independent work, OCF-GAN, also adopts the characteristic function in designing the loss, it still operates under the MMD-GAN framework, without the interpretation of the physical meaning of the characteristic function and the consideration of the t -net proposed in this chapter. More importantly, the reciprocal structure introduced in this chapter, together with the proposed CF loss, significantly improves the image generation performance.

By inspecting the achieved best performances of RCF-GAN, the use of the t -net in outputting optimal $F_{\mathcal{T}}(\mathbf{t})$ proved beneficial. Moreover, we set the default RCF-GAN to use t -net and layer normalisation, which is also a common and fair choice in the existing IPM-GANs [79]. We also show that using the spectral normalisation can further improve the performance of the proposed RCF-GAN. In the default setting, our *critic* and generator were evaluated under almost the same number of model parameters as W-GANs, whereas MMD-GANs need an extra decoder net. The only extra cost in our t -net is negligible because it is a 3-layer fully connected net with the dimension of each layer less than 128.

More importantly, compared to a fluctuated generator loss that is caused by the adversarial module in GANs, we take the advantages of the auto-encoder structure in utilising the reciprocal loss (i.e., $\mathbb{E}_{\mathbf{Z}}[\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2]$ indicates the reciprocal loss in the embedded space), together with the distance between the embedded real distribution $f(\overline{\mathcal{X}})$ and the Gaussian distribution $\mathcal{Z}$ (i.e., $\mathcal{C}_{\mathcal{T}}(f(\overline{\mathcal{X}}), \mathcal{Z})$) to better indicate the convergence, as shown in Figure 6.3. Intuitively, the reciprocal loss measures the convergence on reconstructions, whereas the real image embedding distance $\mathcal{C}_{\mathcal{T}}(f(\overline{\mathcal{X}}), \mathcal{Z})$ indicates the performance on generating images.

6.4.3 Phase and Amplitude in the CF Loss

Besides the illustrative examples in Fig. 6.2, the effects of α in (6.6) on training the overall RCF-GAN are shown in Fig. 6.4. Observe that the proposed RCF-

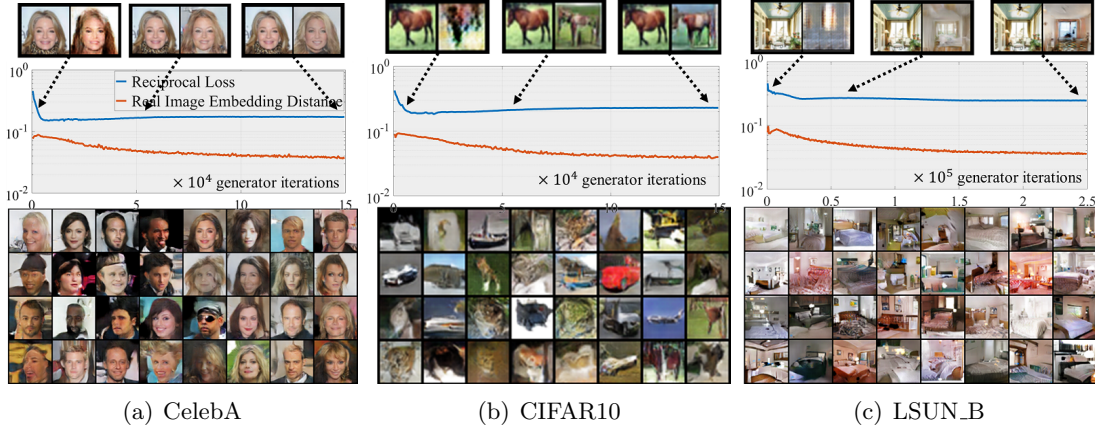


Figure 6.3: Convergence curves and the images generated by the proposed RCF-GAN from Gaussian noise, under the DCGAN [1] structure. The curves represent an average over a moving window, with 500 iterations.

GAN is robust to the choice of α , as when α ranges from 0.1 to 0.9, the RCF-GAN still achieved relatively superior generations. We did not witness any mode collapse generations in any experiment. Although $\alpha = 0.5$ was a default and mainly used in our experiments, varying α could achieve even better performances. For example, for the dataset without complex and diversified scenarios (e.g., CelebA), imposing amplitude by $\alpha = 0.75$ increased the FID (KID) from 15.86 (0.011) to 13.84 (0.009). As the amplitude relates to the diversity measurement in the CF loss, the increment may come from enhancing the richness of generated faces. On the other hand, for some complicated scenarios (e.g., CIFAR-10), keeping the mean of data generation (that is, focusing on the phase) could be more beneficial (e.g., $\alpha = 0.1$).

Fig. 6.4 further shows an illustrative example on some over-weighted examples from CelebA. When over-weighting the phase ($\alpha = 0.1$), the generated images tend to be whitened and blurred, with their interpolation less smooth. This indicates that the RCF-GAN tended to learn the average (mean) information of the data. On the contrary, when the amplitude was over-weighted ($\alpha = 0.9$), the generated images were over-saturated and with noisy artefacts, meaning that the RCF-GAN was likely to learn diversified content, even though some learnt faces were inaccurate. Therefore, the physical meaning of the proposed CF loss can provide a feasible way of understanding and evaluating generation details where the KID and FID metrics

cannot reflect.

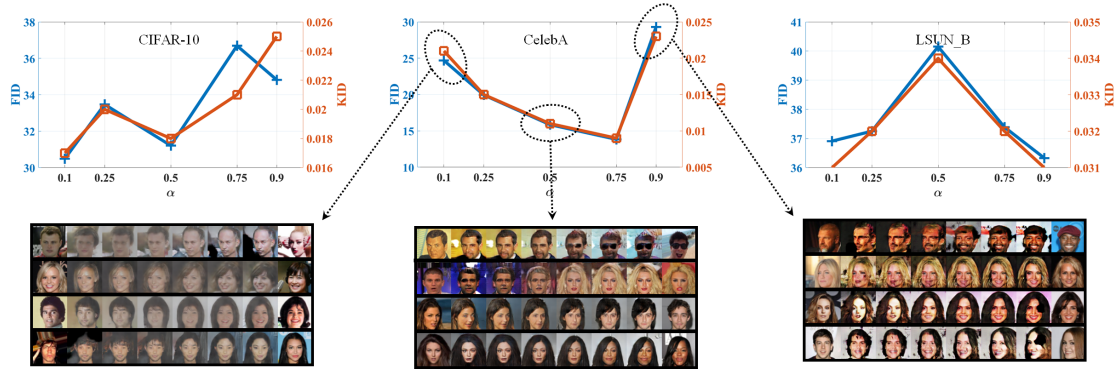


Figure 6.4: The FID and KID scores for different α , under the DCGAN [1] structure. Observe the embedded space (by interpolated images) of the proposed RCF-GAN, which was learnt with different α on CelebA dataset. The details of image interpolation are given in Section 6.4.4.

6.4.4 Image Reconstruction and Interpolation under DCGAN Structure

Benefiting from the reciprocal requirement introduced in Lemma 6.4, the proposed RCF-GAN can also reconstruct images and learn a semantic meaningful space. Images reconstructed and interpolated by RCF-GAN, AGE and MMD-GAN are shown in Fig. 6.5. As seen from this figure, because the RCF-GAN matches the distributions only in the embedded domain, the reconstructed images are clear and semantically meaningful, resulting in superior interpolation and reconstruction. This is beneficial because besides randomly generating real images, our RCF-GAN is able to bi-directionally reconstruct and interpolate real images. In contrast, although MMD-GANs employ a third module to implement an auto-encoder, the decoded images are severely blurred.

The proposed RCF-GAN also achieved better subjective reconstruction and interpolation than the AGE, by generating less blurred and more accurate images (for example, correct skin and hair colours). This is quantified in Table 6.2, which shows that the images reconstructed by our RCF-GAN are superior to those from the AGE. More importantly, by comparing with the FID and KID scores in Table 6.1, the images from the proposed RCF-GAN are consistently superior, whilst the qual-

Table 6.2: The FID and KID scores of reconstructed images under the DCGAN structure. The AGE results were tested from its pre-trained models [7].

Methods	FID		
	CIFAR-10	Celeba	LSUN_B
AGE	47.37 $\pm$ 0.32	30.77 $\pm$ 0.19	—
RCF-GAN	28.70$\pm$0.16	14.82$\pm$0.12	44.16 $\pm$ 0.42
Methods	KID		
	CIFAR-10	Celeba	LSUN_B
AGE	0.022 $\pm$ 0.001	0.024 $\pm$ 0.001	—
RCF-GAN	0.014$\pm$0.001	0.009$\pm$0.000	0.036 $\pm$ 0.001

ity of the reconstructed images by the AGE is significantly inferior to its randomly generated images. This also indicates the effectiveness of the unified structure of our RCF-GAN.

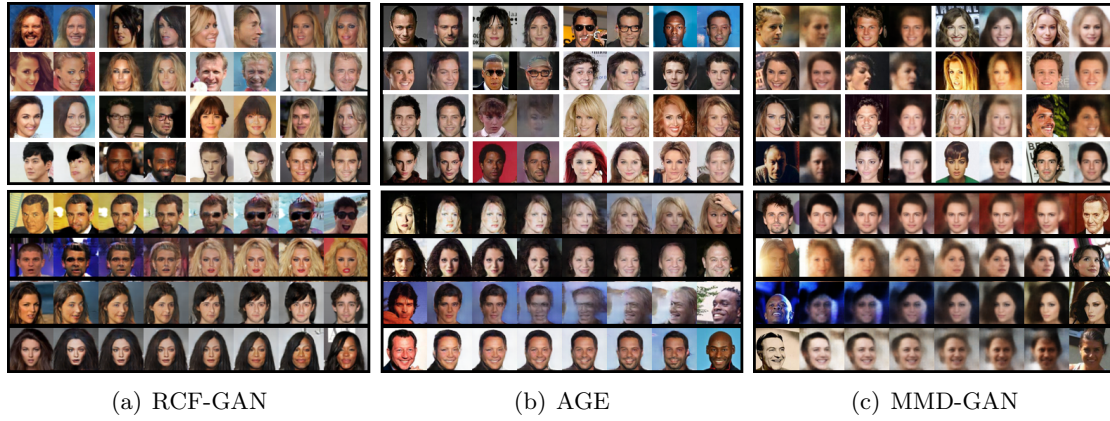


Figure 6.5: Image reconstruction (upper panel) and interpolation (lower panel) by the proposed RCF-GAN, AGE [7] and MMD-GAN [5] for the CelebA dataset, under the DCGAN [1] structure. The upper panel shows the reconstructed images (in even columns) corresponding to the original images (in odd columns). The lower panel displays the linear interpolation in the embedded domain.

6.4.5 Ablation Study

The roles of the two key distinguishing elements, the term λ that controls the reciprocal and the anchor design, of the proposed RCF-GAN are next evaluated via an ablation study on the CelebA dataset. The results in Table 6.3 shows that without the reciprocal loss (by setting $\lambda = 0$) the overall generation was largely degraded and the reconstruction even completely failed. This, on the one hand, highlights the

Table 6.3: Ablation study on the CelebA dataset. The proposed RCF-GAN was evaluated and compared to the one without (w/o) reciprocal requirement ($\lambda = 0$) and without anchor design.

FID			
	w/o reciprocal	w/o anchor	RCF-GAN
G.	59.39 $\pm$ 0.37	17.80 $\pm$ 0.20	15.86 $\pm$ 0.08
R.	>100	>100	14.82 $\pm$ 0.12
KID			
	w/o reciprocal	w/o anchor	RCF-GAN
G.	0.046 $\pm$ 0.001	0.010 $\pm$ 0.000	0.011 $\pm$ 0.001
R.	>0.060	>0.060	0.009 $\pm$ 0.000

Note: G. stands for random image generation and R. for image reconstruction.

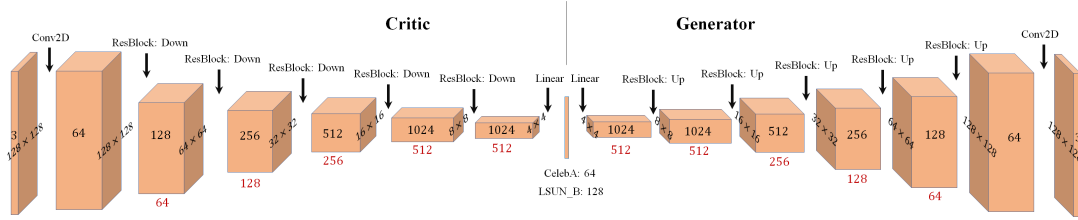


Figure 6.6: The ResNet 128×128 structure adopted in this work. The numbers in red colour represent the channel setting of ResNet 64×64 .

necessity of the reciprocal loss in our work; on the other hand, it also validates the correctness of the theoretical guarantee in Lemma 6.4, which is an important requirement that also motivates the auto-encoder structure in our RCF-GAN. Moreover, Table 6.3 also validates the effectiveness of the proposed anchor design. Without the anchor design, the generation still works, however, because the minimisation of $\mathcal{C}_{\mathcal{T}}(f(\overline{\mathcal{X}}), \mathcal{Z})$ by the anchor does no longer exist, the mapping of real images $f(\overline{\mathcal{X}})$ might not completely fall into the support of $\mathcal{Z}$, thus leading to poor reconstructions. Therefore, in order to successfully generate and reconstruct images, the reciprocal and anchor architecture are necessary in the proposed RCF-GAN.

