Imperial College London
Department of Computing

Adversarial Learning for Image-to-Image Generative Creativity

Simiao Yu

October 2018

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

Abstract

Achieving generative creativity in the context of visual data, i.e. the generation of novel and valuable images, is a long-standing goal in computer vision and artificial intelligence. Generative adversarial networks (GANs) are prominent deep generative models that can successfully generate visually-appealing images. However, the generated images are mostly simple memorisation or imitation of training samples, which exhibits limited generative creativity. To obtain higher-degree generative creativity, we focus on more challenging image-to-image generation tasks, in which the generated images are not only more practically valuable, but also more distinct from existing data.

The challenges of achieving image-to-image generative creativity lie in three aspects: whether the generated images 1) are truly useful, especially for critical applications (e.g. in the field of medical imaging), and 2) can demonstrate a clear difference from training samples, and 3) are varied and diverse for one input image, which is a natural requirement for many image generation tasks.

In this thesis, we aim to develop deep conditional adversarial networks for challenging image-to-image generation tasks, each of which respectively exhibits one type of image-to-image generative creativity. We make the following contributions. First, we propose EnrichGAN for fast compressed sensing magnetic resonance imaging (CS-MRI) reconstruction that exhibits enrichment creativity. We demonstrate that EnrichGAN qualitatively and quantitatively outperforms various conventional and state-of-the-art methods, with a much faster processing time that enables real-time applications. Second, we propose SimGAN for semantic image manipulation. It requires learning good mappings

between visual and text features. We show that SimGAN achieves superior results on this challenging image-to-image generation task that demonstrates high-level transformative creativity. Finally, we propose DesignGAN for automating the process of shape-oriented bionic design. It requires learning to combine features of images from different domains, in an unsupervised fashion. We demonstrate that DesignGAN learns to achieve image-to-image combinatorial creativity.

Copyright Declaration

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

Declaration of Originality

I hereby declare that this thesis is composed by myself, and the work it presents is my own, except where otherwise stated. For collaborative research, my individual contributions are summarised at the end of the corresponding chapter. All use of the previously published work of others has been listed in the bibliography.

Acknowledgements

I would like to express my sincerest gratitude to the following people.

- My supervisor, Professor Yike Guo, for his continuous support and guidance throughout my PhD study. He has supervised me with constant inspiration, immense knowledge, constructive criticism, great patience and endless enthusiasm, which I will always appreciate.

- My examiners, Professor William Knottenbelt from Imperial College London and Professor Mahesan Niranjan from University of Southampton, for their insightful comments and valuable feedback that have helped improve my thesis.

- My collaborators, including Mr Hao Dong, Dr Fangde Liu, Dr Akara Supratak, Ms Pan Wang, Dr Chao Wu and Dr Guang Yang. I have enjoyed working with them on various research projects.

- My friends and colleagues at Imperial College London, including Dr Kai Sun, Mr Yuanhan Mo, Mr Jingqing Zhang, Dr Wei Pan, Dr Guangwei Chen, Dr Xian Yang, Dr Douglas McIlwraith, Dr Miguel Molina-Solana, Dr David Birch, Mr David Akroyd, and Ms Diana O'Malley, for their all kinds of help and support.

- My parents Wei and Dianlin, my wife Chaoge (and our feline family members Yudan and Handou) and all the rest of my family, for their unconditional love and unwavering encouragement.

To my parents and my wife

"What I cannot create, I do not understand."

—

Richard Feynman

"Creativity is just connecting things."

—

Steve Jobs

Contents

1. Introduction **25**
1.1. Motivation and objectives 25
1.1.1. Image-to-image generation for higher-degree generative creativity 25
1.1.2. Three paradigms of image-to-image generative creativity 27
1.2. Contributions 28
1.3. Publications 30
1.4. Thesis structure 31

2. Background **33**
2.1. Deep neural networks (DNNs) 33
2.1.1. DNNs for learning deep features 33
2.1.2. Training DNNs 36
2.1.3. Modular layers for constructing DNNs 39
2.2. Generative adversarial networks (GANs) 49
2.2.1. Architectures of GANs 51
2.2.2. Training GANs 56
2.3. Conditional GANs for image-to-image generation . . . 58
2.3.1. Supervised frameworks 59
2.3.2. Unsupervised frameworks 61
2.3.3. Typical applications of image-to-image generative creativity 67

3. Learning to Achieve Enrichment Creativity **73**
3.1. Motivation 73

3.2. Background 75
3.3. Problem formulation 77
3.3.1. Forward and inverse model 77
3.3.2. General CS-MRI reconstruction model 77
3.3.3. Deep neural networks based CS-MRI model . . 78
3.4. Method 79
3.4.1. Objective function 80
3.4.2. Enrichment learning 83
3.4.3. Network architecture 83
3.5. Experiments 85
3.5.1. Experimental settings 85
3.5.2. Results 88
3.6. Conclusions 102

4. Learning to Achieve Transformative Creativity 105
4.1. Motivation 105
4.2. Background 107
4.3. Baseline method 109
4.4. Method 110
4.4.1. Objective function 110
4.4.2. Network architecture 113
4.4.3. Better image representations 115
4.4.4. Visual-semantic text representations 115
4.4.5. Generation of diverse images 116
4.5. Experiments 117
4.5.1. Experimental settings 117
4.5.2. Results 118
4.6. Conclusions 129

5. Learning to Achieve Combinatorial Creativity 133
5.1. Motivation 133
5.2. Problem formulation 136

5.3. Method 137
5.3.1. A path of evolution from CycleGAN 138
5.3.2. DesignGAN 141
5.3.3. Network architecture 146
5.4. Experiments 147
5.4.1. Experimental settings 147
5.4.2. Results 148
5.5. Conclusions 155

6. Conclusions **159**
6.1. Summary of the thesis 159
6.2. Future work 161

Bibliography **163**

A. Supplementary for Chapter 2 **179**

B. Supplementary for Chapter 3 **181**

C. Supplementary for Chapter 4 **183**

D. Supplementary for Chapter 5 **187**

List of Tables

2.1. Commonly used nonlinear activation functions with their definitions and ranges. 40

3.1. Quantitative results (NMSE and PSNR) of our proposed EnrichGAN model (with variants) and the comparison methods, using 1D Gaussian undersampling masks of different ratios. 89

4.1. Comparison results of human evaluation between the baseline method, our proposed SimGAN and SimGAN-V on images from Caltech-200 bird dataset and Oxford-102 flower dataset. 124

5.1. Comparison results of human evaluation between the investigated and our proposed approaches for the bionic design problem. 152

List of Figures

1.1. Schema of three paradigms of image-to-image generative creativity. 29

2.1. An illustrative example of one DNN architecture used for image classification, with visualisation of learned representations in different layers. 35
2.2. Visualisation of 2D convolution process in the convolutional layer. 42
2.3. Visualisation of a forward convolution (for downsampling) and the corresponding transposed convolution (for upsampling). 45
2.4. Visualisation of recurrent layers with (a) the basic RNN cell, and (b) the LSTM cell. 46
2.5. Different normalisation methods used in the normalisation layer. 48
2.6. Schema of the original GAN framework. 51
2.7. Schema of the LAPGAN architecture. 52
2.8. Latent code arithmetic for visual concepts using DCGAN. 53
2.9. The training scheme of PGGAN. 55
2.10. Schema of the architecture of conditional GANs, InfoGAN and AC-GAN. 57
2.11. Schema of the PLDT architecture. 60
2.12. Schema of the Pix2Pix architecture. 60
2.13. Schema of the BicycleGAN architecture. 62
2.14. Schema of the DTN architecture. 63
2.15. Schema of the UNIT architecture. 64

2.16. Schema of the CycleGAN architecture. 65
2.17. Schema of the StarGAN architecture. 66
2.18. Examples of image-to-image generation tasks that exhibit enrichment creativity. 68
2.19. Examples of image-to-image generation tasks that exhibit transformative creativity. 70
2.20. Neural style transfer as an example of image-to-image generation tasks that exhibit combinatorial creativity. . 71

3.1. An example of CS-MRI reconstruction that demonstrates enrichment creativity. 74
3.2. Schema of our proposed EnrichGAN architecture for CS-MRI reconstruction. 80
3.3. Qualitative results of PFPGE using 1D Gaussian undersampling masks with different ratios of retained data. . 90
3.4. Comparison of horizontal line profiles of our proposed EnrichGAN model (with variants), GT and ZF. 91
3.5. Qualitative results of all the comparison methods and our proposed EnrichGAN (with variants), using the 1D Gaussian undersampling mask with 30% retained data. 93
3.6. Quantitative comparison (PSNR) of noise tolerance between the representative methods of CS-MRI reconstruction and our proposed EnrichGAN model (with variants). 95
3.7. Quantitative comparison (PSNR) of representative methods of CS-MRI and our proposed EnrichGAN (with variants) using difference undersampling masks with 30% retained data. 96
3.8. Zero-shot inference on pathological brain MRI images using our proposed EnrichGAN model. 97
3.9. Zero-shot inference on pathological cardiac MRI images using our proposed EnrichGAN model. 98
3.10. Qualitative visualisation of intermediate results of our proposed EnrichGAN model. 100

3.11. Convergence analysis on the use of enrichment learning (PPG vs. PPGE). 101
3.12. Quantitative comparison (PSNR) of different hyperparameter settings of our proposed EnrichGAN model. . . 102

4.1. Examples of semantic image manipulation that exhibits transformative creativity. 106
4.2. Schema of our proposed SimGAN architecture for semantic image manipulation. 111
4.3. Qualitative comparison results of the baseline method and our proposed SimGAN on images from the training set of the Caltech-200 bird dataset. 119
4.4. Zero-shot comparison results of the baseline method and our proposed SimGAN and SimGAN-V on images from the test set of the Caltech-200 bird dataset. 120
4.5. Zero-shot comparison results of our proposed SimGAN and SimGAN-V on images from the test set of the Oxford-102 flower dataset. 121
4.6. Zero-shot generated images by linearly interpolating between representations of two source images with one target text description. 126
4.7. Zero-shot generated images by linearly interpolating between representations of two target text descriptions with one source image. 127
4.8. Results of diverse generation from our proposed SimGAN on same source image and target description. . . 128
4.9. Results of failure cases of SimGAN on images from the Caltech-200 bird dataset and the Oxford-102 flower dataset. 131
4.10. Zero-shot generation results of 256 $\times$ 256 images on the test set of the Caltech-200 bird dataset and the Oxford-102 flower dataset. 132

5.1. Examples of bionic design that demonstrates combinatorial creativity. 135

5.2. The assumption of the bionic design problem. 137
5.3. Schema of CycleGAN architecture for bionic design. . . 138
5.4. Schema of CycleGAN+N architecture for bionic design. 139
5.5. Schema of CycleGAN+2E architecture for bionic design. 141
5.6. Schema of our proposed DesignGAN architecture for bionic design. 142
5.7. Qualitative comparison results between the investigated and our proposed approaches for the shape-oriented bionic design problem. 150
5.8. Comparison results between employing different relative strengths of the regression loss $\mathcal{L}_r$ and the cycle loss $\mathcal{L}_c$ in the DesignGAN architecture. 153
5.9. Generated results of latent variable interpolations by the DesignGAN architecture. 155
5.10. Comparison results between machine design and human design for the shape-oriented bionic design problem. . . 156

A.1. Detailed architecture of the VGG (VGG16) network. . 179

B.2. Employed random undersampling masks 181
B.1. Detailed network architecture of our proposed EnrichGAN model for CS-MRI reconstruction. 182

C.1. Detailed network architecture of our proposed SimGAN model for semantic image manipulation. 184
C.2. Additional zero-shot results of our proposed SimGAN on images from the test set of the Caltech-200 bird dataset. 185
C.3. Additional zero-shot results of our proposed SimGAN on images from the test set of the Oxford-102 flower dataset. 186

D.1. Detailed network architecture of our proposed DesignGAN for bionic design. 187

1. Introduction

1.1. Motivation and objectives

Advances in deep learning techniques over recent years have enabled machines to achieve unprecedented human-level performance on various challenging tasks that involve learning visual features [91, 124, 48, 106]. Generative creativity [87, 135], which refers to the generation of ***novel*** and ***valuable*** images in the context of visual data, still remains a significant challenge in the field of machine learning. Machine with such generative creativity can be used to assist humans for certain tasks (e.g. inspiring designers for fashion design [118]) and produce novel images (e.g. artwork generation [33]), which is a long-term goal in the field of artificial intelligence (AI) and computer vision (CV).

1.1.1. Image-to-image generation for higher-degree generative creativity

To achieve generative creativity for visual data, a prerequisite condition is the capability of generating realistic images. In recent years, several deep generative models, such as variational autoencoder (VAE) [67, 111], autoregressive models (PixelRNN [97] and PixelCNN [98]) and generative adversarial networks (GANs) [41], have been proposed and have obtained impressive results in ***unconstrained image generation***. In this task, images are directly generated either from latent variables (for VAE and GANs), or from a learned joint distribution on pixels over an image (for PixelRNN and PixelCNN), without conditioned on extra information. Among these approaches, GANs are

able to produce the most visually-appealing images, due to its novel adversarial training process, where a learnable objective can be implicitly defined (in feature space) and automatically adaptive to complex image generation tasks. However, for most of the unconstrained image generation problems, GANs can only be employed as a sampling function of synthesising images that are similar to those in the training dataset. Although the synthesised images in such case may still have never existed and are realistic and plausible, which to some degree satisfies the requirements of generative creativity, achieving generative creativity of higher level (i.e. synthesising more creative and useful images) still remains an important need for many practical applications in CV.

In this thesis, we focus on the more challenging ***image-to-image generation***, in which a synthesised image is based on a given input image as an additional condition. Tasks of image-to-image generation typically exhibit generative creativity of higher degree, compared with those of unconstrained image generation, due to the following two reasons. First, numerous practical applications of image synthesis require the manipulation of given images, in which case the generated results tend to be practically more useful and valuable. Second and more importantly, the additional condition imposes a non-negligible constraint on the image generation process, which enforces synthesised images to contain relevant features of input images. As a result, it naturally enables the generation of more creative images that differ from any existing training samples (e.g. a generated image based on a previously unseen input image).

The original GAN architecture can be further extended to the conditional GAN framework [90], based on which different approaches (such as [134, 59, 162, 65, 152, 78]) have been applied for various image-to-image generation problems. However, there still exist other more challenging image-to-image generation tasks that can demonstrate the achievement of higher-degree generative creativity. The challenges lie in three aspects. First, whether the generated images are truly valu-

able and useful, especially for those of critical applications (e.g. in the field of medical imaging). Second, whether the generated images are able to exhibit higher level of novelty. In other words, they should not only have never existed, but also have a clear distinction from any existing samples. Last but not least, whether diverse images can be generated based on one input image, which is a necessary requirement of many image-to-image generation problems.

The primary goal of this thesis is to develop novel architectures and algorithms of deep conditional adversarial networks for challenging image-to-image generation tasks that demonstrate generative creativity.

1.1.2. Three paradigms of image-to-image generative creativity

A typical pipeline of image-to-image generation that requires feature learning includes three steps: 1) features of an input image are extracted by an encoding function, and 2) the encoded features are then manipulated (e.g. adding new features or discarding original features) according to specific tasks, and 3) the manipulated features are finally transformed into generated images by a decoding function. Image-to-image generative creativity comes from the manipulation on input image features. Specifically, according to how features of input images are manipulated, image-to-image generative creativity can be further divided into three categories: enrichment creativity, transformative creativity and combinatorial creativity [87], which are presented in Figure 1.1 and are illustrated as follows.

1. Enrichment creativity, in which output images are generated by preserving most features of the corresponding input images with new features added, in a novel fashion (Figure 1.1 (a)). It can be seen as enriching details of the input images, with their main underlying structure unchanged.
2. Transformative creativity, which involves transforming some fea-

tures of the input images (with other features maintained) into new features, resulting in the generation of creative images (Figure 1.1 (b)). The feature transformation is performed mainly on the input images, i.e. the generated images still largely resemble the input images.

3. Combinatorial creativity, which in general refers to the generation of novel ideas by combining or connecting existing ones, is considered as an essential way for human beings to generate innovations [87, 34]. In the context of image-to-image generation, it involves extracting key features of input images and combine them with salient features of other images, often leading to generated images of high-level creativity (Figure 1.1 (c)).

In this thesis, our main objectives are therefore to develop different deep conditional adversarial networks, which address challenging tasks of image-to-image generation that respectively exhibit enrichment creativity, transformation creativity and combinatorial creativity.

1.2. Contributions

This thesis makes the following three contributions.

- We propose EnrichGAN, a novel conditional adversarial architecture for fast compressed sensing magnetic resonance imaging (CS-MRI) reconstruction, as a challenging application of image-to-image generation that exhibits enrichment creativity. Generating high-quality MRI images from those of low quality is a main challenge of CS-MRI. Many conventional methods and some deep learning approaches have been developed for this task. By contrast, we formulate the task to an image-to-image generation problem, in which the proposed EnrichGAN enriches the local details and structures of input images to match those of ground truth images. It is the first work that addresses this task by conditional GAN-based framework. EnrichGAN outperforms

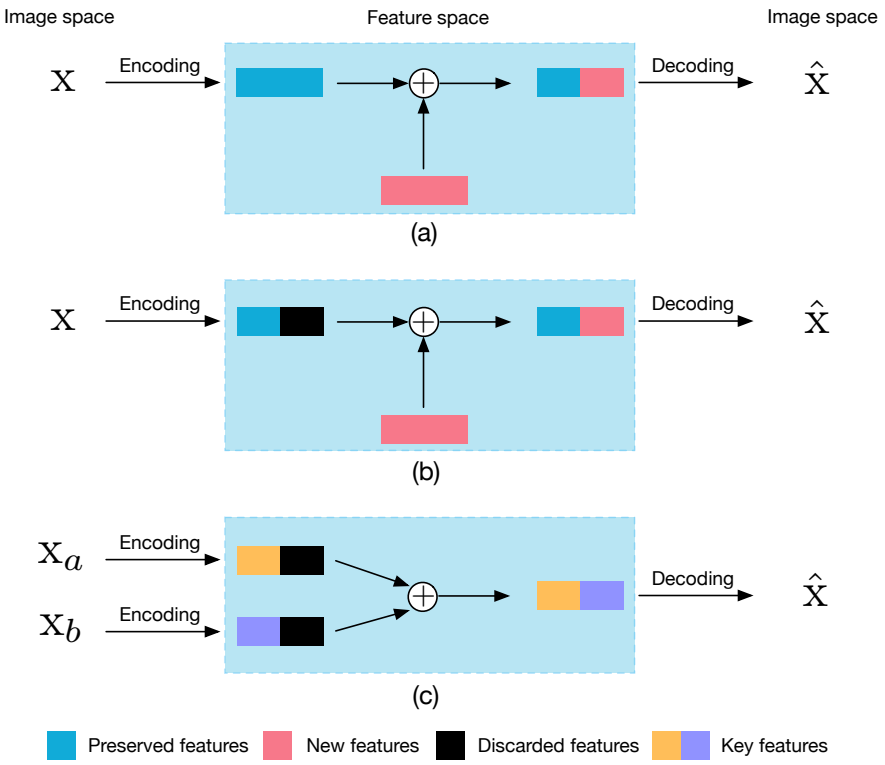


Figure 1.1.: Schema of three paradigms of image-to-image generative creativity according to how features of input images are manipulated. x denotes input images, x_a and x_b denote input images from different domains, $\hat{x}$ denotes generated images. (a) Enrichment creativity. (b) Transformative creativity. (c) Combinatorial creativity.

various conventional and state-of-the-art methods, both qualitatively and quantitatively, with a much faster processing time that enables real-time applications. It is also able to perform zero-shot inference and achieve faithful results. EnrichGAN successfully learns to achieve high-degree enrichment creativity, due to the generation of practically valuable and useful images.

- We propose SimGAN, an innovative conditional adversarial framework for semantic image manipulation. Given an input source image and a target text description, the output images of this

task should meet three conditions: 1) matching the target text description, and 2) preserving other features of the source image that are irrelevant to the description, and 3) being realistic and plausible. This is a challenging image-to-image generation task that demonstrates transformative creativity. It requires to learn a good mapping between visual features and semantic features of natural languages. We perform qualitative and quantitative experiments to show that the proposed SimGAN achieves impressive results on this task. More importantly, given an input source image, SimGAN is capable of generating images that are both diverse and clearly different from any existing images. This indicates that it successfully learns to achieve high-level transformative creativity.

- We propose DesignGAN, a novel conditional GAN-based architecture for automating the process of shape-oriented bionic design. Bionic design refers to a method of product design, where a biologically-inspired object is designed by combining features of a target design object with those of biological source objects. It is a challenging image-to-image generation task that exhibits combinatorial creativity. It requires learning to combine features of images from different domains in an unsupervised manner. We show that our proposed DesignGAN is able to generate creative and plausible shape-oriented bionic designs. Furthermore, given an input image, DesignGAN can also generate diverse images that are significantly distinct from existing samples. It thus learns to achieve high-degree combinatorial creativity.

1.3. Publications

The works of this thesis are based on the following publications and preprints. Those where I am the first or co-first author are marked with *.

- **DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction***, In *IEEE Transactions on Medical Imaging*, 2018. [148]
- **Semantic image synthesis via adversarial learning***, In *International Conference on Computer Vision (ICCV)*, 2017. [24]
- **TensorLayer: a versatile library for efficient deep learning development**, In *ACM on Multimedia Conference (ACM-MM)*, 2017. [23] (best open-source software award)
- **Deep de-aliasing for fast compressive sensing MRI***, In *Preprint arXiv:1705.07137*, 2017. [156]
- **Generative creativity: adversarial learning for bionic design***, In *Preprint arXiv:1805.07615*, 2018. [154]
- **Spatial feature combination for generative creativity: a case study of bionic design***, In *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2018. [155]

1.4. Thesis structure

The rest of this thesis is organised as follows. In Chapter 2, we present the background and related work. In Chapter 3, we describe the proposed EnrichGAN architecture for CS-MRI reconstruction that demonstrates enrichment creativity. In Chapter 4, we present the proposed SimGAN architecture for semantic image manipulation that exhibits transformative creativity. In Chapter 5, we introduce the proposed DesignGAN architecture for the shape-oriented bionic design problem that demonstrates combinatorial creativity. Finally in Chapter 6, we conclude this thesis and describe possible future work. Supplementary materials of Chapter 2, Chapter 3, Chapter 4 and Chapter 5 are presented respectively in the Appendix.

2. Background

This chapter presents background and related work to deep conditional adversarial networks for image-to-image generation. We start with an introduction to deep neural networks, including how they can be applied for learning useful and relevant representations, how they can be trained and how they can be constructed by employing modular layers of different categories. We then review principles behind generative adversarial networks (GANs), with their varied architectures illustrated in detail. Techniques and tricks for stabilising the training process of GANs are also discussed. Finally, we present several prominent frameworks of conditional GANs for image-to-image generation. Typical applications (with conditional GANs applied) that demonstrate image-to-image generative creativity are briefly introduced.

2.1. Deep neural networks (DNNs)

2.1.1. DNNs for learning deep features

Deep neural networks (DNNs) are a class of machine learning algorithms with the ability to learn good representations directly from raw data. This automatic process of representation learning is a key advantage of DNNs, compared with conventional machine learning methods, whose performance heavily depends on manually-designed feature extractors that typically require a substantial amount of domain expertise and engineering competence. Learning representations from raw data in a general-purpose fashion enables artificial intelligence (AI) systems to not only perform better on original tasks, but also adapt to new

tasks more rapidly without much human interference [40].

DNNs are composed of multilayer computational neurons, all of which learn a nonlinear transformation on the outputs of neurons in the previous layer. This multilayer structure enables DNNs to carry two crucial advantages of learning data representations: feature re-use and abstraction [9].

1. Feature re-use. The multiple layers of DNNs make them capable of learning hierarchical features, in which higher-level features are constructed via the composition (i.e. re-use) of lower-level features.
2. Abstraction. The higher-level features learned by DNNs are more abstract than lower-level features. Such abstract features are generally insensitive to variations in observed data, whose intricate structures can then be captured compactly for better addressing AI-related tasks.

DNNs are capable of learning complex functions by hierarchically composing simple nonlinear functions. Figure 2.1 illustrates one DNN architecture with visualised learned representations in multiple layers, for the image classification problem as an example of supervised tasks. Directly mapping the input image (i.e. high-dimensional pixel data) to the correct object identity is very complicated. Instead, the DNN architecture breaks this complex mapping into simple ones. It thus learns to correctly recognise the input image, via the combination of complex object parts (high-level features extracted by the third hidden layer), which are in turn defined by the combination of simple corners/contours and edges (low-level features extracted by the first and second hidden layer).

DNNs can also be applied to learn features from raw data in an unsupervised manner. One such example is the autoencoder framework [51], which consists of an encoder network $\mathrm{h} = f(\mathrm{x})$ (that encodes an input image x into its latent representation h), and a decoder network $\hat{\mathrm{x}} = g(\mathrm{h})$ (that maps h back to image space and produces a reconstructed image $\hat{\mathrm{x}}$). Being trained to reduce the reconstruction error

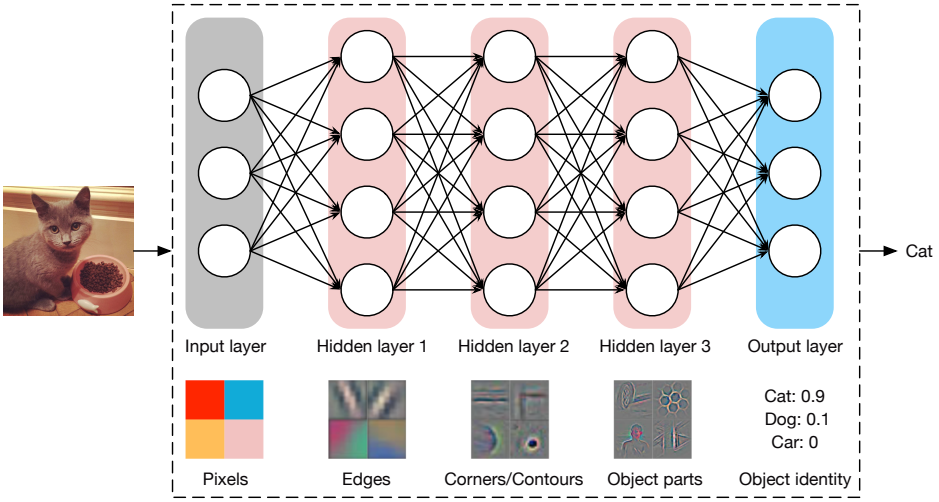


Figure 2.1.: An illustrative example of one DNN architecture used for image classification, with visualisation of learned representations in different layers. The shown DNN architecture consists of multiple layers of neurons: an input layer, three hidden layers and an output layer. The input layer takes as input an image (i.e. pixels). The three hidden layers then learn to extract the increasingly abstract representations of the input image (i.e. edges, corners and contours). Finally, the output layer computes the probability of each category according to the learned highest-level features (i.e. object parts). Images reproduced from [157, 40].

$\mathcal{L}(\mathrm{x}, \hat{\mathrm{x}})$ (e.g. the mean squared error), the autoencoder can capture the most useful and relevant features of the given training data, without any external supervision signal.

The autoencoder framework can be further generalised as a paradigm of generative modelling. For example, in this thesis, we exploit variants of such encoder-decoder framework for various image-to-image generation tasks, in which the original reconstruction loss is replaced by other losses (e.g. the adversarial loss), as a constraint to enforce the manipulation of the extracted features of input images for achieving generative creativity (as shown in Figure 1.1). This will be detailed in Section 2.3 and later chapters.

2.1.2. Training DNNs

The three hidden layers in the example DNN architecture presented in Figure 2.1 are fully-connected layers with activation function, in which each neuron performs the following nonlinear transformation:

$$\mathrm{y} = f(\sum_i (w_i \mathrm{x}_i + b)), \tag{2.1}$$

where x_i represents the output of the i-th neuron in the previous layer, w_i represents the corresponding connection parameter, b represents a bias parameter, f represents a nonlinear activation function. DNNs can then be viewed as a complex nonlinear function $g(\mathrm{x}|\theta)$ parameterised by θ that contains parameters of all neurons.

To find the optimal θ (denoted as θ^*), DNNs are typically trained by employing the gradient descent optimisation strategy. More specifically, this training strategy involves iterative optimisation steps (until convergence), each of which can be divided into the following four phases:

1. Forward pass. Samples from the training set are fed into DNNs and forward propagated through each layer, resulting in the final output scores (e.g. class probabilities computed by the softmax function for the classification problem).
2. Error measurement. The error between the output scores and the ground truth (GT) (denoted as y) is computed by using an objective function $\mathcal{L}(g(\mathrm{x}|\theta), \mathrm{y})$ (e.g. the cross-entropy loss function for the classification problem).
3. Backpropagation. We then compute the gradients of the objective function with respect to the parameters θ of DNNs, by adopting the backpropagation algorithm [113]. Starting from the output layer, backpropagation enables the gradients to be propagated backwards to the input layer, which is achieved by recursively applying the chain rule for derivatives.
4. Parameter update. Finally, each parameter is updated by taking

steps in the negative direction of the corresponding computed gradient, so as to minimise the error measured by the objective function.

The gradient descent optimisation algorithm requires the use of all training samples for computing gradients. However, if the training set is too large, this algorithm will have prohibitively high computational and memory complexity. In practice, DNNs are mostly trained by employing stochastic gradient descent (SGD) or its variants, in which a small number of training samples (termed mini-batch) are used to approximately estimate the gradients over all training samples. The parameters θ are then updated as follows:

$$\theta \leftarrow \theta - \eta g, \tag{2.2}$$

where η denotes the learning rate and $g = \frac{1}{n}\sum_i^n \nabla_\theta \mathcal{L}(g(\mathrm{x}_i|\theta), \mathrm{y}_i)$ denotes the average gradients with respect to θ over a mini-batch. η is a critical hyperparameter for the SGD optimisation and requires to be tuned carefully. Also, rather than using a fixed η, one common practice is to gradually reduce the initial η over the training process to improve convergence performance.

To further accelerate the convergence rate, several extensions to the SGD algorithm have been proposed. For instance, the momentum method [102] and its variant Nesterov momentum [129] employ the exponential moving average of previously computed gradients to update the parameters. These methods will encourage the update of parameters if their previous gradients are of same direction, and will damp the update if the gradients are of distinct directions (i.e. the oscillations are reduced), which enables faster and more stable convergence. In addition, there are other methods that compute separate learning rates adaptively for each parameter during the training process. The AdaGrad method [26] scales the individual learning rate of every parameter by the square root of sum of squares (SRSS) of all the corresponding historical gradients. As a result, all the learning rates are monoton-

ically decreasing throughout the learning process, and the decreasing speed is adaptively dependent on the gradient magnitude. One problem of AdaGrad is that the learning rates decrease too aggressively, because of which the training process may stop too early. To solve this problem, an extension of AdaGrad is proposed: the RMSProp algorithm [136], which employs the exponential moving average of SRSS of the gradients for scaling the learning rates.

Algorithm 2.1 The Adam algorithm [66]

Require: learning rate η
Require: exponential decay rates $\beta_1, \beta_2 \in [0, 1)$
Require: small constant to ensure numerical stability ϵ
Require: initial parameters θ
$m \leftarrow 0$ ▷ initialise 1^{st} moment variable
$v \leftarrow 0$ ▷ initialise 2^{nd} moment variable
$t \leftarrow 0$ ▷ initialise time step
while θ not converged **do**
  $t \leftarrow t + 1$
  get a mini-batch of training samples
  $g \leftarrow \frac{1}{n} \sum_i^n \nabla_\theta \mathcal{L}(g(\mathrm{x}_i|\theta), \mathrm{y}_i)$ ▷ compute gradients
  $m \leftarrow \beta_1 m + (1 - \beta_1) g$ ▷ update 1^{st} moment estimate
  $v \leftarrow \beta_2 v + (1 - \beta_2) g \odot g$ ▷ update 2^{nd} raw moment estimate
  $\hat{m} \leftarrow m / (1 - \beta_1^t)$ ▷ correct bias in 1^{st} moment
  $\hat{v} \leftarrow v / (1 - \beta_2^t)$ ▷ correct bias in 2^{nd} raw moment
  $\theta \leftarrow \theta - \eta \hat{m} / (\sqrt{\hat{v}} + \epsilon)$ ▷ update parameters
end while
return θ

Specifically for training the various DNNs proposed in this thesis, we employ the Adam algorithm [66], which can be viewed as combining the advantages of momentum and RMSProp. Adam computes the adaptive learning rates for separate parameters from the exponentially weighted estimates of the 1^{st} moment (the mean) and the 2^{nd} raw moment (the uncentered variance) of the gradients. Algorithm 2.1 presents the pseudocode of the Adam algorithm.

2.1.3. Modular layers for constructing DNNs

Various architectures of DNNs can be constructed by stacking different categories of compositional layers in a modular fashion. In this section, we introduce a number of such layers, based on which our proposed network architectures for generative creativity in this thesis are composed of.

Fully-connected layer

As already mentioned in the above section, a fully-connected layer contains neurons that have connections to all neurons in the previous layer. It thus can be seen as globally learning relatively higher-level representations from combinations of every lower-level representation learned in the preceding layer.

The fully-connected layer can be applied as a classifier that performs final classification based on the extracted high-level features. However, it is impractical to directly learn features from high-dimensional data (e.g. images). This is because in such case the full connectivity requires a large number of parameters to be optimised, which makes it computationally expensive and tends to cause overfitting.

Activation layer

DNNs involve a nonlinear transformation of representations through different levels. Such nonlinearity is introduced by the activation layer (i.e. nonlinear activation function). There are several commonly used activation functions, such as sigmoid, tanh, ReLU, Leaky-ReLU (L-ReLU) [83], and Parametric-ReLU (P-ReLU) [47], whose definitions and ranges are summarised in Table 2.1. In our proposed DNN architectures, sigmoid is used to produce probability output, while tanh is employed for generating pixels (whose values are from -1 to 1). We also apply ReLU and its variants as the activations in the hidden layers of the networks.

Table 2.1.: Commonly used nonlinear activation functions with their definitions and ranges. Note the difference between L-ReLU and P-ReLU: α in L-ReLU is a constant while ω in P-ReLU is a learnable parameter.

Activation function	Definition	Range
Sigmoid	$\sigma(\mathrm{x}) = \dfrac{1}{1+e^{-\mathrm{x}}}$	$(0, 1)$
Tanh	$\tanh(\mathrm{x}) = \dfrac{e^{\mathrm{x}} - e^{-\mathrm{x}}}{e^{\mathrm{x}} + e^{-\mathrm{x}}}$	$(-1, 1)$
ReLU	$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$	$[0, \infty)$
L-ReLU	$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{otherwise} \end{cases}$	$(-\infty, \infty)$
P-ReLU	$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \omega x, & \text{otherwise} \end{cases}$	$(-\infty, \infty)$

Convolutional layer

Convolutional neural networks (CNNs) [72, 73] are a class of neural networks specifically designed for processing data in the form of arrays [71] (e.g. image data that can be viewed as two-dimensional (2D) or three-dimensional (3D) arrays of pixels). Several deep CNN architectures proposed in past few years (such as AlexNet [69], VGGNet [126], GoogLeNet [132] and ResNet [48]) have achieved spectacular results on large-scale image recognition and object detection. These deep CNNs have further inspired the design of many other DNN architectures, which have obtained state-of-the-art results on various challenging tasks (e.g. [80, 91, 144, 124]).

The core component of the CNN architectures is the convolutional layer, which contains one or more groups of learnable parameters (termed filters or kernels) that perform convolution on the input volume. Figure 2.2 illustrates the 2D convolution process in the convolutional layer.

Specifically, the input volume is of size $W \times H \times D$, and the filter is of size $K \times K \times D$. Note that the filter depth is always equal to the input volume depth. This filter then convolves with the input volume by moving along the width and height direction, resulting in a 2D output map. Multiple filters can be applied with activation functions to produce an output volume (i.e. feature maps), in which each entry is computed by the following equation:

$$\mathrm{y}_{i,j,l} = f(\sum_{d=0}^{D-1}\sum_{k=0}^{K-1}\sum_{k=0}^{K-1} w_{k,k,d,l}\mathrm{x}_{Si+k,Sj+k,d} + b_l), \tag{2.3}$$

where i, j and l denote the position of the entry y, x denotes the entries in the input volume, w and b denote the parameters of the filter, S denotes stride that specifies the moving steps of the filter when performing the convolution, f denotes the nonlinear activation function. The depth of the output volume (denoted as D') is equal to the number of filters, while the weight and height (denoted as W' and H') are determined by:

$$\begin{aligned} W' &= \frac{W + 2P - K}{S} + 1, \\ H' &= \frac{H + 2P - K}{S} + 1, \end{aligned} \tag{2.4}$$

where P denotes the size of zero-padding, in which the input volume is padded with border of zeros. Zero-padding can flexibly control the size of the output volume, e.g. maintaining it same as that of the input volume.

The key ideas behind the convolutional layer are local connectivity and parameter sharing, which respectively correspond to two prior assumptions about the representations of image data. First, the representations (especially low-level ones) tend to be local (e.g. edges in an image are formed by few adjacent pixels). Local connectivity enforces the filters in the convolutional layer to be of small size and only interact with local patches of images. Second, the representations are

translationally invariant (e.g. a primitive shape may appear at different positions of an image). The filters therefore preserve same parameters to learn similar representations at all locations of images. These two ideas enable the convolutional layer to extract better representations with much fewer parameters (i.e. better generalisation), compared with the fully-connected layer.

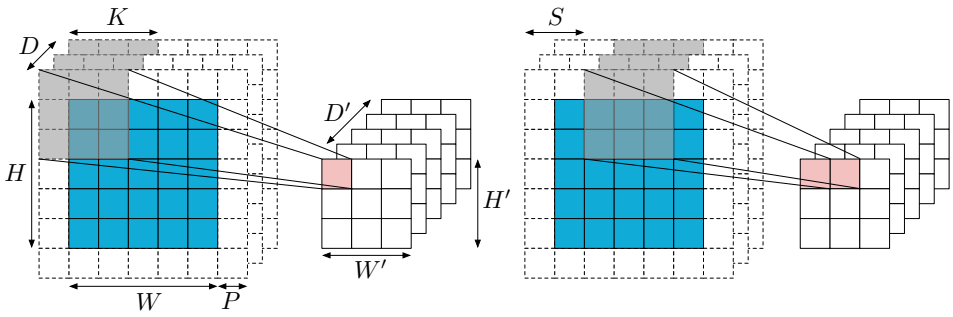


Figure 2.2.: Visualisation of 2D convolution process in the convolutional layer. The filter (in grey) is of size $K \times K \times D$. The input volume (in blue) is of size $W \times H \times D$, with zero-padding of size P. The stride is of size S. Each entry in the output volume (in pink) is determined by Equation 2.3. The output volume is of size $W' \times H' \times D'$, which is computed by Equation 2.4.

