UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Advanced Techniques of Cross Domain Translation Learning

by

Wanming Huang

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2020

Certificate of Original Authorship

I, Wanming Huang, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature: Signature removed prior to publication.

Date: 17/09/2020

ABSTRACT

Advanced Techniques of Cross Domain Translation Learning

by

Wanming Huang

Cross domain translation, such as image captioning, fashion synthesis from text descriptions, music composition in a particular style, has attracted considerable interest in the deep learning community lately. Despite significant progress in this field, certain drawbacks in previous methods have been identified. First, although the attention mechanism has been widely applied to domain transfer and achieved remarkable outcomes, cross-domain translation remains an open research question on cross domain transfer learning because of the different data structures. Second, most domain translation algorithms address only a pair of domains, and there is a need for $2 \times \binom{N}{2}$ transfer functions given N image domains. This makes training prohibitively unmanageable. We have proposed a set of solutions to solve these two problems, as described in detail in Chapter 3. Third, most generative model based domain-transfer algorithms uses single-mode distribution to model the latent space. This does not work well on datasets that contain diversified samples that form multiple clusters. Our study applies mixture models to cross-domain generation, of which the effects and properties are illustrated in Chapter 4. Finally, cross-domain translation models usually suffer from long training time and are difficult to converge. Indeed, this applies to most deep neural network training that involves complex network designs and large datasets. Our work in Chapter 5 accelerates deep neural network training with a specially designed mini-batch sampling strategy.

Dissertation directed by Associate Professor Richard Yi Da Xu

School of Electrical and Data Engineering

Acknowledgements

First and foremost, I would like to express deep feelings of gratitude to my supervisor, Associate Professor Richard Yi Da Xu, who has guided me through my PhD journey. He has taught me how to conduct research, and how interesting and amazing machine learning can be. He has spent time with me and my colleagues from day one, taught us calculus, statistics and all other mathematical theories required to conduct machine learning research. He has also shared his insights and the latest breakthrough in many areas of machine learning, which enlightens our own research. Further, he has provided guidance to each one of my paper and my thesis, not only on research ideas, but also on the structuring and writing. He helped me overcome each obstacle I met in my research. I appreciate all his time, effort, and funding that supported my PhD research. I am also thankful for his dedication and hardworking towards research, which exemplified moral and ethical research.

I would also like to thank my external supervisor Dr. Ian Oppermann. Ian and I have arranged meetings regularly to discuss my research progress and exchange ideas on the application of machine-learning algorithms to industry projects. I am deeply grateful for these meetings because they provided insights into my research. Ian also shared his own experience in research and discussed with me possible applications of my work in the industry, which shed light upon my future career path. Discussions with Ian further allowed me to examine problems in a more systematic way and realised what could be improved and what was left to be addressed. He also provided me with valuable advice on PhD study and thesis structuring.

The members of the research group have also contributed immensely to my research and personal life. Discussions with my colleagues, Shuai Jiang, Xuan Liang, Chen Deng, Yi Huang, Ying Li, Wei Huang and all other group members have allowed me to conquer numerous challenges. We have collaborated in multiple projects, and I greatly appreciate the inspiration and suggestions they provided from multiple fields. Shuai Jiang helped me a great deal with optimisation and statistics; Xuan Liang has enlightened me with his experience in neural translation models. I would also like to thank Haodong Chang and Wei Huang for their help in finding a proofreader.

Regarding my research in latent space modelling of generative adversarial networks with applications in anomaly detection, I would like to thank Dr. Hongbo Xie from the University of Queensland. He not only provided me with his knowledge and experience from a statistical perspective, he has also provided me with valuable advice on paper writing.

Further, I would like to thank Elite Editing for their effort in proofreading this thesis, and editorial intervention was restricted to Standards D and E of the *Australian Standards for Editing Practice*.

Finally, I would like to thank my parents for always being supportive of my research and my life.

Wanming Huang

Sydney, Australia, 2020.