6.4.6 Advancements under ResNet Structure

The scalability of the proposed RCF-GAN was further evaluated over complex net structures and larger image sizes. Specifically, we trained RCF-GAN under the

Table 6.4: The FID and KID scores under the ResNet structure, for 64×64 and 128×128 image sizes. The result of Sphere GAN was obtained from the original article [8]. We ran the available code of the OCF-GAN-GP [6] with its implemented ResNet structure because it failed to converge in our structure given in Fig. 6.6.

64×64	FID		KID	
	Celeba	LSUN_B	Celeba	LSUN_B
Sphere GAN	—	16.9 [8]	—	—
RCF-GAN	9.02 ± 0.22	8.76 ± 0.07	0.006 ± 0.001	0.005 ± 0.001
RCF-GAN (R.)	8.06 ± 0.08	7.89 ± 0.05	0.003 ± 0.000	0.002 ± 0.000
128×128				
OCF-GAN-GP	20.78 ± 0.15	21.82 ± 0.20	0.015 ± 0.001	0.014 ± 0.001
RCF-GAN	10.71 ± 0.11	10.32 ± 0.13	0.006 ± 0.000	0.005 ± 0.001
RCF-GAN (R.)	13.01 ± 0.15	8.64 ± 0.10	0.006 ± 0.000	0.003 ± 0.000

Note: R. stands for image reconstruction.



Figure 6.7: Random generation, reconstruction and interpolation by the proposed RCF-GAN, using ResNet and images of size 128×128 pixels. The upper panel shows images of the CelebA dataset and the lower panel images from the LSUN_B dataset.

ResNet structure, over image sizes of 64×64 and 128×128 . The ResNet structure for image size 64×64 , was exactly the same as that in [79]. We extended this structure to the image size of 128×128 in a similar way to the DCGAN, as is shown⁴ in Fig. 6.6. We adopted spectral normalisation instead of the layer normalisation in the ResNet experiments; we refer to our implementations for more detail.

The FID and KID scores are given in Table 6.4, and the results of randomly generating, reconstructing and interpolating 128×128 images are provided in Fig. 6.7. Again, it is obvious that the proposed RCF-GAN achieved the lowest (and best) FID and KID scores, compared to the existing state-of-the-art models. The generated images given in Fig. 6.7 also validate the fidelity and superior performances of our generation. More importantly, under a more powerful net structure than the DCGAN, the proposed RCF-GAN learned a more semantically meaningful embedded space, thus reconstructing and interpolating images in a highly realistic style. Therefore, the proposed RCF-GAN, only requiring two network modules with a simple training strategy, achieved the above superior generation, reconstruction and interpolation in a stable and scalable way, compared to other GANs that either require additional nets (e.g., ALI and MMD-GAN) or employ complicated training strategies (e.g., W-GAN and MMD-GAN for training the *critic* and generator at different times).

6.5 Appendix

6.5.1 Proof of Lemma 6.1

We here prove the non-negativity, symmetry and triangle properties that are required as a valid distance metric.

⁴Please note that the *critic* of our ResNet 128×128 structure is slightly different from that in the spectral GAN [80]. We adopted a symmetric (mirror) structure of the generator, whereby the spectral GAN used an asymmetric one. Although the RCF-GAN still works under the structure of the spectral GAN, we believe that the mirror structure can well reflect the proposed reciprocal idea and also represents a natural extension of the ResNet 64×64 in [79]. The parameter size in our ResNet structure is slightly smaller than that in the spectral GAN.

Non-negativity: Based on the definition of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ in (6.4), the term $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ is non-negative, because $c(\mathbf{t}) \geq 0$ for all $\mathbf{t}$. We next prove that the equality holds for:

- $\mathcal{X} =^d \mathcal{Y} \rightarrow \mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = 0$: This is evident because $\Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ for all $\mathbf{t}$.
- $\mathcal{X} =^d \mathcal{Y} \leftarrow \mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = 0$: Given that the support of $\mathcal{T}$ is $\mathbb{R}^m$, $\int_{\mathbf{t}} \sqrt{c(\mathbf{t})} dF_{\mathcal{T}}(\mathbf{t}) = 0$ exists if and only if $c(\mathbf{t}) = 0$ everywhere. Therefore, $\Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ for all $\mathbf{t} \in \mathbb{R}^m$. According to the *Uniqueness Theorem* of the CF, we have $\mathcal{X} =^d \mathcal{Y}$.

Therefore, $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) \geq 0$, and the equality holds if and only if $\mathcal{X} =^d \mathcal{Y}$.

Symmetry: This is obvious for the symmetry of $c(\mathbf{t})$, thus yielding $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = \mathcal{C}_{\mathcal{T}}(\mathcal{Y}, \mathcal{X})$.

Triangle inequality: Because the CFs $\Phi_{\mathcal{X}}(\mathbf{t})$ and $\Phi_{\mathcal{Y}}(\mathbf{t})$ are elements of the normed vector space, we have the following inequality (also known as the Minkowski inequality),

$$\begin{aligned} & \int_{\mathbf{t}} |\Phi_{\mathcal{X}}(\mathbf{t}) - \Phi_{\mathcal{Z}}(\mathbf{t}) + \Phi_{\mathcal{Z}}(\mathbf{t}) - \Phi_{\mathcal{Y}}(\mathbf{t})| dF_{\mathcal{T}}(\mathbf{t}) \\ & \leq \int_{\mathbf{t}} |\Phi_{\mathcal{X}}(\mathbf{t}) - \Phi_{\mathcal{Z}}(\mathbf{t})| dF_{\mathcal{T}}(\mathbf{t}) + \int_{\mathbf{t}} |\Phi_{\mathcal{Z}}(\mathbf{t}) - \Phi_{\mathcal{Y}}(\mathbf{t})| dF_{\mathcal{T}}(\mathbf{t}). \end{aligned} \quad (6.9)$$

Therefore, the triangle property of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ follows as

$$\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) \leq \mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Z}) + \mathcal{C}_{\mathcal{T}}(\mathcal{Z}, \mathcal{Y}). \quad (6.10)$$

This means that $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ is a valid distance metric for measuring discrepancies between two random variables $\mathcal{X}$ and $\mathcal{Y}$.

This completes the proof.

6.5.2 Proof of Lemma 6.2

The proof of the triangle and symmetry properties is the same as those for Lemma 6.1. The non-negativity is also evident (the same as that in Lemma 6.1), but the equality holds for different conditions.

Before proceeding with the proof, we first quote Theorem 3 from Essen [212].

Theorem 3 ([212]) *The distributions of two random variables $\mathcal{X}$ and $\mathcal{Y}$ are the same when*

- $\Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ in an interval around $\mathbf{0}$;
- $\beta_k = \int_{\mathcal{X}} x^k dF_{\mathcal{X}}(x) < \infty$ for $k = 0, 1, 2, 3, \dots$
- $\sum_{k=1}^{\infty} 1/\beta_{2k}^{1/2k}$ diverges, which means that the moment problem of β_k is determined and unique.

It is the fact that only requiring $\Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ in an interval around $\mathbf{0}$ does not ensure the equivalence between two distributions without any other constraints, also given in the counterexample provided in [212]. This equivalence cannot be ensured even when all the moments are matched. The third condition, intuitively, guarantees this equivalence by a restriction that the moment does not increase “extremely” fast when $k \rightarrow \infty$.

In Lemma 6.2 of this work, we bound $\mathcal{X}$ and $\mathcal{Y}$ by $[-1, 1]$, thus having $|\beta_k| \leq 1 < \infty$ and $1/\beta_{2k}^{1/2k} \geq 1$ so that $\sum_{k=1}^{\infty} 1/\beta_{2k}^{1/2k}$ diverges. In this case, according to Theorem 3, we have $\Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ when $\mathbf{t}$ samples around $\mathbf{0} \rightarrow \mathcal{X} =^d \mathcal{Y}$. Conversely, it is obvious that $\mathcal{X} =^d \mathcal{Y} \rightarrow \Phi_{\mathcal{X}}(\mathbf{t}) = \Phi_{\mathcal{Y}}(\mathbf{t})$ for all $\mathbf{t}$. Therefore, for bounded $\mathcal{X}$ and $\mathcal{Y}$, sampling around $\mathbf{0}$ is sufficient to ensure the symmetry, triangle, non-negativity (together with the uniqueness when the equality holds) properties of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$.

This completes the proof.

6.5.3 Proof of Lemma 6.3

We first show the boundedness of $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y})$ by observing

$$\begin{aligned} 0 \leq \mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) &= \int_{\mathbf{t}} |\Phi_{\mathcal{X}}(t) - \Phi_{\mathcal{Y}}(t)| dF_{\mathcal{T}}(\mathbf{t}) \\ &\leq \int_{\mathbf{t}} |\Phi_{\mathcal{X}}(t)| dF_{\mathcal{T}}(\mathbf{t}) + \int_{\mathbf{t}} |\Phi_{\mathcal{Y}}(t)| dF_{\mathcal{T}}(\mathbf{t}) \leq 1 + 1 = 2, \end{aligned} \quad (6.11)$$

where the second inequality is obtained via the Minkowski inequality and the third one by the fact that the maximum modulus of the CF is 1. It should be pointed out that this property is important and advantageous because in this way our cost becomes bounded automatically. Otherwise, we may need to bound $f \in \mathcal{F}$ to ensure an existence of the supremum of some IPMs (such as the dual form of the Wasserstein distance used in the W-GAN).

To prove the differentiable property, we first expand $c(\mathbf{t})$ in $\mathcal{C}_{\mathcal{T}}(\mathcal{X}, \mathcal{Y}) = \int_{\mathbf{t}} \sqrt{c(\mathbf{t})} dF_{\mathcal{T}}(\mathbf{t})$ as

$$c(\mathbf{t}) = (\operatorname{Re}\{\Phi_{\mathcal{X}}(\mathbf{t})\} - \operatorname{Re}\{\Phi_{\mathcal{Y}}(\mathbf{t})\})^2 - (\operatorname{Im}\{\Phi_{\mathcal{X}}(\mathbf{t})\} - \operatorname{Im}\{\Phi_{\mathcal{Y}}(\mathbf{t})\})^2, \quad (6.12)$$

where $\operatorname{Re}\{\Phi_{\mathcal{X}}(\mathbf{t})\} = \mathbb{E}_{\mathcal{X}}[\cos(\mathbf{t}^T \mathbf{x})]$ denotes the real part of the CF and $\operatorname{Im}\{\Phi_{\mathcal{X}}(\mathbf{t})\} = \mathbb{E}_{\mathcal{X}}[\sin(\mathbf{t}^T \mathbf{x})]$ for its imaginary part. Therefore, by regarding $c(\mathbf{t})$ as a mapping $\mathbb{R}^m \rightarrow \mathbb{R}$, it is differentiable almost everywhere⁵.

This completes the proof.

6.5.4 Proof of Lemma 6.4

Because $\mathbb{E}_{\mathcal{Z}}[\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2] = 0$ and $\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2 \geq 0$, we have $\mathbf{z} = f(g(\mathbf{z}))$ for any $\mathbf{z}$ and $g(\mathbf{z})$ under the supports of $\mathcal{Z}$ and $\overline{\mathcal{Y}}$, respectively. We can obtain $g(\mathbf{z}) = g(f(g(\mathbf{z})))$ under the supports of $\mathcal{Z}$ and $\overline{\mathcal{Y}}$ as well; given that $\overline{\mathbf{y}} = g(\mathbf{z})$ by

⁵We note that $c(\mathbf{t})$ is not necessarily complex differentiable because it does not satisfy the Cauchy-Riemann equations. In other words, *nonconstant purely real-valued functions* are not complex differentiable because their Cauchy-Riemann equations are not satisfied. However, in our case, $\mathcal{C}_{\mathcal{T}}$ is differentiable as it is treated within the $\mathbb{C}\mathbb{R}$ calculus. Please refer to [206, 213] for more detail on the $\mathbb{C}\mathbb{R}$ calculus.

the definition, this results in $\bar{\mathbf{y}} = g(f(\bar{\mathbf{y}}))$. Then, we have $\mathbb{E}_{\bar{\mathbf{y}}}[\|\bar{\mathbf{y}} - g(f(\bar{\mathbf{y}}))\|_2^2] = 0$. Therefore, the function $g(\cdot)$ is a unique inverse of the function $f(\cdot)$, and vice versa, which also indicates that the two functions are bijective.