Pooling layer

The pooling layer is another building block used extensively in many CNN architectures. Specifically, one type of the pooling layer (including max pooling, average pooling and l_2-norm pooling) is typically used to downsample the feature maps (produced by the convolutional layer) spatially along its width and height direction. For instance, the (2D) max pooling layer with window of size 2 and stride of size 2 will transform the input volume of size $W \times H \times D$ into the output volume of size $W/2 \times H/2 \times D$, by progressively applying the max function on the spatial 2×2 patches. This pooling process can be seen as enforcing the learned features to be invariant to small shifts and distortions of

the input [71, 40]. The downsampled feature maps also imply that the number of required parameters is decreased, which helps reduce the possibility of overfitting. It is worth noting that such downsampling process can also be achieved by employing the convolutional layer with stride [127, 105]. In this case, the downsampling function is learned by the CNN (with more parameters), rather than being deterministic in the non-parametric pooling layer.

Another commonly used pooling layer is the global average pooling [76]. Given feature maps of size $W \times H \times D$, the global average pooling averages over each of the feature maps, resulting in an output of size $1 \times 1 \times D$. It is typically used after the last convolutional layer and before the fully-connected layer or the final output layer (e.g. softmax for classification), as a structural regulariser to prevent overfitting [76]. More importantly, the global average pooling enables CNNs to process input images of varied sizes, as their extracted feature maps (also of varied sizes in width and height) can always be transformed into the output of same size.

Many CNN architectures are composed of multiple convolutional layers, pooling layers and fully-connected layers. An example is the VGG (VGG16) network [126], which is employed as the (pretrained) feature extractor in Chapter 3 and Chapter 4. A detailed visualisation of the network architecture of VGG is presented in Figure A.1 in Appendix A.

Transposed convolutional layer

Transforming representations in latent space back into image space is an essential process for many image generation tasks. It typically requires the use of multiple upsampling functions that map the lower-dimensional input volume to the higher-dimensional output volume. The transposed convolutional layer is one way to perform the upsampling function, in which the transposed convolution [157] can be seen as the backward pass of a forward convolution [80] (i.e. the process

of computing gradients of convolution with respect to its input [27]). A clearer view to understand this upsampling process is to consider the transposed convolution as a forward convolution on the fractionally strided input volume with zeros added [29], which is illustrated in Figure 2.3.

The transposed convolutional layer together with the activation layer enables the learning of nonlinear upsampling function. However, it tends to produce the checkerboard-patterned artifacts in the generated images [94], and thus requires careful tuning of the hyperparameters to avoid such artifacts.

Upsampling layer

The nonlinear upsampling function can also be learned by employing an upsampling layer with a convolutional layer and an activation layer. Specifically, the upsampling layer refers to non-parametric interpolation methods that directly rescale input volume to be of higher dimension. Commonly used interpolation methods include nearest-neighbour interpolation and bilinear interpolation. The rescaled input volume is then fed into the convolutional layer (normally with stride of size 1 and zero-padding to preserve the dimension) with nonlinear activation. Such learned upsampling function tends to generate smoother images without checkerborad artifacts [94], though it is at the expense of higher computational cost (because the convolution is performed on the input volume of larger size), compared with the one learned by the transposed convolutional layer.

Recurrent layer

Recurrent neural networks (RNNs) are a class of neural networks specialised for processing sequential data. A RNN can be used as either an encoding function to extract representations of a given sequential input (e.g. as employed in Chapter 4 for learning text features), or a decoding function to generate sequential data. RNNs have achieved impressive

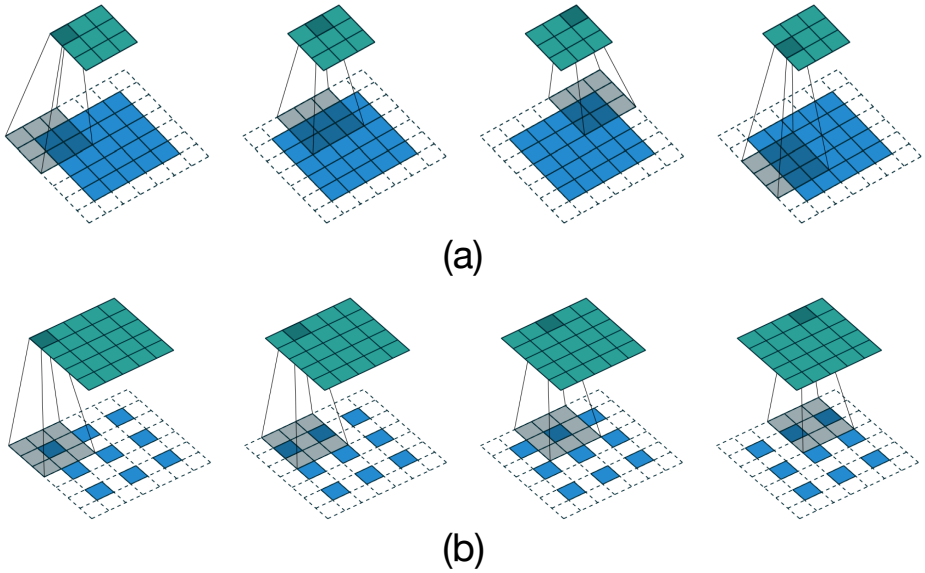


Figure 2.3.: Visualisation of a forward convolution (for downsampling) and the corresponding transposed convolution (for upsampling). (a) A convolution (with filter size of 3×3, stride of size 2×2 and zero-padding of size 1) that downsamples a 5×5 input to a 3×3 output. (b) A fractionally strided convolution on the $1/2$ strided input with zero-padding, which is equivalent to the transpose of convolution in (a). Images reproduced from [29].

successes in a variety of tasks, such as language modelling [89], machine translation [130, 6], image captioning [63, 144], etc.

A RNN is composed of variable number of recurrent layers, each of which is associated with a same type of cell that contains parameters and performs operations to update hidden states. For example, each recurrent layer with the basic RNN cell (Figure 2.4 (a)) takes as input a concatenation of an element $\mathrm{x_t}$ in the given sequential data at timestep t (e.g. the t-th word in the given sentence) and the hidden state $\mathrm{h_{t-1}}$ from the previous recurrent layer. It then produces an updated hidden state $\mathrm{h_t}$ according to the follow equation:

$$\mathrm{h_t} = \tanh(W[\mathrm{h_{t-1}}; \mathrm{x_t}] + b), \tag{2.5}$$

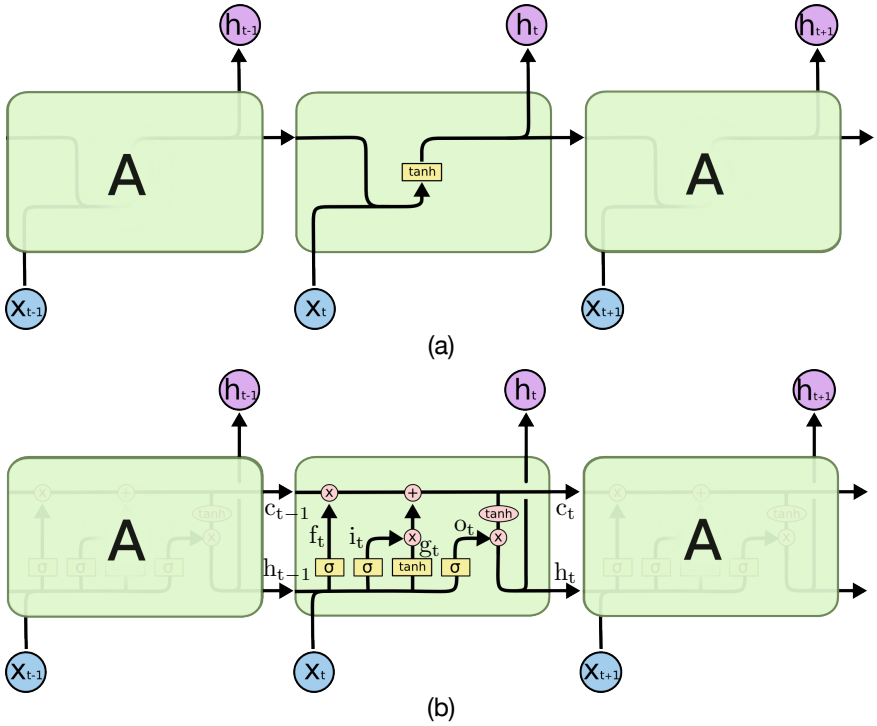


Figure 2.4.: Visualisation of recurrent layers with (a) the basic RNN cell, and (b) the LSTM cell. Images reproduced from [96].

where W and b are parameters that are shared across all recurrent layers of different timesteps (i.e. a RNN repeatedly uses one recurrent layer). The produced hidden state h_t can be seen as the learned representations for the observed sequential data by timestep t.

Compared with feedforward neural networks (e.g. CNNs or fully-connected networks), the most important advantage of RNNs is their ability to model temporal dependencies in the sequential data. However, in practice, RNNs with the basic cell are unable to learn long-term dependencies, due to the problem of gradient vanishing and exploding [10]. Long short-term memory (LSTM) cell [52, 37] addresses this limitation by introducing an extra hidden state (termed memory cell state), which enables the gradient to flow through all timesteps. Specifically, as illustrated in Figure 2.4 (b), the memory cell state c_{t-1}

is controlled by three gates (with learnable parameters): 1) the forget gate $\mathrm{f_t}$ that decides the old information to be removed from memory, and 2) the input gate $\mathrm{i_t}$ that decides the new input information (scaled by $\mathrm{g_t}$) to be added into the memory, and 3) the output gate $\mathrm{o_t}$ that decides the information in the updated memory cell state $\mathrm{c_t}$ to be outputted as new hidden state $\mathrm{h_t}$. The equations for computing forward propagation of recurrent layer with the LSTM cell are given as follows:

$$\begin{aligned}
\mathrm{f_t} &= \sigma(W_\mathrm{f}[\mathrm{h_{t-1}}; \mathrm{x_t}] + b_\mathrm{f}), \\
\mathrm{i_t} &= \sigma(W_\mathrm{i}[\mathrm{h_{t-1}}; \mathrm{x_t}] + b_\mathrm{i}), \\
\mathrm{g_t} &= \tanh(W_\mathrm{g}[\mathrm{h_{t-1}}; \mathrm{x_t}] + b_\mathrm{g}), \\
\mathrm{o_t} &= \sigma(W_\mathrm{o}[\mathrm{h_{t-1}}; \mathrm{x_t}] + b_\mathrm{o}), \\
\mathrm{c_t} &= \mathrm{f_t} \odot \mathrm{c_{t-1}} + \mathrm{i_t} \odot \mathrm{g_t}, \\
\mathrm{h_t} &= \mathrm{o_t} \odot \tanh(\mathrm{c_t}),
\end{aligned} \tag{2.6}$$

where $\odot$ denotes the element-wise multiplication, σ denotes the sigmoid function.

Many variants of the introduced standard LSTM cell have been proposed, e.g. LSTM with peephole connections [38] (in which the gates also take the memory cell state into account for better learning "precise timings" [38]) and gated recurrent unit [16] (GRU, a simplified LSTM cell with low computational cost). For a comprehensive comparison between the typical LSTM variants, please refer to [42].

Normalisation layer

Training DNNs via SGD algorithms suffers from the problem of "covariate shift" [58], in which distribution of inputs to one layer constantly changes during the training course, due to the update of parameters of the preceding layers. This results in a much slower training process, in which lower learning rate and better initialisation of parameters are required [58].

The normalisation layer reduces the covariate shift by normalis-

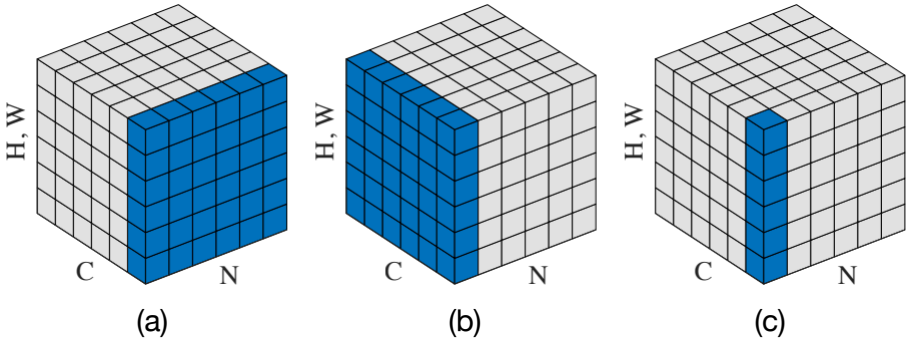


Figure 2.5.: Different normalisation methods used in the normalisation layer. The input volume is defined by three axes: the (mini-)batch axis (N), the channel axis (C) and the spatial (height and width) axis (H, W). The same mean and variance is computed over the elements in blue for the normalisation. (a) Batch normalisation. (b) Layer normalisation. (c) Instance normalisation. Images reproduced from [143].

ing the inputs by their mean and variance over a certain dimension (which dimension depends on the specific normalisation method). Figure 2.5 illustrates three commonly used normalisation methods (including batch normalisation [58], layer normalisation [5] and instance normalisation [138]) on a batch of feature maps for 2D images. The feature maps are integrated as one input volume, which is defined by three axes: the (mini-)batch axis N, the channel axis C and the spatial (height and width) axis (H, W). All the normalisation methods can be formulated as follows [143]:

$$\begin{aligned} \mu_i &= \frac{1}{|S_i|} \sum_{k \in S_i} \mathrm{x}_k, \\ \sigma_i^2 &= \frac{1}{|S_i|} \sum_{k \in S_i} (\mathrm{x}_k - \mu_i)^2, \\ \hat{\mathrm{x}}_i &= \frac{\mathrm{x}_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \\ \mathrm{y}_i &= \gamma \hat{\mathrm{x}}_i + \beta, \end{aligned} \tag{2.7}$$

where i denotes the position of an entry in the input volume, S_i denotes a set of entries for computing the mean μ_i and variance σ_i^2, ϵ denotes a small constant to ensure numerical stability, γ and β are learnable parameters that scale and shift the normalised entry for restoring the representation capability of the network [58]. The main difference between the normalisation methods lies in the definition of S_i: batch normalisation is computed along (N, H, W), layer normalisation is computed along (C, H, W) and instance normalisation is computed along (H, W). In practice, the normalisation layer is typically applied before the activation layer, resulting in a more stable distribution of activations.

Batch normalisation has been demonstrated as a very effective method to accelerate the training of various DNNs. It not only improves the flow of gradient through a network, but also enables a much higher learning rate and reduces the dependence on parameter initialisation. In addition, it introduces extra noise into the training process (due to the uncertainty involved in the mini-batch statistics), which regularises the network and thus reduces the need of using dropout [128]. By contract, layer normalisation is commonly applied in RNNs, in which batch normalisation cannot be effectively used due to the reuse of parameters and variable length of sequential input. Batch normalisation is also not applicable if small batch size is required, in which case instance normalisation becomes a good alternative.

2.2. Generative adversarial networks (GANs)

Deep generative models refer to DNN-based approaches that learn to approximate a data distribution in an unsupervised manner. Specifically in the context of image synthesis, given images sampled from an empirical data distribution p_{data} as training data, a deep generative model either explicitly learns an estimated distribution p_{model}, or im-

plicitly learn to sample images from p_{model}, where $p_{\text{model}} \approx p_{\text{data}}$ [39]. Currently, there are three most popular deep generative models for image generation. Two of them are autoregressive models (e.g. PixelRNN [97] and PixelCNN [98]) and variational autoencoder (VAE) [67, 111, 43], which learn an explicit p_{model} respectively over pixels and latent representations. By contrast, generative adversarial networks (GANs) learn to draw samples from an implicitly-learned p_{model}. GANs tend to generate higher-quality images than those generated by other approaches.

The original GAN framework, as demonstrated in Figure 2.6, is made up of two networks: a generator G and a discriminator D. Specifically, G takes as input a noise vector z sampled from a prior distribution $p_{\text{z}}(\text{z})$ (e.g. a Gaussian distribution), and generates a fake image $\hat{\text{x}} = G(z)$. $\hat{\text{x}}$ can be seen as a sample from the learned generative distribution p_G. D discriminates between $\hat{\text{x}}$ (as fake data) and x sampled from real data distribution $p_{\text{data}}(\text{x})$ (as real data). The generator G attempts to generate samples that resemble real data. G and D are thus trained by playing the two-player minimax game in a competing manner, according to the following objective function:

$$\begin{aligned} \min_G \max_D \mathcal{L}_a(G, D) = \mathbb{E}_{\text{x}\sim p_{\text{data}}(\text{x})}[\log D(\text{x})] \\ + \mathbb{E}_{\text{z}\sim p_{\text{z}}(\text{z})}[\log(1 - D(G(\text{z})))], \end{aligned} \tag{2.8}$$

where $\mathcal{L}_a(G, D)$ denotes the adversarial loss for G and D, and $D(\text{x})$ denotes a probability that x is real. Note that at early training stage, when G only generates poor images, D tends to assign them low probability, which makes $\log(1 - D(G(\text{z})))$ saturate and unable to provide sufficient gradient to update G [41]. A standard way in practice is to maximize $\log D(G(\text{z}))$ for better training G.

It has been theoretically shown that the minimax game will achieve an equilibrium in which the generative distribution is identical to real data distribution ($p_G = p_{\text{data}}$) and $D(\text{x})$ outputs 0.5 for all x (i.e. the discriminator cannot distinguish fake images from real ones), provided

that the generator and discriminator have enough capacity [41].

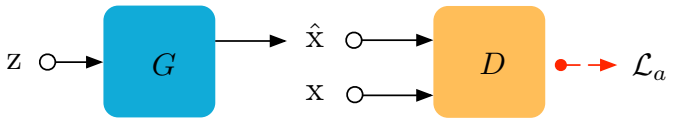


Figure 2.6.: Schema of the original GAN framework. It consists of a generator G (that takes as input noise vector z and outputs generated images $\hat{x}$), and a discriminator D that distinguishes $\hat{x}$ from real images x. G and D are trained adversarially in a two-player minimax game, according to Equation 2.8.

A major advantage of GANs is that the discriminator D can be seen as defining a learnable loss that can be flexibly adapted to various tasks and datasets, in which it would be difficult or impossible to explicitly define one. Such adaptive loss enables GANs and their variants to be applicable to solve many challenging image generation problems, as presented in this thesis.

2.2.1. Architectures of GANs

Various extensions of the original GAN architecture have been proposed for generating images of either higher resolution or conditional constraints. In this section, we introduce several representative architectures that have been shown to perform well in practice.

LAPGAN

One way to generate higher-resolution images is to stack multiple GANs to form a multi-stage generation process, in which images are generated in a coarse-to-fine manner. LAPGAN [19] is one such example that employs a cascade of GANs to generate images of high resolution from those of low resolution. As illustrated in Figure 2.7, LAPGAN consists of four stages of generation. At the first stage, a generator G_3 takes as input a noise vector z_3 and generates $\hat{x}_3$ of low

resolution (8×8). At the second stage, $\hat{x}_3$ is first upsampled to $\hat{u}_2$, which is then fed into another generator G_2 with a noise vector z_2, resulting in a difference image $\hat{d}_2$. $\hat{u}_2$ is added with $\hat{d}_2$ to create a higher-resolution image $\hat{x}_2$ (of size 16×16). With another two similar stages, LAPGAN generates the final output image $\hat{x}_0$ of size 64×64. Note that the generator and the discriminator in each stage are trained separately.

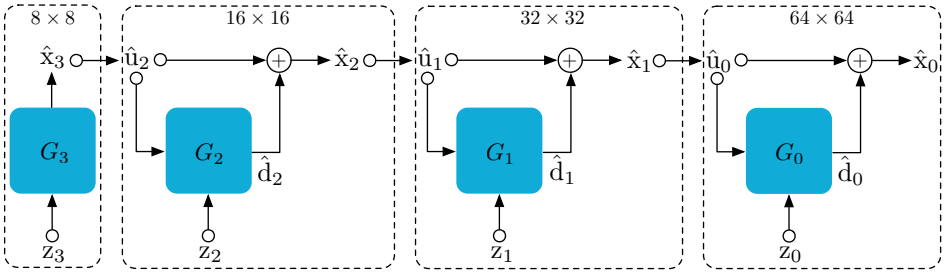


Figure 2.7.: Schema of the LAPGAN architecture. It is able to generate higher-resolution images in a coarse-to-fine manner by cascading multiple GANs.

LAPGAN is the first GAN model that is able to generate plausible 64×64 images. Other GAN frameworks that consist of multiple stacked GANs include S^2-GAN [141], StackGAN [159, 158], SGAN [55] and AttnGAN [145].

DCGAN

DCGAN [105] is the first GAN architecture that is capable of generating images of high resolution by using a single generator. It has affected the design of many different GAN architectures for various tasks of image generation. Specifically, DCGAN is composed of a generator and a discriminator, both of which are made up of convolutional layers with the following key architectural constraints and tricks [105].

1. Use the convolutional layer (with stride of 2) in the discriminator and the transposed convolutional layer (on the $1/2$ strided input

with zero-padding) in the generator, rather than any pooling layer.

2. Use the normalisation layer (with batch normalisation [58]) in the generator (expect its last layer) and the discriminator (expect its first layer).
3. Use the ReLU activation in the generator and the Leaky-ReLU activation in the discriminator.
4. Remove hidden fully-connected layers.
5. Use the Adam [66] optimiser.



Figure 2.8.: Latent code arithmetic for visual concepts using DCGAN. Simple arithmetic operations in latent space reflect semantic correspondence in pixel space, which infers that distributed representation (i.e. compact features that are independently varied and not mutually exclusive) has been successfully learned in an unsupervised fashion. Images reproduced from [105].

These architectural constraints and tricks significantly stabilise the training procedure of GANs in most settings, resulting in generated images of high quality. In addition, DCGAN for the first time demonstrates that the GAN architecture is able to learn relevant and useful representations for generative modelling in an unsupervised manner, as presented in Figure 2.8. Distinct underlying factors of variation (e.g. male/female and with/without glasses) can be automatically captured in each direction of latent feature space by the GAN architecture.

PGGAN

PGGAN [64] enables the generation of images of unprecedented high resolution (1024×1024). It learns to generate images in multiple stages as similar to LAPGAN, but with a critical difference: only one single GAN is used rather than the cascaded ones. PGGAN introduces a training scheme, in which layers are progressively added to the networks along with the training process for generating images of increasing resolutions, as demonstrated in Figure 2.9. The number of layers in the generator G and the discriminator D synchronously grows, with the newly added layers linearly faded in, which avoids sudden influence on the already trained layers [64]. Such training methodology enables PGGAN to first generate the underlying structure, and then add fine details progressively, which stabilises and accelerates the training of GANs dramatically.

Conditional GANs

The original GANs can be further extended to the conditional GANs, in which the generator G and the discriminator D are conditioned on additional constraints. In general, the conditional information can be of different types, such as categorical labels, images and texts. Figure 2.10 (a) presents the architecture of the conditional GANs (first proposed in [90]), where G and D take as input an extra conditional information c. The objective function of the conditional GANs is defined as follows:

$$\begin{aligned}\min_G \max_D \mathcal{L}_a(G, D) = \mathbb{E}_{(\mathrm{x},\mathrm{c})\sim p_{\mathrm{data}}(\mathrm{x},\mathrm{c})}[\log D(\mathrm{x}, \mathrm{c})] \\ + \mathbb{E}_{\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z}),\mathrm{c}\sim p_{\mathrm{data}}(\mathrm{c})}[\log(1 - D(G(\mathrm{z}, \mathrm{c})))].\end{aligned} \tag{2.9}$$

Compared with the original GANs, the conditional GANs are able to provide direct control on the generated images according to the imposed conditional constraints. Such control makes the GAN architecture suitable to be applied in a variety of more complicated im-

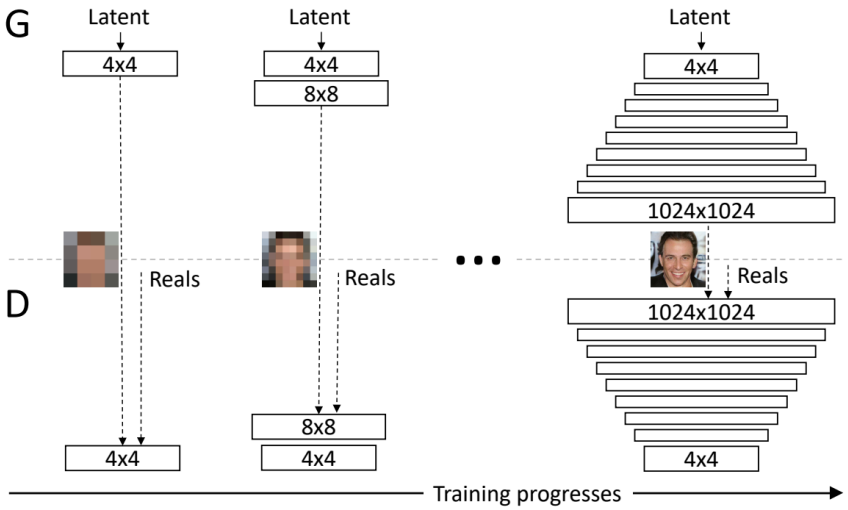


Figure 2.9.: The training scheme of PGGAN. The initial image to be generated is of size 4×4. Layers are then added synchronously to both the generator G and the discriminator D in a progressive fashion during the training process, resulting in generated images of increasing resolution. All the existing layers are trainable during the entire training procedure. The final output images are of unprecedented high resolution (1024×1024). Image from [64].

age generation tasks (e.g. generating images from class labels, texts, images, etc.). In this thesis, our proposed approaches for image-to-image generation can be seen as variants of the conditional GANs (i.e. conditioning the original GANs on images). The relevant conditional GAN-based methods for image-to-image generation will be reviewed in detail in Section 2.3.

InfoGAN

A variant of the conditional GANs is InfoGAN [14], which is proposed to learn useful representations in an unsupervised fashion. The key idea is to decompose the noise vector of the original GANs into an incompressible noise z and a "latent code" c (as an extra condition). A regularisation term is further used to ensure that c is related to the

generated images $\hat{\mathrm{x}} = G(\mathrm{z}, \mathrm{c})$, i.e. the information in c is preserved as much as possible by the generative function G. It has been shown that InfoGAN successfully discovers and disentangles the underlying factors of variation: the learned representations are interpretable and semantically meaningful (e.g. the hair styles, emotions and poses of the generated face images). Figure 2.10 (b) presents the schema of InfoGAN architecture.

AC-GAN

AC-GAN [95] is another variant of the conditional GANs for generating class-conditional images of high resolution. Specifically, it employs an auxiliary classifier in the discriminator D, which not only distinguishes between real and fake samples, but also performs classification on the given images. This is a simple but effective extension, which stabilises the training of GANs and produces sharp images of all 1000 classes of the complicated ImageNet dataset [114]. The schema of the AC-GAN architecture is presented in Figure 2.10 (c).

2.2.2. Training GANs

Training the original GANs in practice can be remarkably challenging and unstable. Empirically, there are three most common failures occurred within the training process [18].

1. Non-convergence: the generator and the discriminator fail to reach an equilibrium. Different samples can be generated along with the training procedure, but their quality is not clearly improved.
2. Mode collapse: the generator is only able to produce similar samples given different inputs [116].
3. Quick convergence of the discriminator loss: the optimal discriminator will be unable to provide sufficient gradient to update the generator [4].

The unstable GAN training can be improved by either employing

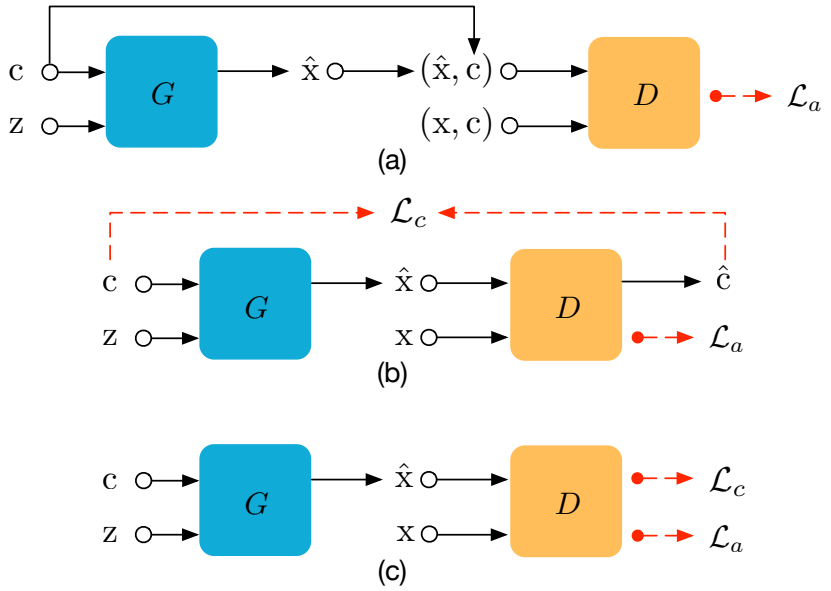


Figure 2.10.: Schema of the architecture of conditional GAN, InfoGAN and AC-GAN. Among these three frameworks, all the generators G similarly take as input an extra conditional information c, while the discriminators D have different structures and losses. (a) Conditional GANs. (b) InfoGAN. (c) AC-GAN.

carefully designed architectures (e.g. DCGAN and PGGAN as introduced previously), or using heuristic training tricks. Specifically, the most commonly used training tricks are summarised as follows.

1. One-sided label smoothing [116], which replaces the target label of real samples in the discriminator from 1 to 0.9. This can be regarded as a regulariser to reduce confidence of the discriminator (in the correct class), which encourages it to give strong gradients to update the generator. A similar idea is to add noise to both real and fake samples, before they are fed into the discriminator [133].
2. Experience replay [122], in which the discriminator is updated using previously generated samples stored in memory, rather than the latest generated ones. This greatly improves the convergence

of adversarial learning.

3. Mini-batch discrimination [116], in which the discriminator takes as input an extra term of similarity between images in a same mini-batch. The discriminator can then easily distinguish generated samples (if they share great similarities) from real samples, which helps alleviate the mode collapse problem.
4. Historical averaging [116], in which each of the network parameters is regularised by its corresponding historical average (maintained along with the training process). This trick helps address the non-convergence problem.

Many other heuristic methods for improving GAN training can be found in [15, 116, 88].

2.3. Conditional GANs for image-to-image generation

Image-to-image generation refers to a broad category of image synthesis problems, where a generated image is conditioned on a corresponding input image. Compared with the ordinary unconditional image synthesis, the image-to-image generation problem is generally more challenging, due to the strong constraint imposed by the input images. Furthermore, for many applications of image-to-image generation, one input image could be corresponded to multiple reasonable images to be generated, which requires the modelling of one-to-many mapping (or multimodality).

The image-conditioned conditional GAN architecture [90] and its variants have been successfully applied in many image-to-image generation tasks, most of which are to find a mapping function that translates images from source domain into ones from target domain (i.e. image-to-image translation [59]). These approaches can be further divided into two categories: 1) supervised frameworks, where paired input and output images (i.e. drawn from a joint distribution) are required, and

2) unsupervised frameworks, where only unpaired images from two domains (i.e. drawn from two marginal distributions) are given, resulting in a more difficult image-to-image generation problem.

In this section, we first present several prominent conditional GAN-based frameworks for image-to-image generation, in both supervised and unsupervised settings. We then briefly introduce typical applications of image-to-image generation (using conditional GAN frameworks) that exhibit generative creativity.

2.3.1. Supervised frameworks

PLDT

The first conditional GAN-based architecture proposed for supervised image-to-image generation is the PLDT framework [153], which consists of a generator G and two discriminators D_r and D_d. G is an autoencoder network [51] that takes as input an image x_s from source domain and transfers it into target domain as $\hat{\mathrm{x}}_t$. D_r discriminates between real and generated images via an adversarial real/fake loss $\mathcal{L}_r$, and thus enforces $\hat{\mathrm{x}}_t$ to be realistic. D_d refers to a domain discriminator, which takes as input a pair of source image and target image (selected from generated image $\hat{\mathrm{x}}_t$, ground truth paired image x_t and irrelevant image $\bar{\mathrm{x}}_t$), and determines whether the images are associated via an adversarial domain loss $\mathcal{L}_d$. D_d therefore ensures $\hat{\mathrm{x}}_t$ to be related to x_s. The schema of the PLDT framework is demonstrated in Figure 2.11. The limitations of PLDT include 1) the generated images are of relatively low quality, and 2) the training process is complicated due to the use of two discriminators, and 3) it is unable to generate diverse results given one input image.

Pix2Pix

Pix2Pix [59] is a simple but effective conditional GAN framework for supervised image-to-image generation. The schema of Pix2Pix is pre-

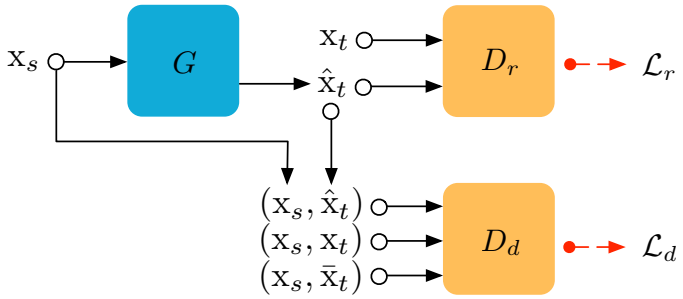


Figure 2.11.: Schema of the PLDT architecture. It is composed by a generator G, a real/fake discriminator D_r and a domain discriminator D_d. The corresponding two adversarial losses $\mathcal{L}_r$ and $\mathcal{L}_d$ respectively ensures the generated image $\hat{x}_t$ to be realistic and related to the input source image x_s.

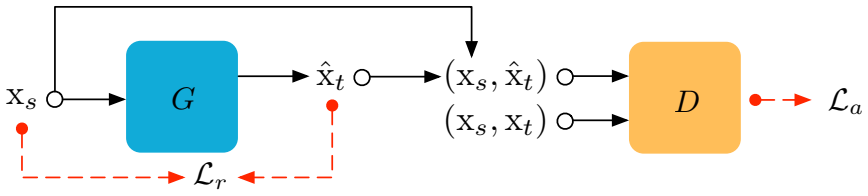


Figure 2.12.: Schema of the Pix2Pix architecture. It consists of one generator G and one discriminator D, with an adversarial loss $\mathcal{L}_a$ and a regression loss $\mathcal{L}_r$ (l_1-norm).

sented in Figure 2.12. Compared with PLDT, Pix2Pix has four major differences. First, the U-Net architecture [112] (i.e. an autoencoder with skip connections between corresponding layers in encoder and decoder) is adopted in the generator G, which allows low-level information to be shared throughout the network. Second, only one discriminator D is employed to distinguish real source-target image pairs (x_s, x_t) from pairs of real source image and generated target image (x_s, $\hat{x}_t$). Third, G is constrained by an additional regression loss $\mathcal{L}_r$ (l_1-norm), such that $\hat{x}_t$ is close to x_s in pixel space. Last, the final output of D is computed by averaging scores of local 70×70 patches across the input 256×256 images (i.e. the PatchGAN method [75, 59]),

which contributes to generating sharper results. One limitation of Pix2Pix is that it is still unable to produce stochastic outputs, even with dropout [128] applied in G.

BicycleGAN

The BicycleGAN framework [163] is proposed to learn many-to-many mappings for supervised image-to-image generation tasks. Given one input source image x_s, the trained generator G^* is able to generate diverse results by sampling different latent vectors z as input: $\hat{\mathrm{x}}_t = G^*(\mathrm{x}_s, \mathrm{z})$. In order to prevent G from ignoring z, an explicit encoder E (mapping $\hat{\mathrm{x}}_t$ back to a latent vector $\hat{\mathrm{z}}$) is jointly trained to preserve the correspondence between different z and $\hat{\mathrm{x}}_t$. Also, E is employed to directly encode the ground truth target image x_t into a more useful and relevant $\hat{\mathrm{z}}$. Specifically, BicycleGAN learns such mappings between the target and latent space in two directions ($\mathrm{x}_t \rightarrow \mathrm{z} \rightarrow \hat{\mathrm{x}}_t$ and $\mathrm{z} \rightarrow \hat{\mathrm{x}}_t \rightarrow \hat{\mathrm{z}}$), each of which is achieved by employing a specific architecture as shown in Figure 2.13. BicycleGAN is therefore a hybrid model that combines both approaches. Despite the complex training process, BicycleGAN is capable of generating more realistic and diverse results than Pix2Pix.

2.3.2. Unsupervised frameworks

DTN

The DTN framework [134] attempts to address the unsupervised image-to-image generation problem by employing an explicit encoder E pretrained on images from source domain. As presented in Figure 2.14, DTN consists of a generator G, a discriminator D and the pretrained encoder E. Specifically, E takes as input either a source image x_s or a target image x_t, and outputs the corresponding latent codes $\hat{\mathrm{z}}_s$ or $\hat{\mathrm{z}}_t$. G then generates either a translated image $\hat{\mathrm{x}}_t$ (from $\hat{\mathrm{z}}_s$) or a reconstructed image $\tilde{\mathrm{x}}_t$ (from $\hat{\mathrm{z}}_t$). Three loss functions are used in DTN: 1)

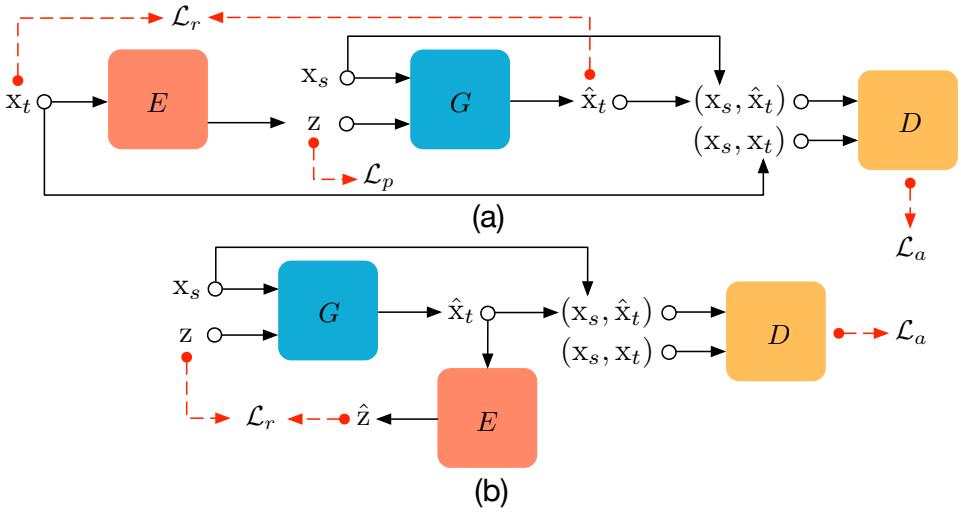


Figure 2.13.: Schema of the BicycleGAN architecture. (a) The framework for learning $x_t \rightarrow z \rightarrow \hat{x}_t$, with an adversarial loss $\mathcal{L}_a$, a pixel-wise reconstruction loss $\mathcal{L}_r$ and a prior loss $\mathcal{L}_p$. (b) The framework for learning $z \rightarrow \hat{x}_t \rightarrow \hat{z}$, with an adversarial loss $\mathcal{L}_a$ and a reconstruction loss $\mathcal{L}_r$ in latent space. BicycleGAN is a hybrid model that combines both of the frameworks, which encourages the connection between the target and latent space in two directions, in order to achieve diverse and realistic results for supervised image-to-image generation tasks.

an adversarial loss $\mathcal{L}_a$ imposed by D, which encourages both $\hat{x}_t$ and $\tilde{x}_t$ to be indistinguishable from real target images, and 2) a feature-wise constancy loss $\mathcal{L}_c$, which enforces $\hat{x}_t$ to preserve high-level features (e.g. complex shapes) of x_s, and 3) a pixel-wise identity loss $\mathcal{L}_i$, which encourages G to maintain the low-level features (e.g. colors and textures) of the input image. DTN is able to yield appealing images, but it has two limitations: 1) it handles the source and target domain differently, so that the translation can only be performed in one direction, and 2) it cannot produce diverse results for one input image.