List of Publications

Conference Papers

C-1. W. Huang, R. Y. D. Xu, and I. Oppermann, "Realistic image generation using region-phrase attention," in Proceedings of The Eleventh Asian Conference on Machine Learning, ser. Proceedings of Machine Learning Research, W. S. Lee and T. Suzuki, Eds., vol. 101. Nagoya, Japan: PMLR, 17–19 Nov 2019, pp. 284–299.

C-2. W. Huang, R. Y. D. Xu, and I. Oppermann, "Efficient diversified mini-batch selection using variable high-layer features," in Proceedings of The Eleventh Asian Conference on Machine Learning, ser. Proceedings of Machine Learning Research, W. S. Lee and T. Suzuki, Eds., vol. 101. Nagoya, Japan: PMLR, 17–19 Nov 2019, pp. 300–315.

C-3. "GAN-based Gaussian Mixture Model Responsibility Learning" accepted by ICPR 2020

Contents

Certificate ii

Abstract iii

Acknowledgments iv

List of Publications vi

List of Figures xi

Abbreviation xvi

Notation xviii

1 Introduction 1

1.1 Background 1

1.2 Problems 3

1.3 Contributions 10

1.4 Research Objectives 14

1.5 Thesis Organisation 14

2 Literature Survey 16

2.1 Introduction 16

2.2 Cross Domain Translation 16

2.3 GAN and its Variants 18

2.3.1 GAN 18

2.3.2 GAN Variants 19

2.3.3 GAN Based Text-to-Image Generation 20
2.3.4 Attention Mechanism and its Applications in Text-to-Image Generation 23
2.3.5 Text-to-Image Alignment 25
2.3.6 Image Domain Transfers With GANs 26
2.3.7 Normalisation Methods and Applications in GANs 27
2.4 Latent Space Modelling of GANs 29
2.4.1 GMM 29
2.4.2 GMM in GANs 30
2.4.3 Dirichlet Process 30
2.4.4 DP Applications in Generative Models 32
2.4.5 Other Latent Space Modelling 33
2.5 Mini-Batch and SGD 34
2.5.1 DPP 35
2.5.2 K-DPP 36
2.5.3 Markov DPP 36
2.5.4 DPP for Mini-Batch Sampling 36

3 Applications of Cross Domain Transfer **39**

3.1 Realistic Image Generation using Region-Phrase Attention 40
3.1.1 Text Encoder 42
3.1.2 Overall Text Embedding Loss 48
3.1.3 Attentional Text-to-Image Generation 48
3.1.4 Bounding Box Prediction 53
3.1.5 Experiments 54

3.2 Transfer of One Thousand Styles 62
3.2.1 Updating Chain Ordering 63
3.2.2 Alternative Mechanic using Scheduled Sampling 66
3.2.3 Experiments 67
3.3 Summary 73

4 Latent Space Modelling in Style Transfer 75

4.1 GAN-based GMM Responsibility Modelling 76
4.1.1 Posterior Consistency Module (PCM) 76
4.1.2 The Generator 79
4.1.3 The Discriminator 79
4.1.4 Training Parameters for the Prior Distribution 80
4.1.5 Experiments 80
4.2 Compressing GANs with Gaussian Mixture Prior 87
4.2.1 Method 87
4.2.2 Experiments 90
4.3 GMM in Text-to-image Generation 98
4.4 Dirichlet Process GAN 101
4.4.1 Generate z via Particle Filter 102
4.4.2 Updating DPGMM Parameters 106
4.4.3 Updating Other Parameters 106
4.4.4 Experiments 108
4.5 Summary 111

5 Training Acceleration in Style Transfer 117

5.1 Efficient Diversified Mini-Batch Selection Using Variable High-Layer Features . . . 118

5.1.1 Gram-Matrix Construction from Variable Higher-Layer Features . . . 119

5.1.2 Computation Complexity Analysis . . . 126

5.1.3 Experiments . . . 128

5.2 Diversified Mini-Batch Selection in Text-to-Image Generation . . . 134

5.3 Summary . . . 137

6 Conclusion 138

7 Appendix 143

7.1 Network Details for the Text-to-Image Generation . . . 143

7.1.1 Hyperparameters . . . 143

7.1.2 Basic Network Blocks . . . 143

7.1.3 Network Architecture for the Generator . . . 145

7.1.4 Network Architecture for the Discriminators . . . 146

Bibliography 148

List of Figures

1.1 Illustrative example of the mapping between latent distribution and data distribution when the latent distribution is modelled with (a) a single-mode Gaussian distribution; and (b) a mixture of Gaussians. . 6