The bijection of the function $f(\cdot)$ possesses many desirable properties between the domains of $\bar{\mathbf{y}}$ and $\mathbf{z}$, thus ensuring the equivalences between their CFs. More specifically, without loss of generality, we assume $\mathcal{C}_{\mathcal{T}}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = 0$, which means

$$\int_{\bar{\mathbf{x}}} e^{j\bar{\mathbf{t}}^T \bar{\mathbf{x}}} dF_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \int_{\bar{\mathbf{y}}} e^{j\bar{\mathbf{t}}^T \bar{\mathbf{y}}} dF_{\bar{\mathbf{y}}}(\bar{\mathbf{y}}), \quad \text{for all } \bar{\mathbf{t}}. \quad (6.13)$$

Then, given the bijection $f(\cdot)$ by $\mathbf{x} = f(\bar{\mathbf{x}})$ and $\mathbf{y} = f(\bar{\mathbf{y}})$, we obtain $\mathbf{x} = f(\bar{\mathbf{x}}) = f(\bar{\mathbf{y}}) = \mathbf{y} \Leftrightarrow \bar{\mathbf{x}} = \bar{\mathbf{y}}$, for any realisations $\mathbf{x}$ and $\mathbf{y}$ from $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$. Then, the following equivalence between the CFs of $\mathbf{x} = f(\bar{\mathbf{x}})$ and $\mathbf{y} = f(\bar{\mathbf{y}})$ holds

$$\begin{aligned} \int_{\bar{\mathbf{x}}} e^{j\bar{\mathbf{t}}^T \bar{\mathbf{x}}} dF_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) &= \int_{\bar{\mathbf{y}}} e^{j\bar{\mathbf{t}}^T \bar{\mathbf{y}}} dF_{\bar{\mathbf{y}}}(\bar{\mathbf{y}}), \quad \text{for all } \bar{\mathbf{t}} \\ \Leftrightarrow \int_{\bar{\mathbf{x}}} e^{j\bar{\mathbf{t}}^T f(\bar{\mathbf{x}})} dF_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) &= \int_{\bar{\mathbf{y}}} e^{j\bar{\mathbf{t}}^T f(\bar{\mathbf{y}})} dF_{\bar{\mathbf{y}}}(\bar{\mathbf{y}}), \quad \text{for all } \bar{\mathbf{t}} \\ \Leftrightarrow \int_{\mathbf{x}} e^{j\bar{\mathbf{t}}^T \mathbf{x}} dF_{\mathbf{x}}(\mathbf{x}) &= \int_{\mathbf{y}} e^{j\bar{\mathbf{t}}^T \mathbf{y}} dF_{\mathbf{y}}(\mathbf{y}), \quad \text{for all } \bar{\mathbf{t}}. \end{aligned} \quad (6.14)$$

Therefore, we have $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{x}}), f(\bar{\mathbf{y}})) = 0$. Furthermore, we also have $f(\bar{\mathbf{y}}) =^d \mathbf{z}$ due to $\mathbb{E}_{\mathbf{z}}[\|\mathbf{z} - f(g(\mathbf{z}))\|_2^2] = 0$. Hence the following equivalences: $\mathcal{C}_{\mathcal{T}}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = 0 \Leftrightarrow \mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{x}}), f(\bar{\mathbf{y}})) = 0 \Leftrightarrow \mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{y}}), \mathbf{z}) = 0$ and $\mathcal{C}_{\mathcal{T}}(f(\bar{\mathbf{x}}), \mathbf{z}) = 0$.

This completes the proof.

6.6 Summary

We have introduced an efficient GAN structure that seamlessly combines the IPM-GANs and auto-encoders. In this way, the reciprocal in the proposed RCF-GAN ensures an equivalence between the embedded and data domains, whereas in the embedded domain the comparison of two distributions is rigorously supported by the proposed powerful CF loss, together with the physically meaningful phase and am-

plitude information, and an efficient sampling strategy. The reciprocal, accompanied with the proposed anchor design, has been shown to also stabilise the convergence of the adversarial learning in the proposed RCF-GAN, and at the same time to benefit from meaningful comparisons in the embedded domain. Consequently, the experimental results have demonstrated the superior performances of the proposed RCF-GAN in tasks on image learning tasks.

Chapter 7

Application in Graph Link Prediction

In many situations where the graph topology cannot be directly observed or even when the data is partially observed, the inference of graph structure is a key first step. Unfortunately, given the observed graph data, graph learning is an ill-posed problem, of which a reasonable solution requires some priors. Thus, it is natural to connect and summarise those techniques under the umbrella of probabilistic generative models. We herein propose to an advanced graph RCF-GAN (GRCF-GAN) for graph topology inferencing, with the state-of-the-art performances achieved over a range of applications.

7.1 Introduction

Oftentimes, it is necessary to establish additional connections, in addition to the existing edges in a graph, such as in the cases of graph completion (interpolation) and graph extension (expansion) [214, 215]. This is achieved through so called link prediction. A direct way to perform link prediction would be to employ some heuristic similarity method on the nodes and sub-graphs in order to predict the missing

links between nodes, as is the case with PageRank, SimRank [216] and SEAL [217]. Alternatively, we can employ learning approaches to infer such links in some “well-behaved” embedded spaces of graphs; this is highly related to the field of graph representation learning, that is, learning a representative latent space given an existing graph. The spectrum of a graph is a simple yet effective candidate for an embedding space, since it reflects smoothness (frequency) of graphs. In this way, spectral clusters can be utilised to train a classifier to predict links [10]. In addition to spectral clustering, more advanced methodologies include Deepwalk [218], Node2vec [219] and Line [220], all of which learn meaningful and continuous low-dimensional latent spaces by preserving (encoding) neighbouring information of nodes. For more detail, we refer to some recent well-documented reviews [221–223].

More recently, the graph convolutional net (GCN), a powerful tool for implicitly processing local information in graphs, has also been applied to link prediction tasks, through probabilistic generative models. Within GCNs, two types of generative models are typically used, graph variational auto-encoder (VAE) and graph generative adversarial model (GAN) [224–226]. Autoregressive models have also been considered to progressively generate graphs [227, 228]. We here focus on the VAE-based methods because they are straightforwardly designed to learning representations for link prediction, while GAN related methods are motivated by graph generation. We should also point out that the two VAE- and GAN- based approaches are not independent, as some VAE-based works take advantage of additional adversarial modules to improve learning capacity [229, 230]. The proposed RCF-GAN in Chapter 6, being a seamless combination of auto-encoder and GANs, however, is also a promising candidate for graph generation, part our future work.

The graph VAE [11] employs the VAE framework proposed in [18], with the underpinning idea that the data (or the graph connections in graph VAEs) are also generated by Gaussian random samples. However, different from the linear model approaches, the VAE employs a neural net (decoder) as a non-linear way of image generation. To avoid trivial generation from random noise, we need to find a rea-

sonable set that is likely to generate meaningful signals, thus leading to an encoder structure which minimises its distance to the true posterior. We refer to [231] for a detailed tutorial on the VAE.

The initial attempt, called the variational graph auto-encoder (VGAE) [11], uses the connection information to the encoder, implemented as a GCN, whilst the decoder employs a simple inner product operation to reconstruct the adjacency matrix. To enable a GCN in the decoder and complete the theory, work in [232] proposes an intermediate and learnable adjacency matrix within the decoder. Other improvements include regularising VGAE through semantic validity [233], use of rich models (such as mixture models) as a prior [234], and an asynchronous message passing scheme for directed acyclic graphs [235].

A potential problem with the VGAE is that its reconstruction loss forces the training of the VGAE to recover an incomplete training adjacency matrix, whereby the test and validation edges are masked out. Therefore, if a perfect reconstruction was achieved in training, in the test stage, the VGAE would only recover the training edges, whilst ignoring the test and validation edges. The way of relieving this issue by the VGAE lies in its usage of variational inference i.e., the representation (embedding) of each graph node as a Gaussian distribution given by the learnt mean and covariance. This, however, adds disturbance and difficulties to the decoder when reconstructing, increasing the reconstruction loss in an implicit adversarial way and forcing the VGAE to learn robust and meaningful graph representations.

On the other hand, because the proposed RCF-GAN in Chapter 6 can learn a meaningful representation in the embedded space, graph links can be directly predicted by the embedded features, instead of being reconstructed by the decoder as the VGAE does. Thus, the graph RCF-GAN (GRCF-GAN) in graph link prediction differs from the VGAE in two main aspects: 1) the decoder within the GRCF-GAN no longer reconstructs the adjacency matrix but outputs the node features, to satisfy the reciprocal requirement of the proposed GRCF-GAN; 2) the reconstruction loss in GRCF-GAN is based on node features, in the embedded space, instead of

the adjacency matrix in the data domain of the VGAE. The so learnt embedded features are therefore graph representations and generalise well, thus enabling link prediction via our GRCF-GAN to effectively avoid overfitting to the training edges. Experimental results show that the embedded space learnt from only reconstructing graph node features can accurately predict graph links via a fixed inner product operation, in this way outperforming the existing VGAEs [11] that aim to reconstruct graph adjacency matrices.

7.2 Probabilistic Generative Models in Graph Link Prediction

Generally speaking, a graph consists of a set of vertices (nodes), $\mathcal{V}$, connected by a collection of edges $\mathbf{A} \in |\mathcal{V}| \times |\mathcal{V}|$, where $|\mathcal{V}|$ represents the cardinality of the set $\mathcal{V}$. The connection by edges is typically denoted by an adjacency matrix $\mathbf{A}$, where its element $A_{i,j} > 0$ if nodes $\mathcal{V}_i$ and $\mathcal{V}_j$ are connected, and $A_{i,j} = 0$ otherwise. Alternatively, the Laplacian matrix $\mathbf{L}$ can also be employed to represent the connectivity, by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is called the degree matrix and is diagonal by $D_{i,i} = \sum_j A_{i,j}$. Because the diagonal matrix $\mathbf{D}$, the non-diagonal elements in $\mathbf{L}$ are non-positive. When the graph is undirected, an important property of the Laplacian matrix is that it is positive semi-definite (psd), which allows for eigen-decomposition and definitions of spectrum in graphs [236].

Graph data analytics with known or given typologies $\mathbf{A}$ or $\mathbf{L}$ is feasible for applications that involve physically meaningful structures, such as citation networks, transport networks and observable social networks. In those applications, the graph convolutional nets (GCNs) and other spectral techniques have been successfully implemented and developed to filter, analyze or visualize graph signals [237]. However, in many situations where the graph topology cannot be directly observed and sometimes even when the data is partially observed, the inference of graph structure is a key first step, because different graph structures can lead to totally different types

of results.

Unfortunately, given the observed graph data, graph learning is an ill-posed problem. In other words, different types of graphs can generate the same data. Thus, to infer graph topology we need to employ some priors, for example, to match statistics via imposing sparsity or smoothness conditions on the graph. A straightforward way would be on the basis of some fundamental statistical models, such as the covariance or precision matrices of the Gaussian distribution (due to their positive definiteness property), the Gaussian Markov random field with local independence prior, and the factor analysis model with smoothness assumptions. We envisage further progress of generative models to be based on the concept of diffusion processes on graphs. The generating model can be regarded as the graph signal that has been diffused by some graph kernels (e.g., polynomial kernels) from a white Gaussian distribution. By using unified notations across the thesis, graph learning can be treated as an inverse problem to a graph generation process, that is, $\mathbf{V} = g(\mathbf{Z})$, where $\mathbf{V}$ denotes the observed data, while the data are considered to be the output of an unknown graph transform $g(\cdot)$ of some initial status, $\mathbf{Z}$. The existing literature of learning a graph can be thought of as an attempt to infer the generative process, $g(\cdot)$, by matching the statistics of data, $\mathbf{V}$, with different priors on $\mathbf{Z}$.

