

Generative Modelling and Adversarial Learning



Chaoyue Wang

Faculty of Engineering and Information Technology

University of Technology Sydney

A thesis submitted for the degree of

Doctor of Philosophy

2018

I dedicate this thesis to my loving families

Hongzhen Wang and *Wenru Wang*

Yanbin Wang and *Xiangfa Wu*

and *Chenlu Li*

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Chaoyue Wang

Acknowledgements

I would like to thank everyone who has helped me to finish my doctoral studies.

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dacheng Tao for his continuous support of my Ph.D study. Thanks for his consistent patience and motivation, for his encouraging attitude and expert knowledge for my research. His strict academic attitude and diligent work style have played a role model for me and will continue to benefit me through my life. It is no exaggeration to say without his help steering my research direction, I would not have finished this thesis so smoothly and on time. Also I want to thanks for his advice and help on my career development.

Besides my principal supervisor, I would like to thank Dr. Chang Xu and Prof. Chaohui Wang. I want to thanks for them never bored discussion with my seemingly endless simple questions from research motivation, model development, algorithm implementation, and paper drafting. With these specific and detailed technical discussion, I have gained practical skills to effectively and efficiently develop and implement my research problems.

I also wish to give special thanks to Ying Wu, Dongang Wang, Guoliang Kang, Dayong Tian, Yuxuan Du, Xinyuan Chen, Dalu Guo, Huan Fu, Xiyu Yu, Jue Wang, Jiang Bian, Yali Du, Shan You, Jianfeng Dong, Liu Liu, Baosheng Yu, Zhe Chen, Jiayan Qiu and Erkun Yang. Without their helps, my life in Sydney would not be so easy. I would like to give my gratitude to Changxing Ding, Maoying Qiao, Shaoli Huang and Tongliang Liu for their helps on my research.

Last but not the least, my gratitude extends to my family who have been patiently encouraging and waiting for the finish of this thesis.

Abstract

A main goal of statistics and machine learning is to represent and manipulate high-dimensional probability distributions of real-world data, such as natural images. Generative adversarial networks (GAN), which are based on the adversarial learning paradigm, are one of the main types of methods for deriving generative models from complicated real-world data. GAN and its variants use a generator to synthesise semantic data from standard signal distributions and train a discriminator to distinguish real samples in the training dataset from fake samples synthesised by the generator. As a confronter, the generator aims to deceive the discriminator by producing ever more realistic samples. Through a two-player adversarial game played by the generator and discriminator, the generated distribution can approximate the real-world distribution and generate samples from it.

This thesis aims to both improve the quality of generative modelling and manipulate generated samples by specifying multiple scene properties. A novel framework for training GAN is proposed to stabilise the training process and produce more realistic samples. Unlike existing GANs, which alternately train a generator and a discriminator using a pre-defined adversarial objective function, different adversarial training objectives are utilised as mutation operations and train a population of generators to adapt to the environment (i.e. the discriminator). The samples generated by different iterations of generators are evaluated and only well-performing generators are preserved and used for further training. In this way, the proposed framework overcomes the limitations of an individual adversarial training objective and always preserves the best offspring, contributing to the progress and success of GANs.

Based on the GANs framework, this thesis devised a novel model, called a perceptual adversarial network (PAN). The proposed PAN consists of two feed-forward convolutional neural networks: a transformation network and a discriminative network. Besides generative adversarial loss, which is widely used in GANs, this thesis proposes to employ perceptual adversarial loss, which undergoes adversarial training between the transformation network and hidden layers of the discriminative network. The hidden layers and output of the discriminative network are upgraded to constantly and automatically discover discrepancies between a transformed image and the corresponding ground truth, and the image transformation network is trained to minimise the discrepancy identified by the discriminative network.

Furthermore, to extend the generative models to perform more challenging re-rendering tasks, this thesis explores disentangled representations encoded in real-world samples and proposes a principled tag disentangled generative adversarial network for re-rendering new samples of the object of interest from a single image by specifying multiple scene properties. Specifically, from an input sample, a disentangling network extracts disentangled and interpretable representations, which are then used to generate new samples using the generative network. In order to improve the quality of the disentangled representations, a tag mapping net determines the consistency between the image and its tags.

Finally, experiments with different challenging datasets and image synthesis tasks demonstrate the good performance of the proposed frameworks regarding the problem of interest.

Contents

Contents	x
List of Figures	xiii
1 Introduction	1
1.1 Background	1
1.2 Related Works	7
1.3 Generative adversarial networks (GAN)	8
1.4 Variants of GANs	9
1.5 Summary of Contributions.	9
1.6 Thesis Structure.	12
2 Stabilizing adversarial training process via selecting optimization	14
2.1 Introduction	15
2.2 Related Works	16
2.2.1 Generative Adversarial Networks	17
2.2.2 Evolutionary Algorithms	18
2.3 Method	18
2.3.1 Generative Adversarial Networks	18
2.3.2 Evolutionary Algorithm	20
2.3.3 Mutations	21
2.3.4 Evaluation	24
2.3.5 E-GAN	27
2.4 Experiments	28

CONTENTS

2.4.1	Implementation Details	28
2.4.2	CIFAR-10 and Inception Score	32
2.4.3	LSUN and Architecture Robustness	33
2.4.4	Synthetic Datasets and Model Collapse	35
2.4.5	CelebA and Space Continuity	37
2.5	Summary	37
3	Learning perceptual information through adversarial paradigm	39
3.1	Introduction	40
3.2	Related works	44
3.2.1	Image-to-image transformation with feed-forward CNNs	44
3.2.2	GANs-based works	44
3.2.3	Perceptual loss	46
3.3	Method	46
3.3.1	Generative adversarial loss	46
3.3.2	Perceptual adversarial loss	48
3.3.3	The perceptual adversarial networks	51
3.3.4	Network architectures	52
3.4	Experiments	53
3.4.1	Experimental setting up	53
3.4.2	Evaluation metrics	57
3.4.3	Analysis of the loss functions	57
3.4.4	Comparing with existing works	61
3.5	Summary	68
4	Interpretable and disentangled representations in adversarial learning	69
4.1	Introduction	70
4.2	Related Works	72
4.3	Method	73
4.3.1	Main Framework	73
4.3.2	Training Process	76
4.3.3	Image Re-rendering	78

CONTENTS

4.4	Experiments	79
4.4.1	Implementation Details	79
4.4.2	Performance Criteria	80
4.4.3	Experimental Results	82
4.5	Summary	89
5	Conclusions	90
	Appendix A	92
5.1	Proof of the Optimal Discriminator	92
5.2	Network Architectures	93
5.3	Synthetic Datasets	94
5.4	Cifar-10	96
5.5	LSUN Bedrooms	97
5.6	CelebA Faces	103
	References	105

List of Figures

1.1	The training process of generative adversarial networks (GANs). An adversarial game is performed between a generator and a discriminator.	3
2.1	(a) The original GAN framework. A generator G and a discriminator D play a two-player adversarial game. The updating gradients of the generator G are received from the adaptive objective, which depends on discriminator D . (b) The proposed E-GAN framework. A population of generators $\{G_\theta\}$ evolves in a dynamic environment, the discriminator D . Each evolutionary step consists of three sub-stages: variation, evaluation, and selection. The best offspring are kept.	19
2.2	The mutation (or objective) functions that the generator G receives given the discriminator D	23
2.3	Experiments on the CIFAR-10 dataset. CIFAR-10 inception score over generator iterations (left), over wall-clock time (right).	26
2.4	Experiments on the CIFAR-10 dataset. The graph of selected mutations in the E-GAN training process	28
2.5	KDE plots of the target data and generated data from different GANs trained on mixtures of Gaussians.	29
2.6	Generated samples on 128×128 LSUN bedrooms.	30
2.7	Experiments to test architecture robustness. Different GAN architectures corresponding to different training challenges and trained with five different GAN methods.	31
2.8	Keep different numbers of candidatures.	33

LIST OF FIGURES

2.9	Generated human face images on the 128×128 CelebA dataset. .	34
2.10	Interpolating in latent space. For selected pairs of the generated images from a well-trained E-GAN model, we record their latent vectors z_1 and z_2 . Then, samples between them are generated by linear interpolation between these two vectors.	36
3.1	Image-to-image transformation tasks. Many tasks in image processing, computer graphics, and computer vision can be regarded as image-to-image transformation tasks, where a model is designed to transform an input image into the required output image. We proposed Perceptual Adversarial Networks (PAN) to solve the image-to-image transformation between paired images. For each pair of the images we demonstrated, the left one is the input image, and the right one is the transformed result of the proposed PAN. . . .	42
3.2	PAN framework. PAN consists of an image transformation network T and a discriminative network D . The image transformation network T is trained to synthesize the transformed images given the input images. It is composed of a stack of Convolution-BatchNorm-LeakyReLU encoding layers and Deconvolution-BatchNorm-ReLU decoding layers, and the skip-connections are used between mirrored layers. The discriminative network D is also a CNN that consists of Convolution-BatchNorm-LeakyReLU layers. Hidden layers of the network D are utilized to evaluate the perceptual adversarial loss, and the output of the network D is used to distinguish transformed images from real-world images.	47
3.3	Comparison of snow-streak removal using different losses functions. Given the same input image (leftmost), each column shows results trained under different losses. The loss function of ID-CGAN [178] combined the pixel-wise loss (least squares loss), cGANs loss and perceptual loss, i.e., $L_2 + \text{cGAN} + \text{perceptual}$. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.	49

LIST OF FIGURES

3.4	Transforming the semantic labels to cityscapes images use the perceptual adversarial loss. Within the perceptual adversarial loss, a different hidden layer is utilized for each experiment. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images. For higher layers, the transformed images look sharper, but less color information is preserved.	50
3.5	Comparison of transforming the semantic labels to facades images by controlling the hyper-parameter θ . Given the same input image (leftmost), each column shows results trained under different θ . For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.	53
3.6	Comparison of rain-streak removal using the ID-CGAN with the proposed PAN on real-world rainy images. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.	56
3.7	Comparison of image in-painting results using the Context-Encoder(CE) with the proposed PAN. Given the central region missed input image (leftmost), the in-painted images and the ground-truth are listed on its rightside.	58
3.8	Comparison of transforming the semantic labels to cityscapes images using the pix2pix-cGAN with the proposed PAN. Given the semantic labels (leftmost), the transformed cityscapes images and the ground-truth are listed on the rightside.	59
3.9	Comparison of transforming the object edges to corresponding images using the pix2pix-cGAN with the proposed PAN. Given the edges (leftmost), the generated images of shoes and handbags are listed on the rightside.	63
3.10	Comparison of some other tasks using the pix2pix-cGAN with the proposed PAN. In the first row, semantic labels are generated based on the real-world cityscapes images. And, the second row reports the generated maps given the aerial photos as input.	65

LIST OF FIGURES

4.1	Model architecture. <i>TD-GAN</i> is composed of four parts: a tag mapping net g , a disentangling network R , a generative network G and a discriminative network D . (a) During training, the tag mapping net g and the generative network G are trained to render images with their tags. The disentangling network R aims to extract disentangled representations, which can be decoded by the network G . The discriminative network D plays a minimax game with networks G and R based on the adversarial training strategy. (b) During test, the disentangling network R extracts disentangled representations from the input image. After replacing one or multiple disentangled representations with the specified representations generated by the tag mapping net g , the image can be re-rendered through the generative network G	74
4.2	Novel view synthesis results of two previous method and ours. For each method, the leftmost image is the input image, and the images on its right side were re-rendered under different viewpoints. . . .	81
4.3	Novel view synthesis results of <i>TD-GAN</i> trained in three settings. The images are arranged similarly to Fig. 4.2.	83
4.4	<i>MSE</i> of <i>TD-GAN</i> trained in three settings. <i>MSE</i> was calculated over all the chair images under 0° in the test set as inputs. Chair images are used to indicate target viewpoints.	84
4.5	Illumination transformation of human face (best view in color). Given an image (leftmost) of human face, we reported a set of re-rendered images on its right side.	85
4.6	Multi-factor transformation (best view in color). Given a single image as input (the up-left one), its viewpoint and expression were jointly transformed.	87
4.7	Multi-factor transformation (best view in color). Given a single image as input (the up-left one), its viewpoint and expression were jointly transformed.	88
5.1	Network architectures for generating 128×128 images.	93

LIST OF FIGURES

5.2	Different GANs learning a mixture of 8 Gaussians arranged in a circle.	94
5.3	Different GANs learning a mixture of 25 Gaussians arranged in a grid.	95
5.4	Generated samples on CIFAR-10 dataset.	96
5.5	Generated images on 128×128 LSUN bedrooms.	97
5.6	Method: DCGAN	98
5.7	Method: LSGAN	99
5.8	Method: WGAN	100
5.9	Method: WGAN-GP	101
5.10	Method: E-GAN	102
5.11	Generated human face images on 128×128 CelebA dataset. . . .	103
5.12	Interpolating in latent space. For selected pairs of the generated images from a well-trained E-GAN model, we record their latent vectors z_1 and z_2 . Then, samples between them are generated by linear interpolation between these two vectors.	104

Chapter 1

Introduction

1.1 Background

One main goal of statistics and machine learning is to represent and manipulate high-dimensional probability distributions of real-world data, such as natural images. A set of real-world images, $\{x\}$, can be regarded as samples of a high-dimensional probability distribution. After observing these samples, generative models are devised to estimate corresponding real data distributions and generate unobserved samples from them.

Determining the high-dimensional distributions of real-world images is of fundamental importance for a wide variety of computer vision and image processing tasks. For example, identification of high-dimensional distributions from training data contributes to data augmentation [167]. Modern deep learning algorithms typically require numerous training samples, especially from labeled data, with good generalisation. Generative models can generate unobserved data from the target distribution and synthesise desired samples from specific labels, significantly reducing the demand for handcrafted labels. In addition, they can be trained with missing data and can provide predictions for inputs that are missing data. One important application of such models is semi-supervised learning, in which some training examples are unlabeled. According to experimental results, many generative models can perform semi-supervised learning tasks reasonably well.

In addition, generative models enable machine learning algorithms to model real-world multimodal data distributions. In many real-world situations, a single input may have multiple correct outputs. For example, there may be many acceptable predictions for the next frame in a video [31, 86, 149]. In contrast, some traditional training methods, such as minimising mean square error, could only produce a single prediction for the next frame.

Furthermore, generative models can be used to learn the distributions of data from different domains and explore the relationship between these domains. Therefore, generative models can be utilised in a wide range of image transformation tasks and improve related experimental results, such as image super-resolution [78, 112], style transfer [70, 138] and image translation [18, 20, 69], *etc.*

To estimate the probability distributions of real-world data, various generative models have been proposed for different tasks and targets. Most are based on the principle of maximum likelihood. Specifically, the maximum likelihood process takes samples from real-world data and attempts to assign probability values to each of them. By optimising the model parameters to maximise the sum of the likelihood of all training data, these generative models can gradually understand and represent data distributions. Those that depend on the principle of maximum likelihood can be divided into different categories. Some directly define an explicit density function, $p_{\text{model}}(x|\theta)$, for each example of training. For these models, calculation of maximum likelihood is straightforward; one simply models the density function of the training data and follows the likelihood uphill. However, it is difficult to maintain computational tractability, especially when dealing with high-dimensional complex data distributions. In contrast, some generative models are devised to model real-world distributions without the need to define an explicit density function. Instead of directly interacting with the training data, these models enable implicit learning of the data distribution, such as through drawing samples. Since there is no longer a need to explicitly define the density function, implicit generative models are widely used to approximate the distribution of real-world data. However, many implicit models depend on a Markov chain transition operator that must be run several times to obtain a sample. Markov chains easily fail to sample from high-dimensional spaces and usually increase the computational cost of using the generative model.

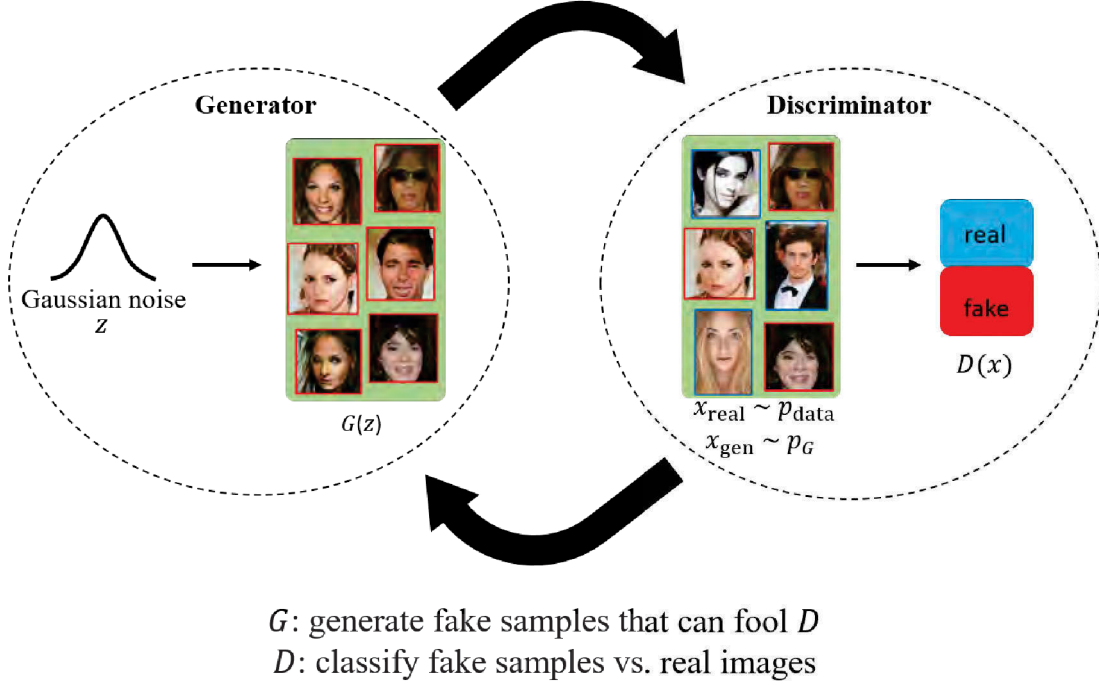


Figure 1.1: The training process of generative adversarial networks (GANs). An adversarial game is performed between a generator and a discriminator.

Recently, an adversarial learning paradigm was introduced into training generative models, and it has become increasingly important in various applications. [54] is the first work to obtain generative models through adversarial game played by two deep neural networks. One deep neural network is called a **generator**, and it is trained to synthesise semantic data from standard signal distributions. It must learn to synthesise samples from the same distribution as the training data. The other is called a **discriminator**. Using traditional supervised learning techniques, the discriminator acts as a binary classification network and aims to divide input samples into two classes (*i.e.*, *real* samples in the training dataset or *fake* samples synthesised by the generator). As a confronter, the generator is asked to fool the discriminator by producing ever more realistic samples, and the discriminator is asked to improve its ability to recognise fake samples. This training procedure continues until the generator wins the adversarial game (*i.e.* the discriminator can only randomly guess whether a particular sample is fake or

real). The whole training process is demonstrated in Fig. 1.1.

Neither the generator nor the discriminator can explicitly represent the target high-dimensional distribution of training data. However, according to theoretical analysis, if the discriminator is trained to achieve optimal performance in each adversarial iteration, updating the generator to fool the discriminator is equivalent to minimising the difference between the training data distribution and generated distribution. Here, different objective functions actually correspond to minimising different distances.

By introducing the adversarial learning paradigm into the training process, GANs can avoid many of the inherent difficulties of previous generative models. Firstly, compared to most explicit generative models, the generator function of GANs has very few restrictions. For example, Boltzmann machines ask their generator functions to admit tractable Markov chain sampling, and the generator of non-linear ICA must be invertible and the dimension of latent code must be the same as that of generated samples. Secondly, GANs can produce samples much easier than most existing generative models; a GANs generator can generate samples in parallel and never depends on Markov chains. Thirdly, GANs have no variational bounds. The usable model families within the GAN framework can be regarded as universal approximators, enabling GANs to be asymptotically consistent. Although it has been proposed that some variational auto-encoders are also asymptotically consistent, this has not been proven yet. Last but not least, experiments show that GANs are good at learning high-dimensional data distributions, such as those of images, videos and 3D objects, and usually produce visually better and more realistic samples [12, 53, 54, 73, 152, 182].

Recently, GANs have been successfully applied to various image synthesis tasks and have achieved promising performance. Unsupervised or semi-supervised image synthesis is of fundamental interest in the fields of computer vision and graphics. Through observation of real-world images, GANs can be trained to generate visually realistic samples (*i.e.*, images from the same high-dimensional distribution that have not been observed in the training set). Moreover, some conditional GANs can learn the conditional distributions of different kinds of images and further augment data in many computer vision tasks. For example, if object labels of different kinds of images (*e.g.*, ‘flower’, ‘cat’, ‘train’) are given as

input, conditional GANs can generate corresponding images that contain different objects [84, 117]. Secondly, image-to-image transformation is another type of image processing task that is widely performed by GAN models. In most image-to-image transformation tasks, learned models aim to translate input images to desired output images. Usually, the input images can be regarded as samples that lie on a specific domain and the target images correspond to another image domain. Utilising the adversarial learning paradigm, trained models can identify differences between the two domains and explore the mapping relationship between them. For example, GAN models are used to generate high-quality images from degraded (*e.g.*, simplified, corrupted or low-resolution) images, translate semantic labels to real-world images or perform de-noising, image colourisation, image segmentation and so on [28, 106]. Thirdly, generative models and the adversarial learning paradigm can be employed to re-render object images. In the fields of computer vision and graphics, re-rendering of new images of the object of interest from a single image by specifying the expected scene properties (*e.g.*, viewpoint, illumination, expression) is of particular interest. For example, image re-rendering can be used in architecture, simulators, video games, movies and visual effects, and faces can be re-rendered for continuous illumination direction, poses and various expressions. This kind of task requires the learned generative model to be able to understand the input data and flexibly manipulate the corresponding properties according to various descriptions.

Although GANs have been widely used in computer vision tasks and are capable of producing visually appealing samples, they still face some challenges. First, GAN models are notoriously difficult to train. If the data distribution and generated distribution do not substantially overlap, the gradients of the generator will more or less point to random directions or lead to gradient vanishing. Second, GAN models often suffer from mode collapse issues, in which a trained model assigns all its probability to a small region in the target space [135, 141]. Third, appropriate network architectures and training parameters, such as batch size, learning rate and updating steps, are critical as unsuitable settings will significantly reduce the training stability and generative performance [77, 90]. Stabilising the training of GANs is not only a fundamental problem of learning deep generative models but also can largely improve the generative performance

of existing deep generative models. Many recent efforts related to GANs have focused on overcoming these training difficulties by developing various adversarial training objectives. Typically, assuming the optimal discriminator for the given generator is learned, different objective functions of the generator measure the distance between the data distribution and the generated distribution using different metrics. The original GAN uses Jensen-Shannon divergence as a metric, but a number of metrics have been introduced to improve its performance, such as least-squares [104], absolute deviation [182], Kullback-Leibler divergence [114, 127] and Wasserstein distance [2, 162]. However, according to both theoretical analyses and experimental results, each method has pros and cons. For example, although measuring Kullback-Leibler divergence largely eliminates the vanishing gradient issue, it easily results in model collapse [1, 127].

Although the adversarial learning paradigm has been successfully applied to train deep generative models, few works have explored the supervised transformation relationship between different data domains utilising the adversarial learning paradigm [169]. In most existing works, GANs are introduced into image-to-image transformation tasks and aim to generate visually realistic results. Specifically, GAN models (or adversarial learning paradigms) explore the distribution of the input and target data. Learning these real-world data distributions enables penalisation of images generated from the same distribution as the training data. However, most image-to-image transformation tasks also require the training model to be able to explore the mapping relations between paired training data (*i.e.*, input images and their corresponding ground truth). If the transformation information contained in the paired training data can be further explored, the performance of image-to-image transformation tasks may be further improved.

Most existing GAN-based frameworks perform end-to-end image generation. Although this is an easy task for training and testing, it is usually difficult to understand what happens during transformations. On the one hand, for different image synthesis tasks, different transformation models need to be trained, which requires much time. On the other hand, if the input image is not fully understood, it is difficult to accurately and flexibly manipulate the generated image. In order to improve models ability to re-render new images of the object of interest from a single image by specifying multiple scene properties, the properties of images

must be further understood. Therefore, many works have explored the disentangled representation of input images [7, 14, 45]. Unlike most deep learning models, which focus on learning of hierarchical representations, disentangled representations correspond to different factors (*e.g.*, identity, viewpoint) of the input image in the model described in this paper. Existing methods of exploring disentangled representations of input images share some important limitations [163]. For most existing methods, (disentangled) representations are mostly extracted from images themselves and the valuable tag information (*e.g.*, photographing conditions and object characterisations) associated with images has not been finely explored. In addition, there have been few attempts to make the re-rendering results more realistic by increasing the difficulty of distinguishing genuine and re-rendered images.

1.2 Related Works

Over the past few decades, there has been substantial growth in deep generative models [87]. [63] proposed an unsupervised learning algorithm for a deep generative model termed deep belief networks (DBN). This work stimulated a significant increase in the number of deep generative models. In statistics, a generative model can randomly generate observable values based on a joint probability distribution. For machine learning researchers, generative models are generally used to generate values for any variable in a model [87]. The deep generative model has a deep, multi-layered architecture. Comparing with shallower architectures, it achieves better performance in terms of generalisation and expression. Nowadays, the deep generative model has been used in a variety of machine learning applications for object detection [51], deblurring [32] and image generation [36].

GAN is an excellent framework for learning deep generative models, which aim to capture probability distributions within given data. GANs consist of generative and discriminative networks. GAN is more easily trained than other generative models by alternately updating a generator and discriminator using the back-propagation algorithm. In addition, for many generative tasks, GANs produce better samples than other generative models [53]. In the following section, the original GAN's formulate is reviewed. Then, some variants of GAN models are

presented. Finally, some recent works that use GAN for synthesising and editing real-world images are discussed.

1.3 Generative adversarial networks (GAN)

GANs [54] are an important approach to learning generative models that generate samples from real-world data distributions. GANs consist of a generative network, G , and a discriminative network, D . Given a noisy sample, $z \sim p(z)$ (sampled from a uniform or normal distribution), as the input, G outputs new data, $G(z)$, whose distribution, p_g , should be close to that of the data distribution, p_{data} . Meanwhile, D is employed to distinguish the true data sample, $x \sim p_{\text{data}}(x)$, and the generated sample, $G(z) \sim p_g(G(z))$. G tries to confuse D by generating increasingly realistic samples. In the original GAN, this adversarial training process can be formulated as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (1.1)$$

According to this adversarial loss function, D is trained to maximise the log-likelihood of assigning the correct label to both training (real) samples and generated (fake) samples, while G is trained to minimise $\log(1 - D(G(z)))$ in an attempt to prevent D from assigning a fake label to generated samples. Using the adversarial training process, G will be trained to generate ever more realistic samples.

According to theoretical analysis, the effect of training the generator was equal to that of minimising the Jensen-Shannon divergence between the data distribution and generated distribution. However, some problems still exist in the training process for GANs. For example, when the generated distribution and data distribution do not substantially overlap, the trained GAN model may suffer from gradient vanishing since the Jensen-Shannon divergence becomes a constant. Therefore, although the original GAN is easy to analyse and performs well with the MNIST dataset, it usually fails to handle high-dimensional data distributions.

1.4 Variants of GANs

Due to their great performance for learning real-world distributions, a large number of works have examined GANs. Some aimed to train a better generative model, such as the DCGAN, least-square GAN, energy-based GAN [182], WGAN [2] and WGAN with gradient penalty (WGAN-GP). Others integrated the GANs into their models to improve the performance of classical tasks. For example, the PGN [100] can be used for video prediction, SRGAN [88] for super-resolution, ID-CGAN [178] for image de-raining, the iGAN [185] for interactive applications, IAN [16] for photo modification and context-encoder for image in-painting [121]. Most recently, Isola *et al.* [68] proposed to use pix2pix-cGANs for several image-to-image transformation tasks (termed image-to-image translations in their work), such as translating semantic labels into a street scene, object edges into pictures, aerial photos into maps and so on [64, 74, 79, 105, 118, 155].

Although many GANs-based methods have been proposed and have achieved promising results, they are often difficult to train. If the data distribution and generated distribution do not substantially overlap (usually at the beginning of training), the generator gradients can point to more or less random directions or even result in gradient vanishing. GANs also suffer from model collapse (i.e. the generator assigns all its probability mass to a small region in the space) [3]. In addition, appropriate hyper-parameters (e.g. learning rate and updating steps) and network architectures are critical as unsuitable settings reduce GAN's performance or fail to produce reasonable results [60, 156].

1.5 Summary of Contributions.

The main aim of this thesis is to utilise generative modeling and the adversarial learning paradigm to synthesise real-world images and overcome the inherent limitations of existing methods.