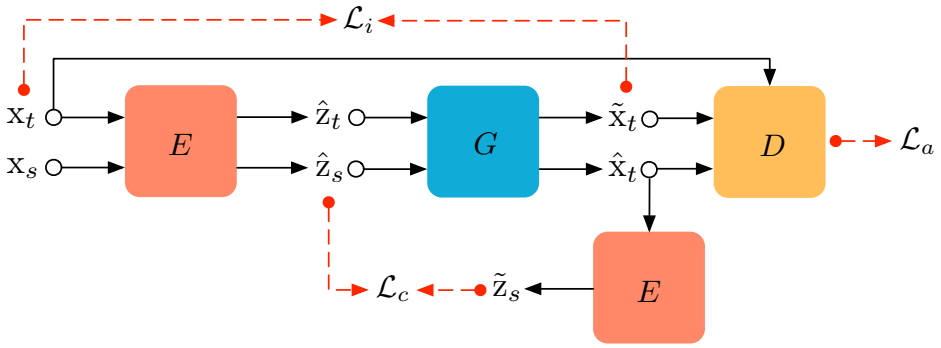


Figure 2.14.: Schema of the DTN architecture. It is made up of a (pretrained) encoder E, a generator G and a discriminator D, with an adversarial loss $\mathcal{L}_a$, a (feature-wise) constancy loss $\mathcal{L}_c$ (l_2-norm) and a (pixel-wise) identity loss $\mathcal{L}_i$ (l_2-norm).

UNIT

The UNIT framework [78] for unsupervised image-to-image generation is based on a shared latent space assumption, which is used in CoGAN framework [79] for learning a joint distribution of images from different domains. Specifically, the assumption states that a pair of images from source and target domain $(\mathrm{x}_s, \mathrm{x}_t)$ can be mapped into a same latent variable z in a shared latent space [79, 78]. Therefore, there exist encoding functions (E_s^* and E_t^*) and generating functions (G_s^* and G_t^*), such that $\mathrm{z} = E_s^*(\mathrm{x}_s) = E_t^*(\mathrm{x}_t)$, $\mathrm{x}_s = G_s^*(\mathrm{z})$ and $\mathrm{x}_t = G_t^*(\mathrm{z})$. The image-to-image generation can then be performed by $\mathrm{x}_s = G_s^*(E_t^*(\mathrm{x}_t))$ and $\mathrm{x}_t = G_t^*(E_s^*(\mathrm{x}_s))$.

The UNIT framework is made up of two generators G_s and G_t, two encoders E_s and E_t and two discriminators D_s and D_t, all of which are trained jointly. Figure 2.15 presents the schema of UNIT. For brevity, we only show the training process for one direction of generation $\mathrm{x}_s \rightarrow \hat{\mathrm{x}}_t$ (the opposite direction follows similarly). Four different loss functions are employed in UNIT: 1) a prior loss $\mathcal{L}_p$, which enforces the distribution of z to be close to a prior distribution that allows easy

sampling, and 2) an adversarial loss $\mathcal{L}_a$, which encourages $\hat{\mathrm{x}}_t$ to resemble real images in target domain, and 3) a reconstruction loss $\mathcal{L}_r$, which ensures x_s can be reconstructed via $G_s(E_s(\mathrm{x}_s))$, and 4) a cycle-consistency loss $\mathcal{L}_c$, which imposes a cycle-consistency constraint, i.e. x_s can be further reconstructed by $G_s(E_t(G_t(E_s(\mathrm{x}_s))))$. UNIT further employs a weight-sharing constraint, in which parameters of last few layers of E_s and E_t (for extracting high-level features) are shared, and parameters of first few layers of G_s and G_t (for decoding high-level features) are shared, as a practical way to implement the shared latent space assumption. The UNIT framework is able to produce images of high quality, but it is difficult to train and unable to produce diverse results.

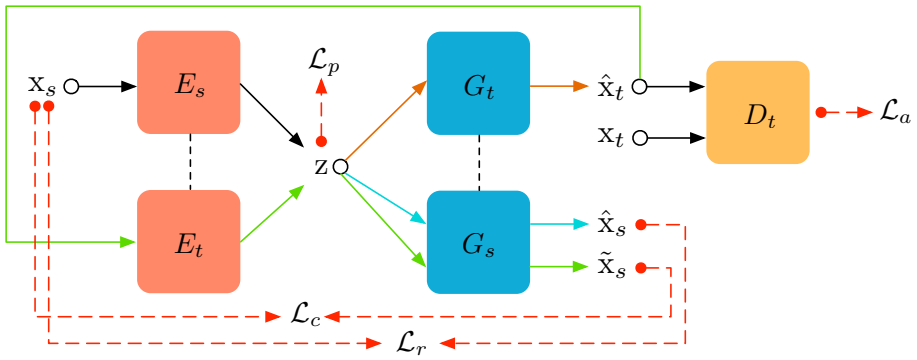


Figure 2.15.: Schema of the UNIT architecture. Only one direction of generation ($\mathrm{x}_s \rightarrow \hat{\mathrm{x}}_t$) is presented for brevity. It consists of two generators G_s and G_t, two encoders E_s and E_t and two discriminators D_s (not shown) and D_t, with a prior loss $\mathcal{L}_p$, an adversarial loss $\mathcal{L}_a$, a reconstruction loss $\mathcal{L}_r$ and a cycle-consistency loss $\mathcal{L}_c$. A weight-sharing constraint is used in high-level layers of the encoders and the generators (indicated by the dashed lines).

CycleGAN

The CycleGAN framework [162, 65, 152] can be regarded as a simplified version of the UNIT framework (without the shared latent space as-

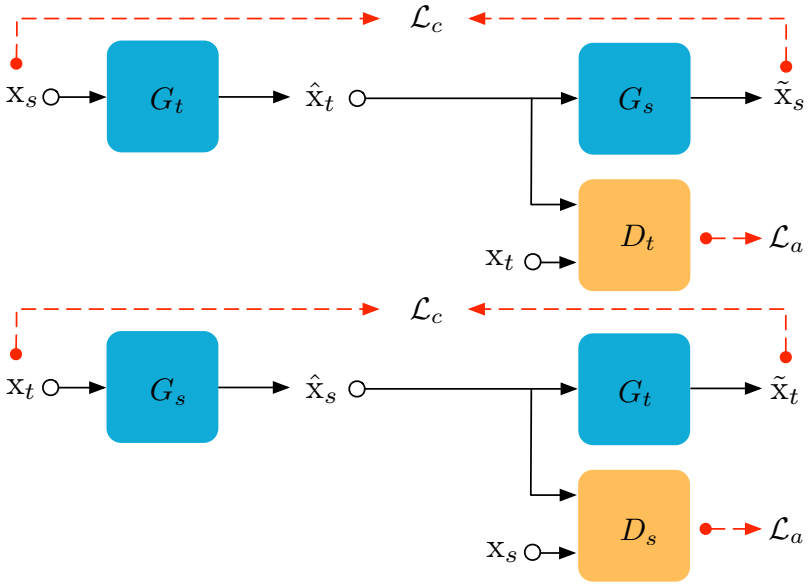


Figure 2.16.: Schema of the CycleGAN architecture. It is composed by two generators (G_s and G_t) and two discriminators (D_s and D_t), with an adversarial loss $\mathcal{L}_a$ and a cycle-consistency loss $\mathcal{L}_c$. All the networks are trained jointly.

sumption). As demonstrated in Figure 2.16, CycleGAN consists of two generators (G_s and G_t) and two discriminators (D_s and D_t), without the use of any explicit encoders. All the networks are trained simultaneously, which enables the generation of two directions ($\hat{\mathrm{x}}_t = G_t(\mathrm{x}_s)$ and $\hat{\mathrm{x}}_s = G_s(\mathrm{x}_t)$), given a source image x_s and a target image x_t. An adversarial loss $\mathcal{L}_a$ (imposed by D_s and D_t) is used to ensure that $\hat{\mathrm{x}}_t$ (and $\hat{\mathrm{x}}_s$) are indistinguishable from real target images (and real source images). A cycle-consistency loss $\mathcal{L}_c$ is further applied to reduce the possible space of the generative functions, which enforces the generated images to be relevant to the input images. It is implemented by minimising the l_1 distance between the input image and the reconstructed image, such that $\mathrm{x}_s \approx \tilde{\mathrm{x}}_s = G_s(G_t(\mathrm{x}_s))$ and $\mathrm{x}_t \approx \tilde{\mathrm{x}}_t = G_t(G_s(\mathrm{x}_t))$. Despite its simplicity, the CycleGAN framework is an effective approach to generating compelling images for various tasks of unsupervised image-to-image generation. However, similar to UNIT, Cycle-

GAN is also unable to produce diverse results.

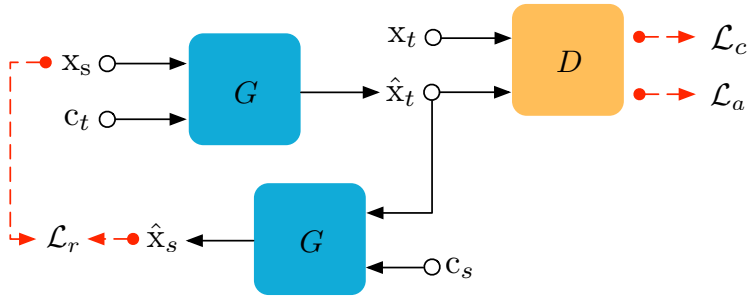


Figure 2.17.: Schema of the StarGAN architecture. It is proposed for multi-domain unsupervised image-to-image generation. It consists of one generator G and one discriminator D, with an adversarial loss $\mathcal{L}_a$, a domain classification loss $\mathcal{L}_c$ (similar to AC-GAN [95]) and a reconstruction loss $\mathcal{L}_r$ (similar to CycleGAN [162, 65, 152], but using the same generator).

StarGAN

The aforementioned frameworks are limited in scalability in multi-domain image-to-image generation tasks, in that two generators are required for each pair of image domains. The StarGAN framework [17] addresses this limitation: it is capable of performing unsupervised image-to-image generation for multiple domains using one single generator. Figure 2.17 presents the schema of the StarGAN architecture, which only consists of one generator G and one discriminator D. Specifically, G takes as input an image x_s (from one of the source domains) and a domain label c_t (for one of the target domains), and generates an image $\hat{x}_t$ of the target domain. D learns to both distinguish real images x_t from $\hat{x}_t$ via the adversarial loss $\mathcal{L}_a$, and classify the input image to its corresponding correct class of domain via the domain classification loss $\mathcal{L}_c$ (similar to AC-GAN [95]). To further ensure $\hat{x}_t$ is related to x_s, a reconstruction loss $\mathcal{L}_r$ is applied, which minimises the l_1 distance between x_s and $\hat{x}_s = G(G(x_s, c_t), c_s)$. It is

worth mentioning that the same generator G is used in this reconstruction process, unlike CycleGAN. StarGAN is not only scalable, but also able to generate images of high quality.

2.3.3. Typical applications of image-to-image generative creativity

Conditional GAN based architectures have been applied for various image-to-image generation tasks and have achieved impressive results. Here we briefly introduce some of the most popular ones that exhibit enrichment creativity, transformative creativity or combinatorial creativity.

Enrichment creativity

Image-to-image generation applications that demonstrate enrichment creativity involve the synthesis of images by adding new representations to input images, while maintaining most representations of them. Conditional GAN based frameworks have been applied in the following tasks.

- Image inpainting [99, 147, 151, 57], in which the input images have missing regions to be filled in, resulting in the completed output images (Figure 2.18 (a)).
- Image super resolution [74, 146, 133], which refers to the generation of high-resolution images from their low-resolution counterparts (Figure 2.18 (b)).
- Sketch-to-photo synthesis [117, 152, 59], in which the input images are sketches and the generated images are realistic photos (Figure 2.18 (c)).
- Image colorization [59, 12], in which the input grayscale images are to be transformed into the generated colour images (Figure 2.18 (d)).

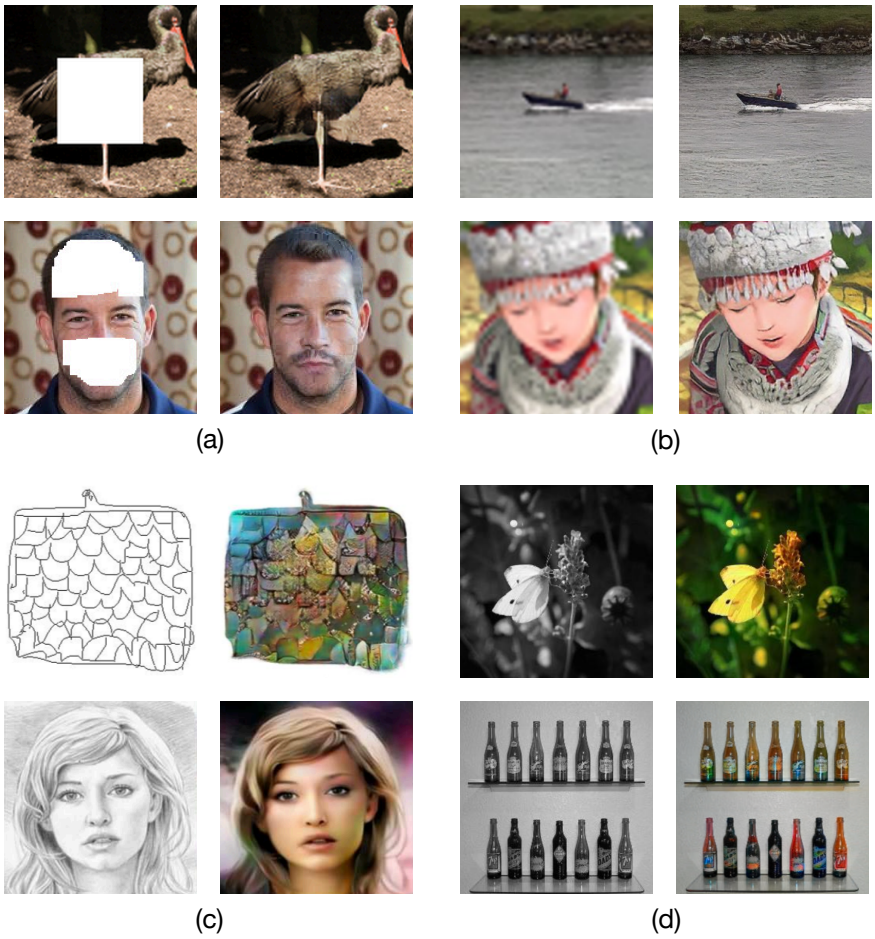


Figure 2.18.: Examples of image-to-image generation tasks that exhibit enrichment creativity. Conditional GAN frameworks have been applied in these tasks. In each subplot, input images are on the left and generated images are on the right. (a) Image inpainting. (b) Image super resolution. (c) Sketch-to-photo synthesis. (d) Image colorization. Images reproduced from [99, 57, 74, 117, 59].

Transformative creativity

Transformative creativity can be demonstrated in image-to-image generation tasks, where representations of input images are partially transformed into other ones. The generated images are still strongly related

to the input images. Conditional GAN architectures have been employed in such tasks as follows.

- Facial attribute and expression transfer [101, 3, 121, 65, 78, 17], which involves transforming the attributes (e.g. age, gender, hair colour, skin colour, etc.) or the expressions (e.g. angry, happy, fearful, sad, etc.) of a given facial image into another one (Figure 2.19 (a)).
- User-guided image manipulation [161, 11], in which the input images are edited according to the constraints (e.g. colour, shape, etc.) imposed by users (Figure 2.19 (b)).
- Other various image-to-image translation tasks [59, 162, 78, 163, 56], such as season transfer (e.g. from winter to summer), object transfiguration (e.g. from apple to orange) and daylight transfer (e.g. from night to day) (Figure 2.19 (c)).

Combinatorial creativity

Combinatorial creativity in the context of image-to-image generation refers to the combination of salient features of an input image and other images, resulting in generated images of high-level creativity. There are fewer applications that can exhibit combinatorial creativity, compared with enrichment creativity and transformative creativity. One good example is image style transfer [30, 50], in which a novel image can be synthesised by combining semantic content of a given input image with style of another image. A variety of CNN based approaches [36, 62, 137, 28, 54] for this task (termed neural style transfer) have been proposed in recent years. The conditional GAN architecture has also been applied in this task and has obtained compelling results [75] (Figure 2.20). For a more comprehensive review of neural style transfer, please refer to [61].

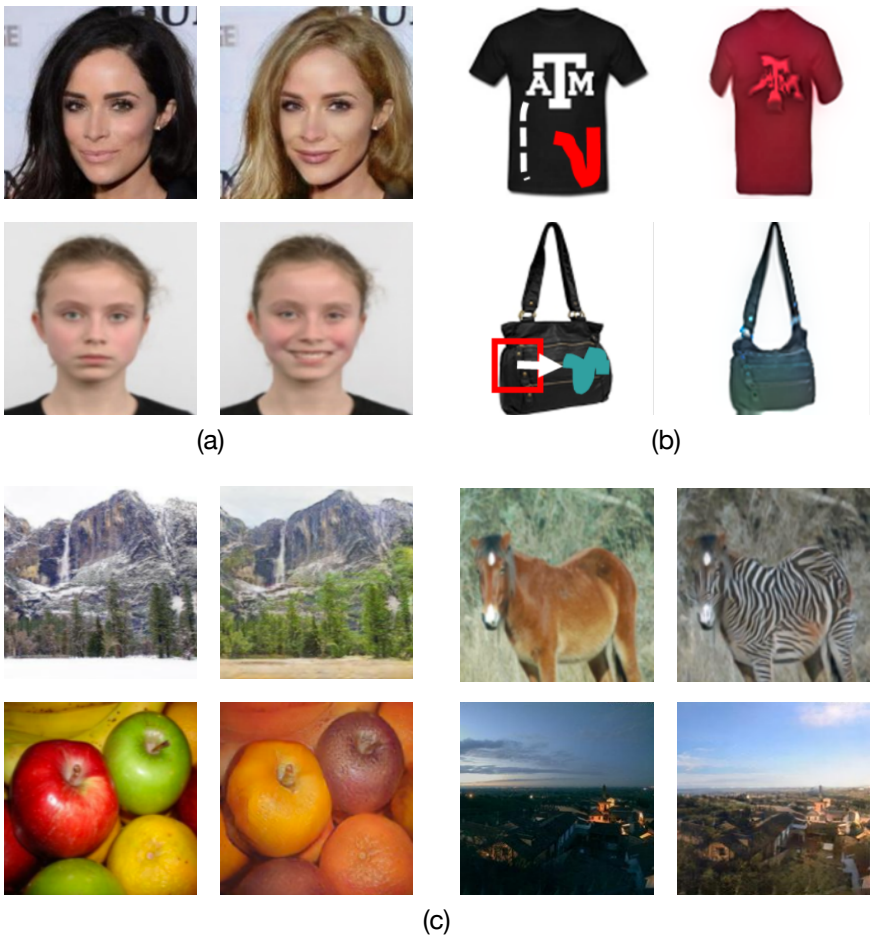


Figure 2.19.: Examples of image-to-image generation tasks that exhibit transformative creativity. Conditional GAN frameworks have been applied in these tasks. In each subplot, input images are on the left and generated images are on the right. (a) Facial attribute and expression transfer. (b) User-guided image manipulation. (c) Various image translation tasks, including season transfer, object transfiguration and daylight transfer. Images reproduced from [17, 161, 162, 163].

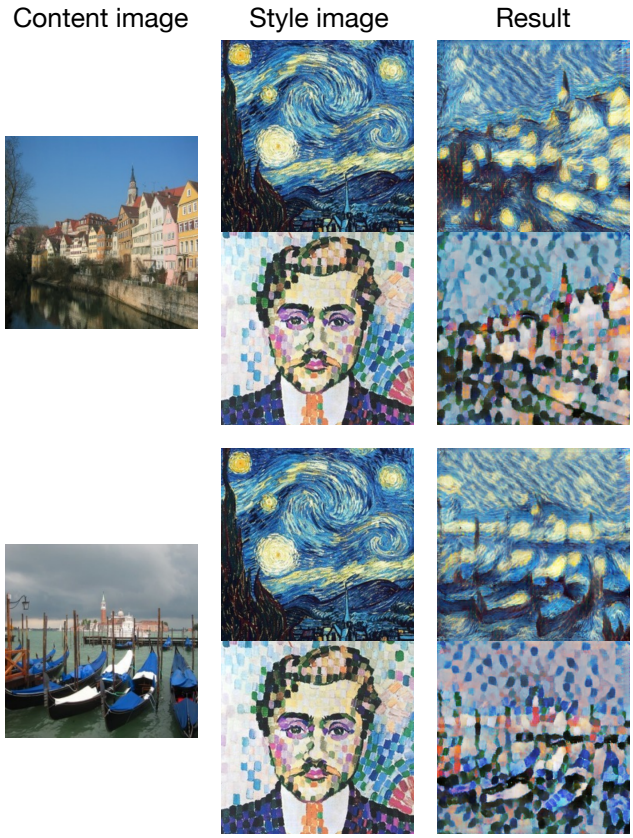


Figure 2.20.: Neural style transfer as an example of image-to-image generation tasks that exhibit combinatorial creativity. Conditional GAN frameworks have been applied in this task. Image from [75].

3. Learning to Achieve Enrichment Creativity

3.1. Motivation

In the context of image-to-image generation, a straightforward way to produce a novel and useful output image is to inject task-specific representations (e.g. colour representations for image colorization [160]) directly into an input image. This process of generation demonstrates enrichment creativity, because the generated image can be seen as a result of enriching details of the original image while maintaining the main underlying structure of it, in a novel manner.

In this chapter, we study how deep conditional adversarial networks can learn to achieve enrichment creativity from challenging image-to-image generation tasks. Specifically, we propose EnrichGAN, a novel conditional generative adversarial network (GAN) based architecture for fast compressed sensing magnetic resonance imaging (CS-MRI) reconstruction, which is a challenging application that exhibits enrichment creativity. Magnetic resonance imaging (MRI) is a widely used modality of medical imaging for many clinical applications. The process of MRI acquisition is inherently slow. CS-MRI enables fast acquisition, which is of great importance in easing patient's burden and reducing cost of scanning. Our proposed approach is to address a main challenge of CS-MRI, which is the generation of high-quality images from low-quality ones that are reconstructed from the highly undersampled data in the spatial frequency domain (k-space). Figure 3.1 demonstrates an example of CS-MRI reconstruction.

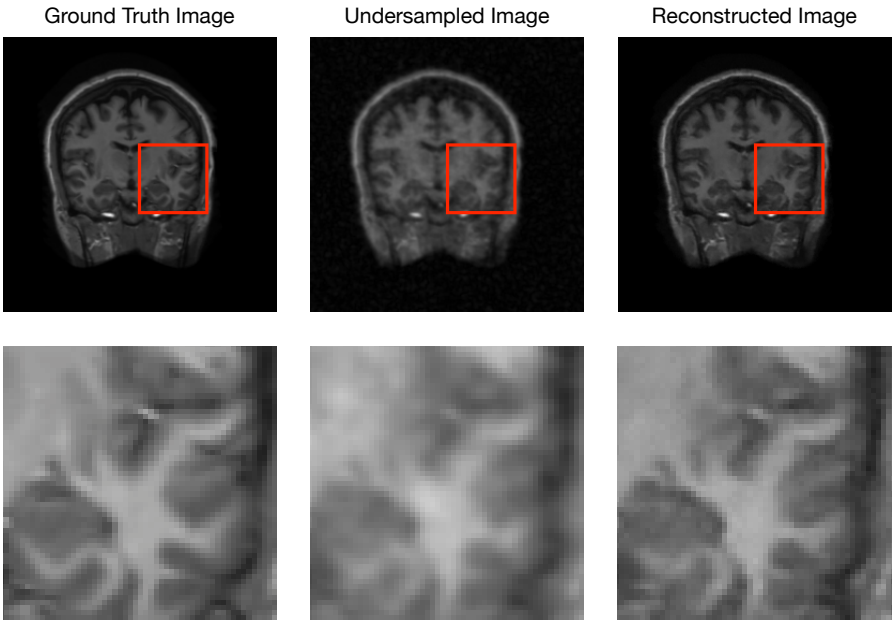


Figure 3.1.: An example of CS-MRI reconstruction that demonstrates enrichment creativity. (*top left*) The ground truth image. (*top middle*) The undersampled (20% remaining in k-space) image. (*top right*) The reconstructed image by our model taking as input the undersampled image. (*bottom*) Visualisation of details of the corresponding images. Our reconstructed image enriches the details to match those of the ground truth image.

The reasons why we focus on CS-MRI as an application of enrichment creativity are twofold. First, deep learning based CS-MRI reconstruction essentially can be viewed as an image-to-image generation problem that involves enrichment of missing details, which is analogous to image super resolution [62, 74]. Such tasks are inherently ideal examples to show enrichment creativity. Compared with image super resolution that is mainly focused on natural images, CS-MRI reconstruction is for critical medical images only, which makes it a more challenging generation task due to limited searching space (for matching the unique and crucial ground truth image) as well as the

requirement of reconstructing fine local details. Second, in general, evaluating the performance of models of generative creativity is always a challenging problem. CS-MRI reconstruction focuses more on the effectiveness of the generated results than the originality, which in turn makes the evaluation process much easier. This enables us to make fair and straightforward comparison between our proposed approaches and other methods.

The rest of this chapter is organised as follows. We first briefly introduce the general background and related work of CS-MRI reconstruction in Section 3.2. We then formulate the problem in Section 3.3. After that, we describe in detail our proposed methods in Section 3.4. Finally, we present and discuss the results of comprehensive experiments in Section 3.5, and conclude our work in Section 3.6.

3.2. Background

MRI is a widely-applied medical imaging technique. It measures the anatomical and functional information of tissue in a non-invasive fashion. One major disadvantage of MRI is the slow process of data acquisition. MRI data is collected in k-space, where spatial-frequency information is obtained sequentially line-by-line, and constructing a high-quality MRI image typically requires 64 to 512 lines of data. Such slow acquisition process not only gives rise to serious artefacts because of the movement and physiological motion of patients (e.g. cardiac pulsation and respiratory excursion), but also raises the scanning cost and degrades the comfort and compliance of patients [53]. In addition, in the case of applying contrast media injection, the prolonged acquisition process leads to contrast washout, which severely impairs image quality.

Conventionally, for the prescribed spatial resolution of MRI images, the required raw data in k-space is determined by the Nyquist-Shannon sampling criteria [93]. A possible way of accelerating the acquisition process is to undersample k-space data. CS-MRI is an approach that

enables fast acquisition with aggressive undersampling scheme that can break the Nyquist-Shannon sampling barrier. In theory, CS-MRI can reconstruct MRI images without deterioration of quality, by employing nonlinear optimisation algorithms on compressible and randomly undersampled raw data. There have been extensive works of CS-MRI research, mainly focusing on three directions: 1) to find random undersampling schemes, 2) to learn sparse representations, and 3) to develop nonlinear optimisation algorithms for image reconstruction. These directions correspond to the three requirements of a successful application of CS [82]. For more comprehensive reviews of classic CS-MRI methods, please refer to [82, 60].

In this work, we mainly focus on the nonlinear reconstruction algorithm for CS-MRI with the use of deep generative networks. There are several studies proposed recently on deep neural networks based CS-MRI reconstruction. Wang et al. [140] proposed a convolutional neural network (CNN) based CS-MRI reconstruction, where a trained CNN network was 1) integrated into the CS-MRI as an extra regularisation term, or 2) applied to generate a reconstruction image to initialise the ordinary CS-MRI method, resulting in a two-stage reconstruction process. Yang et al. [150] introduced ADMM-Net, a learnable deep architecture defined over a data flow graph for iterative procedures of alternating direction method of multipliers (ADMM) to optimise CS-MRI models. Schlemper et al. [120] proposed a cascade CNN architecture for MRI reconstruction. Hammernik et al. [46] introduced a deep variational network to learn a reconstruction procedure for MRI data.

Different from these previous works, we consider CS-MRI reconstruction as a task of enriching local structures and details, and propose a conditional GAN based approach to resolve this problem. We design a novel objective function and an enrichment learning process to achieve better reconstructed details, which leads to superior reconstruction results. Furthermore, unlike the aforementioned works that involve time-consuming iterative optimisations, our proposed method

performs a single feed-forward computation for reconstruction of each image at inference stage, which dramatically reduces the processing time to a few milliseconds and thus enables real-time applications.

3.3. Problem formulation

3.3.1. Forward and inverse model

The general forward model of image reconstruction can be approximated as a linear system [44]:

$$\mathrm{y} = \mathrm{F_u x} + \varepsilon, \tag{3.1}$$

where $\mathrm{x} \in \mathbb{C}^N$ represents a complex-valued image (of $\sqrt{N} \times \sqrt{N}$ pixels and formatted as a column vector) to be reconstructed, $\mathrm{y} \in \mathbb{C}^M$ denotes the (degraded) observation vector, $\mathrm{F}_u \in \mathbb{C}^{M \times N}$ represents a linear operator $\mathrm{F}_u : \mathbb{C}^N \mapsto \mathbb{C}^M$ that defines the forward model for the observation, and ε denotes the additive noise. In the case of CS-MRI reconstruction, F_u is the undersampled Fourier operator in k-space.

The inverse model for reconstructing an estimate of x is typically ill-posed [44], in that the problem is usually underdetermined with $M \ll N$. Moreover, the inverse problem could be unstable because of the numerically ill-conditioned operator F_u and the noise ε.

3.3.2. General CS-MRI reconstruction model

To address the ill-posed inverse problem of CS-MRI reconstruction, a-priori knowledge of x should be utilised, which formulates the following optimisation problem:

$$\min_{\mathrm{x}} \ \frac{1}{2}||\mathrm{F}_u\mathrm{x} - \mathrm{y}||_2^2 + \lambda \mathcal{R}(\mathrm{x}), \tag{3.2}$$

where $\frac{1}{2}||\mathrm{F}_u\mathrm{x} - \mathrm{y}||_2^2$ represents the term of observation fidelity, and λ denotes a regularisation parameter of the regularisation term $\mathcal{R}(\mathrm{x})$

that usually employs l_0-norm or l_1-norm in the sparsifying transform domain of x as a sparse prior [82].

3.3.3. Deep neural networks based CS-MRI model

Instead of using the regularisation term $\mathcal{R}(\mathrm{x})$, deep neural network based CS-MRI model employs a new regularisation term that enforces x to approximate the reconstruction image generated by a deep generative model, which formulates an optimisation problem as follows [140, 120]:

$$\min_{\mathrm{x},\theta} \ \frac{1}{2}||\mathrm{F}_u\mathrm{x} - \mathrm{y}||_2^2 + \zeta||\mathrm{x} - g(\mathrm{x}_u;\theta)||_2^2, \tag{3.3}$$

where $g(\mathrm{x}_u;\theta)$ denotes the deep neural networks based generating function parameterised by θ, x_u represents the reconstruction image from zero-filled undersampled k-space measurements, and ζ is the regularisation parameter. The generative networks can be either pretrained, or trained simultaneously within the entire optimisation process, given input image pairs of the undersampled reconstruction x_u and target ground truth image x_t.

One limitation of the optimisation problem in Equation 3.3 is that the reconstruction process of the generating function g does not take the a-priori knowledge of the data in k-space into account, as it operates in the image domain only [120]. The inclusion of data fidelity should (presumably) be able to improve the performance of the generative model. In addition, to solve this optimisation problem, either iterative optimisation for x_u (e.g. [140]) or cascaded neural networks (e.g. [120]) is required, resulting in a time-consuming optimising process. To address the limitations, we propose to employ one single deep generative network trained with the information of data fidelity for CS-MRI reconstruction, which can be formulated as follows:

$$\min_{\theta} \mathbb{E}_{(\mathrm{x}_t,\mathrm{x}_u)\sim p_{<t,u>}(\mathrm{x}_t,\mathrm{x}_u)}[\mathcal{L}(\mathrm{x}_t, g(\mathrm{x}_u;\theta);\theta)], \tag{3.4}$$

where $\mathcal{L}$ denotes the objective function containing multiple loss com-

ponents such as data fidelity term in sparsifying transform domain and reconstruction term in image domain (detailed in the next section), x_u and x_t are paired undersampled and fully-sampled (ground truth) reconstruction of MRI images sampled from the joint data distribution $p_{<t,u>}(\mathrm{x}_t, \mathrm{x}_u)$. This optimisation problem can be seen as enriching the local structures and details of x_u to match x_t in a data-driven manner. The trained generative network $g^*(\mathrm{x}_u; \theta^*)$ will be capable of creating a high-quality reconstruction via an efficient feed-forward inference, given one undersampled reconstruction of a MRI image.

MRI data encodes both magnitude and phase information in the format of complex number. There are two common strategies to deal with such complex number for deep neural networks. One strategy is to directly leverage the real and imaginary data as two separate channels for the input image x_u, i.e. $\mathbb{C}^N$ is replaced as $\mathbb{R}^{2N}$ [120]. Another strategy is to employ magnitude only, following a standard assumption that many acquisition sequence involves negligibly small phase [31]. In our work, we adopt the latter strategy to avoid extra computational burden.

3.4. Method

We leverage the idea of enrichment creativity and propose EnrichGAN, an image-conditioned conditional GAN architecture for resolving the CS-MRI reconstruction problem formulated in Equation 3.4. Figure 3.2 presents the overall architecture of EnrichGAN. It consists of a generator G as the generating function g in Equation 3.4, a discriminator D, and a pretrained feature extractor network F. G, D and F are built on multiple convonlutional layers. The generator G takes as input a zero-filling undersampled reconstruction of MRI image x_u, learns to enrich local details and structures, and generates a high-quality detail-enriched reconstructed image $\hat{\mathrm{x}}_u$. In the following sections, we first introduce the proposed novel objective function that is made up of four designated loss functions (an adversarial loss $\mathcal{L}_a$,

a pixel-wise image domain loss $\mathcal{L}_i$, a frequency domain loss $\mathcal{L}_f$ and a perceptual loss $\mathcal{L}_p$), which enables our model to generate superior results. We then describe the proposed enrichment learning process and the designed network architecture.

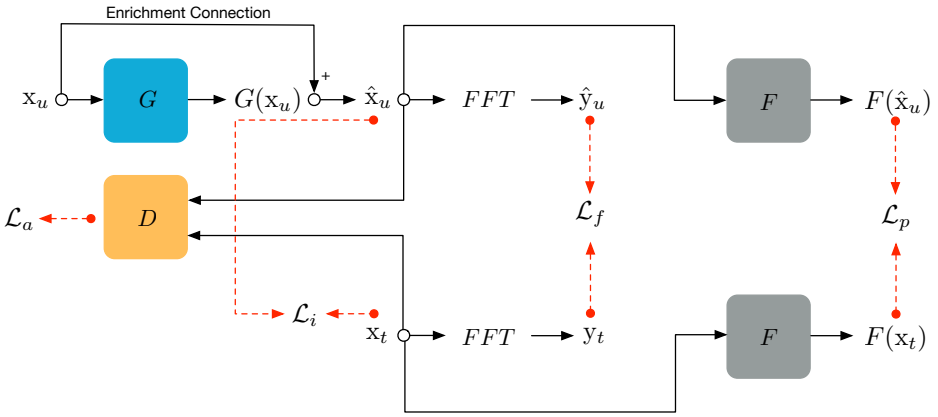


Figure 3.2.: Schema of our proposed EnrichGAN architecture for CS-MRI reconstruction. It is composed of a generator G, a discriminator D, and a feature extractor F, all built on CNN. The objective function consists of four losses, including an adversarial loss $\mathcal{L}_a$, a pixel-wise image domain loss $\mathcal{L}_i$, a frequency domain loss $\mathcal{L}_f$ and a perceptual loss $\mathcal{L}_p$. Our approach also employs an enrichment learning process with the use of the enrichment connection.

3.4.1. Objective function

Adversarial loss

We first employ an adversarial loss [41] $\mathcal{L}_a$ as follows:

$$\mathcal{L}_a(G, D) = \mathbb{E}_{x_t \sim p_t(x_t)}[\log D(x_t)] + \mathbb{E}_{x_u \sim p_u(x_u)}[\log(1 - D(G(x_u)))], \tag{3.5}$$

where the discriminator D learns to distinguish between the fully-sampled reconstruction of MRI images x_t (sampled from the distri-

bution $p_t(\mathrm{x}_t)$) and the generated ones from the generator G, while G is trained to generate indistinguishable results from x_t. The input to G is the undersampled reconstruction of MRI images x_u (sampled from the distribution $p_u(\mathrm{x}_u)$) as a condition of GAN. Unlike other proposed losses, $\mathcal{L}_a$ does not require x_u and x_t to be paired. The aim of applying $\mathcal{L}_a$ is to make the conditional generative distribution $p_G(\hat{\mathrm{x}}_u|\mathrm{x}_u)$ match the empirical distribution $p_t(\mathrm{x}_t)$ via the enrichment of local structures and details on x_u.