Example 7.1. *We show in Figures 7.1 and 7.2 that different graph connections can have different smoothness features, given the same observed samples. As indicated in Figure 7.2, the observed sample retains the lowest frequency components on Graph-(a) and the highest frequency components on Graph-(c). This also results in the smoothest nodal sample when evaluated on Graph-(a). In signal processing, we typically aim at “de-noising” a signal to obtain low-frequency based (or smooth) signals. Therefore, given the observed samples, the prior of smoothness can also be helpful when learning a graph.*

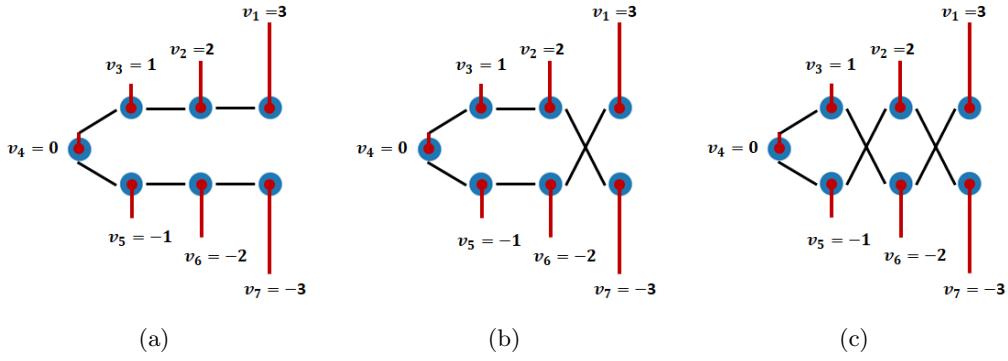


Figure 7.1: Three types of graph connections given the observed nodal sample $\mathbf{v} = [3, 2, 1, 0, -1, -2, -3]^T$.

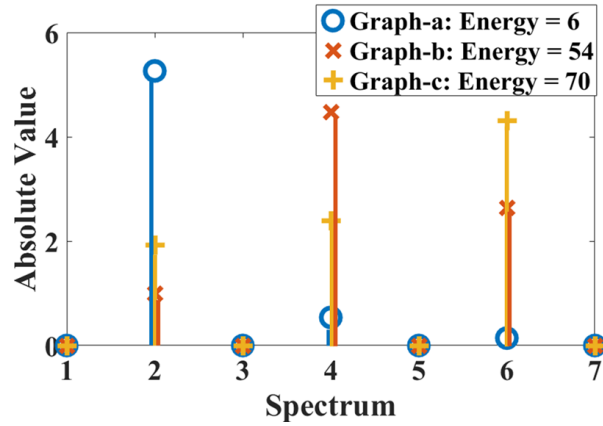


Figure 7.2: The spectrum values corresponding to the three types of graph connections in Figure 7.1. The energy is calculated via $\mathbf{v}^T \mathbf{L} \mathbf{v}$, where small values indicate a smooth graph.

7.2.1 Basic Gaussian Model

The simplest way of constructing a graph would be to associate edge weights with the covariance of graph data; this is reasonable under the Gaussian assumption, since the first two moments fully capture the whole statistics of the data. Given the covariance matrix, its non-zero elements naturally provide consistent estimation of the connectivity within a graph. Furthermore, by associating the adjacency matrix $\mathbf{A}$ with the covariance Σ of a Gaussian distribution, a more sophisticated approach would be to use hypothesis testing via setting a false alarm rate (Chapter 7.3.1 of [238]). However, the limitation of this model is that by employing individual tests, the number of implementation scales in inferring the graph grows up to $\mathcal{O}(m^2)$, where m is the number of nodes or the dimension of the node vector. This, on the

one hand, results in high computational complexity in relatively large graphs; on the other hand, this leads to increasingly false judgements even with a constant false alarm rate. A further possible misleading of the correlation models is due to the fact that the *correlation does not mean the causation*. In other words, the i -th and j -th vertices can show a strong correlation when they are all highly influenced by a middle vertex, however, they are not the causation of one another.

A way of addressing this ambiguity is to use the conditional correlation given by the precision matrix $\mathbf{Q}$ as an indicator of $\mathbf{A}$ because as introduced in Chapter 2, the precision matrix $\mathbf{Q}$ in the Gaussian distribution reflects the independence so as to show the Markov property. A similar hypothesis testing method can also be employed and is still limited to large-scale graphs. More importantly, this method has one additional limitation, in that it requires the number of samples to be larger than the dimension of covariance to ensure a proper inverse of covariance; this does not necessarily hold, especially for large-scale graphs. Therefore, an advancement of this model is to impose a prior on the precision matrix, which results into a regularisation term in estimating the precision matrix and the graph LASSO method can be applied to solving the advanced model [239]. More importantly, constraining the elements in $\mathbf{A}$ to be non-negative is also a common regularisation type in graphical learning. For more detail, we refer to [240–242].

7.2.2 Factor Model and Polynomial Models

We can further use the factor model introduced in Chapter 2.1.1 to infer the graph connection $\mathbf{A}$ or $\mathbf{L}$ [243–246]. To allow for more flexible generating procedures, a tractable non-linear model by the polynomial transform has been proposed in the following form,

$$\mathbf{v} = \sum_{i=1}^l \lambda_i \mathbf{S}^i \mathbf{z}_i + \boldsymbol{\epsilon}, \quad (7.1)$$

where $\mathbf{z}_i$ is independent white Gaussian noise $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, while $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha^2 \mathbf{I})$, similar to the factor model in (2.6). From (7.1), $\mathbf{S}$ is the (symmetric) shift operator

which can be chosen as e.g., the adjacency matrix, or the Laplacian matrix, to name but a few. Furthermore, (7.1) can be compactly written as

$$\mathbf{v} = \sum_{i=0}^l \lambda_i \mathbf{S}^i \mathbf{z}_i, \quad (7.2)$$

where $\mathbf{S}^0 = \mathbf{I}$ and $\lambda_0 = \alpha^2$ retain the same statistics as those in (7.1).

On the basis of (7.2), the covariance of $\mathbf{v}$ can be calculated as

$$\begin{aligned} \Sigma &= \mathbb{E}[\mathbf{v}\mathbf{v}^T] = \left(\sum_{i=0}^l \lambda_i \mathbf{S}^i \right) \mathbb{E}[\mathbf{z}\mathbf{z}^T] \left(\sum_{i=0}^l \lambda_i \mathbf{S}^i \right)^T \\ &= \sum_{i=0}^l \lambda_i \mathbf{S}^i \left(\sum_{i=0}^l \lambda_i \mathbf{S}^i \right)^T = \mathbf{U}^T \left(\sum_{i=0}^l \lambda_i \Lambda^i \right)^2 \mathbf{U}, \end{aligned} \quad (7.3)$$

where we have used the eigen-decomposition $\mathbf{S} = \mathbf{U}^T \Lambda \mathbf{U}$.

Eigenvector estimation. From (7.3), we can see that the eigenvectors of $\mathbf{S}$ are the same as those of the covariance of $\mathbf{v}$. This means, in a straightforward way, that we can infer the eigenvectors of the $\mathbf{S}$ by the empirical covariance from the observed data, $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$.

Eigenvalue estimation. After obtaining the eigenvectors, the remaining task is to estimate the eigenvalues of $\mathbf{S}$. Without any additional constraints, it is obvious that arbitrary values can be chosen as the eigenvalues of $\mathbf{S}$ because we can always find a corresponding set of $\lambda_1, \lambda_2, \dots, \lambda_l$ that satisfies (7.3). Thus, to achieve a unique solution, we need to employ some prior on the function $f(\cdot)$ [247], which is modelled as

$$\min_{\mathbf{S}, \Lambda} f(\mathbf{S}), \text{ subject to } \mathbf{S} = \mathbf{U}^T \Lambda \mathbf{U}. \quad (7.4)$$

For example, when $f(\mathbf{S}) = \|\mathbf{S}\|_0$, the objective function minimises the number of edges, whereas $f(\mathbf{S}) = \|\mathbf{S}\|_1$ minimises the energy of graph edges.

Equation (7.2) assumes that the diffusion process starts from the same initial status, white Gaussian noise. An enhanced diffusion model has been proposed in [248] by

assuming that the signals are generated from multiple heat diffusion processes

$$\mathbf{v} = \sum_{i=1}^l e^{-\lambda_i \mathbf{L}} \mathbf{z}_i. \quad (7.5)$$

Here, $\mathbf{z}_i$ represents the initial state that can also be optimised, and λ_i controls the diffusion time (depth). This means that with a small λ_i , the i -th column of $e^{-\lambda_i \mathbf{L}}$ is localised at the i -th vertex. This model can be solved via a dictionary-learning solver by regarding $[e^{-\lambda_1 \mathbf{L}}, e^{-\lambda_2 \mathbf{L}}, \dots, e^{-\lambda_l \mathbf{L}}]$ as the dictionary $\mathbf{D}$ and $[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_l]$ as coefficients $\mathbf{Z}$. The objective function can now be formulated as

$$\min_{\mathbf{L}, \mathbf{Z}, \lambda_i} \|\mathbf{v} - \mathbf{DZ}\|_F^2 + \text{reg}(\mathbf{Z}) + \text{reg}(\mathbf{L}), \quad \text{subject to } \{\lambda_i\}_{i=1}^l \leq 0, \quad (7.6)$$

where $\text{reg}(\cdot)$ denotes a certain regularisation; for more detail, we refer to [248].

7.2.3 Deep Graph Generative Models

A further existing extension of the framework to link prediction would be to estimate the missing graph connections using the VAE, as introduced in Chapter 2.2.2. The initial attempt, called the VGAE [11], models the connectivity (adjacency matrix) through the encoder, by $q(\mathbf{z}|\mathbf{A}, \mathbf{v})$, whilst the decoder remains the standard neural network used to reconstruct the connectivity. In this way, the encoder in the VGAE can be implemented as a GCN, while the decoder may be simplified to only a product operation $p(\mathbf{A}|\mathbf{z}) = \text{sigmoid}(\mathbf{z}\mathbf{z}^T)$, where the sigmoid function is used to satisfy the probability constraint. The overall structure of the VGAE is given by Figure 7.3. To enable the use of a GCN also in the decoder and to comply with graph theory, work in [232] proposes to employ an intermediate and learnable adjacency matrix within the decoder. Other improvements include regularising VGAE through semantic validity [233], use of rich models (such as mixtures models) as a prior, $p(\mathbf{z})$ [234], and an asynchronous message passing scheme for directed acyclic graphs [235].

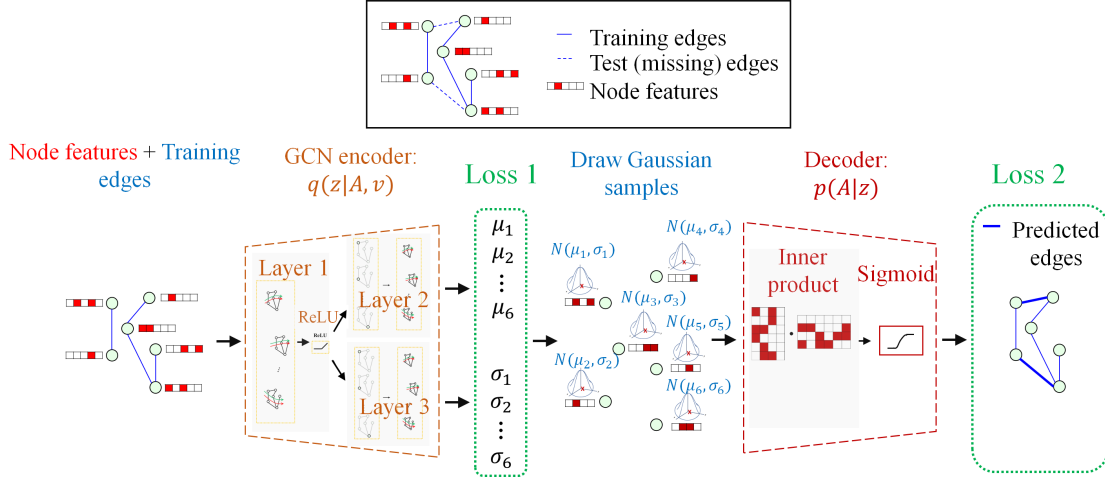


Figure 7.3: Principle of the use of VGAE in link prediction. The training process of the VGAE is based on the provided training edges and node features, whilst the test edges are not connected. Different from the standard auto-encoder, the encoder of VGAE outputs the estimated mean and covariance values of Gaussian distributions ($\mu_1, \mu_2, \dots, \mu_6$ and $\sigma_1, \sigma_2, \dots, \sigma_6$), while the input to the decoder is randomly drawn from the corresponding Gaussian distributions. Being a graph variance of the VAE, the loss consists of two parts: the term $\text{KL}(q(\mathbf{z}|\mathbf{A}, \mathbf{v})||p(\mathbf{z}))$ of *Loss 1* which reflects how well the output Gaussian approaches the standard Gaussian distribution, $p(\mathbf{z})$ (zero mean and identity covariance), while the term $\mathbb{E}_{q(\mathbf{z}|\mathbf{A}, \mathbf{v})}[\log p(\mathbf{A}|\mathbf{z})]$ of *Loss 2* corresponds to the quality of reconstruction of the adjacency matrix, $\mathbf{A}$.