It first stabilised the training process of GANs. To exploit the advantages and avoid the weaknesses of different metrics (*i.e.*, GAN objectives), this thesis devises a framework that utilises different metrics to jointly optimise the generator. In doing so, the proposed framework improves both training stability and gen-

erative performance. Specifically, an evolutionary generative adversarial network (E-GAN) is built that treats the adversarial training procedure as an evolutionary problem. A discriminator acts as the *environment* (*i.e.*, performs adaptive loss functions), and a *population* of generators evolves in response to the environment. During each adversarial iteration, the discriminator is trained to recognise real and fake samples. Generators undergo different *mutations*, acting as parents and producing offspring that are increasingly better adapted to the environment. Different adversarial objective functions aim to minimise different distances between the generated distribution and data distribution, leading to different mutations. Given the current optimal discriminator, we measure the quality and diversity of samples generated by the updated offspring. Finally, according to the principle of survival of the fittest, poorly-performing offspring are removed and the remaining well-performing offspring are preserved and used for further training. Based on the adversarial and evolutionary paradigms, the proposed E-GAN overcomes the inherent limitations of individual adversarial training objectives and always preserves the best offspring produced by different training objectives (*i.e.* mutations). In this way, the proposed framework contributes to the progress and success of GANs.

Second, utilising the adversarial learning paradigm within the GAN framework, this thesis developed a novel deep generative framework, PAN, to perform related image synthesis tasks. Both generative and perceptual adversarial loss are employed to train the PAN. Similar to GANs, the generative adversarial loss is utilised as a static measurement to evaluate the distribution of output images. Representations on the hidden layers of the discriminative network are adopted to perform dynamic perceptual measurement (*i.e.* perceptual adversarial loss) of output and ground truth images. Specifically, the network is trained to generate an output image that has the same high-level features as the corresponding ground truth image. On the other hand, if less discrepancy is measured on a particular hidden layer, the layer will be updated for a new high-dimensional space that increases the discrepancy between the output and ground truth images. The proposed perceptual adversarial loss undergoes an adversarial training process with the image transformation network and discriminative network, and it has the ability to discover and decrease the discrepancy on hidden layers that

have been constantly updated.

Third, in order to improve the ability to re-render new images of the object of interest from a single image by specifying multiple scene properties, this thesis devises a principled tag disentangled generative adversarial network (*TD-GAN*). The whole framework consists of four parts: a disentangling network, generative network, tag mapping net and discriminative network, which are trained jointly based on a given set of images that are completely or partially tagged (for supervised or semi-supervised settings, respectively) [116]. The disentangling network extracts disentangled and interpretable representations from an input image, which are then used to generate images using the generative network. In order to improve the quality of the obtained disentangled representations, a tag mapping net is integrated to explore the consistency between the image and its tags. Since the image and its tags record the same object from two different perspectives, they should share the same disentangled representations. Furthermore, the discriminative network is introduced to implement the adversarial training strategy to generate more realistic images.

In summary, this thesis makes the following contributions:

- A novel GAN framework, E-GAN, is proposed for stable GAN training and improved generative performance. Unlike existing GANs, which employ a pre-defined adversarial objective function to alternately train a generator and discriminator, we utilise different adversarial training objectives as mutation operations and evolve a population of generators to adapt to the environment (*i.e.*, the discriminator). We also design a novel evaluation mechanism to measure the quality and diversity of generated samples, and only well-performing generators are preserved and used for further training. In this way, E-GAN overcomes the limitations of an individual adversarial training objective and always preserves the best offspring, contributing to the progress and success of GANs.
- PAN is proposed for image-to-image transformation tasks. Unlike existing application-driven algorithms, PAN provides a generic framework of learning to map input images to desired images, such as a rainy image to its de-rained counterpart, object edges to its photo, semantic labels to a

scene image and so on. Besides generative adversarial loss, which is widely used in GANs, we propose to use perceptual adversarial loss, which undergoes an adversarial training process between the image transformation network and hidden layers of the discriminative network. The hidden layers and the output of the discriminative network are updated to constantly and automatically discover the discrepancy between the transformed image and corresponding ground truth, while the image transformation network is trained to minimise the discrepancy explored by the discriminative network.

- A principled TD-GAN is devised to re-render new images of the object of interest from a single image by specifying multiple scene properties (such as viewpoint, illumination, expression). The framework consists of a disentangling network, generative network, tag mapping net and discriminative network, which are jointly trained based on a set of images that are completely or partially tagged (for supervised or semi-supervised settings, respectively). Experiments with two challenging datasets demonstrate the state-of-the-art performance of the proposed framework regarding the problem of interest.

1.6 Thesis Structure.

This thesis aims to synthesise and manipulate real-world images through generative modelling and adversarial training. The remainder of this thesis is organised as follows (note that the background and related works are separately summarized in each of the following chapters):

- Chapter 2 develops a novel framework for training GAN models. The proposed E-GAN will be illustrated and discussed. Related experiments demonstrate that E-GAN indeed has the ability to stabilise the GANs training process [154].
- Chapter 3 introduces the adversarial learning paradigm into image-to-image transformation tasks. The devised perceptual adversarial loss will be illustrated and explained. When perceptual adversarial loss is combined with

generative adversarial loss, a novel framework, PAN, performs well for a series of image-to-image transformation tasks [153].

- Chapter 4 proposes a novel TD-GAN for image re-rendering tasks. Within the GANs framework, disentangled representations are explored by penalising the consistency between images and corresponding descriptions [152].
- Finally, Chapter 5 concludes this thesis and proposes recommendations for future work.

Chapter 2

Stabilizing adversarial training process via selecting optimization

Generative adversarial networks (GAN) have been effective for learning generative models for real-world data. However, existing GANs (GAN and its variants) tend to suffer from training problems such as instability and model collapse. In this chapter, I propose a novel GAN framework called evolutionary generative adversarial networks (E-GAN) for stable GAN training and improved generative performance. Unlike existing GANs, which employ a pre-defined adversarial objective function alternately training a generator and a discriminator, different adversarial training objectives are utilized as mutation operations and evolve a population of generators to adapt to the environment (i.e., the discriminator). We also design an evaluation mechanism to measure the quality and diversity of generated samples, such that only well-performing generator(s) are preserved and used for further training. In this way, E-GAN overcomes the limitations of an individual adversarial training objective and always preserves the best offspring, contributing to progress in and the success of GANs. Experiments on several datasets demonstrate that E-GAN achieves convincing generative performance and reduces the training problems inherent in existing GANs.

2.1 Introduction

Generative adversarial networks (GAN) [54] are one of the main groups of methods used to learn generative models from complicated real-world data. As well as using a generator to synthesize semantically meaningful data from standard signal distributions, GANs (GAN and its variants) train a discriminator to distinguish *real* samples in the training dataset from *fake* samples synthesized by the generator. As the confronter, the generator aims to deceive the discriminator by producing ever more realistic samples. The training procedure continues until the generator wins the adversarial game; that is, the discriminator cannot make a better decision than randomly guessing whether a particular sample is fake or real. GANs have recently been successfully applied to image generation [21, 47, 113, 176], image editing [69, 93, 147, 152, 186], video prediction [5, 44, 146, 148, 165], and many other tasks [66, 101, 120, 143, 180].

Although GANs already produce visually appealing samples in various applications, they are often difficult to train. If the data distribution and the generated distribution do not substantially overlap (usually at the beginning of training), the generator gradients can point to more or less random directions resulted in the vanishing gradient issue. GANs also suffer from model collapse, *i.e.*, the generator assigns all its probability mass to a small region in the space [3]. In addition, appropriate hyper-parameters (*e.g.*, learning rate and updating steps) and network architectures are critical configurations in GANs. Unsuitable settings reduce GAN’s performance or even fail to produce any reasonable results.

Many recent efforts on GANs have focused on overcoming these training difficulties by developing various adversarial training objectives. Typically, assuming the optimal discriminator for the given generator is learned, different objective functions of the generator aim to measure the distance between the data distribution and the generated distribution under different metrics. The original GAN uses Jensen-Shannon divergence as the metric. A number of metrics have been introduced to improve GAN’s performance, such as least-squares [104], absolute deviation [182], Kullback-Leibler divergence [114, 127], and Wasserstein distance [2]. However, according to both theoretical analyses and experimental results, minimizing each distance has its own pros and cons. For example, al-

though measuring Kullback-Leibler divergence largely eliminates the vanishing gradient issue, it easily results in model collapse [1, 127]. Likewise, Wasserstein distance greatly improves training stability but sometimes leads to pathological behavior [58].

To exploit the advantages and suppress the weaknesses of different metrics (*i.e.*, GAN objectives), we devise a framework that utilizes different metrics to jointly optimize the generator. In doing so, we improve both the training stability and generative performance. We build an evolutionary generative adversarial network (E-GAN), which treats the adversarial training procedure as an evolutionary problem. Specifically, a discriminator acts as the *environment* (*i.e.*, provides adaptive loss functions) and a *population* of generators evolves in response to the environment. During each adversarial (or evolutionary) iteration, the discriminator is still trained to recognize real and fake samples. However, in our method, acting as parents, generators undergo different *mutations* to produce offspring to adapt to the environment. Different adversarial objective functions aim to minimize different distances between the generated distribution and the data distribution, leading to different mutations. Meanwhile, given the current optimal discriminator, we measure the quality and diversity of samples generated by the updated offspring. Finally, according to the principle of “survival of the fittest”, poorly-performing offspring are removed and the remaining well-performing offspring (*i.e.*, generators) are preserved and used for further training.

Based on the evolutionary paradigm to optimize GANs, the proposed E-GAN overcomes the inherent limitations in the individual adversarial training objectives and always preserves the best offspring produced by different training objectives (*i.e.*, mutations). In this way, we contribute to progress in and the success of GANs. Experiments on several datasets demonstrate the advantages of integrating different adversarial training objectives and E-GAN’s convincing performance for image generation.

2.2 Related Works

In this section, we first review some previous GANs devoted to reducing training instability and improving the generative performance. We then briefly summarize

some evolutionary algorithms on deep neural networks.

2.2.1 Generative Adversarial Networks

Generative adversarial networks (GAN) provides an excellent framework for learning deep generative models, which aim to capture probability distributions over the given data. Compared to other generative models, GAN is easily trained by alternately updating a generator and a discriminator using the back-propagation algorithm. In many generative tasks, GANs (GAN and its variants) produce better samples than other generative models [53].

However, some problems still exist in the GANs training process. In the original GAN, training the generator was equal to minimizing the Jensen-Shannon divergence between the data distribution and the generated distribution, which easily resulted in the vanishing gradient problem. To solve this issue, a non-saturating heuristic objective (*i.e.*, ‘ $-\log D$ trick’) replaced the minimax objective function to penalize the generator [54]. Then, [127] and [133] designed specified network architectures (DCGAN) and proposed several heuristic tricks (*e.g.*, feature matching, one-side label smoothing, virtual batch normalization) to improve training stability. Meanwhile, energy-based GAN [182] and least-squares GAN [104] improved training stability by employing different training objectives. Although these methods partly enhanced training stability, in practice, the network architectures and training procedure still required careful design to maintain the discriminator-generator balance. More recently, Wasserstein GAN (WGAN) [2] and its variant WGAN-GP [58] were proposed to minimize the Wasserstein-1 distance between the generated and data distributions. Since the Wasserstein-1 distance is continuous everywhere and differentiable almost everywhere under only minimal assumptions [2], these two methods convincingly reduce training instability. However, to measure the Wasserstein-1 distance between the generated distribution and the data distribution, they are asked to enforce the Lipschitz constraint on the discriminator (*aka* critic), which may restrict critic capability and result in some optimization difficulties [58, 126].

2.2.2 Evolutionary Algorithms

Over the last twenty years, evolutionary algorithms have achieved considerable success across a wide range of computational tasks including modeling, optimization and design [27, 37]. Inspired by natural evolution, the essence of an evolutionary algorithm is to equate possible solutions to individuals in a population, produce offspring through variations, and select appropriate solutions according to fitness [38].

Recently, evolutionary algorithms have been introduced to solve deep learning problems. To minimize human participation in designing deep algorithms and automatically discover such configurations, there have been many attempts to optimize deep learning hyper-parameters and design deep network architectures through an evolutionary search [108, 128, 173]. Evolutionary algorithms have also demonstrated their capacity to optimize deep neural networks [61, 85, 171]. Moreover, [134] proposed a novel evolutionary strategy as an alternative to the popular MDP-based reinforcement learning (RL) techniques, achieving strong performance on RL benchmarks. Last but not least, an evolutionary algorithm was proposed to compress deep learning models by automatically eliminating redundant convolution filters [158].

2.3 Method

In this section, we first review the GAN formulation. Then, we introduce the proposed E-GAN algorithm. By illustrating E-GAN’s mutations and evaluation mechanism, we further discuss the advantage of the proposed framework. Finally, we conclude with the entire E-GAN training process.

2.3.1 Generative Adversarial Networks

GAN, first proposed in [54], studies a two-player minimax game between a discriminative network D and a generative network G . Taking noisy sample $z \sim p(z)$ (sampled from a uniform or normal distribution) as the input, the generative network G outputs new data $G(z)$, whose distribution p_g is supposed to be close to that of the data distribution p_{data} . Meanwhile, the discriminative network D is

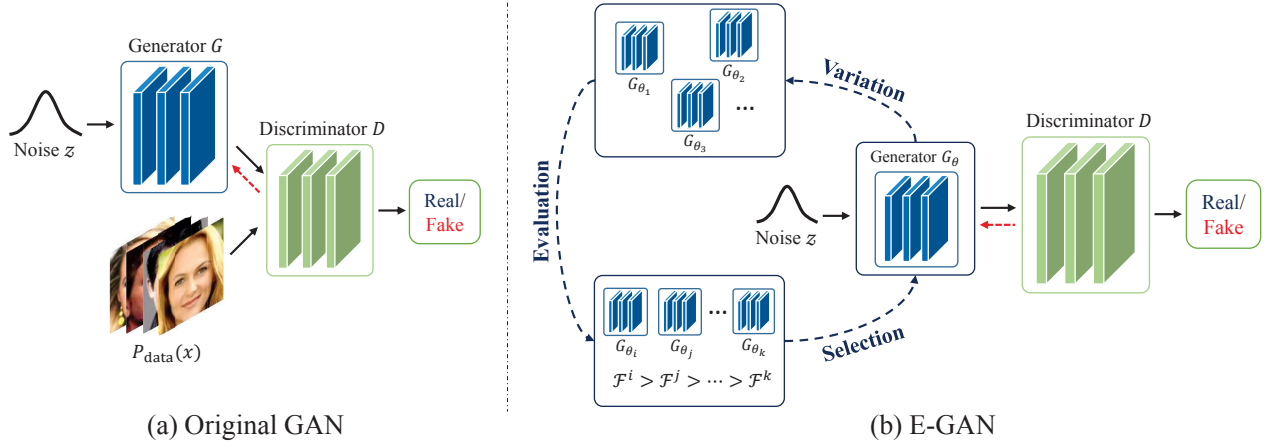


Figure 2.1: (a) The original GAN framework. A generator G and a discriminator D play a two-player adversarial game. The updating gradients of the generator G are received from the adaptive objective, which depends on discriminator D . (b) The proposed E-GAN framework. A population of generators $\{G_{\theta}\}$ evolves in a dynamic environment, the discriminator D . Each evolutionary step consists of three sub-stages: variation, evaluation, and selection. The best offspring are kept.

employed to distinguish the true data sample $x \sim p_{\text{data}}(x)$ and the generated sample $G(z) \sim p_g(G(z))$. In the original GAN, this adversarial training process was formulated as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (2.1)$$

The adversarial procedure is illustrated in Fig. 2.1 (a). Most existing GANs perform a similar adversarial procedure in different adversarial objective functions.

2.3.2 Evolutionary Algorithm

In contrast to conventional GANs, which alternately update a generator and a discriminator, we devise an evolutionary algorithm that evolves a population of generator(s) $\{G\}$ in a given environment (*i.e.*, the discriminator D). In this population, each *individual* represents a possible solution in the parameter space of the generative network G . During the evolutionary process, we expect that the population gradually adapts to its environment, which means that the evolved generator(s) can generate ever more realistic samples and eventually learn the real-world data distribution. As shown in Fig. 2.1 (b), during evolution, each step consists of three sub-stages:

- **Variation:** Given an individual G_θ in the population, we utilize the variation operators to produce its offspring $\{G_{\theta_1}, G_{\theta_2}, \dots\}$. Specifically, several copies of each individual—or *parent*—are created, each of which modified by different *mutations*. Then, each modified copy is regarded as one *child*.
- **Evaluation:** For each child, its performance—or *individual’s quality*—is evaluated by a *fitness* function $\mathcal{F}(\cdot)$ that depends on the current environment (*i.e.*, discriminator D).
- **Selection:** All children will be selected according to their fitness value, and the worst part is removed—that is, they are *killed*. The rest remain *alive* (*i.e.*, free to act as parents), and evolve to the next iteration.

After each evolutionary step, the discriminative network D (*i.e.*, the environment) is updated to further distinguish real samples x and fake samples y generated by the evolved generator(s), *i.e.*,

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{y \sim p_g}[\log(1 - D(y))]. \quad (2.2)$$

Thus, the discriminative network D (*i.e.*, the environment) can continually provide the adaptive losses to drive the population of generator(s) evolving to produce better solutions. Next, we illustrate and discuss the proposed variation (or mutation) and evaluation operators in detail.

Through selecting from multiple generators in each iteration, our framework leverages strengths of different functions, and avoids ‘bad’ updating during training. Firstly, gradient vanishing (or point to random) happens when D easily distinguishes real and fake samples. Choosing from multiple G s according to \mathcal{F}_q (*i.e.*, Eq.1.6) can find the suitable G that deceives D as much as possible. Moreover, [111] explores that mode collapse usually accompanies with occasional surges of D ’s gradient-norm. Choosing from multiple G according to \mathcal{F}_d (*i.e.*, Eq.1.7) can find the suitable G that suppresses the mode collapse as much as possible.

2.3.3 Mutations

We employ *asexual reproduction* with different mutations to produce the next generation’s individuals (*i.e.*, children). Specifically, these mutation operators correspond to different training objectives, which attempt to narrow the distances between the generated distribution and the data distribution from different perspectives. In this section, we introduce the mutations used in this work¹. To analyze the corresponding properties of these mutations, we suppose that, for each evolutionary step, the optimal discriminator $D^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$, according to Eq. (2.2), has already been learned [54].

Here, all three G ’s losses are carefully verified and selected. Firstly, they all achieve reasonable results in their original models, and share the equivalent optimal D . Then, theoretically, these three losses have complementary advantages. For example, the minimax loss easily encounters gradient vanishing when the generated distribution and data distribution do not have substantially overlapping (since JSD is a constant), while heuristic loss suppresses this issue through measuring KL divergence which provides effective gradients in such case

¹More mutation operations were tested, but the mutation approaches described already delivered a convincing performance.

2.3.3.1 Minimax mutation

The minimax mutation corresponds to the minimax objective function in the original GAN:

$$\mathcal{M}_G^{\text{minimax}} = \frac{1}{2} \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (2.3)$$

According to the theoretical analysis in [54], given the optimal discriminator D^* , the minimax mutation aims to minimize the Jensen-Shannon divergence (JSD) between the data distribution and the generated distribution. Although the minimax game is easy to explain and theoretically analyze, its performance in practice is disappointing, a primary problem being the generator’s vanishing gradient. If the support of two distributions lies in two manifolds, the JSD will be a constant, leading to the vanishing gradient [1]. This problem is also illustrated in Fig. 2.2. When the discriminator rejects generated samples with high confidence (*i.e.*, $D(G(z)) \rightarrow 0$), the gradient trends to vanishing. However, if the generated distribution overlaps with the data distribution, meaning that the discriminator cannot completely distinguish real from fake samples, the minimax mutation provides effective gradients and continually narrows the gap between the data distribution and the generated distribution.

2.3.3.2 Heuristic mutation

Unlike the minimax mutation, which minimizes the log probability of the discriminator being correct, the heuristic mutation aims to maximize the log probability of the discriminator being mistaken, *i.e.*,

$$\mathcal{M}_G^{\text{heuristic}} = -\frac{1}{2} \mathbb{E}_{z \sim p_z} [\log(D(G(z)))]. \quad (2.4)$$

Compared to the minimax mutation, the heuristic mutation will not saturate when the discriminator rejects the generated samples. Thus, the heuristic mutation avoids vanishing gradient and provides useful generator updates (Fig. 2.2). However, according to [1], given the optimal discriminator D^* , minimizing the heuristic mutation is equal to minimizing $[KL(p_g || p_{\text{data}}) - 2JSD(p_g || p_{\text{data}})]$, *i.e.*, inverted KL minus two JSDs. Intuitively, the JSD sign is negative, which means pushing these two distributions away from each other. In practice, this may lead

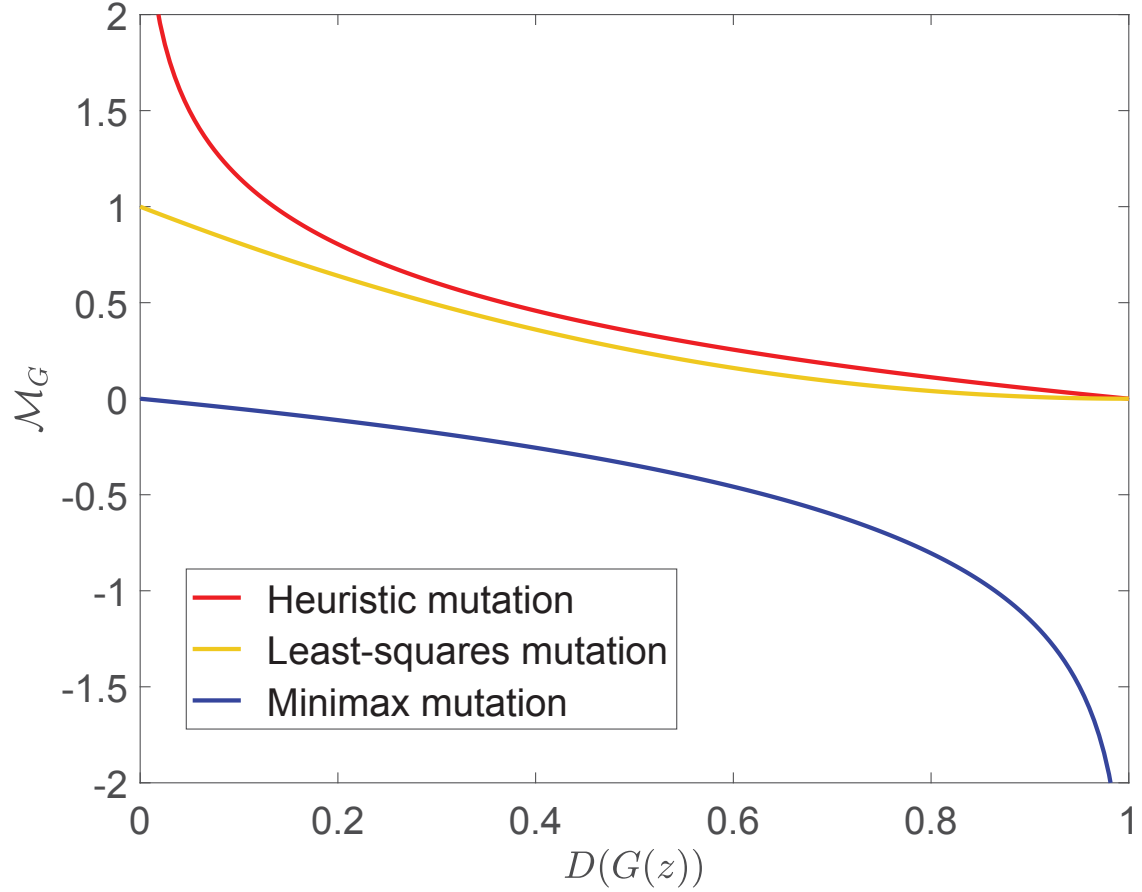


Figure 2.2: The mutation (or objective) functions that the generator G receives given the discriminator D .

to training instability and generative quality fluctuations [58].

2.3.3.3 Least-squares mutation

The least-squares mutation is inspired by LSGAN [104], where the least-squares objectives are utilized to penalize its generator to deceive the discriminator. In

this work, we formulate the least-squares mutation as:

$$\mathcal{M}_G^{\text{least-square}} = \mathbb{E}_{z \sim p_z} [(D(G(z)) - 1)^2]. \quad (2.5)$$

As shown in Fig. 2.2, the least-squares mutation is non-saturating when the discriminator can recognize the generated sample (*i.e.*, $D(G(z)) \rightarrow 0$). When the discriminator output grows, the least-squares mutation saturates, eventually approaching zero. Therefore, similar to the heuristic mutation, the least-squares mutation can avoid vanishing gradient when the discriminator has a significant advantage over the generator. Meanwhile, compared to the heuristic mutation, although the least-squares mutation will not assign an extremely high cost to generate fake samples, it will also not assign an extremely low cost to mode dropping¹, which partly avoids model collapse [104].

Note that, different from GAN-minimax and GAN-heuristic, LSGAN employs a different loss (‘least-squares’) from ours (Eq. (2.2)) to optimize the discriminator. Yet, as shown in the Appendix , the optimal discriminator of LSGAN is equivalent to ours. Therefore, although we employ only one discriminator as the environment to distinguish real and generated samples, it is sufficient to provide adaptive losses for mutations described above.

Overall, we utilize different adversarial objectives as mutations to search possible updating (or evolutionary) directions for generator(s). In addition, we employ a single discriminator, which is trained by the sigmoid cross-entropy (Eq. (2.2)), as environment to provide adaptive losses for generator(s).

2.3.4 Evaluation

In an evolutionary algorithm, evaluation is the operation of measuring the quality of individuals. To determine the evolutionary direction (*i.e.*, individuals’ selection), we devise an evaluation (or fitness) function to measure the performance of evolved individuals (*i.e.*, children). Typically, we focus on two generator properties: 1) the quality and 2) the diversity of generated samples. First, we simply

¹ [1] demonstrated that the heuristic objective suffers from model collapse since $KL(p_g || p_{\text{data}})$ assigns a high cost to generating fake samples but an extremely low cost to mode dropping.

Algorithm 1 E-GAN. Default values $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$, $n_D = 2$, $n_p = 1$, $n_m = 3$, $m = 16$.

Require: the batch size m . the discriminator’s updating steps per iteration n_D . the number of parents n_p . the number of mutations n_m . Adam hyper-parameters α, β_1, β_2 , the hyper-parameter γ of evaluation function.

Require: initial discriminator’s parameters w_0 . initial generators’ parameters $\{\theta_0^1, \theta_0^2, \dots, \theta_0^{n_p}\}$.

```

1: for number of training iterations do
2:   for  $k = 0, \dots, n_D$  do
3:     Sample a batch of  $\{x^{(i)}\}_{i=1}^m \sim p_{\text{data}}$  (training data), and a batch of
        $\{z^{(i)}\}_{i=1}^m \sim p_z$  (noise samples).
4:      $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m \log D_w(x^{(i)})$ 
5:        $+ \frac{1}{m} \sum_{j=1}^{n_p} \sum_{i=1}^{m/n_p} \log(1 - D_w(G_{\theta^j}(z^{(i)})))]$ 
6:      $w \leftarrow \text{Adam}(g_w, w, \alpha, \beta_1, \beta_2)$ 
7:   end for
8:   for  $j = 0, \dots, n_p$  do
9:     for  $h = 0, \dots, n_m$  do
10:      Sample a batch of  $\{z^{(i)}\}_{i=1}^m \sim p_z$ 
11:       $g_{\theta^j, h} \leftarrow \nabla_{\theta^j} \mathcal{M}_G^h(\{z^{(i)}\}_{i=1}^m, \theta^j)$ 
12:       $\theta_{\text{child}}^{j, h} \leftarrow \text{Adam}(g_{\theta^j, h}, \theta^j, \alpha, \beta_1, \beta_2)$ 
13:       $\mathcal{F}_q^{j, h} \leftarrow \mathcal{F}_q^{j, h} + \gamma \mathcal{F}_d^{j, h}$ 
14:    end for
15:   end for
16:    $\{\mathcal{F}_q^{j_1, h_1}, \mathcal{F}_q^{j_2, h_2}, \dots\} \leftarrow \text{sort}(\{\mathcal{F}_q^{j, h}\})$ 
17:    $\theta^1, \theta^2, \dots, \theta^{n_p} \leftarrow \theta_{\text{child}}^{j_1, h_1}, \theta_{\text{child}}^{j_2, h_2}, \dots, \theta_{\text{child}}^{j_{n_p}, h_{n_p}}$ 
18: end for

```

feed generator produced images into the discriminator D and observe the average value of the output, which we name the *quality fitness score*:

$$\mathcal{F}_q = \mathbb{E}_z[D(G(z))]. \quad (2.6)$$

Note that discriminator D is constantly upgraded to be optimal during the training process, reflecting the quality of generators at each evolutionary (or adversarial) step. If a generator obtains a relatively high quality score, its generated samples can deceive the discriminator and the generated distribution is further approximate to the data distribution.

