Image Diversification via Deep Learning based Generative Models

HIROSHI SASAKI

Image Diversification via Deep Learning based Generative Models

Hiroshi Sasaki

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
July 2023

Abstract

Machine learning driven pattern recognition from imagery such as object detection has been prevalenting among society due to the high demand for autonomy and the recent remarkable advances in such technology. The machine learning technologies acquire the abstraction of the existing data and enable inference of the pattern of the future inputs. However, such technologies require a sheer amount of images as a training dataset which well covers the distribution of the future inputs in order to predict the proper patterns whereas it is impracticable to prepare enough variety of images in many cases.

To address this problem, this thesis pursues to discover the method to diversify image datasets for fully enabling the capability of machine learning driven applications. Focusing on the plausible image synthesis ability of generative models, we investigate a number of approaches to expand the variety of the output images using image-to-image translation, mixup and diffusion models along with the technique to enable a computation and training dataset efficient diffusion approach. First, we propose the combined use of unpaired image-to-image translation and mixup for data augmentation on limited non-visible imagery. Second, we propose diffusion image-to-image translation that generates greater quality images than other previous adversarial training based translation methods. Third, we propose a patch-wise and discrete conditional training of diffusion method enabling the reduction of the computation and the robustness on small training datasets.

Subsequently, we discuss a remaining open challenge about evaluation and the direction of future work. Lastly, we make an overall conclusion after stating social impact of this research field.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to thank numerous people who have helped my greatly in achieving and inspiring the production of this thesis, which would not be accomplished without their support.

Firstly my supervisor, Professor Toby P. Breckon, for his time, patience and encouragement in tackling challenging, novel and scientific work and in always pushing me to achieve beneficial results. His openness and supportiveness with regard to my research ultimately led me onto the thesis contained with this document. Additionally, I would definitely like to thank my second supervisor, Dr Chris G. Willcocks, for his insightful help, advice and consultation during my PhD studies. His informative academic and scientific advice has polished my research activities. They have plentifully provided their precious expertise and mental support with me not only as academic advisors but also as respectful mentors; as a result, their supervision have brought me scientific mind, challenging spirit and high-productivity of the research work as well as led to high-quality of my academic publications including this thesis.

Besides, I am also very grateful to my fellow labmates for the endless discussions and the brilliant time during the study periods. Specifically, I would like to thank Dr Neelanjan Bhowmik, Dr Yona Falinie Abd. Gaus and other computer vision members for the insightful discussions for the research problems we have faced and Dr Samet Akcay, an alumni of our department, for the document template of this thesis.

This work has used Durham University's NCC cluster. NCC has been purchased through Durham University's strategic investment funds, and is installed and maintained by the Department of Computer Science. Also, this work made use of the facilities of the N8 Centre of Excellence in Computationally Intensive Research (N8 CIR) provided and funded by the N8 research partnership and EPSRC (Grant No. EP/T022167/1). The Centre is co-ordinated by the Universities of Durham, Manchester and York.

Financial assistance for my PhD study have gratefully been received from

Acquisition, Technology, and Logistic Agency (ATLA), Japan Ministry of Defense. I would like to present a sincere acknowledgement that Durham University offered me such the great PhD study opportunity whereas many universities in Japan prohibit themselves to enroll such the students funded by the Japanese defence body in their academic communities.

Above all, I would truly like to express my heartfelt appreciation to my wife, Mayu Sasaki, for her understanding, patience and cordial support within the 3 years of my PhD study in such the foreign country geographically and culturally quite far from her native place. Her kindness is, needless to say, one of the essential factors for the completion of this thesis.

Contents

List of Figures

List of Tables

Dedication

To my wife and to my parents

CHAPTER 1

Introduction

This thesis considers the common problem of machine learning based automated pattern recognition within imagery.

The demand for automated pattern recognition, especially automatic object detection and classification in imagery, is continuously expanding. There are many applications within computer vision utilising such pattern recognition, for example, optical character recognition [44], video surveillance [45], agricultural analysis from satellite imagery [46], and defect detection in factory automation [47]. This expanding demand within imagery is not limited to two-dimensional nature images in visible spectrum but also is being enhanced to three-dimensional images, non-visible spectrum images, and other visualised data, which will also expand the research field on the image-based pattern recognition (Figure 1.1).

The functions of pattern recognition have been historically enabled by matching between the appearances of reference and input images, e.g. edge matching [48]. This matching has been evolved by using further well-designed features extracted from images such as Scale-Invariant Feature Transform (SIFT) [49]. Moreover, this matching decision mechanism has been replaced by a data-driven approach to automatically define the decision boundary based on machine learning based

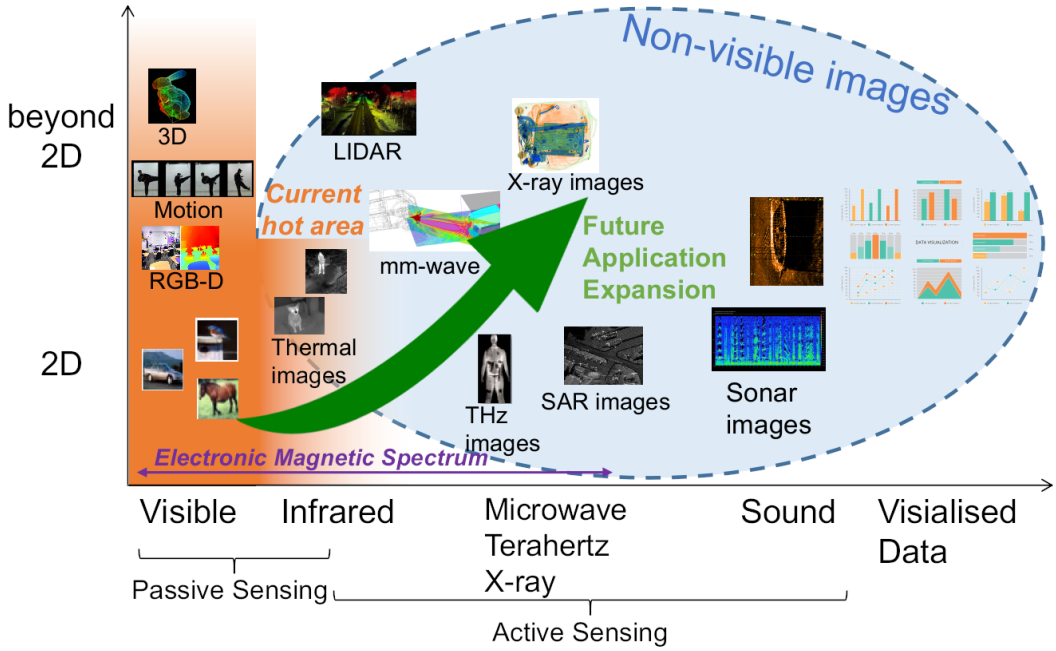


Figure 1.1: Expansion of computer vision field.

classifiers. The machine learning based classifiers, such as Linear Discriminant Analysis (LDA) [50], Support Vector Machine (SVM) [51], enable quite complex decision boundaries humans can not design and have improved the performances of various pattern recognition tasks. Artificial Neural Networks (ANN) [52] have played a key role in the machine learning field as they can learn not only the feature extractors but also feature representations. Multi-layer ANN are designed to act as a universal function approximator and to learn the mapping between the input and output end-to-end. The recent advances of ANN enable constructing a deep structured ANN, namely Deep Neural Networks (DNN) [53]. DNN has provided an unprecedented improvement in pattern recognition tasks along with the availability of big data and high-performance computing. In particular in the result in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [54] in 2012, DNN demonstrated a significantly better performance than other former popular approaches. Since this impressive emergence, DNN has enabled hitherto unprecedented performance on various challenging computer vision tasks such as image classification, object detection, semantic segmentation and temporal video analysis [55].

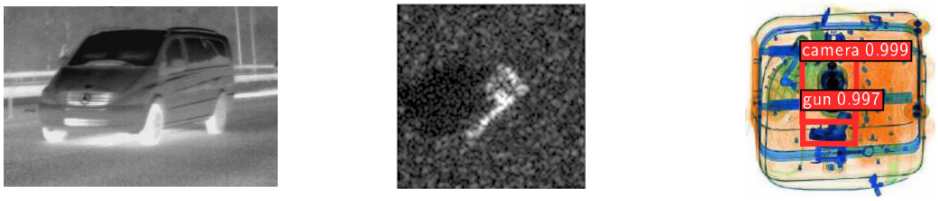


Figure 1.2: Thermal image [1] (left), SAR image [2] (centre), and X-ray image [3] (right).

Whilst contemporary DNN approaches generally perform well with large amounts of data available, within some cases, data availability is often more limited and it can be difficult to collect enough image samples to provide sufficient variability and coverage of the data distribution expected at inference (test, deployment) time. In particular, the applications within non-visible imagery such as infrared (thermal) [1], Synthetic Aperture Radar (SAR) [2] and X-ray images [3] (Figure 1.2) tend to have such issues due to their much limited availability than visible imagery. For example, SAR imagery and X-ray imagery are far less readily available and accessible due to both the lesser prevalence of this sensing technology and the higher associated sensor costs. In addition, such imagery significantly differs from visible-band imagery because it results from active sensing by backscatter projection of energy emission, whilst visible images are captured passively according to the intensity of reflected scene illumination. Moreover, such active sensing imagery is significantly impacted by sensor specification and its sensing configuration. This variation from conventional imagery precludes the direct applicability of commonplace transfer learning solutions, coupled with the lack of data availability, and inhibits inter-task applications with such diverse sensor imagery.

The limited number of training samples drives machine learning processes into the failure of capturing the underlying logic on the samples, so-called underfitting issue. Additionally, this situation of insufficient training samples is prone to bring the high-risk of excessively focusing on particular subsets in data and missing the general abstraction over the all samples, so-called overffing issue (Figure 1.3). Simply changing the size of the training dataset is not effective because random undersampling removes important examples and oversampling leads to the overfitting [56]. Those general phenomenon of machine learning within

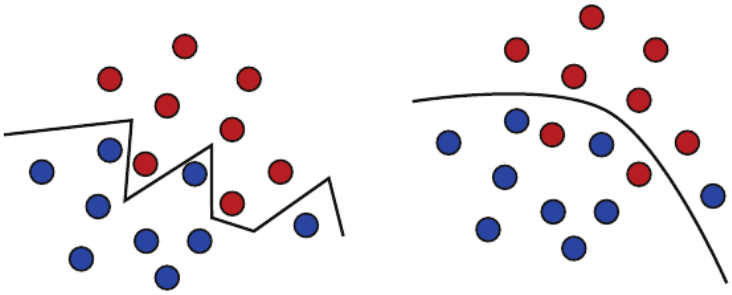


Figure 1.3: Conceptual illustrations of overfitting from [4]. (left) Overfitted example. The boundary is consistent over 2 classes (blue and red) training samples but it is a too complex separation surface that is not likely to generalize well in inference. (right) Well generalised example. The boundary is simpler and might be well fitted to abstraction across the entire training samples despite of it has misclassification of a few points.

small training datasets harms the accuracy of the recognition results in inference / deployment time and has limited the prevalence of this technology in the field of less data availability.

In order to address this issue of limited data availability, data augmentation methods by creating new images are traditionally adopted. These methods have commonly been conducted by some predefined image processing operations such as flipping / rotation [10] [9]. It mitigates those issues within some tasks by providing the variety of position or angle shifts of objects on the projected two-dimensional image spaces with the target model. However, this approach does not always contribute to enhancing the capability of covering future unseen images because the simple image transformation does neither consider what the images semantically look nor diversify the semantic contents in the dataset.

1.1 Motivation

We focus on stochastic generative models for expanding the variety of a given set of images. The generative models can generate new but similar images to those existing in the original dataset. Our top-level motivation stems from the idea that the generative models may be applicable for fabricating images that diversify the original dataset and well match human perception. Many contributions

(a) (b) (c)

Figure 1.4: Examples of the created images via recent generative models: (a) StyleGAN (from [5]), (b) VQ-VAE2 (from [6]), and (c) DDPM (from [7]).

in the field of stochastic generative models have been introduced; in particular, recent research generally focuses on the DNN architecture, Deep Generative Models (DGM) [12] [57] [58] [7], because of their potential modelling capability of data distributions in the real world.

DGM has dramatically been improved to synthesise realistic images by the rapid growth in this research field (Figure 1.4). Such images generated by state-of-the-art models have the potential to diversify limited datasets. However, generating effective images for actual pattern recognition applications is still challenging due to the highly restrictive requirements (discussed in Section 1.2). The research of this thesis pursues solutions to leverage DGM-based image diversification.

1.2 Requirements of Deep Generative Models for Image Diversification

We establish the following requirements of DGM for the sake of image diversification.

Wide Mode Coverage The ideal outcome of the image diversification is increasing data that hardly exist in training but appear in inference, whereas a standard DGM generally tries to learn itself to sample images based on the distribution of the training data. In real situations, the distribution of the existing dataset is often biased and different from the true distribution due to an imbalanced

or insufficient collection of training data. The trained models constructed in such a situation are unable to accurately model patterns or trends that appear within the test data distribution that are infrequent within the training data distribution. This means that DGM for our purpose needs to appropriately generate such low frequency data.

High Resolution Recent pattern recognition applications nowadays tend to use large size images. Moreover, recent DNN-based algorithms greedily extract faint and fine-detailed features from images to achieve high performance in application. Those demands of the fine-detail of the input images require high-resolution synthesis for DGM. Although recent DGM have increased the size of the support resolution, most work on DGM for imagery synthesise lower resolution images than many object classification or detection applications use.

Practical Computation and Training Data The rapid growth of the DGM research field toward synthesising high quality images leads to the increase in required computational resources and amount of training samples required. For example, StyleGANv2 [59] takes more than 1 month on a Nvidia V100 GPU to produce output of 256×256 pixel images. Furthermore DGM might suffer serious degradation in the case of training on small datasets [60]. DGM that is used for producing images applied to downstream tasks requires higher efficiency in the computation of data resources, or otherwise it might limit the applications.

1.3 Research Question

To achieve the requirements in Section 1.2, this thesis investigates the following main questions as:

1. *How to realise a wide mode coverage of DGM outputs?*
   This thesis attempts to find the methodologies to diversify the modality of the DGM outputs. We consider the following 3 strategies:
   - Generative Model based Image-to-image Translation

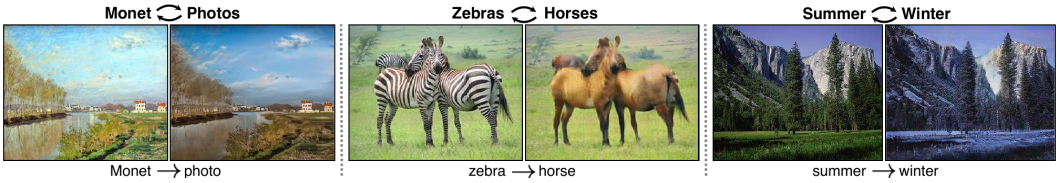


Figure 1.5: Image-to-image translation (from [8]).

- Image Feature Fusion
- High-modality Generative Models

Generative model based Image-to-Image (I2I) translation (Figure 1.5) can be a potential strategy to diversify image datasets because it enables to augment less available datasets by transferring from other domain datasets which are readily available. This technology is a special kind of DGM designed for I2I translation, whose aim is to acquire a model that relates different image domains. I2I translation tasks are classified as paired translation, which trains the model from image pairs of different domains, and unpaired I2I translation, which trains the model from two different domain (unpaired) image datasets. Unpaired I2I translation is a much applicable approach for data augmentation since it allows the difference in the scenes and the number of samples in source and target domain datasets. We review the related I2I translation work in Section 2.3.

Image feature fusion is a kind of interpolation approach that creates new images by mixing the latent features encoded from multiple images and decoding them back to fused images. The encoder and decoder are implemented as DNN as nonlinear mapping from data spaces to semantic feature spaces. The interpolation on this feature space can produce a semantically midpoint image between existing images and this approach may contribute to enhance the mode coverage of the outputs of DGM by incorporating itself into the generation process. The prior work about the image fusion techniques are reviewed in Section 2.4.

There is still a possible strategy to pursue generative models themselves that

enable to produce a wide mode coverage of data. Some recent DGM families using adversarial training (reviewed in Section 2.2.1) are popular because of their ability to synthesise high-quality images: however, those often loose capturing a low-density area of the distribution of the training data due to its aggressive adversarial training scheme. On the other hand, a non-adversarial scheme, furthermore combined with a progressive sampling strategy, such as diffusion models (reviewed in Section 2.2.3) have recently demonstrated the ability to produce simultaneously higher quality and more varied data compared to that of other contemporary DGM. The images generated by such diffusion approaches may cover the wider mode coverage of a desired image domain.

2. *How to realise high mode coverage, fidelity, and efficiency of the DGM?*
   This thesis will investigate approaches to realise a high mode coverage DGM with high-resolution outputs and efficiency with regard to both computation and training samples. We consider the following 3 strategies:

   - Patch-wise Training
   - Hierarchical Architectures
   - Combination of Different DGM Methods

   Patch-wise training is a technique that cuts large training images into small sub-regions (patches) before inputting them to the models during the training but aims to generate images of the original large size in the sampling process. This pre-cutting approach reduces the computation by decreasing the dimensions of the input in the model training, which would consume problematically huge computation when accepting the raw dimension of the images. Also, the model can better focus on learning the detail of the large images. This small patch training should be supported by supplemental techniques to model the distribution differences depending on the global position. We discuss this in Section 2.2.4.

   The use of hierarchical architectures is a beneficial idea for DGM that enables modelling high-resolution images. This hierarchical approach assumes that

images comprise hierarchical semantic information and attempts to disentangle and separately model such different levels of image information. The model training under this scheme can focus on each level of the image information and gain the temporal efficiency of the learning process as a result. The recent high-fidelity DGM adopt various hierarchical strategies, which are reviewed in Section 2.2.4.

Regardless of many types of DGM have demonstrated prominent performances, there is no decisive one satisfying all requirements of high quality, coverage, and sampling time efficiency at the same time as far as it relies on only one type of DGM. Due to the challenges of simultaneously achieving the requirements, the combined use of different DGMs can be a potential strategy. This hybrid strategy can manage the drawback each DGM type has. The recent examples of the combination methods in Section 2.2.5.

1.4 Contributions

In summary, the primary contributions of this thesis can be considered as follows:

- The proposal of a novel approach for data augmentation on limited non-visible imagery based on the generation of inter-class interpolated images using I2I translation and image feature fusion.
- The proposal of a novel I2I translation based on a diffusion model to utilise the characteristics of high fidelity and wide coverage of this prominent model.
- The proposal of a novel technique for a diffusion model within a limited sample number of a training dataset using patch-wise training, discrete variable autoencoders and transformers.

Portions of the work presented in this thesis have previously been published in the following papers:

- H. Sasaki, C. G. Willcocks and T. P. Breckon, "Data augmentation via mixed class interpolation using cycle-consistent generative adversarial networks

applied to cross-domain imagery," *25th International Conference on Pattern Recognition (ICPR)*, 2021 [1].

- H. Sasaki, C. G. Willcocks and T. P. Breckon. "UNIT-DDPM: Unpaired image translation with denoising diffusion probabilistic models." *arXiv preprint arXiv:2104.05358*, 2021.

1.5 Thesis Structure

This thesis reviews and compares four key methodologies contributing to the diversification of image datasets as well as reviewing approaches to improve DGM performances based on the requirements of Section1.2 (Chapter 2). Subsequently, we propose the combination method of I2I translation and image feature fusion to augment non-visible images by transferring visible images and analyse the effect on the object classification performance when applying this augmentation method (Chapter 3). Another new I2I translation approach that uses a diffusion model is proposed and evaluated the impact on the quality of the output images in Chapter 4. These two chapters describe solutions for Research Question 1 in Section 1.3. We also propose the diffusion model that enables less computation and a small dataset in training by combining with patch-wise training, discrete variable autoencoders and transformers (Chapter 5), which is a solution for Research Question 2 in Section 1.3. Lastly, we summarise the thesis and discuss the future research direction as a conclusion in Chapter 6.

[1] The schedule of ICPR2020 was shifted to 2021 due to the COVID-19 pandemic.

CHAPTER 2

Related Work

This chapter reviews four methodologies for image diversification: traditional data augmentation (Section 2.1), generative models (Section 2.2), image-to-image translation (Section 2.3), and image fusion (Section 2.4), before critiquing in Section 2.5. Subsequently, evaluation methods for the images generated via such methodologies are reviewed in Section 2.6.

2.1 Image Diversification via Traditional Data Augmentation

To improve the performance of machine learning driven applications, simple transform operations on images have been widely adopted to increase the number of training samples, namely data augmentation (DA). Those transformations traditionally use geometric and pixel-wise image processing (Section 2.1.1) but recently consider further complex operations such as changing an artistic style of an image (Section 2.1.2).

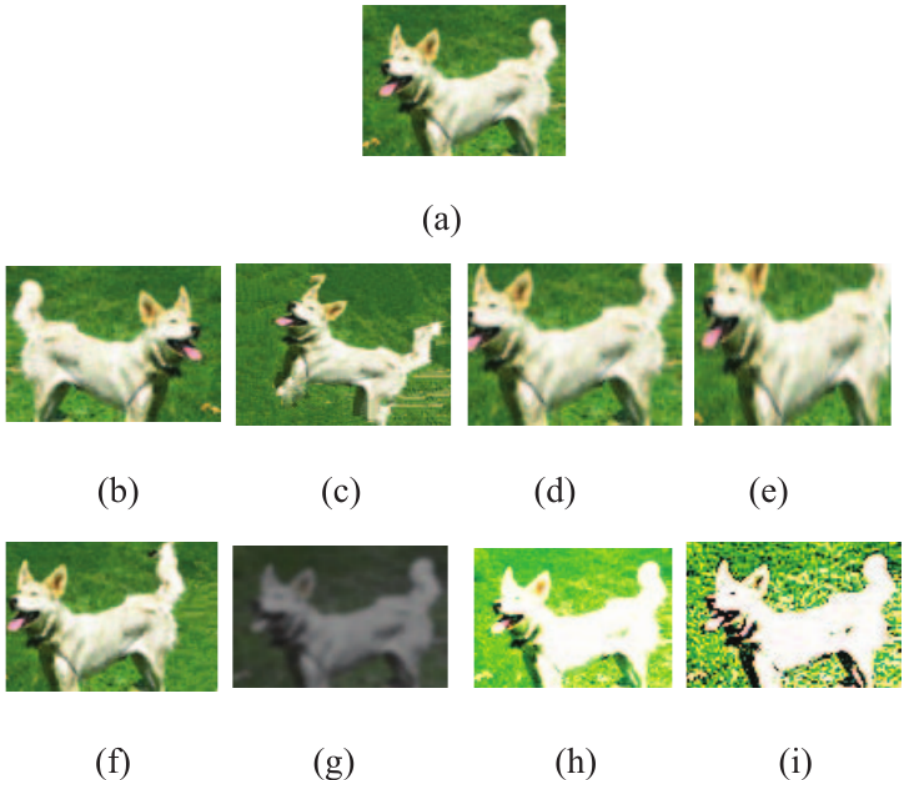


Figure 2.1: Examples of static image operations (from [9]): (a) original image (b) flipping (c) rotation (d)cropping (e) random-cropping (f) shifting (g) noise (h) color-jittering (i) PCA-jittering [10].

2.1.1 Geometric Transformation

Image diversification for data augmentation traditionally adopts a set of static geometric and pixel-wise image processing operations to transform an existing dataset image (e.g. flipping, rotation, cropping, adding noise, etc. [10] [9], Figure 2.1). They can mitigate overfitting of machine learning based model training by applying such transform operations across the training dataset used for model optimisation.

2.1.2 Image Style Transfer and Randomisation

In addition to the geometric operation in Section 2.1.1, the style of images can be diversified using Deep Neural Networks (DNN) based methods. A well trained DNN model can be used for controlled variation of the style of images. The gram matrices of the feature maps from each layer of the VGG19 network [61] represent the style

Figure 2.2: A result of image style transfer (from [11]): (left) an original image (right) a generated image using the style of the bottom left small image.

and can be used for diversifying images by changing its style to a style of other domain images [11] (Figure 2.2). This style transfer approach is applied not only to artistic images but also to photorealistic images [62]. Similarly, randomising such an image style using a style transfer network [63] can diversify the image datasets [64].

Whilst such unsupervised methods can reduce overfitting during model training, the trained models are often unable to accurately model patterns or trends that appear within the test distribution that are infrequent within the training data distribution. This is largely due to the fact that such unsupervised approaches transform data sampled from the same underlying training distribution, therefore their outputs reflect the inherent biases and patterns in this original training distribution.

2.2 Image Synthesis via Generative Models

A generative model is a stochastic model that approximates the probability distribution of a given observation. The model enables us to sample images that do not explicitly exist within the original dataset but are statistically similar to them. The generative model is generally designed as a parametric model p_θ to be learnt itself approximating the true distribution $p(X)$. The training assumes the distribution of the observation comes from the true distribution and is done by

empirical risk minimisation. Given a training dataset $\mathbf{x}_i \in X^{\text{train}}$, θ is optimised as:

$$\hat{\theta} = \arg\max \mathbb{E}_{\mathbf{x}_i}[\log p_\theta(\mathbf{x}_i)]. \tag{2.1}$$

Most generative models are designed as latent variable models, which relate the observation to latent variables. The latent variable model is defined as:

$$\begin{aligned} p_\theta(X) &= \int_{\mathbf{z}} p_\theta(X, \mathbf{z}) d\mathbf{z}, \mathbf{z} \sim p_\theta(\mathbf{z}) && (2.2) \\ &= \int_{\mathbf{z}} p_\theta(X|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}, && (2.3) \end{aligned}$$

where $\mathbf{z}$ is latent variables following the latent distribution $p_\theta(\mathbf{z})$.The trained $p_\theta(X|\mathbf{z})$ can generate data following the observation distribution from the latent variable distribution $p_\theta(\mathbf{z})$.

Recent advances in DNN enables to model quite high-resolution images. The following three sections (Section 2.2.1–2.2.3) review recent dominant generative models: generative adversarial networks, variational autoencoder, and denoising diffusion probabilistic models.

2.2.1 Generative Adversarial Networks

Generative Adversarial Networks (GAN) [12] optimise the model parameters via adversarial training in which the two sub-functions, named a generator $G_{\theta_g}(\mathbf{x})$ and discriminator $D_{\theta_d}(\mathbf{x})$, where θ_g, θ_d are parameters, compete iteratively (Figure 2.3). $G_{\theta_g}(\mathbf{z})$ accepts the prior $\mathbf{z}$ and tries to optimise the parameter to output images $\hat{X}$ maximising $D_{\theta_d}(\hat{X})$ whilst $D_{\theta_d}(\mathbf{x})$ tries to optimise the parameter to simultaneously minimising $D_{\theta_d}(\hat{X})$ and maximising $D_{\theta_d}(X^{\text{train}})$ as:

$$\begin{aligned} \theta_g &= \arg\max \mathbb{E}_{\mathbf{z}}[\log D_{\theta_d}(G_{\theta_g}(\mathbf{z}))], && (2.4) \\ \theta_d &= \arg\max \mathbb{E}_{X^{\text{train}}}[\log D_{\theta_d}(X^{\text{train}})] + \mathbb{E}_{\hat{X}}[\log(1 - D_{\theta_d}(\hat{X})]. && (2.5) \end{aligned}$$

Equation (2.4) and Equation (2.5) define the objective function $V(\theta_g, \theta_d)$ as:

$$V(\theta_g, \theta_d) = \mathbb{E}_{X^{\text{train}}}[\log D_{\theta_d}(X^{\text{train}})] + \mathbb{E}_{\hat{X}}[\log(1 - D_{\theta_d}(\hat{X})], \tag{2.6}$$

$$(\theta_g, \theta_d) = \arg\min_{\theta_g} \max_{\theta_d} V(\theta_g, \theta_d). \tag{2.7}$$

Let the distributions of X^{train} and $G_{\theta_g}(\mathbf{z})p_\theta(\mathbf{z})$ are $p_{\text{train}}(\mathbf{x})$ and $p_g(x)$, D_{θ_d} converges to $\frac{p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x})+p_g(\mathbf{x})}$ by Equation (2.5). When D_{θ_d} converges, the right-hand side of Equation (2.5) can be rewritten as:

$$\mathbb{E}_{X^{\text{train}}}\left[\log \frac{p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})}\right] + \mathbb{E}_{\hat{X}}\left[\log\left(1 - \frac{p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})}\right)\right] \tag{2.8}$$

$$= \int p_{\text{train}}(\mathbf{x}) \log \frac{p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})} d\mathbf{x} + \int p_g(\mathbf{x}) \log\left(1 - \frac{p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})}\right) d\mathbf{x} \tag{2.9}$$

$$= \int p_{\text{train}}(\mathbf{x}) \log \frac{2p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})} d\mathbf{x} - \int p_{\text{train}}(\mathbf{x}) \log 2 d\mathbf{x} + \int p_g(\mathbf{x}) \log\left(1 - \frac{2p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})}\right) d\mathbf{x} - \int p_g(\mathbf{x}) \log 2 d\mathbf{x} \tag{2.10}$$

$$= \int p_{\text{train}}(\mathbf{x}) \log \frac{2p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})} d\mathbf{x} + \int p_g(\mathbf{x}) \log\left(1 - \frac{2p_{\text{train}}(\mathbf{x})}{p_{\text{train}}(\mathbf{x}) + p_g(\mathbf{x})}\right) d\mathbf{x} - 2\log 2 \tag{2.11}$$

$$= 2D_{\text{JS}}(p_{\text{train}}(\mathbf{x}) \parallel p_g(\mathbf{x})) - 2\log 2, \tag{2.12}$$

where D_{JS} is the Jensen-Shannon (JS) divergence. Equation (2.12) means maximising the right-hand side of Equation (2.5) makes $p_{\text{train}}(\mathbf{x})$ and $p_g(\mathbf{x})$ similar distributions. Subsequently, maximising the right-hand side of Equation (2.4) leads to the maximisation of the right-hand side of Equation (2.1). As a result of such optimisation, the output of $G_{\theta_g}(\mathbf{z})$ eventually becomes similar to the training dataset (Figure 2.4).