7.3 Proposed Graph Link Prediction Method

Since the proposed RCF-GAN in Chapter 6 is capable of learning representative embeddings in the latent space, we therefore develop a GRCF-GAN to predict the missing links, following the idea that the observed signals are generated by probabilistic models of graphs. The overall structure is shown in Figure 7.4. Specifically, similar to the VGAE, we use the GCN as the encoder structure to accommodate the graph connection information $\mathbf{A}$. However, for the decoder, different from the VGAE that tries to reconstruct the incomplete adjacency matrix and thus suffers from possibilities of overfitting, we aim to learn representative features in the latent space, and thus are able to infer graph links by a simple product operation on the learnt features, i.e., by $\text{sigmoid}(\mathbf{z}\mathbf{z}^T)$. In other words, for the decoder of the proposed GRCF-GAN, we aim to generate graph node features, so that the learnt feature $\mathbf{z}$ is able to represent both the node features and the connectivity information.

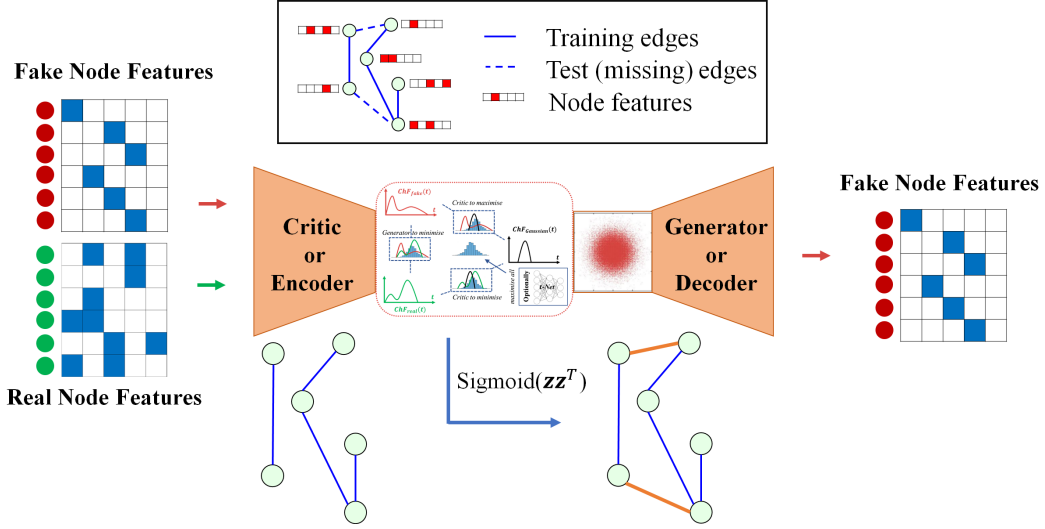


Figure 7.4: The overall structure of the proposed GRCF-GAN.

7.4 Experimental Results

In the following experiments on graph link prediction, we employed two well-known datasets, i.e., Cora and Citeseer [249]. The split of datasets into training, validation and test sets was exactly the same as that in the VGAE [11] for fair comparisons, i.e., 85% for training, 5% for validation and 15% for testing. The RCF-GAN settings from Section 6.4.1 remained the same to verify the robustness and generalisation of the proposed GRCF-GAN, except for the net structures. For the *critic* of GRCF-GAN, we adopted the same basic GCN as the one adopted in the VGAE for a fair comparison. The dimensions of both the *critic* in GRCF-GAN and the VGAE were set to $\{m_{node}, m_{layer}, 128\}$, where m_{node} denotes the dimension of node features and m_{layer} represents the dimension of the middle layer. We set m_{layer} by $\{256, 512, 1024, 2048\}$ in our experiments for comprehensive comparisons. The VGAE was also run with different m_{layer} for a fair comparison by its publicly provided code [11]. For the generator of our GRCF-GAN, we used a simple 3-layer fully connected neural net with dimensions $\{128, m_{layer}, m_{node}\}$. The training and testing process were repeatedly ran 10 times, with both models were trained over 300 epochs. Finally, the mean values of the area under the ROC curve (AUC) and average precision (AP) are shown in Fig. 7.5. Our implementation on graph link prediction is available at <https://github.com/ShengxiLi/grcf>.

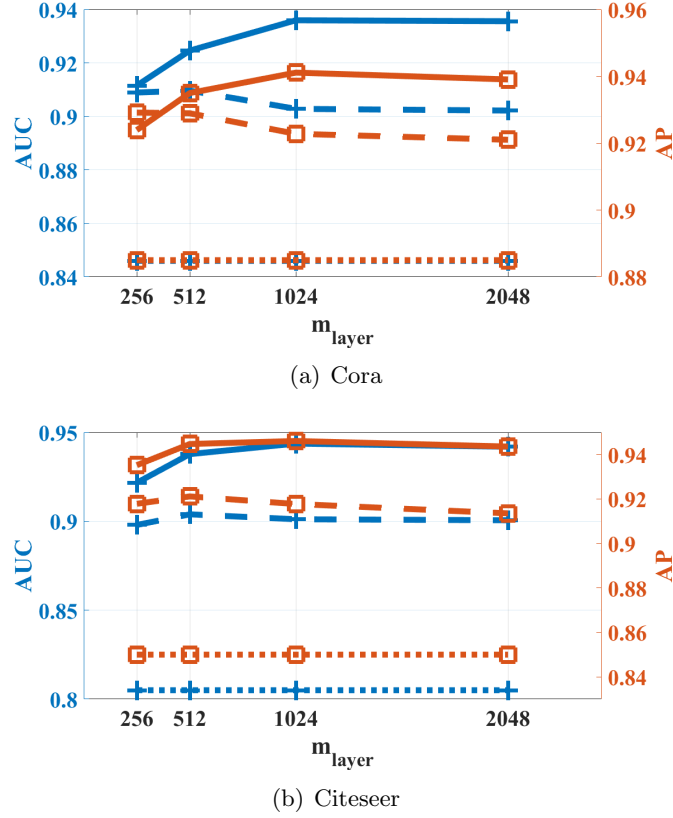


Figure 7.5: The AUC and AP scores of our GRCF-GAN and standard VGAE, with m_{layer} ranging from 256 to 2048. The solid lines show the AUC and AP of our GRCF-GAN in testing, whereas the dashed lines designate for the VGAE. Since spectral clustering [10] is a typical baseline when not using GCNs, spectral clustering results from [11] are shown in dotted lines.

Observe from Fig. 7.5 that for all m_{layer} our GRCF-GAN for link prediction consistently outperformed the VGAE and spectral clustering method by a significant margin. More specifically, the VGAE achieved its best performance at approximately $m_{layer} = 512$, with the AUC (AP) at 0.910 (0.929) for Cora and 0.904 (0.921) for Citeseer. Further increasing m_{layer} may lead to overfitting of the VGAE, thus decreasing its performances in link prediction. However, with the increase of network sizes, the performances of the proposed GRCF-GAN improved accordingly, and the AUC (AP) scores reached 0.936 (0.941) for Cora and 0.944 (0.946) for Citeseer. On the one hand, this may be due to the fact that GRCF-GAN does not directly reconstruct the adjacency matrix but instead learns a semantic embedded space for link prediction. On the other hand, this improvement also validates the effectiveness and efficiency of the learnt embedded space. Therefore, by unifying the autoencoder and

adversarial learning in the proposed GRCF-GAN, we are not only able to precisely mimic data distributions, but also learn semantically meaningful representations of the data.

7.5 Summary

In this chapter, we have developed the GRCF-GAN for the prediction of graph links (edges), which is a key task in graph representation learning. We have first discussed the generative models introduced in previous chapters in the context of the graph link prediction problem, so as to indicate the links between the probabilistic generative model and the graph generation. We have further proposed the GRCF-GAN for graph link prediction, which, uniquely, generates links by learning graph representations instead of minimising reconstruction errors as in existing works. The proposed GRCF-GAN has achieved the state-of-the-art performances, which also verifies the efficient and semantic way of learning of the GRCF-GAN structure.

Chapter 8

Conclusion

8.1 Summary of Thesis Achievements

This thesis makes original contributions to probabilistic generative models, and simultaneously develops the corresponding statistical metrics and estimation solvers. Three important properties, namely robustness, explainability and expressibility, have been improved and generalised for both explicit and implicit generative models, providing advanced AI techniques for modelling and understanding signals. This chapter briefly summaries the main contributions of this thesis, followed by discussion on open problems.

In Chapter 3, we have proposed a skewed elliptical distribution to generalise its symmetric counterpart, by the directional statistics depicted by the von-Mises-Fisher (vMF) distribution. This generalisation is physically meaningful and shares important properties with the elliptical distribution, including robustness, explicit density functions and moments, meaningful stochastic representations, and intrinsic independence and invariance properties. We have further proposed a stable and efficient solver, which has been verified both theoretically and through numerical results.

In Chapter 4, we have proposed the elliptical mixture model (EMM) to address multi-modal signals. We have provided a ready-to-use summary for computable

and identifiable EMMs, given the broad family of elliptical distribution, as well as calculated the influence function to quantify the robustness. The universal manifold solver was further proposed, which is equipped with a re-formulation trick and re-designed cost, to compensate for a mismatch of the statistical manifold established on elliptical distributions. The proposed EMMs, together with the universal solver, have been experimentally validated that they are resistant to outliers, and flexible and powerful in modelling, whilst achieving fast and stable convergence.

In Chapter 5, to overcome the local optima when learning EMMs, we have further proposed a relaxed Wasserstein distance for powerful discrepancy measurement between probabilistic models; the so proposed distance also allows for building explicit statistical manifold on EMMs, rather than on the elliptical distributions. Consequently, we have proved that the relaxed Wasserstein distance is an upper bound of the Wasserstein distance, whilst achieving explicit manifold operations. More importantly, for the universal manifold solver, we have proposed the directional Adam (Dadam), an adaptive stochastic manifold gradient descent method, to accelerate the convergence. Besides robustness and universal convergence, the proposed solver estimates the EMMs to approximately the global optimum with fast and stable convergence speed, which has been validated through comprehensive experiments.

In Chapter 6, we have proposed the reciprocal characteristic function GAN (RCF-GAN), to further enhance model expressibility to complicated signals. The proposed RCF-GAN is built upon the characteristic functions (CFs), a powerful tool that admits all the information about any general distribution. We have further proved that the proposed RCF-GAN is theoretically complete due to the proposed reciprocal theory and demonstrated that, by weighting the phase and amplitude parts of the CFs, we are able to control the generation style. More importantly, we have further proposed an optimal sampling strategy under the umbrella of elliptical distributions, and an anchor design to improve the convergence speed. Consequently, the proposed RCF-GAN is shown to seamlessly combine the auto-encoder and GAN, and thus learn a semantic low-dimensional manifold of signals, which promises both enhanced

expressibility and explainability of models. Whilst benefiting from simple and stable training, the proposed RCF-GAN has been experimentally validated to achieve the state-of-the-art generation and reconstruction.

In Chapter 7, we have introduced radically novel probabilistic generative models for graph link prediction tasks, which makes it possible to learn graphs from statistics of the observed graph signals, instead of using some fixed rules to decide graph connections. We have introduced most of the generative models to the task, and have further proposed a graph RCF-GAN (GRCF-GAN), which learns meaningful representations of graph nodes and achieves the state-of-the-art performances.

8.2 Future Work

Although remarkable advancements have been made in this thesis, we still have a long way to go before the probabilistic generative models are mature in terms of both robustness, explainability and expressibility, the road to the general artificial intelligence. Related to the achievements of this thesis, some of the open problems and future directions are provided below.

8.2.1 Generating Process: Advanced Elliptical Models

This thesis has provided a unifying framework of computable elliptical distributions, as well as the corresponding statistical manifold metrics to obtain universal solvers. The elliptical distribution generalises the Gaussian distribution and also shares its many important properties such as location-scale property, affine transform, and conditional/marginal distribution belonging to the elliptical distributions. It is therefore desirable to generalise many widely applied Gaussian generative models, including factor analysis, (deep) Gaussian process, Kalman filter, to name but a few. Moreover, this thesis has addressed the EMMs, however, the mixture of skewed elliptical distributions is still yet to be investigated. Eventually, the mixture

of skewed elliptical distributions will be able to identify multiple skewed clusters, and therefore comes with enhanced flexibility and explainability.