Pixel-wise image domain loss

Applying the adversarial loss $\mathcal{L}_a$ alone, however, cannot guarantee that the generated image $\hat{\mathrm{x}}_u = G(\mathrm{x}_u)$ shares any similarities with the input image x_u (i.e. the input image would be ignored by G). This is because G could simply map the input x_u to any random samples from $p_t(\mathrm{x}_t)$, when the network capability is large enough. This problem still exists even when the discriminator D is given paired data of input images and ground truth images, as reported in [59]. To address this problem, we further utilise a pixel-wise image domain loss $\mathcal{L}_i$ to restrict the searching space of G:

$$\mathcal{L}_i(G) = \mathbb{E}_{(\mathrm{x}_t,\mathrm{x}_u)\sim p_{<t,u>}(\mathrm{x}_t,\mathrm{x}_u)}[\frac{\|\mathrm{x}_t - G(\mathrm{x}_u)\|_2}{\|\mathrm{x}_t\|_2}], \tag{3.6}$$

in which the paired x_t and x_u are sampled from a joint distribution $p_{<t,u>}(\mathrm{x}_t, \mathrm{x}_u)$. In this work, we employ normalised (root) mean square error (NMSE) in $\mathcal{L}_i$, which is a commonly-used measurement of CS-MRI reconstruction performance defined on pixel-wise image difference [150]. The use of $\mathcal{L}_i$ (together with $\mathcal{L}_a$) will enforce G to not only generate indistinguishable images from real ones, but also to generate reconstruction images that resemble the ground truth images, which is a necessary condition of realising enrichment creativity.

Frequency domain loss

As mentioned in Section 3.3.3, the information of data fidelity is incorporated in our proposed EnrichGAN to train the generator G, by using a frequency domain loss denoted as $\mathcal{L}_f$, which can be seen as follows:

$$\begin{aligned} \mathcal{L}_f(G) &= \mathbb{E}_{(\mathrm{x}_t,\mathrm{x}_u)\sim p_{<t,u>}(\mathrm{x}_t,\mathrm{x}_u)}[\frac{1}{N}\left\|\mathrm{y}_t - \hat{\mathrm{y}}_u\right\|_2^2], \\ \mathrm{y}_t &= FFT(\mathrm{x}_t), \\ \hat{\mathrm{y}}_u &= FFT(G(\mathrm{x}_u)), \end{aligned} \tag{3.7}$$

where FFT represents the two-dimensional (2D) Fourier transform, y_t and $\hat{\mathrm{y}}_u$ represent the corresponding frequency domain data of x_t and $G(\mathrm{x}_u)$, N is the dimension of y_t and $\hat{\mathrm{y}}_u$. $\mathcal{L}_f$ provides G with an updating signal of maintaining data fidelity, which will enable G to generate not only similar images to x_t (as an additional constraint similar to what $\mathcal{L}_i$ imposes), but also smooth and coherent details, resulting in better MRI reconstruction.

Perceptual loss

Previous works [62, 25, 74] have shown that optimising loss defined by differences between high-level representations extracted from pre-trained CNN will make the reconstructed images more visually convincing. In addition, during the training process, it provides a more robust measurement of similarity between images, than using the loss defined by pixel-wise differences. In this work, we also adopt this idea of perceptual similarity and propose to use a perceptual loss $\mathcal{L}_p$ defined as below:

$$\mathcal{L}_p(G) = \mathbb{E}_{(\mathrm{x}_t,\mathrm{x}_u)\sim p_{<t,u>}(\mathrm{x}_t,\mathrm{x}_u)}[\frac{1}{M}\left\|F(\mathrm{x}_t) - F(G(\mathrm{x}_u))\right\|_2^2], \tag{3.8}$$

where F represents the feature extractor, and M is the dimension of the extracted feature maps. By utilising $\mathcal{L}_p$, the generator G is able

to enrich finer and clearer local structural details of MRI images based on x_u.

Full objective

The full objective of our proposed EnrichGAN architecture is:

$$\min_G \max_D \mathcal{L}(G, D) = \mathcal{L}_a(G, D) + \alpha \mathcal{L}_i(G) + \beta \mathcal{L}_f(G) + \gamma \mathcal{L}_p(G), \quad (3.9)$$

where α, β and γ are used to control the relative strength of $\mathcal{L}_i$, $\mathcal{L}_f$ and $\mathcal{L}_p$, respectively.

3.4.2. Enrichment learning

In order to further enforce the generator G to only generate the missing details based on the input undersampled reconstructed image x_u, we propose a method of enrichment learning by introducing an enrichment connection, which can be seen from Figure 3.2. Specifically, instead of employing the original $\hat{\mathrm{x}}_u = G(\mathrm{x}_u)$, we define $\hat{\mathrm{x}}_u$ as:

$$\hat{\mathrm{x}}_u = G(\mathrm{x}_u) + \mathrm{x}_u. \quad (3.10)$$

Enrichment learning dramatically reduces the complexity of the model required for the MRI reconstruction, because generating the missing structural details only is much easier than generating an entire output reconstructed image. Furthermore, the proposed enrichment learning will contribute to stabilising the training process of conditional GAN architecture as well as yielding fast convergence to a large degree. Essentially, the proposed process of enrichment learning demonstrates enrichment creativity.

3.4.3. Network architecture

We briefly describe the network architecture of our proposed EnrichGAN in this section. A more detailed visualisation of the architecture

can be seen in Figure B.1 in Appendix B.

The generator G

For the CS-MRI reconstruction problem, the input image and the output image share a great deal of similarity in the underlying structure, but differ in local details and structures to be enriched. It would be advantageous to propagate different levels of representation of the input image across the generator G for achieving superior enrichment of details.

To this end, we adopt an encoder-decoder based U-Net architecture [112] for the generator G [59]. The encoder is composed of eight convolutional layers, and a layer of batch normalisation [58] and leaky-ReLU [83] is applied after each convolution (except the first and last one). The decoder consists of eight transposed convolutional layers, and each is followed by a layer of batch normalisation and ReLU. Skip connections are applied to the corresponding layers of the encoder and the decoder, such that the decoder is able to receive information of different levels of input image features.

The discriminator D

The discriminator D is composed by eleven convolutional layers, and each of them is followed by a layer of batch normalisation and leaky-ReLU (except the first layer).

The feature extractor F

We employ the VGG (VGG16) network [126] pretrained on ImageNet [114] as the feature extractor F. The feature maps are extracted by the conv4 layer of the VGG network.

3.5. Experiments

3.5.1. Experimental settings

EnrichGAN and variants

In order to evaluate the effectiveness of different loss components in the proposed objective function and the process of enrichment learning, we compared variants of EnrichGAN in our experiments as follows:

1. Pixel-GAN (PG): the method with the pixel-wise image domain loss $\mathcal{L}_i$ and the adversarial loss $\mathcal{L}_a$;
2. Pixel-Perceptual-GAN (PPG): PG with the perceptual loss $\mathcal{L}_p$;
3. Pixel-Perceptual-GAN-Enrichment (PPGE): PPG with enrichment learning;
4. Pixel-Frequency-Perceptual-GAN-Enrichment (PFPGE): PPGE with the frequency domain loss $\mathcal{L}_f$, which is our full model (i.e. EnrichGAN).

Comparison methods

We compared the proposed EnrichGAN and its variants with several representative approaches of CS-MRI reconstruction, including conventional methods (TV [81], SIDWT [8] and RecPF [149]) and state-of-the-art methods (DLMRI [107], PBDW [103], PANO [104], Noiselet [100], BM3D [32] and DeepADMM [150]). For a fair comparison, all methods employed baseline zero-filling (ZF) undersampled reconstruction as initial input.

Evaluation metrics

For quantitative evaluation, we report NMSE, peak signal-to-noise ratio (PSNR in dB) and structural similarity index (SSIM) [142], which are commonly used metrics for measuring the quality of CS-MRI reconstruction. Given a fully-sampled reconstruction of MRI image x_t as ground truth (GT) and a detail-enriched reconstructed image $\hat{\mathrm{x}}_u$,

PSNR is defined as:

$$\mathrm{PSNR}(\mathrm{x}_t, \hat{\mathrm{x}}_u) = 10\log_{10}(\frac{[\max(\mathrm{x}_t)]^2}{\frac{1}{N}\left\|\mathrm{x}_t - \hat{\mathrm{x}}_u\right\|_2^2}), \tag{3.11}$$

where $\max(\mathrm{x}_t)$ computes the maximum pixel value of x_t, N is the dimension of x_t and $\hat{\mathrm{x}}_u$. SSIM is defined as:

$$\mathrm{SSIM}(\mathrm{x}_t, \hat{\mathrm{x}}_u) = \frac{(2\mu_{\mathrm{x}_t}\mu_{\hat{\mathrm{x}}_u} + C_1)(2\sigma_{\mathrm{x}_t\hat{\mathrm{x}}_u} + C_2)}{(\mu_{\mathrm{x}_t}^2 + \mu_{\hat{\mathrm{x}}_u}^2 + C_1)(\sigma_{\mathrm{x}_t}^2 + \sigma_{\hat{\mathrm{x}}_u}^2 + C_2)}, \tag{3.12}$$

where μ_{x_t} and $\mu_{\hat{\mathrm{x}}_u}$ represent the average of x_t and x_u, σ_{x_t} and $\sigma_{\hat{\mathrm{x}}_u}$ represent the variance of x_t and x_u, $\sigma_{\mathrm{x}_t\hat{\mathrm{x}}_u}$ represents the covariance of x_t and x_u, C_1 and C_2 are regularisation constants to avoid instability of certain image regions where local mean or variance is near zero [142]. Unlike NMSE and PSNR that estimate the absolute error of image pixels, SSIM is a perception-based method that evaluates image quality by taking the impact of luminance, contrast and structure into account.

Datasets

We used a dataset from MICCAI 2013 grand challenge[1] to evaluate our model. More specifically, 100 T1-weighted brain MRI datasets were randomly selected for training (70%) and validation (30%), which contain 16095 and 5033 valid 2D images. Another 50 datasets (containing 9854 valid 2D images) were randomly selected for testing. We further randomly chose 50 2D images for comparing our proposed approach with the aforementioned representative CS-MRI reconstruction methods. Furthermore, in order to evaluate the diagnostic value of our EnrichGAN model, the trained model was used to perform zero-shot inference [35] on pathological MRI images, including cardiac MRI images with atrial scarring and brain lesion MRI images[2]. For more details of the datasets please refer to our project page[3].

[1] `https://my.vanderbilt.edu/masi/workshops/`
[2] `http://www.brainlesion-workshop.org/`
[3] `https://github.com/nebulaV/DAGAN`

Random undersampling masks

We used three categories of undersampling patterns in the experiments: 1D Gaussian (G1D), 2D Gaussian (G2D) and 2D Poisson disc (P2D). For the G1D mask, 10%, 20%, 30%, 40% and 50% retained k-space data were fabricated as the simulation of different accelerating ratios of MRI acquisition, respectively 10×, 5×, 3.3×, 2.5× and 2×. For G2D and P2D masks, 30% retained k-space data were generated. For visualisation of the random undersampling masks, please refer to Figure B.2 in Appendix B.

Data augmentation

We performed a number of data augmentation methods to generate additional training data for improving the performance of our model, including (random) rotating, shifting, flipping, zooming, brightness adjustment and elastic distortion [125] (a way of producing shape variations).

Training details

For the weights of losses in Equation 3.9, we balanced the magnitude of different loss terms into roughly similar scales and accordingly set $\alpha = 15$, $\beta = 0.1$, $\gamma = 0.0025$. We used the Adam optimiser [66] with a batch size of 25. The initial learning rate was set to 0.0001 and was halved every 5 epochs. All our models were trained with early stopping strategy as a way of regularisation to avoid overfitting [13]. We used the same set of hyperparameters in all experiments to demonstrate the robustness of our EnrichGAN models that require minimal parameter tuning. We implemented our models using TensorFlow [1] and TensorLayer [23].

3.5.2. Results

Comparison of EnrichGAN and variants

The quantitative comparison results of EnrichGAN and variants (i.e. PG, PPG, PPGE, PFPGE) can be seen in Table 3.1. Overall, all our models are able to generate significantly better results than those of the baseline ZF reconstruction. Specifically, our full model (PFPGE) obtains the best quantitative results of both NMSE and PSNR, for all ratios of the random undersampling 1D Gaussian mask. Also, compared with PG and PPG, PPGE achieves comparable results for less than 30% retained data and much better results for 40% and 50% retained data. This demonstrates the effectiveness of the proposed enrichment learning approach, especially for the higher ratio of retained data, in which input and GT MRI images become more similar, and thus enrichment of fine details and local structures become a more critical factor for generating more superior results.

Moreover, it can be found in Table 3.1 that the use of the perceptual loss does not improve the quantitative performance of PPG, compared with PG. However, employing the perceptual loss contributes to better generation with clearer and finer details and less blurring, as can be seen from Figure 3.5 (l) vs. (m), (n) and (o). This indicates the importance of utilising representation-based constraints on the model for generating images with perceptually convincing details.

Figure 3.3 demonstrates examples of generated MRI images using our full EnrichGAN model (PFPGE) from different ratios of retained raw data. When the undersampling ratio is larger than 20%, the enriched local structural details of the generated images are very close to those of GT, as can be seen in the zoomed-in ROIs and their difference images. However, for 10% retained data, the generated image has obvious difference of local details and structures (e.g. edges of organ) from the GT. This is because the highly undersampled k-space data leads to substantial loss of information. As a consequence, the input image does not incorporate critical representations of underlying

Table 3.1.: Quantitative results (NMSE and PSNR) of our proposed EnrichGAN model (with variants) and the comparison methods, using 1D Gaussian undersampling masks of different ratios.

Method	10%		20%		30%		40%		50%		Testing Time (30%)
	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	CPU (s) / GPU (ms)
ZF	0.37±0.09	28.22±4.30	0.21±0.06	34.04±4.58	0.16±0.04	35.61±4.79	0.14±0.04	37.65±4.92	0.10±0.03	40.65±4.95	**0.002±0.003** / —
TV	0.37±0.09	28.23±4.30	0.20±0.06	34.09±4.57	0.16±0.04	35.68±4.79	0.13±0.04	37.81±4.92	0.09±0.03	40.90±4.95	10.5±1.2 / —
SIDWT	0.36±0.10	28.96±4.79	0.15±0.04	36.99±5.34	0.11±0.03	38.93±5.45	0.10±0.03	40.51±6.31	0.07±0.02	43.53±6.26	126.0±22.7 / —
RecPF	0.36±0.08	28.62±4.46	0.13±0.02	36.21±5.57	0.11±0.02	37.88±4.90	0.09±0.03	41.37±5.88	0.06±0.02	44.78±6.12	35.9±0.8 / —
DLMRI	0.39±0.10	28.25±4.24	0.16±0.03	36.54±4.21	0.12±0.02	38.24±4.29	0.09±0.02	42.11±4.17	0.06±0.01	45.40±4.08	49.1±9.8 / —
PBDW	0.21±0.05	31.14±3.65	0.11±0.04	37.56±3.07	0.10±0.04	38.07±3.02	0.07±0.02	41.91±2.95	**0.04±0.01**	46.05±3.05	223.5±45.5 / —
PANO	0.22±0.05	30.84±4.48	**0.09±0.02**	38.54±4.04	**0.08±0.02**	39.16±4.02	**0.05±0.01**	43.05±3.64	**0.04±0.01**	45.44±3.42	108.6±17.4 / —
Noiselet	0.25±0.07	30.19±5.40	0.13±0.04	36.97±5.94	0.10±0.03	39.01±5.96	0.08±0.03	41.51±6.84	0.06±0.02	44.44±6.79	31.1±0.5 / —
BM3D	0.20±0.05	31.72±5.23	0.10±0.02	39.40±4.58	0.09±0.02	39.93±4.59	0.07±0.01	43.66±4.48	0.05±0.01	45.89±4.41	87.5±3.0 / —
DeepADMM	0.24±0.03	30.70±4.78	0.16±0.09	37.31±3.07	0.15±0.04	37.36±3.27	0.10±0.06	42.36±2.41	0.09±0.04	42.60±2.60	3.2±0.2 / —
PG	0.19±0.04	32.82±3.80	**0.09±0.02**	39.10±3.84	0.09±0.02	39.72±4.30	0.06±0.01	43.28±4.17	0.05±0.01	44.39±4.18	0.2±0.1 / 5.4±0.1
PPG	0.23±0.06	31.33±2.81	0.10±0.03	38.88±3.41	0.09±0.02	39.78±3.95	0.07±0.03	42.31±2.98	0.05±0.01	45.61±3.73	0.2±0.1 / 5.4±0.1
PPGE	0.19±0.04	32.84±4.46	0.10±0.02	39.05±3.82	0.09±0.02	39.53±4.12	**0.05±0.01**	44.45±4.14	**0.04±0.01**	47.30±3.93	0.2±0.1 / 5.4±0.1
PFPGE	**0.17±0.03**	**33.79±4.26**	**0.09±0.02**	**39.44±3.95**	**0.08±0.02**	**40.20±4.07**	**0.05±0.01**	**44.83±4.16**	**0.04±0.01**	**47.83±4.10**	0.2±0.1 / **5.4±0.1**

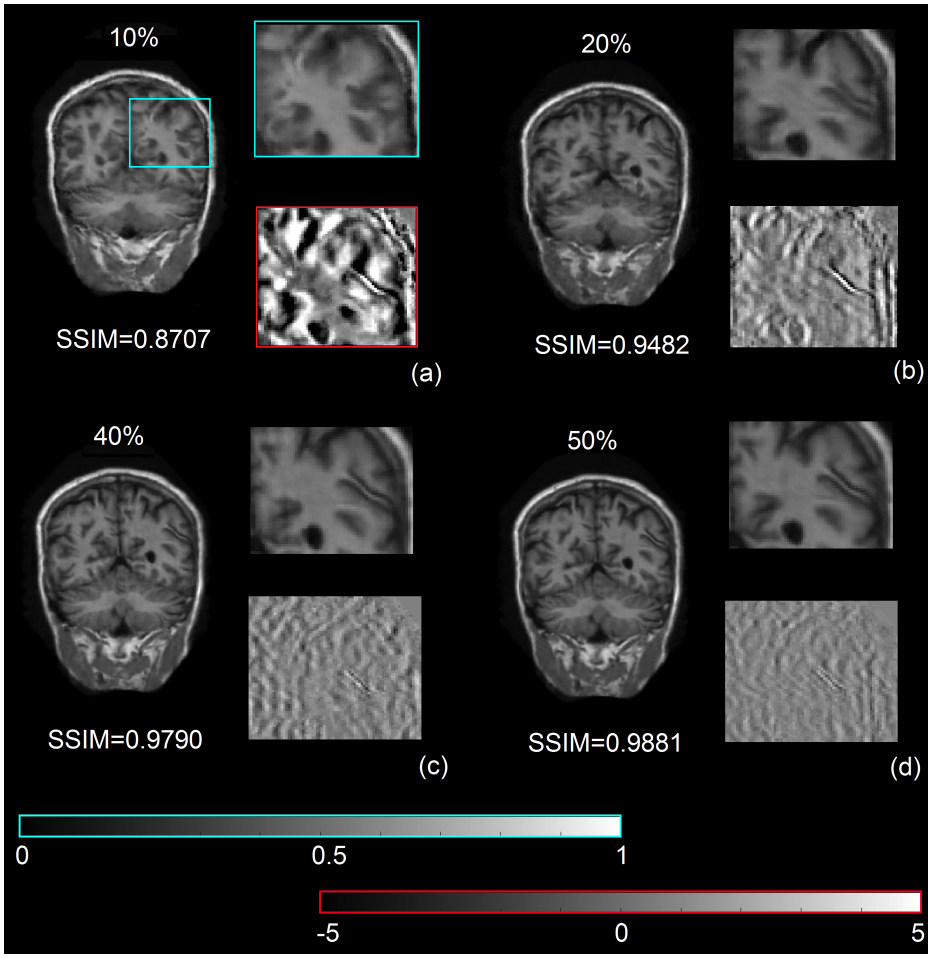


Figure 3.3.: Qualitative results of PFPGE using 1D Gaussian undersampling masks with different ratios of retained data: (a) 10%, (b) 20%, (c) 40%, (d) 40%. Results for 30% retained data and GT can be found respectively in Figure 3.5 (a) and (o). The cyan frame in (a) presents the zoomed-in ROI, and the red frame presents the difference image (50×) between the generated image and the GT. Corresponding zoomed-in ROIs are shown in (b-d). Bottom left and right present the colour bar for images and difference images respectively.

structures from which our model could enrich fine details to generate decent results.

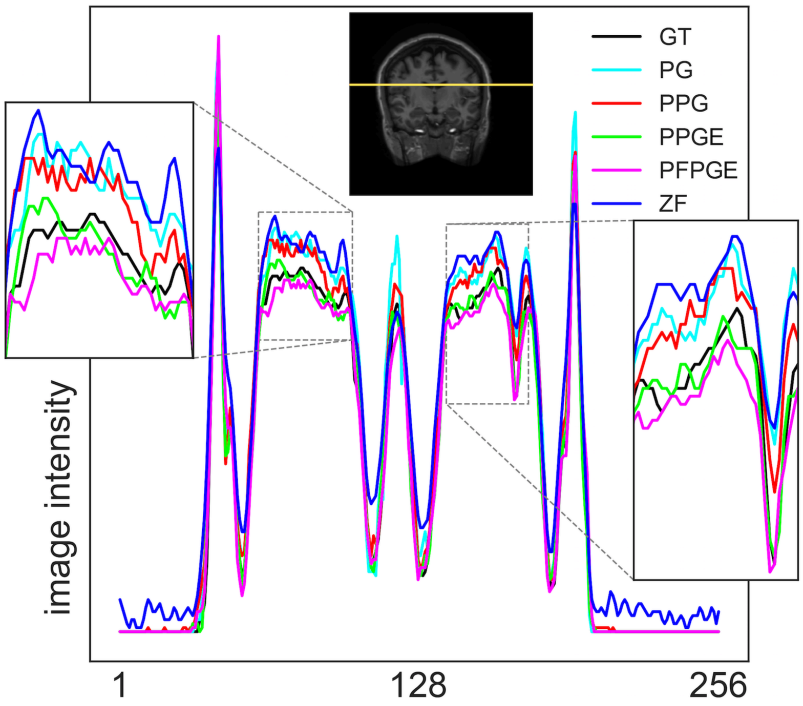


Figure 3.4.: Comparison of horizontal line profiles of our proposed EnrichGAN model (with variants), GT and ZF. ZF is reconstructed from 30% retained data. The vertical axis represents image intensity. The horizontal axis represents position of pixels on the target yellow line shown in the (randomly selected) MRI image.

Figure 3.4 presents comparison of horizontal line profiles of our proposed EnrichGAN model (with variants), GT and ZF. ZF is reconstructed from 30% retained data. Specifically, compared with the GT, the line profile of the ZF reconstruction shows a significant amount of difference in the intensity values. For our proposed models, PG and PPG achieve a clear improvement over the ZF, while PPGE and PFPGE obtain the most accurate line profile that is highly close to that of the GT. This demonstrates that the proposed enrichment learning process has an crucial impact on generating superior MRI reconstruction.

Comparison with other methods

The quantitative and qualitative comparison results of representative CS-MRI approaches can be seen in Table 3.1 and Figure 3.5, respectively. Conventional CS-MRI methods (TV, SIDWT and RecPF) have limited performance in terms of both quantitative and qualitative evaluation (e.g. the reconstructed MRI image contains no clear details but much blurring effect, as can be seen in Figure 3.5 (c), (d) and (e)). Dictionary learning (DLMRI) and patch-based (PBDW and PANO) methods achieve better reconstruction results, but the details of the reconstruction image tend to be overly smoothed (Figure 3.5 (f), (g) and (h)). The reconstruction image of Noiselet method also contains visible blurring effect (Figure 3.5 (i)). BM3D and DeepADMM are capable of producing high-quality reconstruction of MRI images (Figure 3.5 (j) and (k)). Compared with all these methods, our proposed EnrichGAN model (PFPGE) generates the most visually-convincing result that contains clear and fine local structural details (Figure 3.5 (o)), with much better quantitative performance of NMSE, PSNR and SSIM.

In addition, as can be seen from Table 3.1, the testing time of our proposed models is much faster than that of the other comparison methods (around 0.2 seconds using a CPU or 5 milliseconds using a GPU). This is an important advantage, as it enables real-time applications of CS-MRI reconstruction.

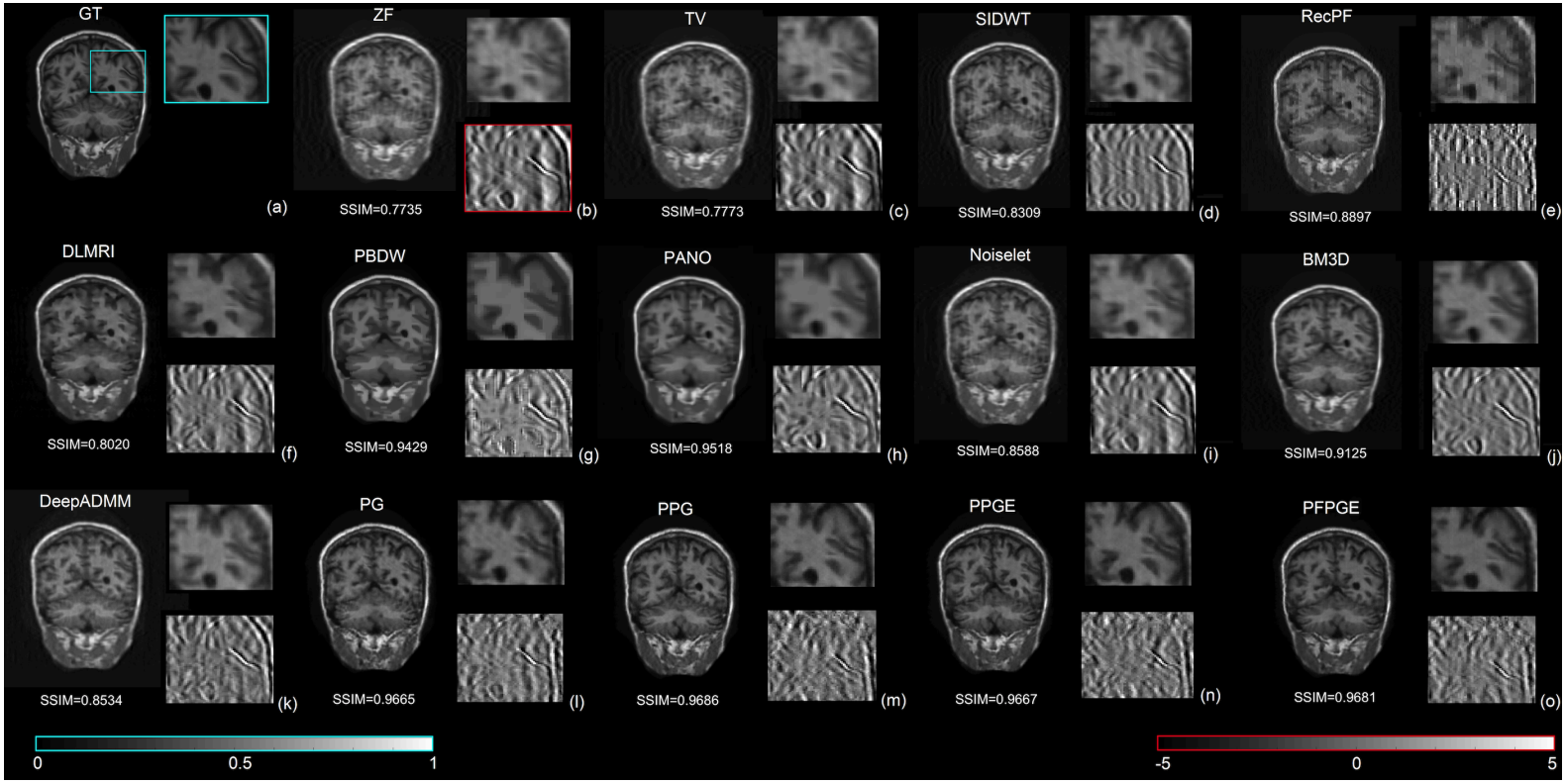


Figure 3.5.: Qualitative results of all the comparison methods and our proposed EnrichGAN (with variants), using the 1D Gaussian undersampling mask with 30% retained data. The cyan frame in (a) presents the zoomed-in ROI, and the red frame in (b) presents the difference image (50×) between the generated image and the GT. Corresponding zoomed-in ROIs are shown in (c-o). Bottom left and right present the colour bar for images and difference images respectively.

Study on noise tolerance

In practice, the raw data collected from a MRI scanner may contain some levels of noise. We would like to evaluate the noise tolerance of CS-MRI methods, including the comparison methods and our proposed models, as a way of demonstrating their applicability in practical use. We fabricated different levels of additive white Gaussian noise and added the noise to the k-space data before the undersampling mask was used. The added white Gaussian noise is a good approximation of the real noise distribution for the MRI data in k-space [7].

Figure 3.6 presents the comparison of noise tolerance between the representative methods and our proposed EnrichGAN and variants. The performance of the comparison methods (measured in PSNR) is significantly degraded due to the addition of noise, which demonstrates that their capability of noise tolerance is limited. By contrast, our proposed methods maintain similar performance when less than 10% level of noise is applied. Furthermore, when the added noise level is 20%, our full model (PFPGE) can still be able to achieve 34dB PSNR, which outperforms the comparison methods to a large degree. Therefore, our proposed models demonstrate relatively higher noise tolerance.

Study on different undersampling patterns

We further evaluated the quantitative performance (PSNR) of the comparison methods and our proposed models with respect to different undersampling patterns, including 1D Gaussian (G1D), 2D Gaussian (G2D) and 2D Poisson disc (P2D). 30% retained data was use in the experiments. As can be seen in Figure 3.7, for every different undersampling pattern, our full model (PFPGE) always achieves the best PSNR result, compared with all other methods. This demonstrates that our proposed EnrichGAN model has general applicability to various undersampling masks. Such important property of generalisation further implies that EnrichGAN could perform even better with op-

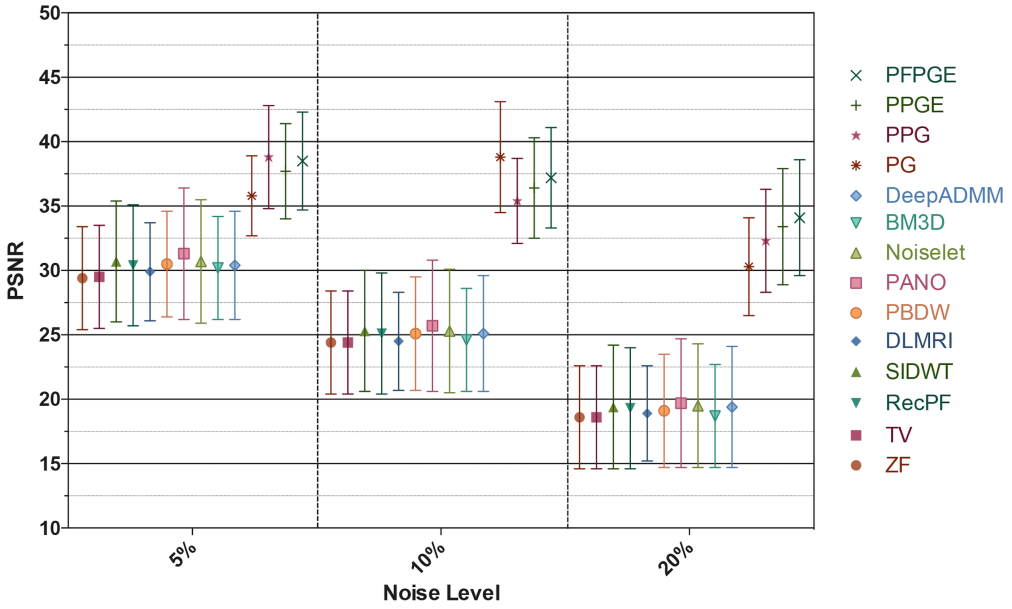


Figure 3.6.: Quantitative comparison (PSNR) of noise tolerance between the representative methods of CS-MRI reconstruction and our proposed EnrichGAN model (with variants). The vertical axis represents PSNR value. The horizontal axis represents different levels of white Gaussian noise added to the k-space data before applying the undersampling (1D Gaussian undersampling mask with 30% retained data).

timal undersampling masks applied, in which finding the best undersampling scheme is a major direction of CS-MRI research.

Zero-shot inference on pathological cases

Compared with normal MRI images, reconstruction of the pathological MRI images reveals higher diagnostic value, which is a highly desirable property of CS-MRI methods. However, in general, the size of pathological MRI image dataset is more limited than that of normal MRI image dataset. Rather than training the data-driven model on limited data, we instead studied whether our proposed EnrichGAN model is capable of performing zero-shot inference [35] on pathological cases.

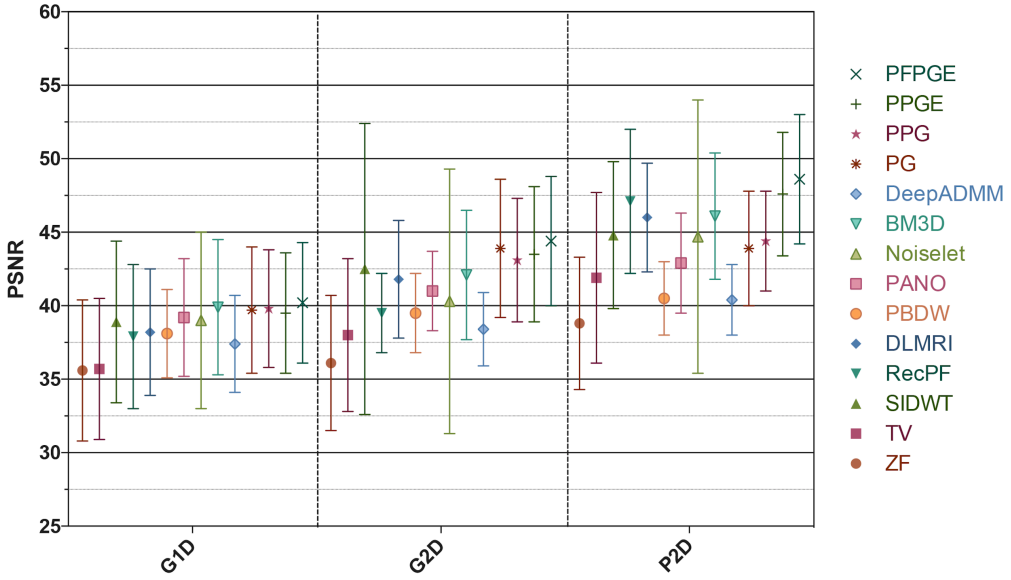


Figure 3.7.: Quantitative comparison (PSNR) of representative methods of CS-MRI and our proposed EnrichGAN (with variants) using difference undersampling masks with 30% retained data. The vertical axis represents PSNR value. Three masks are applied for comparison: 1D Gaussian (G1D), 2D Gaussian (G2D) and 2D Poisson disc (P2D).

Specifically, we employed the trained full model (PFPGE) on normal brain MRI data to reconstruct pathological MRI images that were not contained in the training dataset. We used 1D Gaussian undersampling mask with 30% retained data for the reconstruction of ZF as the input image to the model.

Figure 3.8 shows the generated reconstruction results of pathological brain MRI images using our full model PFPGE. PFPGE achieves superior reconstruction images with clear and faithful structural details of various brain lesions. Compared with the ZF reconstruction, our model is able to enrich finer details of pathological patterns (e.g. edges and textures of brain tumour pointed by green arrows in Figure 3.8 (a-c)) that are close to those of the GT.

For pathological cardiac MRI images, our EnrichGAN model is still

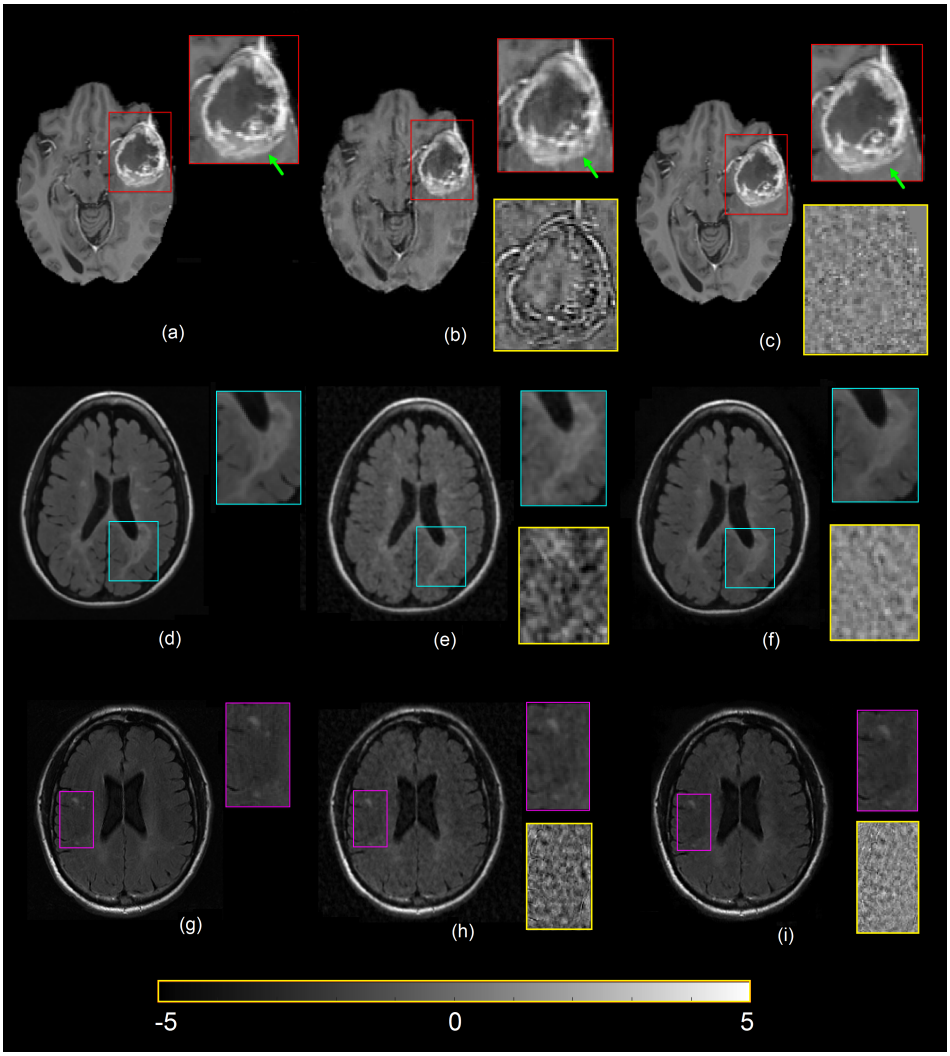


Figure 3.8.: Zero-shot inference on pathological brain MRI images using our proposed EnrichGAN model. (*left*) GT. (*middle*) ZF. (*right*) PFPGE. 1D Gaussian undersampling mask with 30% retained data is applied for the reconstruction of ZF. Coloured frames present the zoomed-in ROI and the difference image (50×) between the generated image and the GT. Bottom presents the colour bar for difference images.

able to generate reasonable reconstruction with clear local structures and details of cardiac anatomy, as can be seen in Figure 3.9. However,

there is an obvious loss of structures around the peripheral regions of the generated image (pink frames in Figure 3.9). This could be because our model is only trained on normal brain MRI images, where the pixel value of the corresponding peripheral regions is always zero. As a result, for the zero-shot inference on cardiac MRI images, the model learns to simply enforce the pixel values of those regions to be zero. Although this structure loss of peripheral regions is clinically unimportant for cardiac MRI, it still implies that our EnrichGAN model may not perform zero-shot inference well on all categories of MRI images.