In order to apply the Convolutional Neural Network approach that specifically targets convolutional feature extraction from images to GAN, the variant of GAN called Deep Convolutional GAN (DCGAN) [65] was proposed.

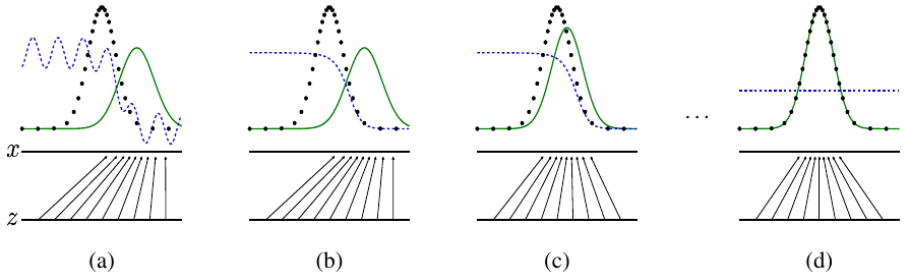


Figure 2.3: Visualisation of the training procedure of GAN (from [12]). The black dots shows real data $p_{\text{train}}(\mathbf{x})$ and green line illustrates the model distribution $G_{\theta_g}(\mathbf{z})$. The blue dashed line means the boundary of the discriminator D_{θ_d}. The training is started from (a) and eventually converged to (d) by iteratively updating D_{θ_d} and G_{θ_g}.

Figure 2.4: Examples of samples from GAN (from [12]) trained within a) MNIST [13] and b) Toront Face Database [14] datasets.

Conditional GAN

Whilst a basic (vanilla) DCGAN generates images based on whether they are determined as real or not by the discriminator without any other constraints and hence does not have the ability to output class dependent images, conditional GAN (cGAN) [16] modifies the GAN architecture to take account of classes by adding class labels into the inputs of the generator and discriminator. Equation (2.6) is modified as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}(\mathbf{x})}}[\log D(\mathbf{x} \mid \mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{x}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{y}) \mid \mathbf{y}))], \quad (2.13)$$

where $\mathbf{y}$ is the category label given in the objective function. Following this original cGAN, which uses the product of the input images and the one-hot class vectors to incorporate the class labels, other cGANs that apply the class conditions in

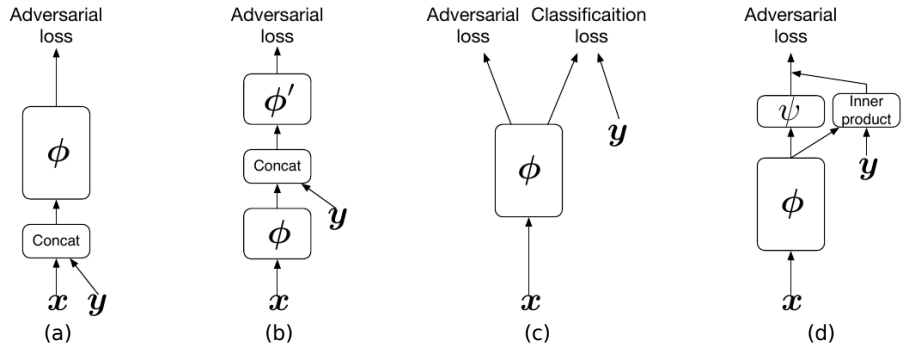


Figure 2.5: Different implementations of cGAN discriminators (from [15]): (a) the discriminator of the original cGAN [16] accepts the concatenation of the input images and classes, (b) the text-conditional GAN [17] concatenates the embedding features of the input images and the conditional information, (c) ACGAN [18] trains its discriminator also as a classifier, and (d) the project discriminator [15] use the inner product of the original discriminator outputs and the embedded vectors of the conditional information.

different manners have been proposed to improve the quality of the class dependent image generation (Figure 2.5). The text-conditional GAN [17], which is specified to text-to-image tasks, implements cGAN as a concatenation of latent features of the input images and conditional text information. This latent features of the images, which is also called embedding features, are extracted by the additional encoder (Figure 2.5(b)). Auxiliary Classifier GAN (ACGAN) [18] implements a classification model in addition to the generative model. This architecture trains its network to minimise the distance between both the real and fake data examples and the actual and predicted category labels (Figure 2.5(c)). Whilst such conditional information was implemented as a concatenation of the input and output of the networks, methods applying embedded features of the condition to the factors of the normalisation layers of the generator networks have been proposed [66] [67]. The normalisation layers in such methods are called the conditional normalisation layers and the generators are modified as $G(\mathbf{z}, e(\mathbf{y}))$, where e is the embedding function. These extensions to the GAN concept have illustrated strong improvement in the quality of the images generated. Furthermore, the effectiveness of embedding condition labels not only to the generator but also to the discriminator was illustrated [15]. This discriminator, which is called the projection discriminator, is implemented with an inner product of the original discriminator outputs and the

Figure 2.6: An illustration of the mode collapse problem on a 2D toy dataset (from [19]). This visualises a heatmap of the generator distribution after increasing numbers of training steps. The generator rotates through the modes of the target distribution and assigns significant probability mass to a single data mode at once (never converges to a fixed distribution).

embedded vectors of the labels as the outputs (Figure 2.5(d)).

Mode Collapse Problem and Solutions

Whilst the minimax game of Equation (2.4) and (2.5) requires finding a Nash equilibrium of a non-convex game with continuous and high-dimensional parameters, the GAN training uses gradient descent techniques to minimise a specified cost function [68]. This exceedingly simplified training of GAN often leads to unstable optimisation, sensitiveness of hyperparameters, and a failure of convergence. One typical consequence of such malignant characteristics of GAN training is a wrongly trained generator that always outputs only one or few modes of training dataset, so-called the "mode collapse" issue [69]. The mode collapse hugely limits the variety of the generated samples (Figure 2.6).

This unconverged behaviour is occurred by the alternating gradient descent that optimises θ_g fixing θ_d and vice versa iteratively. The two optimisation procedures circulate around to chase each other without convergence. Unrolled GAN [19] addresses this issue by taking into account the optimisation path of θ_d during the optimisation of θ_g. PacGAN [70] modifies the discriminator to accept the pack of samples to mitigate the mode collapse by expanding the learning targets p_{train}, p_g to the product distributions of $p^m_{\text{train}}, p^m_g$. AdaGAN [71] employs a weighted mixture of multiple generators to prevent failing to learn the modes in data distribution. Dp-GAN [72] and MSGAN [73] directly regularise the diversity of the generator outputs.

Another factor of the unstable GAN training comes from the uncontinuousness of the objectives such as the JS divergence based loss function like Equation (2.12). WGAN [74] addresses this discontinuous loss function by replacing it with the Earth Mover distance restricted as a 1-Lipschitz continuous function. Meanwhile, spectral normalization [75] considers a much more general approach to realise the Lipschitz continuity by normalising the weights of a discriminator. Such a variety of studies to improve GAN training have significantly reduced the chance of mode collapse within contemporary GAN formulations [19] [70] [71] [72] [73] [74] [75].

2.2.2 Variational Autoencoders

Variational Autoencoders (VAE) [57] are generative models using Autoencoder (AE) networks. AE is a kind of dimension reduction process of $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}} \rightarrow \mathbf{z} \in \mathbb{R}^{d_\mathbf{z}}$, $(d_\mathbf{x} > d_\mathbf{z})$ using a encoder $E_{\theta_e}(\mathbf{x})$ and decoder $G_{\theta_g}(\mathbf{z})$ networks. The encoder and decoder are jointly trained as:

$$\theta_e, \theta_g = \arg\min \mathbb{E}_{X^{\text{train}}}[l(\mathbf{x}, G_{\theta_g}(E_{\theta_e}(\mathbf{x})))], \tag{2.14}$$

where l is an error function such as squared errors. Optimising Equation (2.14), $E_{\theta_e}(\mathbf{x})$ is learnt to squeeze significant information of $\mathbf{x}$ as codes $\mathbf{z}$.

VAE uses such AE networks but originates from a different mathematical formulation as generative models. It is not feasible to directly solve Equation (2.1) because X^{train} contains high dimensional information and Equation (2.3) requires the calculation of all combinations of X and $\mathbf{z}$. It is solvable if $p_\theta(\mathbf{z}|X)$ is provided because the $p_\theta(\mathbf{x})$ can be straightforwardly calculated by $p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}|\mathbf{x})$. Since it is not possible to directly obtain $p_\theta(\mathbf{z}|X)$, they try to approximate another function $q_\phi(\mathbf{z}|X)$ to $p_\theta(\mathbf{z}|X)$, namely the variational inference method, by minimising the Kullback-Leibler (KL) divergence $D_{\text{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z}|X))$, which can be expanded

as:

$$
\begin{aligned}
D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z}|X)) &= \int q_\phi(\mathbf{z}|X) \log \frac{q_\phi(\mathbf{z}|X)}{p_\theta(\mathbf{z}|X)} d\mathbf{z} && (2.15)\\
&= \int q_\phi(\mathbf{z}|X) \log \frac{q_\phi(\mathbf{z}|X) p_\theta(X)}{p_\theta(\mathbf{z}, X)} d\mathbf{z} && (2.16)\\
&= \int q_\phi(\mathbf{z}|X) \left(\log p_\theta(\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|X)}{p_\theta(\mathbf{z}, X)} \right) d\mathbf{z} && (2.17)\\
= \log p_\theta(\mathbf{x}) &+ \int q_\phi(\mathbf{z}|X) \log \frac{q_\phi(\mathbf{z}|X)}{p_\theta(\mathbf{z}, X)} d\mathbf{z} && (2.18)\\
= \log p_\theta(\mathbf{x}) &+ \int q_\phi(\mathbf{z}|X) \log \frac{q_\phi(\mathbf{z}|X)}{p_\theta(X|\mathbf{z}) p_\theta(\mathbf{z})} d\mathbf{z} && (2.19)\\
= \log p_\theta(\mathbf{x}) &+ \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|X)} \left[\log \frac{q_\phi(\mathbf{z}|X)}{p_\theta(\mathbf{z})} - \log p_\theta(X|\mathbf{z}) \right] && (2.20)\\
= \log p_\theta(\mathbf{x}) &+ D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z})) \\
& \quad -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|X)} \left[\log(p_\theta(X|\mathbf{z})) \right]. && (2.21)
\end{aligned}
$$

At this point, the objective function to be minimised is defined as:

$$
L_{(\theta,\phi)} = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|X)} \left[\log \frac{p_\theta(\mathbf{z}, X)}{q_\phi(\mathbf{z}|X)} \right] = -\log(p_\theta(\mathbf{x})) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z}|X)) \quad (2.22)
$$

Equation (2.22), called negative evidence lower bound (ELBO), is rewriten using Equation (2.21) as:

$$
L_{(\theta,\phi)} = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|X)} \left[\log(p_\theta(X|\mathbf{z})) \right] + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z})). \quad (2.23)
$$

$q_\phi(\mathbf{z}|X)$ and $p_\theta(X|\mathbf{z})$ can be implemented like $E_{\theta_e}(\mathbf{x})$ and $G_{\theta_g}(\mathbf{z})$. $p_\theta(X|\mathbf{z})$, which is optimised by minimising Equation (2.23), samples images similar to training dataset from the prior $p_\theta(\mathbf{z})$. VAE commonly makes the assumption that the prior $p_\theta(\mathbf{z})$ in Equation (2.23) is distributed with the Gaussian distribution. Since Equation (2.23) is not differentiable, the optimisation process employs the reparametisation trick [76].

Posterior Collapse Problem and Solutions

Such Gaussian VAE have often suffered from the issue that the latent space has little information of the input, namely 'posterior collapse' [77]. In order to well approximate the posterior related $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z}))$ in Equation (2.23), the training requires enough variety of samples to be able to imply the abstraction of the concept of the target data. Otherwise, the posterior approximation tends to provide a too weak or noisy signal, due to the inappropriate variance induced by the stochastic gradient approximation. As a result, the decoder may learn to ignore $\mathbf{z}$ and instead rely solely on the autoregressive properties of X, causing X and $\mathbf{z}$ to be independent [78]. Since the issue is highly related on $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|X) \parallel p_\theta(\mathbf{z}))$ in Equation (2.23) [79], δ-VAE [77] constrains this term to prevent the issue. Other solutions use alternative priors of the latents such as Gaussian mixture models [80], autoregressive models [81], Dirichlet prior [82], and the stick-breaking prior [83] based process [84].

In particular, Vector Quantised Variational AutoEncoder (VQ-VAE) [20] mitigates the posterior collapse by assuming a discrete latent prior, significantly improving the quality of image synthesis. Such discrete representations allow for image information to be represented as a smaller set of discrete countable vectors (a 'codebook') improving the performance of the generative models within imagery [85] [20] [23]. (Also, Feature Quantised Generative Adversarial Networks (FQ-GAN) [85] show that this discrete latent approach can be universally applied to GAN to improve performance.) VQ-VAE employs a discrete latent posterior using vector quantisation to capture important features from the input data (Figure 2.7), making the model robust against large variances and mitigating the posterior collapse issue. Assuming input data $\mathbf{x}$, the encoder of the model outputs $z_e(\mathbf{x})$ and the discrete latents $z_q(\mathbf{x})$,

$$z_{\mathrm{q}}(\mathbf{x}) = \mathbf{e}_k, \text{where } k = \arg\min_i \|z_{\mathrm{e}}(\mathbf{x}) - \mathbf{e}_i\|_2, \tag{2.24}$$

where $\mathbf{e}_i$ are the K embedding vectors $\mathbf{e}_i \in R^D, i \in 1, 2, \cdots, K$ in the codebook.

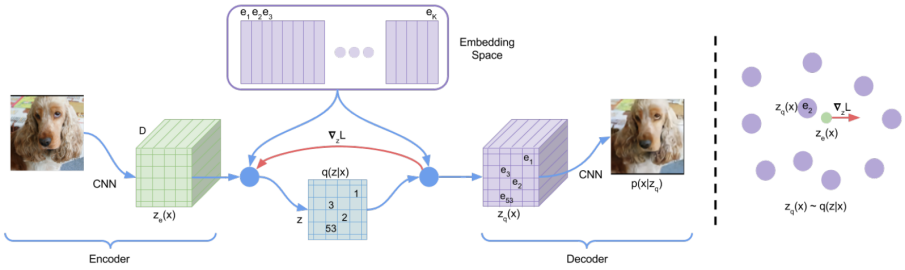


Figure 2.7: Conceptual illustration of VQ-VAE (from [20]). Images are encoded to a smaller set of discrete countable vectors and the the decoder maps between the space of the discrete vectors and the image data space.

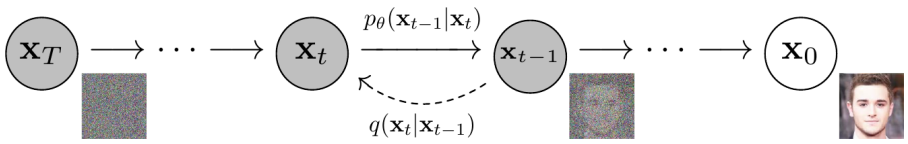


Figure 2.8: An illustration of DDPM (from [7]). The images $\mathbf{x}_0$ are sampled via Markov Chain Monte Carlo process progressively recovering from noise $\mathbf{x}_T$.

The training objective function is defined as:

$$L_{\text{vqvae}} = \log p\left(\mathbf{x}|z_{\text{q}}(\mathbf{x})\right) + \|\text{sg}[z_{\text{e}}(\mathbf{x})] - z_{\text{q}}(\mathbf{x})\|_2^2 + \beta\|z_{\text{e}}(\mathbf{x}) - \text{sg}[z_{\text{q}}(\mathbf{x})]\|_2^2, \quad (2.25)$$

where sg[·] represents the stop-gradient operator. Subsequently, the codebook is modelled by a CNN-based autoregressive model (PixelCNN) [86].

2.2.3 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) [7] have recently demonstrated the ability to produce higher quality and more varied data compared to that of other contemporary generative models. They are trained in a forward process based on Langevin dynamics [87], by gradually adding noise over a fixed number of timesteps, until all signal information is lost and the data closely resembles noise. Consequently, the trained model can be sampled by reversing the Markov Chain Monte Carlo (MCMC) process, starting from white noise and then iteratively denoising the transformed samples until they resemble meaningful high-quality images (Figure 2.8).

DDPM models data as latent variables of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})dx_{1:T}$,

where $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ are images, T is the length of the Markov chain, and $\mathbf{x}_1, ..., \mathbf{x}_T$ are latents of the same dimensions as the images. $p_\theta(\mathbf{x}_{0:T})$ is a Markov chain with learnt Gaussian transitions (the reverse process) where:

$$
\begin{aligned}
p_\theta(\mathbf{x}_{0:T}) &:= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) && (2.26)\\
p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\right) && (2.27)\\
p(\mathbf{x}_T) &= \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}). && (2.28)
\end{aligned}
$$

DDPM additionally approximates the posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ in the forward process. This Markov chain gradually adds Gaussian noise to the images:

$$
\begin{aligned}
q(\mathbf{x}_{1:T}|\mathbf{x}_0) &:= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) && (2.29)\\
q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right), && (2.30)
\end{aligned}
$$

where $\alpha_t \in \{\alpha_1, ..., \alpha_T\}$ are scheduled weights of the noise, therefore Equation (2.29) gradually adds Gaussian noise according to a variance schedule α_t. Equation (2.30) is a linear interpolation function of noise and images, which admits sampling $\mathbf{x}_t$ at an arbitrary timestep t as:

$$
\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{2.31}
$$

where $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To approximate $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, DDPM optimises the model parameter θ via denoising score matching (DSM) [88]. The loss function is thus redefined as a simpler form as:

$$
\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t)\|^2], \tag{2.32}
$$

where ϵ_θ is a non-linear function predicting the added noise $\boldsymbol{\epsilon}$ from $\mathbf{x}_t$ and t. Using an approximated ϵ_θ, μ_θ can be predicted as:

$$
\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right). \tag{2.33}
$$

Σ_θ in Equation (2.27) is set as $\Sigma_\theta(\mathbf{x}_t, t) = (1-\alpha_t)\mathbf{I}$. This admits sampling $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$:

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t, t) + \Sigma_\theta(\mathbf{x}_t, t)\boldsymbol{\epsilon}, \tag{2.34}$$

which leads to being able to sample $\mathbf{x}_0$.

DDPM has been used not only for unconditional image synthesis [89] [7] [90] [91], but they have also widely and successfully applied to various tasks including shape generation [92], super resolution [93] [94], image-to-image translation [95], and text-to-speech [96] [97].

The subsequent sections (Sections 2.2.4–2.2.5) review techniques that enhance the capability of such generative models.

2.2.4 Approaches for Large Image Generation

Whilst the recent advance of deep generative models enables synthesising quite high quality images, such models require very large neural networks with a huge number of parameters to accurately learn such high-dimensional data spaces. This in turn requires a significant amount of GPU-based memory for high resolution image output.

Patch-wise Training

To enable large image synthesis in a small GPU environment, it is a possible solution to separate large images into multiple smaller patches and train the patches. Stitching Across Frontier Network (SAFRON) [21] is a variant of GAN employing the generator accepting small patches of large images on cancer histology images and the discriminator accepting the stitched images of the patch-wise output of the generator (Figure 2.9). As a result, the generator outputs small patch samples that can be stitched to large images. Conditional Coordinating GAN (COCO-GAN) [98] trains the model with the small patches of large images and its spacial coordinates as conditions. Using this positional condition, the trained model can produce full-size

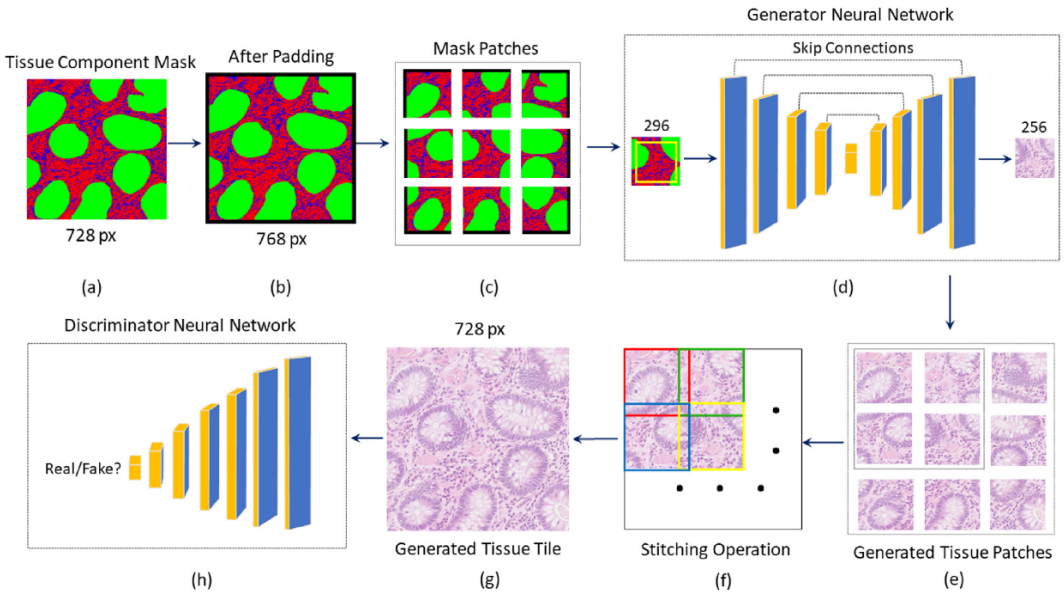


Figure 2.9: The training procedure of SAFRON (from [21]). The images (a) are separated into small patches (c) before the generator training (d). The discriminator (h) classify the stitched images (g) fabricated from the outputs of the patch-wise generator (e).

images in sampling. InfinityGAN [99] achieves seamless image patch generation by patch-wise training along with the condition of its global semantic contexts and coordinates. ∞-GAN [100] attempts to make consistency in image and latent spaces between input and its sub-region to generate image patches that have a smoothness to the neighbour patches.

Hierarchical Architecture

Hierarchical architectures for generative models assume images comprise hierarchical latent information and attempt to model such different levels of image information. Stacked GAN (StackGAN) [101] uses top-level and bottom-level hierarchical GANs for text-to-image generation. The top-level network creates primitive shape and colour of objects in low-resolution images and the bottom-level network subsequently refines the low-resolution outputs to synthesise high-resolution images. FineGAN [102] disentangles images to background, object shape, and object appearance features and hierarchically models each information. VQ-VAE-2 [6] modifies VQ-VAE to be able to generate high-fidelity images by employing top-

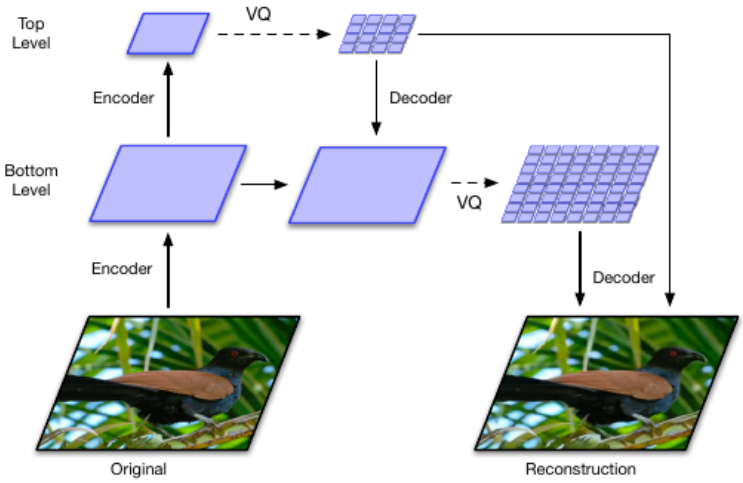


Figure 2.10: The hierarchical architecture of VQ-VAE2 (from [6]). The top and bottom level encoders and decoders separately model high-resolution images in the different levels of the information.

level and bottom-level hierarchical encoders/decoders and a self-attention powered PixelCNN (PixelSnail) [103] to enable high-resolution image synthesis (Figure 2.10). Further deeper hierarchical models of VAE [104] [105] achieve high-resolution and high-quality outputs. Similarly, ProGAN [106] and StyleGAN [5] attempt to model semantically coarse-to-fine information and progressively generate them by many-stacked hierarchical networks. DDPM indigenously has such a characteristic as the many denoising networks in the long Markov chain [1].

2.2.5 Generative Learning Trilemma and Combination Approaches

Whilst different kinds of generative models (Section 2.2.1–2.2.3) have different advantages and disadvantages, it is difficult for one type of generative models to simultaneously satisfy the following three key requirements: high-quality output, inexpensive sampling computation, and wide mode coverage [22] (Figure 2.11). To overcome this trilemma, the combined use of different models is considered as a solution.

[1]Recent work on diffusion models such as Latent Diffusion Models [40] introduce further improvement, which are discussed in Chapter 6

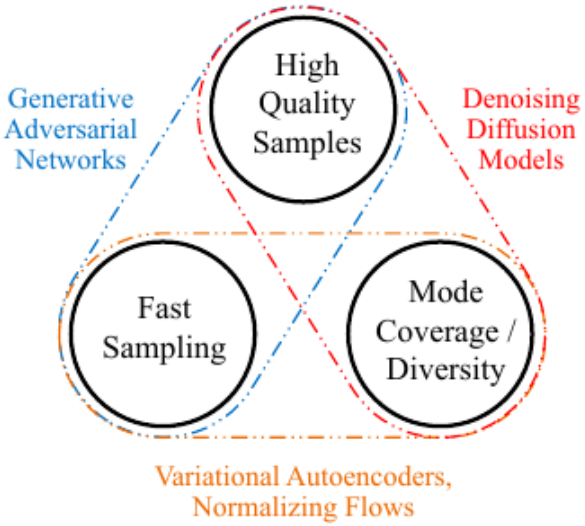


Figure 2.11: The trilemma on generative models (from [22]). Current generative learning framework cannot yet simultaneously satisfy high-quality sampling, mode coverage / diversity, and fast / computationally inexpensive sampling.

VAE can be used with GAN in combination [107] [108] [109]. VAE-GAN [107] replaces the element-wise loss in the discriminator in GAN with feature-wise loss to better capture the data distribution. Introspective Adversarial Networks (IAN) [110] trains the encoder via the same discriminator outputs the generator uses instead of the reconstruction loss of the decoder outputs. Adversarial Generator-Encoder (AGE) [108] employs two reconstruction losses of real and fake images on both data and latent spaces: $\mathbb{E}_{\mathbf{x}\sim P_{\text{data}}(\mathbf{x})}[\mathbf{x} - G(e(\mathbf{x}))], \mathbb{E}_{\mathbf{z}\sim p_x(\mathbf{z})}[\mathbf{z} - e(G(\mathbf{z}))]$. α-GAN [109] achieved better reconstruction performance than AGE by employing two discriminators on data and latent spaces.