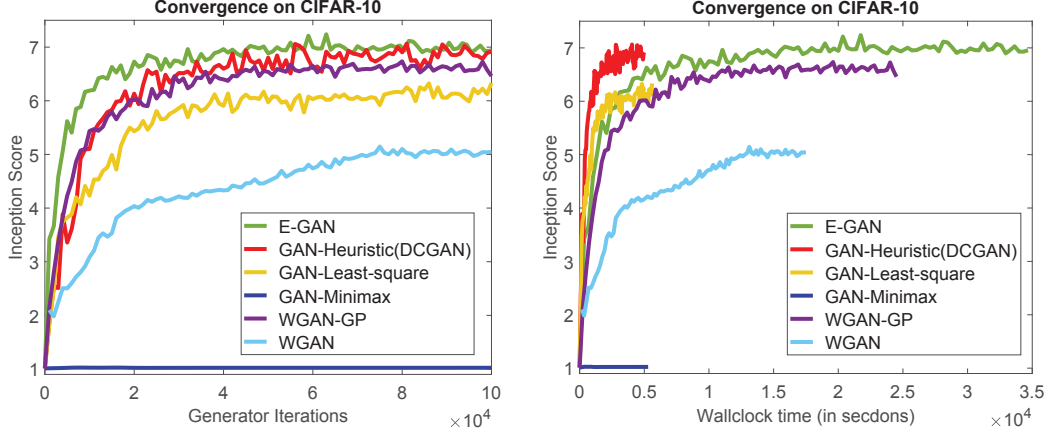


Figure 2.3: Experiments on the CIFAR-10 dataset. CIFAR-10 inception score over generator iterations (left), over wall-clock time (right).

Besides generative quality, we also pay attention to the diversity of generated samples and attempt to overcome the model collapse issue in GAN optimization. Recently, [111] proposed a gradient-based regularization term to stabilize the GAN optimization and suppress model collapse. The regularization term provides some “foresight” for training generator G , *i.e.*, when the generator collapses to a small region, the discriminator will subsequently label these collapsed points as fake data with “obvious countermeasure” (*i.e.*, big gradients).

We employ a similar principle to evaluate generator optimization stability and generative diversity. Formally, the *diversity fitness score* is defined as:

$$\mathcal{F}_d = -\log \left\| \nabla_D - \mathbb{E}_x[\log D(x)] - \mathbb{E}_z[\log(1 - D(G(z)))] \right\|. \quad (2.7)$$

Instead of predicting the generator’s “foresight” as in [111], The log gradient value of updating D is utilized to measure the diversity of generated samples. If the updated generator obtains a relatively high diversity score, which corresponds to small discriminator gradients, its generated samples trend to spread out enough, to avoid the discriminator has obvious countermeasures. Thus, the model collapse issue can be suppressed and the discriminator will change smoothly, which helps to improve the training stability. The samples generated by a generator are fed to the latest discriminator, and we calculate the log gradient value of updating the discriminator, thereby avoiding the “double backprop” in the GP term optimization.

Based on the aforementioned two fitness scores, we can finally give the evaluation (or fitness) function of the proposed evolutionary algorithm:

$$\mathcal{F} = \mathcal{F}_q + \gamma \mathcal{F}_d, \quad (2.8)$$

where $\gamma \geq 0$ balances two measurements: generative quality and diversity. Overall, a relatively high fitness score \mathcal{F} , leads to higher training efficiency and better generative performance.

In experiments, \mathcal{F}_q (*i.e.*, Eq.1.6) lies in $[0,1]$, while the gradient-norm in Eq.1.7 can vary from 1 to 1000 w.r.t. different D . The logarithm shrinks \mathcal{F}_d (*i.e.*, Eq.7) to have similar magnitude with \mathcal{F}_q , which helps to decrease the sensitiveness of γ . We run grid search to find γ 's value based on the observed 'fitness scores' (*i.e.*, \mathcal{F}_q , \mathcal{F}_d) and sample quality. In practice, we find γ is highly related to D 's scale and choose $\gamma = 0.5$ for toy data, 0.02 for real-world data.

2.3.5 E-GAN

Having introduced the proposed evolutionary algorithm and corresponding mutations and evaluation criteria, the complete E-GAN training process is concluded in Algorithm 1. Overall, in E-GAN, generators $\{G\}$ are regarded as an evolutionary population and discriminator D acts as an environment. For each evolutionary step, generators are updated with different objectives (or mutations) to accommodate the current environment. According to the principle of "survival of the fittest", only well-performing children will survive and participate in future adversarial training. Unlike the two-player game with a fixed and static adversarial training objective in conventional GANs, E-GAN allows the algorithm to integrate the merits of different adversarial objectives and generate the most competitive solution. Thus, during training, the evolutionary algorithm not only largely suppresses the limitations (vanishing gradient, model collapse, *etc.*) of individual adversarial objectives, but it also harnesses their advantages to search for a better solution.

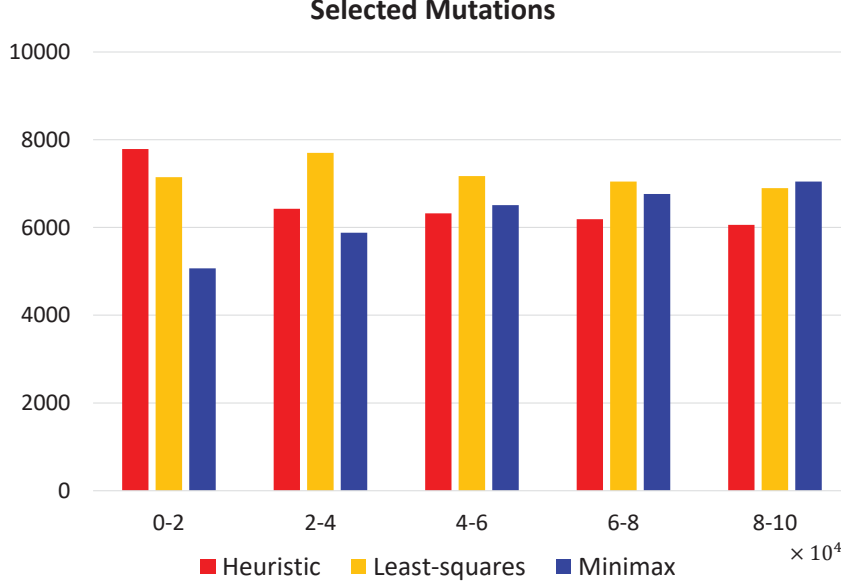


Figure 2.4: Experiments on the CIFAR-10 dataset. The graph of selected mutations in the E-GAN training process

2.4 Experiments

To evaluate the proposed E-GAN, in this section, we run and analyze experiments on several generation tasks.

2.4.1 Implementation Details

We evaluate E-GAN on two synthetic datasets and three image datasets: CIFAR-10 [81], LSUN bedroom [175], and CelebA [98]. For all of these tasks, the network architectures are based on DCGAN [127] and are briefly introduced here, more details can be found in the Appendix . We use the default hyper-parameter values listed in Algorithm 1 for all experiments. Note that the number of parents

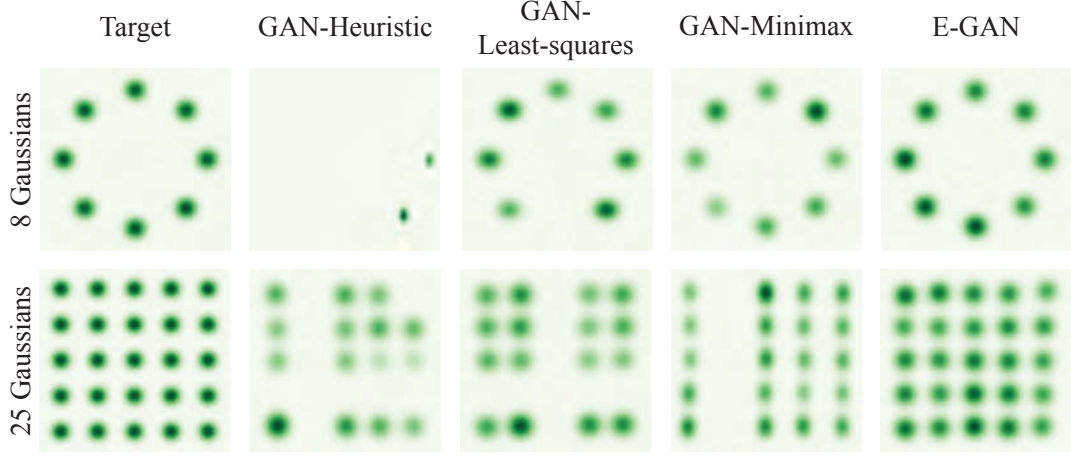


Figure 2.5: KDE plots of the target data and generated data from different GANs trained on mixtures of Gaussians.

n_p is set as 1, which means only one (*i.e.*, the best) child is retained in each evolutionary step. On the one hand, this reduces E-GAN’s computational cost, thereby accelerating training. On the other hand, our experiments show that E-GAN already achieves impressive performance and stability even with only one survivor at each step. Furthermore, all experiments were trained on Nvidia GTX 1080Ti GPUs. To train a model for 64×64 images using the DCGAN architecture cost around 30 hours on a single GPU.

In the first experiment, we adopt the experimental design proposed in [107], which trains GANs on 2D Gaussian mixture distributions. The model collapse issue can be accurately measured on these synthetic datasets, since we can clearly observe the data distribution and the generated distribution. As shown in Fig. 2.5, we employ two challenging distributions to evaluate E-GAN, a mixture of 8 Gaussians arranged in a circle and a mixture of 25 Gaussians arranged in a grid.¹

We first compare the proposed evolutionary adversarial training framework with one using an individual adversarial objective (*i.e.*, conventional GANs). We

¹We obtain both 2D distributions and network architectures from the code provided in [58].



Figure 2.6: Generated samples on 128×128 LSUN bedrooms.



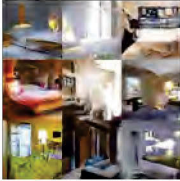


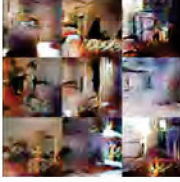
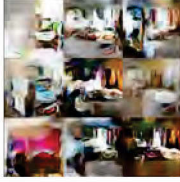
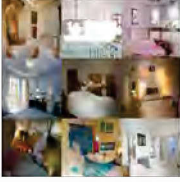
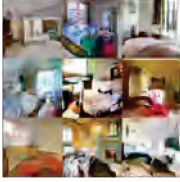

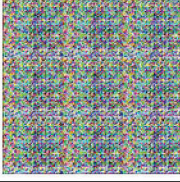

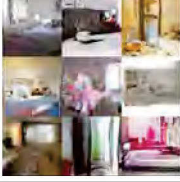
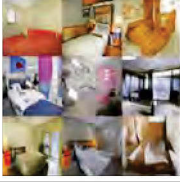
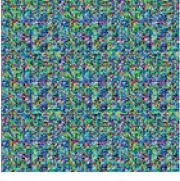
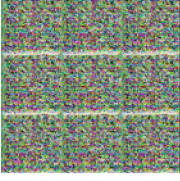
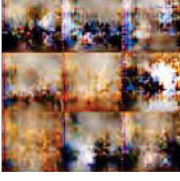
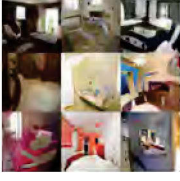
DCGAN	LSGAN	WGAN	WGAN-GP	ECGAN (ours)
Baseline (G: DCGAN, D: DCGAN)				
				
No BN and a constant number of filters in G and D				
				
G: No BN and a constant number of filters, D: DCGAN				
				
G: DCGAN, D: 2-Conv-1-FC LeakyReLU				
				

Figure 2.7: Experiments to test architecture robustness. Different GAN architectures corresponding to different training challenges and trained with five different GAN methods.

train each method 50K iterations and report the KDE plots in Fig. 2.5. The results show that all of the individual adversarial objectives suffer from model collapse to a greater or lesser degree. However, by combining different objectives in our evolution framework, model performance is largely improved and can accurately fit the target distributions. This further demonstrates, during the evolutionary procedure, the proposed evaluation mechanism can recognize well-performing updatings (*i.e.*, offspring), and promote the population to a better evolutionary direction.

When evaluating a GAN model, sample quality and convergence speed are two important criteria. We train different GANs on CIFAR-10 and plot inception scores [133] over the course of training (Fig. 5.10). The same network architecture based on DCGAN is used in all methods.

As shown in Fig. 5.10, E-GAN can get higher inception score with less training steps. Meanwhile, E-GAN also shows comparable stability when it goes to convergence. By comparison, conventional GANs expose their different limitations, such as instability at convergence (GAN-Heuristic), slow convergence (GAN-Least square) and invalid (GAN-minimax). As mentioned above, different objectives aim to measure the distance between the generated and data distributions under different metrics which have different pros and cons. Here, utilizing the evolutionary framework, E-GAN not only overcomes the limitations of these individual adversarial objectives, but it also outperforms other GANs (the WGAN and its improved variation WGAN-GP). Furthermore, although E-GAN takes more time for each iteration, it achieves comparable convergence speed in terms of wall-clock time (Fig. 5.10-right).

2.4.2 CIFAR-10 and Inception Score

During training E-GAN, we recorded the selected objective in each evolutionary step (Fig. 2.4). At the beginning of training, the heuristic mutation and the least-square mutation are selected more frequently than the minimax mutation. It may be due to the fact that the minimax mutation is hard to provide effective gradients (*i.e.*, vanishing gradient) when the discriminator can easily recognize generated samples. Along with the generator approaching convergence (after 20K steps),

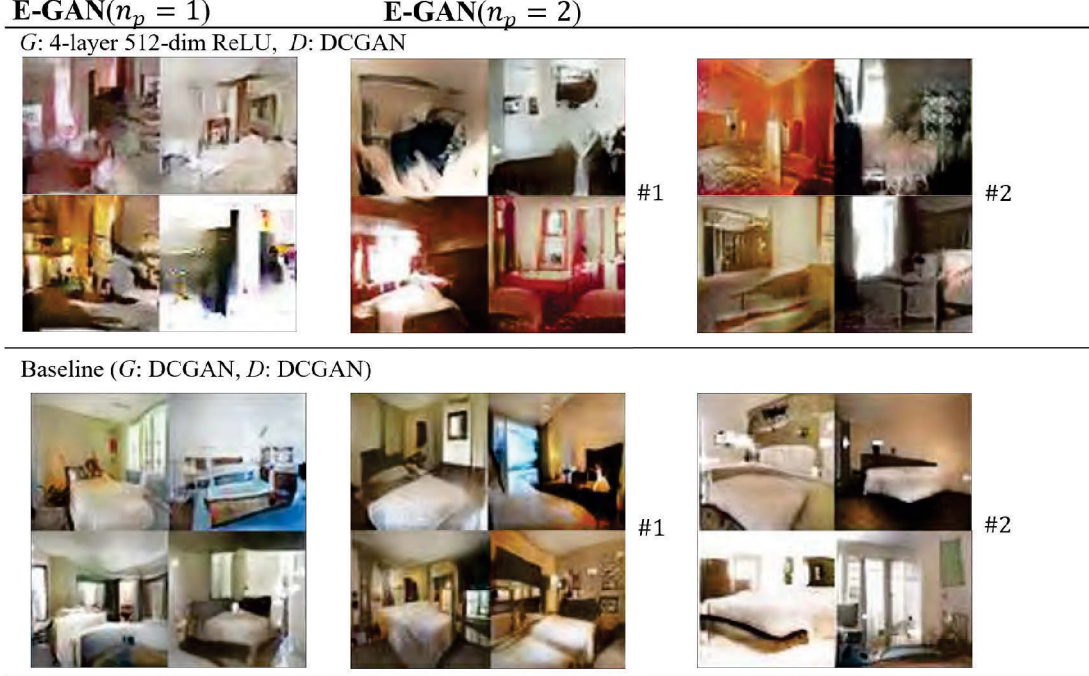


Figure 2.8: Keep different numbers of candidatures.

ever more minimax mutations are employed, yet the number of selected heuristic mutations is falling. As aforementioned, the minus JSDs of the heuristic mutation may trend to push the generated distribution away from data distribution and lead to training instability. However, in E-GAN, beyond the heuristic mutation, we have other options of mutation, which improves the stability at convergence.

In addition, we test Eq. 2.8 on cifar-10, and achieve an inception score around 6.2, which is inferior to those of E-GAN(~ 6.8) and WGAN-GP (~ 6.5). All three G 's losses currently used in our model are carefully selected due to their complementary advantages. Compare to using Eq.1.8 alone, selecting from multiple generators can lead to better results.

2.4.3 LSUN and Architecture Robustness

The architecture robustness is another advantage of E-GAN. To demonstrate the training stability of our method, we train different network architectures on the

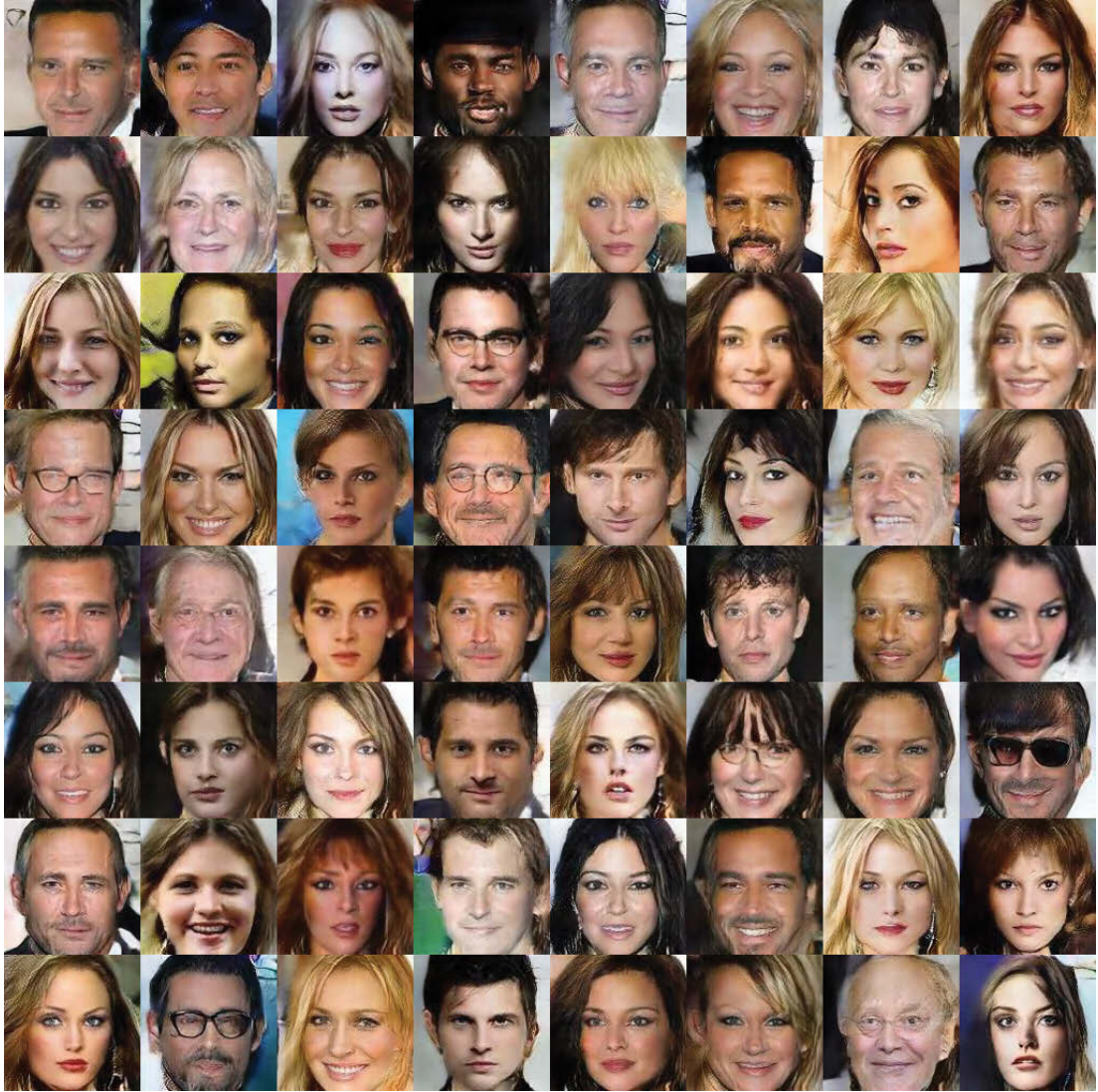


Figure 2.9: Generated human face images on the 128×128 CelebA dataset.

LSUN bedroom dataset [175] and compare with several existing works. In addition to the baseline DCGAN architecture, we choose three additional architectures corresponding to different training challenges: (1) limiting the recognition capability of the discriminator D , *i.e.*, 2-Conv-1-FC LeakyReLU discriminator; (2) limiting the expression capability of the generator G , *i.e.*, no batchnorm and a constant number of filters in the generator; (3) reducing the network capability of the generator and discriminator together, *i.e.*, remove the BN in both the generator G and discriminator D . For each architecture, we test five different methods: DCGAN, LSGAN, standard WGAN (with weight clipping), WGAN-GP (with gradient penalty) ,and our E-GAN. For each method, we used the default configurations recommended in the respective studies (these methods are summarized in [58]) and train each model for 200K iterations. As shown in Fig. 2.7, E-GAN generated reasonable results even when other methods failed. Furthermore, based on the DCGAN architecture, we train E-GAN to generate 128×128 bedroom images¹ (Fig. 2.6). Observing the generated images, we demonstrate that E-GAN can be trained to generate diversity and high-quality images from the target data distribution.

2.4.4 Synthetic Datasets and Model Collapse

Moreover, we preserved multiple children in our framework and have the following observations. As shown in Fig. 2.8 (first line), given extreme imbalance between G and D , preserving more children is beneficial for further stabilizing training progress and leading to better results. While, in the baseline setting (second line), preserving the best candidature is already capable of overcoming most inherent limitations within existing GANs, and preserving multiple children achieves a very limited increase in sample quality. As regard to training efficiency, if we regard updating and evaluating a child G as an operation and define mutations' number as n , keeping p children will cost $O(np)$ operations in each iteration.

¹We remove batchnorm layers in the generator. The detailed architecture and more generated images are reported in the Appendix .

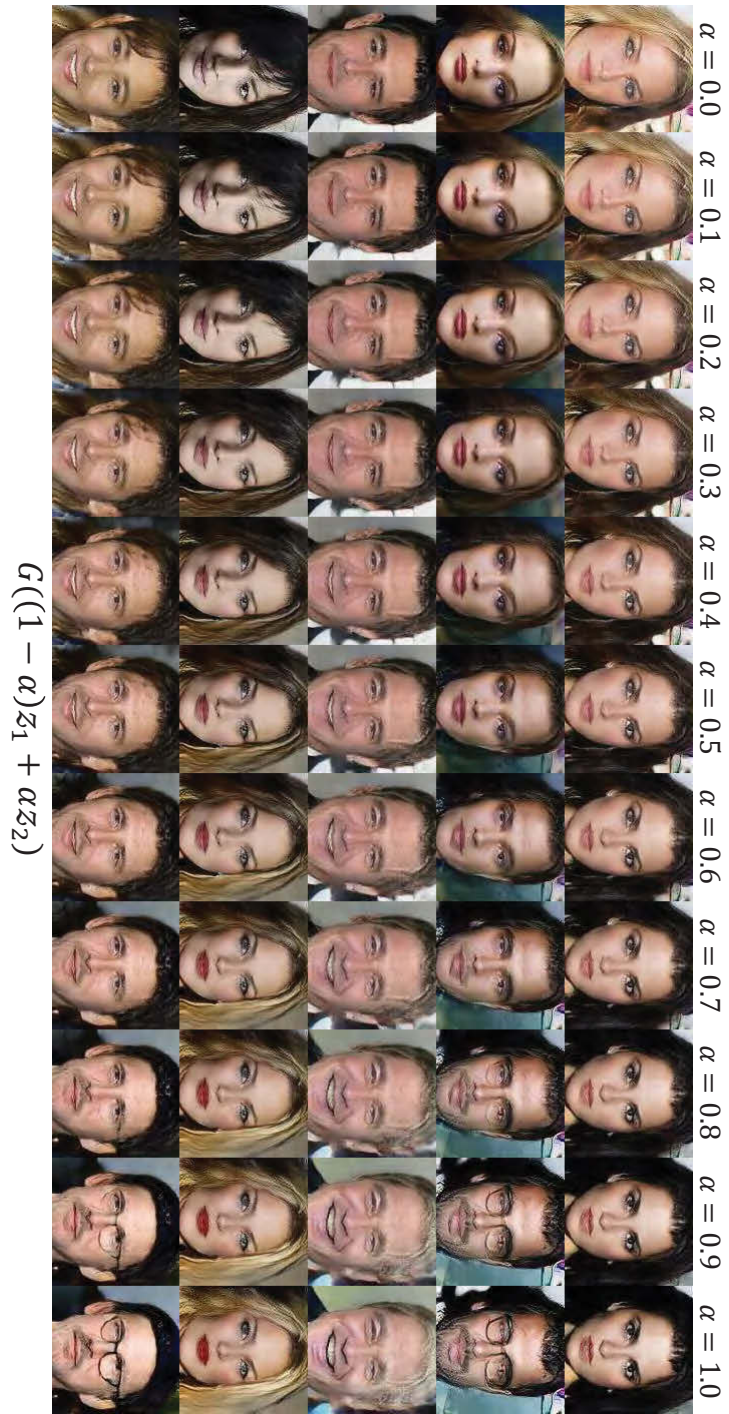


Figure 2.10: Interpolating in latent space. For selected pairs of the generated images from a well-trained E-GAN model, we record their latent vectors z_1 and z_2 . Then, samples between them are generated by linear interpolation between these two vectors.

2.4.5 CelebA and Space Continuity

Since humans excel at identifying facial flaws, generating high-quality human face images is challenging. Similar to generating bedrooms, we employ the same architectures to generate 128×128 RGB human face images (Fig. 2.9). In addition, given a well-trained generator, we evaluate the performance of the embedding in the latent space of noisy vectors z . We aim to test model’s space continuity on CelebA and demonstrate the learned generator does not merely memorize training samples. Sample quality of images can be compared under some metrics, but it is hard to qualitatively and quantitatively evaluate the spatial continuity of generative models. Following DCGAN, there is no comparison on CelebA. In Fig. 5.12, we first select pairs of generated faces and record their corresponding latent vectors z_1 and z_2 . The two images in one pair have different attributes, such as gender, expression, hairstyle, and age. Then, we generate novel samples by linear interpolating between these pairs (*i.e.*, corresponding noisy vectors). We find that these generated samples can seamlessly change between these semantically meaningful face attributes. This experiment demonstrates that generator training does not merely memorize training samples but learns a meaningful projection from latent noisy space to face images. Meanwhile, it also shows that the generator trained by E-GAN does not suffer from model collapse, and shows great space continuity.

2.5 Summary

In this chapter, we present an evolutionary GAN framework (E-GAN) for training deep generative models. To reduce training difficulties and improve generative performance, we devise an evolutionary algorithm to evolve a population of generators to adapt to the dynamic environment (*i.e.*, the discriminator D). In contrast to conventional GANs, the evolutionary paradigm allows the proposed E-GAN to overcome the limitations of individual adversarial objectives and preserve the best offspring after each iteration. Experiments show that E-GAN improves the training stability of GAN models and achieves convincing performance in several image generation tasks. Future works will focus on further exploring the relation-

ship between the environment (*i.e.*, discriminator) and evolutionary population (*i.e.*, generators) and further improving generative performance.

Chapter 3

Learning perceptual information through adversarial paradigm

In this Chapter, we propose Perceptual Adversarial Networks (PAN) for image-to-image transformation tasks. Different from existing application driven algorithms, PAN provides a generic framework of learning to map from input images to desired images (Fig. 3.1), such as a rainy image to its de-rained counterpart, object edges to its photo, semantic labels to a scenes image, etc.. The proposed PAN consists of two feed-forward convolutional neural networks (CNNs): the image transformation network T and the discriminative network D . Besides the generative adversarial loss widely used in GANs, we propose the perceptual adversarial loss, which undergoes an adversarial training process between the image transformation network T and the hidden layers of the discriminative network D . The hidden layers and the output of the discriminative network D are upgraded to constantly and automatically discover the discrepancy between the transformed image and the corresponding ground-truth, while the image transformation network T is trained to minimize the discrepancy explored by the discriminative network D . Through integrating the generative adversarial loss and the perceptual adversarial loss, the networks D and T can be trained alternately to solve image-to-image transformation tasks. Experiments evaluated on several image-to-image transformation tasks (e.g., image de-raining, image inpainting, etc.) demonstrate the effectiveness of the proposed PAN approach and its advantages

over many related state-of-the-art methods.

3.1 Introduction

Image-to-image transformations aim to transform an input image into the desired output image, and they exist in a number of applications about image processing, computer graphics, and computer vision. For example, generating high-quality images from corresponding degraded (e.g. simplified, corrupted or low-resolution) images, and transforming a color input image into its semantic or geometric representations. More examples include, but not limited to, image de-noising [42], image in-painting [11], image super-resolution [112], image colorization [102, 130], image segmentation [72, 94], etc..

In recent years, convolutional neural networks (CNNs) are trained in a supervised manner for various image-to-image transformation tasks [33, 46, 121, 179]. They encode input image into hidden representation, which is then decoded to the output image. By penalizing the discrepancy between the output image and ground-truth image, optimal CNNs can be trained to discover the mapping from the input image to the transformed image of interest. These CNNs are developed with distinct motivations and differ in the loss function design.

One of the most straightforward approaches is to pixel-wisely evaluate output images [23, 33, 137, 183], e.g., least squares loss or least absolute loss to calculate the distance between the output and ground-truth images in the pixel space. Though pixel-wise evaluation can generate reasonable images in many image-to-image transformation tasks, there are some unignorable defects associated with the outputs, such as image blur and image artifacts.