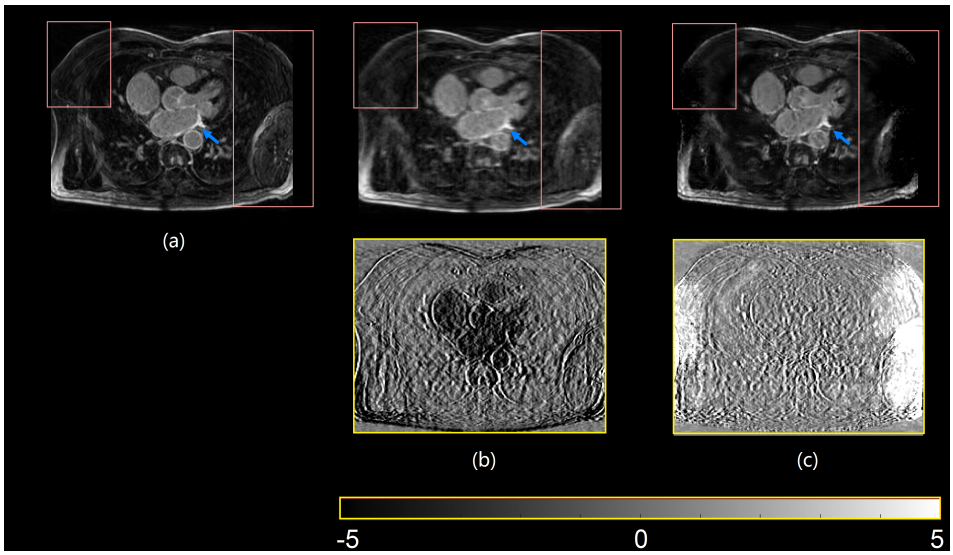


Figure 3.9.: Zero-shot inference on pathological cardiac MRI images using our proposed EnrichGAN model. (*left*) GT. (*middle*) ZF. (*right*) PFPGE. 1D Gaussian undersampling mask with 30% retained data is applied for the reconstruction of ZF. Pink frame presents the peripheral regions. Yellow frame presents the difference image (50×) between the generated image and the GT. Bottom presents the colour bar for difference images.

Study on intermediate results

In the context of CS-MRI reconstruction, it is of great importance to ensure that our conditional GAN based model does not learn to enrich unrealistic details and structures. This applies not only to the final generated images, but also to the intermediate results. We studied this by thoroughly scrutinising the intermediate results and final reconstruction images. Figure 3.10 demonstrates the visualisation of intermediate generated results of our proposed EnrichGAN model (using 1D Gaussian undersampling mask with 30% retained data). The intermediate images are generated by the model trained after 0, 5, 10 and 15 (final) epochs. We can see that in both normal and pathological cases, there are no apparent unreal details and structures generated by the model. This is because the input ZF reconstruction and the applied multiple loss terms impose strong constraints on our model to only enrich realistic details that match the empirical data distribution of GT.

Convergence analysis of enrichment learning

Figure 3.11 presents the convergence analysis comparison on employing the proposed enrichment learning process (PPGE) or not (PPG). In general, enrichment learning can not only make the convergence speed much faster (Figure 3.11 (a)), but also stabilise the training process of our model (Figure 3.11 (b)). This is due to the fact that enrichment learning enforces the generator to generate the missing details and structures only. As a result, the required complexity of the model is significantly reduced, which leads to a faster convergence speed and a more stable training process.

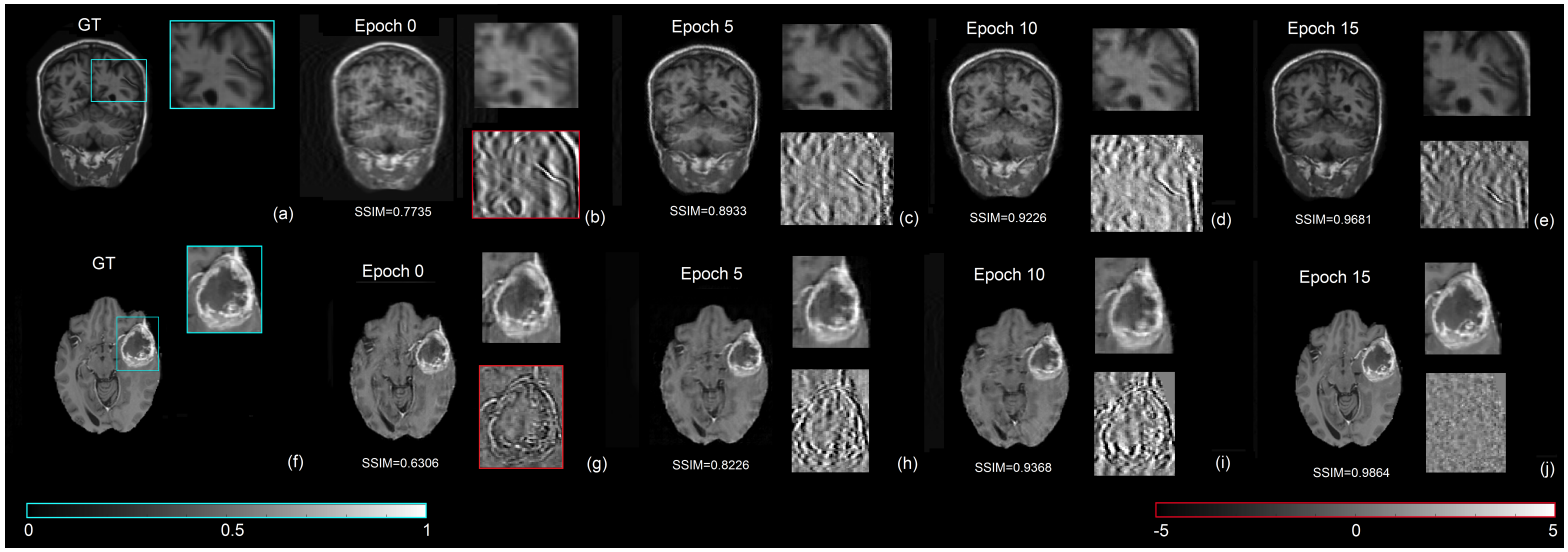


Figure 3.10.: Qualitative visualisation of intermediate results of our proposed EnrichGAN model. 1D Gaussian undersampling mask with 30% retained data is used for the reconstruction of ZF as the input image to the model. Intermediate results are generated by the model trained after 0, 5, 10 and 15 (final) epochs. The cyan frames present the zoomed-in ROI, and the red frames present the difference image (50×) between the generated image and the GT. Corresponding zoomed-in ROIs are shown in (c-e) and (h-j). Bottom left and right present the colour bar for images and difference images respectively.

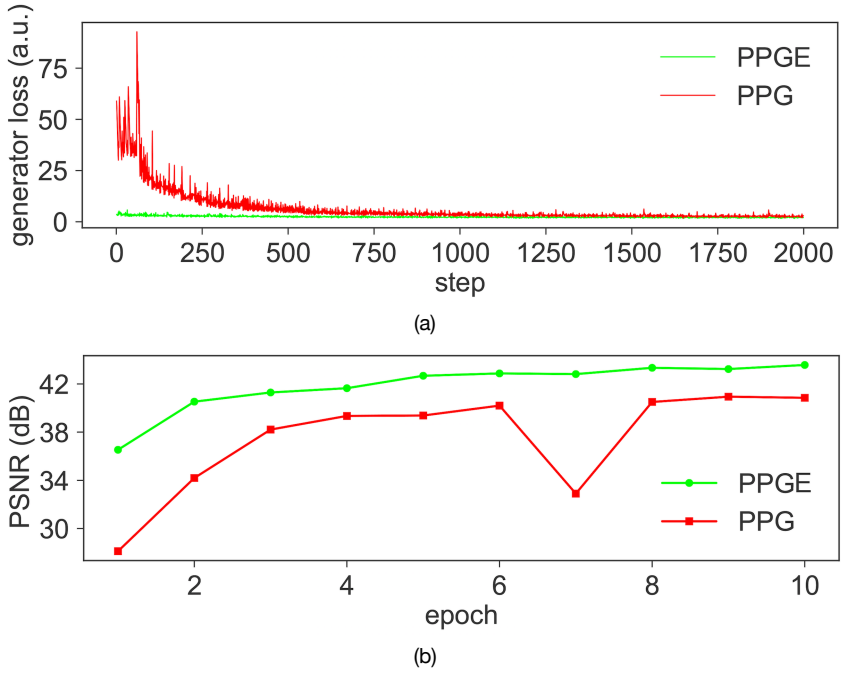


Figure 3.11.: Convergence analysis on the use of enrichment learning (PPG vs. PPGE). (a) Speed of convergence. The vertical axis presents generator loss and the horizontal axis presents the training steps. (b) Stability of convergence. The vertical axis presents value of PSNR and the horizontal axis presents training epochs.

Study on different hyperparameter settings

We finally performed a quantitative comparison to demonstrate that tuning the hyperparameters of our proposed EnrichGAN model will not significantly affect the performance of the model, which is a desirable requirement for medical imaging processing tasks such as CS-MRI reconstruction. Specifically, we tuned one of the hyperparameters (α, β and γ) by 0.5×, 5× and 10×, while maintaining the others unchanged. As can be seen from Figure 3.12, apart from 5×β and 10×β that lead to clearly worse performance, all the other hyperparameter settings achieve comparable or even better results. This demonstrates that our

EnrichGAN model is robust with minimal hyperparameter tuning.

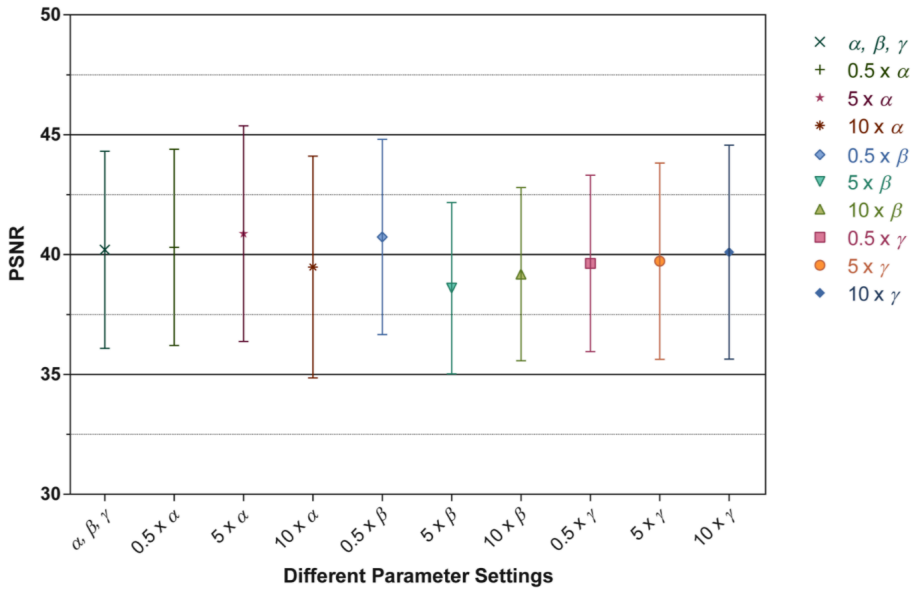


Figure 3.12.: Quantitative comparison (PSNR) of different hyperparameter settings of our proposed EnrichGAN model. 1D Gaussian undersampling mask with 30% retained data is applied for the reconstruction of ZF as the input image to the model. The original parameters are $\alpha = 15$ (for pixel-wise image domain loss $\mathcal{L}_i$), $\beta = 0.1$ (for frequency domain loss $\mathcal{L}_f$) and $\gamma = 0.0025$ (for perceptual loss $\mathcal{L}_p$). The vertical axis presents PSNR value and the horizontal axis presents different hyperparameter settings.

3.6. Conclusions

In this chapter, we proposed EnrichGAN, a novel conditional GAN based architecture for CS-MRI reconstruction. We formulated CS-MRI reconstruction as an image-to-image generation task, in which our proposed approach was applied to enrich the local structural details of the input images to match those of the ground truth images. In this sense, CS-MRI reconstruction can be regarded as a challenging

application of image-to-image generation that exhibits enrichment creativity. Moreover, an innovative objective function (consisting of four designated loss terms) as well as an enrichment learning process were proposed to contribute to generating fine, clear and accurate details and structures of MRI images.

We performed comprehensive experiments to evaluate our proposed EnrichGAN model and its variants. Specifically, EnrichGAN outperformed the comparison methods (including conventional and state-of-the-art methods of CS-MRI reconstruction) qualitatively and quantitatively, with a much faster processing time that enables real-time applications. Also, our approaches demonstrated better noise tolerance and more general applicability to various undersampling schemes. For pathological MRI images, EnrichGAN was able to perform zero-shot inference and achieved faithful results. We further demonstrated that EnrichGAN not only enriched realistic structural details in both intermediate and final reconstruction images, but also was robust with minimal hyperparameter tuning, which is of great significance for tasks of medical image processing. Finally, we showed the effectiveness of the different designated loss terms as well as the enrichment learning process via qualitative and quantitative ablation studies.

In summary, our proposed EnrichGAN model achieved superior results of fast CS-MRI reconstruction. To the best of our knowledge, this is the first work that utilises conditional GAN based architecture for CS-MRI reconstruction. Such superior results demonstrated that conditional GAN based approaches can learn to achieve enrichment creativity from challenging image-to-image generation tasks, such as CS-MRI reconstruction.

Author's contributions. This work was completed in collaboration with Dr. Guang Yang. I mainly contributed to the design and the development of the proposed methods (EnrichGAN and all its variants, with their network architectures and objective functions), and the conduction of all the experiments related to these proposed methods. Dr. Guang Yang initiated the study, conducted the designed experiments

related to the comparison methods, provided test datasets with noise and random undersampling masks.

4. Learning to Achieve Transformative Creativity

4.1. Motivation

Human beings have the ability to manipulate representations of imaginary pictures in their minds [115]. For example, if a person first imagines "a flower with red and thin petals" as a mental image, this person could then be capable of making modifications to this mental image, such as changing the red colour to yellow, adjusting the thin petals to thick ones, increasing the number of presented flowers, etc. Such capability of flexible manipulation is of great significance in the generation of creative thoughts for visual arts [119].

In this chapter, we attempt to endow machines with the ability of manipulating visual features in a goal-oriented fashion. We achieve this goal by formulating it as an image-to-image generation task. Specifically, given an input source image and a target text description, the generated output image should satisfy the following three requirements: 1) matching what the target description specifies; 2) maintaining other features of the input source image that are irrelevant to the target description; 3) being realistic and plausible. We refer to such image-to-image generation task as semantic image manipulation. Some examples of this task are presented in Figure 4.1. Essentially, semantic image manipulation demonstrates transformative creativity of image-to-image generation, because it involves transformation of partial representations of the input image, resulting in the generation of new images that never existed before.

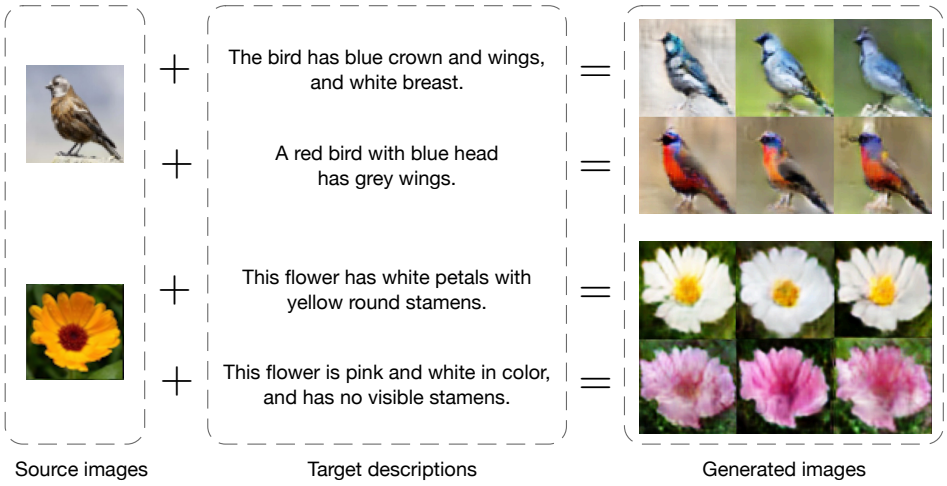


Figure 4.1.: Examples of semantic image manipulation that exhibits transformative creativity. Given an input source image and a target natural language description, the generated image should 1) correspond to the target description; 2) keep the irrelevant features to the target description unchanged; 3) be realistic and plausible. Note that it should be able to generate varied output images that meet these three conditions.

A key challenge for semantic image manipulation is to learn a good mapping between the visual features of images and the semantic representations of natural languages. Such learned mapping is a prerequisite for the generated images to meet the aforementioned requirements. To this end, we propose SimGAN, a novel conditional generative adversarial network (GAN) based architecture for semantic image manipulation. SimGAN utilises the adversarial learning process and a specified objective function to implicitly learn the representation mapping between the two modalities. Based on this learned mapping and the designed architecture, SimGAN is able to generate compelling results of semantic image manipulation.

The rest of this chapter is organised as follows. We first introduce the background related to our work in Section 4.2. We then describe

in detail a baseline method for comparison in Section 4.3 and our proposed SimGAN model in Section 4.4. After that, we demonstrate and discuss the results of the conducted experiments in Section 4.5. Finally, we draw conclusions of our work in Section 4.6.

4.2. Background

Our proposed SimGAN model for semantic image manipulation is based on the conditional GAN framework. Unlike ordinary image-to-image generation tasks (as introduced in Chapter 2.3.3) that typically condition GANs on images only, semantic image manipulation requires an extra condition of texts. There are several works that employ text-conditioned conditional GAN framework to address the text-to-image generation problem. Reed et al. [110] for the first time proposed an effective GAN architecture (termed GAN-INT-CLS) and training strategies, which successfully enabled the generation of realistic bird and flower images (of size 64×64) from text descriptions. They further introduced Generative Adversarial What-Where Network (GAWWN) [109], which was capable of synthesising plausible 128×128 images. This was achieved by conditioning GANs on both text descriptions and object locations (specified by bounding boxes or keypoints). Zhang et al. [159, 158] proposed StackGAN, which was able to generate compelling 256×256 images from text descriptions only. StackGAN was composed of two GANs to perform two-stage generation. Given a text description, GAN of the first stage generated a corresponding coarse image only, and GAN of the second stage generated more realistic details. More recently, Xu et al. [145] proposed AttnGAN that incorporated attention mechanism and multi-stage refinement to perform fine-grained text-to-image generation. Our proposed approach differs from these works, in that both image and text information are conditioned to generate images that satisfy the requirements of semantic image manipulation.

As mentioned in Section 4.1, it is required to learn a mapping be-

tween representations of images and texts for semantic image manipulation. Before learning such a mapping, we need to first obtain the representations of images and texts by training the corresponding encoders. For the text representations, we employ a pretrained text encoder in this work, which will be described in detail in Section 4.4.4. For the image representations, within GAN framework, there exist two methods for training an encoder: 1) on the real images from real data distribution, or 2) on the generated images from the learned generative distribution.

Several GAN-based approaches adopt the first method to explicitly train an encoder to obtain image representations. For example, Donahue et al. [21] and Dumoulin et al. [27] proposed BiGAN architecture, where a generator and an encoder were jointly trained, and a discriminator discriminated in both data space and latent space (i.e. real images and their encoded latent representations vs. generated images and input latent variable). Larsen et al. [70] combined variational autoencoder (VAE) [67, 111] with GANs by collapsing the VAE decoder and GAN generator into one module, such that the default pixel-wise reconstruction loss of VAE was replaced by the feature-wise loss imposed by the discriminator of GANs. Makhzani et al. [84] proposed adversarial autoencoder (AAE), which was composed of a standard autoencoder and a discriminator. The autoencoder reconstructed images from the encoded latent variables. The discriminator distinguished between these latent variables (as negative samples) and samples from a prior distribution specified by the user (as positive samples). Similarly, our proposed SimGAN model employs the method of learning the image representation from real data.

The second method of training the image encoder on generated images is adopted in a baseline method of this work, as described in detail in the next section.

4.3. Baseline method

We adopt the two-step approach proposed by Reed et al. [110] as a baseline method for comparison with our proposed SimGAN model. To the best of our knowledge, this is the only approach that is capable of addressing the same semantic image manipulation task as in our work.

Specifically, in the first step, a generator G is trained for text-to-image generation by using the text-conditioned conditional GAN framework. G takes as input a latent variable z and the embedding of a target text description $\varphi(\mathrm{t})$, and outputs a generated image $\hat{\mathrm{x}}$ that matches the description t. In the second step, an encoder E is trained to encode $\hat{\mathrm{x}}$ (generated by the trained G) into latent representations $\hat{\mathrm{z}}$. A l_2 loss is used to train the encoder E as follows:

$$\mathcal{L}_e(E) = \mathbb{E}_{\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z}),\mathrm{t}\sim p_{\mathrm{t}}(\mathrm{t})}[\|\mathrm{z} - E(G(\mathrm{z}, \varphi(\mathrm{t})))\|_2^2], \tag{4.1}$$

where $p_{\mathrm{z}}(\mathrm{z})$ represents the prior distribution over latent variable z and $p_{\mathrm{t}}(\mathrm{t})$ represents the real text distribution.

At inference stage, given a source image x and a target text description t, the trained encoder E first generates the latent representations of x: $\hat{\mathrm{z}} = E(\mathrm{x})$. After that, the trained generator G generates the final result of semantic image manipulation based on $\hat{\mathrm{z}}$ and $\varphi(\mathrm{t})$: $\hat{\mathrm{x}} = G(\hat{\mathrm{z}}, \varphi(\mathrm{t}))$.

Similar two-step methods were proposed by Perarnau et al. [101] and Dong et al. [22] for attribute-based image manipulation and unsupervised image-to-image translation, respectively.

The two-step methods introduced above have a main limitation, as also discussed in [21]. The encoder is trained on generated images only and never takes real images into account. This would not be an issue if the generative distribution of the trained generator perfectly matches the real data distribution. However, in practice, this almost never happens due to the high complexity of real data distribution (e.g.

natural images). As a consequence, at inference stage, the trained encoder could produce poor representations of real images, which would degrade the quality of final generated images.

4.4. Method

We propose SimGAN architecture for semantic image manipulation, which is a challenging image-to-image generation task that demonstrates transformative creativity. Figure 4.2 presents the overall schema of SimGAN architecture. Specifically, SimGAN is composed of five networks: an image encoder E, a residual transformation unit, a generator G, a pretrained text encoder φ and a discriminator D. The encoder E takes as input a source image x and outputs the encoded latent representations $E(\mathrm{x})$. $E(\mathrm{x})$ and the text representations of a target text description (encoded by the text encoder φ) are concatenated and processed by the residual transformation unit. The generator G takes in the output of residual transformation unit, and generates an image that satisfies the requirements of semantic image manipulation.

In the following sections, we first describe in detail how we design the objective function that enables our approach to generate superior results. We then introduce the network architecture. After that, we describe another option of utilising a pretrained image encoder as well as the employed pretrained text encoder. We finally introduce how to enable the model to generate diverse images, given one source image and a target description.

4.4.1. Objective function

The generated image for semantic image manipulation is required to not only be realistic and match the given text description, but also maintain representations of the source image that are irrelevant to the description. It would be difficult to explicitly define a corresponding learning objective for this complicated generation problem. Instead,

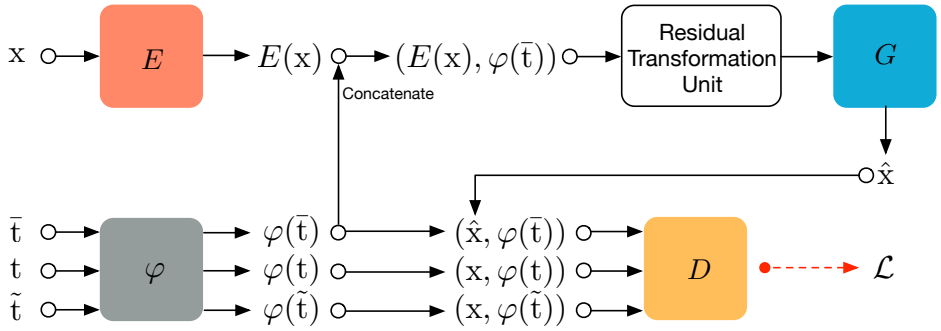


Figure 4.2.: Schema of our proposed SimGAN architecture for semantic image manipulation. It consists of an image encoder E, a residual transformation unit, a generator G, a (pretrained) text encoder φ and a discriminator D. SimGAN can be regarded as a variant of conditional GAN conditioned on both image and text information. We employ adversarial learning with an designated adversarial loss $\mathcal{L}$ to train our model.

we employ adversarial learning with a designated objective function to implicitly learn an adaptive loss for semantic image manipulation.

Given a source image x, we first define three categories of text descriptions as follows:

1. Matching text (denoted as t), which precisely describes the content of x;
2. Mismatching text (denoted as $\tilde{\text{t}}$), which refers to all available text descriptions except t from the dataset;
3. Semantically relevant text (denoted as $\bar{\text{t}}$), which contains semantics that are partially related to the representations of x. For example, the two text descriptions of birds in Figure 4.1 are semantically relevant to the source image of bird, but share no relationship with the flower images.

We then feed three types of pairs of image and text representations to the discriminator D, which learns to discriminate among these input pairs:

1. $s_r^+ = D(\text{x}, \varphi(\text{t}))$ (real image, matching text);

2. $s_w^- = D(\mathrm{x}, \varphi(\tilde{\mathrm{t}}))$ (real image, mismatching text);
3. $s_s^- = D(\hat{\mathrm{x}}, \varphi(\bar{\mathrm{t}}))$ (generated image $\hat{\mathrm{x}} = G(E(\mathrm{x}), \varphi(\bar{\mathrm{t}}))$, semantically relevant text),

where s_r^+, s_w^- and s_s^- represent the output probability score of D for the corresponding input pairs, $+$ and $-$ respectively represent real or fake samples of input pairs discriminated by D. s_r^+ and s_s^- provide the generator G with an updating signal for generating realistic images. They also implicitly encourage the generated images to match the target description. In addition, the use of s_w^- introduced by Reed et al. [110] further imposes an additional constraint on our model, which is enforced to learn stronger alignment of images and text descriptions explicitly.

It is worth emphasising the importance of using $\bar{\mathrm{t}}$ (rather than t or $\tilde{\mathrm{t}}$) in the term s_s^-, which is a critical factor to ensure the effectiveness of the adversarial training process for semantic image manipulation. Specifically, at early training stage, G only generates unrealistic images, which will be easily discriminated by D as fake samples via the term s_r^+, without necessarily considering the text condition. As the training proceeds, G gradually learns to generate realistic images. D in the meanwhile gradually learns to evaluate the alignment between images and texts via s_w^-.

When G is able to generate realistic images, it is the type of text description used in s_s^- that determines whether the generated images will satisfy the requirements of semantic image manipulation.

If mismatching text $\tilde{\mathrm{t}}$ is used in s_s^-, then the input pairs to D are realistic images with mismatching texts. Due to the term s_w^-, D will easily discriminate these input pairs as fake samples, and will emit a strong updating signal to enforce G to generate images that correspond to the mismatching texts. As a result, G will eventually learn to generate images that share little or no relationship to the input source images. In other words, the target text descriptions dominate and the source images are completely ignored by G.

If matching text t is employed in s_s^-, then the input pairs to D are

always realistic images with completely matching texts. In this case, because of the term s_w^-, G will be enforced to generate images that correspond to the matching texts. Since the source images directly match t, G will simply generate the source images without considering the text information. In other words, the source images dominate and the target text descriptions are completely ignored by G.

In our method we employ $\bar{\mathrm{t}}$ in s_s^-, which can be seen as a trade-off between the use of $\tilde{\mathrm{t}}$ and t. Since $\bar{\mathrm{t}}$ is guaranteed to be semantically relevant to the source image, both information of images and texts will be taken into account by G in the adversarial training procedure. This enables the generation of plausible results for semantic image manipulation.

We therefore define the full objective of our proposed SimGAN as follows:

$$\begin{aligned}\min_{G,E}\max_{D}\mathcal{L}(G,E,D) = {} & \mathbb{E}_{(\mathrm{x},\mathrm{t})\sim p_{<\mathrm{x},\mathrm{t}>}(\mathrm{x},\mathrm{t})}[\log D(\mathrm{x},\varphi(\mathrm{t}))] \\ & + \mathbb{E}_{(\mathrm{x},\tilde{\mathrm{t}})\sim p_{<\mathrm{x},\tilde{\mathrm{t}}>}(\mathrm{x},\tilde{\mathrm{t}})}[\log(1-D(\mathrm{x},\varphi(\tilde{\mathrm{t}})))] \\ & + \mathbb{E}_{(\mathrm{x},\bar{\mathrm{t}})\sim p_{<\mathrm{x},\bar{\mathrm{t}}>}(\mathrm{x},\bar{\mathrm{t}})}[\log(1-D(G(E(\mathrm{x}),\varphi(\bar{\mathrm{t}})),\varphi(\bar{\mathrm{t}})))].\end{aligned} \tag{4.2}$$

4.4.2. Network architecture

In this section, we briefly describe the network architecture of our proposed SimGAN as follows. For a more detailed visualisation of the architecture, please refer to Figure C.1 in Appendix C.

The encoder E

The encoder E encodes input source images (of size 64×64) into feature maps (of dimension $16 \times 16 \times 512$). E consists of three convolutional layers, with the latter two being followed by a layer of batch normalisation [58] and ReLU.

The target text descriptions are encoded into semantic representations (of dimension 128) by a pretrained text encoder. The semantic

representations are then spatially replicated to be of dimension $16 \times 16 \times 128$, and concatenated with the encoded image representations, resulting in the input to the residual transformation unit.

The residual transformation unit

The residual transformation unit is composed of four residual blocks [48]. Each block consists of two convolutional layers, each of which is followed by a layer of batch normalisation and ReLU. The reasons to employ the residual transformation unit in SimGAN architecture are twofold. First, residual connections help retain the representations of underlying structures in the source images [62]. This is an attractive property for semantic image manipulation, which requires the generated images to maintain features of the source images that are irrelevant to the target text description. Second, the use of residual blocks enables a deeper encoding process, from which SimGAN could learn a better mapping between finer representations of both images and texts.

The generator G

The generator G transforms the resulting representations of images and texts from the residual transformation into generated images of size 64×64. G consists of two upsampling blocks, each of which is made up of an upsampling layer, a convolutional layer and a layer of batch normalisation and ReLU.

The discriminator D

The discriminator D first encodes input images into feature maps of dimension $4 \times 4 \times 512$, by four convolutional layers and a residual block. Then the image representations are concatenated with representations of text descriptions. Finally, two convolutional layers are employed to produce final results. Batch normalisation and leaky-ReLU [83] are applied after each convolutional layer (except the first and last layer).

4.4.3. Better image representations

For semantic image manipulation, the target text descriptions typically describe in detail foreground objects of the source images. This implies that for most cases backgrounds of the generated images should be preserved as similar to those of the source images as possible. However, the image encoder trained along with the entire model might not learn to produce good representations of various backgrounds, due to limited size of the datasets employed in this work. We therefore propose to employ a (pretrained) much deeper convolutional neural network (CNN) to obtain better image representations.

More specifically, we employ the VGG (VGG16) network [126] (pretrained on ImageNet [114]) to replace the original encoder in SimGAN. We employ the output of conv4 layer of the VGG, which produces feature maps of dimension $16 \times 16 \times 512$. Other components of our architecture remain unchanged. We term this architecture with VGG as SimGAN-V.

4.4.4. Visual-semantic text representations

For the pretrained text encoder φ, we adopt the approach proposed by Kiros et al. [68], which is capable of encoding text descriptions into visual-semantic text representations. More specifically, given pairs of an image x and its corresponding text description t, we first utilise a CNN model ϕ and a long short-term memory (LSTM) recurrent neural network (RNN) model [52, 37] φ to respectively encode x and t into representations $\phi(\mathrm{x})$ and $\varphi(\mathrm{t})$. In order to achieve the visual-semantic text representations, we then learn to find a joint embedding space of images and texts via the pair-wise ranking loss, which is defined as follows:

$$\begin{aligned} \min_{\theta} & \sum_{\mathrm{x}} \sum_{k} \max\{0, \alpha - s(\phi(\mathrm{x}), \varphi(\mathrm{t})) + s(\phi(\mathrm{x}), \varphi(\mathrm{t}_k))\} \\ + & \sum_{\mathrm{t}} \sum_{k} \max\{0, \alpha - s(\varphi(\mathrm{t}), \phi(\mathrm{x})) + s(\varphi(\mathrm{t}), \phi(\mathrm{x}_k))\}, \end{aligned} \tag{4.3}$$

where θ represents all parameters in ϕ and φ, t_k and x_k are mismatching texts (to x) and images (to t), $s(\phi(\mathrm{x}), \varphi(\mathrm{t}))$ measures the cosine similarity between $\phi(\mathrm{x})$ and $\varphi(\mathrm{t})$, α is the margin. This loss can be seen as minimising the cosine similarity between representations of mismatching pairs of images and texts, while maximising the cosine similarity between representations of matching pairs of them. After training, visual-semantic representations of given texts will be produced by the text encoder φ, which is applied as the pretrained text encoder in our SimGAN model.

4.4.5. Generation of diverse images

As similar to many of other image-to-image generation tasks that demonstrate transformative creativity, semantic image manipulation essentially is a one-to-many generation problem. Specifically, given a source image and a target text description, the model should be able to generate diverse images, all of which satisfy the requirements of the task. A typical way to enable the generation of multiple varied outputs is to directly provide noise as the input to generative model. However, in the case of conditional GAN based architecture for image-to-image generation problem, the generator would simply learn to ignore the input noise and produce deterministic results based on the input images [86, 59].

There are several studies proposed to resolve this limitation of image-to-image generation problems in both supervised [163] and unsupervised [56, 2, 154] settings. Different from these studies that mainly focused on image representation learning, we attempt to enable the generation of diverse images via producing additional text representations for the same target text description.

Given an encoded text representation $\varphi(\mathrm{t})$, we apply an approach of conditioning augmentation proposed by Zhang et al. [159]. Specifically, additional text representations can be sampled from a Gaussian distribution $\mathcal{N}(\mu(\varphi(\mathrm{t})), \Sigma(\varphi(\mathrm{t})))$, in which the mean $\mu(\varphi(\mathrm{t}))$ and diagonal

covariance matrix $\Sigma(\varphi(\mathrm{t}))$ are functions of $\varphi(\mathrm{t})$. Conditioning augmentation can be regarded as reducing the discontinuity in the latent space of text descriptions [159], thus enables additional text representations. A regularisation loss can be incorporated into our objective function (Equation 4.2) to further reduce overfitting and enhance the smoothness of the text latent space, which is defined as follows [67, 20, 159]:

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(\mathcal{N}(\mu(\varphi(\mathrm{t})), \Sigma(\varphi(\mathrm{t}))) \parallel \mathcal{N}(0, I)), \tag{4.4}$$

where $\mathcal{L}_{\mathrm{KL}}$ is the Kullback-Leibler (KL) divergence loss between the conditioning Gaussian distribution and the standard Gaussian distribution.

At inference stage, given a target text description t, additional text representations $\varphi(\mathrm{t})$ can be directly sampled from $\mathcal{N}(\mu(\varphi(\mathrm{t})), \Sigma(\varphi(\mathrm{t})))$, which presumably allows our proposed model to generate diverse and reasonable results.

4.5. Experiments

4.5.1. Experimental settings

In our experiments, we compared the baseline method, our proposed SimGAN and SimGAN-V for semantic image manipulation. We evaluated the three approaches on Caltech-200 bird dataset [139] and Oxford-102 flower dataset [92]. The Caltech-200 bird dataset contains 200 classes of birds, with 11788 images. We split it into 150 training classes and 50 test classes. The Oxford-102 flower dataset has 102 classes of flowers, with 8189 images. It was split into 82 training classes and 20 test classes. Each image in both datasets is associated with ten descriptive sentences (collected by Reed. et al. [108]). To improve the performance of the models, we generated additional images for training via a number of data augmentation methods, including (random) flipping, rotating, cropping and zooming.

To train our proposed model, we used the Adam optimiser [66], with an initial learning rate of 0.0002. The networks were trained for 600 epochs, and the learning rate was halved every 100 epochs. Batch size was set to 64. We implemented our methods by using TensorFlow [1] and TensorLayer [23].

4.5.2. Results

Qualitative results

We first qualitatively compared our proposed approaches with the baseline method. Figure 4.3 presents the results of the baseline method (extracted from [110] for a direct comparison) and SimGAN for the bird dataset. It is of note that all input source images and top six target text descriptions were selected from the training set, as in [110]. In general, the baseline method is able to generate images that match the target descriptions, though some generated birds appear to lack of details. However, the baseline method often fails to maintain the backgrounds and bird poses of the source images, which significantly degrades the quality of the generated images. For example, the second generated image for the description "This bird is completely red." (in Figure 4.3 (a)) is of poor quality, because the branch, which should be preserved unchanged, is mistakenly coloured red by the model. By contrast, SimGAN performs much better in preserving the backgrounds and bird poses, with clearer and finer details of the generated images that correspond to the target descriptions (e.g. various branches in the source images are maintained well). Moreover, we selected source images in the training set that contain more varied bird shapes and poses. As shown in the last five columns of Figure 4.3 (b), SimGAN successfully generates realistic images that match the target descriptions, while maintaining the various bird poses of the source images. This demonstrates that our proposed SimGAN outperforms the baseline method on bird images in the training set.