The adversarial training of GAN can be applied to VQ-VAE (VQ-GAN, Figure 2.12) [23] to improve the output quality along with mitigating the mode collapse issue. ImageBART [111] employs a discrete DDPM to model the discrete latents in VQ-GAN. Taming Transformer [23] is the latest extension of VQ-VAE that is designed to learn the encoder and decoder not only using Equation (2.25) but also with both an adversarial loss [12] and a perceptual loss [107] to further improve latent quality and hence enable the decoder to synthesise higher quality images. Since the CNN-based autoregressive model of the conventional VQ-VAE relies only on convolutional density estimation and exhibits hard to capture long-

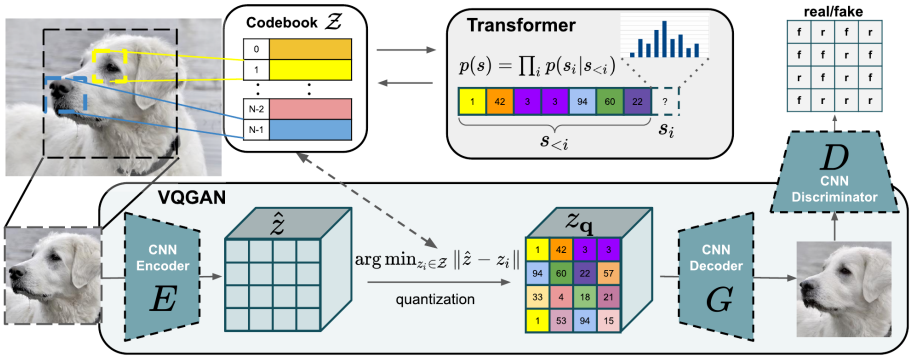


Figure 2.12: VQ-GAN (used in taming transformer [23]): Discrete latents are obtained via adversarial training applied VQ-VAE. The latent variables are modelled via transformer-based auroregressive models.

range interactions within high-resolution data, they apply a transformer [112] based autoregressive model to learn the prior of the codes. In general, a transformer is defined as a multi-layer function, as follows:

$$T(\mathbf{x}) = T_0(\mathbf{x}) \circ T_1(\mathbf{x}) \circ \ldots \circ T_N(\mathbf{x}), \tag{2.35}$$

$$T_n(\mathbf{x}) = f_n(A_n(\mathbf{x}) + \mathbf{x}), \tag{2.36}$$

where f_n is a small sub-function usually implemented as fully-connected and A_n is the self-attention function after projecting the input into three representations query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ as:

$$A_n(\mathbf{x}) = A'_n(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.37}$$

$$\mathbf{Q} = \mathbf{x}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{x}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{x}\mathbf{W}^V. \tag{2.38}$$

When the transformer is used for an autoregressive model, all entries below the diagonal of $\mathbf{Q}\mathbf{K}^T$ are masked (e.g. set to $-\infty$) in order to predict logits of the next sequence element without referring to future elements. To parallelise the attention function, a multi-head attention mechanism is used. This approach splits into h multiple sub-matrices $\mathbf{Q}_i$, $\mathbf{K}_i$, and $\mathbf{V}_i$ and linearly projects the sub-matrices

respectively as:

$$\mathbf{Q}_i = \mathbf{Q}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{K}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{V}\mathbf{W}_i^V, \quad i \in [1, h]. \tag{2.39}$$

Subsequently, the attention (Equation 2.37) is applied to each $\mathbf{Q}_i$, $\mathbf{K}_i$, and $\mathbf{V}_i$ in parallel and the outputs (heads) are unified as one output using concatenation and a linear projection $\mathbf{W}^o$ as:

$$A_n^{\text{mult}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{cat}\left(A'_n(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1, \ldots, A'_n(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h)\right) \mathbf{W}^o, \tag{2.40}$$

where $\text{cat}(\cdot, \ldots, \cdot)$ is a concatenation operation.

2.3 Generative Model based Image-to-image Translation

Image-to-image (I2I) Translation is a class of computer vision tasks where the goal is to obtain the mapping functions between different image domains, such as style transfer [63], colourisation [113], super-resolution [114], photorealistic image synthesis [115], and domain adaptation [116]. This I2I mapping can be learnt as a conditional generative model that accepts a source image as a condition and models the density of the target image. In this perspective, $p_\theta(X|\mathbf{z})$ in Equation (2.3) is rewritten as $p_\theta(X^t|X^s)$, where X^s, X^t are the source and target images, respectively. It can be recognised that I2I translation is a generative model using X^s as the prior instead of known-prior $\mathbf{z}$. I2I translation is classified into two types by the training procedures: paired and unpaired.

2.3.1 Paired Image-to-image Translation

Paired I2I translation assumes the pairs of source domain images and corresponding target domain images are given in training.

Earlier work proposed to use a pre-trained CNN and Gram matrices to obtain the perceptual decomposition of images [11]. This separates the image content and

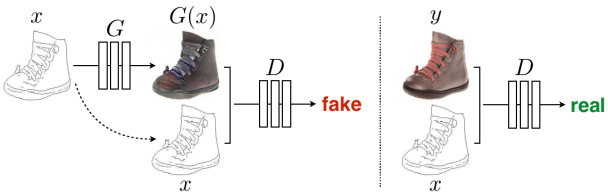


Figure 2.13: The conceptual illustration of the training process of pix2pix within a edges → photos task (from [24]). The generator G is learnt the mapping of edges → photos. The discriminator D is learnt as cGAN conditioned with the input edge images.

style, enabling style variation whilst preserving the semantic content. Many recent I2I approaches are trained adversarially with a GAN [12]. Pix2Pix [24] provides a general-purposed adversarial framework to transform an image from one domain to another (Figure 2.13). Instead of an autoencoder, U-Net [117] is utilized to share the low-level information between the input and output. BicycleGAN [118] combines conditional VAE-GAN (CVAE-GAN) with an approach to recover the latent codes, which improves performance, where the CVAE-GAN reconstructs category specific images [119].

2.3.2 Unpaired Image-to-image Translation

Whilst paired I2I translation requires a pair-wise training dataset of source and target domain images, it is hard for many tasks to prepare such a paired image dataset. Unpaired I2I translation is a solution for such limitations of the training dataset requirement. This unpaired approach learns the function that accepts source domain images and outputs images that resemble target domain images but preserve its semantic content information using a training dataset comprising unaligned source and target image sets.

Cycle-Consistent GAN (CycleGAN) [8] is one of the expansions of GAN specified in unpaired I2I translation. In this method, G and D in Equation 2.7 are trained to transfer from source images $\mathbf{x}_s \in X_s$ to target images $\mathbf{x}_t \in X_t$. Not only a lateral transform G, it learns bilateral transform paths $G_t(\mathbf{x}_s), G_s(\mathbf{x}_t)$. In addition, this adopts a new loss measure named a cycle-consistency loss $L_{\text{cyc}}(G_s, G_t)$ (Figure 2.14),

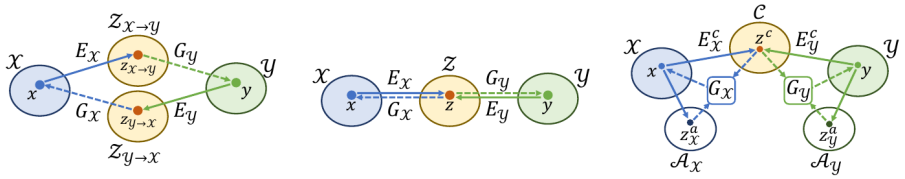


Figure 2.14: Architectures of (left) CycleGAN, (centre) UNIT, and (right) DRIT++ (from [25]).

which is represented as:

$$L_{\text{cyc}}(G_s, G_t) = \mathbb{E}_{\mathbf{x}_s \in X_s}[\|G_s(G_t(\mathbf{x}_s)) - \mathbf{x}_s\|_1] + \mathbb{E}_{\mathbf{x}_t \in X_t}[\|G_t(G_s(\mathbf{x}_t)) - \mathbf{x}_t\|_1]. \quad (2.41)$$

The full objective function of CycleGAN is defined as:

$$\min_{G_s, G_t} \max_{D_s, D_t} V(D_s, G_s) + V(D_t, G_t) + \lambda_{\text{cyc}} L_{\text{cyc}}(G_s, G_t), \quad (2.42)$$

where λ_{cyc} is a cycle-consistency loss weight. Unsupervised Image-to-Image Translation Networks (UNIT) [120] further make a share-latent assumption and adopt coupled GAN [121] in their method (Figure 2.14). To address the multimodal problem, Multimodal UNIT (MUNIT) [122], Diverse Image-to-Image Translation via Disentangled Representations (DRIT++) [25] (Figure 2.14), augmented CycleGAN [123] adopt a disentangled representation to further learn diverse I2I translation from unpaired training data. Moreover, [124] employs shared and exclusive representations, which are associated with the content and style information of images, respectively, for the translation of the styles.

2.4 Image Fusion

Blending two or multiple image samples is a possible remedy for overfitting problem or a less diversity in the datasets. Even simple alpha-blending of randomly chosen pairs of images contributes improving object classification performance and blending in a feature space learnt via DNN provides further plausible mixed images (reviewed in Section 2.4.1). The feature blending idea is also used in few-shot generative model tasks. This fusion-base few-shot training approach learns the generator that fuses

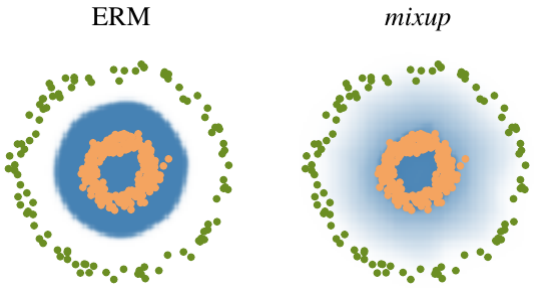


Figure 2.15: Visualisations of the probability of a classifier output $p(\mathbf{y} = 1|\mathbf{x})$ (blue) in a binary classification task (green: $\mathbf{y} = 0$ and orange: $\mathbf{y} = 1$) without mixup (left) and with mixup (right) (from [26]).

the features of multiple samples and outputs fused images that are as natural as the training real samples (reviewed in Section 2.4.2).

2.4.1 Class-wise Interpolation for Object Classification

Object classification learns the statistics of each class from training samples and infers the classes of unseen samples based on the likelihood in the statistics. Such inference is often degraded by an erroneously learnt manifold due to insufficient variation or imbalanced training samples. To improve the training, Mixup [26] blends the randomly sampled pairs of training data and class labels before using training. Assuming the pairs of training data and class labels as $(\mathbf{x}_i, \mathbf{y}_i)$, the blended pairs $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$ are defined as:

$$\bar{\mathbf{x}}_k = \lambda \mathbf{x}_i + (1 - \lambda \mathbf{x}_j), \bar{\mathbf{y}}_k = \lambda \mathbf{y}_i + (1 - \lambda \mathbf{y}_j), \tag{2.43}$$

where $\lambda \in [0, 1]$ is a weight following the beta distribution $\mathrm{Beta}(\alpha, \alpha)$, in which α is constantly set. This mixup training approach provides smoother classification boundaries (Figure 2.15) that enable wider coverage of unseen samples. Similarly, cutmix [125], which draws from mixup and cutout [126] as replacing a sub-region of an image with a region of the same size from another image, also provides an improvement in performance. Mixmo [127] applies mixup and cutmix to a multi-input multi-output (MIMO) training [128] to further improve the performance on image classification tasks. These blending approaches are used for adversarial

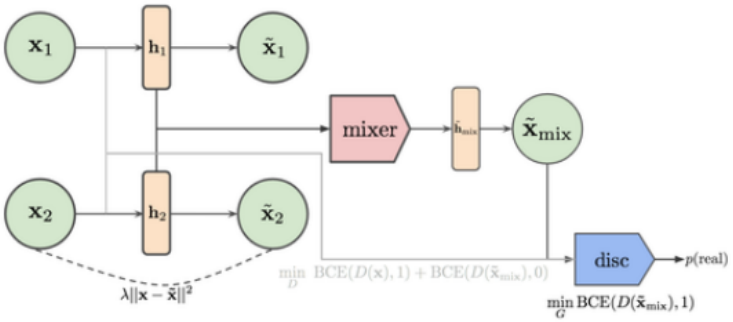


Figure 2.16: Architecture of Adversarial Mixup Synthesis (from [27]). The latent features of two images are mixed and decoded to a fused image. The synthesised image is input to a GAN discriminator.

training of GAN [26] [129] [130] and contribute to the stability of the training.

Manifold Mixup [131] blends instances not only in the input data space but also in feature spaces in the model to enable fusing higher level information. Similarly, Adversarial Mixup Synthesis [27] fuses the latent variables in the autoencoder and trains the decoder to make the fused images similar to natural images using a GAN framework (Figure 2.16). AlignMixup [132] employs positional feature alignment using Sinkhorn Distances [133] in ManifoldMixup to match the meaningful positions of the blending images. Meanwhile, OptTransMix and AutoMix [134] realise such higher level blending in different ways using the KL divergence and Wasserstein distance.

2.4.2 Multiple Instance Fusion for Few-shot Learning

Many feature fusion approaches have been adopted in few-shot learning tasks. Generative Matching Networks [135] combine Matching Networks [136] with VAE by decoding fused features after the matching network in order to diversify the generative model outputs in few-shot training tasks. MatchingGAN [28] replaces the VAE with GAN to achieve more plausible and high-resolution outputs.

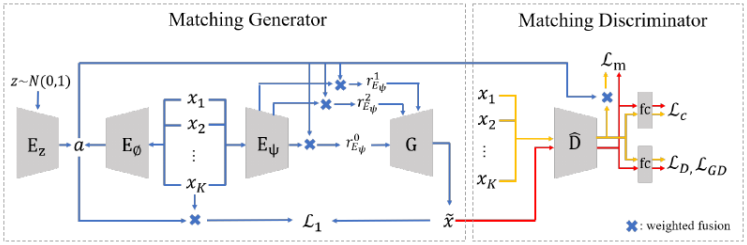


Figure 2.17: Architecture of MatchingGAN (from [28]). The matching generator creates a randomly weighted fusion of multiple images. The matching discriminator is trained with the real multiple images and the fake synthesised image.

2.5 Comparison of Image Diversification Approaches

Traditional DA (Section 2.1) generally improves performances on object classification tasks by diversifying the appearance of the training images. However, this approach is not always applicable; for example, a classification task on the MNIST dataset [13] requires the recognition of number digits that are not allowed flipped/rotated variances. Class-wise interpolation methods such as mixup (Section 2.4.1) improve object classification performances without the transform operation by smoothing the decision boundary but this linear blending operation produces unnatural images. Unlike these simple operation approaches, GAN (Section 2.2.1) and CycleGAN (Section 2.3) synthesise new but plausible images from the model learnt from training datasets but the generated images do not have a significant effect on object classification tasks [137]. To sum up, these diversification approaches have different advantages and disadvantages.

To develop a stronger image diversification method, this thesis investigates two strategies mitigating the disadvantages and enhancing the advantages. The first one is the combined use of CycleGAN and mixup to create synthetic images that is simultaneously similar to the training dataset and improve object classification performances (Chapter 3). The second one is DDPM [7] based I2I translation instead of GAN to diversify the output by making use of the high mode coverage ability of DDPM (Chapter 4).

Whilst such generative models fabricate new images by learning existing images, these data-driven approaches require huge computation resources and the outputs

are degraded when the number of the training dataset is small [60]. To overcome these limitations, we investigate an approach incorporating a patch-wise training (Section 2.2.4) into a high mode coverage DDPM and employing a hierarchical architecture (Section 2.2.4) of VQ-VAE [20] and the DDPM (Chapter 5).

2.6 Evaluation Techniques for Quality and Mode Coverage

This thesis aims to find methods to realise high quality and wide mode coverage of generative model outputs. However, it is intrinsically difficult to evaluate the images sampled via generative models in terms both of quality and mode coverage due to no ground truth of this unsupervised task. Under this difficulty, various methods have been proposed to pursue a plausible evaluation of such generative model outputs. The generated images can be evaluated as qualitative and quantitative measures.

2.6.1 Qualitative Evaluation

The most straightforward and trustworthy approach is showing people the actual outputs. Humans can be convinced as the output images are plausible by seeing actual outputs. Asking people to respond a questionnaire about how the images look provides a better analysis and increasing the number of the respondents can statistically strengthen the evaluation. For this purpose, a crowdsoucing service such as Amazon Mechanical Turk [138] has been widely used to collect a large number of answers within a short term with low cost. Meanwhile, this crowdsoucing approach has also been criticised as exploitation of workers [139] and increasingly recognised as an unsuitable approach for long-term evaluation within this field.

Along with observing images themselves, low-dimensional projection of the images is commonly adopted for evaluation. The t-distributed stochastic neighbor embedding (t-SNE) [29] maps high-dimensional data to 2 or 3-dimensional points by non-linear dimensional reduction techniques (Figure 2.18). Comparing the projected points of real and generated images, we can evaluate how close those image sets are

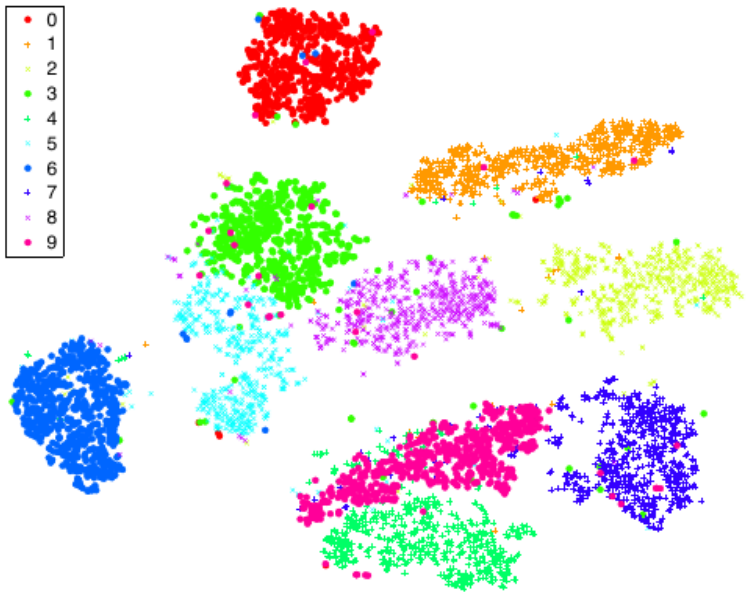


Figure 2.18: Visualizations of 6,000 handwritten digits from the MNIST data set. (from [29]).

and the scatterness of the points enables us to observe the diversity of the images.

2.6.2 Quantitative Evaluation

Many measures have been proposed for numerically calculating similarity between given image datasets and diversity of generated images. Generative models can be quantitatively evaluated by comparing such measures. Also, it is convincing to observe the numerical performances of downstream machine learning tasks when the synthetic images are used as a training dataset.

Comparing pre-trained model outputs

One of the most popular method for the quantitative evaluation is Inception Score (IS) [68], which uses the outputs of Inception-v3 classification network [140] trained on ImageNet [141]. They define the realness by how the output is unique on a specific class and the diversity by how the mean output of an entire dataset is close to uniform. To jointly evaluate both as unified values, they use the Kullback-Leibler divergence of the predicted class distribution on each image and the mean of the predicted class distribution across all images. Whilst this method has

provided common evaluation on many research, there are some limitations such as that it does not well support image classes out of the training set that is used for Inception-v3 training [142]. Therefore, another popular method Fréchet Inception Distance (FID) [33] uses the features that are output from the activation vector of the last pooling layer instead of the final classification layer. Unlike IS, FID calculates Fréchet distance [143] between two gaussian distributions whose means and variances are fit to the features from real images and generated images, respectively. This comparison of means and variances simultaneously measures the quality and diversity of the generated images. They state this method provides human-like perception of the similarity between real and generated images. FID is one of the most popular method to evaluate outputs of generative models as many researchers adopt it to compare their method with other methods. Spacial FID (sFID) [144] is a variant of FID that uses the first 7 channels from the intermediate mixed 6/conv feature maps along with the last pooling layer to consider spacial variability. IS and FID often create bias, i.e. dependence on the number of samples, that is, the scores are varied by the sample quantity [145]. To mitigate this bias, $\overline{IS}_\infty$ and $\overline{FID}_\infty$ [145] calculate the scores assuming an infinite number of samples.

Meanwhile, Learned Perceptual Image Patch Similarity (LPIPS) [34] uses the intermediate features of VGG16 [61] to measure the distance between two images. This distance has been used to evaluate the diversity of generated images by calculating the empirical mean of the distances of randomly chosen pairs of the images. Perceptual Path Length (PPL) [5] evaluates how much an image space and learnt latent space are related by analysing the smoothness of LPIPS when changing the latent variables.

Unlike IS, FID, and LPIPS, the precision and recall for distribution (PRD) [146] measures how many generated images are close to the given real images and how many the real images are close to the generated images. However, PRD can not measure an identicalness of the distributions of real and generated images and is not robust against outliers. To address those issues, PRDC [147] proposes the new measures of 'density' and 'coverage'. The density counts how many regions abound real samples cover generated samples and the coverage counts how many generated

samples are covered in regions around real samples.

Evaluation on downstream tasks

Applying to actual downstream tasks is a convincing evaluation from a practical aspect. For example, observing the impact on an object classification performance of the case that includes generated images provides plausible insights into the image diversification analysis. This approach evaluates performances only on specific tasks but it can be a strong evaluation by using popular datasets or tasks from publicly available resources such as Kaggle [148].

These evaluation methods provide validation from different perspectives. Therefore, this thesis adopts the use of multiple evaluation methods.

CHAPTER 3

Image Fusion in Unpaired Image-to-image Translation

This chapter discusses exploiting the potential of image-to-image (I2I) translation as a dataset diversification strategy and develops a new I2I translation model, adopted from Cycle-Consistent Generative Adversarial Networks (CycleGAN) [8]. In particular, we modify CycleGAN by manipulating class conditional information and generating class-interpolated fused images (Figure 3.1), as described in detail in Section 3.2. The proposed approach enables the expansion of the mode of the generated images as the solution of Research Question 1 in Section 1.3. The experiments supporting our method, within the context of Synthetic Aperture Radar (SAR) image object classification using a variation of the Statoil/C-CORE Iceberg Classifier Challenge dataset [149][1], are presented in Section 3.3 with a subsequent summary presented in Section 3.4.

3.1 Motivation

As discussed in Section 2.3, I2I translation can produce new images similar to a target domain by transferring images in another (source) domain. In particular,

[1] We ablate this dataset in order to use for our evaluation. The detail is described in Section 3.3.1.

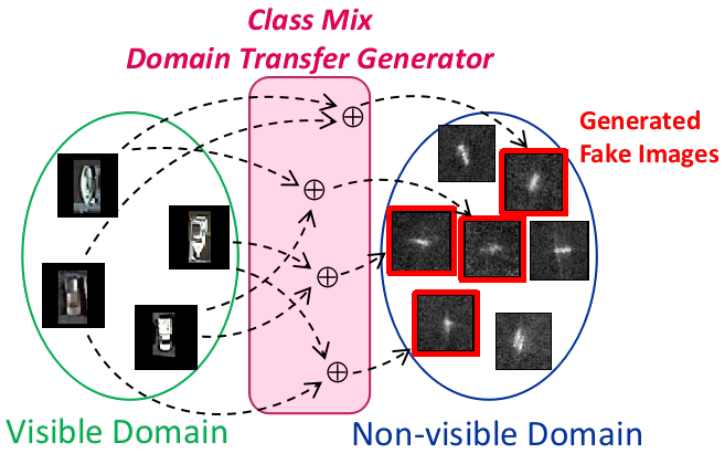


Figure 3.1: Conceptual illustration of our novel data augmentation approach for generating cross-domain, class-interpolated image instances.

unpaired I2I translation such as CycleGAN allows the use of unaligned training datasets of source and target domains. In other words, unpaired I2I models can be learnt by a small amount of a target domain dataset and a large amount of another domain dataset and the trained model can increase the small target domain dataset by the translation from another domain. However, especially in such a small dataset situation, the translation produces images that do not improve object classification performances as mentioned in Section 2.5. Furthermore, the mode collapse issue (Section 2.2.1) occurs when the number of target domain samples is small. The mixup operation of real and generated samples in GAN training mitigates this issue [26]. This mixup GAN training alters Equation (2.6) as:

$$V(\theta_g, \theta_d) = \mathbb{E}_{X^{\text{train}}}[\log \lambda D_{\theta_d}(\lambda X^{\text{train}} + (1-\lambda)\hat{X})], \tag{3.1}$$

where $\hat{X}$ is the generated samples and λ is the mixup ratio as same as Equation (2.43). MixCycleGAN [130] applies CutMix [125] operation to the CycleGAN process to stabilise the training. This method splits an input image into two rectangular regions vertically or horizontally and replaces one region with that of another image:

$$\bar{\mathbf{x}} = \text{cat}(\mathbf{x}_1[:\lambda H, :], \mathbf{x}_2[(1-\lambda)H :, :]) \text{ or } \text{cat}(\mathbf{x}_1[:, :\lambda W], \mathbf{x}_2[:, (1-\lambda)W :]), \tag{3.2}$$

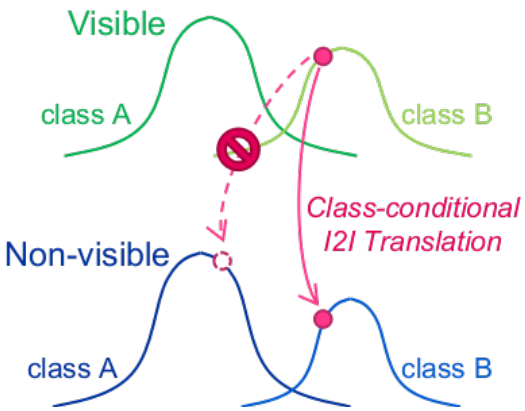


Figure 3.2: Class conditional I2I translation to match the classes between input and output images.

where $\bar{\mathbf{x}}$ is the mixed image, $\mathbf{x}_1, \mathbf{x}_2 \in X$ are the input images, H, W are the height and width of the input images respectively, and $\text{cat}(\cdot, \cdot)$ is a concatenation function. The preprocessed mixed image $\bar{\mathbf{x}}$ is input to the generator G of CycleGAN to synthesise a fake image. The discriminator D is modified to estimate the mixup ratio from the alpha-blended real and fake images, which is optimised as:

$$\hat{\theta}_d, \hat{\theta}_g = \arg\max_{\theta_g} \arg\min_{\theta_d} \mathbb{E}_{\mathbf{x} \in X} \left[\log \| D_{\theta_d}(\lambda \mathbf{x} + (1-\lambda) G_{\theta_g}(\bar{\mathbf{x}})) - \lambda \| \right]. \tag{3.3}$$

Although the use of the mixup operation provides the stability in the adversarial training, those are designed without the consideration of the trend of the image classes within image datasets; therefore, in the case of applying an object classification dataset that has an imbalance of samples over the classes, the translation result may not match the class of the input but mislead to the class that is frequently occurred in the training dataset whichever the class of the input is. In order to prevent these concerns, we employ a class conditional training approach for I2I translation, which forces the transition between source and target domains in the same classes (Figure 3.2). We assume that this class conditional model can also be used to diversify the output of I2I translation. Like Manifold Mixup [131], the trained class conditional model may produce images whose classes seem a mixture of two classes when the model receives the mixed class conditions and images. The fused class-interpolated images may provide much diversity in the target dataset.

In the following sections, we look into how the idea of this class-interpolated I2I translation works for data augmentation.

3.2 Methodology

This section describes the detailed process of our proposed method. The proposed method assumes a source domain dataset $(\mathbf{x}_s^i, \mathbf{y}_s^i) \in X_s^N$ and a target domain dataset $(\mathbf{x}_t^j, \mathbf{y}_t^j) \in X_t^M$ which consist of N and $M(\ll N)$ samples respectively. $\mathbf{x}_s^i$ and $\mathbf{x}_t^j$ are the images themselves and $\mathbf{y}_s^i$ and $\mathbf{y}_t^j$ are class labels. The types of classes are common in both domains.