Besides pixel-wise losses, the generative adversarial losses were largely utilized in training image-to-image transformation models. GANs (and cGANs) [54, 109, 150] perform an adversarial training process alternating between identifying and faking, and generative adversarial losses are formulated to evaluate the discrepancy between the generated distribution and the real-world distribution. Experimental results show that generative adversarial losses are beneficial for generating more realistic images. Therefore, there are many GANs (or cGANs) based works to solve image-to-image transformation tasks, resulting in sharper and more real-

istic transformed images [68, 121]. Meanwhile, some GANs variants [73, 172, 187] investigated cross-domain image translations and performed image translation tasks in absence of paired examples [25, 110, 188]. Although these unpaired works achieved reasonable results in some image-to-image translation tasks, they are inappropriate for some image-to-image problems. For example, in image inpainting tasks, it is difficult to define the domain and formulate the distribution of corrupted images, especially when these images are from various classes. In addition, paired information between training data are beneficial for learning image transformations, but they cannot be utilized by unpaired image translation methods. Thus, at this stage, it is still important to study paired training, especially for performance-driven situations and applications, such as high-resolution image synthesis [157], photo-realistic image synthesis [88], real world image inpainting [19], etc..

Moreover, perceptual losses emerged as a novel measurement for evaluating the discrepancy between high-level perceptual features of the output and ground-truth images [17, 34, 70, 140]. Hidden layers of a well-trained image classification network (e.g., VGG-16 [139]) are usually employed to extract high-level features (e.g., content or texture) of both output images and ground-truth images. It is then expected to encourage the output image to have the similar high-level feature with that of the ground-truth image. Recently, perceptual losses were introduced in aforementioned GANs-based image-to-image transformation frameworks for suppressing artifacts [178] and improving perceptual quality [88] of the output images. Though integrating perceptual losses into GANs has produced impressive image-to-image transformation results, existing works are used to depend on external well-trained image classification network (e.g. VGG-Net) out of GANs, but ignored the fact that GANs, especially the discriminative network, also has the capability and demand of perceiving the content of images and the difference between images. Moreover, since these external networks are trained on specific classification datasets (e.g., ImageNet), they mainly focus on features that contribute to the classification and may perform inferior in some image transformation tasks (e.g., transfer aerial image to maps). Meanwhile, since specific hidden layers of pretrained networks are employed, it is difficult to explore the difference between generated images and ground-truth images from more points

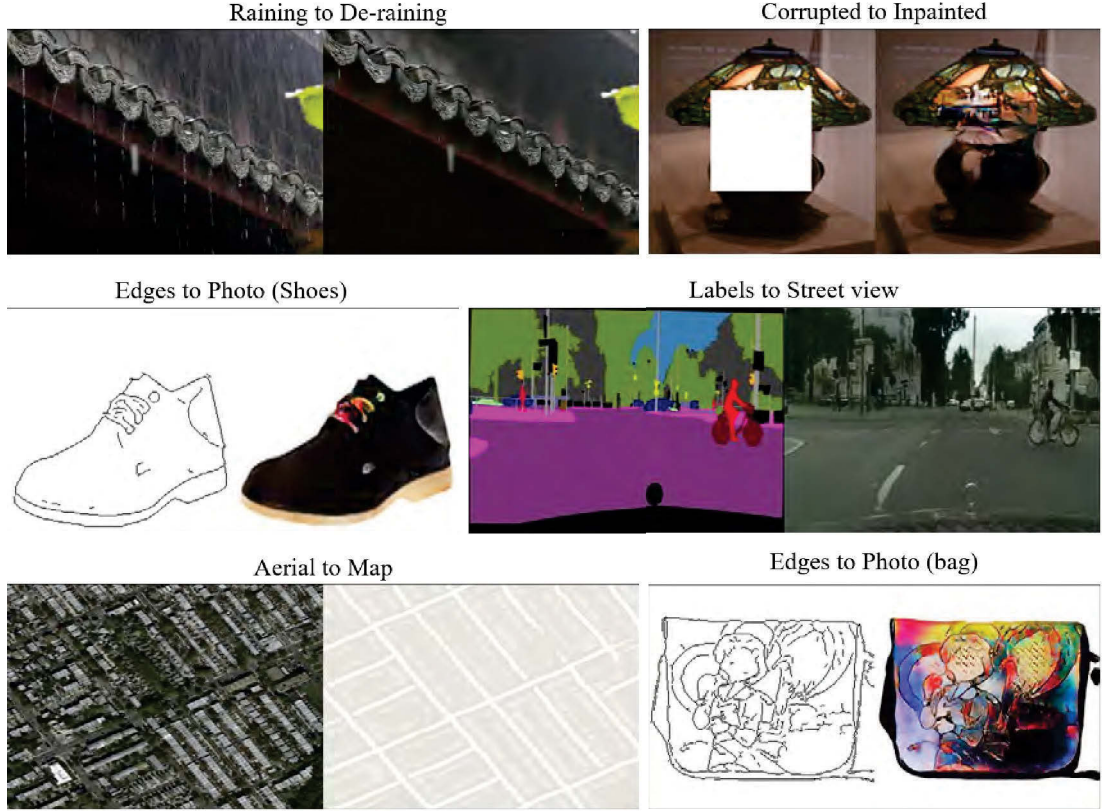


Figure 3.1: Image-to-image transformation tasks. Many tasks in image processing, computer graphics, and computer vision can be regarded as image-to-image transformation tasks, where a model is designed to transform an input image into the required output image. We proposed Perceptual Adversarial Networks (PAN) to solve the image-to-image transformation between paired images. For each pair of the images we demonstrated, the left one is the input image, and the right one is the transformed result of the proposed PAN.

of view.

In this Chapter, we proposed the perceptual adversarial networks (PAN) for image-to-image transformation tasks. Inspired from GANs, PAN is composed of an image transformation network T and a discriminative network D . Both generative adversarial loss and perceptual adversarial loss are employed to train

the PAN. Firstly, similar with GANs, the generative adversarial loss is utilized to measure the distribution of the generated images, i.e., penalizing the generated images to lie in the desired target domain, which usually contributes to produce more visually realistic images. Meanwhile, to comprehensively evaluate transformed images, we devised the perceptual adversarial loss to form dynamic measurements based on the hidden layers of the discriminative network D . Specifically, given hidden layers of the network D , the network T is trained to generate the output image that has the same high-level features with that of the corresponding ground-truth. If the difference between images measured on existing hidden layers of the discriminator is smaller, these hidden layers will be updated to discover discrepancy between images from a new point of view. Different from the pixel-wise loss and conventional perceptual loss, our perceptual adversarial loss undergoes an adversarial training process, and aims to discover and decrease the discrepancy under constantly explored dynamic measurements.

In summary, this chapter makes the following contributions:

- We proposed the perceptual adversarial loss, which utilizes the hidden layers of the discriminative network to evaluate the discrepancy between the output and ground-truth images through an adversarial training process.
- Through combining the perceptual adversarial loss and the generative adversarial loss, we presented the PAN framework for solving image-to-image transformation tasks.
- We evaluated the performance of the PAN on several image-to-image transformation tasks (Fig. 3.1). Experimental results demonstrate that the proposed PAN has a great capability of accomplishing image-to-image transformations.

The rest of this chapter is organized as follows: after a brief summary of previous related works in section 3.2, we illustrate the proposed PAN together with its training losses in section 3.3. Then we exhibit the experimental validation of the whole method in section 3.4. Finally, we conclude this chapter with some future directions in section 3.5.

3.2 Related works

In this section, we first introduce some representative image-to-image transformation methods based on feed-forward CNNs, and then summarize related works on GANs and perceptual losses.

3.2.1 Image-to-image transformation with feed-forward CNNs

Recent years have witnessed a variety of feed-forward CNNs developed for image-to-image transformation tasks. These feed-forward CNNs can be easily trained using the back-propagation algorithm [82, 131], and the transformed images are generated by forwardly passing the input image through the well-trained CNNs in the test stage.

Individual pixel-wise loss or pixel-wise loss accompanied with other losses are employed in a number of image-to-image transformations. Image super-resolution tasks estimate a high-resolution image from its low-resolution counterpart [33, 70, 88]. Image de-raining (or de-snowing) methods attempt to remove the rain (or snow) strikes in the pictures brought by the uncontrollable weather conditions [40, 46, 170, 178]. Given a damaged image, image inpainting aims to recover the missing part of the input image [92, 121, 125, 132]. Image semantic segmentation methods produce dense scene labels based on a single input image [39, 43, 99, 115]. Given an input object image, some feed-forward CNNs were trained to synthesize the image of the same object from a different viewpoint [142, 168]. More image-to-image transformation tasks based on feed-forward CNNs, include, but not limited to, image colorization [23, 65], depth estimations [39, 41], etc..

3.2.2 GANs-based works

Generative adversarial networks (GANs) [54] provide an important approach for learning a generative model which generates samples from the real-world data distribution. GANs consist of a generative network and a discriminative network. Through playing a minimax game between these two networks, GANs are trained to generate more and more realistic samples. Since the great performance on learning real-world distributions, there have emerged a large number of GANs-

based works. Some of these GANs-based works are committed to training a better generative model, such as the InfoGAN [21], the WGAN [2] and the Energy-based GAN [182]. There are also some works integrating the GANs into their models to improve the performance of classical tasks. For example, the PGAN [91] is proposed for small object detection. Specifically, [91] devised a novel perceptual discriminator network, which contains an adversarial branch and a perception branch. The adversarial branch utilizes the adversarial loss to distinguish representations of real and synthesized objects; the perception branch (or loss) employs a classification loss L_{cls} and a bounding-box regression loss L_{loc} to encourage the synthesized ‘super-resolved’ objects representation to retain the same perception information as the input small objects representation. In addition, these kind of works include, but not limited to, the PGN [100] for video prediction, the SRGAN [88] for super-resolution, the ID-CGAN for image de-raining [178], the iGAN [185] for interactive application, the IAN [16] for photo modification, and the Context-Encoder for image in-painting [121]. Most recently, Isola *et al.* [68] proposed the pix2pix-cGANs to perform several image-to-image transformation tasks (also known as image-to-image translations in their work), such as translating semantic labels into the street scene, object edges into pictures, aerial photos into maps, etc..

Moreover, some GANs variants [15, 48, 73, 172, 187] investigated cross-domain image translations through exploring the cyclic mapping (or primal-dual) relation between different image domains. Specifically, a primal GAN aims to explore the mapping relations from source images to target images, while a dual (or inverse) GAN performs the invert task. These two GANs form a closed loop and allow images from either domain to be translated and then reconstructed. Through combining the GAN loss and cycle consistency loss (or recovery loss), these works can be used for performing image translation tasks in absence of paired examples. However, if paired training data are available in some applications, [30, 73, 172, 187] neglecting paired information between data often has inferior performance to that of paired methods [35, 68, 123].

3.2.3 Perceptual loss

Recently, some theoretical analysis and experimental results suggested that the high-level features extracted from a well-trained image classification network have the capability to capture the perceptual information from real-world images [49, 67, 70]. Specifically, representations extracted from hidden layers of well-trained image classification network are beneficial to interpret the semantics of input images, and image style distribution can be captured by the Gram matrix of hidden representations. Hence, high-level features extracted from hidden layers of a well-trained classifier are often introduced in image generation models. Dosovitskiy and Brox [34] took Euclidean distances between high-level features of images as the deep perceptual similarity metrics to improve the performance of image generation. Johnson *et al.* [70], Bruna *et al.* [17] and Ledig *et al.* [88] used features extracted from a well-trained VGG network to improve the performance of single image super-resolution task. In addition, there are works applying high-level features in image style-transfer [49, 70], image de-raining [22, 178] and image view synthesis [89, 119, 124] tasks.

3.3 Method

In this section, we introduce the proposed Perceptual Adversarial Networks (PAN) for image-to-image transformation tasks. Firstly, we explain the generative and perceptual adversarial losses, respectively. Then, we give the whole framework of the proposed PAN. Finally, we illustrate the details of the training procedure and network architectures.

3.3.1 Generative adversarial loss

We begin with the generative adversarial loss in vanilla GANs. A generative network G is trained to map samples from noise distribution p_z to real-world data distribution p_{data} through playing a minimax game with a discriminative network D . In the training procedure, the discriminative network D aims to distinguish the real samples $y \sim p_{\text{data}}$ from the generated samples $G(z)$. In contrary, the generative network G tries to confuse the discriminative network D by generating

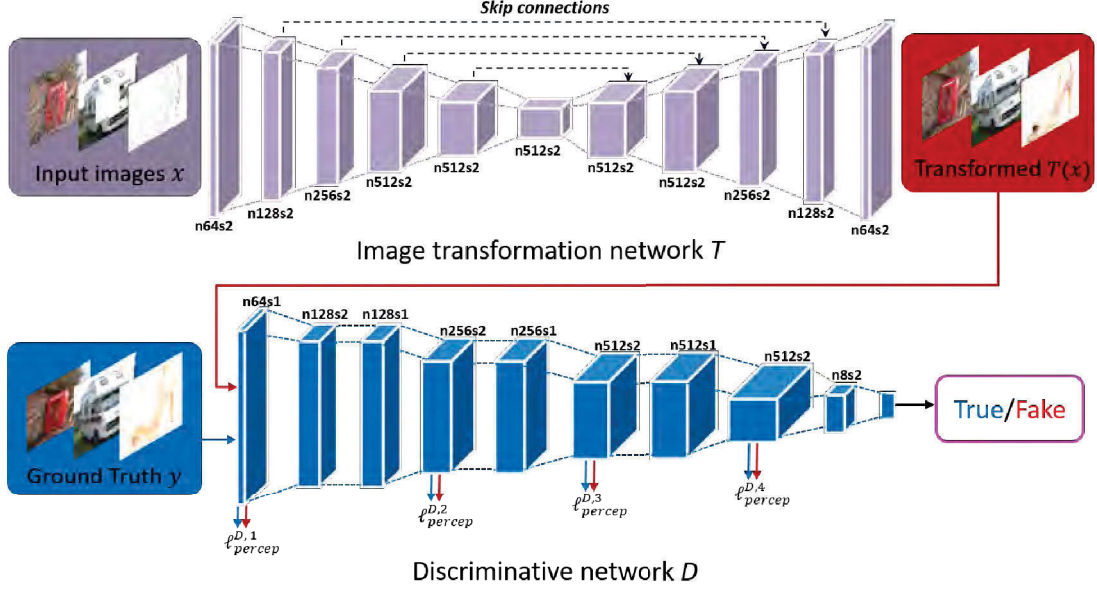


Figure 3.2: PAN framework. PAN consists of an image transformation network T and a discriminative network D . The image transformation network T is trained to synthesize the transformed images given the input images. It is composed of a stack of Convolution-BatchNorm-LeakyReLU encoding layers and Deconvolution-BatchNorm-ReLU decoding layers, and the skip-connections are used between mirrored layers. The discriminative network D is also a CNN that consists of Convolution-BatchNorm-LeakyReLU layers. Hidden layers of the network D are utilized to evaluate the perceptual adversarial loss, and the output of the network D is used to distinguish transformed images from real-world images.

increasingly realistic samples. This minimax game can be formulated as:

$$\min_G \max_D \mathbb{E}_{y \sim p_{\text{data}}} [\log D(y)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3.1)$$

Nowadays, GANs-based models have shown the strong capability of learning generative models, especially for image generation [2, 21, 100]. We, therefore, adopt the GANs learning strategy to solve image-to-image transformation tasks as well. As shown in Fig. 3.2, the image transformation network T is used to

generate transformed image $T(x)$ given the input image $x \in \mathcal{X}$. Meanwhile, each input image x has a corresponding ground-truth image y . We suppose that all target image $y \in \mathcal{Y}$ obey the distribution p_{real} , and the transformed image $T(x)$ is encouraged to have the same distribution with that of targets image y , i.e., $T(x) \sim p_{\text{real}}$. To achieve the generative adversarial learning strategy, a discriminative network D is additionally introduced, and the generative adversarial loss can be written as:

$$\min_T \max_D \mathcal{V}_{D,T} = \mathbb{E}_{y \in \mathcal{Y}} [\log D(y)] + \mathbb{E}_{x \in \mathcal{X}} [\log(1 - D(T(x)))] \quad (3.2)$$

The generative adversarial loss acts as a statistical measurement to penalize the discrepancy between the distributions of transformed images and the ground-truth images.

3.3.2 Perceptual adversarial loss

Different from vanilla GANs that randomly generate samples from the data distribution p_{data} , our goal is to infer the transformed image according to the input images. Therefore, it is a further step of GANs to explore the mapping from the input image to its ground truth.

As mentioned in Sections 3.1 and 3.2, pixel-wise losses and perceptual losses are widely used in existing works for generating images towards the ground truth. The pixel-wise losses penalize the discrepancy occurred in the pixel space, but often produce blurry results [121, 179]. The perceptual losses explore the discrepancy between high-dimensional representations of images extracted from a well-trained classifier, e.g., the VGG net trained on the ImageNet dataset [139]. Although hidden layers of well-trained classifier have been experimentally validated to map the image from pixel space to high-level feature spaces, how to extract the effective features for image-to-image transformation tasks from hidden layers has not been thoroughly discussed.

Here, we employ hidden layers of the discriminative network D to evaluate the perceptual adversarial loss between transformed images and ground-truth images. In our experiments, given the training sample $\{(x_i, y_i) \in (\mathcal{X} \times \mathcal{Y})\}_{i=1}^N$, the least absolute loss is employed to calculate the discrepancy of the high-dimensional



Figure 3.3: Comparison of snow-streak removal using different losses functions. Given the same input image (leftmost), each column shows results trained under different losses. The loss function of ID-CGAN [178] combined the pixel-wise loss (least squares loss), cGANs loss and perceptual loss, i.e., L2+cGAN+perceptual. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.

representations on the hidden layers of the network D , e.g.,

$$\ell_{percep}^{D,j} = \frac{1}{N} \sum_{i=1}^N ||d_j(y_i) - d_j(T(x_i))|| \quad (3.3)$$

where $d_j()$ is the image representation on the j -th hidden layer of the discriminative network D , and $\ell_{percep}^{D,j}$ calculates the discrepancy measured by the j -th hidden layer of D .

Similar to what has been done with the Energy-Based GAN [182], we use two different losses, one (\mathcal{L}_T) to train the image transformation network T , and the other (\mathcal{L}_D) to train hidden layers of the discriminative network D . Therefore, the image transformation network T and hidden layers of the discriminative network D play a non-zero-sum game and form the perceptual adversarial loss. Formally, the perceptual adversarial loss \mathcal{L}_T for the image transformation network T can be written as:

$$\mathcal{L}_T = \sum_{j=1}^F \lambda_j \ell_{percep}^{D,j} \quad (3.4)$$

and, given a positive margin m , the loss \mathcal{L}_D for hidden layers of the discriminative



Figure 3.4: Transforming the semantic labels to cityscapes images use the perceptual adversarial loss. Within the perceptual adversarial loss, a different hidden layer is utilized for each experiment. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images. For higher layers, the transformed images look sharper, but less color information is preserved.

network D is defined as:

$$\mathcal{L}_D = [m - \mathcal{L}_T]^+ = \left[m - \sum_{j=1}^F \lambda_j \ell_{percep}^{D,j} \right]^+ \quad (3.5)$$

where $[\cdot]^+ = \max(0, \cdot)$, $\{\lambda_j\}_{j=1}^F$ are hyper-parameters balancing the influence of F different hidden layers.

By minimizing the perceptual adversarial loss function \mathcal{L}_T with respect to parameters of T , we encourage the network T to generate image $T(x)$ that has similar high-level features with its ground-truth y on the hidden layers. If the weighted sum of discrepancy between transformed images and ground-truth images on different hidden layers is less than the positive margin m , the loss function \mathcal{L}_D will upgrade the discriminative network D for some new latent feature spaces, which preserve the discrepancy between the transformed images and their ground-truth. Therefore, based on the perceptual adversarial loss, the discrepancy be-

tween the transformed and ground-truth images can be constantly explored and exploited.

Compared to our perceptual adversarial loss which measures the difference between the transformed image and ground-truth image in hidden layers of the discriminator, the conditional GAN loss indicates whether the transformed image forms the appropriate image pair with the input image, and can also explore supervised information of paired images during the training process. However, they utilize different methods to minimize high-level feature differences explored by the discriminator. The perceptual adversarial loss directly penalizes the high-level representations of transformed images and ground-truth images to be as same as possible. In contrast, the conditional GAN loss aims to model the mapping relation from the input x to its output y_{real} and encourages the generated image pairs $(x, T(x))$ obeying the same conditional distribution $P_{\text{real}}(y|x)$. Compared to conditional GAN loss that indirectly guides the generated images $T(x)$ sharing the same features with corresponding ground-truth y_{real} , our perceptual adversarial loss directly measures and minimizes differences between the generated images and ground-truth images from different perspectives.

3.3.3 The perceptual adversarial networks

Based on the aforementioned generative adversarial loss (Eq. 3.2) and perceptual adversarial loss (Eq. 3.4 and Eq. 3.5), we develop the PAN framework, which consists of an image transformation network T and a discriminative network D . These two networks are trained alternately to perform an adversarial learning process, the loss functions of image transformation network \mathcal{J}_T and discriminative network \mathcal{J}_D are formally defined as:

$$\begin{aligned}\mathcal{J}_T &= \theta \mathcal{V}_{D,T} + \mathcal{L}_T \\ \mathcal{J}_D &= -\theta \mathcal{V}_{D,T} + \mathcal{L}_D \\ &= -\theta \mathcal{V}_{D,T} + [m - \mathcal{L}_T]^+\end{aligned}\tag{3.6}$$

where θ is the hyper-parameter balance the influence of generative adversarial loss and perceptual adversarial loss. When $\mathcal{L}_T < m$, minimizing \mathcal{J}_D with respect to the parameters of D is consistent with maximizing \mathcal{J}_T . Otherwise, when $\mathcal{L}_T \geq m$,

the second term of \mathcal{J}_D will have zero gradients, because of the positive margin m . In general, the discriminative network D aims to distinguish transformed image $T(x)$ from ground-truth image y from both the statical (the first term of \mathcal{J}_D) and dynamic perceptual (the second term of \mathcal{J}_D) aspects. On the other hand, the image transformation network T is trained to generate increasingly better images by reducing the discrepancy between the output and ground-truth images.

3.3.4 Network architectures

Fig. 3.2 illustrates the framework of the proposed PAN, which is composed of two CNNs, i.e., the image transformation network T and the discriminative network D .

3.3.4.1 Image transformation network T

The image transformation network T is designed to generate the transformed image given the input image. Following the network architectures in [68, 127], the network T firstly encodes the input image into high-dimensional representation using a stack of Convolution-BatchNorm-LeakyReLU layers, and then, the output image can be decoded by the following Deconvolution-BatchNorm-ReLU layers. Note that the output layer of the network T does not use batchnorm and replaces the ReLU with Tanh activation. Moreover, the skip-connections are used to connect mirrored layers in the encoder and decoder stacks. More details of the transformation network T are listed in Table 3.2. The same architecture of the network T is used for all experiments in this chapter, except there is an additional explanation¹.

3.3.4.2 Discriminative network D

In the proposed PAN framework, the discriminative network D is introduced to compute the discrepancy between the transformed images and the ground-truth images. Specifically, given an input image, the discriminative network D extracts high-level features using a series of Convolution-BatchNorm-LeakyReLU layers.

¹In analysis of the loss functions and the image inpainting task, different architectures of the network T were used.

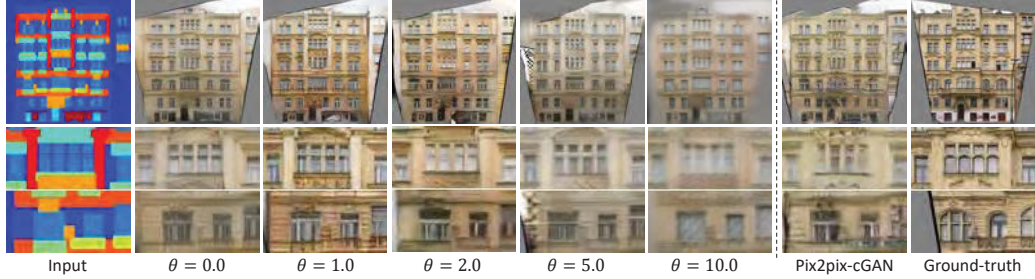


Figure 3.5: Comparison of transforming the semantic labels to facades images by controlling the hyper-parameter θ . Given the same input image (leftmost), each column shows results trained under different θ . For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.

The 1st, 4th, 6th, and 8th layers are utilized to measure the perceptual adversarial loss for every pair of transformed image and its corresponding ground-truth in the training data. Finally, the last convolution layer is flattened and then fed into a single sigmoid output. The output of the discriminative network D estimates the probability that the input image comes from the real-world dataset rather than from the image transformation network T . The same discriminative network D is applied for all tasks demonstrated in this chapter, and details of the network D are shown in Table 3.1.

3.4 Experiments

In this section, we evaluate the performance of the proposed PAN on several image-to-image transformation tasks, which are popular in fields of image processing (e.g., image de-raining), computer vision (e.g., semantic segmentation) and computer graphics (e.g., image generation).

3.4.1 Experimental setting up

For fair comparisons, we adopted the same settings with existing works, and reported experimental results using several evaluation metrics. These tasks and

Table 3.1: The architecture of the discriminative network.

Discriminative network D	
Input: Image	
[layer 1]	Conv. (3, 3, 64), stride=1; $LReLU$;
	(Perceptual adversarial loss: $\ell_{percep}^{D,1}$)
[layer 2]	Conv. (3, 3, 128), stride=2; Batchnorm; $LReLU$;
[layer 3]	Conv. (3, 3, 128), stride=1; Batchnorm; $LReLU$;
[layer 4]	Conv. (3, 3, 256), stride=2; Batchnorm; $LReLU$;
	(Perceptual adversarial loss: $\ell_{percep}^{D,2}$)
[layer 5]	Conv. (3, 3, 256), stride=1; Batchnorm; $LReLU$;
[layer 6]	Conv. (3, 3, 512), stride=2; Batchnorm; $LReLU$;
	(Perceptual adversarial loss: $\ell_{percep}^{D,3}$)
[layer 7]	Conv. (3, 3, 512), stride=1; Batchnorm; $LReLU$;
[layer 8]	Conv. (3, 3, 512), stride=2; Batchnorm; $LReLU$;
	(Perceptual adversarial loss: $\ell_{percep}^{D,4}$)
[layer 9]	Conv. (3, 3, 8), stride=2; $LReLU$;
[layer 10]	Fully connected (1); $Sigmoid$;
Output: Real or Fake (Probability)	

data settings include:

- *Single image de-raining*, on the dataset provided by ID-CGAN [178].
- *Image Inpainting*, on a subset of ILSVRC'12 (same as context-encoder [121]).
- *Semantic labels \leftrightarrow images*, on the Cityscapes dataset [26] (same as pix2pix [68]).
- *Edges \rightarrow images*, on the dataset created by pix2pix [68]. The original data is from [185] and [174], and the HED edge detector [164] was used to extract edges.

Table 3.2: The architecture of the image transformation network.

Image transformation network T	
Input: Image	
[layer 1]	Conv. (3, 3, 64), stride=2; <i>LReLU</i> ;
[layer 2]	Conv. (3, 3, 128), stride=2; Batchnorm;
[layer 3]	<i>LReLU</i> ; Conv. (3, 3, 256), stride=2; Batchnorm;
[layer 4]	<i>LReLU</i> ; Conv. (3, 3, 512), stride=2; Batchnorm;
[layer 5]	<i>LReLU</i> ; Conv. (3, 3, 512), stride=2; Batchnorm;
[layer 6]	<i>LReLU</i> ; Conv. (3, 3, 512), stride=2; Batchnorm; <i>LReLU</i> ;
[layer 7]	DeConv. (4, 4, 512), stride=2; Batchnorm;
	Concatenate Layer(Layer 7, Layer 5); <i>ReLU</i> ;
[layer 8]	DeConv. (4, 4, 256), stride=2; Batchnorm;
	Concatenate Layer(Layer 8, Layer 4); <i>ReLU</i> ;
[layer 9]	DeConv. (4, 4, 128), stride=2; Batchnorm;
	Concatenate Layer(Layer 9, Layer 3); <i>ReLU</i> ;
[layer 10]	DeConv. (4, 4, 64), stride=2; Batchnorm;
	Concatenate Layer(Layer 10, Layer 2); <i>ReLU</i> ;
[layer 11]	DeConv. (4, 4, 64), stride=2; Batchnorm; <i>ReLU</i> ;
[layer 12]	DeConv. (4, 4, 3), stride=2; <i>Tanh</i> ;
Output: Transformed image	

- *Aerial*→*map*, on the dataset from pix2pix [68].

Furthermore, all experiments were trained on Nvidia Titan-X GPUs using Theano [10]. Given the generative and perceptual adversarial losses, we alternately updated the image transformation network T and the discriminative network D . Specifically, Adam solver [75] with a learning rate of 0.0002 and a first momentum of 0.5 was used in network training. After one update of the discriminative network D , the image transformation T will be updated three times.