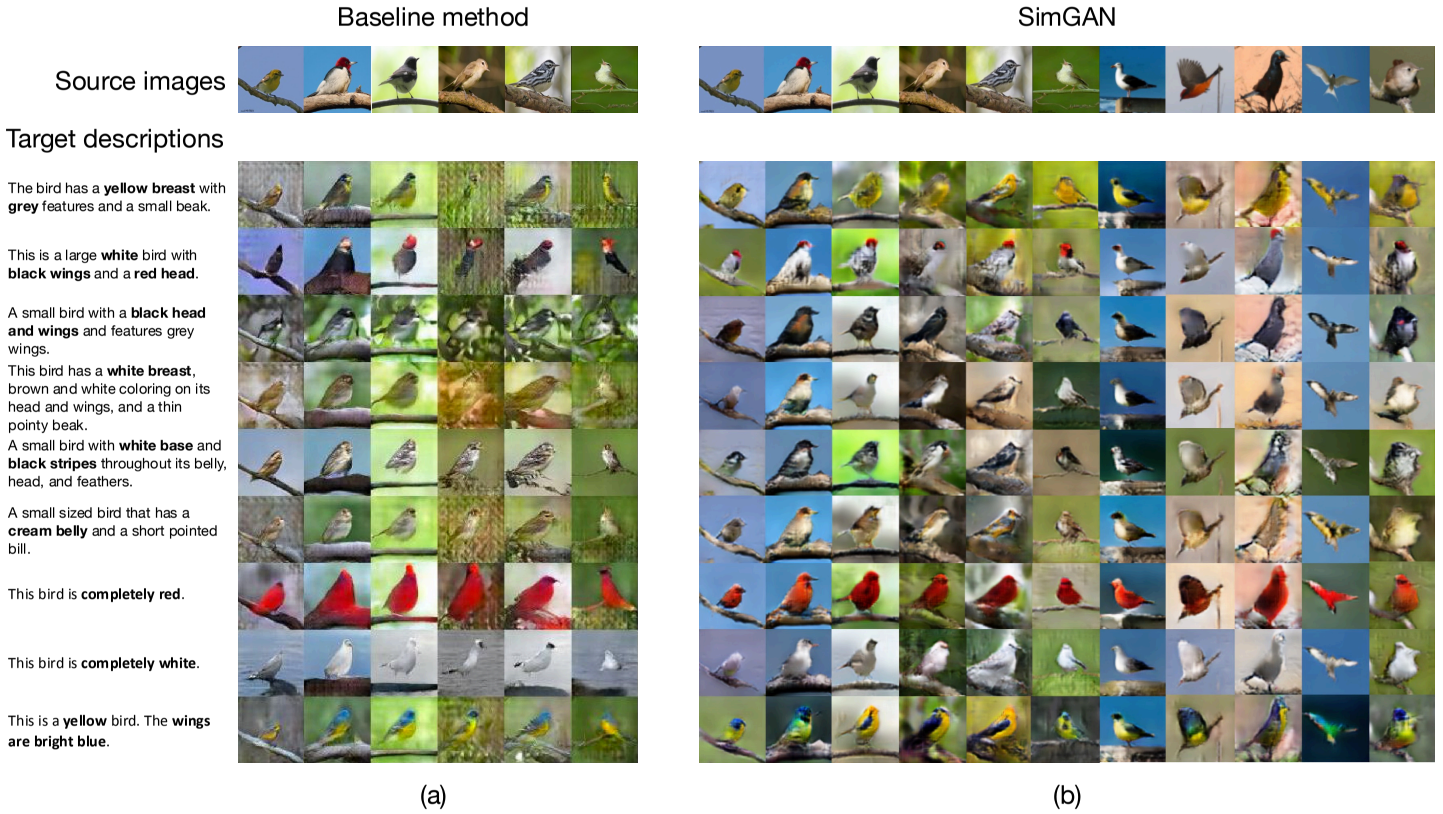


Figure 4.3.: Qualitative comparison results of the baseline method and our proposed SimGAN on images from the training set of the Caltech-200 bird dataset. (a) Results of the baseline method. These results are extracted from [110] for a direct comparison. (b) Results of our proposed SimGAN.

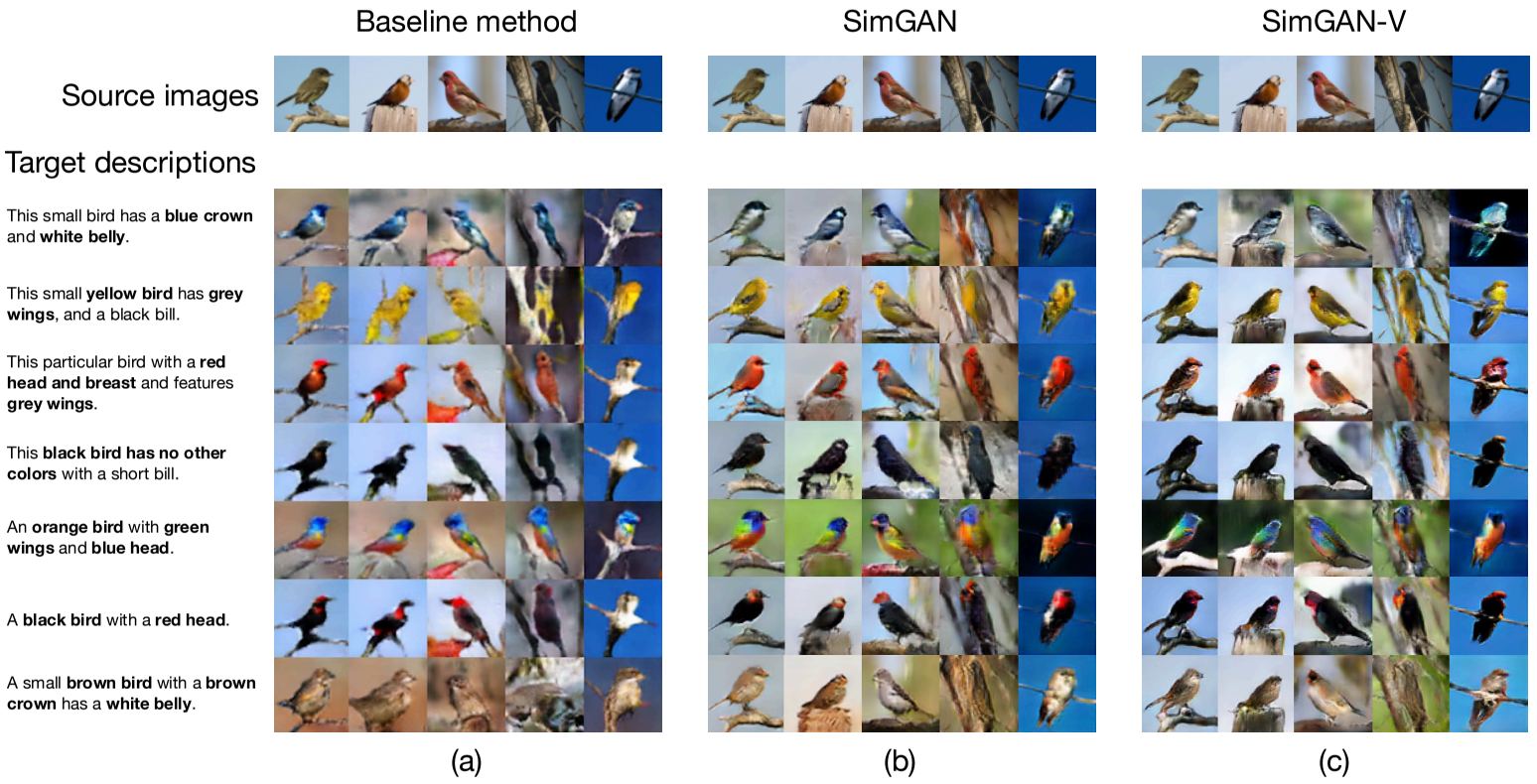


Figure 4.4.: Zero-shot comparison results of the baseline method and our proposed SimGAN and SimGAN-V on images from the test set of the Caltech-200 bird dataset. (a) Results of the baseline method. (b) Results of SimGAN. (c) Results of SimGAN-V.



Figure 4.5.: Zero-shot comparison results of our proposed SimGAN and SimGAN-V on images from the test set of the Oxford-102 flower dataset. (a) Results of SimGAN. (b) Results of SimGAN-V.

Figure 4.4 demonstrates zero-shot results of the baseline method, our proposed SimGAN and SimGAN-V on the images from the test set of the bird dataset. More specifically, the generated images of the baseline method are of poor quality: the birds tend to be mixed up with the background objects. When the input source images have more complicated backgrounds (e.g. several tree branches or an overlapping cable, as shown in the last two columns in Figure 4.4 (a)), the baseline method fails to generate realistic images to meet the requirements of semantic image manipulation. For our proposed SimGAN and SimGAN-V, they are able to generate images that not only preserve the backgrounds and bird poses of the source images, but also show clearer details than those of the baseline method. Even for the images with complex backgrounds, SimGAN and SimGAN-V can still produce reasonable results (last two columns of Figure 4.4 (b) and (c)). Compared with SimGAN, SimGAN-V tends to generate finer details of birds.

Figure 4.5 presents the zero-shot results of SimGAN and SimGAN-V on the flower dataset. Note that the results of the baseline method are not included, as it generally fails to produce plausible flower images, given the source images in the test set. One possible reason is that the trained text-to-image generator in the baseline method is only able to generate realistic images that are similar to the training images. However, flowers of different classes generally have more diverse and distinct shapes, compared with those of birds. For zero-shot generation, the input source images of flowers may contain new shapes that are never observed by the generator during the training process. As a result, for the new flower shapes that are irrelevant to the text descriptions, it is almost impossible for the generator to maintain them, resulting in low quality generated images. By contrast, our proposed SimGAN and SimGAN-V learn a good mapping between the representations of images and texts, and thus are able to directly manipulate the source images according to the descriptions while preserving other irrelevant features, regardless of whether they are observed or not in

training stage. As can be viewed in Figure 4.5, both SimGAN and SimGAN-V are able to generate superior results for semantic image manipulation on the challenging flower dataset. Also, SimGAN-V is able to generate images of flowers with clearer background, compared with SimGAN.

The presented qualitative results also show that our proposed approaches are capable of generating novel images for semantic image manipulation. These generated images have never existed before, with a clear distinction from the ones sampled from the real data distribution. In other words, the proposed methods learn to achieve higher-degree transformative creativity. For additional zero-shot results of semantic image manipulation, please refer to Figure C.2 (for the bird dataset) and Figure C.3 (for the flower dataset) in Appendix C.

Quantitative results

Evaluating the performance of generative models for creative tasks is a difficult problem. In this work, we performed a human study to quantitatively compare the quality of images generated by the baseline method, our proposed SimGAN and SimGAN-V. Although subjective factors exist, human evaluation would still be the most reliable approach to estimating whether the generated images satisfy the requirements of semantic image manipulation.

More specifically, we conducted the human study on Amazon Mechanical Turk (AMT). We recruited ten subjects to evaluate the quality of generated images. Every subject was shown five source images from each bird and flower test dataset. For every source image, seven target text descriptions were given. We then used the baseline method (for the bird dataset only but not the flower dataset, from which it fails to generate plausible images), SimGAN and SimGAN-V to generate the corresponding result images, given one paired source image and target text description. The generated images as well as the corresponding source image and target text description were all presented

to each subject, without identifying which approach was used for the generative results.

The subjects were then required to rank the generated images of each group (from 1 to 3 for the bird dataset and from 1 to 2 for the flower dataset, 1 for the best) based on three criteria as follows:

1. whether the generated image well preserved the bird pose (for bird images) or the flower shape (for flower images) of the source image;
2. whether the generated image well maintained a clear background of the source image;
3. whether the generated image well corresponded to the target text description.

Finally, the recorded ranks from the subjects were averaged as the score of quality for semantic image manipulation.

Table 4.1.: Comparison results of human evaluation between the baseline method, our proposed SimGAN and SimGAN-V on images from Caltech-200 bird dataset and Oxford-102 flower dataset. The scores are the averaged human ranks for different aspects of the generated images.

		Baseline	SimGAN	SimGAN-V
Bird images	Pose	2.69 ± .06	1.57 ± .07	1.73 ± .08
	Background	2.40 ± .08	1.73 ± .08	1.87 ± .08
	Description	2.48 ± .08	1.67 ± .07	1.86 ± .08
Flower images	Shape	/	1.50 ± .05	1.50 ± .05
	Background	/	1.59 ± .05	1.41 ± .05
	Description	/	1.52 ± .05	1.48 ± .05

Table 4.1 exhibits comparison results of the quality score between the baseline method, our proposed SimGAN and SimGAN-V. Specifically, for the bird dataset, the baseline method performs significantly worse than SimGAN and SimGAN-V in all aspects. SimGAN also

obtains better quality score than that of SimGAN-V in terms of retaining backgrounds and bird poses of the source images as well as matching the text descriptions. For the flower dataset, SimGAN and SimGAN-V achieve comparable scores in maintaining flower shapes, while SimGAN-V performs slightly better in preserving background of source images and corresponding to what target text descriptions specify. In summary, the presented quantitative comparison results indicate that our proposed SimGAN and SimGAN-V outperform the baseline method in all aspects of quality of generated images for semantic image manipulation.

Interpolation results

If traversing in the learned latent space leads to semantic changes to the generated images (i.e. a smooth latent space), then we can reason that the model successfully learns relevant and useful representations, rather than simply memorising the training data [105]. Specifically for semantic image manipulation, in order to generate superior results, our proposed SimGAN is required to (implicitly) learn a good mapping between the relevant representations of the two modalities (i.e. images and texts). In this section, we investigate this by respectively interpolating between two representations of one modality while fixing representations of another modality. If in both cases the generated images demonstrate meaningful gradual changes, then it is reasonable to conclude that our model successfully learns a good mapping between image and text representations.

We first linearly interpolated the representations between two input source images, and fed them with a fixed representation of a target description to the trained generator of SimGAN. The source images were selected from the test set for zero-shot generation. As can be seen in Figure 4.6, our model is capable of generating realistic images that present continuous semantic changes, all of which correspond well to the target text description. For example, the first

row of Figure 4.6 shows two source images that have distinct backgrounds and bird poses. The corresponding generated images from the interpolated image representations demonstrates a smooth transition between the two original generated images, e.g. varying appearances of the branches, orientations and shapes of the bird, etc. Such smooth transition indicates that our model is able to learn useful and relevant representations of images for semantic image manipulation.

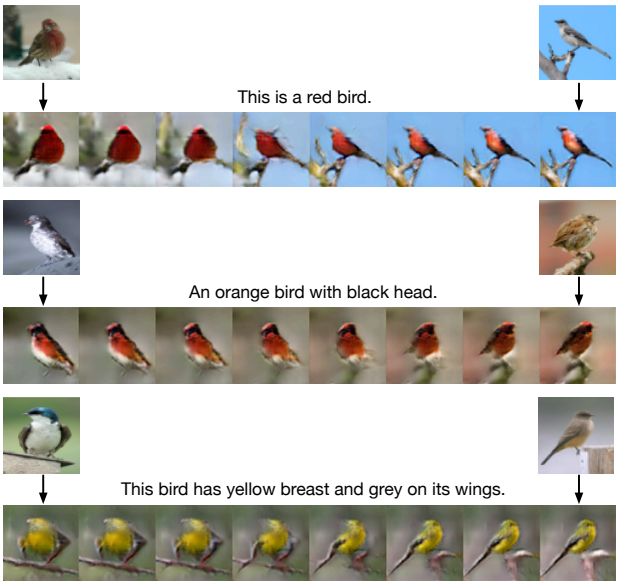


Figure 4.6.: Zero-shot generated images by linearly interpolating between representations of two source images with one target text description.

We then fed the linearly interpolated representations of two different target text descriptions and one fixed representation of a source image to SimGAN. The source images were again selected from the test set. As presented in Figure 4.7, the generated images from interpolated representations clearly demonstrate smooth variations between the semantics of the two target descriptions. Moreover, all the generated images not only remain realistic, but also maintain features of the input source image that are irrelevant to the target description. For in-

stance, the plausible interpolation results between "A yellow bird with black on wings." and "A bird with red head and breast." (shown in the second row of Figure 4.7) exhibit gradual changes of colours from yellow to red, with the appearance of orange as a result of yellow and red mixture. The irrelevant background and pose remain unchanged. This continuous transition indicates that relevant and useful text representations are learned by our model.

Since the interpolation results of both images and texts demonstrate smooth semantic changes, we can reason that our proposed SimGAN learns a good mapping between image and text representations.

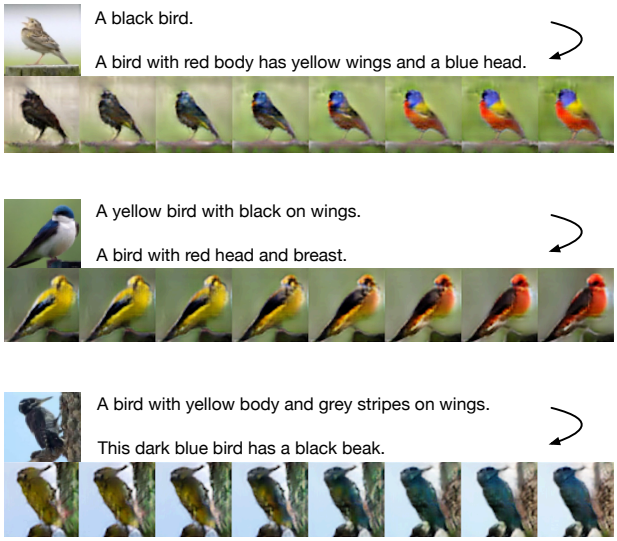


Figure 4.7.: Zero-shot generated images by linearly interpolating between representations of two target text descriptions with one source image.

Results of diverse generation

Given one source image and a target description, our proposed SimGAN is capable of generating diverse images, all of which meet the requirements of semantic image manipulation. As introduced in Section 4.4.5, this was achieved by applying the conditional augmenta-

tion approach [159], in which additional text representations can be sampled from the Gaussian distribution specified by the target text representations. Some examples can be seen in Figure 4.8.

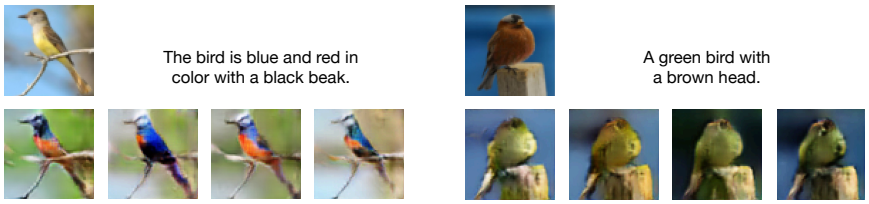


Figure 4.8.: Results of diverse generation from our proposed SimGAN on same source image and target description. All the generated images satisfy what semantic image manipulation requires.

Results of failure cases

We have demonstrated that our proposed SimGAN and SimGAN-V are able to generate superior results for semantic image manipulation. However, there still exist cases in which the generated images fail to meet the requirements of semantic image manipulation. Figure 4.9 illustrates some examples of failure cases for both bird and flower images. There are two main reasons for these failure examples. First, the backgrounds of input source images 1) contain unrecognised objects (e.g. the first image in Figure 4.9 (a)); 2) are too complicated (e.g. the second image in Figure 4.9 (b)); 3) have similar colours or textures, which mix up with the foreground objects (e.g. the third image in Figure 4.9 (c)). Second, the appearances of foreground objects are significantly different from those in training set (e.g. the second image in Figure 4.9 (d) and the third image in Figure 4.9 (h)). To generate better images in these cases, larger training datasets that contain more varied samples, or better captions that describe the images in more detail, are needed.

Generating higher-resolution images

Finally, we considered how to extend our proposed SimGAN architecture to generate higher-resolution images with finer details for semantic image manipulation. In our preliminary experiments, we attempted to directly replace the current networks with deeper ones. However, the model with such simple modification was unable to generate high-resolution images. The main reason is that the discriminator can distinguish the generated higher-resolution images from the real images more easily (especially at early training stage), which will severely degrade the stability of the learning process [95, 64].

To address this problem, we made two further modifications to our model. First, we proposed to employ an additional cycle loss [162, 65, 152] for the generator G and the encoder E:

$$\mathcal{L}_c(G, E) = \mathbb{E}_{(\mathrm{x},\bar{\mathrm{t}},\mathrm{t})\sim p_{<\mathrm{x},\bar{\mathrm{t}},\mathrm{t}>}(\mathrm{x},\bar{\mathrm{t}},\mathrm{t})}[\|G(E(\hat{\mathrm{x}}), \varphi(\mathrm{t})) - \mathrm{x}\|_2^2] \tag{4.5}$$

where $\hat{\mathrm{x}} = G(E(\mathrm{x}), \varphi(\bar{\mathrm{t}}))$ represents the generated image. By enforcing $G(E(\hat{\mathrm{x}}), \varphi(\mathrm{t}))$ to match x, the cycle loss significantly reduces the space of the generative function, resulting in a faster and easier learning process of G and E. Second, instead of using objectives of the vanilla GAN [41], we employed the least square loss of LSGAN [85], which is able to not only generate higher-quality images, but also make the learning process more stable.

Figure 4.10 presents the zero-shot results of 256 $\times$ 256 images for bird and flower datasets. The generated images not only satisfy the requirements of semantic image manipulation, but also contain much finer details than 64 $\times$ 64 images generated by the original SimGAN.

4.6. Conclusions

In this chapter, we proposed SimGAN, a novel conditional GAN based architecture for semantic image manipulation, which is a challenging

task of image-to-image generation that exhibits transformative creativity. We designed an innovative objective function, which enabled SimGAN to generate superior results that meet the requirements of semantic image manipulation. We further proposed to replace the original encoder in SimGAN with the (pretrained) deeper VGG network, resulting in the SimGAN-V method, in order to learn better image representations of various backgrounds. We utilised a pretrained text encoder for producing visual-semantic text representations. Since semantic image manipulation is essentially an one-to-many generation problem, an approach of conditioning augmentation on text representations was employed to generate diverse images.

We performed both qualitative and quantitative experiments to compare our proposed approaches with the baseline method, on Caltech-200 bird dataset and Oxford-102 flower dataset. The results indicated that our methods outperformed the baseline method, qualitatively and quantitatively, in all aspects of quality of generated images. SimGAN-V achieved slightly better results than those of SimGAN. We also demonstrated that our model was able to learn a good mapping between the image and text representations, as well as generate diverse images. Some examples of failure cases were presented, with possible reasons discussed. Finally, we demonstrated that our SimGAN architecture can be extended to generate compelling higher resolution images (of size 256×256).

In conclusion, our proposed SimGAN is able to generate superior results for semantic image manipulation. More importantly, the generated images not only have never existed before, but also show a clear difference from existing images. Such difference indicates that conditional GAN based architecture can learn to achieve high-level transformative creativity of image-to-image generation.

Author's contributions. This work was completed in collaboration with Mr. Hao Dong. I mainly contributed to the design of SimGAN and SimGAN-V (including their objective functions and network architectures), conducted the qualitative experiments for the flower

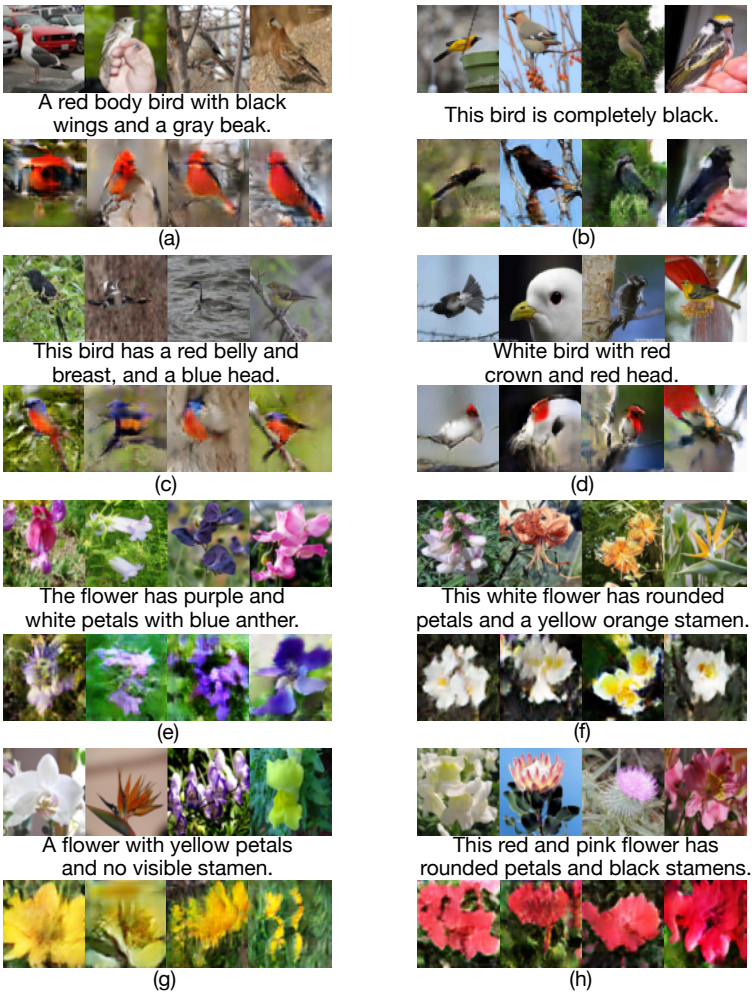


Figure 4.9.: Results of failure cases of SimGAN on images from the Caltech-200 bird dataset and the Oxford-102 flower dataset. (a-d) Bird images. (e-h) Flower images. In each section, the first row presents source images, and the second row presents the (failed) generated images.

dataset and the quantitative experiments on Amazon Mechanical Turk, and performed all the corresponding analysis. Mr. Hao Dong also contributed to the development of the proposed methods, implemented the comparison baseline method, and conducted the qualitative experiments for the bird dataset.

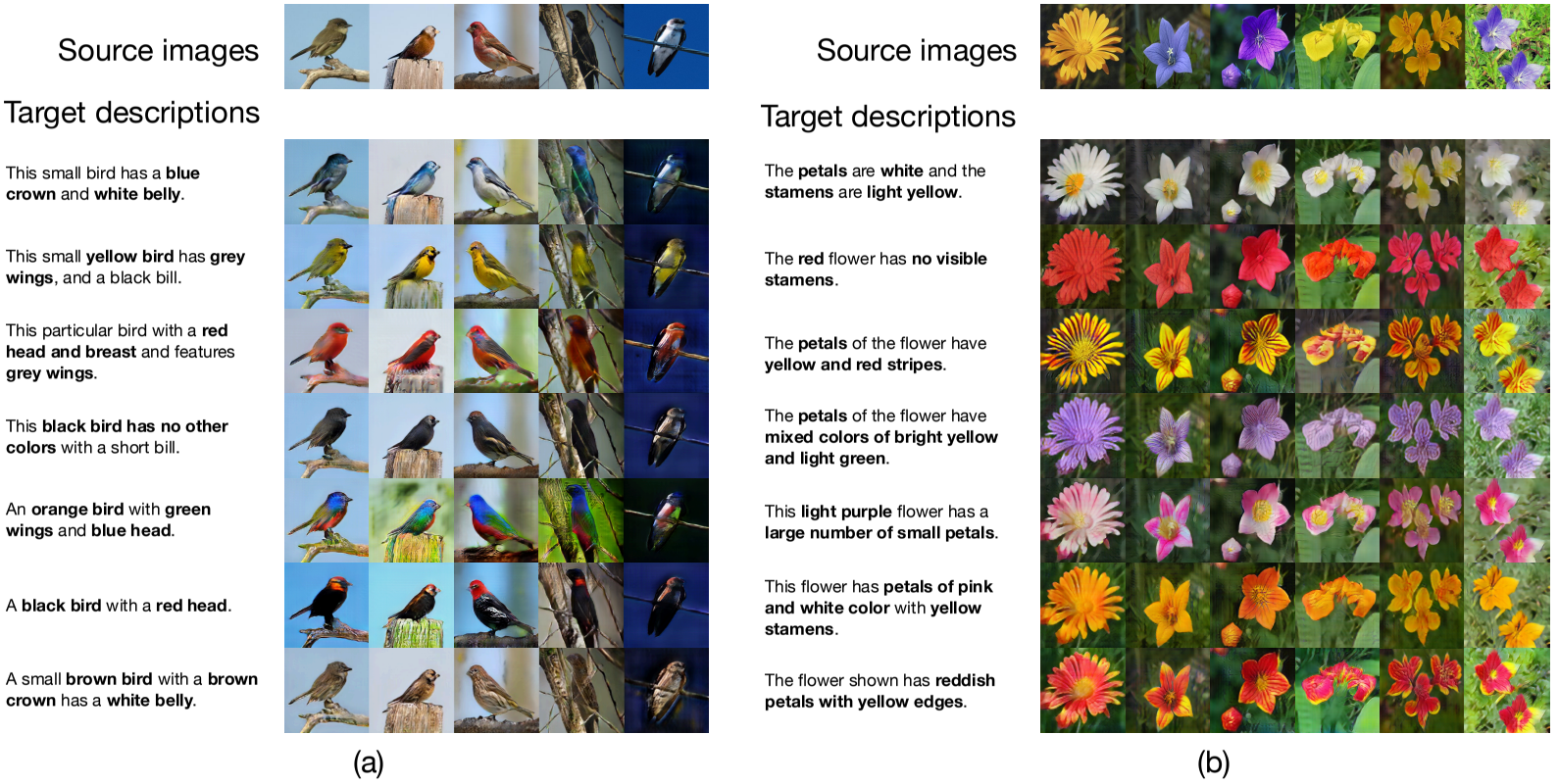


Figure 4.10.: Zero-shot generation results of 256 × 256 images on the test set of the Caltech-200 bird dataset and the Oxford-102 flower dataset. (a) Bird images. (b) Flower images.

5. Learning to Achieve Combinatorial Creativity

5.1. Motivation

Combinatorial creativity is defined as the generation of new and creative ideas by connecting existing ones, which is thought to be a crucially important pattern of innovation for human beings [87]. In computer vision, applications that exhibit combinatorial creativity involve the generation of novel images by composing representations of different domains. Specifically for image-to-image generation problems, one typical example is image style transfer [30, 50, 36], in which a creative image is generated by combining the content of a target image with the style of another artwork image.

In this chapter, we attempt to automate the process of bionic design [49, 123], which is a challenging image-to-image generation problem that demonstrates combinatorial creativity. Bionic design refers to a product design method, in which a biologically-inspired object is created by combining the representations of a target design object with those of biological source objects. Specifically in this work, we mainly focus on shape-oriented bionic design, which is the fundamental step in studying more general bionic design problems.

More specifically, given an input design target image, the generated images for bionic design should meet the following three requirements: 1) maintaining the shape features of the input image; 2) containing the shape features of images from the biological source domain; 3) remaining plausible and diverse. Figure 5.1 illustrates some examples

of bionic design.

Bionic design is a challenging image-to-image generation task due to the following three reasons. First, this task is of unsupervised learning, because the nature of its combinatorial creativity determines that there is no or very few available results of biologically-inspired designs. In our case, we only have unpaired samples of design target images and biological source images. Second, there should exist multiple ways of integrating features of biological source images into the given design target image. In other words, bionic design is essentially a one-to-many generation problem. This requires the generative model to be able to generate diverse results. Third, the generated biologically-inspired design should preserve key features of both input design target image and biological source images. Therefore, the generative model should be capable of learning and merging features from different sources to generate superior results.

To resolve the challenges mentioned above, we propose DesignGAN, an innovative conditional generative adversarial network (GAN) based architecture for bionic design. In particular, a novel objective function with multiple loss terms is designed to enforce the generated images to satisfy the requirements of bionic design. Specifically, the generator of DesignGAN takes as input both an image and a latent variable sampled from a prior Gaussian distribution, which enables the model to generate diverse output images. This is realised by the use of an explicit encoder and a latent loss. DesignGAN also employs a cycle loss [162, 65, 152] and a regression loss to maintain the key features of the design target image. Finally, an adversarial loss is employed to integrate the features of biological source images into the design target image.

We conduct both qualitative and quantitative experiments to show that our proposed model is capable of generating plausible and diverse biologically-inspired design images. Figure 5.1 (c) presents examples of 3D product modelling designed by a human designer, who is inspired by the generated creative bionic design images from our model.

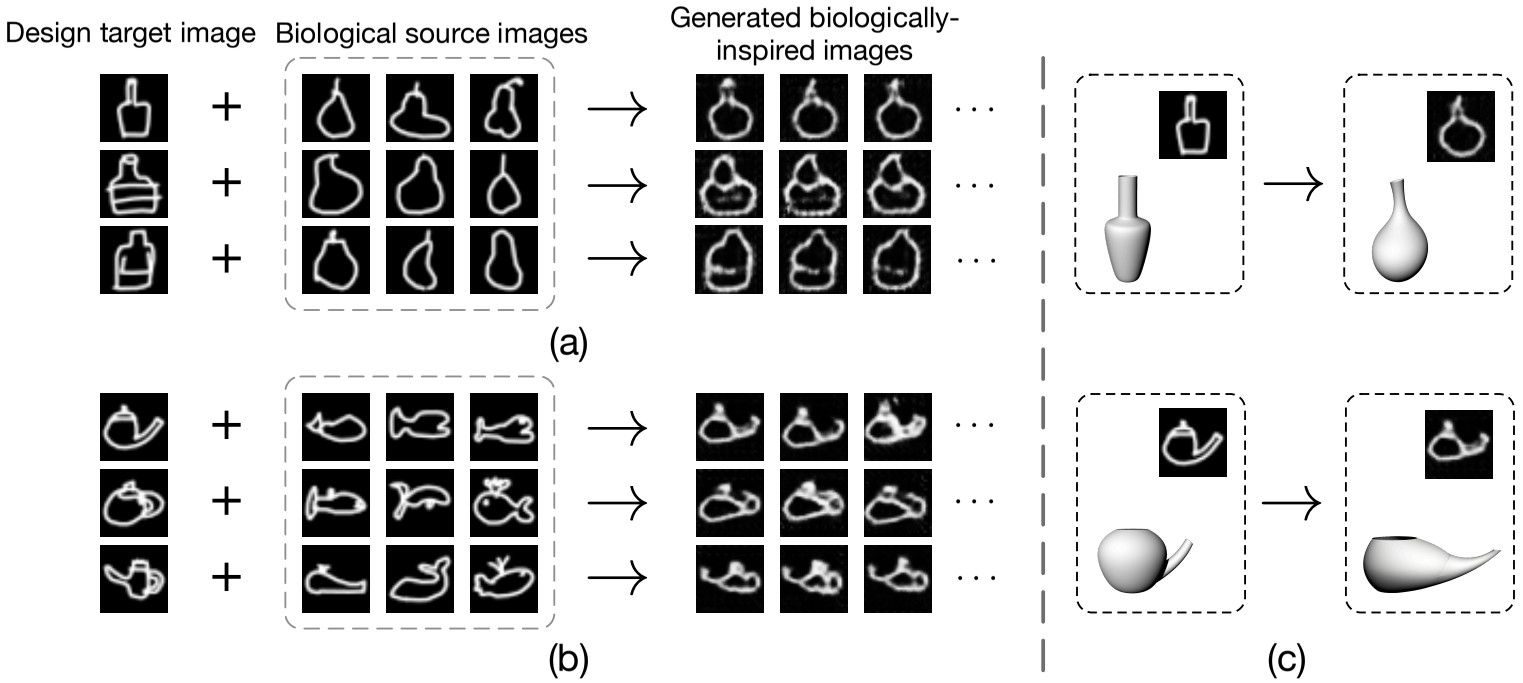


Figure 5.1.: Examples of bionic design that demonstrates combinatorial creativity. Given a design target image and biological source images, our proposed model generates biologically-inspired images, which should 1) retain features of the design target image; 2) contain features of biological source images; 3) be plausible and varied. (a) Wine bottle + pear generates pear-like bottles. (b) Teapot + whale generates whale-shaped teapots. (c) Examples of 3D product modelling designed by a human designer, who is inspired by the generated creative images from our model.

The rest of this chapter is organised as follows. We first formulate the problem of bionic design to be addressed in this work in Section 5.2. We then describe the baseline methods and our proposed methods in detail in Section 5.3. After that, we demonstrate and discuss the results of experiments in Section 5.4. Finally, we conclude our work in Section 5.5.

5.2. Problem formulation

The problem of bionic design can be formulated as follows. Let D be the domain of design target images (e.g. floor lamps), and B be the domain of biological source images (e.g. flowers). We are given the (unpaired) training samples $\{\mathrm{d}_k\}_{k=1}^{M} \in \mathrm{D}$ and $\{\mathrm{b}_k\}_{k=1}^{N} \in \mathrm{B}$, which are respectively sampled from the data distributions $p_{\mathrm{d}}(\mathrm{d})$ and $p_{\mathrm{b}}(\mathrm{b})$ (the subscript k are omitted for convenience). We further denote the corresponding latent representation space of D and B as $\mathrm{Z_d}$ and $\mathrm{Z_b}$, respectively.

We then make two key assumptions of the bionic design problem as follows:

1. there exists an "intermediate" domain I that contains samples of the biologically-inspired design images $\mathrm{i} \in \mathrm{I}$ (e.g. flower-like floor lamps);

2. the representations of i in the latent space of I (denoted as Z) are formed by a combination of those in $\mathrm{Z_d}$ and $\mathrm{Z_b}$.

These two assumptions are illustrated in Figure 5.2.

Based on these two assumptions, to address the bionic design problem, our goal is to learn a generative function $G_{\mathrm{DB}} : \mathrm{D} \times \mathrm{Z} \rightarrow \mathrm{I}$, such that the generative distribution $p_G(\mathrm{d}, \mathrm{z})$ matches the distribution of biologically-inspired design images (denoted as $p_{\mathrm{i}}(\mathrm{i})$). Specifically in this work, there is no existing data of biologically-inspired design images. Therefore, it is impossible to explicitly learn such generative

distribution. Nonetheless, we could still be able to learn it implicitly via the empirical data distributions $p_{\mathrm{d}}(\mathrm{d})$ and $p_{\mathrm{b}}(\mathrm{b})$ with a good design of the model architecture. It is such implicit learning process of generating creative images that gives rise to combinatorial creativity.

Since G_{DB} takes as input the latent variable $\mathrm{z} \in \mathrm{Z}$ sampled from the distribution $p_{\mathrm{z}}(\mathrm{z})$, the requirement of diversity for bionic design is naturally satisfied. By sampling different z from $p_{\mathrm{z}}(\mathrm{z})$, multiple results can be generated for one input design target image d.

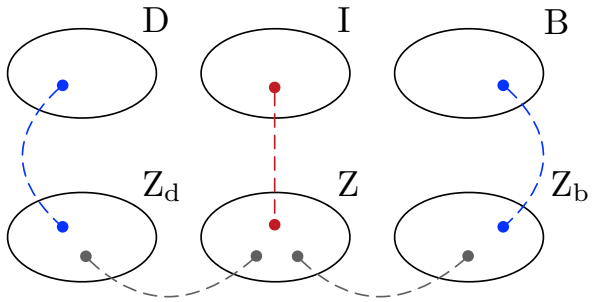


Figure 5.2.: The assumption of the bionic design problem. We assume that there exists an intermediate domain I that contains images of biologically-inspired design, whose representations (in latent space Z) are formed by the combination of those of design target images (in latent space $\mathrm{Z_d}$ of D) and biological source images (in latent space $\mathrm{Z_b}$ of B).

5.3. Method

At first glance, the shape-oriented bionic design problem could be tackled by employing the CycleGAN architecture [162, 65, 152]. However, we reveal the limitation of CycleGAN for this problem, which motivates us to develop new architectures. In this section, we start with a brief discussion on the applicability of CycleGAN model and its extensions. We then describe in detail our proposed DesignGAN model and the designed objective function. We finally introduce the network architecture of DesignGAN.