1.2 Illustrative example of the mismatch between latent distribution and data distribution. 7

1.3 Illustration of clustering (a) low-dimensional data and (b) high-dimensional data. 8

1.4 Illustrative example of the mapping between latent distribution and data distribution when the number of modes in the latent distribution is (a) smaller than; and (b) larger than the actual number of data modes. 9

2.1 In both figures, magenta points are drawn from k-DPP. Cyan points are its subsequent draw. Green points are samples selected in both steps, where (a) is using an independent k-DPP, and (b) is drawn from a Markov k-DPP, conditioning the first. 37

3.1 Network structure for text embedding and GAN network. 41

3.2 Text embedding with a two-layer LSTM networks. 42

3.3 Text embedding with CNN layers. n represents the number of words in the sentence. “heads = 4” means four scaled dot-product attention run in parallel. 44

3.4 Examples of full image region, the regular-grid region and object-grid region 45

3.5 Thumbnail generator 49

3.6 Thumbnail bounding box conditioned discriminator 49

3.7 Example of attention being paid to a phrase when generating each object-grid region. White rectangles on the left figure highlight the object-grid regions in the image. The matched pair of phrase and object-grid image region is highlighted in the right image. 52

3.8 Validation losses of coordinates prediction and number prediction in terms of whether to use the sentences that include position related words for the training. 62

3.9 Overall architecture of the proposed framework in two epochs. The graph shows the structure when in t^{th} epoch, the order of styles is $3, 1, 2, \ldots, N$, the order is updated to $1, 3, 2, \ldots, N$ in the next epoch. The style ordering shown in the graph is for illustration only; the actual ordering is decided during training. 63

3.10 NOD=1 and NOD=2. Numbers are the index of each style, the two nodes being swapped in each step are highlighted in yellow. 65

3.11 Forward and backward generation results for the toy dataset. Real samples are in blue and the generated samples are in orange. 68

3.12 Forward and backward generation results on MNIST. The first row of both graphs is the real images; the following three rows are synthetic samples by taking 1, 2, and 3 steps from the real samples. . 70

3.13 Forward and backward generation results on cars from multiple angles. 71

3.14 Forward and backward generation results on images with multiple brightness. 73

4.1 The overall architecture. The feed-forward logic of the classifier C, the generator G and the discriminator D are marked with different colours. 76

4.2 Samples generated by the proposed models trained on the MNIST (left column), Fashion-MNIST (middle column) and CIFAR-10 (right column) datasets. The top row contains images generated using random vectors sampled from a different Gaussian. The bottom row shows the linear interpolation result from one Gaussian to another. The index of Gaussian does not necessarily correspond to the actual digit, generated images are reordered for demonstration purpose. 83

4.3 Linear interpolation over three Gaussians on the MNIST, Fashion-MNIST and Oxford-102 datasets. 84

4.4 Inception score and FID scores over training epochs on the Oxford dataset. 84

4.5 Inception and FID scores over epochs for the proposed GMM-based GAN with various size of the network compared with the vanilla GAN. 86

4.6 Performance on highly imbalanced dataset. 87

4.7 The complete architecture of our model. K is the number of Gaussians in the GMM; z_i is the random vector sampled from the GMM. x_i and $\hat{x}_i$ are the real and fake sample. G, C and D are the generator, discriminator and classifier. 88

4.8 Inception and FID scores over training epochs under different compression rates of GM-GAN and vanilla GAN on the CIFAR-10 and Oxford-102 dataset. 93