3.2.1 Training a Conditional CycleGAN Model

Initially, an I2I translation model, which transfers between two different domains, is built using the conditional CycleGAN approach, which is the conditional GAN (Section 2.2.1) applied CycleGAN and implemented as replacing $V(\cdot, \cdot)$ in Equation (2.42) with one in Equation (2.13). The overall flow is shown in Figure 3.3a where, unlike ordinary CycleGAN, the generator and discriminator functions are conditioned on the class labels. The objective function is defined as a simple sum of weighted terms:

$$L = \lambda_s L_{G_s} + \lambda_t L_{G_t} + \lambda_s L_{D_s} + \lambda_t L_{D_t} + \lambda_s \lambda_{\text{cyc}} L_{\text{cyc}_s} + \lambda_t \lambda_{\text{cyc}} L_{\text{cyc}_t}, \tag{3.4}$$

where:

$$L_{G_s} = \mathbb{E}_{(\mathbf{x}_t^j, \mathbf{y}_t^j) \in X_t}[\log(1 - D_s(G_s(\mathbf{x}_t^j, e_t(\mathbf{y}_t^j)), e_s(\mathbf{y}_t^j)))], \tag{3.5}$$

$$L_{G_t} = \mathbb{E}_{(\mathbf{x}_s^i, \mathbf{y}_s^i) \in X_s}[\log(1 - D_t(G_t(\mathbf{x}_s^i, e_s(\mathbf{y}_s^i)), e_t(\mathbf{y}_s^i)))], \tag{3.6}$$

$$\begin{aligned} L_{D_s} &= \mathbb{E}_{(\mathbf{x}_s^i, \mathbf{y}_s^i) \in X_s}[\log(1 - D_s(\mathbf{x}_s^i, e_s(\mathbf{y}_s^i)))] + \mathbb{E}_{(\mathbf{x}_t^j, \mathbf{y}_t^j) \in X_t}[\log(D_s(G_s(\mathbf{x}_t^j, e_t(\mathbf{y}_t^j)), e_s(\mathbf{y}_t^j)))] \\ &+ \lambda_{\text{gp}} \mathbb{E}_{(\hat{\mathbf{x}}_s^j, \hat{\mathbf{y}}_s^j) \sim \mathbb{P}_{\hat{\mathbf{x}}_s, \hat{\mathbf{y}}_s}}[(\|\nabla D_s(\hat{\mathbf{x}}_s^j, e_s(\hat{\mathbf{y}}_s^j))\|_2 - 1)], \end{aligned} \tag{3.7}$$

$$\begin{aligned} L_{D_t} &= \mathbb{E}_{(\mathbf{x}_t^j, \mathbf{y}_t^j) \in X_t}[\log(1 - D_t(\mathbf{x}_t^j, e_t(\mathbf{y}_t^j)))] + \mathbb{E}_{(\mathbf{x}_s^i, \mathbf{y}_s^i) \in X_s}[\log(D_t(G_t(\mathbf{x}_s^i, e_s(\mathbf{y}_s^i)), e_t(\mathbf{y}_s^i)))] \\ &+ \lambda_{\text{gp}} \mathbb{E}_{(\hat{\mathbf{x}}_t^j, \hat{\mathbf{y}}_t^j) \sim \mathbb{P}_{\hat{\mathbf{x}}_t, \hat{\mathbf{y}}_t}}[(\|\nabla D_t(\hat{\mathbf{x}}_t^j, e_t(\hat{\mathbf{y}}_t^j))\|_2 - 1)], \end{aligned} \tag{3.8}$$

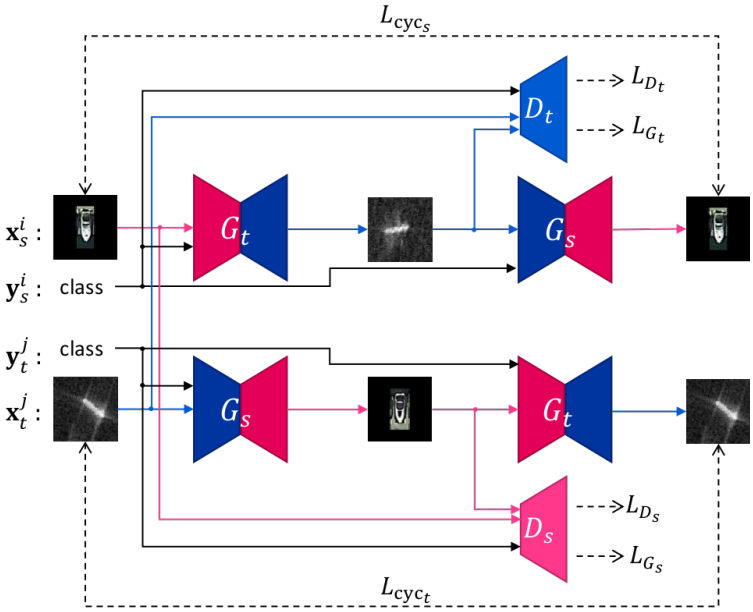
$$L_{\text{cyc}_s} = \mathbb{E}_{(\mathbf{x}_s^i, \mathbf{y}_s^i) \in X_s}[\|(G_s(G_t(\mathbf{x}_s^i, e_s(\mathbf{y}_s^i)), e_t(\mathbf{y}_s^i)) - \mathbf{x}_s^i)\|_1], \tag{3.9}$$

$$L_{\text{cyc}_t} = \mathbb{E}_{(\mathbf{x}_t^j, \mathbf{y}_t^j) \in X_t}[\|(G_t(G_s(\mathbf{x}_t^j, e_t(\mathbf{y}_t^j)), e_s(\mathbf{y}_t^j)) - \mathbf{x}_t^j)\|_1], \tag{3.10}$$

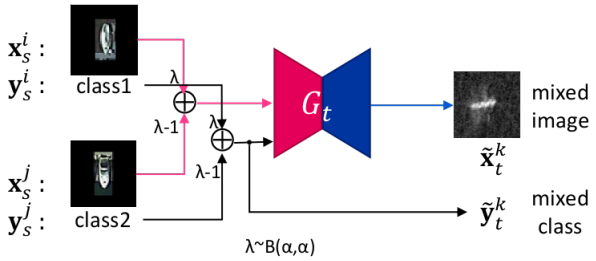
λ_s and λ_t are source domain and target domain weights, respectively, which balance the corresponding generator and discriminator functions with the cycle-consistency losses for both the source and target domains accordingly. λ_{gp} is a weight of the gradient penalty [150]. We implement the conditional regularisation of cGAN to our CycleGAN generators by using conditional batch normalization [151] in the networks instead of batch normalization [152]. Also, the projection discriminators [15] are applied to the discriminators in our networks both to improve the output quality and smooth the transition of the outputs across the class labels, which is a key technique to realise the synthesis of the class-interpolated images. In order to prevent mode collapse and stabilise training, Spectral Normalization [75] is combined with the gradient penalty as proposed in [153].

3.2.2 Adding Class-interpolated Domain-transferred Images

After training in Section 3.2.1, the model is used for the synthesis of new class-conditioned images via the domain transfer. A pair of images and class labels in the source domain dataset $(\mathbf{x}_s^i, \mathbf{y}_s^i), (\mathbf{x}_s^j, \mathbf{y}_s^j) \in X_s^N$ are used as an input. Subsequently, the input is processed to produce a tuple of a mixed image, label, and embedded



(a) Training



(b) Sampling

Figure 3.3: Overall flow of our conditional CycleGAN model. (a) The generator and discriminator are trained with the condition of object classes. (b) The generator synthesises a fused image from two images and the class conditions.

feature vector $(\bar{\mathbf{x}}_s^k, \bar{\mathbf{y}}_s^k, \bar{e}_s^k)$, defined by:

$$\bar{\mathbf{x}}_s^k = \mathbf{x}_s^i * \lambda + \mathbf{x}_s^j * (1 - \lambda), \tag{3.11}$$

$$\bar{\mathbf{y}}_s^k = \mathbf{y}_s^i * \lambda + \mathbf{y}_s^j * (1 - \lambda), \tag{3.12}$$

$$\bar{e}_s^k = e_s(\mathbf{y}_s^i) * \lambda + e_s(\mathbf{y}_s^j) * (1 - \lambda), \tag{3.13}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$ is the mixup ratio sampled from the beta distribution Beta, in which α is set as a constant as in [26]. As a result, the mixed pair $(\tilde{\mathbf{x}}_t^k, \tilde{\mathbf{y}}_t^k)$ that

is input to the generator and discriminator is defined as:

$$(\tilde{\mathbf{x}}_t^k, \tilde{\mathbf{y}}_t^k) = (G_t(\bar{\mathbf{x}}_s^k, \bar{e}_s^k), \bar{\mathbf{y}}_s^k). \tag{3.14}$$

Finally, N samples $\tilde{X}_t^N = \{(\tilde{\mathbf{x}}_t^k, \tilde{\mathbf{y}}_t^k)\}$ are synthesised as in Figure 3.3b. The new fake samples are combined with the original dataset as $X_t^M \cup \tilde{X}_t^N$, where we denote this method as Conditional CycleGAN Mixup Augmentation (C2GMA).

3.3 Evaluation

The method is evaluated in the context of the ships/icebergs SAR classification task using a variation of the Statoil/C-CORE Iceberg Classifier Challenge dataset [149]. Results are compared between classification models trained with and without existing dataset augmentation approaches in addition to our proposed CycleGAN driven C2GMA (Section 3.2) approaches.

3.3.1 Dataset

We choose SAR images as a small dataset to be increased in our experiments. SAR is a kind of radar which is generally mounted on aeroplanes or satellites to observe ground and whose imaging procedures and resulting images are quite different from visible cameras and images. The imaging process is realised by visualising the backscatter of the microwave signals that the radar emits (Figure 3.4). SAR has a stronger observation capability than visible sensors such as availability in bad weather conditions, providing penetration vision behind foliage, and so on, because microwaves have much more penetrability than the visible spectrum. Furthermore, some SAR equipment support polarimetry by capturing the polarisation between transmitted microwave and received echos. This polarisation is occurred depending on the material types of the reflection surfaces; therefore, this polarimetry SAR imagery can be used for material analysis. Whilst this imaging technology has many beneficial characteristics, developed images look different from visible natural images because of this active sensing process and the difference of the spectrum band.

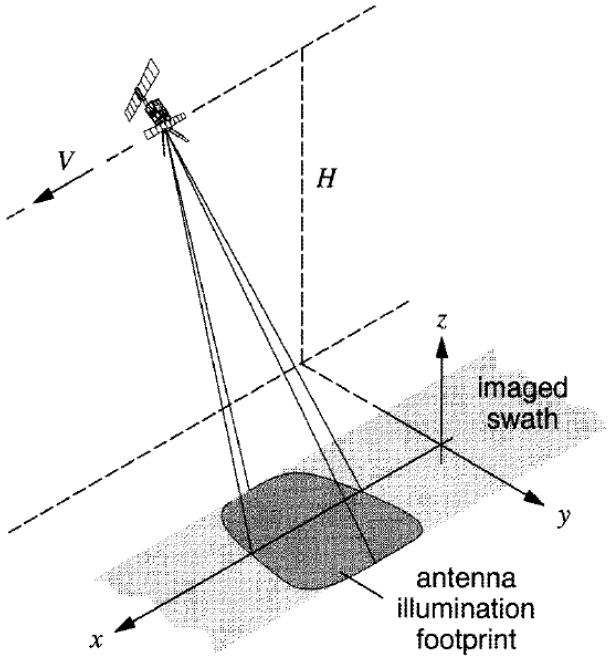


Figure 3.4: The principle of SAR (from [30]). The images are developed by comparing the intensities of the transmitted microwaves and received echoes. Simultaneously, the backprojeced information is synthesised across the moving path of the radar in order to compose an area of images.

This difference in the special imaging technology requires individual data collection because the appearances are differed by the specifications of SAR equipment. Moreover, data collection within SAR imagery is much more difficult than capturing visible imagery because the equipment and its associated cost are quite expensive. To realise the alternative way to increase the number of the samples in SAR image dataset, we apply our proposed method.

For the experiment for SAR imagery, we use the Statoil/C-CORE Iceberg Classifier Challenge dataset [149], which has a collection of satellite C-band SAR images of ships and icebergs from Sentinel-1 [154]. The training and test datasets are provided as combined JSON format [155] which contain 1,604 and 8,424 images respectively. Each image has two bands of 75×75 pixels of floating-point values with the unit being dB. The two bands represent the different channels of microwave echos: HH and HV. The values in the HH channel are the intensity of the horizontal echoes of the horizontaly transmitted microwave, whereas the HV channel is the intensity of the vertical echoes of the same transmitted microwave. We initially

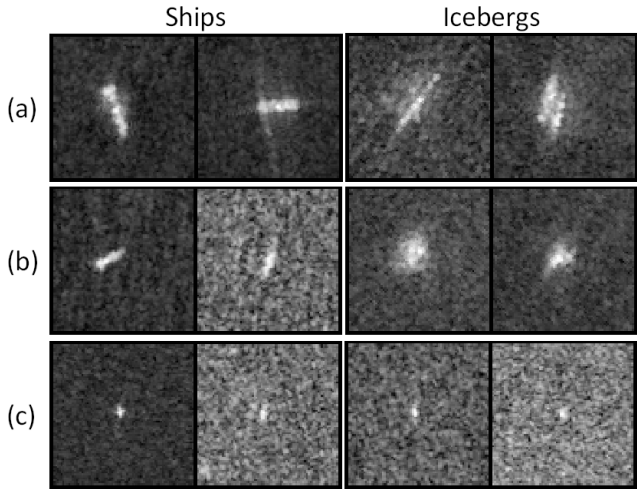


Figure 3.5: SAR ships/icebergs images divided into three groups based on difficulty of discrimination by distance, angle, object size, etc.

combine the two channels into one channel:

$$I(x,y) = \sqrt{I_{\mathrm{HH}}(x,y)^2 + I_{\mathrm{HV}}(x,y)^2}, \tag{3.15}$$

where $I(x,y)$, $I_{\mathrm{HH}}(x,y)$, and $I_{\mathrm{HV}}(x,y)$ are the pixel values of the combined image, the HH image, and the HV image at (x,y) respectively. Since the images in the test set do not have any class labels whilst the images in the training set are labelled as either a ship or an iceberg, we use only the labelled training data in our experiments (we split this labelled data into different groups for evaluation, discussed subsequently). A challenge of assessing the generalisation performance, given a dataset sampled from a single distribution, is that it does not reflect the case where the distribution of data under the expected testing conditions differs from the distribution of data sampled for training. Therefore, we split the dataset into three groups of discriminable classes, from which the images are sampled at different ratios between training and testing. The dataset is then subdivided into three groups by hand for each class: (a) easily discriminable sets, (b) moderately discriminable sets, and (c) difficult cases (Figure 3.5). Each of the groups is partitioned into training and testing splits and the training split is subsampled to three sets at different ratios, where specifically we distort the distribution of the training sets to simulate further imbalance and mismatch between the training distribution and the expected testing

Table 3.1: The number of samples in the experiment dataset separated by the test set and the three different training sets. The columns (a), (b), and (c) represent: easily identifiable samples, moderate samples, and difficult samples.

	Ship				Iceberg			
	(a)	(b)	(c)	total	(a)	(b)	(c)	total
Test	97	158	171	426	99	137	141	377
Train #1	96	15	17	128	99	13	14	126
Train #2	96	15	17	128	9	137	14	160
Train #3	96	15	17	128	9	13	140	162

data distribution. These splits, and the corresponding skewed subsamplings, are shown in Table 3.1.

In order to augment the training datasets using our proposed method, we use the satellite visible image dataset named DOTA [31], which is a collection of commercial satellite images containing many objects such as vehicles annotated with bounding boxes and class labels (Figure 3.6). Therefore we use SAR and visible image pairs with SAR images originating from the Statoil/C-CORE Iceberg Classifier Challenge dataset [149] and visible images from the DOTA [31] dataset. Due to the lack of iceberg visible images within either dataset, we pair iceberg SAR images from the Statoil/C-CORE Iceberg Classifier Challenge dataset [149] with representative non-ship images from the DOTA [31] dataset, for which purposes we use visible images of vehicles. Despite this obvious semantic mismatch in the second pairing, our I2I translation model specifically synthesises images conforming to the true distribution of the SAR iceberg images as enforced by the discriminator criteria of the loss function in Equation (3.8).

Initially, visible object images are extracted from the visible dataset using the annotations. Each extracted image is resized in the same way as the SAR image, and its rotations are adjusted accordingly. The backgrounds are set to black, which prevents including surrounding objects, which would be undesirable (Figure 3.7). The source domain visible dataset exhibits several images that are unclear or incorrect, as in Figure 3.8. Such images are eliminated based on their distances from the median of all of the images within each class. These distances are measured in the latent spaces trained by a Variational Autoencoder (VAE) [57] on individual

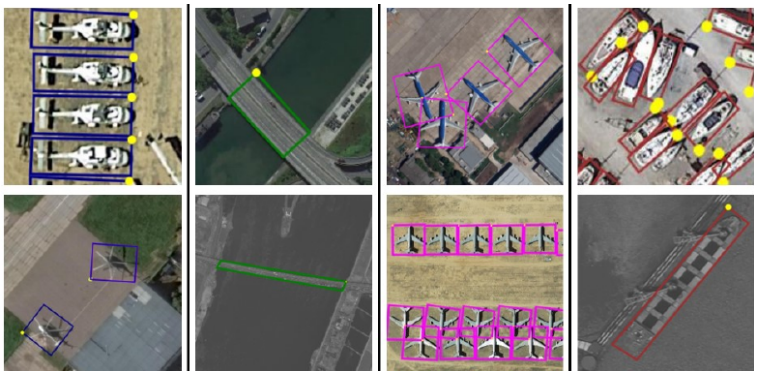


Figure 3.6: DOTA satellite image dataset with object annotations [31]. Our experiment use this visible dataset as source domain images for I2I translation.

Figure 3.7: Training samples within visible domain (domain transfer source) extracted from DOTA.

classes. The VAE is implemented based on a CNN-based architecture as shown in Figure 3.9. Using the encoder, all of the images are embedded in a lower dimensional latent space that follows an approximate normal distribution, and the distances of each sample $d(\mathbf{x}_i^c)$ are calculated:

$$d(\mathbf{x}_i^c) = \sqrt{(f_e^c(\mathbf{x}_i^c) - \mathcal{M}^c)^T S^{c-1} (f_e^c(\mathbf{x}_i^c) - \mathcal{M}^c)}, \tag{3.16}$$

$$S^c = \mathbb{E}[(f_e(\mathbf{x}_i^c) - \mathcal{M}^c)(f_e^c(\mathbf{x}_i^c) - \mathcal{M}^c)^T], \tag{3.17}$$

Figure 3.8: Poor quality visible images illustrating blurriness and multiple objects (which we eliminate).

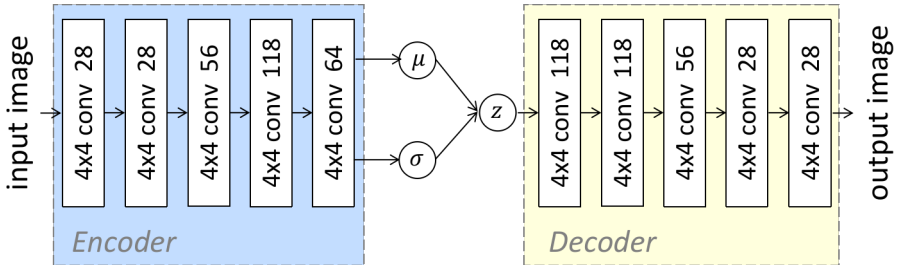


Figure 3.9: The architecture of the VAE for poor quality sample removal. The encoder and decoder consist of 5 convolution down/up sampling layers.

where $\mathbf{x}_i^c$ is the i-th input sample of class c, f_e is the encoder, and $\mathcal{M}^c$ is the median of the encoded features in class c. S^c is a normalisation factor for each dimension of the feature vectors in class c. Half of the shorter distance samples are selected for each class, subsampling 14,034 visible ship images and 13,063 visible vehicles, resulting in clearer data and higher-quality annotations for use as our source domain[2].

3.3.2 Training Domain Transfer Model

Domain transfer models, as described in Secion 3.2, are trained using the SAR images for each training split, where 1,500 ships and 1,500 vehicles images are subsampled from the visible images, prepared as previously outlined. The network architecture used in this experiment is shown in Figure 3.10, which follows a standard residual generative network, and the discriminator function uses spectral normalization on the convolutional layers. The network training parameters are: $\lambda_{\mathrm{s}} = \lambda_{\mathrm{t}} = 10.0$, $\lambda_{\mathrm{cyc}} = 1.0$, $\lambda_{\mathrm{gp}} = 0.01$, batch size $B = 32$, and update rate of discriminators $= 2$, 187,500 training iterations and optimised with Adam [156] (initial learning rate $\eta = 0.0001$, $\beta_1 = 0.5$, $\beta_2 = 0.999$).

3.3.3 Image Generation and Data Augmentation

Fake SAR images are synthesised using the visible images as the input of our transfer model, as discussed. This synthesis results in 3,000 generated SAR images, where

[2]This cleanup technique on the translation source images emphasises the content information that the domain transfer model training needs to extract, however, this thesis skips a quantitative discussion of the effect.

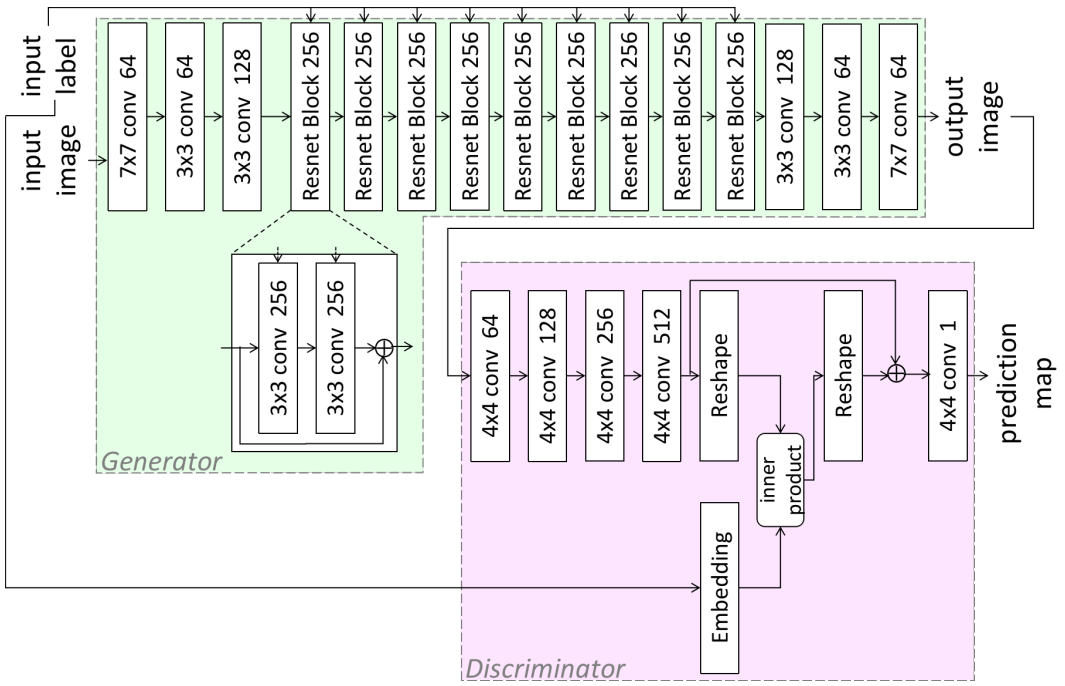
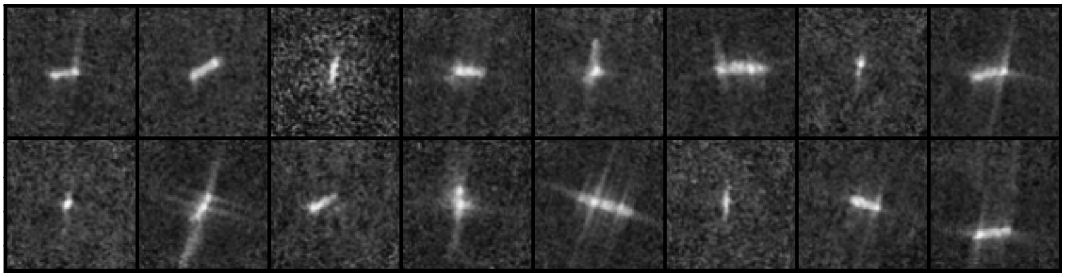


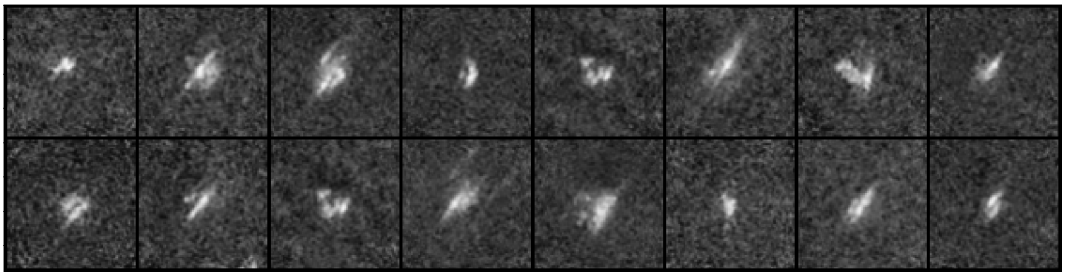
Figure 3.10: Our network architecture:- Conditional batch normalisation layers are applied to every convolutional layer within the generator whilst instance normalisation layers and spectral normalization are applied to every convolutional layer within the discriminator.

examples of these generated images are shown in Figure 3.11. Additionally, we plot the real SAR images and fake SAR images using t-SNE [29] (Figure 3.12) to show how the different distributions interrelate. This plot shows that the fake SAR images are well-distributed around the real SAR images.

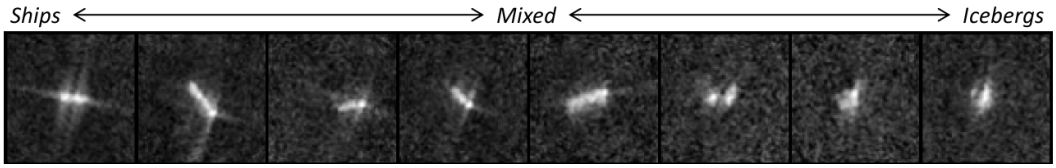
Also, we supplementary show the class-interpolated image outputs within the visible image domain in Figure 3.13. This result within the visible domain suggests that the trained network can produce semantically interpolated images also in the I2I translation from SAR to visible images and has a consistency between the bilateral I2I translation paths.



(a) Ships



(b) Icebergs



(c) Mixed

Figure 3.11: Examples of the generated SAR images (Train #1): (a) and (b) are the individual class images. (c) are the inter-class images sorted by the class labels from ship to iceberg.

3.3.4 Experiment on Object Classification Task

The evaluation of the classifier performance uses the simple Alexnet architecture [10][3], where the classifier performance is compared under the conditions in Table 3.2.

The classifiers are trained with the three training datasets, as denoted in Table 3.1, where the hyperparameters are optimised with the stochastic gradient descent [157] algorithm ($\eta = 0.02$, number of epochs $= 200$, $B = 512$). Performance is quantitatively assessed via the testing dataset also outlined in Table 3.1, using

[3]We choose the simple Alexnet classifier for the simple task but more advanced classifiers like Inception-v3 [140] should be selected for more complex tasks, such as multiple classification tasks. Our method can be applied to any classification task.

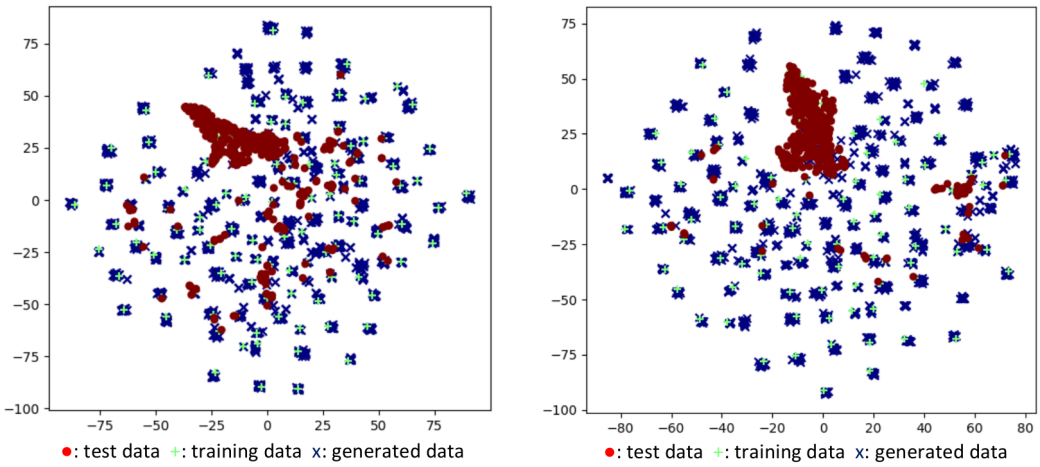


Figure 3.12: t-SNE [29] plot of ship (top) and iceberg (bottom) images from the test, training and generated datasets (Train #1).

Figure 3.13: Class-interpolated images within visible domain from our method.

statistical accuracy (A), precision (P), recall (R) and F1-score (F1) (Table 3.3), alongside the additional individual per-class classification performances for ships and icebergs, shown in the confusion matrices in Figure 3.14.

The overall results show that our proposed C2GMA data augmentation approach significantly outperforms the other approaches (BL, ROT, MIXUP [26], and MIXCG [130]). We find that generating new images using our approach increases training data appropriately, where the process of synthesising inter-class images is shown to provide significant improvements for the overall classification

Table 3.2: Compared augmented datasets on our object classification experiment. The MixCycleGAN model in this experiment (MIXCG) is trained with the same training dataset and parameters that our method uses.

BL	Only using the original training data [149]
ROT	BL + rotated 90, 180, and 270 degrees
MIXUP	Mixup ($\alpha = 0.2$) [26]
MIXCG	BL + MixCycleGAN [130] (α=0.2)
C2GMA (Ours)	BL + C2GMA (α=0.2, Section 3.2)

performance (C2GMA, Table 3.3).

Table 3.3: Overall classification results: accuracy (A), precision (P), recall (R), and F1-score (F1) on the common test set for each of training sets #1–3.