5.3.1. A path of evolution from CycleGAN

CycleGAN

Our initial attempt to address the bionic design problem is to directly employ the CycleGAN architecture. Figure 5.3 presents the schema of CycleGAN for bionic design. It includes two generators (G_{DB} and G_{BD}) and two discriminators (D_{B} and D_{D}). G_{DB} takes as input a design target image d, and outputs a biologically-inspired image $\hat{\mathrm{i}}_{\mathrm{DB}}$. An adversarial loss $\mathcal{L}_a^{\mathrm{DB}}$ from D_{B} is used to integrate the features of biological source images into $\hat{\mathrm{i}}_{\mathrm{DB}}$. A cycle loss $\mathcal{L}_c^{\mathrm{DB}}$ is further applied to maintain the features of the input d, by enforcing the reconstruction $\tilde{\mathrm{d}} = G_{\mathrm{BD}}(\hat{\mathrm{i}}_{\mathrm{DB}})$ to be close to the original input d. The use of the cycle loss requires joint training of two image-based conditional GANs.

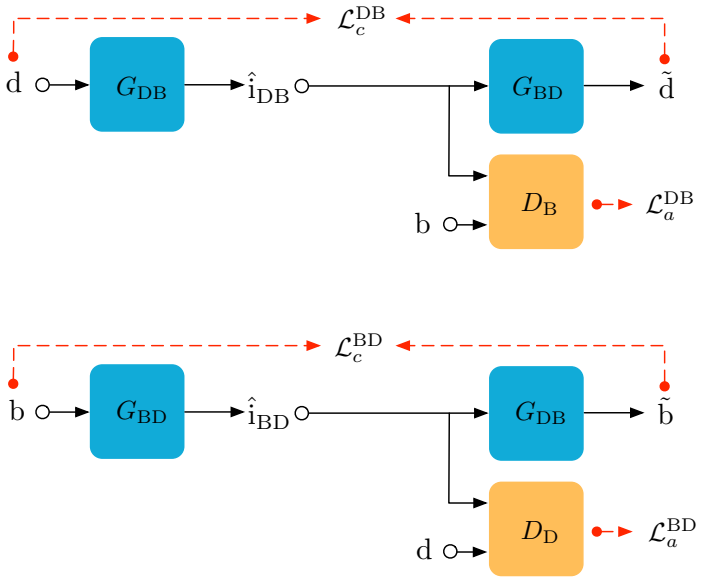


Figure 5.3.: Schema of CycleGAN architecture for bionic design.

Since the images only contain shape representations, these two losses will directly compete with each other, which makes G_{DB} possible to generate images that contain combined shape features of the input design target image and biological source images. However, CycleGAN

only learns a deterministic generation (i.e. one-to-one mapping) and is not able to generate diverse results.

CycleGAN+N

A straightforward way of enabling the CycleGAN architecture to learn one-to-many mapping is to directly provide the generators with noise input. We denote this method as CycleGAN+N, whose schema can be seen in Figure 5.4. In this approach, the generator G_{DB} takes as input both a design target image d and a noise vector z sampled from a prior distribution (e.g. a Gaussian). At inference stage, the trained G_{DB} could generate diverse biologically-inspired images, given one d and different sampled z.

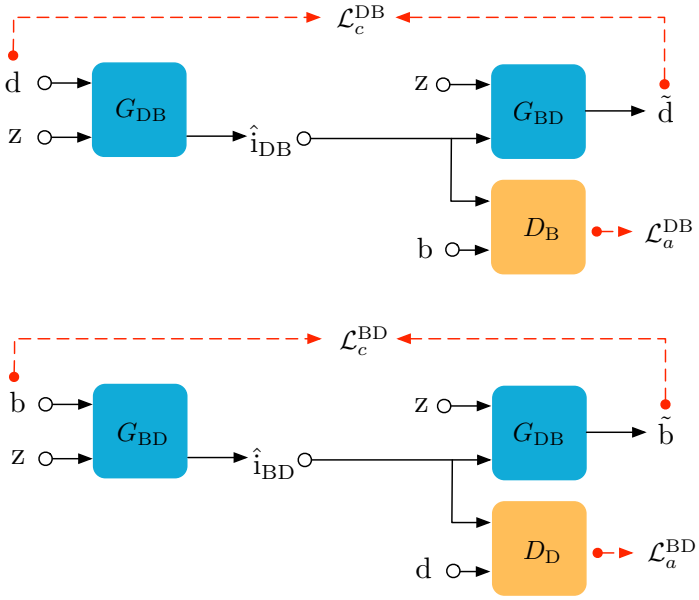


Figure 5.4.: Schema of CycleGAN+N architecture for bionic design. Compared with CycleGAN, the generators in CycleGAN+N are provided with noise input to generate diverse results.

However, as also mentioned in [2, 56], the employed cycle-consistency restriction would make the generators ignore the input noise vector z.

Take the cycle loss $\mathcal{L}_c^{DB}$ as an example. The generator G_{BD} takes as input the (assumed diverse) generated images $\hat{\mathrm{i}}_{\mathrm{DB}}$ and arbitrary noise vector z, and outputs reconstructed images $\tilde{\mathrm{d}}$. Since $\tilde{\mathrm{d}}$ is encouraged to approximate the same d via $\mathcal{L}_c^{DB}$, G_{BD} is then enforced to learn many-to-one mapping, without necessarily taking z into account. Similarly, the cycle loss $\mathcal{L}_c^{BD}$ enforces G_{DB} to learn many-to-one mapping only. Consequently, G_{DB} and G_{BD} will only learn to generate deterministic output images, as similar to CycleGAN.

CycleGAN+2E

To address the limitation of CycleGAN+N, we further propose a new architecture with two explicit encoders E_{D} and E_{B}. We denote this method as CycleGAN+2E, with its schema demonstrated in Figure 5.5. Each encoder takes as input a generated image and encodes it back into a corresponding latent code. The generated latent code will then be encouraged to approximate the input noise vector via a latent loss. Given one design target image d and different noise vectors $\mathrm{z_b}$, the use of the latent loss enforces the generator G_{DB} to generate diverse biologically-inspired images $\hat{\mathrm{i}}_{\mathrm{DB}}$. Compared with CycleGAN+N, CycleGAN+2E will never ignore the input noise vector, due to the use of the explicit encoders.

Nonetheless, CycleGAN+2E has its limitation as well: the generated images tend to depend on the input noise vector only, without taking the input image into account. This is also due to the employed cycle-consistency restriction. More specifically, for the cycle loss $\mathcal{L}_c^{\mathrm{DB}}$, the generator G_{BD} takes as input diverse generated images $\hat{\mathrm{i}}_{\mathrm{DB}}$, as well as the generated latent code $\hat{\mathrm{z}}_{\mathrm{d}}$. Note that $\hat{\mathrm{z}}_{\mathrm{d}}$ will always be a fixed one encoded by the encoder E_{D}, given the same design target image d as the input. The reconstructed image $\tilde{\mathrm{d}}$ also approximates the same d due to $\mathcal{L}_c^{\mathrm{DB}}$. As a result, G_{BD} will learn one-to-one mapping only, between $\hat{\mathrm{z}}_{\mathrm{d}}$ and $\tilde{\mathrm{d}}$. Similarly, the cycle loss $\mathcal{L}_c^{\mathrm{BD}}$ will enforce G_{DB} to learn one-to-one mapping between $\hat{\mathrm{z}}_{\mathrm{b}}$ and $\tilde{\mathrm{b}}$. In this case, the input noise vector

dominates, while the input images are ignored by the generators. In other words, G_{BD} and G_{DB} will only be able to generate results that share no relationship with the input images.

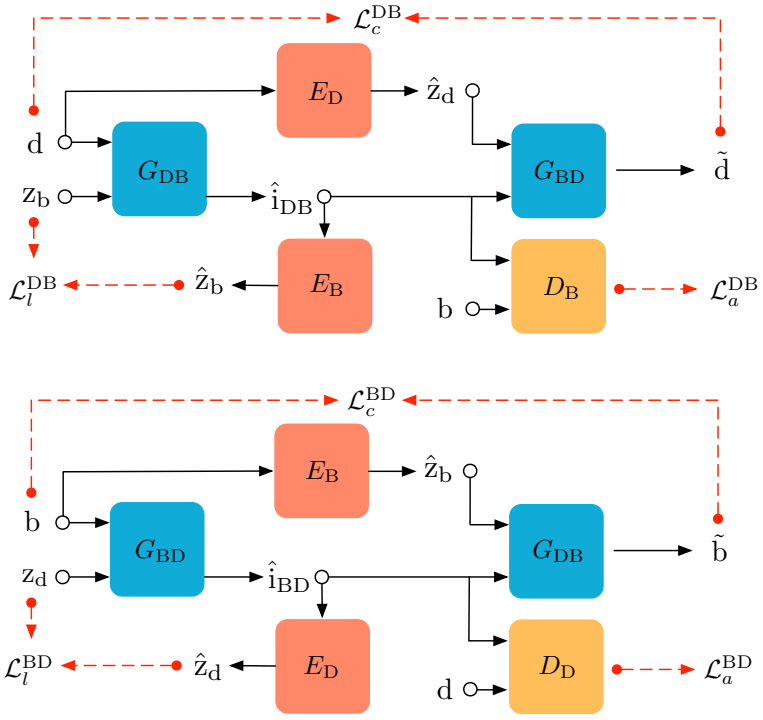


Figure 5.5.: Schema of CycleGAN+2E architecture for bionic design. Compared with CycleGAN+N, CycleGAN+2E employs two explicit encoders with the latent loss to ensure that the input noise will not be ignored by the model.

5.3.2. DesignGAN

To address the limitation of CycleGAN+2E as discussed above, we further propose DesignGAN architecture for the bionic design problem. Figure 5.6 demonstrates the overall schema of DesignGAN. Specifically, it consists of five networks: two generators (G_{DB} and G_{BD}), two discriminators (D_{B} and D_{D}) and one encoder (E). We design a novel objective function with four designated losses, which will be discussed

in detail as follows. DesignGAN is an end-to-end architecture: all the component networks are trained jointly from scratch.

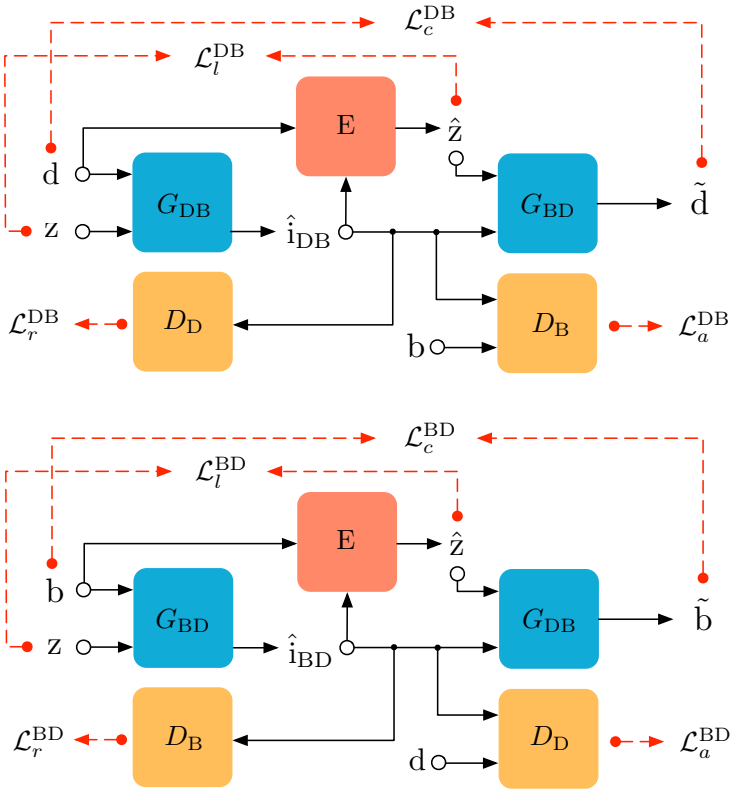


Figure 5.6.: Schema of our proposed DesignGAN architecture for bionic design. It is composed of two generators (G_{DB} and G_{BD}), two discriminators (D_{B} and D_{D}) and one encoder (E). DesignGAN has two enhancements on CycleGAN+2E: 1) it employs a single encoder E as the encoding function to learn the variation for bionic design, and 2) it reuses the discriminators D_{D} and D_{B} as forward regression functions to further preserve the representations of the input image domain via the regression loss ($\mathcal{L}_r^{\mathrm{DB}}$ and $\mathcal{L}_r^{\mathrm{BD}}$). Other loss components include the adversarial loss ($\mathcal{L}_a^{\mathrm{DB}}$ and $\mathcal{L}_a^{\mathrm{BD}}$), the cycle loss ($\mathcal{L}_c^{\mathrm{DB}}$ and $\mathcal{L}_c^{\mathrm{BD}}$) and the latent loss ($\mathcal{L}_l^{\mathrm{DB}}$ and $\mathcal{L}_l^{\mathrm{BD}}$). At inference stage, G_{DB} is used to generate varied biologically-inspired images, given an input design target image d and an input noise vector z.

Adversarial loss

One requirement of the bionic design problem is that the biologically-inspired images to be generated should include the features of biological source images. To satisfy this requirement, we employ an adversarial loss [41] $\mathcal{L}_a^{\mathrm{DB}}$ for G_{DB} and D_{B}, in which D_{B} attempts to discriminate between the generated images $\hat{\mathrm{i}}_{\mathrm{DB}}$ and the biological source images b, while G_{DB} attempts to generate indistinguishable results from b. This adversarial process acts as an approach to integrating representations of biological objects into $\hat{\mathrm{i}}_{\mathrm{DB}}$. A similar adversarial loss $\mathcal{L}_a^{\mathrm{BD}}$ is applied for G_{BD} and D_{D}, which will be employed to compute other losses (detailed later). The overall adversarial loss is defined as follows:

$$
\begin{aligned}
\mathcal{L}_a(G_{\mathrm{DB}}, G_{\mathrm{BD}}, D_{\mathrm{B}}, D_{\mathrm{D}}) &= \mathcal{L}_a^{\mathrm{DB}}(G_{\mathrm{DB}}, D_{\mathrm{B}}) + \mathcal{L}_a^{\mathrm{BD}}(G_{\mathrm{BD}}, D_{\mathrm{D}}), \\
\mathcal{L}_a^{\mathrm{DB}}(G_{\mathrm{DB}}, D_{\mathrm{B}}) &= \mathbb{E}_{\mathrm{b}\sim p_{\mathrm{b}}(\mathrm{b})}[\log D_{\mathrm{B}}(\mathrm{b})] \\
&\quad + \mathbb{E}_{\mathrm{d}\sim p_{\mathrm{d}}(\mathrm{d}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\log(1 - D_{\mathrm{B}}(G_{\mathrm{DB}}(\mathrm{d}, \mathrm{z})))], \\
\mathcal{L}_a^{\mathrm{BD}}(G_{\mathrm{BD}}, D_{\mathrm{D}}) &= \mathbb{E}_{\mathrm{d}\sim p_{\mathrm{d}}(\mathrm{d})}[\log D_{\mathrm{D}}(\mathrm{d})] \\
&\quad + \mathbb{E}_{\mathrm{b}\sim p_{\mathrm{b}}(\mathrm{b}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\log(1 - D_{\mathrm{D}}(G_{\mathrm{BD}}(\mathrm{b}, \mathrm{z})))].
\end{aligned}
\tag{5.1}
$$

where $p_{\mathrm{d}}(\mathrm{d})$ and $p_{\mathrm{b}}(\mathrm{b})$ respectively represent the empirical data distribution of design target images and biological source images, z represents the input noise vector sampled from a prior distribution $p_{\mathrm{z}}(\mathrm{z})$ (a Gaussian distribution is employed in this work).

Cycle loss

The bionic design problem also requires the generated biologically-inspired images to maintain the (shape) representations of the input design target image. One straightforward way to meet such requirement is to enforce the generated images to resemble the input design target image to a large extent. This enables the generated images to be recognised as the same class of the design target (e.g. the generated whale-shaped teapots should still remain as teapots).

We apply a cycle loss [162, 65, 152] to achieve this goal. Specifically, given a generated biologically-inspired image $\hat{\mathrm{i}}_{\mathrm{DB}} = G_{\mathrm{DB}}(\mathrm{d}, \mathrm{z})$, the cycle loss encourages the reconstructed image $\tilde{\mathrm{d}} = G_{\mathrm{BD}}(\hat{\mathrm{i}}_{\mathrm{DB}}, \hat{\mathrm{z}})$ to be close to the input design target image d. Such cycle consistency will ensure that G_{DB} is able to preserve the shape representations of the input image.

The use of the cycle loss introduces an auxiliary generator G_{BD} with a discriminator D_{D}, which will be trained along with G_{DB} and D_{B}, resulting in a dual-learning process for DesignGAN. After training, only the trained G_{DB} will be used for the bionic design problem. The overall cycle loss is defined as below:

$$\begin{aligned}
\mathcal{L}_c(G_{\mathrm{DB}}, G_{\mathrm{BD}}) &= \mathcal{L}_c^{\mathrm{DB}}(G_{\mathrm{DB}}, G_{\mathrm{BD}}) + \mathcal{L}_c^{\mathrm{BD}}(G_{\mathrm{BD}}, G_{\mathrm{DB}}), \\
\mathcal{L}_c^{\mathrm{DB}}(G_{\mathrm{DB}}, G_{\mathrm{BD}}) &= \mathbb{E}_{\mathrm{d}\sim p_{\mathrm{d}}(\mathrm{d}), \mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\|G_{\mathrm{BD}}(G_{\mathrm{DB}}(\mathrm{d}, \mathrm{z}), E(G_{\mathrm{DB}}(\mathrm{d}, \mathrm{z}), \mathrm{d})) - \mathrm{d}\|_2^2], \\
\mathcal{L}_c^{\mathrm{BD}}(G_{\mathrm{BD}}, G_{\mathrm{DB}}) &= \mathbb{E}_{\mathrm{b}\sim p_{\mathrm{b}}(\mathrm{b}), \mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\|G_{\mathrm{DB}}(G_{\mathrm{BD}}(\mathrm{b}, \mathrm{z}), E(\mathrm{b}, G_{\mathrm{BD}}(\mathrm{b}, \mathrm{z}))) - \mathrm{b}\|_2^2].
\end{aligned} \tag{5.2}$$

Regression loss

The cycle loss imposes a pixel-wise constraint to preserve the (shape) representations of a given design target image. However, when applying such pixel-wise loss alone, it would be difficult to adjust its relative strength (to that of the adversarial loss), for generating images that satisfy the requirements of bionic design. The generated images may either fail to include enough biological details (when the pixel-wise constraint is too strong), or be unable to resemble the design target image (when the constraint is too weak). Intuitively, this is because different parts of the input image are of different importance (e.g. some contain salient features of the design target to be maintained while others contain nothing useful and should be replaced by biological features), which is not taken into account by the pixel-wise constraint.

To this end, we propose to apply another feature-wise constraint (simultaneously along with the cycle loss) to maintain the design target

representations. More specifically, we reuse the discriminator D_{D} as a regression function to impose a regression loss L_r^{DB} for the generator G_{DB}, such that the generated image $\hat{\mathrm{i}}_{\mathrm{DB}}$ will contain the design target representations, and will be recognised as the design target class. A similar regression loss L_r^{BD}, imposed by the discriminator D_{B}, is applied for G_{BD}.

It is worth noting that the regression loss requires D_{D} and D_{B} to be regression functions only, in which case D_{D} and D_{B} will not compete with the corresponding generators. This is different from the adversarial loss. This difference can be seen in Figure 5.6, in which there is only one input to D_{D} and D_{B} for the regression loss L_r^{DB} and L_r^{BD}, while there are two inputs to be discriminated by the discriminator for the adversarial loss. The overall regression loss is defined as follows:

$$\begin{aligned}
\mathcal{L}_r(G_{\mathrm{DB}}, G_{\mathrm{BD}}) &= \mathcal{L}_r^{\mathrm{DB}}(G_{\mathrm{DB}}) + \mathcal{L}_r^{\mathrm{BD}}(G_{\mathrm{BD}}), \\
\mathcal{L}_r^{\mathrm{DB}}(G_{\mathrm{DB}}) &= \mathbb{E}_{\mathrm{d}\sim p_{\mathrm{d}}(\mathrm{d}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\log(1 - D_{\mathrm{D}}(G_{\mathrm{DB}}(\mathrm{d},\mathrm{z})))], \\
\mathcal{L}_r^{\mathrm{BD}}(G_{\mathrm{BD}}) &= \mathbb{E}_{\mathrm{b}\sim p_{\mathrm{b}}(\mathrm{b}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\log(1 - D_{\mathrm{B}}(G_{\mathrm{BD}}(\mathrm{b},\mathrm{z})))].
\end{aligned} \tag{5.3}$$

Latent loss

Finally, to learn one-to-many mapping for the bionic design problem, we propose to use one explicit encoder E with a latent loss $\mathcal{L}_l$ in the DesignGAN architecture. For the generator G_{DB} with its input d and z, E encodes the generated biologically-inspired image $\hat{\mathrm{i}}_{\mathrm{DB}}$ (along with d) into the latent code $\hat{\mathrm{z}}$ in the latent space Z of domain I. The latent loss $\mathcal{L}_l^{\mathrm{DB}}$ is then computed by the l_1 distance between $\hat{\mathrm{z}}$ and z, which enforces G_{DB} to generate diverse results. Similarly, the latent loss $\mathcal{L}_l^{\mathrm{BD}}$ is imposed on the generator G_{BD}. Compared with CycleGAN+N and CycleGAN+2E, the use of E and the latent loss in DesignGAN ensures that both the input image and the input noise vector will be taken into account by the generator.

It is noteworthy that the encoder E in DesignGAN has two differences from the encoders in CycleGAN+2E. First, only one E is applied for both latent loss $\mathcal{L}_l^{\mathrm{DB}}$ and $\mathcal{L}_l^{\mathrm{BD}}$. Second, apart from the generated

images ($\hat{\mathrm{i}}_{\mathrm{DB}}$ or $\hat{\mathrm{i}}_{\mathrm{BD}}$), the input image to the generators (d or b) is also fed to E. These two differences make E correspond to our assumption of the bionic design problem (as described in Section 5.2), which is critical for the generation of superior biologically-inspired design results. The overall latent loss is defined as below:

$$\begin{aligned}
\mathcal{L}_l(G_{\mathrm{DB}}, G_{\mathrm{BD}}, E) &= \mathcal{L}_l^{\mathrm{DB}}(G_{\mathrm{DB}}, E) + \mathcal{L}_l^{\mathrm{BD}}(G_{\mathrm{BD}}, E), \\
\mathcal{L}_l^{\mathrm{DB}}(G_{\mathrm{DB}}, E) &= \mathbb{E}_{\mathrm{d}\sim p_{\mathrm{d}}(\mathrm{d}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\|E(G_{DB}(\mathrm{d},\mathrm{z}),\mathrm{d}) - \mathrm{z}\|_1], \\
\mathcal{L}_l^{\mathrm{BD}}(G_{\mathrm{BD}}, E) &= \mathbb{E}_{\mathrm{b}\sim p_{\mathrm{b}}(\mathrm{b}),\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\|E(\mathrm{b}, G_{\mathrm{BD}}(\mathrm{b},\mathrm{z})) - \mathrm{z}\|_1].
\end{aligned} \tag{5.4}$$

Full objective

The full objective function of our proposed DesignGAN is:

$$\begin{aligned}
\min_{G_{\mathrm{DB}},G_{\mathrm{BD}},E}\max_{D_{\mathrm{B}},D_{\mathrm{D}}} \mathcal{L}(G_{\mathrm{DB}}, G_{\mathrm{BD}}, E, D_{\mathrm{B}}, D_{\mathrm{D}}) = \lambda_a\mathcal{L}_a(G_{\mathrm{DB}}, G_{\mathrm{BD}}, D_{\mathrm{B}}, D_{\mathrm{D}}) + \\
\lambda_c\mathcal{L}_c(G_{\mathrm{DB}}, G_{\mathrm{BD}}) + \lambda_r\mathcal{L}_r(G_{\mathrm{DB}}, G_{\mathrm{BD}}) + \lambda_l\mathcal{L}_l(G_{\mathrm{DB}}, G_{\mathrm{BD}}, E),
\end{aligned} \tag{5.5}$$

where λ_a, λ_c, λ_r and λ_l respectively controls the relative strength of the adversarial loss, the cycle loss, the regression loss and the latent loss.

5.3.3. Network architecture

In this section, we describe the network architecture of our proposed DesignGAN in brief as below. A more detailed visualisation of network architecture can be seen in Figure D.1 in Appendix D.

The generator G_{DB} and G_{BD}

For the generator networks, we adopt the encoder-decoder architecture. The latent vector z is first spatially replicated and concatenated to the input image (d or b), resulting in the input to the encoder. The encoder consists of three convolutional layers and the decoder is composed by two transposed convolutional layers, each of which is followed by a

layer of batch normalisation [58] and ReLU. Six residual blocks [48] are applied after the encoder. Each residual block is made up of two groups of a convolutional layer, batch normalisation and ReLU. The residual blocks act as a deeper encoding process, from which better representations of the input image can be obtained. We employ the same architecture for the generator G_{DB} and G_{BD}.

The discriminator D_{B} and D_{D}

The discriminator consists of four convolutional layers. Each convolutional layer (except the first one) is followed by a layer of batch normalisation and leaky-ReLU [83]. This design of architecture is adopted for both discriminators D_{B} and D_{D}.

The encoder E

For the encoder network, the two input images are concatenated as the input to three convolutional layers, each of which followed by a layer of batch normalisation and ReLU. The output of the convolutional layers is then fed into six residual units (of same structure as the one used in the generator).

5.4. Experiments

5.4.1. Experimental settings

Methods

In our experiments, we evaluated and compared all the methods discussed in Section 5.3, including CycleGAN, CycleGAN+N, our proposed CycleGAN+2E and DesignGAN. We adopted the same network architecture (described in Section 5.3.3) in all methods for a fair comparison.

Dataset

We evaluated the methods on the "Quick, Draw!" dataset [45], which contains millions of simple grayscale drawings (of size 28×28) across 345 common objects. It is an ideal dataset for the shape-oriented bionic design problem. We selected several pairs of domains of design targets and biological sources as the varied bionic design problems. We randomly chose 4000 images from each domain of the domain pairs for training.

Training details

The networks were trained for 120 epochs using Adam optimiser [66], with a learning rate of 0.0001 and a batch size of 64. The learning rate was decayed to zero linearly over the last half number of epochs. Due to the distinct complexity of images from different domains, the values of λ_a, λ_c, λ_r and λ_l, as well as the dimension of the latent variable z, were set independently for each specific bionic design problem.

In order to stabilise the adversarial learning process, we replaced the adversarial loss (of the vanilla GAN) by the objectives of least squares GAN [85] for the generators and the discriminators. Moreover, the discriminators were updated using a history of generated images, as proposed in [122], which further made the training process more stable [162].

We applied random horizontal flipping and random ±15 degree rotation to the training images. These images were further resized to 32×32 before being fed into the models. TensorFlow [1] and TensorLayer [23] were used to implement the methods.

5.4.2. Results

Qualitative results

Figure 5.7 presents the qualitative results of the four comparison methods for bionic design. In each group of generation, the values of the in-

put noise vector (to CycleGAN+N, CycleGAN+2E and DesignGAN), used in the corresponding column of generated results for each input design target image, were maintained same.

Specifically, CycleGAN is able to generate reasonable biologically-inspired results in some cases (e.g. Figure 5.7 (a) left and (f) left). However, in other cases, it fails to maintain the features of the input design target image (e.g. Figure 5.7 (d) and (f) right). Also, since CycleGAN learns a one-to-one generation only, there is no diversity in the generated results.

For the CycleGAN+N architecture, the generated images are similar to those of CycleGAN. More importantly, we observe that in most cases the input noise vector to the model does not result in generation of diverse results. This indicates that CycleGAN+N ignores the input noise vector to a large degree, which corresponds to what we discussed in Section 5.3.1.

As for our proposed CycleGAN+2E architecture, it is capable of generating varied images. However, the images are either of low-quality (e.g. Figure 5.7 (b) and (c)), or unable to preserve any representations of the input design target image (e.g. Figure 5.7 (a) and (f)). This implies that the input noise vector dominates, while the design target image is largely ignored by CycleGAN+2E, which again corresponds to our analysis in Section 5.3.1.

By contrast, our proposed DesignGAN is capable of generating plausible and diverse biologically-inspired images that satisfy the requirements of bionic design. This demonstrates that, unlike CycleGAN+N and CycleGAN+2E, DesignGAN takes both the input image and the input noise vector into account for generation. Furthermore, the qualitative results indicate that DesignGAN can generate creative images that not only never existed before, but also show clear distinction from the existing images. This is done by combining the representations of images from two domains in a novel manner. In other words, DesignGAN successfully learns to achieve high-degree combinatorial creativity of image-to-image generation.

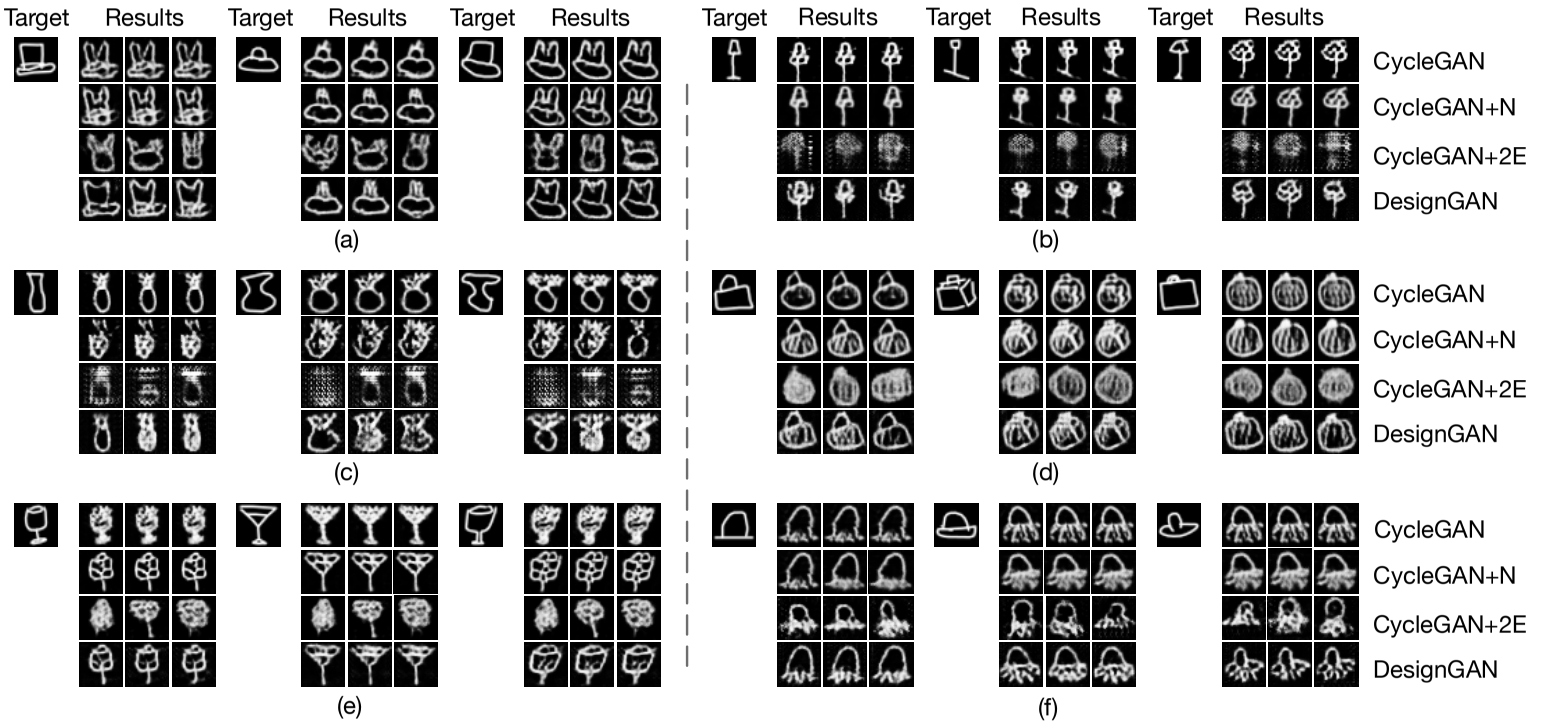


Figure 5.7.: Qualitative comparison results between investigated and our proposed approaches for the shape-oriented bionic design problem. For each input design target image, three biologically-inspired images are generated by each comparison method. (a) Hat + rabbit. (b) Floor Lamp + flower. (c) Vase + pineapple. (d) Suitcase + onion. (e) Wine glass + flower. (f) Hat + octopus.

Quantitative results

How to quantitatively evaluate the performance of generative models for creative tasks remains a challenging problem. In this work, we utilised human judgement to evaluate the investigated and proposed methods for bionic design. Despite subjective factors, it would be the most dependable measurement of creativity and plausibility of generated results.

More specifically, we performed the human evaluation on Amazon Mechanical Turk (AMT). We selected eight domain pairs of design targets and biological sources shown in this paper. For each pair of domains, we used three design target images as the input to the comparison models. For each input image, three biologically-inspired design images were generated by each of the comparison methods. There were twenty subjects recruited, presented all the input and generated images, and required to rank the models, according to the following criteria:

1. whether the generated images contained the key features of input design target images;
2. whether the generated images contained the key features of biological source images;
3. whether the generated images were diverse;
4. whether the generated images were creative and plausible.

Subjects scored 1 for the best and scored 4 for the worst (expect for the evaluation of diversity, in which CycleGAN was not compared and subjects scored 3 for the worst, as CycleGAN is a deterministic model and cannot generate diverse results). We then averaged all the ranking scores over the different criteria.

Table 5.1 presents the human evaluation results of the comparison methods for the bionic design problem. Specifically, all approaches achieve a comparable ranking score in integrating features of biological source images (CycleGAN and DesignGAN perform better than CycleGAN+N and CycleGAN+2E). In terms of maintaining the fea-

tures of design target images, DesignGAN performs better than other comparison methods (CycleGAN+2E obtains very poor result in this criterion). For diversity, CycleGAN+2E achieve better rankings than DesignGAN and CycleGAN+N, though it generally fails to preserve any features of design target images. In the aspect of creativity and plausibility, DesignGAN obtains the best ranking score, which indicates that it is capable of generating the most creative and plausible bionic designs. When all the judging criteria are considered, DesignGAN outperforms other comparison approaches.

Table 5.1.: Comparison results of human evaluation between the investigated and our proposed approaches for the bionic design problem. The scores represent the averaged human ranks (1 for the best) according to four specific criteria.

	CycleGAN	CycleGAN+N	CycleGAN+2E	DesignGAN
Design target features	1.97 ± .08	2.65 ± .09	3.57 ± .07	1.80 ± .08
Biological features	2.28 ± .09	2.70 ± .09	2.73 ± .11	2.29 ± .09
Diversity	/	2.44 ± .07	1.55 ± .07	2.01 ± .06
Creativity & plausibility	2.04 ± .08	2.62 ± .10	3.36 ± .09	1.99 ± .08

Comparison between regression loss and cycle loss

We studied the effect of the regression loss $\mathcal{L}_r$ and the cycle loss $\mathcal{L}_c$ in the DesignGAN architecture, both of which can be used to maintain the representations of the input design target images. Figure 5.8 demonstrates the generated results of employing varied relative strengths of $\mathcal{L}_r$ and $\mathcal{L}_c$. More specifically, when $\mathcal{L}_c$ is used alone, with its weight set to a relatively small value (Figure 5.8 (a)), the generated images fail to contain the important features of the input design target image (e.g. the area pointed by a red arrow in the target image in Figure 5.8). Increasing either the strength of $\mathcal{L}_r$ or that of $\mathcal{L}_c$ can

enable the generated images to preserve more features of the design target image. In the case of applying $\mathcal{L}_c$ alone, we observe that it is only when setting its weight to a relatively larger value will the generated images resemble the input image (Figure 5.8 (e)). However, in such case, the generated results tend to lose the details of biological features. By contrast, applying $\mathcal{L}_r$ (along with $\mathcal{L}_c$) gives rise to better generated biologically-inspired images (Figure 5.8 (b)), which demonstrates the effectiveness of $\mathcal{L}_r$ as expected. Nevertheless, the weight of $\mathcal{L}_r$ needs to be carefully tuned in order to prevent the generated images from being exactly identical to the input image (i.e. not able to integrate the representations of the biological source images, as can be seen in Figure 5.8 (c)).

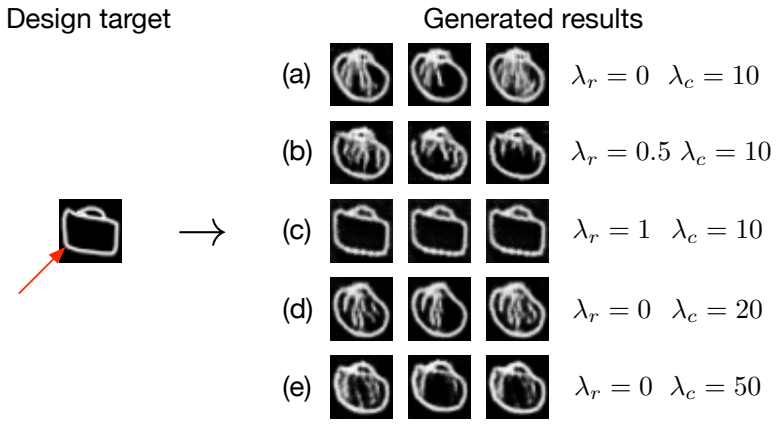


Figure 5.8.: Comparison results between employing different relative strengths of the regression loss $\mathcal{L}_r$ and the cycle loss $\mathcal{L}_c$ in the DesignGAN architecture. We use suitcase + onion as an example.

Latent variable interpolation results

We further investigated whether our proposed DesignGAN can learn relevant and useful representations in the latent space Z of the "intermediate" bionic design domain I. More specifically, for the generative

function $G_{\mathrm{DB}} : \mathrm{D} \times \mathrm{Z} \rightarrow \mathrm{I}$, we fixed the input design target image d (from D), while linearly interpolated two input latent variables z (from Z). Figure 5.9 presents the generated results of latent variable interpolations. It can be observed that all the generated biologically-inspired design images not only well preserve the features of the given design target image, but also present smooth and continuous transitions of the biological features to be integrated (e.g. varying textures of onion (Figure 5.9 (a)) and shape of flower leaves (Figure 5.9 (b))). Such gradual semantic changes clearly indicate that the DesignGAN architecture is capable of learning the required useful representations for the bionic design problem, rather than a simple memorisation of the training samples [105].