8.2.2 Metric: Statistical Manifold and Graphs

Given the fact that the graph Laplacian can be regarded as a discrete realisation of the Riemannian metric, it is therefore beneficial to connect graphs to the statistical manifolds of generative models, e.g., the elliptical distributions and EMMs, thus establishing statistical graphs. This is beneficial to both fields; on the one hand, the generative models can be fast and efficiently solved by using graph techniques, on the other hand, the graph learning can become seamless when combined with the proposed generative models in this thesis. For example, we can predict the graph connections by estimating an EMM, where a robust and universal predictor can be obtained.

8.2.3 Metric: Characteristic Functions in Distribution Alignment

Another potential direction following this thesis is to bring the general concept of the CFs into practice. The CF has been previously studied as a powerful tool in theoretical probabilistic analysis, while its practical applications have been limited due to complicated functional forms. This thesis has preliminarily applied the CF to learning generative adversarial nets by developing efficient sampling methods. It is therefore both necessary and beneficial to develop a more sophisticated and efficient method to evaluate the difference between two CFs, either explicitly or numerically. Furthermore, the complex-valued nature of the CF brings the physical meaning to the CF components. It is a well known experimental phenomenon that the phase of discrete Fourier transform of images captures the saliency information, which motivates a large volume of works in saliency detection. This thesis gives a probabilistic explanation to this phenomenon, paving the way for future work to embark upon this intrinsic relationship

8.2.4 Solver: Auto-Encoding in an Adversarial Way

The studies of GANs and studies on the design of probabilistic auto-encoders basically start from different perspectives because the former serves for the generation, or it “decodes” from random noise, whilst the latter, as its name implies, focuses on encoding to summarise information. Although there are extensive attempts on combining those two structures, they typically embed one into the other merely as components, such as by using an auto-encoder as a discriminator in GANs or using an adversarial idea in an auto-encoder. The RCF-GAN proposed in this thesis provides a way of equally treating the two structures; the proposed structure, which contains only two modules, can be regarded both as an “encoder-decoder” and “discriminator-generator”. We believe that this path is promising and efficient, as it combines both the powerful expressibility of GANs and also explainability by learning a semantic low-dimensional manifold. Following this direction, it is possible to propose many new models. For example, unsupervised classification can be achieved when using mixture distributions in the latent space in training. The so learnt representations can be further tuned and applied in down-streaming tasks, such as semi-supervised learning and supervised classifications.

8.2.5 Application: Image Semantic Compression and Representation

The very nature of the probabilistic generative models is to find the intrinsic and semantic representation of observed signals. The so established representation is able to encode the signals, especially for image and video signals. The current image and video compression standards aim at minimising the entropy in signals, which can be easily calculated within the considered probabilistic generative models. Therefore, application of the data-driven probabilistic generative models to compression is expected to further improve the coding efficiency, whilst achieving semantic codecs instead of meaningful bit streams in the existing image/video compression methods.

Bibliography

- [1] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.
- [3] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “Demystifying MMD GANs,” *arXiv preprint arXiv:1801.01401*, 2018.
- [4] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv preprint arXiv:1701.07875*, 2017.
- [5] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, “MMD GAN: Towards deeper understanding of moment matching network,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.
- [6] A. F. Ansari, J. Scarlett, and H. Soh, “A characteristic function approach to deep implicit generative modeling,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

- [7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “It takes (only) two: Adversarial generator-encoder networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [8] S. W. Park and J. Kwon, “Sphere generative adversarial network based on geometric moment matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4292–4301, 2019.
- [9] C. Hennig *et al.*, “Breakdown points for maximum likelihood estimators of location–scale mixtures,” *The Annals of Statistics*, vol. 32, no. 4, pp. 1313–1340, 2004.
- [10] L. Tang and H. Liu, “Leveraging social media networks for classification,” *Data Mining and Knowledge Discovery*, vol. 23, no. 3, pp. 447–478, 2011.
- [11] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *arXiv preprint arXiv:1611.07308*, 2016.
- [12] A. B. Tsybakov, *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.
- [13] C. E. Antoniak, “Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems,” *The annals of statistics*, pp. 1152–1174, 1974.
- [14] P. J. Diggle and R. J. Gratton, “Monte Carlo methods of inference for implicit statistical models,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 46, no. 2, pp. 193–212, 1984.
- [15] D. H. Wolpert and W. G. Macready, “Coevolutionary free lunches,” *IEEE Transactions on evolutionary computation*, vol. 9, no. 6, pp. 721–735, 2005.
- [16] K. W. Fang, S. Kotz, and K. W. Ng, *Symmetric multivariate and related distributions*. London, U.K.: Chapman & Hall, 1990.
- [17] P. J. Huber, “Robust statistics,” in *International Encyclopedia of Statistical Science*, pp. 1248–1251, Springer, 2011.

- [18] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.
- [20] 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,” *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.
- [21] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” *ArXiv*, vol. abs/1701.04862, 2017.
- [22] G. Monge, *Mémoire sur la théorie des déblais et des remblais*. De l’Imprimerie Royale, 1781.
- [23] S.-I. Amari, “Natural gradient works efficiently in learning,” *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [25] G. R. L. Silva, D. A. Vieira, A. C. Lisboa, and V. Palade, “On a multiobjective training algorithm for RBF networks using Particle Swarm optimization,” in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 282–285, IEEE, 2010.
- [26] M. Alsalamah, S. Amin, and V. Palade, “Heart disease diagnosis based on deep radial basis function kernel machine,” *EAI Endorsed Transactions on Ambient Systems*, vol. 5, no. 17, 2018.
- [27] H. Rue and L. Held, *Gaussian Markov random fields: Theory and applications*. CRC press, 2005.

- [28] Z. Kato, J. Zerubia, *et al.*, “Markov random fields in image segmentation,” *Foundations and Trends® in Signal Processing*, vol. 5, no. 1–2, pp. 1–155, 2012.
- [29] C. Wang, N. Komodakis, and N. Paragios, “Markov random field modeling, inference & learning in computer vision & image understanding: A survey,” *Computer Vision and Image Understanding*, vol. 117, no. 11, pp. 1610–1627, 2013.
- [30] M. B. Christopher, *Pattern recognition and machine learning*. Springer-Verlag New York, 2016.
- [31] K. Petersen and M. Pedersen, “The matrix cookbook,” *Technical Manual*, 2008.
- [32] D. Mandic, *Recurrent Neural Networks for Prediction-architectures and Stability*. PhD thesis, University of London 1999., 1999.
- [33] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [34] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.
- [35] S. Roweis and Z. Ghahramani, “A unifying review of linear gaussian models,” *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.
- [36] L. Xu and M. I. Jordan, “On convergence properties of the EM algorithm for gaussian mixtures,” *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.
- [37] G. Frahm, *Generalized elliptical distributions: Theory and applications*. PhD thesis, Universität zu Köln, 2004.
- [38] J. T. Kent and D. E. Tyler, “Redescending M-estimates of multivariate location and scatter,” *The Annals of Statistics*, pp. 2102–2119, 1991.

- [39] O. Arslan, “Convergence behavior of an iterative reweighting algorithm to compute multivariate M-estimates for location and scatter,” *Journal of Statistical Planning and Inference*, vol. 118, no. 1-2, pp. 115–128, 2004.
- [40] D. Peel and G. J. McLachlan, “Robust mixture modelling using the t distribution,” *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000.
- [41] J. L. Andrews and P. D. McNicholas, “Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions,” *Statistics and Computing*, vol. 22, no. 5, pp. 1021–1029, 2012.
- [42] T.-I. Lin, P. D. McNicholas, and H. J. Ho, “Capturing patterns via parsimonious t mixture models,” *Statistics & Probability Letters*, vol. 88, pp. 80–87, 2014.
- [43] J. Sun, A. Kaban, and J. M. Garibaldi, “Robust mixture modeling using the Pearson type VII distribution,” in *Proceeding of The IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, 2010.
- [44] S. Tan and L. Jiao, “Multivariate statistical models for image denoising in the wavelet domain,” *International Journal of Computer Vision*, vol. 75, no. 2, pp. 209–230, 2007.
- [45] R. P. Browne and P. D. McNicholas, “A mixture of generalized hyperbolic distributions,” *Canadian Journal of Statistics*, vol. 43, no. 2, pp. 176–198, 2015.
- [46] M. Mukhopadhyay, D. Li, and D. B. Dunson, “Estimating densities with nonlinear support using Fisher-Gaussian kernels,” *arXiv preprint arXiv:1907.05918*, 2019.
- [47] M. G. Genton and N. M. Loperfido, “Generalized skew-elliptical distributions and their quadratic forms,” *Annals of the Institute of Statistical Mathematics*, vol. 57, no. 2, pp. 389–401, 2005.

- [48] A. Wiesel, “Geodesic convexity and covariance estimation,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6182–6189, 2012.
- [49] T. Zhang, A. Wiesel, and M. S. Greco, “Multivariate generalized Gaussian distribution: Convexity and graphical models,” *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 4141–4148, 2013.
- [50] S. Sra and R. Hosseini, “Geometric optimisation on positive definite matrices for elliptically contoured distributions,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2562–2570, 2013.
- [51] D. Karlis and L. Meligkotsidou, “Finite mixtures of multivariate Poisson distributions with application,” *Journal of Statistical Planning and Inference*, vol. 137, no. 6, pp. 1942–1960, 2007.
- [52] E. López-Rubio, “Stochastic approximation learning for mixtures of multivariate elliptical distributions,” *Neurocomputing*, vol. 74, no. 17, pp. 2972–2984, 2011.
- [53] R. P. Browne, P. D. McNicholas, and M. D. Sparling, “Model-based learning using a mixture of mixtures of Gaussian and uniform distributions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 814–817, 2012.
- [54] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood and the em algorithm,” *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.
- [55] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the EM algorithm,” *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.
- [56] I. Naim and D. Gildea, “Convergence of the EM algorithm for gaussian mixtures with unbalanced mixing coefficients,” *arXiv preprint arXiv:1206.6427*, 2012.

- [57] R. Salakhutdinov, S. T. Roweis, and Z. Ghahramani, “Optimization with EM and expectation-conjugate-gradient,” in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 672–679, 2003.
- [58] R. Hosseini and S. Sra, “Matrix manifold optimization for Gaussian mixtures,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 910–918, 2015.
- [59] R. Hosseini and S. Sra, “An alternative to EM for Gaussian mixture models: Batch and stochastic riemannian optimization,” *arXiv preprint arXiv:1706.03267*, 2017.
- [60] A. Azzalini and A. Capitanio, “Statistical applications of the multivariate skew normal distribution,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 579–602, 1999.
- [61] J. Wang, J. Boyer, and M. G. Genton, “A skew-symmetric representation of multivariate distributions,” *Statistica Sinica*, pp. 1259–1270, 2004.
- [62] A. Azzalini and A. D. Valle, “The multivariate skew-normal distribution,” *Biometrika*, vol. 83, no. 4, pp. 715–726, 1996.
- [63] M. D. Branco and D. K. Dey, “A general class of multivariate skew-elliptical distributions,” *Journal of Multivariate Analysis*, vol. 79, no. 1, pp. 99–113, 2001.
- [64] S. K. Sahu, D. K. Dey, and M. D. Branco, “A new class of multivariate skew distributions with applications to Bayesian regression models,” *Canadian Journal of Statistics*, vol. 31, no. 2, pp. 129–150, 2003.
- [65] R. B. Arellano-Valle and M. G. Genton, “Multivariate unified skew-elliptical distributions,” *Chilean Journal of Statistics*, vol. 1, no. 1, pp. 17–33, 2010.
- [66] A. Abtahi and M. Towhidi, “The new unified representation of multivariate skewed distributions,” *Statistics*, vol. 47, no. 1, pp. 126–140, 2013.

- [67] C. Adcock and A. Azzalini, “A selective overview of skew-elliptical and related distributions and of their applications,” *Symmetry*, vol. 12, no. 1, p. 118, 2020.
- [68] A. Azzalini and R. B. Arellano-Valle, “Maximum penalized likelihood estimation for skew-normal and skew-t distributions,” *Journal of Statistical Planning and Inference*, vol. 143, no. 2, pp. 419–433, 2013.
- [69] C. Flecher, P. Naveau, and D. Allard, “Estimating the closed skew-normal distribution parameters using weighted moments,” *Statistics & Probability Letters*, vol. 79, no. 19, pp. 1977–1984, 2009.
- [70] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *arXiv preprint arXiv:1912.02762*, 2019.
- [71] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, “Generalization and equilibrium in generative adversarial nets (GANs),” in *Proceedings of the 34th International Conference on Machine Learning*, pp. 224–232, JMLR, 2017.
- [72] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” *arXiv preprint arXiv:1410.8516*, 2014.
- [73] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real NVP,” *arXiv preprint arXiv:1605.08803*, 2016.
- [74] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.
- [75] L. Theis, A. v. d. Oord, and M. Bethge, “A note on the evaluation of generative models,” *arXiv preprint arXiv:1511.01844*, 2015.
- [76] S. Zhao, J. Song, and S. Ermon, “Towards deeper understanding of variational autoencoding models,” *arXiv preprint arXiv:1702.08658*, 2017.
- [77] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?,” *arXiv preprint arXiv:1801.04406*, 2018.