	Train #1			
	A	P	R	F1
BL	0.715	0.746	0.725	0.735
ROT	0.707	0.723	0.714	0.719
MIXUP	0.766	0.794	0.775	0.784
MIXCG	0.760	0.765	0.764	0.765
C2GMA (Ours)	**0.800**	**0.807**	**0.804**	**0.806**

	Train #2			
	A	P	R	F1
BL	0.469	0.469	0.500	0.484
ROT	0.469	0.469	0.500	0.484
MIXUP	0.690	0.728	0.701	0.714
MIXCG	0.757	0.783	0.766	0.776
C2GMA (Ours)	**0.771**	**0.795**	**0.779**	**0.787**

	Train #3			
	A	P	R	F1
BL	0.469	0.469	0.500	0.484
ROT	0.469	0.469	0.500	0.484
MIXUP	0.690	0.694	0.681	0.688
MIXCG	0.676	0.708	0.687	0.697
C2GMA (Ours)	**0.691**	**0.729**	**0.703**	**0.716**

	Average			
	A	P	R	F1
BL	0.551 ± 0.142	0.562 ± 0.160	0.575 ± 0.130	0.568 ± 0.145
ROT	0.549 ± 0.137	0.554 ± 0.146	0.571 ± 0.124	0.562 ± 0.135
MIXUP	0.715 ± 0.044	0.739 ± 0.051	0.719 ± 0.049	0.729 ± 0.050
MIXCG	0.730 ± 0.048	0.752 ± 0.039	0.739 ± 0.045	0.745 ± 0.042
C2GMA (Ours)	**0.754 ± 0.056**	**0.777 ± 0.042**	**0.762 ± 0.053**	**0.769 ± 0.047**

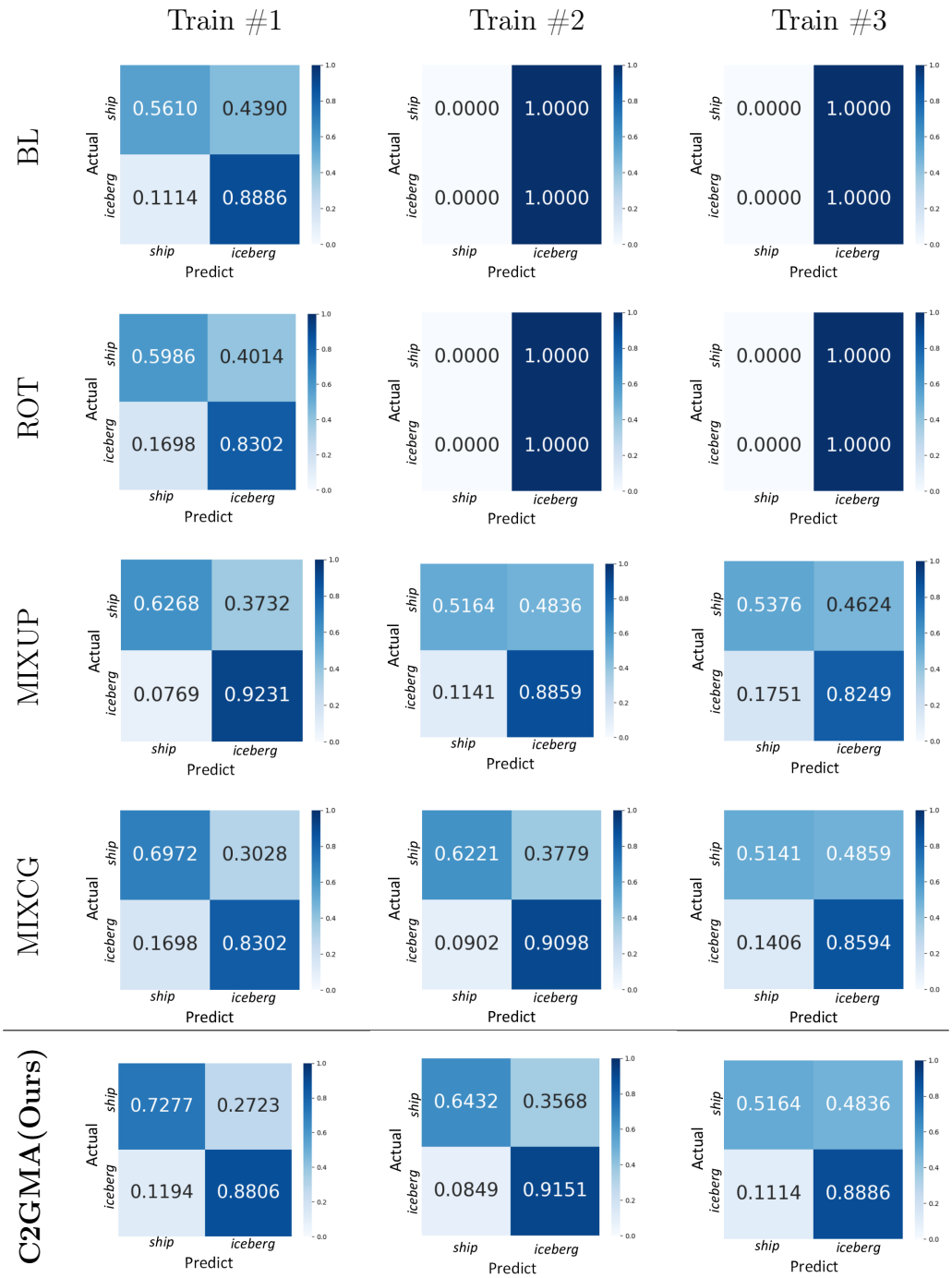


Figure 3.14: Per-class performance (confusion matrices) of our approach (C2GMA) against prior work in the field.

3.4 Summary

This chapter proposes and evaluates a CycleGAN and image fusion enabled data augmentation approach, Conditional CycleGAN Mixup Augmentation (C2GMA),

to address the challenge of effective data augmentation within cross-domain imagery where the availability of one of the domains is limited. In particular, we show that the generation of interpolated mixed class (non-visible domain) image examples via our novel C2GMA methodology leads to a significant improvement in the quality of non-visible domain classification tasks that suffer due to limited data availability and variety. Focusing on classification within the synthetic aperture radar domain, our approach is evaluated on a variation of the Statoil/C-CORE Iceberg Classifier Challenge dataset and achieves 75.4% accuracy, demonstrating a significant improvement when compared against traditional augmentation strategies. Future work will consider other different generative model backends, network architectures, and image feature fusion processes to enable generation of higher quality and variety of images for improved classification results and applications to other imaging domains.

CHAPTER 4

Diffusion-based Unpaired Image-to-image translation

This chapter discusses and evaluates a diffusion model based image-to-image (I2I) translation technique with the proposal of a novel unpaired I2I translation method that uses Denoising Diffusion Probabilistic Models (DDPM) [7] without requiring adversarial training. The proposed method, UNpaired Image Translation with Denoising Diffusion Probabilistic Models (UNIT-DDPM), trains a generative model to infer the joint distribution of images over both domains as a Markov chain by minimising a denoising score matching (DSM) [88] objective conditioned on the other domain. In particular, we update both domain translation models simultaneously, and we generate target domain images by a denoising Markov Chain Monte Carlo approach that is conditioned on the input source domain images, based on Langevin dynamics. Our approach provides stable model training for I2I translation and generates high-quality image outputs. This enables state-of-the-art Fréchet Inception Distance (FID) performance on several public datasets, including both colour and multispectral imagery, significantly outperforming the contemporary adversarial I2I translation methods such as Cycle-Consistent Generative Adversarial Networks (CycleGAN) [8]. The proposed approach enables the expansion of the mode of the translated images as the solution of Research Question 1 in Section 1.3.

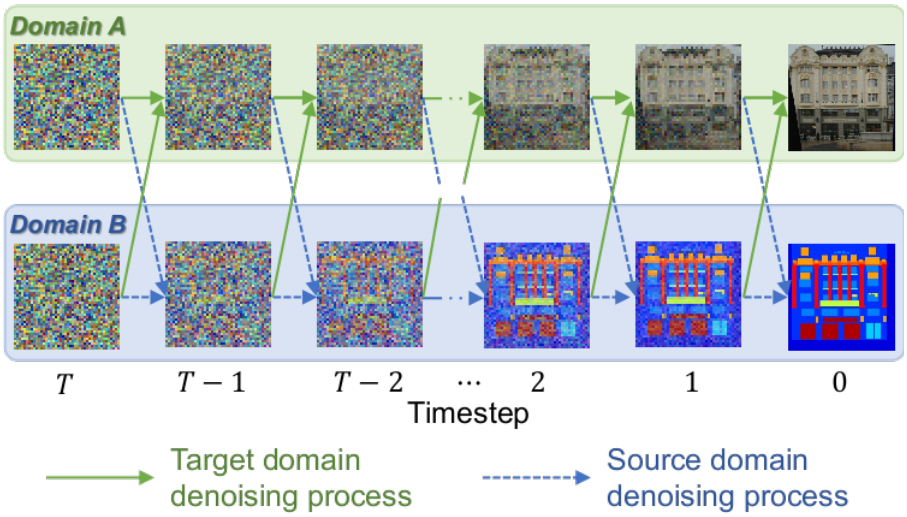


Figure 4.1: Conceptual illustration of our novel image-to-image translation approach using denoising diffusion probabilistic models.

4.1 Motivation

I2I translation fabricates images that are new but similar to target domain images from the information of images in another domain using the theory of generative models. Unpaired I2I translation eliminates the requirement of an aligned pair of source and target domain samples in training to enhance application potential as reviewed in Section 2.3. The current approaches of this unpaired method have generative model backends that mostly rely on adversarial training, namely Generative Adversarial Networks (GAN) [12]. Therefore, the unpaired I2I translation methods inherit the undesired natures of GAN (Section 2.2.1). To overcome this limitation, we apply a diffusion approach (Section 2.2.3) to unpaired I2I translation instead of GAN in order to make use of the advantages of diffusion models: wide mode coverage, high output quality and non-adversarial stable training.

4.2 Contributions

We propose a new I2I translation approach that uses DDPM as its backend, instead of GAN, in order to mitigate the limitation of unstable adversarial training and

improve the mode coverage and plausibility of generated images (Figure 4.1). The main contributions of this work are:

- *Dual-domain Markov Chain based Generative Model* – a Markov chain I2I translation approach is introduced that approximates the data distribution of both the source and target domains, such that they interrelate with each other (Section 4.3).

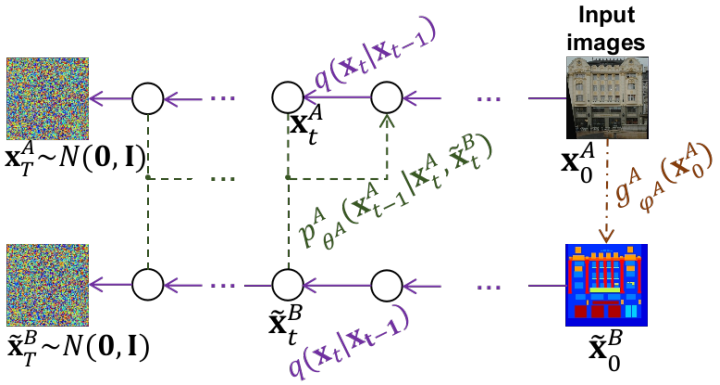
- *Stable Non-GAN-based I2I Translation Training* – the approach does not require adversarial training, however the model generates realistic outputs that capture high-frequency variations according to the perturbations of various levels of noise (Section 4.3.1).

- *Novel use of Markov Chain Monte Carlo Sampling* – the proposed sampling algorithm can be conditioned on unpaired source domain images to synthesise target domain images (Section 4.3.2).

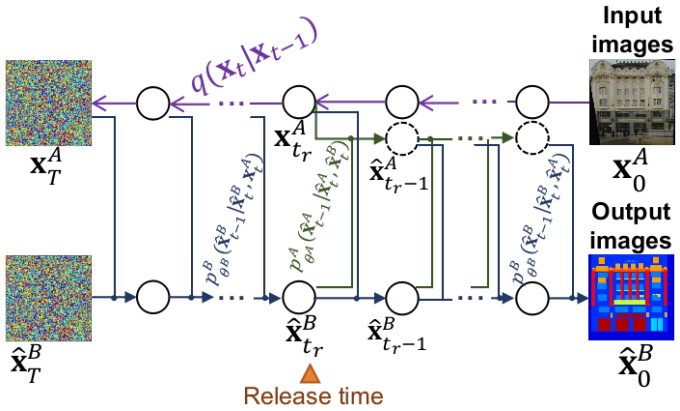
- *State-Of-The-Art Image-to-Image Translation* – the results are found to outperform the prior work of CycleGAN [8], Unsupervised Image-to-Image Translation Networks (UNIT) [120], Multimodal UNIT (MUNIT) [122], and Diverse Image-to-Image Translation via Disentangled Representations (DRIT++) [25] qualitatively and quantitatively for a number of varied benchmark datasets (Facade [158], Photos–Maps [8], Summer–Winter [8], and RGB–Thermal [32]) (Table 4.1 and Figure 4.4), as described in detail in Section 4.4.

4.3 Methodology

Our aim is to develop I2I translation between different domains of images whose distributions are formed as the joint probability of Equation (2.26) respectively. The method needs to learn the parameters of the models from a given dataset of source and target domains via empirical risk minimisation and subsequently be able to infer the target domain images from the corresponding source domain images.



(a) Model Training



(b) Image Translation (Inference)

Figure 4.2: The process of our method. (a) Model training: the reverse process $p^A_{\theta^A}$ is optimised using source domain images $\mathbf{x}_0^A$ and synthetic target domain images $\tilde{\mathbf{x}}_0^B$ created by the domain translation function $g^A_{\phi^A}$. (b) Image translation (inference): the trained model iteratively recovers the target domain images from noise with the condition of the source domain images. The conditional images are also re-generated by the reverse process from an intermediate timestep (*release time*).

4.3.1 Model Training

Assuming a source domain $\mathbf{x}_0^A \in \mathcal{X}^A$ and a target domain $\mathbf{x}_0^B \in \mathcal{X}^B$, we iteratively optimise the reverse process in each domain $p^A_{\theta^A}, p^B_{\theta^B}$ and the domain translation functions $\tilde{\mathbf{x}}_0^B = g^A_{\phi^A}(\mathbf{x}_0^A), \tilde{\mathbf{x}}_0^A = g^B_{\phi^B}(\mathbf{x}_0^B)$, which are only used in the model training to transfer the domain A to B and B to A respectively, via DSM (Figure 4.2 (a)). To enable translation between the source domain and target domain image pairs, $p^A_{\theta^A}(\mathbf{x}^A_{t-1}|\mathbf{x}^A_t), p^B_{\theta^B}(\mathbf{x}^B_{t-1}|\mathbf{x}^B_t)$ is modified as $p^A_{\theta^A}(\mathbf{x}^A_{t-1}|\mathbf{x}^A_t, \tilde{\mathbf{x}}^B_t), p^B_{\theta^B}(\mathbf{x}^B_{t-1}|\mathbf{x}^B_t, \tilde{\mathbf{x}}^A_t)$ such as to be conditional on the generated images. On the reverse process optimisation step, the model parameters θ^A, θ^B are updated to minimise the loss function based

Algorithm 1 UNIT-DDPM Training

repeat
  $\mathbf{x}_0^A \in \mathcal{X}^A, \mathbf{x}_0^B \in \mathcal{X}^B$
  $\tilde{\mathbf{x}}_0^A \leftarrow g_{\phi^B}^B(\mathbf{x}_0^B), \tilde{\mathbf{x}}_0^B \leftarrow g_{\phi^A}^A(\mathbf{x}_0^A)$
  $t^A, t^B \sim \text{Uniform}(\{1, ..., T\})$
  $\boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^B \sim \mathcal{N}(0, \text{I})$
  $\mathbf{x}_{t^A}^A \leftarrow \sqrt{\bar{\alpha}_{t^A}}\mathbf{x}_0^A + \sqrt{1-\bar{\alpha}_{t^A}}\boldsymbol{\epsilon}^A$, $\mathbf{x}_{t^B}^B \leftarrow \sqrt{\bar{\alpha}_{t^B}}\mathbf{x}_0^B + \sqrt{1-\bar{\alpha}_{t^B}}\boldsymbol{\epsilon}^B$
  $\tilde{\mathbf{x}}_{t^B}^A \leftarrow \sqrt{\bar{\alpha}_{t^B}}\tilde{\mathbf{x}}_0^A + \sqrt{1-\bar{\alpha}_{t^B}}\boldsymbol{\epsilon}^A$, $\tilde{\mathbf{x}}_{t^A}^B \leftarrow \sqrt{\bar{\alpha}_{t^A}}\tilde{\mathbf{x}}_0^B + \sqrt{1-\bar{\alpha}_{t^A}}\boldsymbol{\epsilon}^B$
  Take gradient descent step on
    $\nabla_{\theta^A,\theta^B}[\|\boldsymbol{\epsilon}^A - \epsilon_{\theta^A}^A(\mathbf{x}_{t^A}^A, \tilde{\mathbf{x}}_{t^A}^B, t^A)\|^2 + \|\boldsymbol{\epsilon}^B - \epsilon_{\theta^B}^B(\mathbf{x}_{t^B}^B, \tilde{\mathbf{x}}_{t^B}^A, t^B)\|^2]$
  Take gradient descent step on
    $\nabla_{\phi^A,\phi^B}[\|\boldsymbol{\epsilon}^A - \epsilon_{\theta^A}^A(\mathbf{x}_{t^A}^A, \tilde{\mathbf{x}}_{t^A}^B, t^A)\|^2 + \|\boldsymbol{\epsilon}^A - \epsilon_{\theta^A}^A(\tilde{\mathbf{x}}_{t^B}^A, \mathbf{x}_{t^B}^B, t^B)\|^2$
    $+\|\boldsymbol{\epsilon}^B - \epsilon_{\theta^B}^B(\mathbf{x}_{t^B}^B, \tilde{\mathbf{x}}_{t^B}^A, t^B)\|^2 + \|\boldsymbol{\epsilon}^B - \epsilon_{\theta^B}^B(\tilde{\mathbf{x}}_{t^A}^B, \mathbf{x}_{t^A}^A, t^A)\|^2$
    $+\lambda_{\text{cyc}}\|g_{\phi^B}^B(\tilde{\mathbf{x}}_0^B) - \mathbf{x}_0^A\|^2 + \lambda_{\text{cyc}}\|g_{\phi^A}^A(\tilde{\mathbf{x}}_0^A) - \mathbf{x}_0^B\|^2]$
until converged

on Equation (2.32), which is rewritten as:

$$\begin{aligned}\mathcal{L}_\theta(\theta^A, \theta^B) &= \mathbb{E}_{t,\mathbf{x}_0^A,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_{\theta^A}^A(\mathbf{x}_t(\mathbf{x}_0^A, \boldsymbol{\epsilon}), \tilde{\mathbf{x}}_t^B, t)\|^2] \\ &+ \mathbb{E}_{t,\mathbf{x}_0^B,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_{\theta^B}^B(\mathbf{x}_t(\mathbf{x}_0^B, \boldsymbol{\epsilon}), \tilde{\mathbf{x}}_t^A, t)\|^2], \end{aligned} \tag{4.1}$$

The parameters of the domain translation functions ϕ^A, ϕ^B are updated to minimise the DSM objective fixing θ^A, θ^B where:

$$\begin{aligned}\mathcal{L}_{\epsilon^\phi}(\phi^A, \phi^B) &= \mathbb{E}_{t,\mathbf{x}_0^B,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_{\theta^A}^A(\mathbf{x}_t(g_{\phi^B}^B(\mathbf{x}_0^B), \boldsymbol{\epsilon}), \mathbf{x}_t(\mathbf{x}_0^B, \boldsymbol{\epsilon}), t)\|^2 \\ &+ \|\boldsymbol{\epsilon} - \epsilon_{\theta^B}^B(\mathbf{x}_t(\mathbf{x}_0^B, \boldsymbol{\epsilon}), g_{\phi^B}^B(\mathbf{x}_t(\mathbf{x}_0^B), \boldsymbol{\epsilon}), t)\|^2] \\ &+ \mathbb{E}_{t,\mathbf{x}_0^A,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \epsilon_{\theta^B}^B(\mathbf{x}_t(g_{\phi^A}^A(\mathbf{x}_0^A), \boldsymbol{\epsilon}), \mathbf{x}_t(\mathbf{x}_0^A, \boldsymbol{\epsilon}), t)\|^2 \\ &+ \|\boldsymbol{\epsilon} - \epsilon_{\theta^A}^A(\mathbf{x}_t(\mathbf{x}_0^A, \boldsymbol{\epsilon}), g_{\phi^A}^A(\mathbf{x}_t(\mathbf{x}_0^A), \boldsymbol{\epsilon}), t)\|^2], \end{aligned} \tag{4.2}$$

In addition, the training is regularised by the cycle-consistency loss that is proposed in [8] to make both domain translation models bijective. The cycle-consistency loss, Equation (2.41) is rewritten as:

$$\mathcal{L}_{\text{cyc}^\phi}(\phi^A, \phi^B) = \mathbb{E}_{\mathbf{x}_0^B}[\|g_{\phi^A}^A(g_{\phi^B}^B(\mathbf{x}_0^B)) - \mathbf{x}_0^B\|_1] + \mathbb{E}_{\mathbf{x}_0^A}[\|g_{\phi^B}^B(g_{\phi^A}^A(\mathbf{x}_0^A)) - \mathbf{x}_0^A\|_1]. \tag{4.3}$$

The loss function is thus described as follows:

$$\mathcal{L}_{\phi}(\phi^A, \phi^B) = \mathcal{L}_{\epsilon^{\phi}}(\phi^A, \phi^B) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}^{\phi}}(\phi^A, \phi^B), \tag{4.4}$$

where λ_{cyc} is a cycle-consistency loss weight. The overall training process is presented in Algorithm 1.

4.3.2 Inference of Image Translation

Using the trained θ^A, θ^B in Section 4.3.1, the input images are translated from the source to the target domain. The domain translation functions are no longer used in inference. Instead, the target domain images are progressively synthesised from Gaussian noise and the noisy source domain images. During sampling, the generative process is conditioned on the input source domain images that are perturbed by the forward process from $t = 0$ until an arbitrary timetestep $t_r \in [1, T]$. This is then re-generated by the reverse process from this timestep, which we denote as the *release time* (Figure 4.2 (b)). The case of transferring from domain A $\mathbf{x}_0^A$ to domain B $\hat{\mathbf{x}}_0^B$ is described as:

$$\hat{\mathbf{x}}_{t-1}^B = \mu_{\theta^B}(\hat{\mathbf{x}}_t^B, \hat{\mathbf{x}}_t^A, t) + \Sigma_{\theta^B}(\mathbf{x}_t, t)\epsilon^B, \tag{4.5}$$

$$\hat{\mathbf{x}}_{t-1}^A = \begin{cases} \sqrt{\bar{\alpha}_{t^A}}\mathbf{x}_0^A + \sqrt{1-\bar{\alpha}_{t^A}}\boldsymbol{\epsilon}^A & (t > t_r) \\ \mu_{\theta^A}(\hat{\mathbf{x}}_t^A, \hat{\mathbf{x}}_t^B, t) + \Sigma_{\theta^A}(\mathbf{x}_t^A, t)\boldsymbol{\epsilon}^B & (t \leq t_r) \end{cases}, \tag{4.6}$$

$$\hat{\mathbf{x}}_T^B, \boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^B \sim \mathcal{N}(0, \mathrm{I}) \tag{4.7}$$

The overall translation (inference) process is presented in Algorithm 2.

4.4 Evaluation

Our method is evaluated against prior unpaired I2I translation methods [8] [120] [122] [25] on public datasets where ground truth input-output pairs are available [158] [8] [32]. We use Fréchet Inception Distance (FID) [33] to compare the performance because of the measuring capability of the quality and diversity of the output images along with the track record among previous

Algorithm 2 UNIT-DDPM Inference ($\mathcal{X}^A \to \mathcal{X}^B$)

$\mathbf{x}_0^A \in \mathcal{X}^A, \hat{\mathbf{x}}_T^B \sim \mathcal{N}(0, \mathrm{I})$
for $t = T, ..., t_r + 1$ **do**
  $\boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^B \sim \mathcal{N}(0, \mathrm{I})$
  $\hat{\mathbf{x}}_t^A = \sqrt{\bar{\alpha}_{t^A}}\mathbf{x}_0^A + \sqrt{1 - \bar{\alpha}_{t^A}}\boldsymbol{\epsilon}^A$
  $\hat{\mathbf{x}}_{t-1}^B = \frac{1}{\sqrt{1-\alpha_t}}(\hat{\mathbf{x}}_t^B - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\theta^B}(\hat{\mathbf{x}}_t^B, \hat{\mathbf{x}}_t^A, t)) + \sigma_t\boldsymbol{\epsilon}^B$
end for
for $t = t_r, ..., 1$ **do**
  $\boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^B \sim \mathcal{N}(0, \mathrm{I})$ if $t > 1$, else $\boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^B = 0$
  $\hat{\mathbf{x}}_{t-1}^A = \frac{1}{\sqrt{1-\alpha_t}}(\hat{\mathbf{x}}_t^A - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\theta^A}(\hat{\mathbf{x}}_t^A, \hat{\mathbf{x}}_t^B, t)) + \sigma_t\boldsymbol{\epsilon}^A$
  $\hat{\mathbf{x}}_{t-1}^B = \frac{1}{\sqrt{1-\alpha_t}}(\hat{\mathbf{x}}_t^B - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\theta^B}(\hat{\mathbf{x}}_t^B, \hat{\mathbf{x}}_t^A, t)) + \sigma_t\boldsymbol{\epsilon}^B$
end for
return $\hat{x}_0^B$

work [8] [120] [122] [25] as discussed in Section 2.6.2.

4.4.1 Baselines

The inferred output imagery from our proposed method is compared with that of CycleGAN [8], UNIT [120], MUNIT [122], and DRIT++ [25] both quantitatively (Tables 4.1) and qualitatively (Figures 4.4).

4.4.2 Datasets

We prepare the following datasets for the experiment. Every dataset includes two domains (here abbreviated to domain A and B) of images and is separated into training and test datasets. All images are resized 64 × 64 pixels in advance.

Facade The (A) photo and (B) semantic segmentation label images of buildings from the CMP Facades dataset [158]. 400 pairs are included for training and 106 pairs for test.

Photos–Maps The (A) photo and (B) map images were scraped from Google Maps [8]. 1,096 pairs are included for training and 1,098 pairs for test.

Summer–Winter The (A) summer and (B) winter Yosemite images were downloaded using Flickr API [8]. The dataset includes 1,231 summer and 962 winter

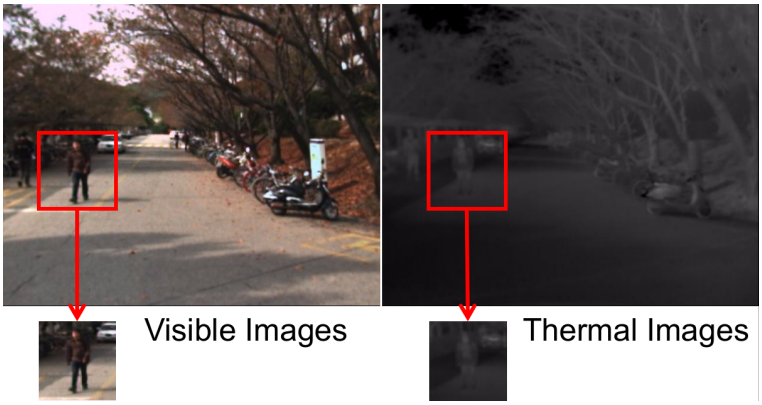


Figure 4.3: RGB–Thermal dataset cropped from the KAIST Multispectral Pedestrian Dataset [32].

images for training and 309 summer and 238 winter images for test.

RGB–Thermal The (A) visible and (B) thermal infrared images of pedestrians from the KAIST Multispectral Pedestrian Dataset [32]. This dataset contains aligned visible and thermal images in various regular traffic scenes. Since the images are annotated in the region of the pedestrians by bounding boxes, we crop 723 pairs of the pedestrian areas (more than 64 $\times$ 64 pixels size) from one scene (set00) for training and 425 pairs from another scene (set06) for test (Figure 4.3).

4.4.3 I2I Translation via UNIT-DDPM

The denoising models of our method are implemented using U-Net [117] based on PixelCNN [159] and Wide ResNet [160]. Transformer sinusoidal position embedding [112] is incorporated into the U-Net to encode the timestep, whose length is $T = 1000$. α_t is linearly decreased from $\alpha_1 = 0.9999$ to $\alpha_T = 0.98$. These configurations are same as original DDPM [7] but replaced Swith [161] with ReLU [162], group normalization [163] with batch normalization [152], and removed self-attention block to reduce the computation. The domain translation functions have ResNet [164] architecture which has the same layer depth as the U-Net.

In training, the pair of the training sample and the transferred sample (in another domain) are concatenated as the input. The model parameters are updated with $\lambda_{\text{cyc}} = 10.0$, batch size $B = 16$, 20,000 epochs via Adam [156] (initial learning

rate $\eta = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$).

We generate the images in both domains using the trained models and the test samples in each dataset with the *release time* $t_r = 1$.

4.4.4 Result

The output images synthesised by each method are shown in Figure 4.4 from which it is clearly apparent that our approach qualitatively generates more realistic images than CycleGAN [8], UNIT [120], MUNIT [122], and DRIT++ [25]. We also found our method does not suffer from mode collapse and the resultant model training was more stable due to not requiring adversarial training. In addition, Figure 4.5 shows the progressive sampling via our method over the course of the reverse process.