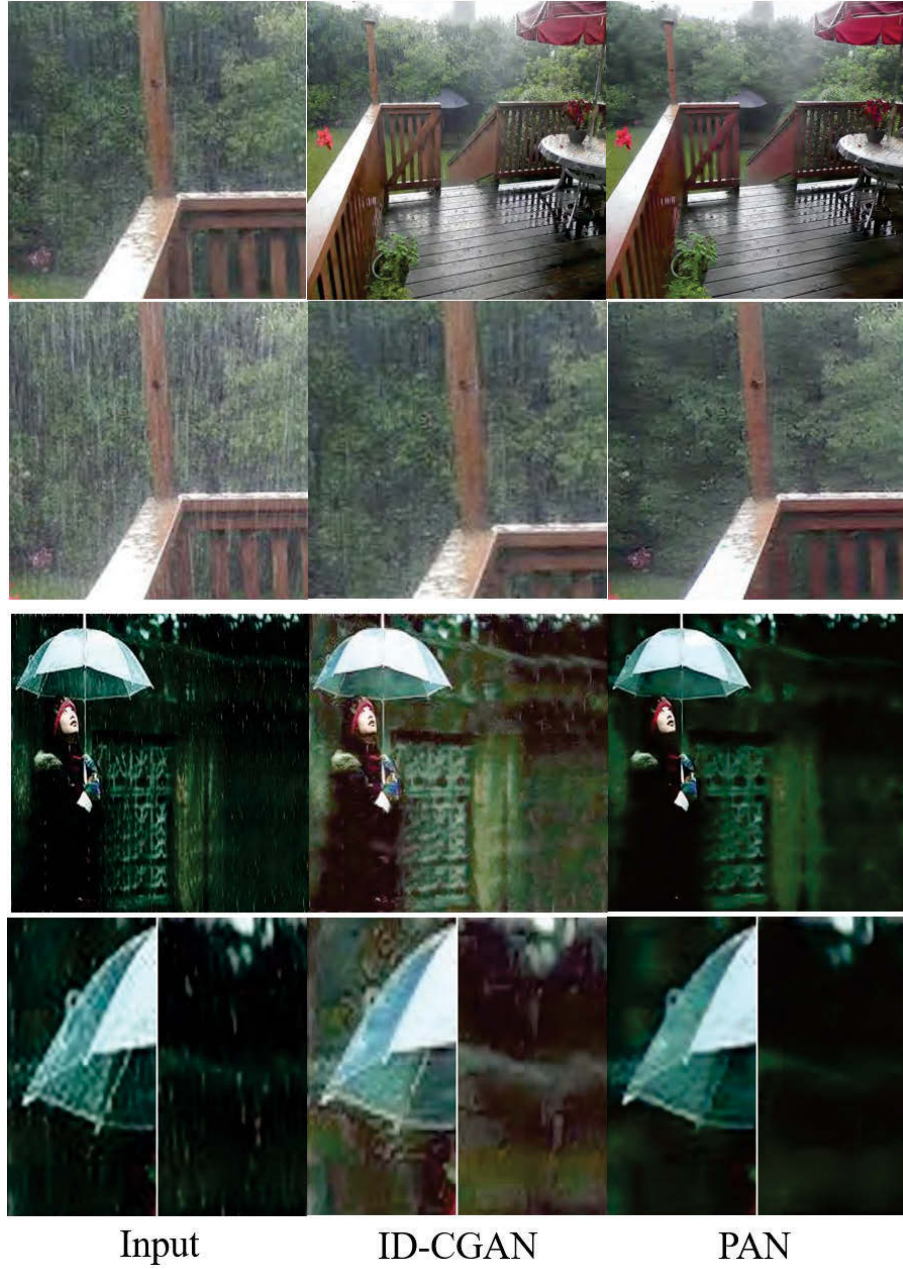


Figure 3.6: Comparison of rain-streak removal using the ID-CGAN with the proposed PAN on real-world rainy images. For better visual comparison, zoomed versions of the specific regions-of-interest are demonstrated below the test images.

Hyper-parameters $\theta = 1$, $\lambda_1 = 5$, $\lambda_2 = 1.5$, $\lambda_3 = 1.5$, $\lambda_4 = 1$, and batch size of 4 were used for all tasks. Since the dataset sizes for different tasks are changed largely, the training epochs of different tasks were set accordingly. Overall, the number of training iterations was around 100k.

3.4.2 Evaluation metrics

To illustrate the performance of image-to-image transformation tasks, we conducted qualitative and quantitative experiments to evaluate the performance of the transformed images. For the qualitative experiments, we directly presented the input and transformed images. Meanwhile, we used quantitative measures to evaluate the performance over the test sets, such as Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) [160], Universal Quality Index (UQI) [159] and Visual Information Fidelity (VIF) [136].

3.4.3 Analysis of the loss functions

As discussed in Sections 3.1 and 3.2, the design of loss function will largely influence the performance of image-to-image transformation. Firstly, the pixel-wise loss (using least squares loss) is widely used in various image-to-image transformation works [32, 76, 142]. Then, the joint loss integrating pixel-wise loss and conditional generative adversarial loss is proposed to synthesize more realistic transformed images [68, 121]. Most recently, through introducing the perceptual loss, i.e., penalizing the discrepancy between high-level features that extracted by a well-trained classifier, the performance of some image-to-image transformation tasks are further enhanced [70, 88, 178]. Different from these existing methods, the proposed PAN loss integrates the generative adversarial loss and the perceptual adversarial loss to train image-to-image transformation networks. Here, we compare the performance of the proposed perceptual adversarial loss with those of existing losses. For a fair comparison, we adopted the same image transformation network and data settings from ID-CGAN [178], and used the combination of different losses to perform the image de-raining (de-snowing) task. The quantitative results over the synthetic test set were shown in Table 3.3, while the qualitative results on the real-world images were shown in Fig. 3.3. From both

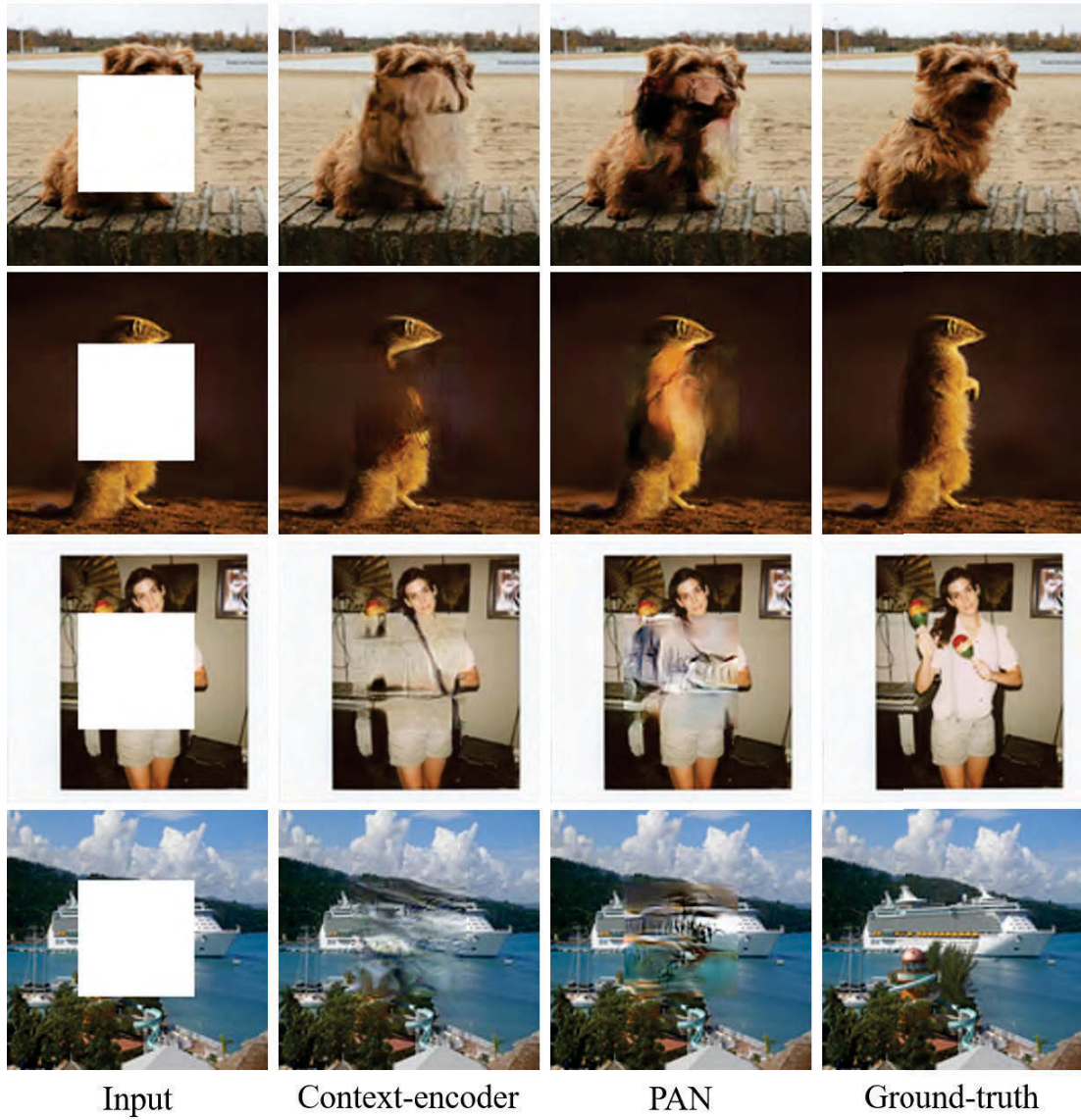


Figure 3.7: Comparison of image in-painting results using the Context-Encoder(CE) with the proposed PAN. Given the central region missed input image (leftmost), the in-painted images and the ground-truth are listed on its rightside.

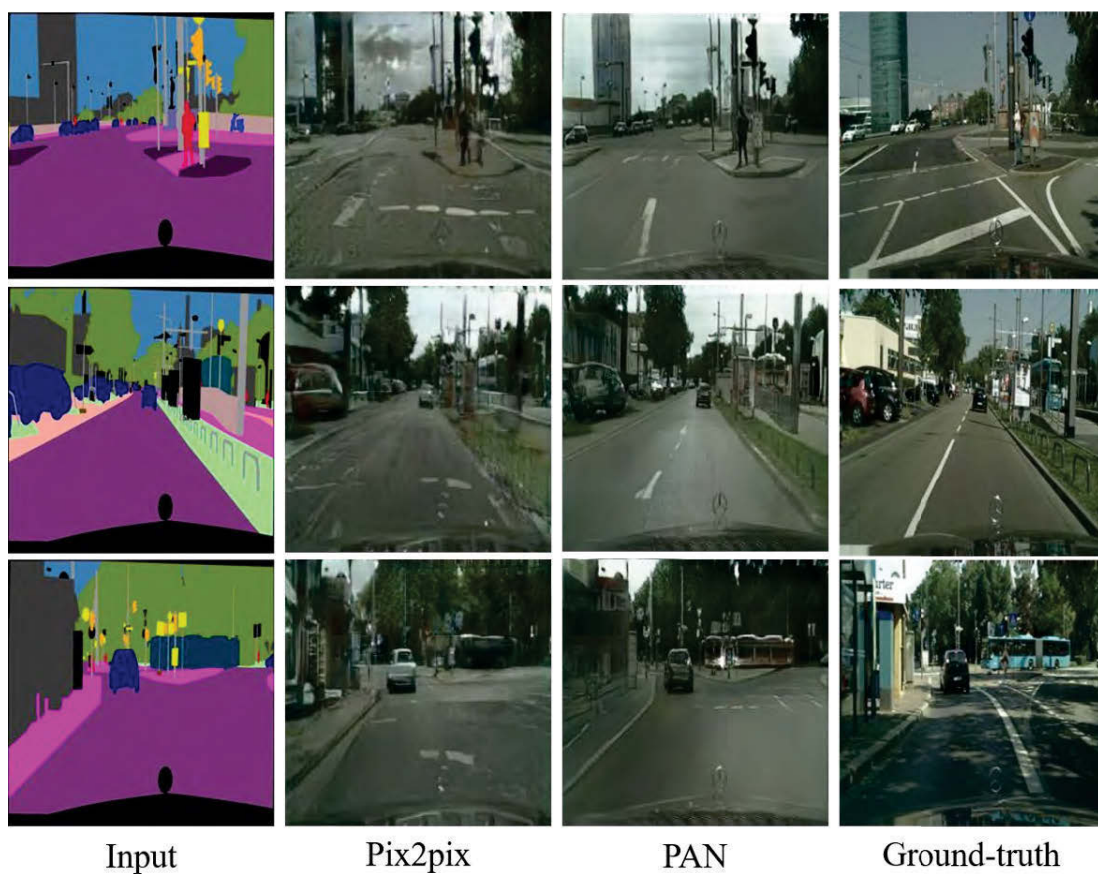


Figure 3.8: Comparison of transforming the semantic labels to cityscapes images using the pix2pix-cGAN with the proposed PAN. Given the semantic labels (left-most), the transformed cityscapes images and the ground-truth are listed on the rightside.

quantitative and qualitative comparisons, we find that only using the pixel-wise loss (least squares loss) achieved the worst result, and there are many snow-streaks in the transformed images (Fig. 3.3). Through introducing the cGANs loss, the de-snowing performance was indeed improved, but artifacts can be observed (Fig. 3.3) and the PSNR performance dropped (Table 3.3). Combining the pixel-wise, cGAN and perceptual loss (VGG-16 [139]) together, i.e., using the loss function of ID-CGAN [178], the quality of transformed images has been further improved on both observations and quantitative measurements. However, from Fig. 3.3, we observe that the transformed images have some color distortion compared to the input images. The proposed PAN loss (i.e., combining the perceptual adversarial loss and original GAN loss) not only removed most streaks without colour distortion, but also achieved much better performance on quantitative measurements. Moreover, we evaluated the performance of combining conditional GAN loss and the perceptual adversarial loss. Comparing with using the cGAN loss independently, introducing the perceptual adversarial loss largely improves the model performance. Yet, comparing with the PAN loss, replacing the original GAN loss with its conditional version does not make a further improvement in both quantitative and qualitative comparisons.

Though variables of the discriminator network are optimized in iterations, the capability of hidden layers is constrained by network architecture. Therefore, in the proposed PAN, we selected four hidden layers of the discriminative network D to calculate the perceptual adversarial loss. We next proceed to analyze the property of these hidden layers. Specifically, we trained four configurations of the PAN to perform the task of transforming the semantic labels to the cityscapes images. For each configuration, we set one hyper-parameter λ_i as 1, and the others $\{\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots\}$ as 0, i.e., we used only one hidden layer to evaluate the perceptual adversarial loss in each configuration. As shown in Fig. 3.4, the lower layers (e.g., $\ell_{percep}^{D,1}$, $\ell_{percep}^{D,2}$) pay more attention to the patch-to-patch transformation and the color transformation, but the transformed images are blurry and lack of fine details. On the other hand, higher layers (e.g., $\ell_{percep}^{D,1}$, $\ell_{percep}^{D,2}$) capture more high-frequency information, but lose the color information. Therefore, by integrating different properties of these hidden layers, the proposed PAN can be expected to achieve better performance, and the final results of this task

Table 3.3: De-raining

	PSNR(dB)	SSIM	UQI	VIF
L2	22.77	0.7959	0.6261	0.3570
cGAN	21.87	0.7306	0.5810	0.3173
L2+cGAN	22.19	0.8083	0.6278	0.3640
ID-CGAN	22.91	0.8198	0.6473	0.3885
PAN	23.35	0.8303	0.6644	0.4050
PA Loss+cGAN	23.22	0.8078	0.6375	0.3904

are shown in Fig. 3.8 and Table 3.5.

In our work, the balance between the perceptual adversarial loss and GAN loss is controlled by the hyper-parameters θ . In the task of transforming labels to facades, we vary the value of θ to test its influence on the proposed PAN. Qualitative samples are reported in Fig 3.5. As shown in Fig. 3.5, only using the perceptual adversarial loss (i.e., $\theta = 0$) has already had the capability of synthesizing visually reasonable images from the input labels. Given the advantage of the GAN loss to promote more realistic images, the transformation performance gets better with increasing θ . However, with the continuous increasing of θ , the role of perceptual adversarial loss will be weakened and the model performance drops, e.g., visual artifacts are observed in certain images.

3.4.4 Comparing with existing works

In this subsection, we compared the performance of the proposed PAN with those of existing algorithms for image-to-image transformation tasks.

3.4.4.1 Context-encoder

Context-Encoder (CE) [121] trained CNNs for the single image inpainting task. Given corrupted images as input, image inpainting can be formulated as an image-to-image transformation task. Pixel-wise loss (least squares loss) and the generative adversarial loss were combined in the Context-Encoder to explore the

Table 3.4: In-painting

	PSNR(dB)	SSIM	UQI	VIF
Context-Encoder	21.74	0.8242	0.7828	0.5818
PAN	21.85	0.8307	0.7956	0.6104

relationship between the input surroundings and its central missed region.

To compare with the Context-Encoder, we applied PAN to inpaint images whose central regions were missed. As illustrated in Section 3.4.1, 100k images were randomly selected from the ILSVRC’12 dataset to train both Context-Encoder and PAN, and 50k images from the ILSVRC’12 validation set were used for test purpose. Moreover, since the image inpainting models are asked to generate the missing region of the input image instead of the whole image, we directly employ the architecture of the image transformation network from [121].

In Fig. 3.7, we reported some example results in the test set. For each input image, the missing part is mixed by the foreground objects and backgrounds. The goal is to predict and recover missing parts with the help of surrounding information. From the inpainted results, we find the proposed PAN performed better on understanding the surroundings and estimating the missing part with semantic contents. However, the context-encoder tended to use the nearest region (usually the background) to inpaint the missing part. PAN can synthesize more details in the missing parts. Last but not the least, in Table 3.4, we reported the quantitative results calculated over all 50k test images, which also demonstrated that the proposed PAN achieves better performance on understanding the context of the input images and synthesizing the corresponding missing parts.

3.4.4.2 ID-CGAN

Image de-raining task aims to remove rain streaks in a given rainy image. Considering the unpredictable weather conditions, the single image de-raining (de-snowing) is a challenge image-to-image transformation task. Most recently, the Image De-raining Conditional Generative Adversarial Networks (ID-CGAN) was proposed to tackle the image de-raining problem. Through combining the pixel-

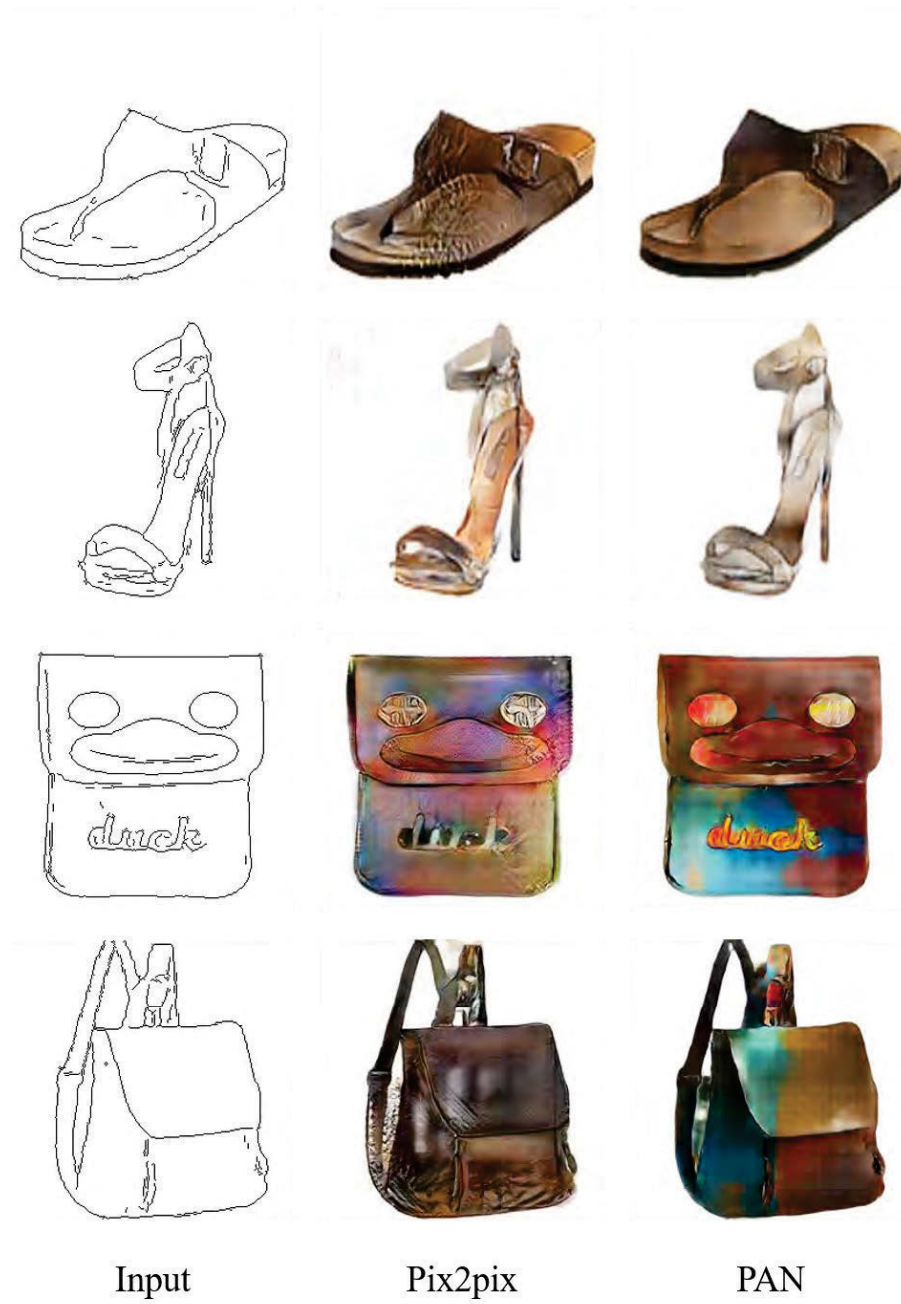


Figure 3.9: Comparison of transforming the object edges to corresponding images using the pix2pix-cGAN with the proposed PAN. Given the edges (leftmost), the generated images of shoes and handbags are listed on the rightside.

wise (least squares loss), conditional generative adversarial, and perceptual losses (VGG-16), ID-CGAN achieved the state-of-the-art performance on single image de-raining.

We attempted to solve image de-raining by the proposed PAN using the same setting with that of ID-CGAN. Since there is a lack of large-scale datasets consisting of paired rainy and de-rained images, we resort to synthesize the training set [178] of 700 images. Zhang *et al.* [178] provided 100 synthetic images and 50 real-world rainy images for test. Since the ground-truth is available for synthetic test images, we calculated and reported the quantitative results in Table 3.3. Moreover, we test both ID-CGAN and PAN on real-world rainy images, and the results were shown in Fig. 3.6. For better visual comparison, we zoomed up the specific regions-of-interest below the test images.

From Fig. 3.6, we found both ID-CGAN and PAN achieved great performance on single image de-raining. However, by observing the zoomed region, the PAN removed more rain-strikes with less color distortion. Additionally, as shown in Table 3.3, for synthetic test images, the de-rained results of PAN are much more similar with the corresponding ground-truth than that of ID-CGAN. Dealing with the uncontrollable weather condition, why the proposed PAN can achieve better results? One possible reason is that ID-CGAN utilized the well-trained classifier to extract the high-level features of the output and ground-truth images, and penalize the discrepancy between them (i.e., the perceptual loss). The high-level features extracted by the well-trained classifier usually focus on the content information, and may be hard to capture other image information, such as color information. Yet, the proposed PAN used the perceptual adversarial loss, which aims to continually and automatically measure the discrepancy between the output and ground-truth images. The different training strategy of PAN may help the model to learn a better mapping from the input to output images, and resulting in better performance.

3.4.4.3 Pix2pix-cGAN

Isola *et al.* [68] utilized cGANs as a general-purpose solution to image-to-image translation (transformation) tasks. In their work, the pixel-wise loss (least ab-

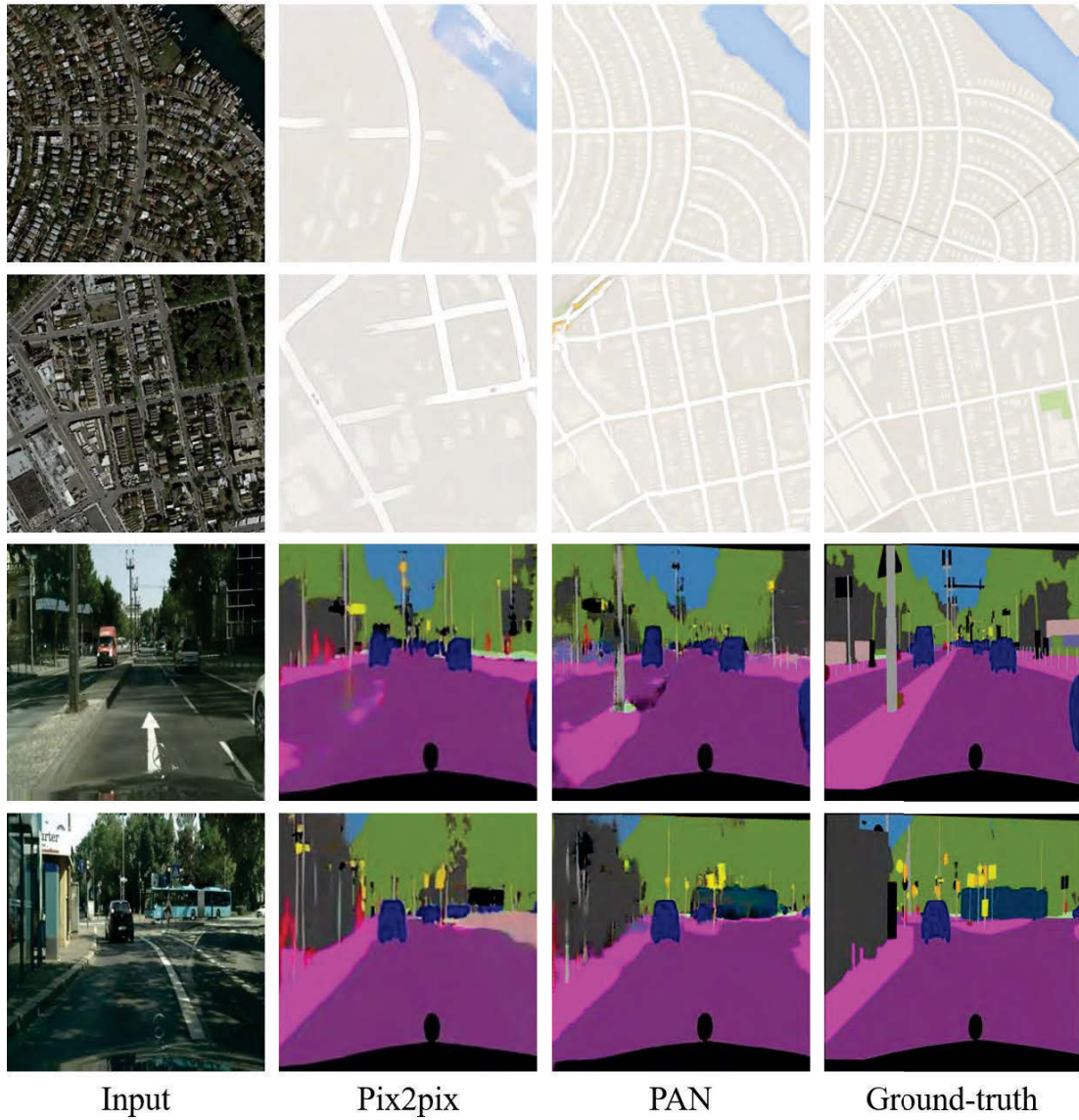


Figure 3.10: Comparison of some other tasks using the pix2pix-cGAN with the proposed PAN. In the first row, semantic labels are generated based on the real-world cityscapes images. And, the second row reports the generated maps given the aerial photos as input.

Table 3.5: Comparison with pix2pix-CGAN

Semantic labels \rightarrow Cityscapes images				
	PSNR(dB)	SSIM	UQI	VIF
pix2pix-cGAN	15.74	0.4275	0.07315	0.05208
PAN	16.06	0.4820	0.1116	0.06581
Edges \rightarrow Shoes				
	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	20.07	0.7504	0.2724	0.2268
PAN	19.51	0.7816	0.3442	0.2393
Edges \rightarrow Handbags				
	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	16.50	0.6307	0.3978	0.1723
PAN	15.90	0.6570	0.4042	0.1841
Cityscapes images \rightarrow Semantic labels				
	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	19.46	0.7270	0.1555	0.1180
PAN	20.67	0.7725	0.1732	0.1638
Aerial photos \rightarrow Maps				
	PSNR(dB)	SSIM	UQI	VIF
ID-cGAN	26.10	0.6465	0.09125	0.02913
PAN	28.32	0.7520	0.3372	0.1617

solute loss) and Patch-cGANs loss are employed to solve a series of image-to-image transformation tasks, such as translating the object edges to its photos, semantic labels to scene images, gray images to color images, etc.. The image-to-image transformation tasks performed by pix2pix-cGAN can also be solved by the proposed PAN. Here, we implemented some of them and compared with pix2pix-cGAN.

Firstly, we attempted to translate the semantic labels to cityscapes images.