- [78] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv:1701.04862*, 2017.
- [79] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.
- [80] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
- [81] S. Arora and Y. Zhang, “Do GANs actually learn the distribution? An empirical study,” *arXiv preprint arXiv:1706.08224*, 2017.
- [82] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos, “The Cramer distance as a solution to biased Wasserstein gradients,” *arXiv preprint arXiv:1705.10743*, 2017.
- [83] Y. Mroueh, T. Sercu, and V. Goel, “McGAN: Mean and covariance feature matching gan,” *arXiv preprint arXiv:1702.08398*, 2017.
- [84] Y. Mroueh and T. Sercu, “Fisher GAN,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2513–2523, 2017.
- [85] D. Berthelot, T. Schumm, and L. Metz, “BEGAN: Boundary equilibrium generative adversarial networks,” *arXiv preprint arXiv:1703.10717*, 2017.
- [86] I. Deshpande, Z. Zhang, and A. G. Schwing, “Generative modeling using the sliced Wasserstein distance,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3483–3491, 2018.
- [87] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv preprint arXiv:1511.05644*, 2015.
- [88] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” *arXiv preprint arXiv:1512.09300*, 2015.

- [89] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” *arXiv preprint arXiv:1605.09782*, 2016.
- [90] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Neural photo editing with introspective adversarial networks,” *arXiv preprint arXiv:1609.07093*, 2016.
- [91] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, “Mode regularized generative adversarial networks,” *arXiv preprint arXiv:1612.02136*, 2016.
- [92] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially learned inference,” *arXiv preprint arXiv:1606.00704*, 2016.
- [93] J. Duchi, “Derivations for linear algebra and optimization,” *Berkeley, California*, vol. 3, no. 1, pp. 2325–5870, 2007.
- [94] S. M. Ali and S. D. Silvey, “A general class of coefficients of divergence of one distribution from another,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.
- [95] Y. Censor and A. Lent, “An iterative row-action method for interval convex programming,” *Journal of Optimization theory and Applications*, vol. 34, no. 3, pp. 321–353, 1981.
- [96] S. Nowozin, B. Cseke, and R. Tomioka, “f-gan: Training generative neural samplers using variational divergence minimization,” *arXiv preprint arXiv:1606.00709*, 2016.
- [97] C. Villani, *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.
- [98] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, “Optimal mass transport: Signal processing and machine-learning applications,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 43–59, 2017.
- [99] A. Müller, “Integral probability metrics and their generating classes of functions,” *Advances in Applied Probability*, vol. 29, no. 2, pp. 429–443, 1997.

- [100] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, “Hilbert space embeddings and metrics on probability measures,” *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1517–1561, 2010.
- [101] A. Smola, A. Gretton, L. Song, and B. Schölkopf, “A Hilbert space embedding for distributions,” in *International Conference on Algorithmic Learning Theory*, pp. 13–31, Springer, 2007.
- [102] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.
- [103] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [104] S. Bonnabel, “Stochastic gradient descent on Riemannian manifolds,” *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.
- [105] Y. Liu, F. Shang, J. Cheng, H. Cheng, and L. Jiao, “Accelerated first-order methods for geodesically convex optimization on riemannian manifolds,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 4868–4877, 2017.
- [106] H. Zhang, S. J. Reddi, and S. Sra, “Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 4592–4600, 2016.
- [107] H. Zhang and S. Sra, “First-order methods for geodesically convex optimization,” in *Proceeding of Conference on Learning Theory*, pp. 1617–1638, 2016.
- [108] S. J. Reddi, A. Hefny, S. Sra, B. Póczos, and A. Smola, “Stochastic variance reduction for nonconvex optimization,” in *Proceeding of International Conference on Machine Learning*, pp. 314–323, 2016.
- [109] H. Zhang and S. Sra, “Towards Riemannian accelerated gradient methods,” *arXiv preprint arXiv:1806.02812*, 2018.

- [110] C. RAO, “Information and accuracy attainable in the estimation of statistical parameters,” *Bull. Calcutta Math. Soc.*, vol. 37, pp. 81–91, 1945.
- [111] L. T. Skovgaard, “A Riemannian geometry of the multivariate normal model,” *Scandinavian Journal of Statistics*, pp. 211–223, 1984.
- [112] C. Atkinson and A. F. Mitchell, “Rao’s distance measure,” *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 345–365, 1981.
- [113] S.-I. Amari, “Differential geometry of curved exponential families—curvatures and information loss,” *The Annals of Statistics*, pp. 357–385, 1982.
- [114] R. Bhatia, *Positive definite matrices*. Princeton university press, 2009.
- [115] F. Nielsen and V. Garcia, “Statistical exponential families: A digest with flash cards,” *arXiv preprint arXiv:0911.4863*, 2009.
- [116] L. Malagò and G. Pistone, “Information geometry of the Gaussian distribution in view of stochastic optimization,” in *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pp. 150–162, 2015.
- [117] M. E. Khan and D. Nielsen, “Fast yet simple natural-gradient descent for variational inference in complex models,” in *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pp. 31–35, IEEE, 2018.
- [118] A. Renaux, P. Forster, E. Chaumette, and P. Larzabal, “On the high-SNR conditional maximum-likelihood estimator full statistical characterization,” *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4840–4843, 2006.
- [119] S. Zozor, C. Ren, and A. Renaux, “On the maximum likelihood estimator statistics for unimodal elliptical distributions in the high signal-to-noise ratio regime,” *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 883–887, 2018.
- [120] S. Li, Z. Yu, M. Xiang, and D. P. Mandic, “Solving general elliptical mixture models through an approximate Wasserstein manifold,” in *Proceedings of the AAAI*, pp. 4658–4666, 2020.

- [121] S. Li, Z. Yu, and D. Mandic, “A universal framework for learning the elliptical mixture model,” *IEEE Transactions on Neural Networks and Learning Systems*, in print, 2020.
- [122] S. Zozor and C. Vignat, “On the MAP estimation in the context of elliptical distributions,” in *17th European Signal Processing Conference*, pp. 2460–2464, IEEE, 2009.
- [123] L. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo, S. Li, A. G. Constantinides, *et al.*, “Data analytics on graphs Part III: Machine learning on graphs, from graph topology to applications,” *Foundations and Trends® in Machine Learning*, vol. 13, no. 4, 2020.
- [124] H. Nurminen, T. Ardeshiri, R. Piche, and F. Gustafsson, “Robust inference for state-space models with skewed measurement noise,” *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, 2015.
- [125] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, “Skew- t filter and smoother with improved covariance matrix approximation,” *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5618–5633, 2018.
- [126] L. Zhang and J. Lan, “Extended object tracking using random matrix with skewness,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 5107–5121, 2020.
- [127] I. Lopez-Moreno, D. Ramos, J. Gonzalez-Dominguez, and J. Gonzalez-Rodriguez, “Von Mises–Fisher models in the total variability subspace for language recognition,” *IEEE Signal Processing Letters*, vol. 18, no. 12, pp. 705–708, 2011.
- [128] M. Costa, V. Koivunen, and H. V. Poor, “Estimating directional statistics using wavefield modeling and mixtures of von-Mises distributions,” *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1496–1500, 2014.

- [129] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, “Unscented von Mises–Fisher filtering,” *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 463–467, 2016.
- [130] S. Sra, “Directional statistics in machine learning: A brief review,” *Applied Directional Statistics: Modern Methods and Case Studies*, p. 225, 2018.
- [131] M. G. Genton, L. He, and X. Liu, “Moments of skew-normal random vectors and their quadratic forms,” *Statistics & Probability Letters*, vol. 51, no. 4, pp. 319–325, 2001.
- [132] T. Hillen, K. J. Painter, A. C. Swan, and A. D. Murtha, “Moments of von-Mises and Fisher distributions and applications,” *Mathematical Biosciences & Engineering*, vol. 14, no. 3, p. 673, 2017.
- [133] A. Ben-Israel, “The change-of-variables formula using matrix volume,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 300–312, 1999.
- [134] M. A. T. Figueiredo and A. K. Jain, “Unsupervised learning of finite mixture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.
- [135] B. G. Lindsay, “Mixture models: Theory, geometry and applications,” in *NSF-CBMS Regional Conference Series in Probability and Statistics*, pp. i–163, JSTOR, 1995.
- [136] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, “Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts,” *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 61–80, 2012.
- [137] D. P. Mandic, D. Obradovic, A. Kuh, T. Adali, U. Trutschell, M. Golz, P. De Wilde, J. Barria, A. Constantinides, and J. Chambers, “Data fusion for modern engineering applications: An overview,” in *Proceeding of International Conference on Artificial Neural Networks*, pp. 715–721, Springer, 2005.

- [138] H. Holzmann, A. Munk, and T. Gneiting, “Identifiability of finite mixtures of elliptical distributions,” *Scandinavian Journal of Statistics*, vol. 33, no. 4, pp. 753–763, 2006.
- [139] D. F. Andrews and C. L. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 99–102, 1974.
- [140] B. Jeuris, R. Vandebril, and B. Vandereycken, “A survey and comparison of contemporary algorithms for computing the matrix geometric mean,” *Electronic Transactions on Numerical Analysis*, vol. 39, no. EPFL-ARTICLE-197637, pp. 379–402, 2012.
- [141] S. Sra, “A new metric on the manifold of kernel matrices with application to matrix geometric means,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 144–152, 2012.
- [142] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, “Kernel methods on the Riemannian manifold of symmetric positive definite matrices,” in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 73–80, IEEE, 2013.
- [143] M. Faraki, M. T. Harandi, and F. Porikli, “A comprehensive look at coding techniques on Riemannian manifolds,” *IEEE Transactions on Neural Networks and Learning Systems*, 2018.
- [144] J. Burbea and C. R. Rao, “Entropy differential metric, distance and divergence measures in probability spaces: A unified approach,” *Journal of Multivariate Analysis*, vol. 12, no. 4, pp. 575–596, 1982.
- [145] F. Hiai and D. Petz, “Riemannian metrics on positive definite matrices related to means,” *Linear Algebra and its Applications*, vol. 430, no. 11-12, pp. 3105–3130, 2009.

- [146] M. Moakher and M. Zéraï, “The riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 2, pp. 171–187, 2011.
- [147] M. Calvo and J. M. Oller, “A distance between multivariate normal distributions based in an embedding into the Siegel group,” *Journal of Multivariate Analysis*, vol. 35, no. 2, pp. 223–242, 1990.
- [148] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan, “Local maxima in the likelihood of Gaussian mixture models: Structural results and algorithmic consequences,” in *Proceedings of the Advances in neural information processing systems*, pp. 4116–4124, 2016.
- [149] S. Miyajima, “Verified computation of the matrix exponential,” *Advances in Computational Mathematics*, vol. 45, no. 1, pp. 137–152, 2019.
- [150] P.-A. Absil and J. Malick, “Projection-like retractions on matrix manifolds,” *SIAM Journal on Optimization*, vol. 22, no. 1, pp. 135–158, 2012.
- [151] N. Boumal and P.-A. Absil, “Low-rank matrix completion via preconditioned optimization on the Grassmann manifold,” *Linear Algebra and its Applications*, vol. 475, pp. 200–239, 2015.
- [152] E. Ollila and D. E. Tyler, “Regularized M -estimators of scatter matrix,” *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 6059–6070, 2014.
- [153] Y. Chung, A. Gelman, S. Rabe-Hesketh, J. Liu, and V. Dorie, “Weakly informative prior for point estimation of covariance matrices in hierarchical models,” *Journal of Educational and Behavioral Statistics*, vol. 40, no. 2, pp. 136–157, 2015.
- [154] B. Rajaratnam, H. Massam, C. M. Carvalho, *et al.*, “Flexible covariance estimation in graphical Gaussian models,” *The Annals of Statistics*, vol. 36, no. 6, pp. 2818–2849, 2008.