4.9 Performance of anomaly detection measured by area under curve. (a) AUC over anomalous classes on the MNIST dataset; (b) AUC over anomalous classes on the Fashion-MNIST dataset; (c) AUC over anomalous classes on the CIFAR-10 dataset. 97

4.10 Overall architecture of GMM based text-to-image generation. 99

4.11 Comparison between the Gaussian and GMM based GANs in terms of the inception and FID scores over epochs on the CUB and MSCOCO dataset. 102

4.12 The complete architecture of DP-GAN. 104

4.13 The overall flow of how the latent DPGMM distribution is updated. Notations on the left are the traditional flow of particles; notations on the right are how it is applied in the proposed architecture. 105

4.14 Samples from the toy dataset and generated from (a) GAN and (b) DP-GAN. Samples from the training set and its PDF are plotted in blue and synthetic samples and the PDF of their distribution are plotted in red. (c) Stick weights for the DPGMM when the model is fully trained. 109

4.15 (a) Samples generated by DP-GAN after trained on the MNIST dataset. Each row contains images sampled from a different Gaussian. Rows are sorted using the components' weights in the descending order. (b) Stick weights for the DPGMM when the model is fully trained. 110

4.16 (a) Samples generated by DP-GAN after training on the MNIST dataset. Each row contains images sampled from a different Gaussian. Rows are sorted using the components' weights in the descending order. (b) Stick weights for the DPGMM when the model is fully trained. 111

5.1 A visualization of three feature vectors being used to construct Gram-matrices. From left to right, each figure represents *raw*, *fixed higher-layer* and *variable higher-layer* features respectively. Each category is represented in a unique colour. Features are dimension reduced using t-SNE. 120

5.2 Duration over T on the Oxford 102 Flower dataset. 128

5.3 Calinski-Harabasz scores over iterations and the validation accuracy over the training duration on the Oxford 102 Flower dataset. 130

5.4 Validation accuracy over the training duration by the proposed sampling methods and the uniform sampling on the Stanford Dogs dataset. 132

5.5 Validation accuracy over the training duration by the proposed sampling methods and the uniform sampling on the Caltech 101 dataset. 133

5.6 Validation accuracy over training duration by the proposed sampling methods and the uniform sampling on the MNIST dataset. 135

5.7 Inception, FID and R-precision over epochs on the CUB dataset. "SGD" denotes the experiment performed using the conventional SG; "FULL-SINGLE" denotes the one performed using the FULL-SINGLE DPP sampling. 136

Abbreviation

3D - three-dimensional

AUC - area under the receiver operating characteristic curve

bi-LSTM - bidirectional long short-term memory

CUB - Caltech-USCD Bird

CNN - convolutional neural network

DCGAN - deep convolutional generative adversarial network

DM-SGD - Diversified Mini-Batch SGD

DNN - deep neural network

DP - Dirichlet process

DPGMM - Dirichlet process Gaussian mixture model

DPP - determinantal point process

FC - fully connected

FID - Fréchet Inception Distance

GAN - generative adversarial network

GAWWN - generative adversarial what-where network

GLU: gated linear unit

GM-GAN: Gaussian mixture generative adversarial network

GMM: Gaussian Mixture Model

GRU: gated recurrent unit

IS: inception score

LSTM - long short-term memory

MSCOCO - Microsoft COCO dataset

NF - normalizing flow

NMT - neural machine translation

NLP - natural language processing

PCM - posterior consistency module

RCM - responsibility consistency module

RCNN - region-based convolutional neural network

ReLU: rectified linear unit

RNN - recurrent neural network

ROC - receiver operating characteristic

RoI - region of interest

SGD - stochastic gradient descent

SMC - sequential Monte Carlo

SVRG - stochastic variance reduced gradient

VAE - variational autoencoder

Nomenclature and Notation

Capital letters denote matrices.

Lower-case alphabets denote column vectors.

$(.)^T$ denotes the transpose operation.

I_n is the identity matrix of dimension $n \times n$.

0_n is the zero matrix of dimension $n \times n$.

$\mathbb{R}$, $\mathbb{R}^+$ denote the field of real numbers, and the set of positive reals, respectively.