Unlike the image segmentation problems, this inverse translation is an ill-posed problem and image transformation network has to learn prior knowledge from the training data. As shown in Fig. 3.8, given semantic labels as input images, we listed the transformed cityscapes images of pix2pix-cGAN, PAN and the corresponding ground-truth on the rightside. From the comparison, we found the proposed PAN captured more details with less deformation, which made the synthetic images looked more realistic. Moreover, the quantitative comparison in Table 3.5 also indicated that the PAN can achieve much better performance.

Generating real-world objects from corresponding edges is also one kind of image-to-image transformation task. Based on the dataset provided by [68], we trained the PAN to translate edges to object photos, and compared its performance with that of pix2pix-cGAN. Given edges as input, Fig. 3.9 presented shoes and handbags synthesized by pix2pix-cGAN and PAN. At the same time, the quantitative results over the test set were shown in the Table 3.5. Observing the generated object photos, we think that both pix2pix-cGAN and PAN achieved promising performance, yet it’s hard to tell which one is better. Quantitative results are also very close, PAN performed slightly inferior to pix2pix-cGAN on the PSNR measurement, yet superior on other quantitative measurements.

In addition, we compared PAN with pix2pix-cGAN on tasks of generating semantic labels from cityscapes photos, and generating maps from the aerial photos. Some example images generated using PAN and pix2pix-cGAN and their corresponding quantitative results were shown in Fig. 3.10 and Table 3.5, respectively. To perform these two tasks, the image-to-image transformation models are asked to capture the semantic information from the input image, and synthesize the corresponding transformed images. Since pix2pix-cGAN employed the pixel-wise and generative adversarial losses to training their model, it may be hard to capture perceptual information from the input image, which causes that their results to be poor, especially on transforming the aerial photos to maps. However, we can observe that the proposed PAN can still achieve promising performance on these tasks. These experiments showed that the proposed PAN can also effectively extract the perceptual information from the input image.

3.5 Summary

In this chapter, we proposed the perceptual adversarial networks (PAN) for image-to-image transformation tasks. As a generic framework of learning mapping relationship between paired images, the PAN combines the generative adversarial loss and the proposed perceptual adversarial loss as a novel training loss function. According to this loss function, a discriminative network D is trained to continually and automatically explore the discrepancy between the transformed images and the corresponding ground-truth images. Simultaneously, an image transformation network T is trained to narrow the discrepancy explored by the discriminative network D . Through the adversarial training process, these two networks are updated alternately. Finally, experimental results on several image-to-image transformation tasks demonstrated that the proposed PAN framework is effective and promising for practical image-to-image transformation applications.

Chapter 4

Interpretable and disentangled representations in adversarial learning

In this chapter, we propose a principled Tag Disentangled Generative Adversarial Networks (TD-GAN) for re-rendering new images for the object of interest from a single image of it by specifying multiple scene properties (such as viewpoint, illumination, expression, etc.). The whole framework consists of a disentangling network, a generative network, a tag mapping net, and a discriminative network, which are trained jointly based on a given set of images that are completely/partially tagged (i.e., supervised/semi-supervised setting). Given an input image, the disentangling network extracts disentangled and interpretable representations, which are then used to generate images by the generative network. In order to boost the quality of disentangled representations, the tag mapping net is integrated to explore the consistency between the image and its tags. Furthermore, the discriminative network is introduced to implement the adversarial training strategy for generating more realistic images. Experiments on two challenging datasets demonstrate the state-of-the-art performance of the proposed framework in the problem of interest.

4.1 Introduction

The re-rendering of new images for the object of interest from a single image of it by specifying expected scene properties (such as viewpoint, illumination, expression, etc.) is of fundamental interest in computer vision and graphics. For example, re-rendering of faces for the continuous pose, illumination directions, and various expressions would be an essential component for virtual reality systems, where one tends to naturally “paste” persons into a virtual environment [59, 97, 103, 122]. More applications of image re-rendering can be found in architecture, simulators, video games, movies, visual effects, etc.

Conventional approaches to addressing the object image re-rendering problem are generally based on the following scheme: a 3D (static or dynamic) model of the object of interest is first reconstructed from the given image(s) and is then projected onto the 2D image plane corresponding to a specified configuration of scene properties. However, 3D model reconstruction from a single 2D object image is a highly ill-posed and challenging problem. Taking the 3D surface reconstruction of the static object for an example, it relies on the exploitation of class-specific statistical priors on the 3D representation of the object of interest [71, 145, 151], while the modeling of such 3D priors necessitates very high expenses in building the training data. Also considering the current state-of-the-art in the study of 3D reconstruction, it is desirable to directly integrate the 3D model reconstruction and 3D-2D projection process together so as to be able to focus on the 2D data only.

Recent works (e.g., [24, 83]) have shown the promise of deep learning models in achieving the goal of interest. The basic idea is based on learning interpretable and disentangled representations [8] of images. Unlike most deep learning models which focus on the learning of hierarchical representations [9], these models aim to extract disentangled representations which correspond to different factors (e.g., identity, viewpoint, etc.) from the input image [181]. Learning the disentangled representations aims to express the objective factors with different high-level representations. Using the human brain to make an analogy, the human understands the world by projecting real-world objects into abstract concepts of different factors and can generate new object images with these concepts via generalization.

Despite the great progress achieved in image re-rendering, existing methods share some important limitations. The first one is regarding the effectiveness and independence of the (disentangled) representations extracted from the input image. For most existing methods, (disentangled) representations are mostly extracted from images themselves, and the valuable tag information (e.g., photograph conditions and object characterizations) associated with images has not been finely explored. Besides, there was little attempt to make the re-rendering result more realistic by increasing the difficulty in distinguishing genuine and re-rendered images. Last but not least, previous works mostly focused on performing object image re-rendering w.r.t. a single scene property and the extension of the developed methods to the setting of multiple scene properties is not straightforward.

In order to boost the performance in re-rendering new images for the object of interest from a single image of it by specifying multiple scene properties, in this chapter, we propose a principled Tag Disentangled Generative Adversarial Networks (*TD-GAN*). The whole framework consists of four parts: a disentangling network, a generative network, a tag mapping net, and a discriminative network, which are trained jointly based on a given set of images that are completely/partially tagged (corresponding to the supervised/semi-supervised setting). Given an input image, the disentangling network extracts disentangled and interpretable representations, which are then used to generate images by the generative network. In order to boost the quality of the obtained disentangled representations, the tag mapping net is integrated to explore the consistency between the image and its tags. Considering that the image and its tags record the same object from two different perspectives [166], they should share the same disentangled representations. Furthermore, the discriminative network is introduced to implement the adversarial training strategy for generating more realistic images. Experiments on two challenging datasets demonstrate the state-of-the-art performance of our framework in the problem of interest.

4.2 Related Works

The last decade has witnessed the emergence of algorithms for image modeling and rendering (e.g., [36, 76, 96]). In particular, a large effort has been devoted to the new view synthesis problem and quality results can be obtained for real-world objects, even provided with only one single object image. To name a few, by integrating auto-encoder with recurrent neural networks, [168] proposed the recurrent convolutional encoder-decoder network (*RCEDN*) to synthesize novel views of 3D objects through rotating the input image with a fixed angle. [142] also adopted convolutional network to generate novel object images using a single object image as input.

Besides the viewpoint, other factors, including illumination and scale, have also been studied in re-rendering object images. *Transforming auto-encoders* [62, 80, 184] is among the earliest works that attempted to learn a whole vector of instantiation parameters, which is then used to generate a transformed version of the input image, through training auto-encoder capsules. [24] introduced an unsupervised cross-covariance penalty (*XCov*) for learning disentangled representations of input images through the hidden layers of deep networks. Novel images can then be generated by resetting these disentangled representations according to specified configurations. [83] proposed the deep convolutional inverse graphics network (*DC-IGN*), which employs convolutional layers to de-render the input images and then re-render the output images using de-convolutional layers. Neurons of the *graphics codes* layer between convolutional and de-convolutional layers are encouraged to represent disentangled factors (identity, illumination, etc.) of the input image. Although the *XCov* and *DC-IGN* methods utilize label (tag) information to guide their models to learn disentangled representations from the input image, they do not consider the consistency between the image and its tags, i.e., the fact that the image and its tags share the same latent (disentangled) representation. In our framework, besides learning disentangled representations from images, we dig up the correlation between tags and disentangled representations, which enables our proposed *TD-GAN* to achieve better performance in learning disentangled representations.

Moreover, image re-rendering is related to image generation, which aims to

generate images using scene-level features. [36] proposed an ‘up-convolutional’ network to generate images of chairs given the label of chair style and sine (co-sine) value of azimuth angle. Experimentally, the ‘up-convolutional’ network has exhibited strong ability to generate images with specified features. However, it can only generate images for the objects existing in the training set and simply interpolate between them. Similarly, generative adversarial nets (*GANs*) [54] and *GANs*-based models [21, 50] are trained to generate images from different variables, such as, unstructured noise vector [55, 127, 177], text [52, 57, 129, 161], latent code [13, 21, 95, 144], etc. Based on the adversarial training strategy, the generative network in *GANs* maps input variables into image space through playing a minimax optimization with the discriminative network.

4.3 Method

A set \mathcal{X}^L of tagged images is considered in the supervised setting. Let $\{(\mathbf{x}_1, \mathbf{C}_1), \dots, (\mathbf{x}_{|\mathcal{X}^L|}, \mathbf{C}_{|\mathcal{X}^L|})\}$ denote the whole training dataset, where $\mathbf{x}_i \in \mathcal{X}^L$ denotes the i^{th} image and $\mathbf{C}_i \in \mathcal{C}$ its corresponding *tag codes* which can be represented as: $\mathbf{C}_i = (\mathbf{c}_i^{\text{ide}}, \mathbf{c}_i^{\text{view}}, \mathbf{c}_i^{\text{exp}}, \dots)$. One-hot encoding vectors are employed to describe tag information (e.g., in the case of expression tag with three candidates [neutral, smile, disgust], $\mathbf{c}_i^{\text{exp}} = [0, 1, 0]$ for an image with “smile” tag). An additional untagged training image set \mathcal{X}^U is considered in the semi-supervised setting.

4.3.1 Main Framework

As shown in Fig. 4.1, *TD-GAN* is composed of four parts: a tag mapping net g , a generative network G , a disentangling network R and a discriminative network D . The tag mapping net g and the disentangling network R aim to map *tag codes* and real-world images into latent disentangled representations, which can then be decoded by the generative network G . Finally, the discriminative network D is introduced to perform the adversarial training strategy which plays a minimax game with the networks R and G . There are three interrelated objectives in the framework of *TD-GAN* for realizing the optimal image re-rendering task.

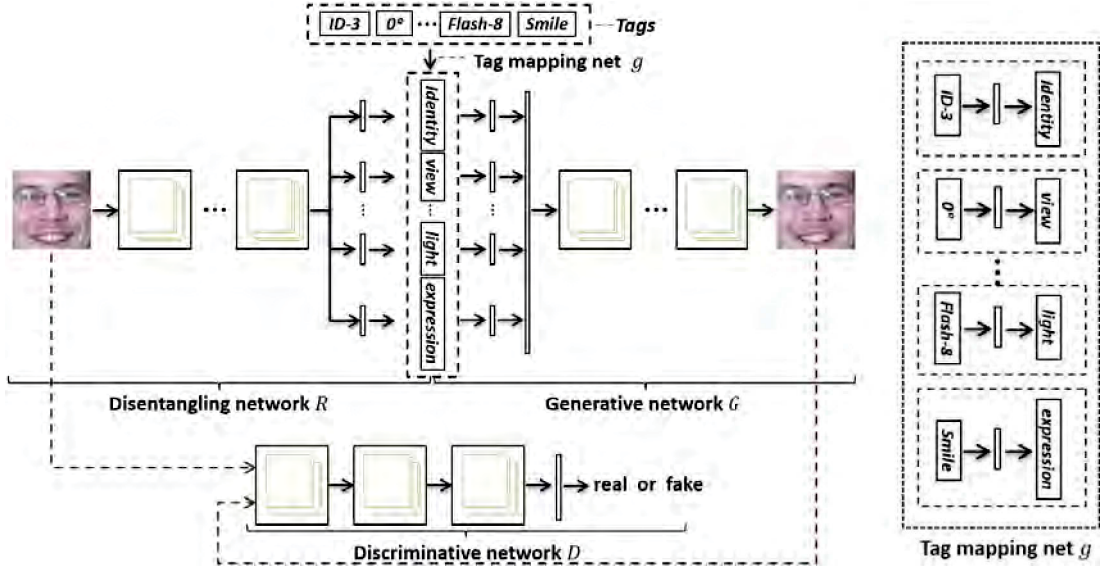


Figure 4.1: Model architecture. *TD-GAN* is composed of four parts: a tag mapping net g , a disentangling network R , a generative network G and a discriminative network D . (a) During training, the tag mapping net g and the generative network G are trained to render images with their tags. The disentangling network R aims to extract disentangled representations, which can be decoded by the network G . The discriminative network D plays a minimax game with networks G and R based on the adversarial training strategy. (b) During test, the disentangling network R extracts disentangled representations from the input image. After replacing one or multiple disentangled representations with the specified representations generated by the tag mapping net g , the image can be re-rendered through the generative network G .

4.3.1.1 Exploring consistency between the image and its tags.

In practice, the image provides a visual approach to recording the real-world object. Meanwhile, tags (identity, viewpoint, illumination, etc.) accompanied with the image can describe object in a textual or parametric way. The image and its tags consistently represent the same object, despite the difference in physical properties. It is therefore meaningful to explore the consistency between the image and its tags in the image re-rendering task. In the training procedure, the disentangling network R aims to extract the disentangled representations $R(\mathbf{x})$ of the input image \mathbf{x} ¹. Besides, *tag codes* \mathbf{C} are fed through the tag mapping net g to obtain the disentangled representations² $g(\mathbf{C})$ of the tags. The first objective of *TD-GAN* is to penalize the discrepancy between the disentangled representations $R(\mathbf{x})$ and $g(\mathbf{C})$ generated from the image and its tags. Using the $L2$ norm to penalize such a discrepancy, the energy function is defined as follows:

$$f_1(R, g) = \frac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^L} \| R(\mathbf{x}_i) - g(\mathbf{C}_i) \|_2^2 \quad (4.1)$$

Compared with valuable tag information, lots of untagged images can be easily harvested in practice. In order to explore useful information contained in these untagged images for network learning in the semi-supervised setting, by applying the generative network G on their disentangled representations $R(\mathbf{x})$, we utilize the following objective function so as to encourage the reconstructed image from the disentangled representations $R(\mathbf{x}_i)$ to be close to the genuine image \mathbf{x}_i :

$$\tilde{f}_1(G, R) = \frac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} \| G(R(\mathbf{x}_i)) - \mathbf{x}_i \|_2^2 \quad (4.2)$$

4.3.1.2 Maximizing image rendering capability.

Given image *tag codes* \mathbf{C} , the generative network G should have the capability to render the image with its disentangled representations $g(\mathbf{C})$. Note that the

¹Formally, $R(\mathbf{x}) = (r_{\text{ide}}, r_{\text{view}}, \dots)$, where those r_* denote disentangled representations extracted from the input image.

²The function g actually consists of a set of independent sub-functions designed to translate the tags into the corresponding representations, i.e., $g(\mathbf{C}) = (g_{\text{ide}}(\mathbf{c}^{\text{ide}}), g_{\text{view}}(\mathbf{c}^{\text{view}}), \dots)$.

disentangled representations extracted from the image and its tags have already been encouraged to be the same (see Eq. (4.1)). To maximize the rendering capability of network G , we tend to minimize the discrepancy between genuine images and rendered images via the following objective function:

$$f_2(G, g) = \frac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^L} \|G(g(\mathbf{C}_i)) - \mathbf{x}_i\|_2^2 \quad (4.3)$$

We can re-render images based on the networks R , G and g discussed above. However, such a way lacks strong driving force for continuously improving the re-rendering performance. Therefore, we introduce the discriminative network D into *TD-GAN* so as to minimize genuine image recognition loss, the detail of which is shown below.

4.3.1.3 Minimizing genuine image recognition loss.

Ideally, image re-rendering should be able to mix re-rendered images with genuine images so that they cannot be distinguished. To this end, we adopt the adversarial training strategy, based on the *GANs* model [54]. Adversarial training suggests the use of the discriminative network D as an adversary of the networks R and G . The discriminative network D outputs the probability that the input image is genuine, and tries its best to detect all those re-rendered images. Competition in this game encourages R , G and D to improve their solutions until the re-rendered images are indistinguishable from those genuine ones. Formally, the objective function can be formulated as:

$$f_3(R, G, D) = \mathbb{E}[\log D(\mathbf{x})] + \mathbb{E}[\log(1 - D(G(R(\mathbf{x})))]) \quad (4.4)$$

where \mathbb{E} is computed over those genuine images in the training set. Higher values of f_3 indicates better discriminative abilities, and vice versa.

4.3.2 Training Process

TD-GAN is composed of R , D , G and g , which can be optimized and learned using the alternating optimization strategy. In the following, we first present the optimization with respect to each component of *TD-GAN* at each iteration

in the context of the supervised setting, and then clarify the difference in the optimization for the semi-supervised setting.

We optimize the tag mapping net g , by fixing G^* , R^* and D^* . Since g maps tags to disentangled representations, optimizing g involves the first and second objectives (f_1 and f_2):

$$\mathcal{L}_g = \min_g \lambda_1 f_1(R^*, g) + \lambda_2 f_2(G^*, g) \quad (4.5)$$

Here, those λ 's are hyper-parameters balancing the influence of the corresponding terms, and R^* and G^* are fixed as the configurations obtained from the previous iteration (similarly for the other optimization steps presented below).

By fixing g^* , D^* and R^* , the search of the generative network G can be formulated as:

$$\mathcal{L}_G = \min_G \lambda_2 f_2(G, g^*) + \lambda_3 f_3(R^*, G, D^*) \quad (4.6)$$

where G determines the re-rendering procedure with disentangled representations as input.

The disentangling network R is trained by fixing g^* , G^* and D^* . Similarly, the main target of the network R is to infer disentangled representations from the input image (i.e., f_1), and it is also a part of adversarial training (i.e., f_3). Hence, the loss function with respect to R should be:

$$\mathcal{L}_R = \min_R \lambda_1 f_1(R, g^*) + \lambda_3 f_3(R, G^*, D^*) \quad (4.7)$$

The discriminative network D is introduced to cooperate with the adversarial training strategy, via the following optimization:

$$\mathcal{L}_D = \max_D \lambda_3 f_3(R^*, G^*, D) \quad (4.8)$$

We can observe that the above processing forms a minimax game, which aims to maximize the image re-rendering capability and to minimize the error in distinguishing between genuine and re-rendered images. In the semi-supervised setting, since G and R are used to re-render the untagged images as well, the objective functions in Eqs. (4.6) and (4.7) also include a weighted loss $\lambda_4 \tilde{f}_1$ for

Disentangling Net R	
Input: Image $\mathbf{x} - (128 \times 128 \times 3)$	
[Layer 1]	Conv. (3, 3, 64) Stride=2. <i>LReLU</i> BatchNorm
[Layer 2]	Conv. (3, 3, 128) Stride=2. <i>LReLU</i> BatchNorm
[Layer 3]	Conv. (3, 3, 256) Stride=2. <i>LReLU</i> BatchNorm
[Layer 4]	Conv. (3, 3, 256) Stride=1. <i>LReLU</i> BatchNorm
[Layer 5]	Conv. (3, 3, 256) Stride=1. <i>LReLU</i> BatchNorm
[Layer 6]	Conv. (3, 3, 512) Stride=2. <i>LReLU</i> BatchNorm
[Layer 7]	Conv. (3, 3, 512) Stride=1. <i>ReLU</i> BatchNorm
[Layer 8]	FC. 5120 <i>LReLU</i>
[Layer 9]	FC. separate <i>Tanh</i>
Output:	Disentagnled Representations $R(\mathbf{x})$

Table 4.1: Details of the disentangling network used for all the experiments.

untagged image re-rendering.

4.3.3 Image Re-rendering

In our framework, the trained disentangling network R is able to transform the input image into disentangled representations corresponding to those scene properties (e.g., identity, viewpoint, etc.). Hence, in the test stage, given an unseen image \mathbf{x}_{test} of the object of interest as input, the disentangling network R will output its disentangled representations. The image re-rendering task is performed simply by replacing one or multiple disentangled representations with the specified representation(s) generated by the tag mapping net g . The obtained disentangled representations are then fed to the generative network G so as to output the re-rendering result.

Discriminative Net D	
Input: Image $\mathbf{x} - (128 \times 128 \times 3)$	
[Layer 1]	Conv. (3, 3, 64) Stride=2. <i>LReLU</i> BatchNorm
[Layer 2]	Conv. (3, 3, 128) Stride=2. <i>LReLU</i> BatchNorm
[Layer 3]	Conv. (3, 3, 256) Stride=2. <i>LReLU</i> BatchNorm
[Layer 4]	Conv. (3, 3, 256) Stride=1. <i>LReLU</i> BatchNorm
[Layer 5]	Conv. (3, 3, 256) Stride=1. <i>LReLU</i> BatchNorm
[Layer 6]	Conv. (3, 3, 512) Stride=2. <i>LReLU</i> BatchNorm
[Layer 7]	Conv. (3, 3, 512) Stride=1. <i>ReLU</i> BatchNorm
[Layer 8]	FC. 2560 <i>LReLU</i>
[Layer 9]	FC. 1 <i>Sigmoid</i>
Output:	Probability of being genuine $D(\mathbf{x})$

Table 4.2: Details of the discriminative network used for all the experiments.

4.4 Experiments

We evaluated the performance of our *TD-GAN* method based on two challenging datasets: *3D-chairs* dataset [4] and *Multi-PIE* database [56]. The obtained results demonstrate the advantage of our method in learning interpretable disentangled representations and re-rendering images of unseen objects with tag configurations.

4.4.1 Implementation Details

In our experiments, all images were resized to $128 \times 128 \times 3$. Meanwhile, all layers, except those fully-connected with disentangled representations were fixed and the details of these layers, are shown in Table 4.1. Distinct *tag codes* (i.e., different architecture configurations of the tag mapping net g) were chosen for different tasks and we summarize in Table 4.4 all the settings used in our experiments.

Generative Net G	
Input:	Disentangled representations $R(\mathbf{x})$
[Layer 1]	Concatenate Layer
[Layer 2]	FC. (4, 4, 1024) <i>ReLU</i> BatchNorm
[Layer 3]	Deconv. (4, 4, 512) Stride=2 <i>ReLU</i> BatchNorm
[Layer 4]	Deconv. (4, 4, 256) Stride=2 <i>ReLU</i> BatchNorm
[Layer 5]	Deconv. (4, 4, 128) Stride=2 <i>ReLU</i> BatchNorm
[Layer 6]	Deconv. (4, 4, 64) Stride=2 <i>ReLU</i> BatchNorm
[Layer 7]	Deconv. (4, 4, 3) Stride=2 <i>Tanh</i> .
Output:	Generated Image $G(R(\mathbf{x})) - (128 \times 128 \times 3)$

Table 4.3: Details of the generative network used for all the experiments.

Moreover, our proposed *TD-GAN* was implemented based on *Theano* [10]. All the parameters were initialized using a zero-mean Gaussian with variance 0.05. The same weights ($\lambda_1 = 10, \lambda_2 = 10, \lambda_3 = 1$, and $\lambda_4 = 50$) were used in all the experiments. Based on the training process described above, we alternatively updated the disentangling network, the discriminative network and the generative network (with the tag mapping net). The optimization was done based on the *ADAM* solver [75] with momentum 0.5, where the learning rate was set as 0.0005 for the disentangling network and 0.0002 for all the other networks. We utilized a minibatch size of 50 and trained for around 900 epochs.

4.4.2 Performance Criteria

We qualitatively and quantitatively evaluated the performance of our method. On the one hand, following existing works [24, 142], we choose to use the qualitative criteria to evaluate the performance of re-rendered images, i.e., evaluating through human observers. To this end, we directly report the input image and its re-rendered images for a set of representative test examples. On the other

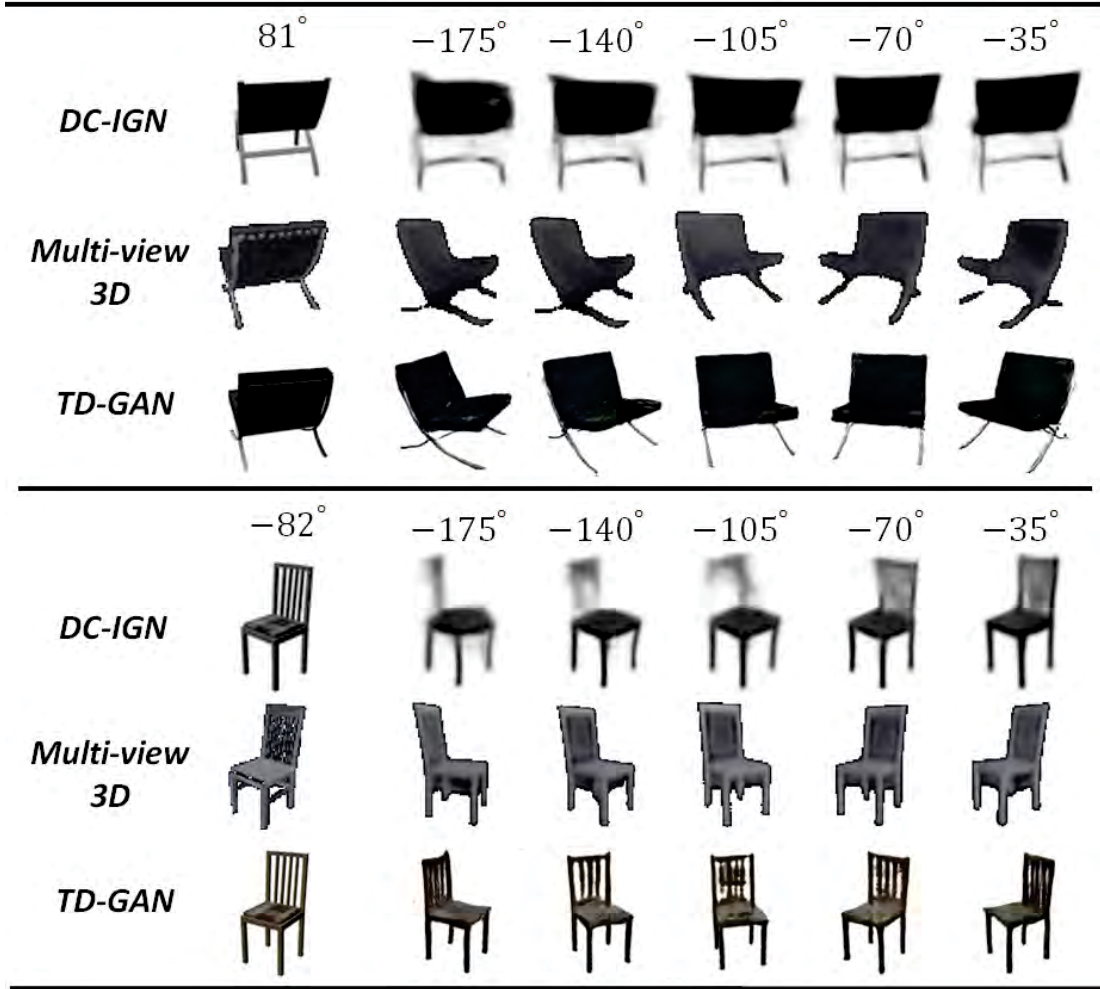


Figure 4.2: Novel view synthesis results of two previous method and ours. For each method, the leftmost image is the input image, and the images on its right side were re-rendered under different viewpoints.

	Tag	Hidden layer	Disentangled
Chair _{fully} ^{identity}	500	FC.1024 <i>ReLU</i>	FC.1024 <i>Tanh</i>
Chair _{semi} ^{identity}	100	FC.1024 <i>ReLU</i>	FC.1024 <i>Tanh</i>
Chair ^{viewpoint}	31	FC. 512 <i>ReLU</i>	FC. 512 <i>Tanh</i>
Face ^{identity}	200	FC.1024 <i>ReLU</i>	FC.1024 <i>Tanh</i>
Face ^{illumination}	19	FC. 512 <i>ReLU</i>	FC. 512 <i>Tanh</i>
Face ^{viewpoint}	13	FC. 256 <i>ReLU</i>	FC. 256 <i>Tanh</i>
Face ^{expression}	3	FC. 256 <i>ReLU</i>	FC. 256 <i>Tanh</i>

Table 4.4: Experimental settings of the tag mapping net g .

hand, as for quantitative criteria, same as previous work [83], we measured the mean squared error (MSE) between the re-rendered images and the corresponding ground truths over the test set.

4.4.3 Experimental Results

4.4.3.1 3D-chairs dataset.

The *3D-chairs* dataset [4] contains 86,366 images rendered from 1,393 3D CAD models of different chairs. For each chair, 62 viewpoints are taken from 31 azimuth angles (with a step of 11 or 12 degrees) and 2 elevation angles (20 and 30 degrees). Since the *3D-chairs* dataset records chair images from different viewpoints, we firstly performed the novel view synthesis task, which takes a single image of an unseen object as input, and attempts to re-render images under novel viewpoints. Following the experimental setting of existing works [36, 142, 168], we selected 809 chair models from all 1,393 chair models by removing near-duplicate (e.g., those differing only in color) and low-quality models. The first 500 models were used for training purpose, while the remaining 309 models for test.