- [155] J. Barnard, R. McCulloch, and X.-L. Meng, “Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage,” *Statistica Sinica*, pp. 1281–1311, 2000.
- [156] M. Fop, T. B. Murphy, and L. Scrucca, “Model-based clustering with sparse covariance matrices,” *Statistics and Computing*, vol. 29, no. 4, pp. 791–819, 2019.
- [157] J. Mulder, L. R. Pericchi, *et al.*, “The matrix- f prior for estimating and testing covariance matrices,” *Bayesian Analysis*, vol. 13, no. 4, pp. 1193–1214, 2018.
- [158] S. Said, L. Bombrun, Y. Berthoumieu, and J. H. Manton, “Riemannian Gaussian distributions on the space of symmetric positive definite matrices,” *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2153–2170, 2017.
- [159] H. Hajri, I. Ilea, S. Said, L. Bombrun, and Y. Berthoumieu, “Riemannian Laplace distribution on the space of symmetric positive definite matrices,” *Entropy*, vol. 18, no. 3, p. 98, 2016.
- [160] R. A. Maronna, “Robust M-estimators of multivariate location and scatter,” *The Annals of Statistics*, pp. 51–67, 1976.
- [161] D. E. Tyler, “Breakdown properties of the M-estimators of multivariate scatter,” *arXiv preprint arXiv:1406.4904*, 2014.
- [162] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, “Complex elliptically symmetric distributions: Survey, new results and applications,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5597–5625, 2012.
- [163] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*, vol. 196. John Wiley & Sons, 2011.
- [164] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “Manopt, a Matlab toolbox for optimization on manifolds,” *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014.

- [165] S. Dasgupta, “Learning mixtures of Gaussians,” in *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer science.*, pp. 634–644, IEEE, 1999.
- [166] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 898–916, May 2011.
- [167] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms.” <http://www.vlfeat.org/>, 2008.
- [168] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [169] J. Xu, D. J. Hsu, and A. Maleki, “Global analysis of expectation maximization for mixtures of two gaussians,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2676–2684, 2016.
- [170] S. Kolouri, G. K. Rohde, and H. Hoffmann, “Sliced Wasserstein distance for learning gaussian mixture models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.
- [171] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2048–2057, 2015.
- [172] Y. Chen and W. Li, “Natural gradient in Wasserstein statistical manifold,” *arXiv preprint arXiv:1805.08380*, 2018.
- [173] L. A. Stefanski, “A normal scale mixture representation of the logistic distribution,” *Statistics & Probability Letters*, vol. 11, no. 1, pp. 69–70, 1991.
- [174] Y. Chen, T. T. Georgiou, and A. Tannenbaum, “Optimal transport for Gaussian mixture models,” *IEEE Access*, vol. 7, pp. 6269–6278, 2018.

- [175] M. Knott and C. S. Smith, “On the optimal mapping of distributions,” *Journal of Optimization Theory and Applications*, vol. 43, no. 1, pp. 39–49, 1984.
- [176] B. Muzellec and M. Cuturi, “Generalizing point embeddings using the Wasserstein space of elliptical distributions,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 10237–10248, 2018.
- [177] A. Takatsu *et al.*, “Wasserstein geometry of Gaussian measures,” *Osaka Journal of Mathematics*, vol. 48, no. 4, pp. 1005–1026, 2011.
- [178] L. Malagò, L. Montrucchio, and G. Pistone, “Wasserstein Riemannian geometry of Gaussian densities,” *Information Geometry*, vol. 1, no. 2, pp. 137–179, 2018.
- [179] P. C. Parks, “Am lyapunov’s stability theory—100 years on,” *IMA journal of Mathematical Control and Information*, vol. 9, no. 4, pp. 275–303, 1992.
- [180] S. Sra and R. Hosseini, “Conic geometric optimization on the manifold of positive definite matrices,” *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 713–739, 2015.
- [181] K.-T. Sturm *et al.*, “On the geometry of metric measure spaces,” *Acta Mathematica*, vol. 196, no. 1, pp. 65–131, 2006.
- [182] J. Martens, “New insights and perspectives on the natural gradient method,” *arXiv preprint arXiv:1412.1193*, 2014.
- [183] J. Rabin, G. Peyré, J. Delon, and M. Bernot, “Wasserstein barycenter and its application to texture mixing,” in *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446, Springer, 2011.
- [184] B. Lévy, “A numerical algorithm for L2 semi-discrete optimal transport in 3d,” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 49, no. 6, pp. 1693–1715, 2015.
- [185] G. Becigneul and O.-E. Ganea, “Riemannian adaptive optimization methods,” in *International Conference on Learning Representations*, 2019.

- [186] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [187] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, 2010.
- [188] O. Barndorff-Nielsen, J. Kent, and M. Sørensen, “Normal variance-mean mixtures and z distributions,” *International Statistical Review/Revue Internationale de Statistique*, pp. 145–159, 1982.
- [189] N. Ghaffari and S. Walker, “On multivariate optimal transportation,” *arXiv preprint arXiv:1801.03516*, 2018.
- [190] F. Farnia and D. Tse, “A convex duality framework for GANs,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5248–5258, 2018.
- [191] S. Liu, O. Bousquet, and K. Chaudhuri, “Approximation and convergence properties of generative adversarial learning,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5545–5553, 2017.
- [192] S. Mohamed and B. Lakshminarayanan, “Learning in implicit generative models,” *arXiv preprint arXiv:1610.03483*, 2016.
- [193] Y. Mroueh, C.-L. Li, T. Sercu, A. Raj, and Y. Cheng, “Sobolev GAN,” *arXiv preprint arXiv:1711.04894*, 2017.
- [194] P. McCullagh, “Does the moment-generating function characterize a distribution?,” *The American Statistician*, vol. 48, no. 3, pp. 208–208, 1994.
- [195] L. A. Waller, “Does the characteristic function numerically distinguish distributions?,” *The American Statistician*, vol. 49, no. 2, pp. 150–152, 1995.
- [196] S. Li, Z. Yu, M. Xiang, and D. Mandic, “Solving general elliptical mixture models through an approximate Wasserstein manifold,” *arXiv preprint arXiv:1906.03700*, 2019.

- [197] D. Williams, *Probability with martingales*. Cambridge University Press, 1991.
- [198] A. Feuerverger, R. A. Mureika, *et al.*, “The empirical characteristic function and its applications,” *The Annals of Statistics*, vol. 5, no. 1, pp. 88–97, 1977.
- [199] E. Lukacs, “A survey of the theory of characteristic functions,” *Advances in Applied Probability*, vol. 4, no. 1, pp. 1–37, 1972.
- [200] T. Epps and K. J. Singleton, “An omnibus test for the two-sample problem using the empirical characteristic function,” *Journal of Statistical Computation and Simulation*, vol. 26, no. 3-4, pp. 177–203, 1986.
- [201] K. P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton, “Fast two-sample testing with analytic representations of probability measures,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1981–1989, 2015.
- [202] S. G. Bobkov, “Proximity of probability distributions in terms of Fourier–Stieltjes transforms,” *Russian Mathematical Surveys*, vol. 71, no. 6, p. 1021, 2016.
- [203] S. C. Douglas and D. P. Mandic, “The least-mean-magnitude-phase algorithm with applications to communications systems,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4152–4155, IEEE, 2011.
- [204] Z. Yu, S. Li, and D. Mandic, “Widely linear complex-valued autoencoder: Dealing with noncircularity in generative-discriminative models,” in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 339–350, Springer, 2019.
- [205] A. V. Oppenheim and J. S. Lim, “The importance of phase in signals,” *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.

- [206] D. P. Mandic and V. S. L. Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*, vol. 59. John Wiley & Sons, 2009.
- [207] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.
- [208] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [209] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop,” *arXiv preprint arXiv:1506.03365*, 2015.
- [210] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [211] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” *arXiv preprint arXiv:1710.10196*, 2017.
- [212] C.-G. Esseen *et al.*, “Fourier analysis of distribution functions: A mathematical study of the Laplace-Gaussian law,” *Acta Mathematica*, vol. 77, pp. 1–125, 1945.
- [213] K. Kreutz-Delgado, “The complex gradient operator and the CR-calculus,” *arXiv preprint arXiv:0906.4835*, 2009.
- [214] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [215] L. Stankovic, D. Mandic, M. Dakovic, M. Brajovic, B. Scalzo, S. Li, and A. G. Constantinides, “Graph signal processing—Part III: Machine

- learning on graphs, from graph topology to applications,” *arXiv preprint arXiv:2001.00426*, 2020.
- [216] G. Jeh and J. Widom, “SimRank: A measure of structural-context similarity,” in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 538–543, 2002.
- [217] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5165–5175, 2018.
- [218] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.
- [219] A. Grover and J. Leskovec, “Node2Vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.
- [220] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.
- [221] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *arXiv preprint arXiv:1901.00596*, 2019.
- [222] Z. Zhang, P. Cui, and W. Zhu, “Deep learning on graphs: A survey,” *arXiv preprint arXiv:1812.04202*, 2018.
- [223] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, “Machine learning on graphs: A model and comprehensive taxonomy,” *arXiv preprint arXiv:2005.03675*, 2020.
- [224] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, “GraphGAN: Graph representation learning with generative adversarial nets,” *arXiv preprint arXiv:1711.08267*, 2017.

- [225] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, “NetGAN: Generating graphs via random walks,” *arXiv preprint arXiv:1803.00816*, 2018.
- [226] N. De Cao and T. Kipf, “MolGAN: An implicit generative model for small molecular graphs,” *arXiv preprint arXiv:1805.11973*, 2018.
- [227] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, “GraphRNN: Generating realistic graphs with deep auto-regressive models,” *arXiv preprint arXiv:1802.08773*, 2018.
- [228] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia, “Learning deep generative models of graphs,” *arXiv preprint arXiv:1803.03324*, 2018.
- [229] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, “Adversarially regularized graph autoencoder for graph embedding,” *arXiv preprint arXiv:1802.04407*, 2018.
- [230] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, “Learning deep network representations with adversarially regularized autoencoders,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2663–2671, 2018.
- [231] C. Doersch, “Tutorial on variational autoencoders,” *arXiv preprint arXiv:1606.05908*, 2016.
- [232] A. Grover, A. Zweig, and S. Ermon, “Graphite: Iterative generative modeling of graphs,” in *International Conference on Machine Learning*, pp. 2434–2444, 2019.
- [233] T. Ma, J. Chen, and C. Xiao, “Constrained generation of semantically valid graphs via regularizing variational autoencoders,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 7113–7124, 2018.
- [234] A. Hasanzadeh, E. Hajiramezanali, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, “Semi-implicit graph variational auto-encoders,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 10712–10723, 2019.

- [235] M. Zhang, S. Jiang, Z. Cui, R. Garnett, and Y. Chen, “D-VAE: A variational autoencoder for directed acyclic graphs,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1588–1600, 2019.
- [236] L. Stanković, D. Mandić, M. Daković, M. Brajović, B. Scalzo, S. Li, and C. G. Anthony, “Data analytics on graphs part i: Graphs and spectra on graphs,” *Foundations and Trends® in Machine Learning*, vol. 13, no. 1, pp. 1–157, 2020.
- [237] L. Stanković, D. Mandić, M. Daković, M. Brajović, B. Scalzo, S. Li, and C. G. Anthony, “Data analytics on graphs part ii: Signals on graphs,” *Foundations and Trends® in Machine Learning*, vol. 13, no. 2-3, pp. 158–331, 2020.
- [238] E. D. Kolaczyk, *Statistical Analysis of Network Data – Methods and Models*. Springer-Verlag New York, 2009.
- [239] L. Stanković, D. Mandić, M. Daković, M. Brajović, B. Scalzo, S. Li, and C. G. Anthony, “Data analytics on graphs part iii: Machine learning on graphs, from graph topology to applications,” *Foundations and Trends® in Machine Learning*, vol. 13, no. 4, pp. 332–530, 2020.
- [240] O. Banerjee, L. E. Ghaoui, and A. d’Aspremont, “Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data,” *Journal of Machine Learning Research*, vol. 9, no. Mar, pp. 485–516, 2008.
- [241] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical LASSO,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [242] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

- [243] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.
- [244] V. Kalofolias, “How to learn a graph from smooth signals,” in *Proceedings of the Artificial Intelligence and Statistics*, pp. 920–929, 2016.
- [245] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, “Learning sparse graphs under smoothness prior,” in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6508–6512, IEEE, 2017.
- [246] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, “Connecting the dots: Identifying network structure via graph signal processing,” *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.
- [247] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, “Network topology inference from spectral templates,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, 2017.
- [248] D. Thanou, X. Dong, D. Kressner, and P. Frossard, “Learning heat diffusion graphs,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 484–499, 2017.
- [249] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.