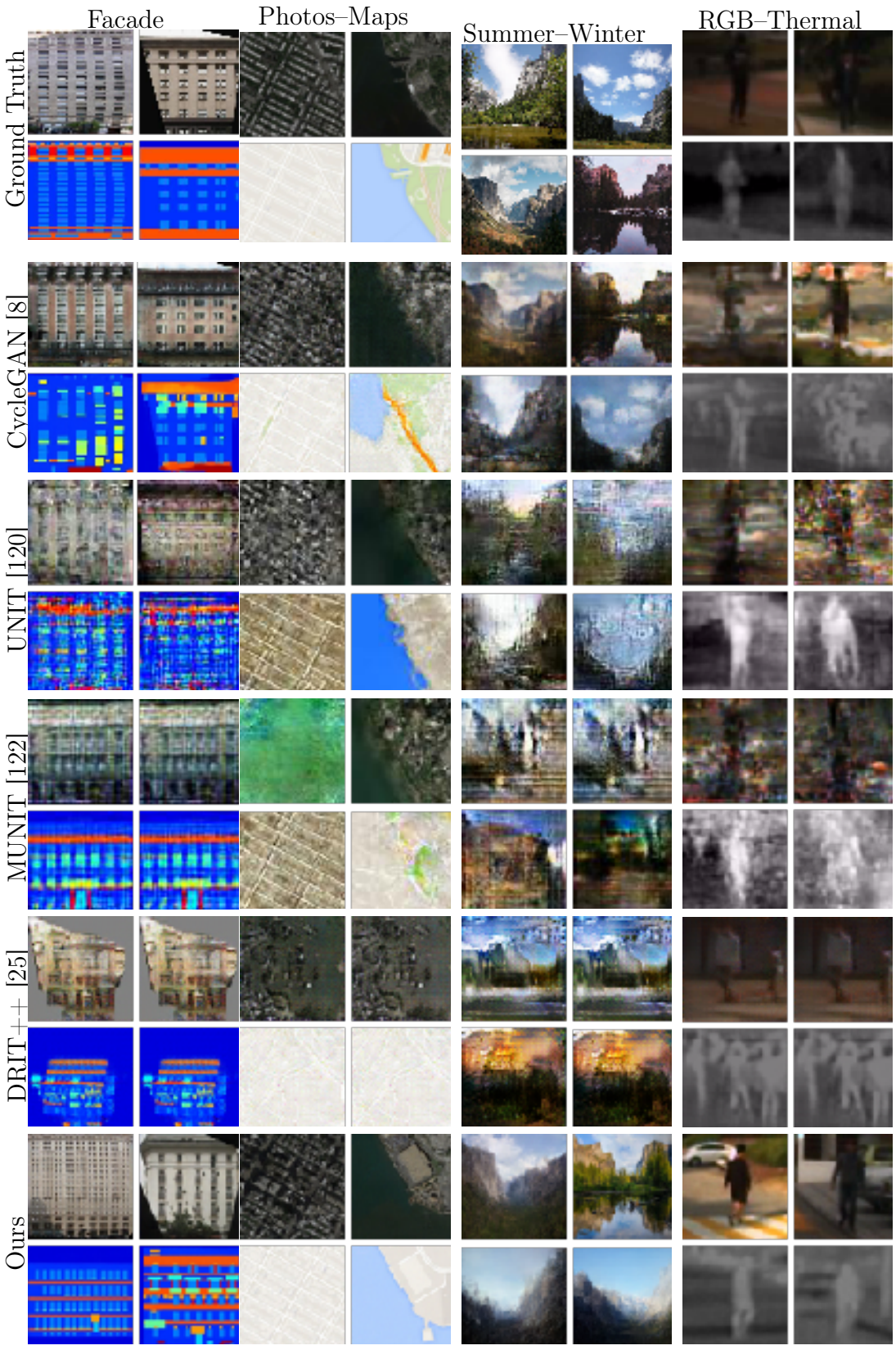


Figure 4.4: The examples of the output images generated by different image-to-image translation methods.

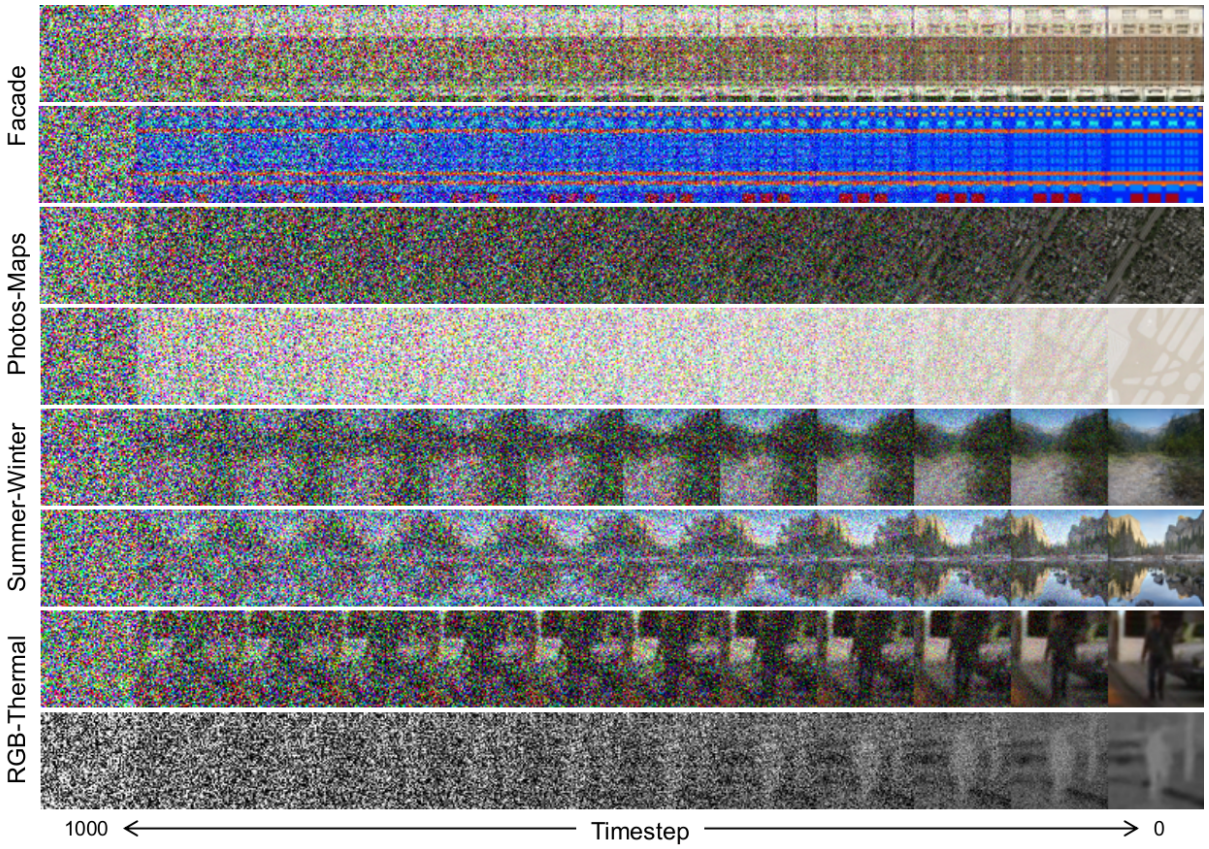


Figure 4.5: Examples of the progressive image generation via our method.

Comparison is conducted via FID between the ground truth and output images, and shown in Table 4.1[1]. Our method outperforms the contemporary approaches of CycleGAN [8], UNIT [120], MUNIT [122], and DRIT++ [25] over all of the benchmark datasets Facade, Photos–Maps, Summer–Winter, and RGB–Thermal offering a significant average increase in performance of $\sim$ 20% against previous approaches across all such datasets.

4.4.5 Ablation Study

We analyse the impact of the release time against the performance by changing from $t_r = 1$ to 900. The comparisons of the FID (Figure 4.6) show there are no significant changes. It can be thought that there are little differences between the

[1]FID score is hugely influenced by the size and number of the input images. This results in quite high values of FID, which are rarely observed in the literature.

Table 4.1: Fréchet Inception Distance (FID) [33] score on different image-to-image translation methods.

	CycleGAN [8]	UNIT [120]	MUNIT [122]	DRIT++ [25]	Ours
Facades					
B→A	232.12	239.58	335.72	336.76	**169.95**
A→B	265.70	216.27	244.78	317.23	**110.13**
Photos–Maps					
B→A	216.89	213.65	240.11	316.42	**193.06**
A→B	150.23	253.98	224.96	237.94	**116.23**
Summer–Winter					
B→A	121.18	202.88	221.11	261.82	**113.70**
A→B	133.16	161.10	205.33	268.45	**109.98**
RGB–Thermal					
B→A	338.30	286.90	305.24	284.21	**198.85**
A→B	169.38	213.08	233.92	226.35	**167.70**

input images and the model estimation. We observe some differences attributable to the release time variation, but this is dataset dependent. This result suggests that tuning the release time hyperparameter is dataset dependent, for which further analysis represents a direction for future work.

4.4.6 Limitations

We also observe the output images when the input image resolution is increased to 256 × 256 pixels. The higher resolution models are trained using the same network architecture and learning parameters as Section 4.4.3. The outputs shown in Figure 4.7 are wrongly coloured across the entire pixels. This suggests that the model fails to learn the global information of the images due to the increased complexity of the higher dimensional image space. One of the possible solutions is adding more layers along with an attention mechanism into the U-Net in the denoising models in order to capture a much more accurate multi-resolutional structure of images. Another possible solution is a single-image super-resolution (SISR) technique that could be applied to the output of our method. Such SISR techniques are realised by sparse coding [165], training CNNs [166], and also building SISR focused generative models [114] [94]. The trained SISR networks via such methods can increase the

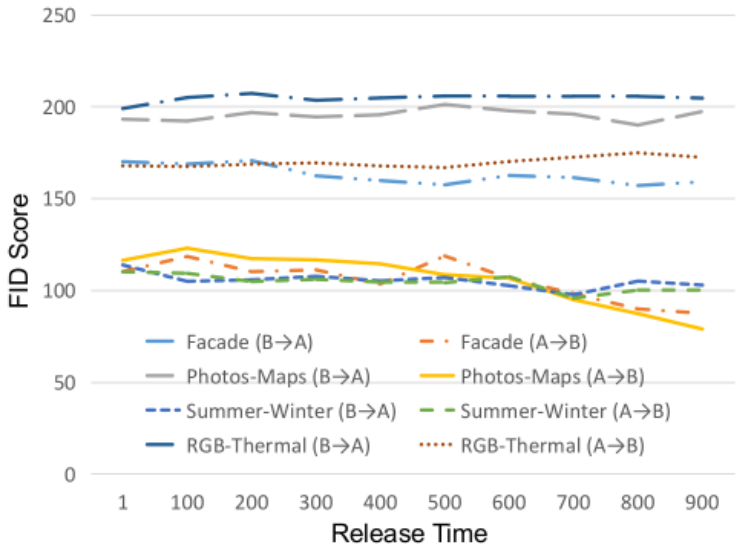


Figure 4.6: The comparison of FID by the release times.

image size of the output of the low resolution version of our method. Such solutions that can support the higher resolution sampling will be investigated in future work.

4.5 Summary

This chapter proposes a novel unpaired I2I translation method that uses DDPM without adversarial training, named UNpaired Image Translation with Denoising Diffusion Probabilistic Models (UNIT-DDPM). Our method trains a generative model to infer the joint distribution of images over both domains as a Markov chain by minimising a DSM objective conditioned on the other domain. Subsequently, the domain translation models are simultaneously updated to minimise this DSM objective. After jointly optimising these generative and translation models, we generate target domain images by a denoising MCMC approach, which is conditioned on the input source domain images, based on Langevin dynamics. Our approach provides stable model training for I2I translation and generates high-quality image outputs. The experimental validation of our approach provides state-of-the-art FID score performance on several public datasets including both colour and multispectral imagery, significantly outperforming the contemporary state-of-the-art I2I translation methods.

Figure 4.7: Examples of 256 × 256 output images generated using the model trained by our method (Facade dataset resized to 256 × 256 pixels).

Although the experiment shows compelling results, the current form of our method is far from universally effective in particular for higher resolution imagery. To address this issue, the implementation needs to be modified to model large images more accurately.

In addition, one remaining drawback of DDPM is the inference time for image generation. However, this can be accelerated by modifying the Markovian process such as denoising diffusion implicit models [90] or reducing the timesteps using a learnable Σ_θ [167].

Future work will consider modifications to enable shorter sampling times and higher quality image outputs, and the evaluation of the performance when the synthesised images are applied to other downstream computer vision tasks such as object classification.

CHAPTER 5

Efficient Diffusion-based Generative Model using Discrete Variables

This chapter considers a technique to enable lower training computation and a small number of training samples for diffusion based generative models. Whilst conventional Denoising Diffusion Probabilistic Models (DDPM) [7] can model fine image details even within small training datasets, their image synthesis capability is both resolution limited and computationally demanding due to the long Markov Chain dependency. As an alternative, Vector Quantised-Variational Autoencoders (VQ-VAE) [20] offer high-resolution modelling, but they subsequently fail to effectively capture such fine image detail. To overcome these issues, we propose a novel generative model that combines a Transformer-based autoregressive VQ-VAE with a smaller conditional DDPM [1]. The former (VQ-VAE) retains the advantages of capturing long-term dependencies and global structure in high-resolution imagery whilst the latter (conditional DDPM) is capable of modelling fine image textures and details. This proposed Discrete Conditional DDPM (DC-DDPM) architecture,

[1]Recent work on diffusion models such as Latent Diffusion Models [40] introduce another combination of VQ-GAN and DDPM providing further improvement. These are discussed in Chapter 6.

is subsequently shown to train much faster (5×) than competing DDPM models whilst, in contrast to competing adversarial methods, is additionally able to both compute per-sample likelihood estimates and remain competitive with state-of-the-art adversarial approaches within both large and small training datasets. The proposed approach enables the high mode and fidelity DDPM to gain data and computational efficiency as the solution of Research Question 2 in Section 1.3.

5.1 Motivation and Contributions

The training of DDPM is quite stable, featuring a simple denoising objective with noise schedules. This is in contrast to the training of Generative Adversarial Networks (GAN) [12], which require a generator and a discriminator components competing against each other, and commonly suffer from high sensitivity of hyperparameters, mode collapse, unstable training [69], and serious degradation in the case of training on small datasets [60]. Non-adversarial approaches including DDPM do not rotate through dataset modes in training in the manner of a GAN (i.e. where the generator and discriminator in a GAN cycle through a subset of the training dataset), and hence offer a strong advantage in terms of their reduced training dataset size requirements that thus enables broader applications (e.g. rare disease or anomaly image classes).

Regardless of their beneficial characteristics, DDPM do require many training iterations due to optimisation of their denoising network, which is conditioned by many steps of noise levels to construct the long Markov Chain, with just one randomly selected step of the noise level in each iteration. Moreover, the denoising network requires enough capacity to model each variational lower bound within many steps of the Markov Chain as a single function. This network requirement further forces the optimisation step to be tiny. These unavoidable traits associated with the long Markov Chain entail the high demand of training computation (e.g. experiments in the original DDPM [7] used 8 Google Cloud v3 TPUs, which are equivalent to 8 NVIDIA Tesla V100 GPUs) and limits scaling to high-resolution imagery. As an alternative, by way of avoiding both a long step-

by-step process or adversarial training, Variational Autoencoders (VAE) [57] offer different complimentary characteristics. VAE can be viewed as an autoencoder whose encoded latent is regularised to fit a tractable distribution which allows efficient sampling. This encoder-decoder architecture enables a lightweight non-iterative transformation from the latent to the image. However, VAE often fail to effectively model fine image detail due to the variational lower bound on the likelihood during training.

The VAE lower bound is optimised by the KL term and the posterior term (Equation (2.22)). This optimisation leads to the self-pruning of the hidden units of the encoder network. As a result, the decoder network is trained with poor latent vectors that do not represent enough abstraction of the input images and/or tend to just memorise a small portion of the mapping between the latent points and the image samples, without learning the distribution. This issue, called 'posterior collapse' [77], typically causes blurry outputs. As in Section 2.2.2, many approaches mitigate the posterior collapse; in particular, Vector Quantised Variational AutoEncoder (VQ-VAE) [20] significantly improves the outcomes. However, even these solutions have a limitation on providing sufficient information to the decoder within a large image generation task. To improve the decoder performance, the adversarial training of GANs can be applied to VQ-VAE (VQ-GAN) [23] but such an adversarial approach leads to unstable training and sensitiveness of hyperparameters, especially within a small training dataset, and requires much computation to converge the oscillating loss values.

To overcome these issues with stability, we uniquely use a smaller denoising network (DDPM) supported by a VAE and patch-based training. First, we redesign the network to be conditioned on the encoded VAE latent, effectively providing a 'hint' to the DDPM network. The well-compressed VAE latent acts as an informative guide to simplify the denoising function. Following the advantages of using a discrete latent representation for image synthesis [20] [23], we use the latents from VQ-VAE as the DDPM conditionals. Second, since the discrete latents are well modelled via a transformer-based autoregressive model [23], we utilise an attention-based training and sampling approach with a transformer architecture for the DDPM

conditionals. Transformers [112] have become the *de facto* standard within natural language modelling tasks [168] but have now also become popular in other domains such as audio [169] and vision [170] [171].

Recently, taming transformer [23] encode images as quantised discrete vectors (called 'codes') using an adversarially trained VQ-VAE (VQ-GAN) and a perceptual loss [107]. Images are generated by decoding from these codes whose structures are modelled via a transformer to capture the long-term interactions between them. We adopt such discrete code sequences modelled by a transformer for our conditional DDPM. We also adopt a patch-wise training approach for our model, facilitated by our proposed conditional strategy. Employing those strategies, our approach trains a VQ-VAE to obtain the discrete latent vectors, and then subsequently trains the DDPM using the patch-wise images with the relevant part of the discrete vectors as additional conditional information (Figure 5.1a). The model is sampled by inferring the discrete code sequence and synthesising images via the conditional DDPM MCMC process (Figure 5.1b). As a result, our approach, termed Discrete Conditional DDPM (DC-DDPM), realises a lightweight DDPM that is fast to train and generates high-quality outputs, and generalises well especially for datasets with limited training examples. The main contributions of this work are:

- we propose a novel approach that reduces the computation of DDPM via a complementary transformer-based autoregressive VQ-VAE (Section 5.2).
- the proposed lightweight DC-DDPM performs very well in cases of exceedingly limited training datasets (even with just 273 images from FFHQ [5] dataset) without any dataset augmentation applied (Section 5.3.3).
- our experiments validate that our non-adversarial approach enables image synthesis with faster training than the conventional DDPM [7] and improved Fréchet Inception Distance (FID) over other methods (StyleGANv2 [59], VQ-VAE2 [6], and DDPM [7]) on a limited training computation setting on the FFHQ dataset (Section 5.3.2).

5.2 Methodology

We describe our proposed DC-DDPM method, which is conceptually illustrated in Figure 5.1.

Introducing the function $\{C_i(\mathbf{x}_0)\}, i \in [0, L]$ dividing image $\mathbf{x}_0$ with L patches, our method approximates the image distribution $p(\mathbf{x}_0)$ as:

$$p(\mathbf{x}_0) \approx p(\mathbf{s}) \prod_{i=0}^{L-1} p(c_i(\mathbf{x}_0)|c_i(\mathbf{s})), \tag{5.1}$$

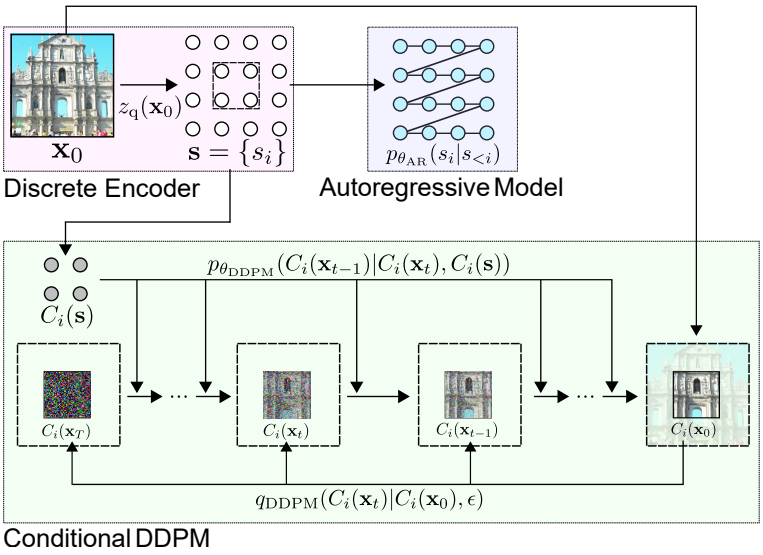
where $\mathbf{s}$ is a discrete latent of $\mathbf{x}_0$ that are coded by an encoder $\mathbf{s} = \phi(\mathbf{x}_0)$. We use a VQ-VAE for the encoder $\phi(\cdot)$, a DDPM for learning the distribution of $p(C_i(\mathbf{x}_0)|C_i(\mathbf{s}))$, and a transformer for $p(\mathbf{s})$ in Equation (5.1). Assuming the reduced dimensional subpatch of the image $C_i(\mathbf{x}_0)$ as an input of the DDPM, training of the DDPM can more efficiently prioritise finer and more local details. This 'local training' approach works effectively due to our modified DDPM which is conditioned by codes $\mathbf{s}$ sampled such as to respect their long-term interactions as well as positional trends. Such positional information acts as a guide to preserve the position of the patch in training and to recover an entire image in inference.

5.2.1 Training

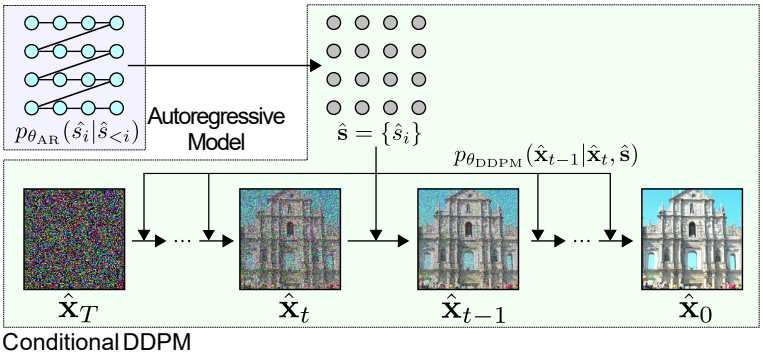
The overall training procedure is illustrated in Figure 5.1a. First, the VQ-VAE encoder $\phi(\cdot)$ is trained using Equation (2.25) to extract discrete codes from $\mathbf{x}_0$. Subsequently, the conditional form of DDPM, which accepts the patch of the codes $C_i(\mathbf{s})$ as the condition, is trained in order to learn $p(C_i(\mathbf{x}_0)|C_i(\mathbf{s}))$. The DDPM reverse process Equation (2.26) is therefore:

$$p_{\theta_{\text{DDPM}}}(C_i(\mathbf{x}_{0:T})|C_i(\mathbf{s})) := p(C_i(\mathbf{x}_T)) \prod_{t=1}^{T} p_{\theta_{\text{DDPM}}}(C_i(\mathbf{x}_{t-1})|C_i(\mathbf{x}_t), C_i(\mathbf{s})). \tag{5.2}$$

This DDPM process models the local coherence of the data depending on the subpatch of the discrete code. Since our conditional DDPM is trained using the patch of the image $C_i(\mathbf{x}_0)$ and conditioned by the codes $C_i(\mathbf{s})$, the model can be



(a) Training



(b) Sampling

Figure 5.1: Conceptual illustration of our proposed generative model. (a) Training: The input image $\mathbf{x}_0$ is coded to discrete vectors $\mathbf{s}$ by the VQ-VAE encoder z_{q}. $\mathbf{s}$ is modelled using a transformer. The subpart of the image $C_i(\mathbf{x}_0)$ is input to the DDPM with the relevant region of the codes $C_i(\mathbf{s})$ as a condition. (b) Sampling: The trained autoregressive transformer infers the discrete codes $\hat{\mathbf{s}}$. The DDPM generates data $\hat{\mathbf{x}}_0$ from Gaussian noise $\hat{\mathbf{x}}_T$ via MCMC sampling conditioned by the estimated $\hat{\mathbf{s}}$.

optimised much faster. The codes $\mathbf{s} = \{s_i\}$ are generated by an autoregressive model where:

$$p_{\theta_{\mathrm{AR}}}(\mathbf{s}) := \prod_{i=1}^{K} p_{\theta_{\mathrm{AR}}}(s_i|s_{<i}). \tag{5.3}$$

This autoregressive model is implemented with a transformer as with the approach in [23]. In contrast to the DDPM part (Equation (5.2)), this transformer models the long-term interactions of the codes $\mathbf{s}$ representing the overall structure of the entire image.

5.2.2 Sampling

The sampling procedure requires both the trained conditional DDPM model and the autoregressive transformer model (Figure 5.1b). First, the codes $\hat{\mathbf{s}} = \{\hat{s}_i\}$ are predicted step-by-step using multinomial sampling from the autoregressive model as:

$$\hat{s}_i \sim \mathrm{Multi}(\hat{s}_i^{c'}/Z_{\hat{s}_i^{c'}}), \tag{5.4}$$

$$\hat{s}_i^{c'} = \{\hat{s}_{ij}^{c'}\}, \quad \hat{s}_{ij}^{c'} = \begin{cases} \hat{s}_{ij}^{c} & (\hat{s}_{ij}^{c} \in \mathrm{top}_k(\hat{s}_i^{c})) \\ 0 & (\text{otherwise}) \end{cases}, \tag{5.5}$$

$$Z_{\hat{s}_i^{c'}} = \sum_j \hat{s}_{ij}^{c'}, \tag{5.6}$$

$$\hat{s}_i^{c} \sim p_{\theta_{\mathrm{AR}}}(\hat{s}_i^{c}|\hat{s}_{<i}), \tag{5.7}$$

where $\mathrm{Multi}(\cdot)$ is a multinomial distribution and $\mathrm{top}_k(\cdot)$ is a collection of top-k elements. In each step, we cut off the probabilities of the lower candidates by top-k selection [172] (Equation (5.5)) in order to avoid unintended sampling on low likelihoods.

Subsequently, the data $\hat{\mathbf{x}}_0$ is inferred using the estimated $\hat{\mathbf{s}}$ and the DDPM model as in Equation (5.8):

$$\hat{\mathbf{x}}_{t-1} = \mu_\theta(\hat{\mathbf{x}}_t, \hat{\mathbf{s}}, t) + \Sigma_\theta(\hat{\mathbf{x}}_t, \hat{\mathbf{s}}, t)\boldsymbol{\epsilon}, \tag{5.8}$$

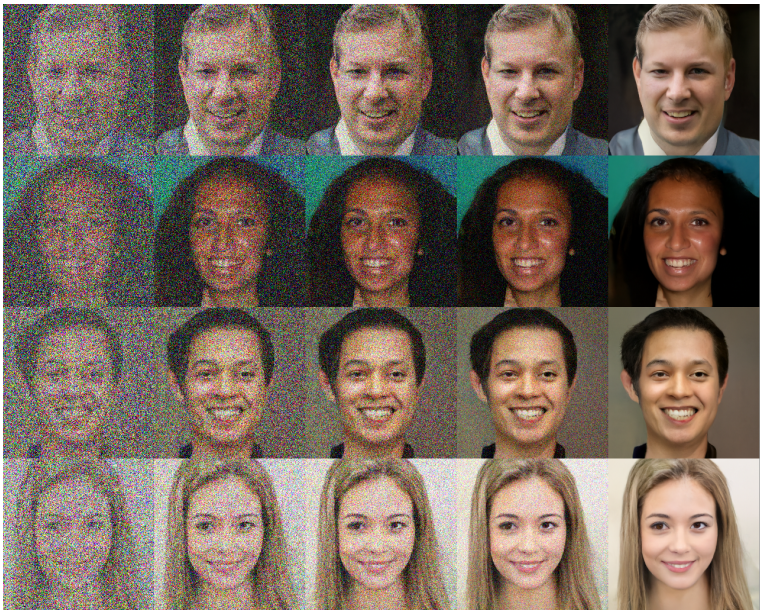


Figure 5.2: The reverse MCMC process gradually transforms from noise to images.

This sampling process gradually synthesises images from noise supported by the codes (Figure 5.2).

5.3 Evaluation

We evaluate the performance of our generative model within an image synthesis task comparing against three contemporary state-of-the-art approaches (VQ-VAE-2 [6], StyleGANv2 [59], DDPM [7]) on the challenging Flickr-Faces-HQ (FFHQ) [5] human faces dataset (70,000 images, 1024×1024 resolution) with image rescaling to 256×256 image resolution. We use Fréchet Inception Distance (FID) [33] to compare the performance because of the measuring capability of the quality and diversity of the output images along with the track record among previous work [6] [59] [7] as discussed in Section 2.6.2.

5.3.1 Implementation and Training

We train both our proposed model and the other comparison models without any use of transfer learning. To evaluate all methods on a same computation setting, all models are trained for a maximum duration of 15 GPU days on a single NVidia

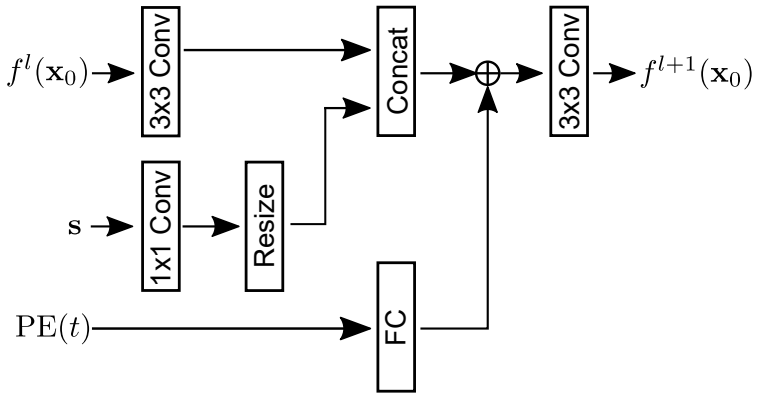


Figure 5.3: The diagram of our conditional ResNet block in the U-Net of the DDPM. $f^l(\mathbf{x}_0)$, $\mathbf{s}$, PE(t), $f^{l+1}(\mathbf{x}_0)$ are the output of the previous layer, the codes as a condition, the encoded timestep, and the output of the block, respectively. The codes are concatenated with the input in the middle of each ResNet block.

GeForce RTX 2080 Ti GPU.

Our Method

Our method is implemented as three components following the outline presented in Section 5.2: the VQ-VAE encoder, the autoregressive transformer and the conditional DDPM.

The **VQ-VAE encoder** is implemented following the original network [20] with the number of downsampling / upsampling layers is increased from 2 to 4, which compresses 256×256×3 dimensional images to a 16×16×256 dimensional encoding. The model parameters are updated with batch size $B = 256$ over 200 epochs via Adam [156] (initial learning rate $\eta = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$).

The 16×16×256 encodings, which are produced by the VQ-VAE encoder, are flattened to 256×256 and learnt by the **autoregressive transformer**. The number of the fully-connected layers in the transformer is 12 and the number of heads is 2. The model parameters are updated with $B = 64$, 1,000 epochs via Adam ($\eta = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$).