Machine design vs. human design

We finally performed a preliminary experiment that compared human design with machine design for the shape-oriented bionic design problem. Specifically, we recruited a subject with industrial design background, who was asked to draw biologically-inspired images according to a design target image and a specified biological source. We then compared them with the ones generated by our proposed DesignGAN. As presented in Figure 5.10, DesignGAN is able to produce bionic designs that are similar to human designs. However, human designs not only contain finer details, but also exhibit much larger diversity. This is for two reasons. First, humans can employ their much wider prior knowledge for the bionic design task, while machine can only learn limited knowledge from the given data. Second, a human designer is able to evaluate the design work reasonably, which can help improve their designs. By contrast, without a good evaluating function, machine can only perform "random" generation. Integrating wider prior knowledge into the deep networks, as well as designing reasonable evaluating functions, are two important directions of future work for generating more creative and plausible results of bionic design.

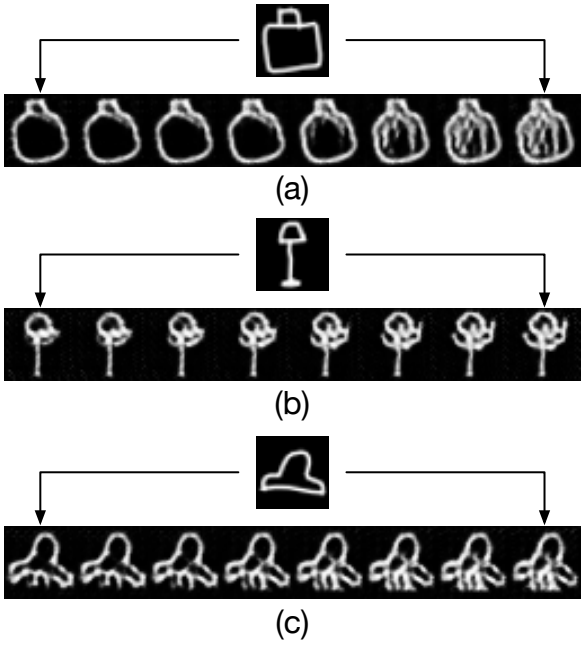


Figure 5.9.: Generated results of latent variable interpolations by the DesignGAN architecture. DesignGAN takes as input a fixed input design target image, and one of the linear interpolation results between two given latent variables. (a) Suitcase + onion. (b) Floor lamp + flower. (c) Hat + octopus.

5.5. Conclusions

In this chapter, we proposed DesignGAN, an innovative conditional GAN based architecture for the shape-oriented bionic design problem, which is a challenging image-to-image generation task that demonstrates combinatorial creativity. We presented a systemic design path of this architecture, including CycleGAN, CycleGAN+N and CycleGAN+2E, with their limitations discussed accordingly. We further proposed a novel objective function consisting of adversarial loss, cycle loss, regression loss and latent loss, which enabled DesignGAN to generate diverse, creative and plausible bionic designs.

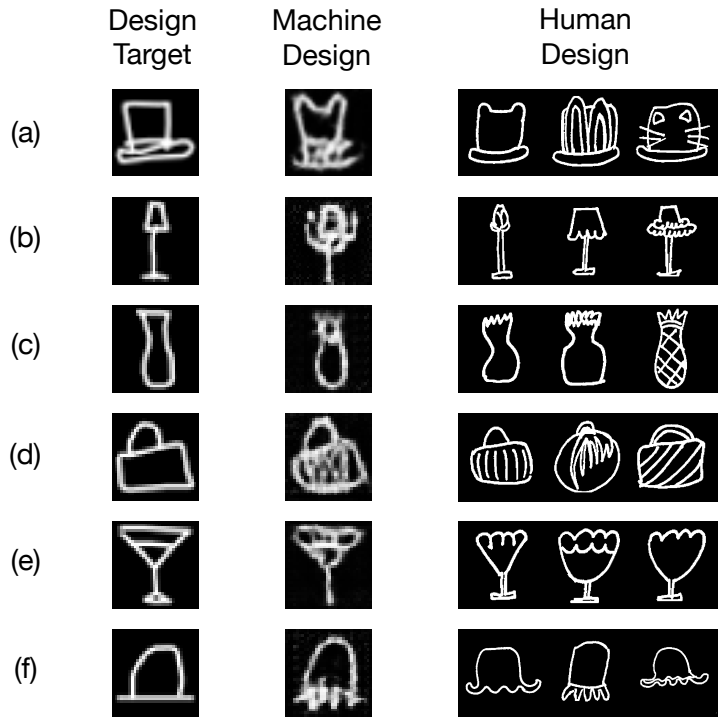


Figure 5.10.: Comparison results between machine design and human design for the shape-oriented bionic design problem. Machine design examples are created by our proposed DesignGAN architecture. Human design is performed by a designer. (a) Hat + rabbit. (b) Floor Lamp + flower. (c) Vase + pineapple. (d) Suitcase + onion. (e) Wine glass + flower. (f) Hat + octopus.

We conducted qualitative and quantitative experiments on the methods of our design path, which demonstrated that our proposed DesignGAN outperformed other methods when all the judging criteria of bionic design were taken into account (including preserving design target features, integrating biological features, diversity, creativity and plausibility). We also showed the effectiveness of the proposed regression loss for better maintaining the representations of the given design target images. We then demonstrated that DesignGAN was able to

learn useful and relevant representations required for the bionic design problem. We finally discussed the limitations of DesignGAN by comparing its generated biologically-inspired designs with those from a human designer.

In summary, our proposed DesignGAN is capable of generating creative and plausible shape-oriented bionic designs. It is worth noticing that the shape-oriented bionic design problem we addressed can be regarded as an essential prerequisite for tackling more comprehensive and complicated bionic design problems that may require the manipulation of colours, textures, etc. The successful generation of the biologically-inspired designs, which have never existed before and are substantially distinct from the existing data, clearly demonstrates that deep conditional adversarial networks can learn to achieve high-degree combinatorial creativity of image-to-image generation.

6. Conclusions

6.1. Summary of the thesis

Enabling machines to achieve generative creativity is a challenging task and a long-term goal in the field of computer vision and artificial intelligence. In this thesis, we proposed different novel conditional adversarial networks for challenging image-to-image generation tasks, each of which respectively exhibited enrichment creativity, transformative creativity and combinatorial creativity. We demonstrated that deep conditional adversarial networks are effective approaches to achieving high-level image-to-image generative creativity. Detailed contributions of this thesis are summarised as follows.

- In Chapter 3, we proposed EnrichGAN, an innovative conditional GAN based architecture for CS-MRI reconstruction, which was formulated as a challenging image-to-image generation task that demonstrated enrichment creativity. EnrichGAN was applied to enrich local structures and details of input low-quality MRI images into finer and clearer ones. We also devised a novel objective function composed of four designated loss terms and an enrichment learning process. We performed comprehensive qualitative and quantitative evaluation, and showed that EnrichGAN outperformed conventional and state-of-the-art methods of CS-MRI reconstruction, with a much faster processing time that allows real-time applications. We also showed that EnrichGAN was able to perform zero-shot inference and obtain faithful results. We further demonstrated the robustness of EnrichGAN: it only

required minimum tuning of hyperparameters, which was of great importance for medical image processing. We finally showed the effectiveness of each loss terms and the enrichment learning process via ablation studies. To the best of our knowledge, this was the first work that employed GAN based framework for CS-MRI reconstruction and achieved superior results.

- In Chapter 4, we proposed SimGAN, a novel conditional adversarial architecture for semantic image manipulation, as a challenging image-to-image generation application that demonstrated transformative creativity. It required learning a good mapping between visual features of images and semantic features of texts. We devised an innovative objective function for this complex task. We further proposed to employ a deeper pretrained image encoder (for learning better visual features) and a pretrained text encoder (for learning visual-semantic text features). By performing both qualitative and quantitative experiments on different datasets, we showed that SimGAN and its variant outperformed the baseline method in all aspects of image quality. We also demonstrated that SimGAN was capable of learning useful mapping between image and text features, and producing diverse results given one input image and text description. We finally showed that SimGAN can be extended to generate images of higher resolution (256×256) with much finer details. The generated images showed a clear distinction from any existing samples in training set, indicating that SimGAN learned to achieve high-level transformative creativity.

- In Chapter 5, we proposed DesignGAN, an innovative conditional adversarial architecture for shape-oriented bionic design, which was a challenging image-to-image task that exhibited combinatorial creativity. It required learning and combining representations from different domains of design target and biological source, all in an unsupervised fashion. We presented a sys-

temic design path of variants of the architecture. We also proposed a novel objective function consisting of four loss terms for this challenging task. We performed qualitative and quantitative experiments, and demonstrated that our proposed DesignGAN outperformed other comparison methods, with all evaluating criteria taken into account. We then showed the effectiveness of loss terms. We further demonstrated that DesignGAN successfully learned the useful and relevant representations for the bionic design problem. We finally compared the generated results with those from a designer. DesignGAN was able to create diverse biologically-inspired designs that were significantly different from existing training samples, which indicated that it learned to achieve high-degree image-to-image combinatorial creativity.

6.2. Future work

There are several possible directions of future work to be investigated, which are discussed as follows.

- We demonstrated that SimGAN achieved superior results of semantic image manipulation on bird and flower dataset. One question is whether it can be applied to datasets with more general images and texts. We plan to use the MS COCO dataset [77] as a more challenging semantic image manipulation problem. This dataset contains much more varied objects but fewer captions per image. Such increased complexity requires the further development or adjustment of the current SimGAN architecture, in order to learn useful mappings between visual and text features for generating visually appealing results.

- DesignGAN was used to perform the task of shape-oriented bionic design for combinatorial creativity. One future work is to apply DesignGAN to color images or images of higher resolution for a more comprehensive bionic design problem. However, currently

there is no appropriate dataset available. We thus plan to create one such dataset, and further extend the DesignGAN framework to achieve more complicated bionic designs.

- One limitation of our proposed approaches (and of most deep learning methods) is that a large amount of data is required for training. This is particularly an issue for medical image processing, in which medical images are expensive and scarce to obtain. In order to enhance the practical use of EnrichGAN, we intend to perform additional experiments by using fewer training samples and further develop techniques (other than data augmentation) to improve data efficiency.

- The objective function used in our proposed EnrichGAN and DesignGAN consists of multiple loss terms. How to quantitatively balance different loss terms during training process (especially for deep generative models) remains an open question. One direction of future work is therefore to comprehensively investigate how each loss term affects the usefulness and novelty of final generated images. This will contribute to enhancing the interpretability of the proposed deep generative networks.

- DesignGAN generates novel and plausible biologically-inspired images in a random manner. A natural question is whether we can improve this automated design process to be goal-oriented as human designers would do. For example, the design results can be iteratively evolved according to some criteria, where techniques such as reinforcement learning [131] can be applied. However, this requires good quantitative metrics on novelty and usefulness of the generated results, which remains a challenging open problem. Two possible directions of future work are 1) to develop quantitative metrics that well correspond to human evaluation, and 2) to incorporate human-in-the-loop in the generation procedure for achieving superior results.

Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: a system for large-scale machine learning. In *OSDI*, 2016.

[2] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville. Augmented CycleGAN: learning many-to-many mappings from unpaired data. In *ICML*, 2018.

[3] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. In *ICIP*, 2017.

[4] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

[5] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[7] P. Bao and L. Zhang. Noise reduction for magnetic resonance images via adaptive multiscale products thresholding. *IEEE Trans. Med. Imag.*, 22(9):1089–1099, 2003.

[8] R. Baraniuk. Compressive sensing. *IEEE Signal Process. Mag.*, 24(4):118–121, 2007.

[9] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[10] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.*, 5(2):157–166, 1994.

[11] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. In *ICLR*, 2017.

[12] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised diverse colorization via generative adversarial networks. In *ECML/PKDD*, 2017.

[13] R. Caruana, S. Lawrence, and L. Giles. Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In *NIPS*, 2000.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[15] S. Chintala, E. Denton, M. Arjovsky, and M. Mathieu. How to train a GAN? Tips and tricks to make GANs work. `https://github.com/soumith/ganhacks`, 2016.

[16] K. Cho, B. van Merrienboer, . Glehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[17] Y. Choi, M.-J. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[18] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: an overview. *IEEE Signal Process. Mag.*, 35(1):53–65, 2018.

[19] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[20] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[21] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2017.

[22] H. Dong, P. Neekhara, C. Wu, and Y. Guo. Unsupervised image-to-image translation with generative adversarial networks. *arXiv preprint arXiv:1701.02676*, 2017.

[23] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, S. Yu, and Y. Guo. TensorLayer: a versatile library for efficient deep learning development. In *ACM-MM*, 2017.

[24] H. Dong, S. Yu, C. Wu, and Y. Guo. Semantic image synthesis via adversarial learning. In *ICCV*, 2017.

[25] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016.

[26] J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[27] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. In *ICLR*, 2017.

[28] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. In *ICLR*, 2017.

[29] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[30] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.

[31] M. J. Ehrhardt and M. M. Betcke. Multicontrast MRI reconstruction with structure-guided total variation. *SIAM J. Imaging Sci.*, 9(3):1084–1106, 2016.

[32] E. M. Eksioglu. Decoupled algorithm for MRI reconstruction using nonlocal block matching model: BM3D-MRI. *J. Math. Imaging Vis.*, 56(3):430–440, 2016.

[33] A. M. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *ICCC*, 2017.

[34] C. Flath, S. Friesike, M. Wirth, and F. Thiesse. Copy, transform, combine: exploring the remix as a form of innovation. *J. Inf. Technol.*, 32(4):306–325, 2017.

[35] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong. Recent advances in zero-shot recognition: toward data-efficient understanding of visual content. *IEEE Signal Process. Mag.*, 35(1):112–125, 2018.

[36] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[37] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with LSTM. *Neural Comput.*, 12(10):2451–2471, 2000.

[38] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with LSTM recurrent networks. *J. Mach. Learn. Res.*, 3:115–143, 2003.

[39] I. Goodfellow. NIPS 2016 tutorial: generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[40] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016.

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[42] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber. LSTM: a search space odyssey. *IEEE Trans. Neural Netw. Learning Syst.*, 28(10):2222–2232, 2017.

[43] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: a recurrent neural network for image generation. In *ICML*, 2015.

[44] B. K. Gunturk and X. Li. *Image restoration: fundamentals and advances*. CRC Press, 2012.

[45] D. Ha and D. Eck. A neural representation of sketch drawings. In *ICLR*, 2018.

[46] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated MRI data. *arXiv preprint arXiv:1704.00447*, 2017.

[47] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In *ICCV*, 2015.

[48] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[49] M. Helms, S. S. Vattam, and A. K. Goel. Biologically inspired design: process and products. *Design Stud.*, 30(5):606–622, 2009.

[50] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *SIGGRAPH*, 2001.

[51] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[52] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

[53] K. G. Hollingsworth. Reducing acquisition time in clinical MRI by data undersampling and compressed sensing reconstruction. *Phys. Med. Biol.*, 60(21):297–322, 2015.

[54] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[55] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In *CVPR*, 2017.

[56] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[57] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, 2017.

[58] S. Ioffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[59] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[60] O. N. Jaspan, R. Fleysher, and M. L. Lipton. Compressed sensing MRI: a review of the clinical literature. *Br. J. Radiol.*, 88(1056):1–12, 2015.

[61] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: a review. *arXiv preprint arXiv:1705.04058*, 2017.

[62] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[63] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[64] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[65] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.

[66] D. Kingma and J. Ba. Adam: a method for stochastic optimization. In *ICLR*, 2014.

[67] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[68] R. Kiros, R. Salakhutdinov, and R. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *TACL*, 2015.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[70] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016.

[71] Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[72] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1989.

[73] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[74] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[75] C. Li and M. Wand. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In *ECCV*, 2016.

[76] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[77] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[78] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[79] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[80] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[81] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58(6):1182–1195, 2007.

[82] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing MRI. *IEEE Signal Process. Mag.*, 25(2):72–82, 2008.

[83] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[84] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. In *ICLR*, 2016.

[85] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[86] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[87] R. E. Mayer. Fifty years of creativity research. In R. J. Sternberg, editor, *Handbook of creativity*, chapter 10. Cambridge University Press, 1999.

[88] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

[89] T. Mikolov, M. Karafiát, L. Burget, J. ernock, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

[90] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[91] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[92] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICCVGIP*, 2008.

[93] H. Nyquist. Certain topics in telegraph transmission theory. *Trans. Am. Inst. Elect. Eng.*, 47(2):617–644, 1928.

[94] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. `https://distill.pub/2016/deconv-checkerboard/`, 2016.

[95] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017.

[96] C. Olah. Understanding LSTM networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015.

[97] A. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

[98] A. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *NIPS*, 2016.

[99] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: feature learning by inpainting. In *CVPR*, 2016.

[100] K. Pawar, G. Egan, and J. Zhang. Multichannel compressive sensing MRI using noiselet encoding. *PLOS ONE*, 10(5):e0126386, 2015.

[101] G. Perarnau, J. v. d. Weijer, B. Raducanu, and J. M. Álvarez. Invertible conditional GANs for image editing. In *NIPS Workshop on Adversarial Training*, 2016.

[102] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.*, 4(5):1–17, 1964.

[103] X. Qu, D. Guo, B. Ning, Y. Hou, Y. Lin, S. Cai, and Z. Chen. Undersampled MRI reconstruction with patch-based directional wavelets. *Magn. Reson. Imag.*, 30(7):964–977, 2012.

[104] X. Qu, Y. Hou, F. Lam, D. Guo, J. Zhong, and Z. Chen. Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator. *Med. Image Anal.*, 18(6):843–856, 2014.

[105] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative Adversarial Networks. In *ICLR*, 2016.

[106] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. CheXNet: radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[107] S. Ravishankar and Y. Bresler. MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Trans. Med. Imag.*, 30(5):1028–1041, 2011.

[108] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[109] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*, 2016.

[110] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[111] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[112] O. Ronneberger, P. Fischer, and T. Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[113] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[114] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015.

[115] A. T. Sack and T. Schuhmann. Hemispheric differences within the fronto-parietal network dynamics underlying spatial imagery. *Front Psychol.*, 3:214, 2012.

[116] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NIPS*, 2016.

[117] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: controlling deep image synthesis with sketch and color. In *CVPR*, 2017.

[118] O. Sbai, M. Elhoseiny, A. Bordes, Y. LeCun, and C. Couprie. DesIGN: design inspiration from generative networks. *arXiv preprint arXiv:1804.00921*, 2018.

[119] A. Schlegel, P. J. Kohler, S. V. Fogelson, P. Alexander, D. Konuthula, and P. U. Tse. Network structure and dynamics of the mental workspace. *Proc. Natl. Acad. Sci. U. S. A.*, 110(40):16277–16282, 2013.

[120] J. Schlemper, J. Caballero, J. V. Hajnal, A. Price, and D. Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. In *IPMI*, 2017.

[121] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *CVPR*, 2017.

[122] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[123] L. Shu, K. Ueda, I. Chiu, and H. Cheong. Biologically inspired design. *CIRP Ann.*, 60(2):673–693, 2011.

[124] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[125] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, 2003.

[126] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[127] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: the all convolutional net. In *ICLR Workshop*, 2014.

[128] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[129] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[130] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[131] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.

[132] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[133] C. K. Snderby, J. Caballero, L. Theis, W. Shi, and F. Huszr. Amortised MAP inference for image super-resolution. In *ICLR*, 2017.

[134] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017.

[135] C. Thornton. Generative creativity course handbook. `http://users.sussex.ac.uk/~christ/crs/gc/`, 2010.

[136] T. Tieleman and G. Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning, 2012.

[137] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *ICML*, 2016.

[138] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: the missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[139] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[140] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang. Accelerating magnetic resonance imaging via deep learning. In *ISBI*, 2016.

[141] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016.

[142] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[143] Y. Wu and K. He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018.

[144] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: neural image caption generation with visual attention. In *ICML*, 2015.

[145] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[146] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang. Learning to super-resolve blurry face and text Images. In *ICCV*, 2017.

[147] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017.

[148] G. Yang, S. Yu, H. Dong, G. G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. R. Arridge, J. Keegan, Y. Guo, and D. N. Firmin. DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Trans. Med. Imaging*, 37(6):1310–1321, 2018.

[149] J. Yang, Y. Zhang, and W. Yin. A fast alternating direction method for TVL1-L2 signal reconstruction from partial fourier data. *IEEE J. Sel. Topics Signal Process.*, 4(2):288–297, 2010.

[150] Y. Yang, J. Sun, H. Li, and Z. Xu. Deep ADMM-Net for compressive sensing MRI. In *NIPS*, 2016.

[151] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, 2017.

[152] Z. Yi, H. Zhang, P. Tan, and M. Gong. DualGAN: unsupervised dual learning for image-to-image translation. In *ICCV*, 2017.

[153] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. In *ECCV*, 2016.

[154] S. Yu, H. Dong, P. Wang, C. Wu, and Y. Guo. Generative creativity: adversarial learning for bionic design. *arXiv preprint arXiv:1805.07615*, 2018.

[155] S. Yu, H. Dong, P. Wang, C. Wu, and Y. Guo. Spatial feature combination for generative creativity: a case study of bionic design. In *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2018.

[156] S. Yu, H. Dong, G. Yang, G. G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. R. Arridge, J. Keegan, D. N. Firmin, and Y. Guo. Deep de-aliasing for fast compressive sensing MRI. *arXiv preprint arXiv:1705.07137*, 2017.

[157] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[158] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. StackGAN++: realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017.

[159] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[160] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016.

[161] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

[162] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[163] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017.

A. Supplementary for Chapter 2

A.1. Detailed network architecture of the VGG network

Figure A.1 demonstrates the detailed architecture of the VGG (VGG16) network [126] employed as a (pretrained) feature extractor in Chapter 3 and Chapter 4.

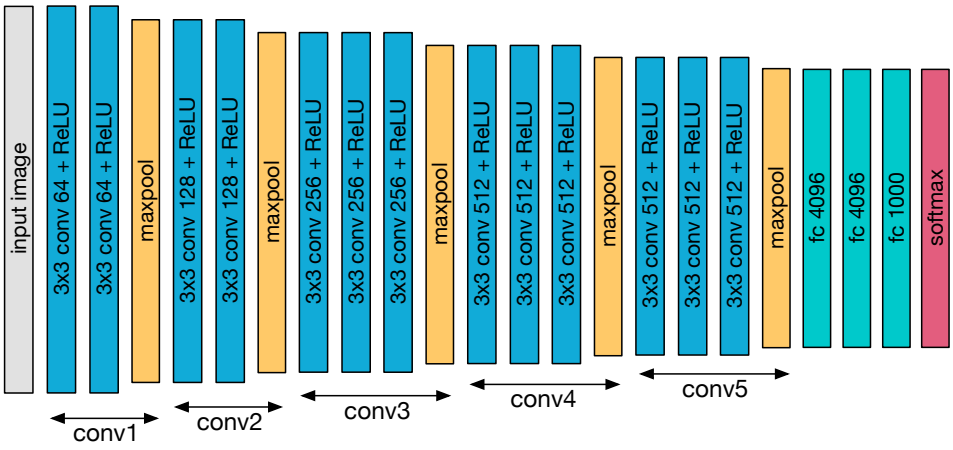


Figure A.1.: Detailed architecture of the VGG (VGG16) network.

B. Supplementary for Chapter 3

B.1. Detailed network architecture of EnrichGAN

The detailed network architecture of our proposed EnrichGAN for CS-MRI reconstruction can be seen in Figure B.1.

B.2. Random undersampling masks

Figure B.2 presents the random undersampling masks employed in the experiments.

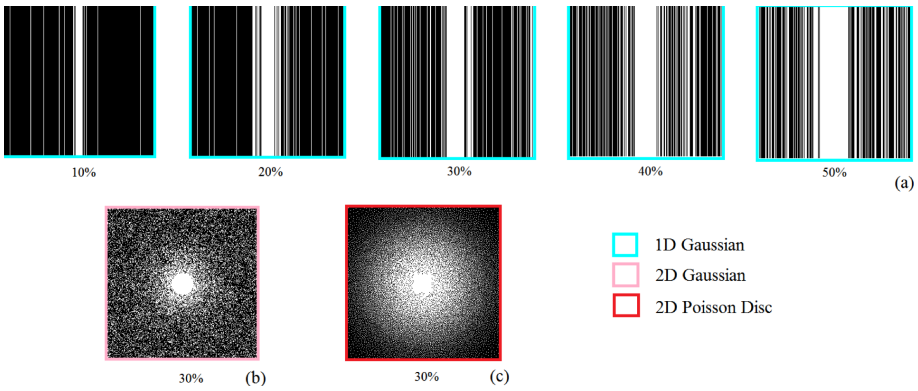


Figure B.2.: The random undersampling masks employed in the experiments. (a) 1D Gaussian. (b) 2D Gaussian. (c) 2D Poisson disc.

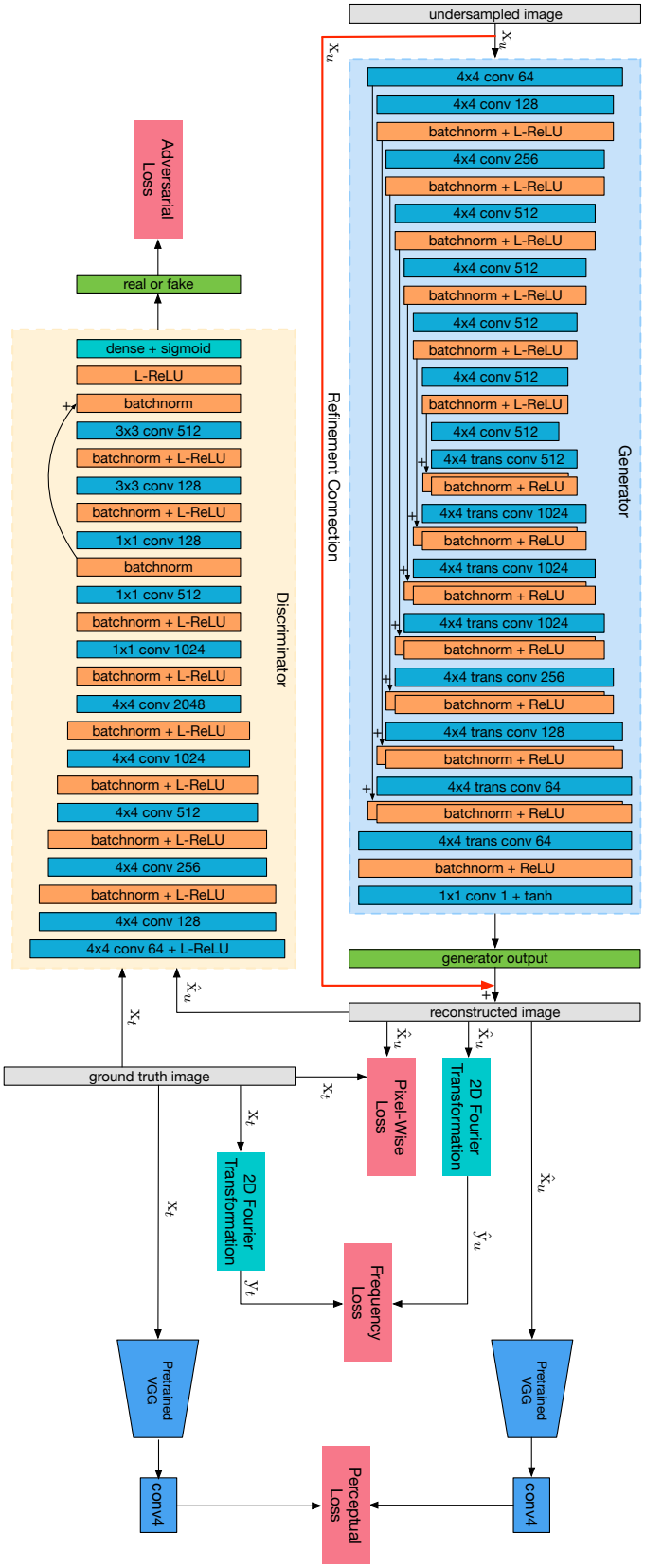


Figure B.1.: Detailed network architecture of our proposed EnrichGAN model for CS-MRI reconstruction.

C. Supplementary for Chapter 4

C.1. Detailed network architecture of SimGAN

The detailed network architecture of our proposed SimGAN for semantic image manipulation is presented in Figure C.1.

C.2. Additional results on Caltech-200 bird dataset

Figure C.2 demonstrates additional zero-shot results of our proposed SimGAN on images from Caltech-200 bird dataset.

C.3. Additional results on Oxford-102 flower dataset

Figure C.3 presents additional zero-shot results of our proposed SimGAN on images from Oxford-102 flower dataset.

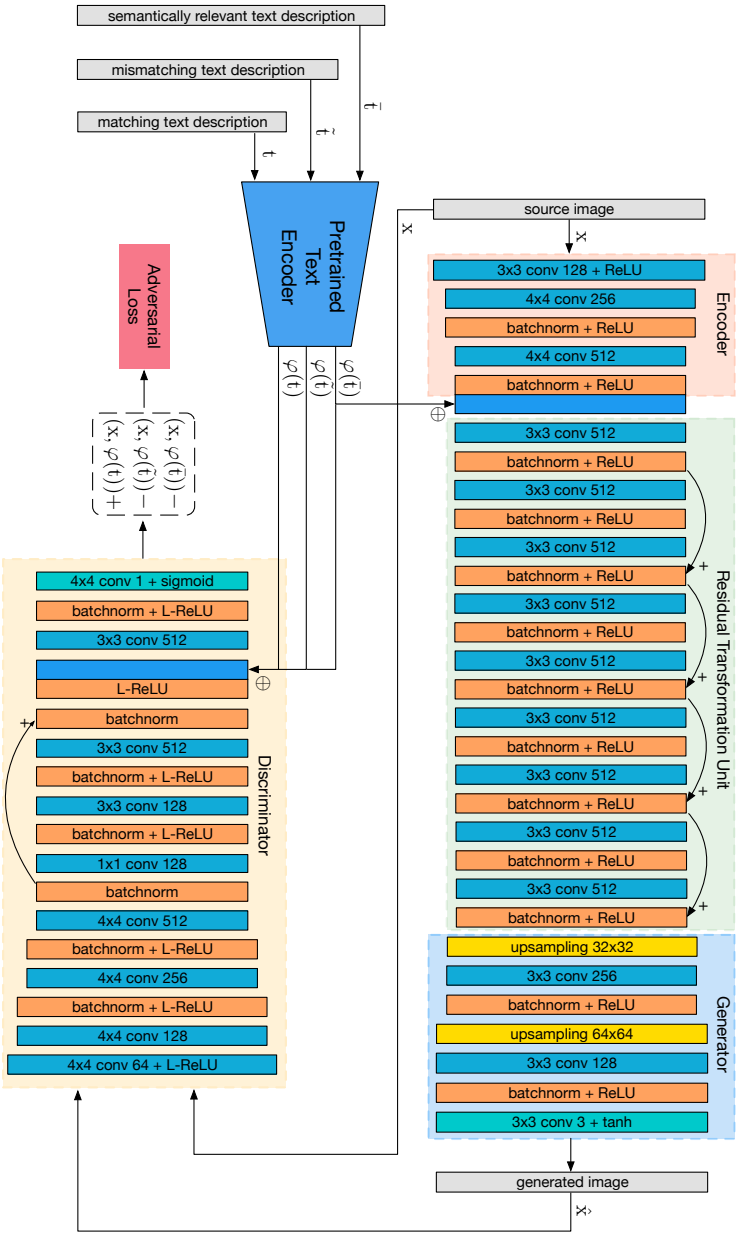


Figure C.1.: Detailed network architecture of our proposed SimGAN model for semantic image manipulation.

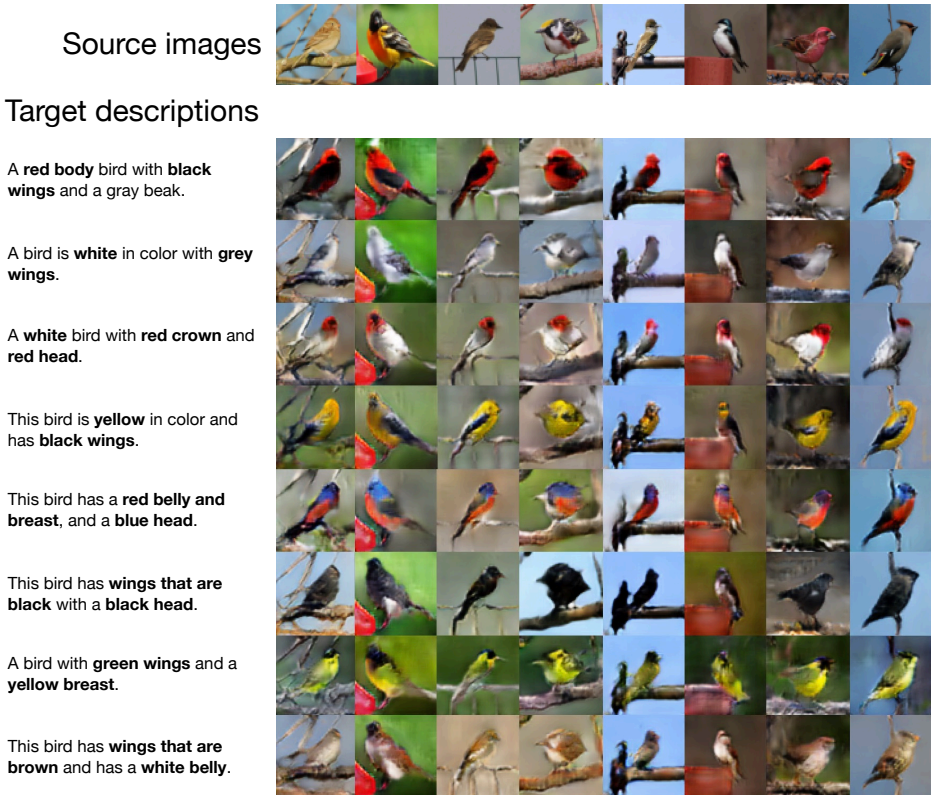


Figure C.2.: Additional zero-shot results of our proposed SimGAN on images from the test set of the Caltech-200 bird dataset.

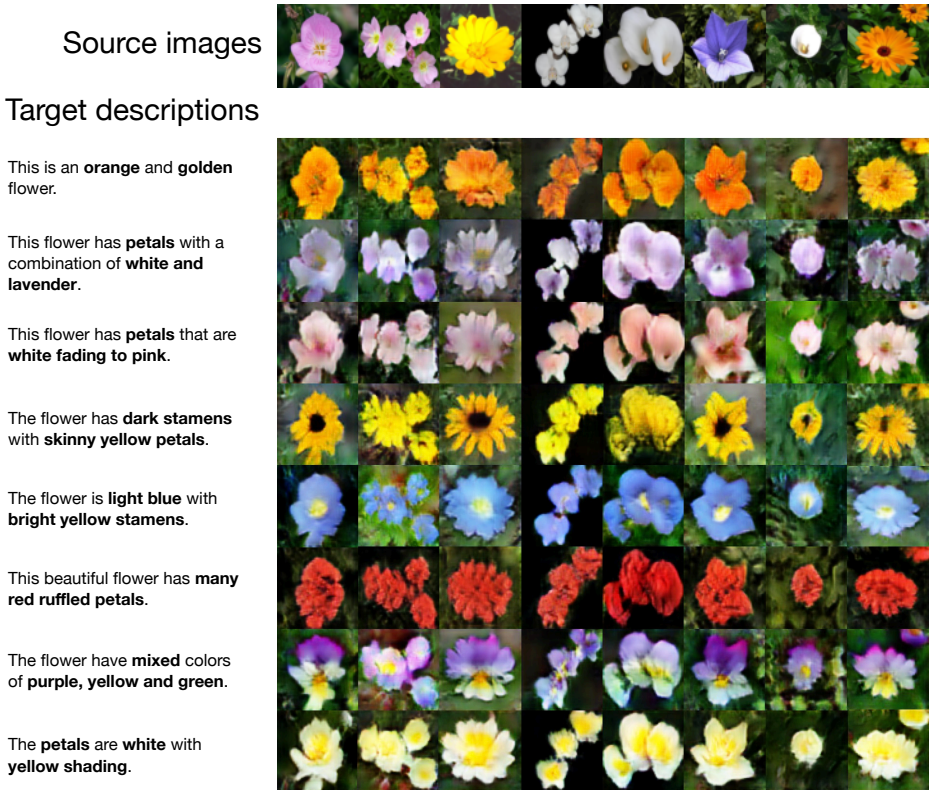


Figure C.3.: Additional zero-shot results of our proposed SimGAN on images from the test set of the Oxford-102 flower dataset.

D. Supplementary for Chapter 5

D.1. Detailed network architecture of DesignGAN

Figure D.1 shows the architecture of DesignGAN for bionic design.

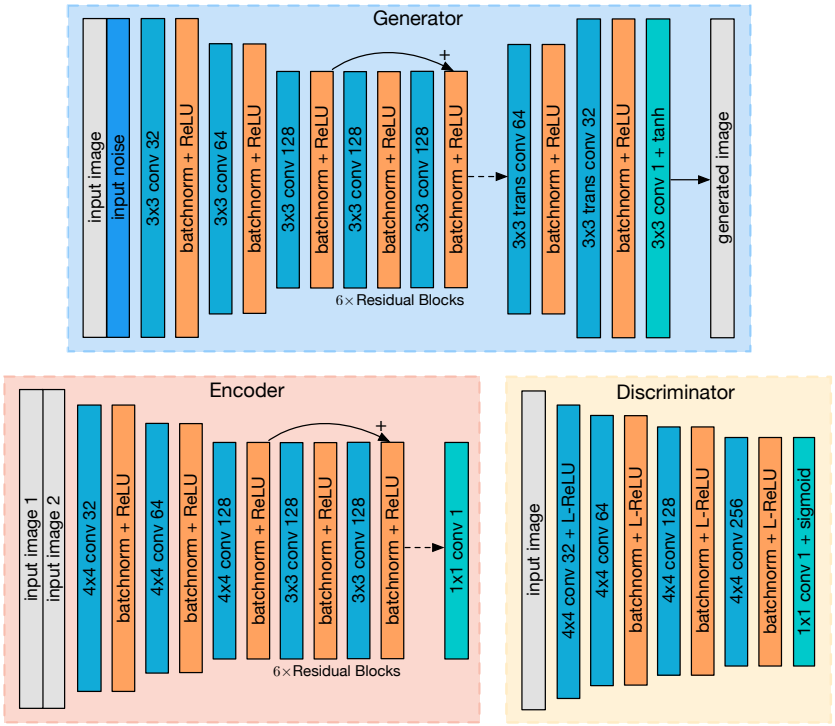


Figure D.1.: Detailed network architecture of our proposed DesignGAN for bionic design.