In Fig. 4.2, we show the novel view synthesis results of *TD-GAN* and two

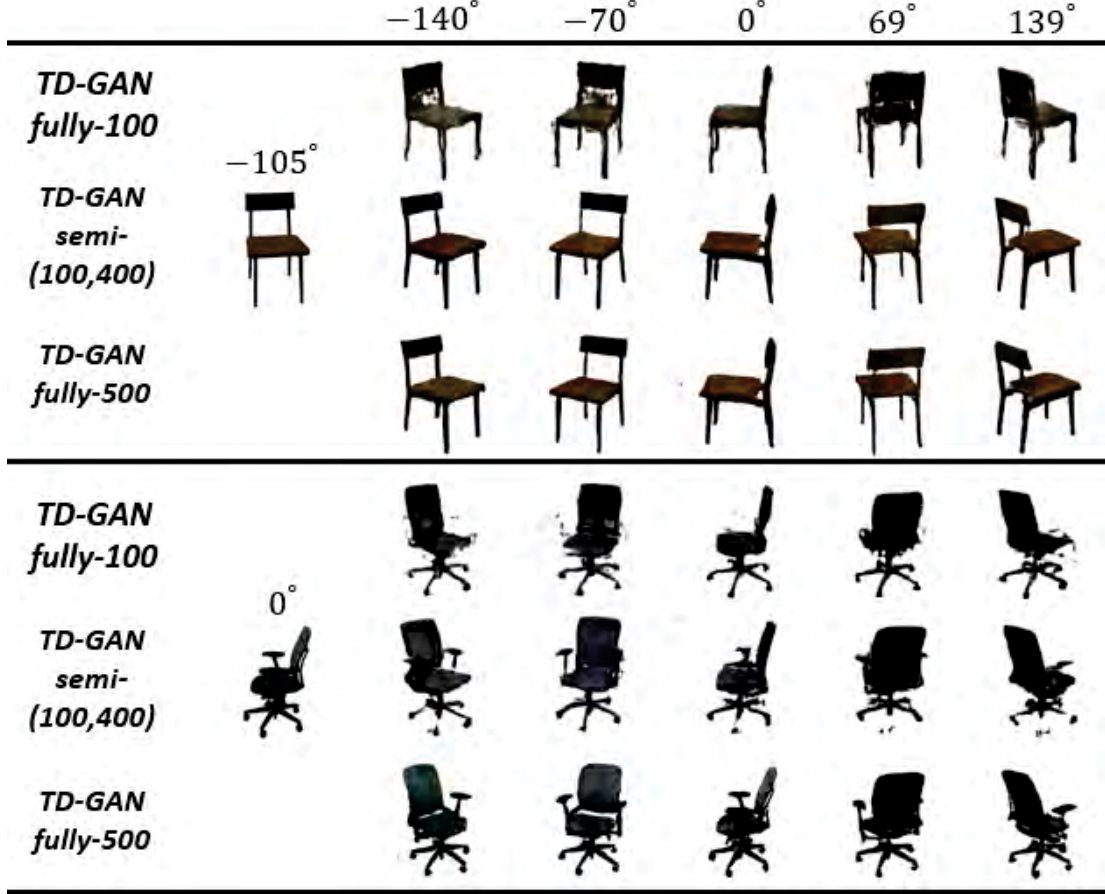


Figure 4.3: Novel view synthesis results of *TD-GAN* trained in three settings. The images are arranged similarly to Fig. 4.2.

previous methods on the *3D-chairs* dataset. Following the comparison in [142], we adopt the results reported by existing works [83,142] as references, and exhibit our results over the same (or very similar) test images. Among them, *Multi-View 3D* [142] was designed to re-render unseen viewpoints of 3D objects. Compared with *Multi-View 3D*, our method not only achieves comparable performance in viewpoint transformation, but also be able to carry out other image re-rendering tasks, such as illumination transformation, expression transformation, etc. Despite the fact that both *DC-IGN* [83] and *TD-GAN* were developed to re-render

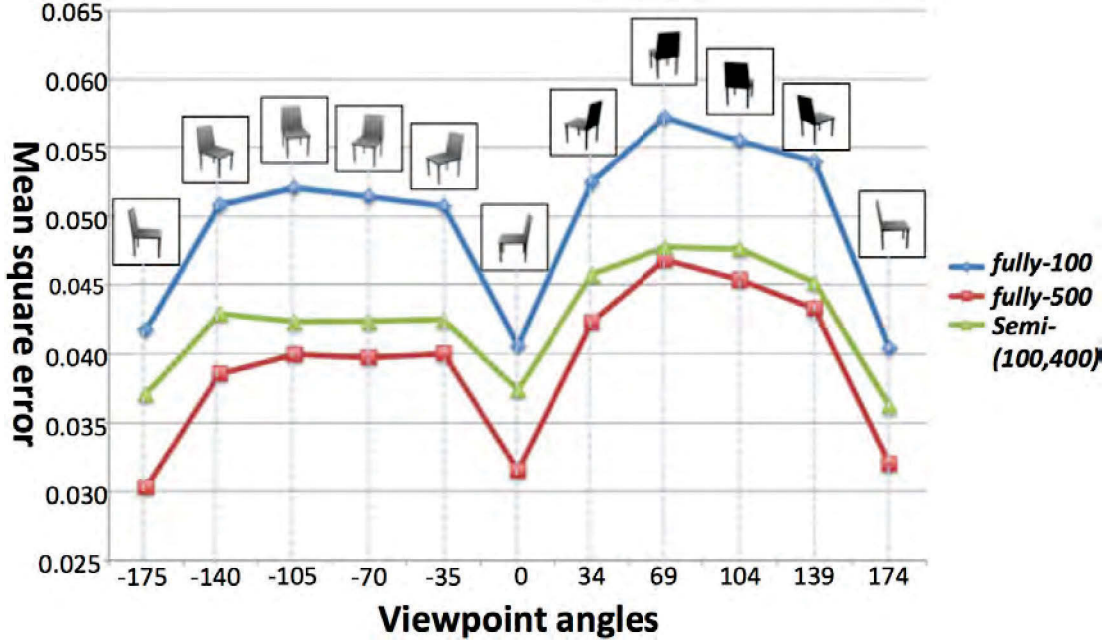


Figure 4.4: *MSE* of *TD-GAN* trained in three settings. *MSE* was calculated over all the chair images under 0° in the test set as inputs. Chair images are used to indicate target viewpoints.

images with multiple different configurations, the obtained experimental results demonstrate that *TD-GAN* achieves much better results. Furthermore, we evaluated the semi-supervised extension of our *TD-GAN* method. To this end, within the aforementioned experimental setting, we tested the following three representative training settings: using all the 500 training models and their tags, only the first 100 models and their tags, and all the 500 training models but only the tags of the first 100 models (referred to as *fully-500*, *fully-100*, and *semi-(100,400)*, respectively). We show in Fig. 4.3 some representative qualitative results on the same test images, from which we can observe that: (i) the fully-supervised setting with all the 500 models (i.e., *fully-500*) achieves the best performance; (ii) the semi-supervised setting (i.e., *semi-(100,400)*) shows slightly degraded performance compared to *fully-500*; and (iii) both of *fully-500* and *semi-(100,400)* perform much better than the fully-supervised setting with only the first 100

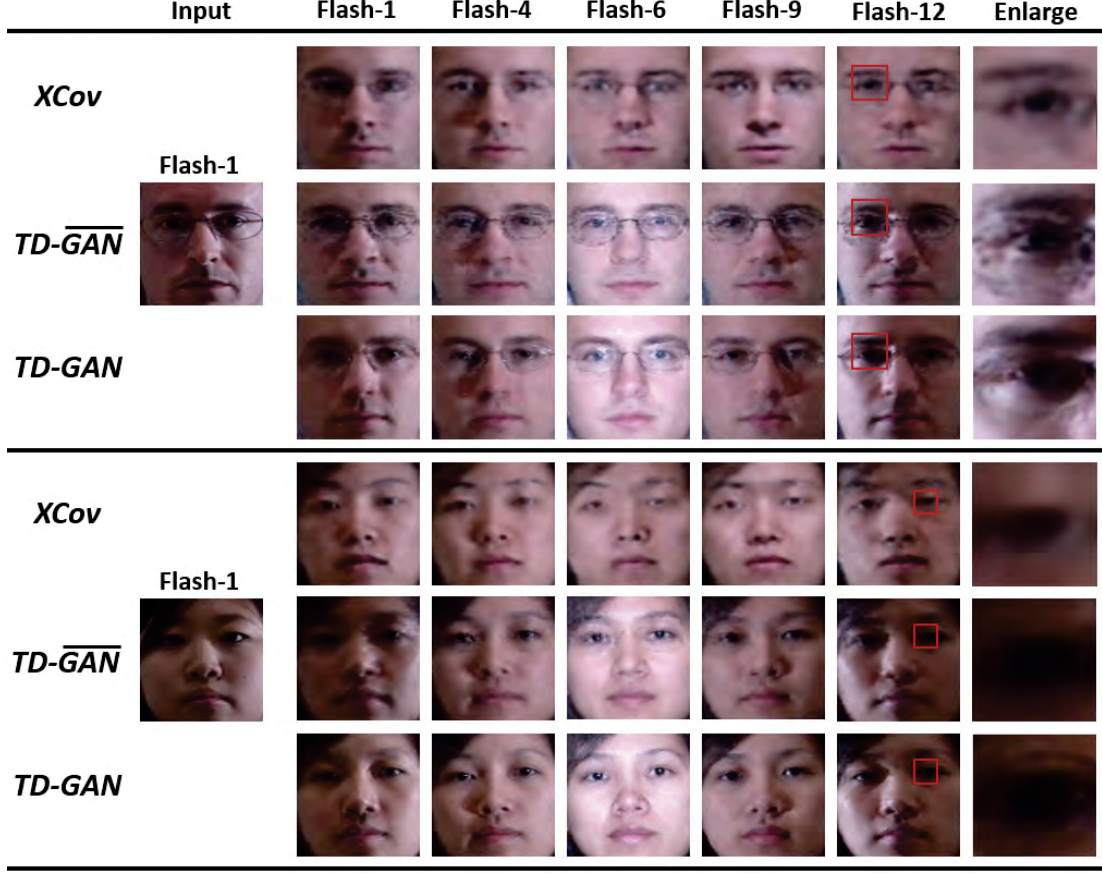


Figure 4.5: Illumination transformation of human face (best view in color). Given an image (leftmost) of human face, we reported a set of re-rendered images on its right side.

models (i.e., *fully-100*). For quantitative results, we selected all the images of chair objects under 0° in the test set as inputs, and re-rendered the images under different viewpoints. We report the *MSE* values for a set of individual target viewpoints in Fig. 4.4, and the global *MSE* over all the target viewpoints is 0.03875, 0.04379 and 0.05018 in the *fully-500*, *semi-(100,400)* and *fully-100* settings, respectively. Finally, from both the qualitative and quantitative results, we can conclude that the introduction of untagged data (semi-supervised learning) is indeed beneficial for improving the performance of the *TD-GAN* method.

	$XCov$	$TD-\overline{GAN}$	$TD-GAN$
Flash-1	0.5776	0.5623	0.5280
Flash-4	5.0692	0.8972	0.8818
Flash-6	4.3991	1.2509	0.1079
Flash-9	3.4639	0.6145	0.5870
Flash-12	2.4624	0.7142	0.6973
All Flash (mean)	3.8675	0.6966	0.6667

Table 4.5: MSE ($\times 10^{-2}$) of illumination transformation.

4.4.3.2 Multi-PIE database.

Multi-PIE [56] is a face dataset containing images of 337 people under 15 camera-poses (13 camera-poses with 15° intervals at head height and 2 additional ones located above the subject) and 19 illumination conditions in up to four sessions. Following previous works [24, 29], we cropped all face images based on manually annotated landmarks on eyes, nose and mouth. Since *Multi-PIE* records several factors (tags) of human faces, it is suitable for training $TD-GAN$ to perform various image re-rendering tasks.

We evaluated the performance of $TD-GAN$ in re-rendering face images under various illuminations. By fixing viewpoint (0°) and expression (neutral), we selected all 337 identities in all four sessions under different illuminations as the data setting. Among the 337 identities, the first 200 ones were used for training, and the remaining 137 ones for test. Besides our $TD-GAN$, we also evaluated $XCov$ [24] and $TD-GAN$ without the use of generative adversarial loss (i.e., f_3 in Eq. (4.4)) in the same task (refer to as $TD-\overline{GAN}$). In Fig. 4.5, we show some images re-rendered by those methods using the same images as inputs. Table 4.5 reports MSE for the images re-rendered with five individual target illuminations and with all the target illuminations, using the images of all human faces under a fixed illumination (*Flash-1*) in the test set as inputs. According to Fig. 4.5 and

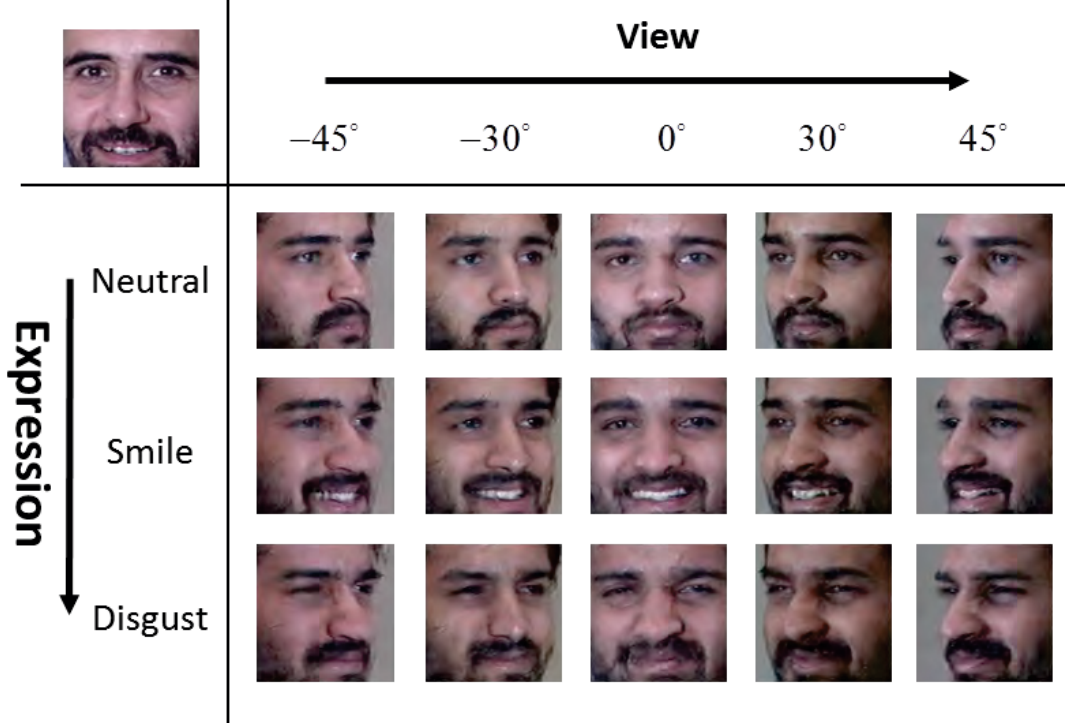


Figure 4.6: Multi-factor transformation (best view in color). Given a single image as input (the up-left one), its viewpoint and expression were jointly transformed.

Table 4.5, we can draw the following two conclusions. Firstly, *XCov* performs much worse in re-rendering images when the specified illumination condition is different from that of the input image, while our method performs well in all those re-rendering with illumination transformation experiments. Indeed, ideally, the re-rendering performance should be independent with the scene properties (e.g., illumination) of the input image, unless the extracted identity representation is correlated to those properties. Secondly, the introduction of generative adversarial loss really helps our method to generate more realistic images with fine details.

Last but not least, in order to validate the capability of re-rendering images with multi-factor transformation, we performed an experiment where the viewpoint and expression were jointly configured. We used the data of session 3 as

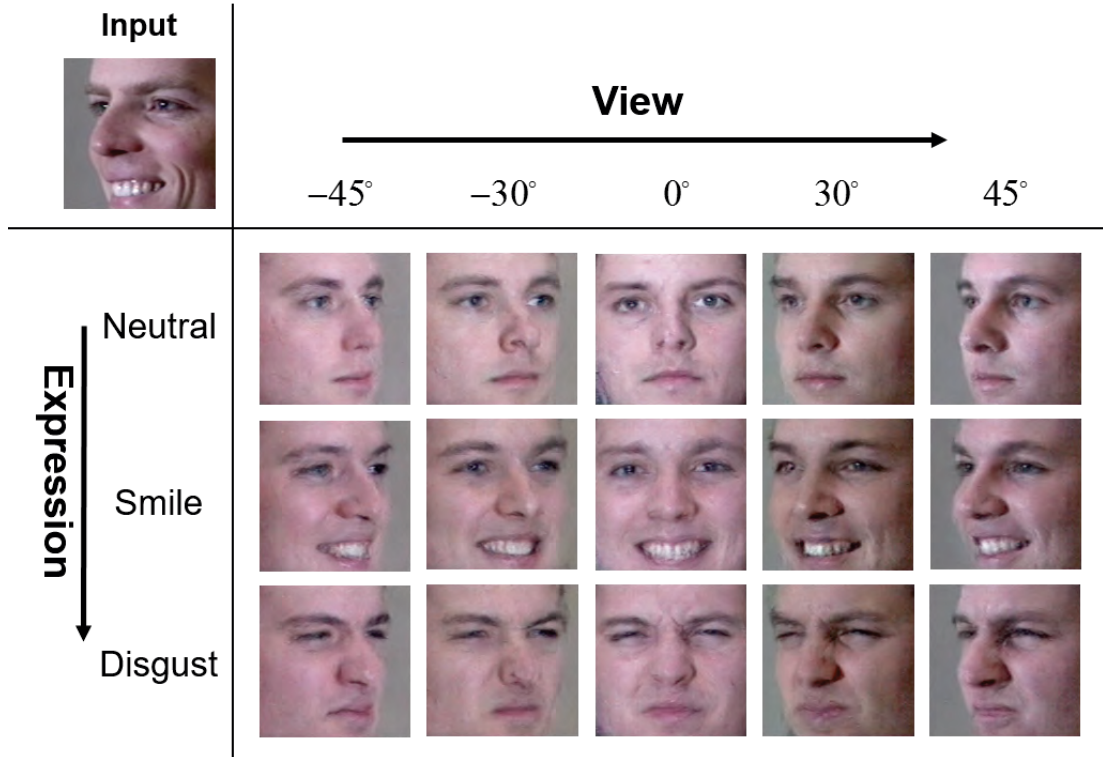


Figure 4.7: Multi-factor transformation (best view in color). Given a single image as input (the up-left one), its viewpoint and expression were jointly transformed.

the dataset, which contains 230 identities with three expressions (neutral, smile and disgust). Similarly, the first 200 identities were used as the training set, and the remaining 30 ones served as the test set. As shown in Fig. 4.6, given a test image, our method is able to effectively re-render images of the same identity with different expressions and viewpoints. The re-rendered images look quite natural and make us believe that they are indeed the genuine images of the same face exhibiting in the input image.

4.5 Summary

In this chapter, we have investigated the image re-rendering problem by developing a principled Tag Disentangled Generative Adversarial Networks (*TD-GAN*). Images and their associated tags have been fully and finely explored to discover the disentangled representations of real-world objects. The whole framework is established with three interrelated objectives, and can be effectively optimized with respect to the involved four essential parts. Experimental results on real-world datasets demonstrate that the proposed *TD-GAN* framework is effective and promising for practical image re-rendering applications.

For many real-world applications, manipulating real-world images by desired properties is a meaningful yet challenging problem. In this work, we propose a principled framework (named TD-GAN) for re-rendering new images for the object of interest from a single image of it by specifying multiple scene properties (such as viewpoint, illumination, expression, etc.) and the experiments on two challenging datasets demonstrate the state-of-the-art performance of this framework. Through adjusting image conditions (e.g., illumination, pose), TDGAN has the capability to synthesize high-quality images for human observation, image classification or object recognition. Meanwhile, utilizing generated continuous pose of the object of interest, a 3D model can be reconstructed easily from a single image. Moreover, it may also be applied to virtual reality systems, where one tends to naturally paste persons into a virtual environment. In general, TDGAN provides a promising framework for editing images using simple descriptions.

Chapter 5

Conclusions

In recent years, GANs have been successfully applied to many image synthesis tasks [6], such as image inpainting, image super-resolution, image transformation and image re-rendering. In this thesis, I contribute to both improving the quality of the synthesized image and the manipulation of an image of it by specifying multiple scene properties.

Firstly, in order to stabilize the training process of GAN models, a novel GAN framework called evolutionary generative adversarial networks (E-GAN) is proposed. Unlike existing GANs, which employ a pre-defined adversarial objective function alternately training a generator and a discriminator, we utilize different adversarial training objectives to optimize a population of generators go against the discriminator. An evaluation mechanism is devised to measure the quality and diversity of generated samples, such that only well-performing generator(s) are preserved and used for further training. In this way, E-GAN overcomes the limitations of an individual adversarial training objective and always preserves the best offspring, contributing to progress in and the success of GANs.

Then, a Perceptual Adversarial Networks (PAN) for image-to-image transformation tasks is devised. Besides the generative adversarial loss widely used in GANs, the perceptual adversarial loss was proposed, which undergoes an adversarial training process between the image transformation network and the hidden layers of the discriminative network. The hidden layers and the output of the discriminative network are upgraded to constantly and automatically discover the discrepancy between the transformed image and the corresponding

ground-truth, while the image transformation network is trained to minimize the discrepancy explored by the discriminative network. Experiments on different challenging datasets and image transformation tasks demonstrate the convincing performance of the proposed framework in the supervised image-to-image transformation tasks.

Finally, a principled Tag Disentangled Generative Adversarial Networks (TD-GAN) is devised for re-rendering new images for the object of interest from a single image of it by specifying multiple scene properties (such as viewpoint, illumination, expression, *etc*). Given an input image, a disentangling network extracts disentangled and interpretable representations, which are then used to generate images by the generative network. In order to boost the quality of disentangled representations, the tag mapping net is integrated to explore the consistency between the image and its tags. Experiments on re-rendering 3D chairs and human faces demonstrate the state-of-the-art performance of the proposed framework in the problem of interest.

Future works will focus on synthesizing and manipulating real-world images based on unsupervised training data, which ask that we further explore the capability of adversarial learning paradigm and combining it with possible generative models.

Appendix A

5.1 Proof of the Optimal Discriminator

First, we consider the optimal discriminator of our proposed E-GAN. Similar to the original GAN and its heuristic variant (*i.e.*, ‘ $-\log D$ trick’), we employ the sigmoid cross-entropy to penalize discriminator D to distinguish real samples and generated samples (Eq. 2). Hence, according to the proof in [54], the optimal discriminator D^* is

$$D_{\text{E-GAN}}^* = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}.$$

In addition, least-squares GAN [104] utilize the ‘least-squares’ objective to train its discriminator, *i.e.*,

$$\begin{aligned}\mathcal{L}_D^{\text{LSGAN}} &= \frac{1}{2}\mathbb{E}_{x \sim p_{\text{data}}}[(D(x) - 1)^2] + \frac{1}{2}\mathbb{E}_{z \sim p_z}[D(G(z))^2] \\ &= \frac{1}{2}\mathbb{E}_{x \sim p_{\text{data}}}[(D(x) - 1)^2] + \frac{1}{2}\mathbb{E}_{z \sim p_g}[D(x)^2] \\ &= \int_x \frac{1}{2}(p_{\text{data}}(x)(D(x) - 1)^2 + p_g(x)D(x)^2)dx.\end{aligned}$$

With respect to $D(x)$, the function $\mathcal{L}_D^{\text{LSGAN}}$ achieves its minimum in $[0, 1]$ at $\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$, which is equivalent to that of E-GAN.

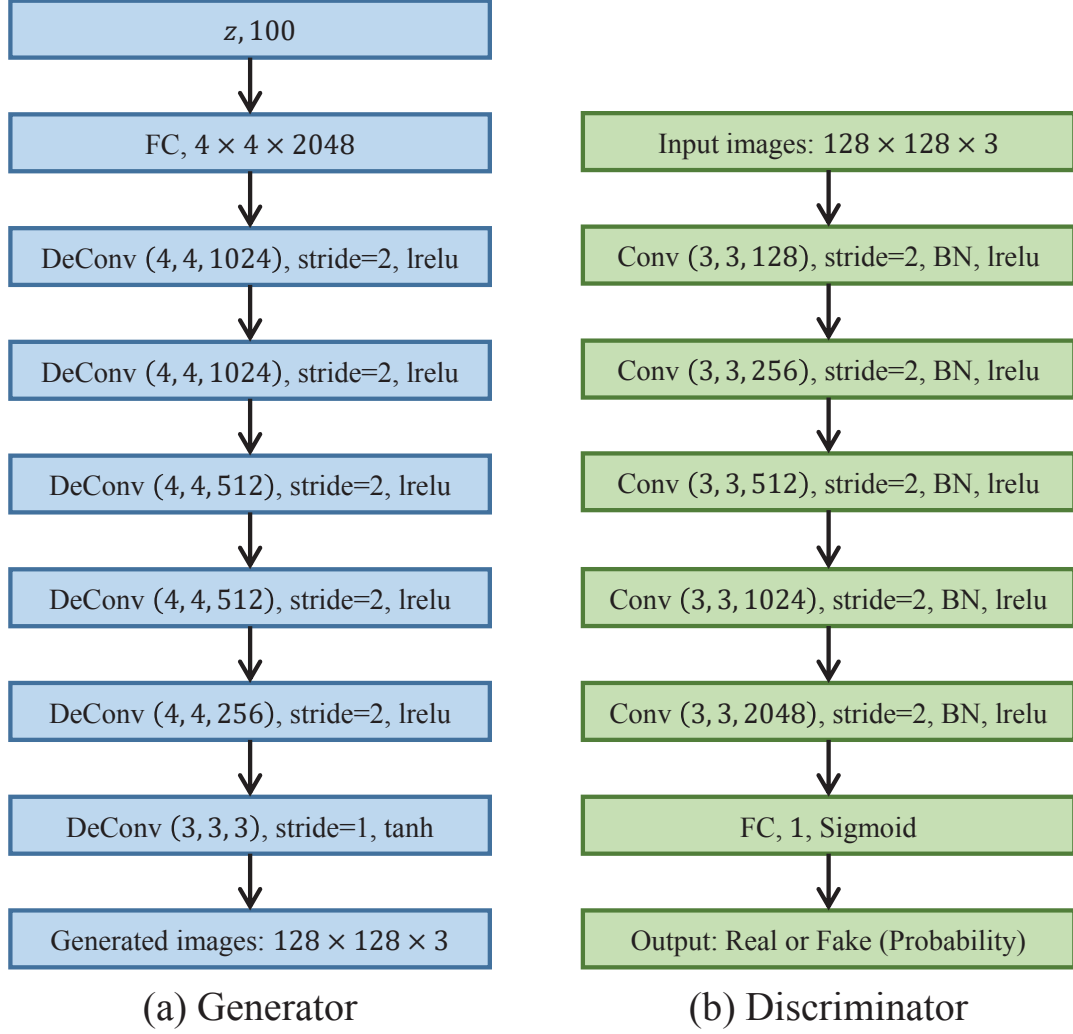


Figure 5.1: Network architectures for generating 128×128 images.

5.2 Network Architectures

In Fig. 5.1, we illustrate the network architectures that we used for generating 128×128 RGB images (both LSUN bedrooms and CelebA human faces). For other experiments, we adopt the same network architectures from existing works [58, 127].

5.3 Synthetic Datasets

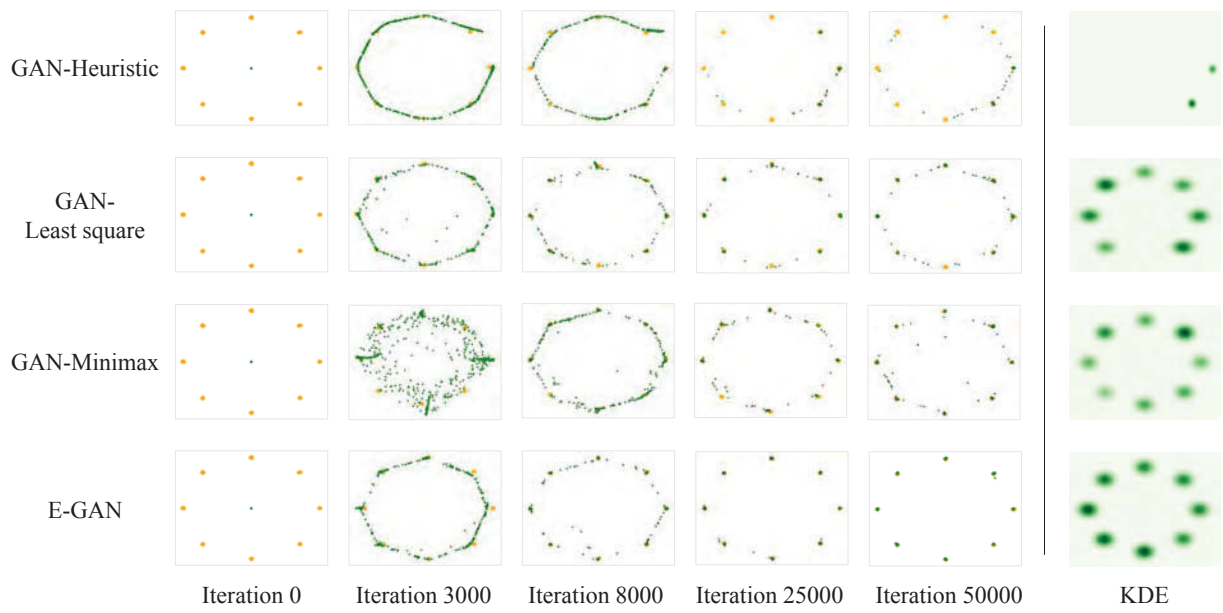


Figure 5.2: Different GANs learning a mixture of 8 Gaussians arranged in a circle.