The **conditional DDPM** is implemented using U-Net [117] based on Wide ResNet [160] and a transformer sinusoidal position embedding [112] to encode the timestep (whose length is $T = 1000$, and α_t is decreased from $\alpha_1 = 0.9999$ to $\alpha_T =$

Table 5.1: The comparison of the average training time by one epoch.

	second / epoch
DDPM [7]	3514
DC-DDPM (ours)	**732**

0.98, as per the original DDPM [7]) but with Swith [161] replaced by ReLU [162], group normalization [163] with batch normalization [152], and the removal of the self-attention block to reduce computation. In addition, we adopt cosine scheduling [167] on the decreasing α instead of the original linear scheduling. To modify the U-Net to accept the condition, we concatenate the outputs of the first CNN layers of the ResNet blocks and the encodings (Figure 5.3). The model parameters are updated with $B = 128$ via Adam ($\eta = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$) within the remainder of the training schedule.

Within this experiment, we further define $c_i(\mathbf{x}_0)$ as a function that randomly crops a 64×64 pixels area from a 256×256 pixels image.

Other Methods

We use existing implementations to obtain the results of the other comparison methods [173] [174] [175] . All training parameters are set to the default values apart from the batch size, B, which is changed to run on the single GPU with training iterations to match that of our method. For VQ-VAE-2, we train the VQ-VAE model for 5 GPU days with $B = 128$ and subsequently the top-level and bottom-level PixelSnail are trained for 5 GPU days with $B = 8$ and $B = 8$, respectively. The StyleGANv2 model is trained with $B = 6$ and DDPM is trained with $B = 4$.

A comparison of the training speed of the original DDPM and our lightweight conditional DDPM is shown in Table 5.1, from which we can see that our method is 5× faster than the original DDPM.

5.3.2 Image Synthesis Results

We generate 10,000 samples at 256×256 image resolution from the trained model of our method following the sampling procedure (Section 5.2.2) and the same number of

Figure 5.4: The examples of output images from our DC-DDPM method trained on the FFHQ dataset [5].

Table 5.2: The comparison of Fréchet Inception Distance (FID) [33] score on a limited training computation setting: 15 days on a single GPU (NVidia Geforce RTX 2080 Ti).

	FID ↓
StyleGANv2 [59]	59.151
VQ-VAE-2 [6]	68.999
DDPM [7]	78.873
DC-DDPM (ours)	**52.574**

samples from each of the comparison methods. During the autoregressive sampling of the codes, we use top-k selection [172] in each step. The parameter k should be carefully considered because a too high k picks many inappropriate codes, while a too low k misses the diversity of the sampling. We empirically found that $k = 32$ gives a well balanced setting for our experiment. The examples of the generated images of our method are shown in Figure 5.4.

To measure quantitative performance, we calculate the FID between the training images, which are randomly selected 10,000 samples, and the generated images from each method using the reference FID implementation [176]. From Table 5.2 we can

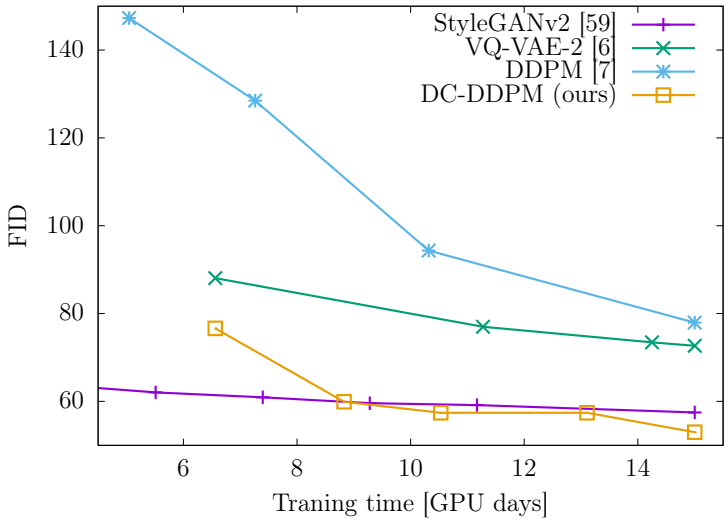


Figure 5.5: The comparison of Fréchet Inception Distance (FID) [33] scores of the generated images in Sec. 5.3.2 over different training time durations on the FFHQ dataset [5]. The time of VQ-VAE-2 is a sum of the training time of their 3 models and recorded by changing the training time of the 2 PixelSnail models equally. The time of our method is a sum of the training time of the 3 models and recorded by changing the training of the conditional DDPM.

observe that the FID score of the images from our method is lower than that from the comparison methods illustrating that method outperforms the others in terms of perceptual similarity between the sampled real and generated image sets. In addition, we also compare the FID scores of the generated images from each method at different stages of training (Figure 5.5) whereby we can see that the FID score of our method is improved more rapidly than the other non-adversarial methods and also gradually outperforms the state-of-the-art GAN based comparative approach.

5.3.3 Reduced Training Samples

In order to evaluate the performance obtained from a lower number of training samples, we respectively train each model with a randomly sampled subset of { 17,500 | 4,375 | 1,093 | 273[2] } images from the training dataset. The model training is run on an NVidia Tesla V100 GPU for 15 GPU days and we use the lowest FID checkpoints for each approach. We generate samples for the reduced training sample

[2]only for our proposed method.

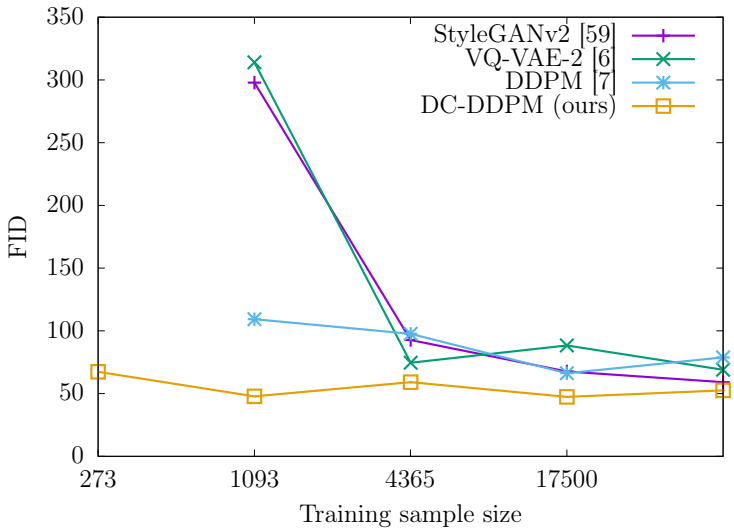


Figure 5.6: The comparison of Fréchet Inception Distance (FID) [33] scores of the generated images in Sec. 5.3.3 over different sizes of the training set (sample size) randomly selected from the FFHQ dataset [5].

models (shown in Figure 5.7) and compare the FID scores using the same approach as in Section 5.3.2. From Figure 5.6 we can observe that the DDPM-based methods perform well on smaller training dataset sizes whilst the performance of the other methods is significantly degraded. In particular, our method is less affected by such a small dataset condition. Subsequently, we extract the nearest neighbour samples of the generated images from the training dataset (Figure 5.8). This nearest neighbour experiment shows that the generated images (trained with exceedingly limited data and without augmentation) have a surprising lack of overfitting to the reduced FFHQ dataset.

5.3.4 Other Datasets

We also train our model using LSUN datasets [35], which has 10 scene categories and around 120,000 to 3,000,000 images in each category. We use the church and bedroom category pictures, which contain 126,227 and 303,125 samples[3], respectively. In addition, we use FLIR Starter Thermal Dataset [36], which contains 14,452 thermal images taken on street and highway scenes using Teledyne FLIR Tau

[3]20% of randomly chosen samples.

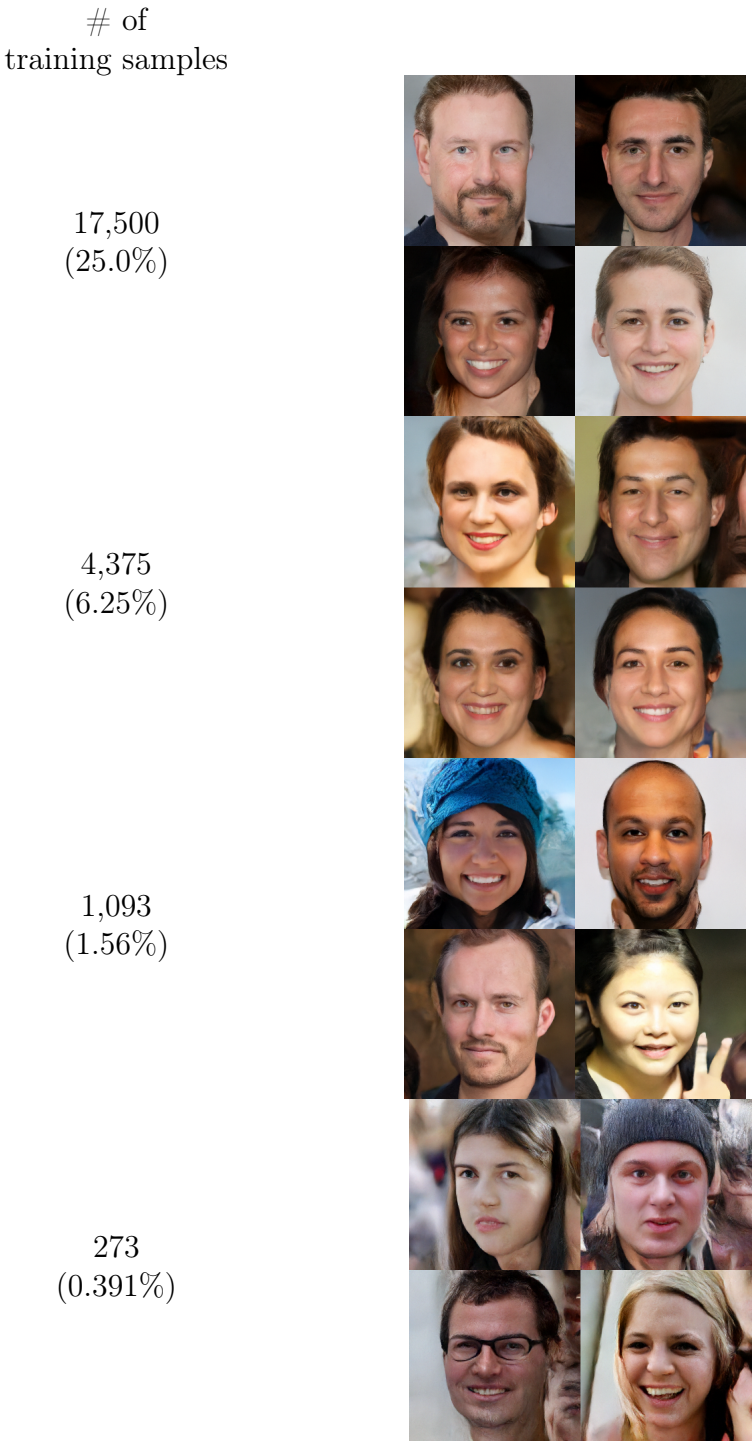


Figure 5.7: Images generated via our DC-DDPM method trained on different numbers of training examples.

Figure 5.8: The nearest neighbors (LPIPS [34] distance) for generated images from our model trained on only 273 randomly sampled images from FFHQ [5]. The leftmost column shows samples from our model and the other columns are the nearest neighbours within the training set (increasing in distance from left to right).

2 and Blackfly S thermal cameras, to build our model to generate thermal images for a evaluation of non-visible spectrum imagery. As another experiment for non-visible imagery, we use X-ray baggage images captured at Durham University using a Smith Detection dualenergy X-ray scanner (Dbf3 Dataset [37]), which consists of 7,603 security scan like images including criminal items such as firearms. The X-ray images are cropped the white spaces in advance. Those samples in the church, bedroom, thermal, and X-ray sets are randomly cropped to 256×256 pixels before use for training the models. A sample of the generated images from our model on each dataset are shown for further qualitative evaluation in Figure 5.9–5.11.

(a) LSUN Church

(b) LSUN Bedroom

Figure 5.9: Images generated via our method trained on the church and bedroom categories from LSUN [35].

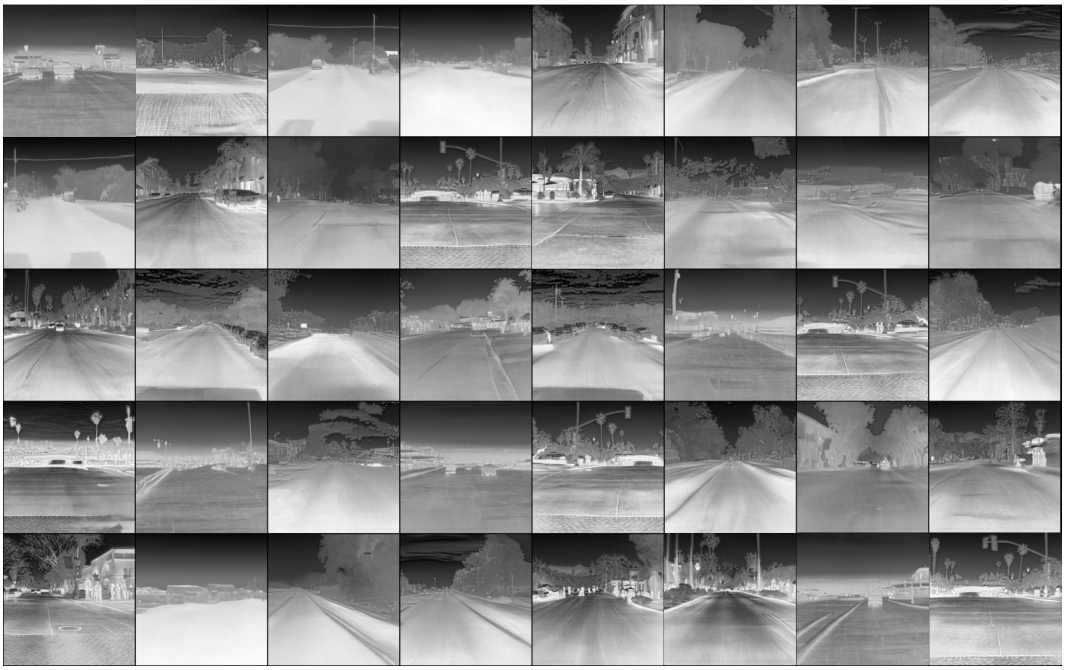


Figure 5.10: Images generated via our method trained on the thermal images from FLIR [36] dataset.

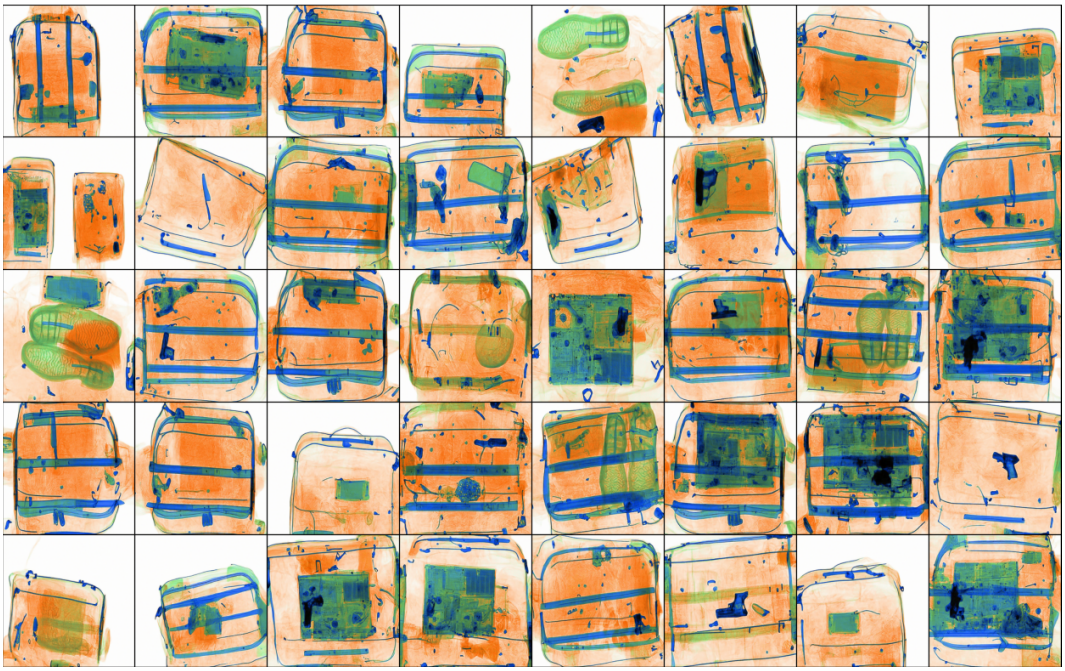


Figure 5.11: Images generated via our method trained on the X-ray baggage images from Dbf3 [37] dataset.

5.3.5 Limitations

Whilst our approach enables high-resolution image synthesis with fast training and a small dataset, the generated images exhibit unrealistic traits related to global image consistency. For example, we can see generated faces with differently coloured left/right eyes. Although a possibility within real world images, the natural occurrence of such phenomena is significantly lower than that of our generated examples (e.g. Figure 5.7). Such limitations are attributable to both the construction of the codebook that has insufficient information to distinguish the eyes and the use of patch-wise training which under-constrains long-term coherence in the patch-wise sequence, and hence global consistency, of the facial sub-regions. A possible solution is to learn a much informative codebook such as a hierarchical [6] or an adversarial approach [23].

Another limitation derives from the use of an autoregressive model to infer the codes. The current approach adopts a simple top-k multinomial sampler (Section 5.2.2) whose estimation is hugely reliant upon the unstable prediction in the first few iterations. This in turn degrades the performance of the code inference and hence the quality of the subsequent DDPM image synthesis. This code sampling can be improved via other recent work such as Gumbel-Softmax [177] [178] and discrete DDPM [179] [180] [111] which we will investigate to enable further high-quality image synthesis in future work.

5.4 Summary

We propose a novel generative model that combines both a transformer-based autoregressive VQ-VAE with a smaller conditional DDPM. The VQ-VAE retains the advantages of capturing long-term dependencies and global structure in high-resolution imagery whilst the conditional DDPM is capable of modelling finer textures and image details. This proposed DC-DDPM architecture is subsequently shown to train much faster than competing DDPM models whilst, in contrast to competing adversarial methods, is additionally able to both compute per-sample likelihood estimates and remain competitive with a state-of-the-art adversarial

approach within both a large and small training datasets. Notably, our proposed approach shows compelling results on exceedingly small training datasets without any dataset augmentation applied.

Future work will explore alternative methodologies for codebook generation within the proposed approach and look to achieve increasingly high-resolution image generation within similar training set and computational bounds.

CHAPTER 6

Conclusions

This chapter reviews key findings of our contributions (Chapters 3–5) and what extent the contributions address the research questions in Section 6.1, compares related concurrent work in Section 6.2 and describes a remaining challenge in Section 6.3. Considering these discussions, we describe the direction of a future work of this thesis in Section 6.4. Lastly, we make an overall conclusion in Section 6.6 after discussing social impact of this field of research in Section 6.5.

6.1 Review of Contributions

Overall from the research work carried out in this thesis, we highlight the following achievements:

- Domain translation and interpolation via image-to-image translation for augmenting a limited dataset (Chapter 3).
- Diffusion-based unpaired image-to-image translation without an unstable adversarial training (Chapter 4).
- Computation and dataset size efficient diffusion models using a patch-wise

training and discrete conditioning (Chapter 5).

Subsequently, we analyse the detailed descriptions of the contributions described in Chapter 3–5 from the two research questions (Section 1.3) individually.

Wide mode coverage of Deep Generative Model (DGM) sampled images Chapter 3 proposes an image generation technique using DGM-based Image-to-Image (I2I) translation and mixup, termed Conditional CycleGAN Mixup Augmentation (C2GMA). This C2GMA approach increases images in a desired domain by transferring images in another domain. To augment an object classification image dataset, which has an imbalance of the amount of classes, to improve the classification performance, the I2I translation model trained by our method synthesises semantic mixup images between two classes (Section 3.2). The experiments (Section 3.3) show the generated class mixed images contribute to improving classification performance. This result suggests that our C2GMA method can produce more variety of images in terms of the applications of object classification tasks. Meanwhile, further validation is needed that this image generation of our method is effective not only on object classification tasks but universally effective on other computer vision tasks. Also, this C2GMA method uses Generative Adversarial Models (GAN) to build the I2I translation model that inherits unstable adversarial training and the possibility of mode missing of output images.

Our another I2I translation method, named UNpaired Image Translation with Denoising Diffusion Probabilistic Models (UNIT-DDPM) (Chapter 4), adopts Denoising Diffusion Probabilistic Models (DDPM) for the backend of its model instead of GAN. DDPM theoretically produces higher quality and wider mode coverage of output images. This DDPM nature indirectly suggests that the generated images via our DDPM-based I2I translation approach can have wide mode coverage along with high quality, which is indicated in the experiments (Section 4.4). Meanwhile, further analysis is needed to directly investigate the mode coverage of the output of UNIT-DDPM, which may include experiments on downstream tasks.

High mode coverage, fidelity, and efficiency of DGM The approaches of C2GMA (Chapter 3) and UNIT-DDPM (Chapter 4) achieve the improvement only on small size images (75×75 and 64×64 pixels). These research are required further study to generate higher-resolution images with the consideration of computation and a number of training samples.

Our proposal of Discrete Conditional DDPM (DC-DDPM) (Chapter 5) reduces the training time of DDPM and enables training on small datasets (shown in Section 5.3). This achievement indicates that our DC-DDPM approach enables DDPM, which originally produces high mode coverage and quality output, to acquire training efficiency. Meanwhile, likewise UNIT-DDPM, experiments to directly see the mode coverage and the improvement within downstream tasks are needed.

6.2 Other Concurrent Work

Since the commencement of this research, much concurrent related work has emerged.

6.2.1 Stochastic Differential Equations for Diffusion Models

As in Section 2.2.3, DDPM uses many but finite timesteps of the transition process from noise to data for the data generation process. Stochastic Differential Equations (SDE) [181] generalise this DDPM process as a continuous time-dependent gradient field based on neural ordinary differential equations [182]. This continuous step model of SDE enables the flexible reverse diffusion process whereas the original DDPM is required to set the number of timesteps that has a trade-off of sampling quality and time. The ideal point of the timestep trade-off depends on datasets. Furthermore, SDE uses a predictor-corrector framework to correct errors in the discretised reverse process in sampling. As a result, SDE achieves a better Fréchet Inception Distance (FID) score than the score of a standard DDPM. Although SDE provides such a sophisticated and powerful upgrade to DDPM, this oversimplified diffusion limits generative model performance [38]. Therefore, Critically-damped Langevin Diffusion (CLD) [38] extends the variable space of

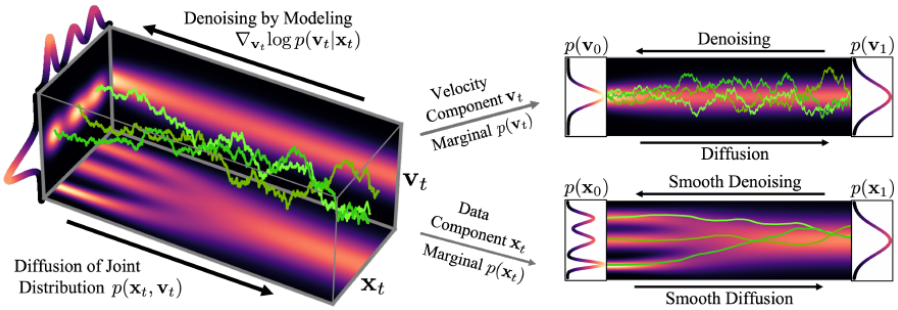


Figure 6.1: Visualisation of the Critically-damped Langevin Diffusion process (from [38]). Data $\mathbf{x}_t$ is augmented with a velocity $\mathbf{v}_t$. The diffusion process is coupled as a joint data-velocity space (probabilities in red). Noise is injected only into $\mathbf{v}_t$, which leads to smooth diffusion trajectories of $\mathbf{x}_t$ (green).

the diffusion process with consideration of velocities (Figure 6.1). This extension improves the performance of SDE.

6.2.2 Latent Diffusion Models

The original DDPM models data as a Markov chain (Equation (2.26)). This formulation causes two issues. The first one is that the reverse diffusion process on probabilistic variables $p_\theta(\mathbf{x}_{1:T})$ and the decoding process $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$ are implemented by the same U-Net network [7]. This unnatural implementation affects the efficiency of the model training. The second issue is that all latent variables $\mathbf{x}_{1:T}$ have the large dimension as same as the dimension of the input data $\mathbf{x}_0$. This training on large dimension variables requires huge computation. To address these issues, recent DDPM variants employ an encoder and a decoder of Variational Autoencoders (VAE) to convert input data to low dimensional latent codes and model the reverse diffusion process of this latent variables [111] [183] [40] [39]. ImageBART [111] uses discrete latents coded via GAN applied vector-quantised VAE (VQ-GAN) and applies diffusion modelling on the codes using a multinomial diffusion process [179]. Similarly, Unleashing Transfomer [183] models such VQ-GAN codes using absorbing state diffusion [180]. Meanwhile, Latent Diffusion Models (LDM) [40] support both continuous and discrete latent codes as employing KL-regularisation of VAE and VQ-regularisation of VQ-VAE. Latent Score-based Generative Model (LSGM) [39] applies such a latent variable approach to SDE (Figure 6.2). This latent SDE

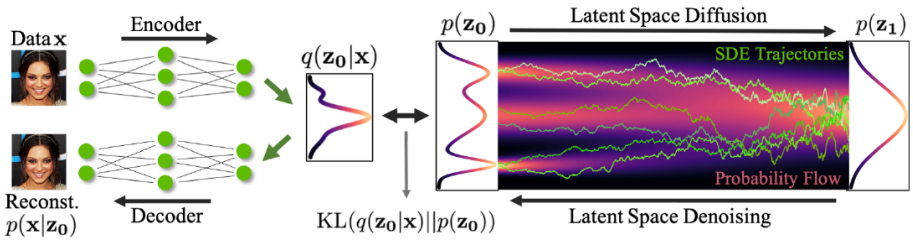


Figure 6.2: Conceptual illustrations of Latent Score-based Generative Model (from [39]). Data $\mathbf{x}_t$ is mapped to lower dimensional latent space via an encoder $q(\mathbf{z}_0|\mathbf{x})$. The diffusion process models the latent variables. The images are sampled via mapping from the latent to data space using a decoder $p(\mathbf{x}|\mathbf{z}_0)$ after synthesis of the latent variables via the reverse diffusion process.

method provides not only less computation in the smaller space of the diffusion model training but also the end-to-end simultaneous training of both VAE and SDE, which well optimises the models.

6.2.3 Controlling Reverse Diffusion Process

DDPM and its successors achieve high mode coverage and quality of image sampling. Recent advances in the analysis of diffusion models have brought techniques to control the image generation process [91] [184] [41]. Such controllable diffusion approaches enable the sampling process to be lead to target modes to be augmented. Along with the adoption of adaptive group normalisation, which is the application of the adaptive instance normalisation [5] and Feature-wise Linear Modulation (Film) [185] to the group normalisation layers in the U-Net of DDPM, a classifier of images is employed to incorporate class information [91] like Auxiliary Classifier GAN (ACGAN) [18]. This classifier-guidance approach trains a classifier $p_\phi(y|x_t, t)$, where y is the class labels, on noisy images x_t. The sampling process uses the gradient of the classifier $\nabla_{x_t} \log p_\phi(y|x_t, t)$ to guide the reverse diffusion to output the target class images. Whilst the classifier-guidance method requires an additional classifier model, Classifier-Free Diffusion Guidance [184] enables class-conditional DDPM without such a classifier by considering the difference between the scores of conditional and unconditional denoising. LDM proposes a conditional DDPM approach by inserting cross-attention layers in the U-Net [40] (Figure 6.3).

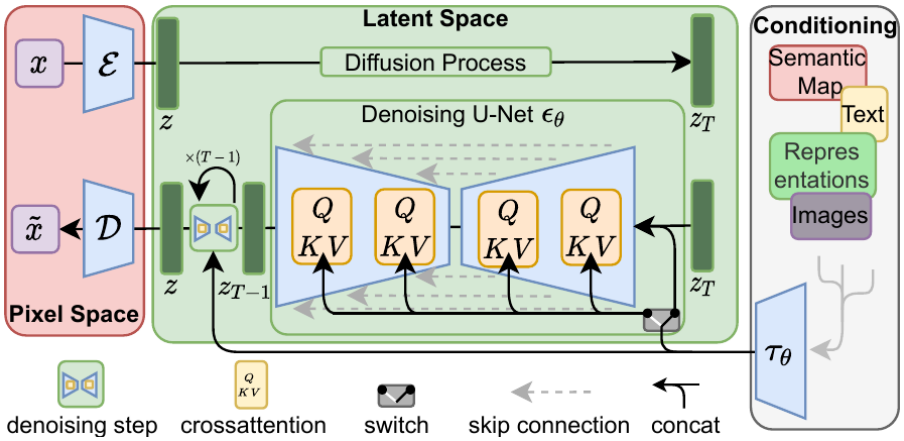


Figure 6.3: Conceptual illustrations of Latent Diffusion Model (from [40]). The reverse diffusion process is controlled by the cross-attention layers and the encoded condition inputs.

This cross-attention mechanism provides much flexible conditioning of the reverse diffusion that is not limited to classes but accepts any information such as text or bounding boxes. Another approach to greedy explore low-frequency modes of trained DDPM [41] employs the calculation of "hardness score", which is calculated based on the distance between a given image and the mean and variance of a given class. The reverse diffusion process supports the synthesis of low-density but plausible images by controlling the rate of the hardness scores of a specific class and all classes as well as the rate of the hardness score of real images and real plus synthesised images (Figure 6.4).

6.3 Open Challenges on Evaluation

As mentioned in Section 2.6, quantitative evaluation of generated images is quite difficult. The ultimate goal of image diversification is to synthesise images out of the distribution of a training dataset but inside a true distribution but it is difficult to conduct a completely accurate judgement whether the sampled images satisfy this criterion due to the unobservable true distribution.

Our contributions (Chapter 4–5) adopt FID score between pre-existing and generated samples, which is a common practice in a generative model research field, but it is not always an appropriate evaluation since it just measures the similarity

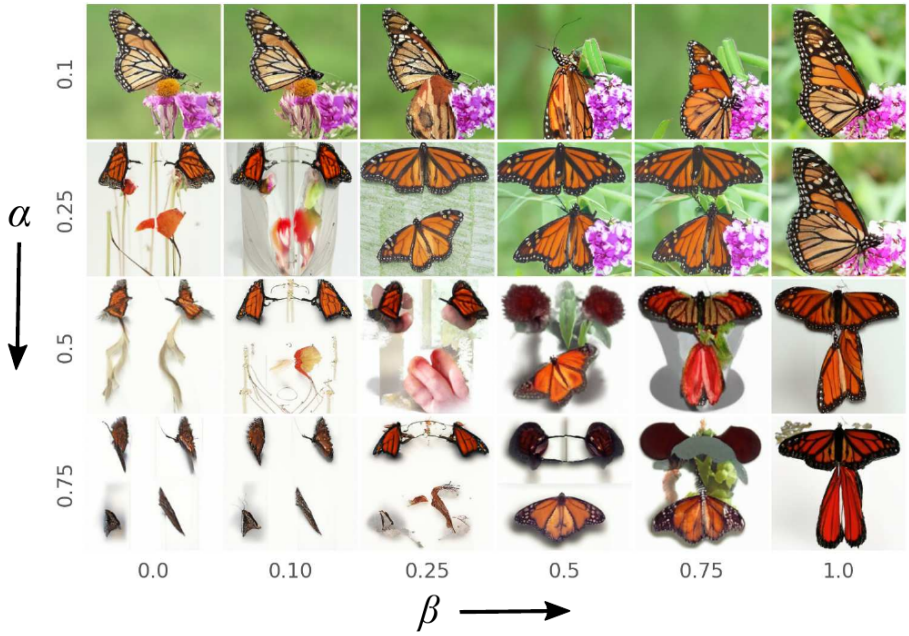


Figure 6.4: Samples from different settings of the hardness scores (from [41]). The setting has two parameters α and β. Decreasing α leads image synthesis to a low-density region. Increasing β leads sampling to a real image distribution.

between the two distributions. Moreover, there is a possible risk of low validity because FID uses the last layer in Inception-v3 classification network [140] trained on ImageNet [141]. This pre-trained network is optimised to discriminate the object classes of ImageNet and might not always capture important features of arbitrary images. Furthermore, the network is trained within visible spectrum imagery and might not fully support non-visible imagery. In order to mitigate these risks, the evaluation using self-supervised models such as 'Swapping Assignments between multiple Views of the same image' (SwAV) [186] can be considered as a solution [187]. However, this self-supervised model based evaluation requires pre-training on a large dataset which is difficult to apply to specific datasets like non-visible spectrum images, which tend to consists of few samples.

As an alternative, evaluation on downstream tasks like the experiment in Chapter 3 plays an important role for the quality assessment of generated images in practice, though, it does not provide a complete justification. This evaluation should be conducted on multiple tasks rather than a single task in order to strengthen the statement.

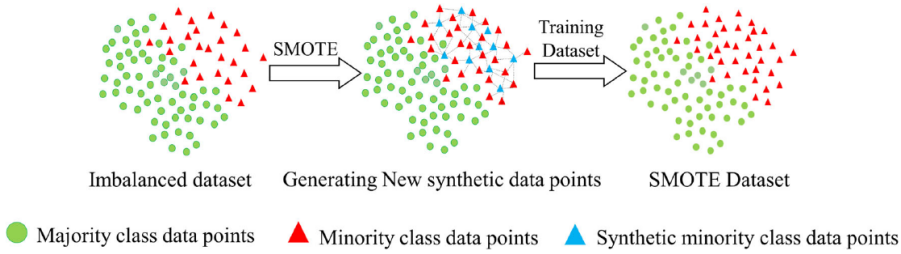


Figure 6.5: The schematic of SMOTE algorithm (from [42]). The samples in a minority class are enhanced by creating interpolated points between neighbour samples.

6.4 Direction of Future Work

Considering Section 6.1–6.3, our future research will be planned toward the following directions.

6.4.1 Large Image Generation of Diffusion-based I2I Translation

The current form of our diffusion-based I2I translation has a limitation of the output resolution (Section 4.4.6) regardless of its potential about the output image quality. Therefore, we will explore methods to overcome this limitation of the resolution. The encoded latent variable approaches (Section 6.2.2) could be a possible solution.

6.4.2 Investigating Other Feature Fusion Techniques

Whilst the contribution in this thesis (Chapter 3) adopts the mixup operation on the condition of class labels, various other fusion techniques [132] [134] [188] [189] [190] [191] can additionally be considered. As reviewed in Section 2.4.1, recent advanced manifold mixup families such as AlignMixup [132], OptTransMix, and AutoMix [134] provide more natural blending of images. Synthetic Minority Oversampling Technique (SMOTE) [188] produces artificial minority samples by interpolating between existing minority samples and their nearest minority neighbours (Figure 6.5). In contrast, Synthetic Majority Undersampling Technique (SMUTE) [189] creates an interpolated point between

two existing majority samples and deletes the two samples keeping the interpolated sample. Combined Synthetic Majority Oversampling and Undersampling Technique (CSMOUTE) [189] fuses SMOTE and SMUTE techniques. Sampling WIth the Majority (SWIM) technique [190] generates samples by inflating minority samples along the density contours of majority samples. MixBoost [191] integrates mixup with SMOTE and SWIM techniques. These feature blending methodologies may be able to expand the diverseness of our work.

6.4.3 Employing State-of-the-art Techniques for Improving Diffusion Models

Since the mechanism of DDPM was proposed, many techniques improving this diffusion approach have been devised. The log-likelihoods during the optimisation of the objective of DDPM are improved by importance sampling [167]. Whereas the network of a standard DDPM needs to be thoroughly trained on entire timesteps due to the imbalance of the loss scales across the timesteps, Soft-truncation [192] eases this training requirement by a dynamic configuration of the range of truncation of the timestep. This truncation approach results in better performances in their experiments. The learnable noise schedule and the Fourier features [193] are effective for the training of the diffusion training [194]. Whilst a standard DDPM uses a noise estimator for its denoising model, Dynamic Dual-Output DDPM [195] employs two models of a noise estimator and data estimator and dynamically combines them. The experimental result of this method is better than the performance of a standard diffusion approach. Subspace DDPM [43] proposes an approach to project a diffusion process to a subspace and its orthogonal space as well as a method to choose the subspace (Figure 6.6). The part of this orthogonal component diffusion is discarded from a specific timestep. This approach enables not only the improvement of the sample quality but also the reduction of the inference time. Our research may be improved by these DDPM improvement techniques as well as the advanced diffusion approaches in Section 6.2.

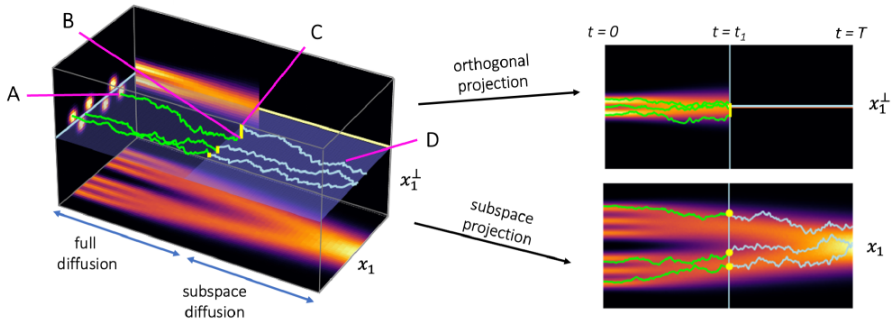


Figure 6.6: Visual schematic of subspace diffusion (from [43]). The diffusion process is projected to a subspace x_1 and its orthogonal space$x_1^\perp$. The orthogonal component is diffused until t_1 and discarded afterwards.

6.4.4 Evaluation on Multiple Downstream Tasks

In the future, we will apply the synthesised images via our proposed approach to multiple downstream tasks such as object classification, segmentation, and detection to investigate the effectiveness of the outcomes. This analysis should provide better evaluations on the contributions than just calculating FID.

6.5 Societal Impact Statement

Recent advances in DGM can have a positive impact both in research and application areas such as data augmentation, corrupted/missing data recovery, anomaly detection and data likelihood estimation which have established routes to broader societal impact.

Conversely, the malicious use of such generative models to generate fake images and videos for inappropriate use in both social and mainstream media outlooks can have negative societal impacts well beyond our research domain. Fortunately, such generated images still contain subtle flaws that readily facilitate expert detection [196] [197] such that inappropriate use can both be identified and counteracted.

Images synthesised via DGM can enhance image datasets, and some techniques can rebalance imbalanced training datasets. This possibly entails rebuilding datasets where some demographics are under-representated [198].

Whilst most DGM require significant computational training resources, putting strain on both energy and semiconductor material resources alike, our contribution (Chapter 5) instead works to address such issues by reducing the computational requirements of such models.

6.6 Conclusion Summary

To summarise this thesis, we draw the following primary conclusions in terms of the general topic, practical DGM-based image diversification, as:

- The diversification of image datasets is achievable using the combined use of unpaired I2I translation and image feature fusion. The proposed approach is valid for data augmentation on limited non-visible imagery based on the generation of inter-class interpolated images transferred from visible images.
- DDPM, which is a kind of DGM having the characteristics of wide mode coverage of the output samples, is applicable to I2I translation using our methodology. Also, the generated images via this diffusion-based I2I translation have greater quality than images generated by other previous I2I translation methods.
- Patch-wise and discrete conditional training of our proposed method enables the reduction of the computation and robustness when trained on small datasets.
- Future work will consider the adoption of state-of-the-art DDPM such as latent diffusion models and incorporate the advanced diffusion method into the backend of I2I translation.

Bibliography

[1] M. E. Kundegorski, S. Akçay, G. P. de La Garanderie, and T. P. Breckon, "Real-time classification of vehicles by type within infrared imagery," in *Proc. Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII*, 2016.

[2] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.

[3] S. Akçay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2203–2215, Sep. 2018.

[4] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[6] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Proc. Advances in Neural Information Processing Systems 32*, 2019.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Advances in Neural Information Processing Systems 33*, 2020.

[8] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2017.

[9] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *Proc. of the IEEE Chinese automation congress*, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems 25*, 2012.

[11] L. Gatys, A. Ecker, and M. Bethge, "A neural algorithm of artistic style," *Journal of Vision*, vol. 16, no. 12, pp. 326–326, 2016.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems 27*, 2014.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," vol. 86, pp. 2278–2324, Ieee, 1998.

[14] J. Susskind, A. Anderson, and G. E. Hinton, "The toronto face dataset," tech. rep., Technical Report UTML TR 2010-001, U. Toronto, 2010.

[15] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[16] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR abs/1411.1784*, 2014.

[17] Z. A. X. Y. L. L. B. S. Reed, Scott and H. Lee, "Generative adversarial text to image synthesis," in *Proc. 33rd Intl. Conf. on Machine Learning*, 2016.

[18] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Intl. Conf. on Machine Learning*, 2017.

[19] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proc. 5th Intl. Conf. on Learning Representations*, 2017.

[20] A. van den Oord, O. Vinyals, and k. kavukcuoglu, "Neural discrete representation learning," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[21] S. Deshpande, F. Minhas, S. Graham, and N. Rajpoot, "Safron: Stitching across the frontier network for generating colorectal cancer histology images," *Medical Image Analysis*, vol. 77, p. 102337, 2022.

[22] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the generative learning trilemma with denoising diffusion GANs," in *Proc. 10th Intl. Conf. on Learning Representations*, 2022.

[23] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2021.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[25] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. of the European Conf. on Computer Vision*, 2018.

[26] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[27] C. Beckham, S. Honari, V. Verma, A. M. Lamb, F. Ghadiri, R. D. Hjelm, Y. Bengio, and C. Pal, "On adversarial mixup resynthesis," in *Proc. Advances in neural information processing systems 32*, 2019.

[28] Y. Hong, L. Niu, J. Zhang, and L. Zhang, "MatchingGAN: Matching-based few-shot image generation," in *Proc. of the IEEE Intl Conf. on Multimedia and Expo*, 2020.

[29] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[30] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse Problems*, vol. 14, pp. R1 – R54, 1998.

[31] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[32] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baselines," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.

[33] 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 *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[35] 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," 2016.

[36] FLIR, "FLIR starter thermal dataset version1.3." `https://www.flir.co.uk/oem/adas/adas-dataset-form/`.

[37] N. Bhowmik, Q. Wang, Y. F. A. Gaus, M. Szarek, and T. P. Breckon, "The good, the bad and the ugly: Evaluating convolutional neural networks for prohibited item detection using real and synthetically composited x-ray imagery," in *Proc. British Machine Vision Conf. Workshops*, 2019.

[38] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped langevin diffusion," in *Proc. 10th Intl. Conf. on Learning Representations*, 2022.

[39] A. Vahdat, K. Kreis, and J. Kautz, "Score-based generative modeling in latent space," in *Proc. Advances in Neural Information Processing Systems 34*, 2021.

[40] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.

[41] V. Sehwag, C. Hazirbas, A. Gordo, F. Ozgenel, and C. Canton, "Generating high fidelity data from low-density regions using diffusion models," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.

[42] M. Aldraimli, D. Soria, J. Parkinson, E. Thomas, J. Bell, M. Dwek, and T. Chaussalet, "Machine learning prediction of susceptibility to visceral fat associated diseases," *Health and Technology*, vol. 10, pp. 925–944, 07 2020.

[43] B. Jing, G. Corso, R. Berlinghieri, and T. Jaakkola, "Subspace diffusion generative models," in *Proc of the European Conf. on Computer Vision*, 2022.

[44] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 18, no. 7, pp. 690–706, 1996.

[45] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[46] A. Vibhute and S. K. Bodhe, "Applications of image processing in agriculture: a survey," *Intl. Journal of Computer Applications*, vol. 52, no. 2, 2012.

[47] S.-H. Huang and Y.-C. Pan, "Automated visual inspection in the semiconductor industry: A survey," *Computers in industry*, vol. 66, pp. 1–10, 2015.

[48] C. F. Olson, "Maximum-likelihood template matching," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[49] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 1999.

[50] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[51] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[52] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[53] I. Goodfellow, Y. Bengio, and A. C. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Intl. Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[55] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36, 2018.

[56] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[57] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Advances in Neural Information Processing Systems 29*, 2016.

[58] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. of the 32nd Intl. Conf. on Machine Learning*, 2015.

[59] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.

[60] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient GAN training," in *Proc. Advances in Neural Information Processing Systems 33*, 2020.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Intl. Conf. on Learning Representations*, 2015.

[62] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[63] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," in *Proc. British Machine Vision Conf.*, 2017.

[64] P. T. G. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: Data augmentation via style randomization," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2019.

[65] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Intl. Conf. on Learning Representations*, 2016.

[66] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. 5th Intl. Conf. on Learning Representations*, 2017.

[67] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[68] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Neural Information Processing Systems 29*, 2016.

[69] I. Goodfellow, "NIPS 2016 Tutorial: Generative adversarial networks," *CoRR abs/1701.00160*, 2017.

[70] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," in *Proc. Advances in Neural Information Processing Systems 31*, 2018.

[71] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "AdaGAN: Boosting generative models," 2017.

[72] S. Pei, R. Y. D. Xu, S. Xiang, and G. Meng, "Alleviating mode collapse in GAN via diversity penalty module," *CoRR abs/2108.02353*, 2021.

[73] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[74] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Intl. Conf. on Machine Learning*, 2017.

[75] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[76] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. 31st Intl. Conf. on Machine Learning*, 2014.

[77] A. Razavi, A. van den Oord, B. Poole, and O. Vinyals, "Preventing posterior collapse with delta-VAEs," in *Proc. 7th Intl Conf. on Learning Representations*, 2019.

[78] A. G. ALIAS PARTH GOYAL, A. Sordoni, M.-A. Côté, N. R. Ke, and Y. Bengio, "Z-forcing: Training stochastic recurrent networks," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[79] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi, "Don't blame the ELBO! a linear VAE perspective on posterior collapse," in *Proc. Advances in Neural Information Processing Systems 32*, 2019.

[80] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *CoRR abs/1611.02648*, 2016.

[81] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, "Deep autoregressive networks," in *Proc. 31st Intl. Conf. on Machine Learning*, 2014.

[82] W. Joo, W. Lee, S. Park, and I.-C. Moon, "Dirichlet variational autoencoder," *Pattern Recognition*, vol. 107, p. 107514, 2020.

[83] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 161–173, 2001.

[84] E. Nalisnick and P. Smyth, "Stick-breaking variational autoencoders," in *Proc. 5th Intl Conf. on Learning Representations*, 2017.

[85] Y. Zhao, C. Li, P. Yu, J. Gao, and C. Chen, "Feature quantization improves gan training," in *Proc. 37th Intl. Conf. on Machine Learning*, 2020.

[86] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. 33rd Intl. Conf. on Machine Learning*, 2016.

[87] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. of Machine Learning Research 37*, 2015.

[88] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[89] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Advances in Neural Information Processing Systems 32*, 2019.

[90] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. 9th Intl Conf. on Learning Representations*, 2021.

[91] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Advances in Neural Information Processing Systems 34*, 2021.

[92] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *Proc. of the European Conf. on Computer Vision*, 2020.

[93] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2022.

[94] "Srdiff: Single image super-resolution with diffusion probabilistic models," *Neurocomputing*, vol. 479, pp. 47–59, 2022.

[95] H. Sasaki, C. G. Willcocks, and T. P. Breckon, "UNIT-DDPM: Unpaired image translation with denoising diffusion probabilistic models," *CoRR abs/2104.05358*, 2021.

[96] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[97] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[98] C. H. Lin, C.-C. Chang, Y.-S. Chen, D.-C. Juan, W. Wei, and H.-T. Chen, "COCO-GAN: Generation by parts via conditional coordinating," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2019.

[99] C. H. Lin, Y.-C. Cheng, H.-Y. Lee, S. Tulyakov, and M.-H. Yang, "InfinityGAN: Towards infinite-pixel image synthesis," in *Proc. 10th Intl. Conf. on Learning Representations*, 2022.

[100] C. Lu, R. Turner, Y. Li, and K. Nate, "Interpreting spatially infinite generative models," in *Proc. Intl. Conf. Machine Llearning Workshop on Human Interpretability in Machine Learning*, 2020.

[101] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2017.

[102] K. K. Singh, U. Ojha, and Y. J. Lee, "Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[103] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "PixelSNAIL: An improved autoregressive generative model," in *Proc. of 35th Intl. Conf. on Machine Learning*, 2018.

[104] A. Vahdat and J. Kautz, "NVAE: A deep hierarchical variational aautoencoder," in *Proc. Advances in Neural Information Processing Systems 33*, 2020.

[105] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[106] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[107] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Intl. Conf. on Machine Learning*, 2016.

[108] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "It takes (only) two: Adversarial generator-encoder networks," in *Proc. The Thirty-Second AAAI Conf. on Artificial Intelligence*, 2018.

[109] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," *CoRR abs/1706.04987*, 2017.

[110] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks," in *Proc. 5th Intl. Conf. on Learning Representations*, 2017.

[111] P. Esser, R. Rombach, A. Blattmann, and B. Ommer, "ImageBART: Bidirectional context with multinomial diffusion for autoregressive image synthesis," in *Proc. Advances in Neural Information Processing Systems 34*, 2021.

[112] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems 31*, 2017.

[113] Z. Dong, S. Kamata, and T. Breckon, "Infrared image colorization using S-shape network," in *Proc. Int. Conf. on Image Processing*, 2018.

[114] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[115] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2017.

[116] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[117] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[118] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[119] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: fine-grained image generation through asymmetric training," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2017.

[120] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[121] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems 29*, 2016.

[122] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. of the European Conf. on Computer Vision*, 2018.

[123] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, "Augmented CycleGAN: Learning many-to-many mappings from unpaired data," in *Proc. of the 35th Intl. Conf. on Machine Learning*, 2018.

[124] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio, "Image-to-image translation for cross-domain disentanglement," in *Proc. Advances in Neural Information Processing Systems 31*, 2018.

[125] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2019.

[126] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *CoRR abs/1708.04552*, 2017.

[127] A. Ramé, R. Sun, and M. Cord, "MixMo: Mixing multiple inputs for multiple outputs via deep subnetworks," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, 2021.

[128] M. Havasi, R. Jenatton, S. Fort, J. Z. Liu, J. Snoek, B. Lakshminarayanan, A. M. Dai, and D. Tran, "Training independent subnetworks for robust prediction," in *Proc. 8th Intl. Conf. on Learning Representations*, 2020.

[129] E. Schönfeld, B. Schiele, and A. Khoreva, "A U-Net based discriminator for generative adversarial networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.

[130] D. Liang, F. Yang, T. Zhang, and P. Yang, "Understanding mixup training methods," *IEEE Access*, vol. 6, pp. 58774–58783, 2018.

[131] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold Mixup: Better representations by interpolating hidden states," in *Proc. 36th Intl. Conf. on Machine Learning*, 2019.

[132] S. Venkataramanan, E. Kijak, L. Amsaleg, and Y. Avrithis, "AlignMixup: Improving representations by interpolating aligned features," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.

[133] M. Cuturi, "Sinkhorn Distances: Lightspeed computation of optimal transport," in *Proc. Advances in Neural Information Processing Systems 26*, 2013.

[134] J. Zhu, L. Shi, J. Yan, and H. Zha, "AutoMix: Mixup networks for sample interpolation via cooperative barycenter learning," in *Proc. of the European Conf. on Computer Vision*, 2020.

[135] S. Bartunov and D. Vetrov, "Few-shot generative modelling with generative matching networks," in *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, 2018.

[136] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Proc. Advances in neural information processing systems 29*, 2016.

[137] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR abs/1712.04621*, vol. abs/1712.04621, 2017.

[138] Amazon, "Amazon Mechanical Turk." `https://www.mturk.com`, August 2022.

[139] F. A. Schmidt, "The good, the bad and the ugly: Why crowdsourcing needs ethics," in *Proc. Intl. Conf. on Cloud and Green Computing*, 2013.

[140] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.

[141] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[142] S. Barratt and R. Sharma, "A note on the inception score," in *Proc. Intl. Conf. on Machine Learning Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[143] M. Fréchet, "Sur la distance de deux lois de probabilité," *Comptes Rendus Hebdomadaires des Seances de L Academie des Sciences*, vol. 244, no. 6, pp. 689–692, 1957.

[144] C. Nash, J. Menick, S. Dieleman, and P. Battaglia, "Generating images with sparse representations," in *Proc. 38th Intl. Conf. on Machine Learning*, 2021.

[145] M. J. Chong and D. Forsyth, "Effectively unbiased FID and inception score and where to find them," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.

[146] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," in *Proc. Advances in Neural Information Processing Systems*, 2018.

[147] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *Proc. of 37th Intl Conf. on Machine Learning*, 2020.

[148] Kaggle, "Kaggle." `https://www.kaggle.com`, August 2022.

[149] Statoil/C-CORE, "Statoil/C-CORE Iceberg Classifier Challenge." `https://www.kaggle.com/c/statoil-iceberg-classifier-challenge`.

[150] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[151] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Proc. Advances in Neural Information Processing Systems 30*, 2017.

[152] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Intl. Conf. on Machine Learning*, 2015.

[153] C. Chu, K. Minami, and K. Fukumizu, "Smoothness and stability in GANs," in *Proc. 8th Intl Conf. on Learning Representations*, 2020.

[154] EuropeanSpaceAgency, "Sentinel-1." `https://sentinel.esa.int/web/sentinel/missions/sentinel-1`.

[155] EuropeanComputerManufacturersAssociation, "ECMA-404 The JSON data interchange syntax 2nd edition." `https://www.ecma-international.org/publications-and-standards/standards/ecma-404/`, December 2017.

[156] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Intl. Conf. on Learning Representations*, 2015.

[157] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Advances in Neural Information Processing Systems 20*, 2007.

[158] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *Proc. German Conf. on Pattern Recognition*, 2013.

[159] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: A PixelCNN implementation with discretized logistic mixture likelihood and other modifications," in *Proc. 5th Intl Conf. on Learning Representations*, 2017.

[160] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. British Machine Vision Conf.*, 2016.

[161] P. Ramachandran, B. Zoph, and Q. Le, "Searching for activation functions," in *Workshop Track Proc. 6th Intl Conf. on Learning Representations*, 2018.

[162] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Intl. Conf. on Machine Learning*, 2010.

[163] Y. Wu and K. He, "Group Normalization," in *Proc. of the European Conf. on Computer Vision*, 2018.

[164] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[165] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang, "Convolutional sparse coding for image super-resolution," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2015.

[166] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[167] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. 38th Intl. Conf. on Machine Learning*, 2021.

[168] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training." `https://openai.com/blog/language-unsupervised/`, 2018.

[169] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *CoRR*, vol. abs/1904.10509, 2019.

[170] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proc. 37th Intl. Conf. on Machine Learning*, 2020.

[171] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[172] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[173] K. Seonghyeon, "vq-vae-2-pytorch." `https://github.com/rosinality/vq-vae-2-pytorch/`.

[174] P. Wang, "Simple stylegan2 for pytorch." `https://github.com/lucidrains/stylegan2-pytorch/`.

[175] P. Wang, "Denoising diffusion probabilistic model, in pytorch." `https://github.com/lucidrains/denoising-diffusion-pytorch/`.

[176] M. Seitzer, "pytorch-fid: FID Score for PyTorch." `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.2.1.

[177] Y. W. T. Chris J. Maddison, Andriy Mnih, "The concrete distribution: A continuous relaxation of discrete random variables," in *Proc. 5th Intl Conf. on Learning Representations*, 2017.

[178] B. P. Eric Jang, Shixiang Gu, "Categorical reparameterization with gumbel-softmax," in *Proc. 5th Intl Conf. on Learning Representations*, 2017.

[179] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, "Argmax Flows: Learning categorical distributions with normalizing flows," in *3rd Symposium on Advances in Approximate Bayesian Inference*, 2021.

[180] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, "Structured denoising diffusion models in discrete state-spaces," in *Proc. Advances in Neural Information Processing Systems 34*, 2021.

[181] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[182] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Advances in Neural Information Processing Systems 31*, 2018.

[183] S. Bond-Taylor, P. Hessey, H. Sasaki, T. P. Breckon, and C. G. Willcocks, "Unleashing Transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes," in *Proc. of the European Conf. on Computer Vision*, 2022.

[184] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[185] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. The Thirty-Second AAAI Conf. on Artificial Intelligence*, 2018.

[186] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Advances in Neural Information Processing Systems 33*, 2020.

[187] S. Morozov, A. Voynov, and A. Babenko, "On self-supervised image representations for GAN evaluation," in *Proc. 9th Intl. Conf. on Learning Representations*, 2021.

[188] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[189] M. Koziarski, "CSMOUTE: Combined synthetic oversampling and undersampling technique for imbalanced data classification," in *Proc. Intl. Joint Conf. on Neural Networks*, 2021.

[190] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *Proc. of the IEEE Intl. Conf. on Data Mining*, 2018.

[191] A. Kabra, A. Chopra, N. Puri, P. Badjatiya, S. Verma, P. Gupta, and B. Krishnamurthy, "MixBoost: Synthetic oversampling using boosted mixup for handling extreme imbalance," in *IEEE Intl. Conf. on Data Mining*, 2020.

[192] D. Kim, S. Shin, K. Song, W. Kang, and I.-C. Moon, "Soft Truncation: A universal training technique of score-based diffusion model for high precision score estimation," in *Proc. 39th Intl. Conf. on Machine Learning*, 2022.

[193] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. Advances in Neural Information Processing Systems 33*, 2020.

[194] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. Advances in Neural Information Processing Systems 34*, 2021.

[195] Y. Benny and L. Wolf, "Dynamic dual-output diffusion models," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.

[196] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.

[197] H. Guo, S. Hu, X. Wang, M.-C. Chang, and S. Lyu, "Eyes Tell All: Irregular pupil shapes reveal GAN-generated faces," in *Proc. the IEEE Intl Conf. on Acoustics, Speech and Signal Processing*, 2022.

[198] J. Vincent, "What a machine learning tool that turns Obama white can (and can't) tell us about AI bias." `https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-pulse-stylegan-obama-bias`, January 2020.