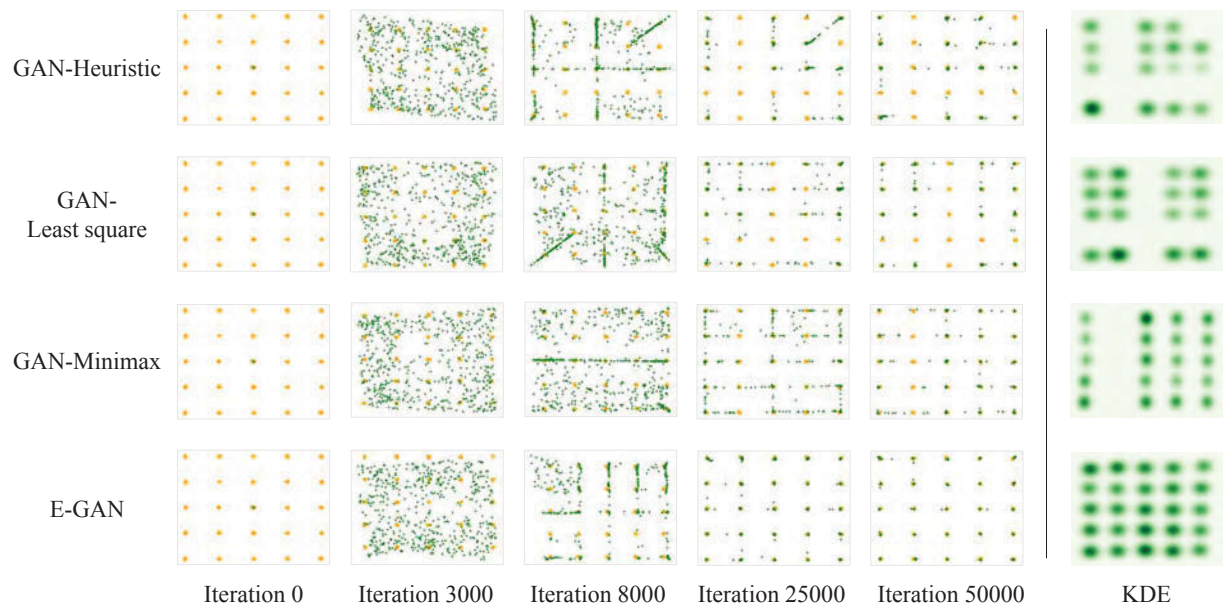


Figure 5.3: Different GANs learning a mixture of 25 Gaussians arranged in a grid.

5.4 Cifar-10

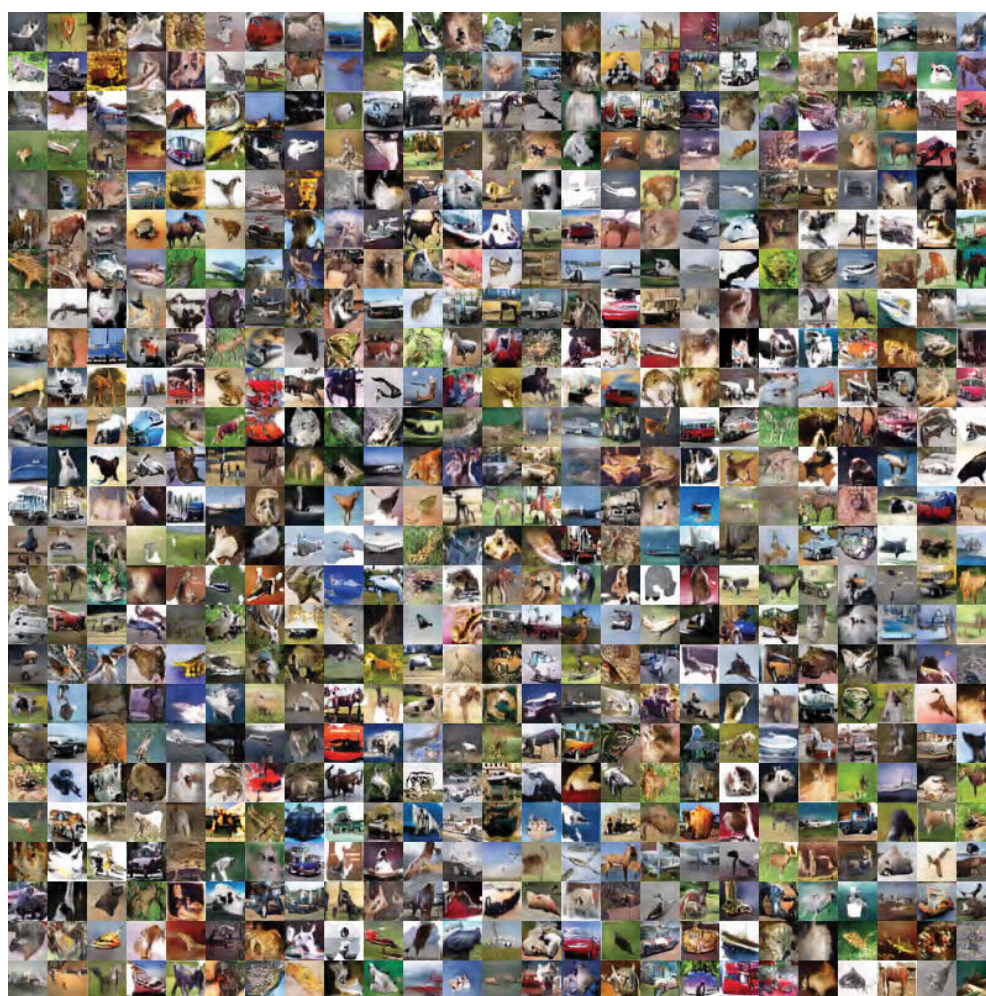
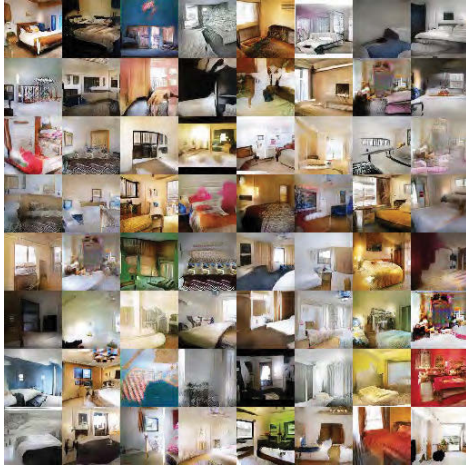


Figure 5.4: Generated samples on CIFAR-10 dataset.

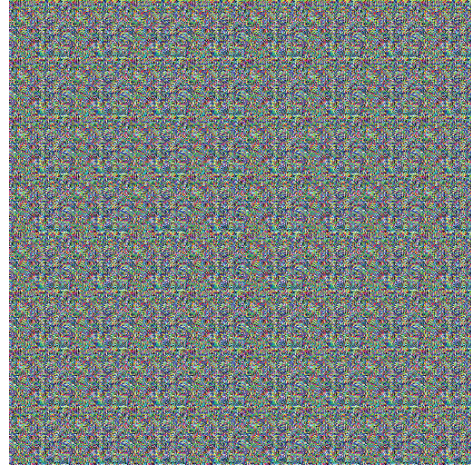
5.5 LSUN Bedrooms



Figure 5.5: Generated images on 128×128 LSUN bedrooms.



G : DCGAN, D : DCGAN



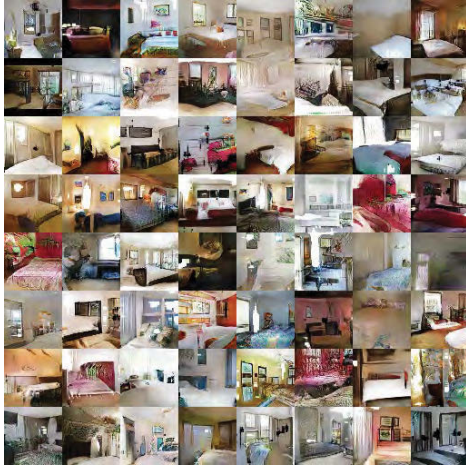
G : DCGAN, D : 2-Conv-1-FC LReLU



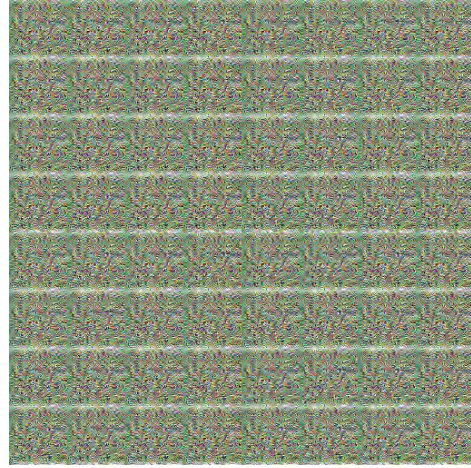
G : No BN and const. number of filters, Both G and D : No BN and const. number of filters
 D : DCGAN



Figure 5.6: Method: DCGAN



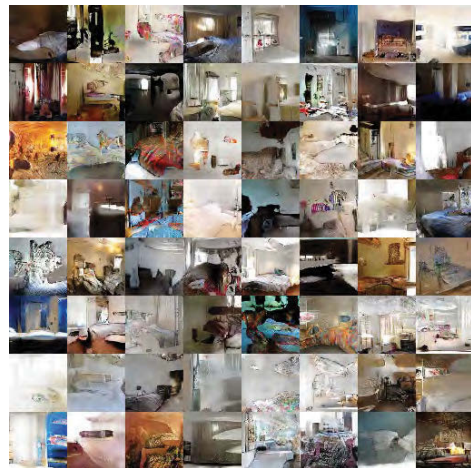
G : DCGAN, D : DCGAN



G : DCGAN, D : 2-Conv-1-FC LReLU

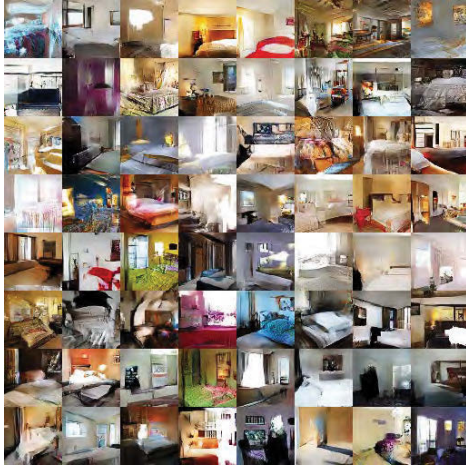


G : No BN and const. number of fillters, Both G and D : No BN and const. number of fillters
 D : DCGAN

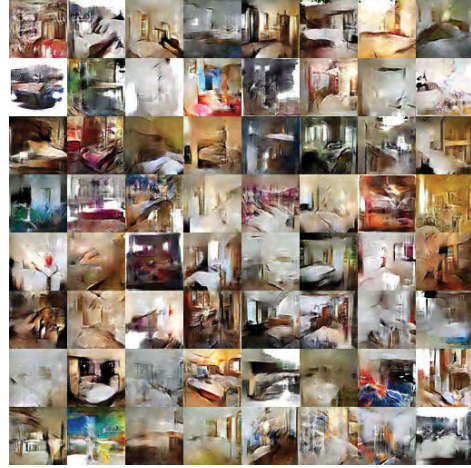


G : No BN and const. number of fillters, Both G and D : No BN and const. number of fillters
 D : DCGAN

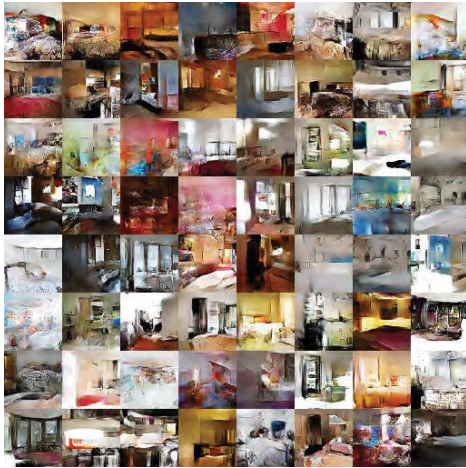
Figure 5.7: Method: LSGAN



G : DCGAN, D : DCGAN



G : DCGAN, D : 2-Conv-1-FC LReLU



G : No BN and const. number of filters, Both G and D : No BN and const. number of filters
 D : DCGAN

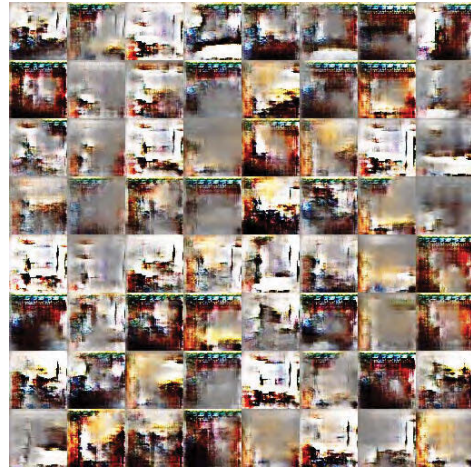
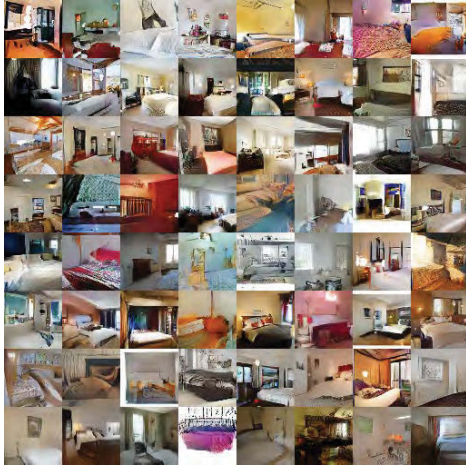


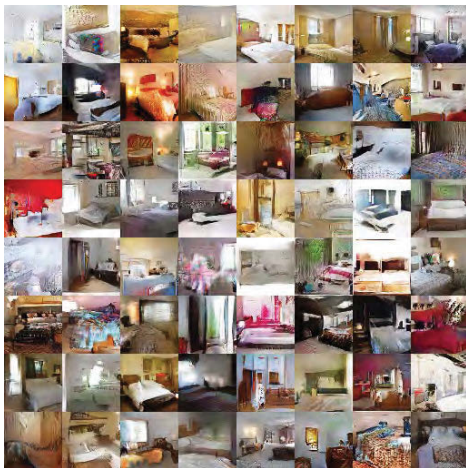
Figure 5.8: Method: WGAN



G : DCGAN, D : DCGAN



G : DCGAN, D : 2-Conv-1-FC LReLU



G : No BN and const. number of fillters, Both G and D : No BN and const. number of fillters
 D : DCGAN

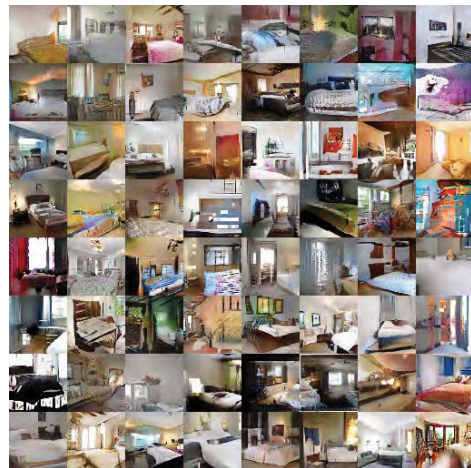
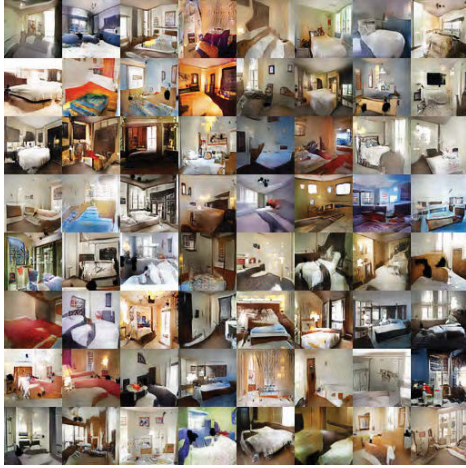
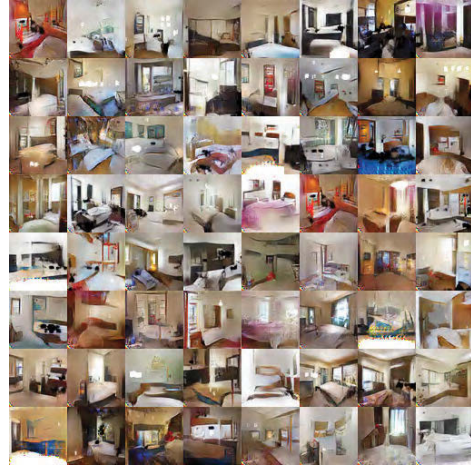


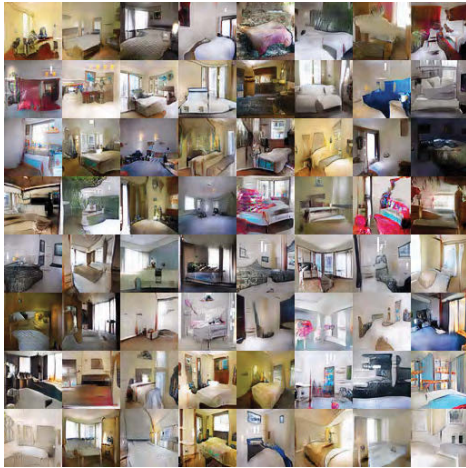
Figure 5.9: Method: WGAN-GP



G : DCGAN, D : DCGAN



G : DCGAN, D : 2-Conv-1-FC LReLU



G : No BN and const. number of fillters, Both G and D : No BN and const. number of fillters
 D : DCGAN



number of fillters

Figure 5.10: Method: E-GAN

5.6 CelebA Faces

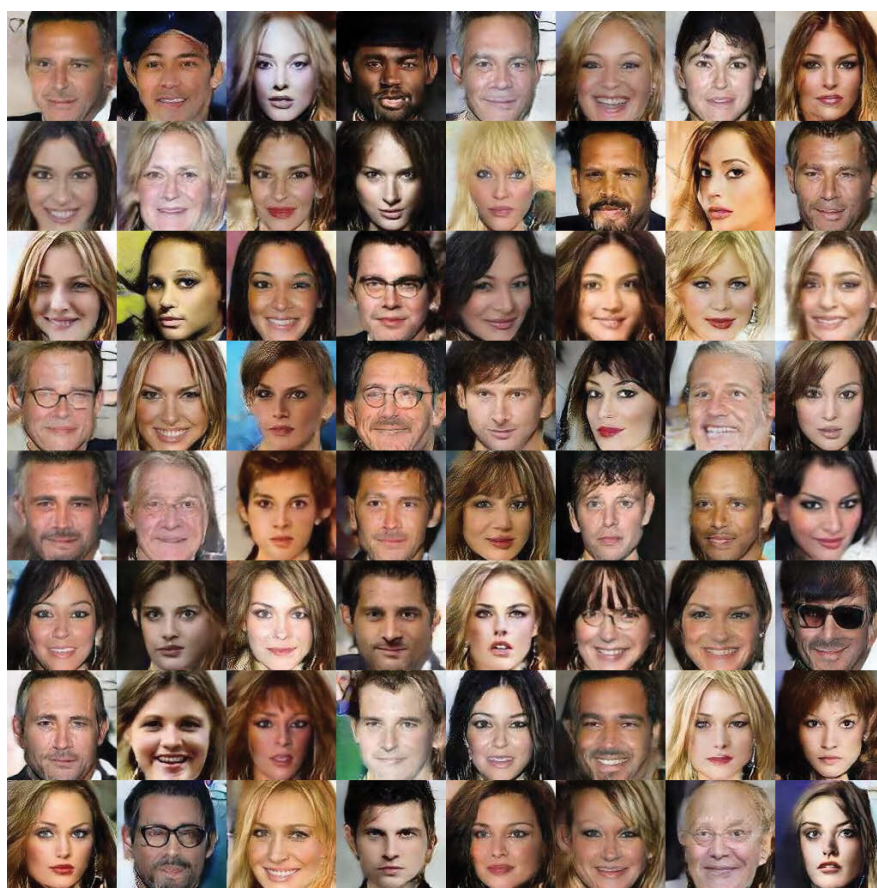


Figure 5.11: Generated human face images on 128×128 CelebA dataset.

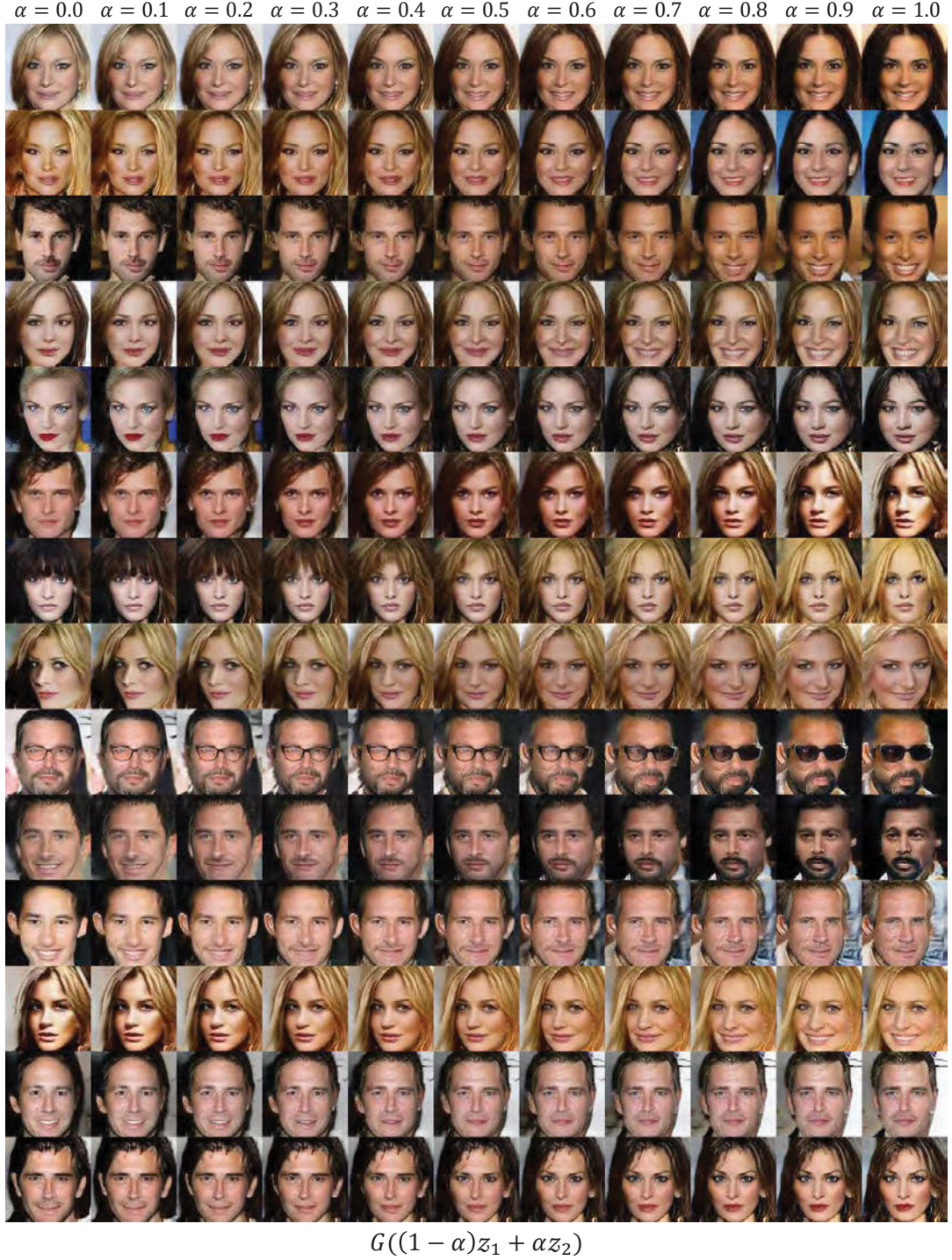


Figure 5.12: Interpolating in latent space. For selected pairs of the generated images from a well-trained E-GAN model, we record their latent vectors z_1 and z_2 . Then, samples between them are generated by linear interpolation between these two vectors.

References

- [1] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 6, 16, 22, 24
- [2] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 214–223. 6, 9, 15, 17, 45, 47
- [3] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, “Generalization and equilibrium in generative adversarial nets (GANs),” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 224–232. 9, 15
- [4] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, “Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3762–3769. 79, 82
- [5] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, “Stochastic variational video prediction,” 2017. 15
- [6] A. Bansal, Y. Sheikh, and D. Ramanan, “Pixelnn: Example-based image synthesis,” 2017. 90
- [7] S. Benaim and L. Wolf, “One-sided unsupervised domain mapping,” 2017.

REFERENCES

- [8] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. 70
- [9] Y. Bengio, I. J. Goodfellow, and A. Courville, “Deep learning,” *An MIT Press book in preparation. Draft chapters available at [http://www. iro. umontreal. ca/ bengioy/dlbook](http://www.iro.umontreal.ca/~bengioy/dlbook)*, 2015. 70
- [10] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pasfcanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: A cpu and gpu math compiler in python,” in *Proc. 9th Python in Science Conf*, 2010, pp. 1–7. 55, 80
- [11] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424. 40
- [12] D. Berthelot, T. Schumm, and L. Metz, “Began: Boundary equilibrium generative adversarial networks,” 2017. 4
- [13] P. Bojanowski, A. Joulin, D. Lopezpaz, and A. Szlam, “Optimizing the latent space of generative networks,” 2017. 73
- [14] D. Bouchacourt, R. Tomioka, and S. Nowozin, “Multi-level variational autoencoder: Learning disentangled representations from grouped observations,” 2017. 7
- [15] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, and K. Konolige, “Using simulation and domain adaptation to improve efficiency of deep robotic grasping,” 2017. 45
- [16] A. Brock, T. Lim, J. Ritchie, and N. Weston, “Neural photo editing with introspective adversarial networks,” *arXiv preprint [arXiv:1609.07093](https://arxiv.org/abs/1609.07093)*, 2016. 9, 45

REFERENCES

- [17] J. Bruna, P. Sprechmann, and Y. LeCun, “Super-resolution with deep convolutional sufficient statistics,” *arXiv preprint arXiv:1511.05666*, 2015. 41, 46
- [18] L. Chen, S. Srivastava, Z. Duan, and C. Xu, “Deep cross-modal audio-visual generation,” pp. 349–357, 2017. 2
- [19] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2017. 41
- [20] Q. Chen, J. Xu, and V. Koltun, “Fast image processing with fully-convolutional networks,” in *IEEE International Conference on Computer Vision*, 2017, pp. 2516–2525. 2
- [21] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 2172–2180. 15, 45, 47, 73
- [22] X. Chen, J. Yu, S. Kong, Z. Wu, X. Fang, and L. Wen, “Towards quality advancement of underwater machine vision with generative adversarial networks,” 2017. 46
- [23] Z. Cheng, Q. Yang, and B. Sheng, “Deep colorization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 415–423. 40, 44
- [24] B. Cheung, J. A. Livezey, A. K. Bansal, and B. A. Olshausen, “Discovering hidden factors of variation in deep networks,” *arXiv preprint arXiv:1412.6583*, 2014. 70, 72, 80, 86
- [25] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, “Generating multi-label discrete electronic health records using generative adversarial networks,” 2017. 41

REFERENCES

- [26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223. 54
- [27] K. A. De Jong, *Evolutionary computation: a unified approach*. MIT press, 2006. 18
- [28] A. Deshpande, J. Lu, M. C. Yeh, M. J. Chong, and D. Forsyth, “Learning diverse image colorization,” 2016. 5
- [29] C. Ding, C. Xu, and D. Tao, “Multi-task pose-invariant face recognition,” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 980–993, 2015. 86
- [30] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” *Computer Science*, 2014. 45
- [31] C. Donahue, A. Balsubramani, J. McAuley, and Z. C. Lipton, “Semantically decomposing the latent spaces of generative adversarial networks,” 2017. 2
- [32] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199. 7, 57
- [33] —, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016. 40, 44
- [34] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 658–666. 41, 46
- [35] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766. 45

REFERENCES

- [36] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, “Learning to generate chairs with convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1538–1546. 7, 72, 73, 82
- [37] A. E. Eiben, J. E. Smith *et al.*, *Introduction to evolutionary computing*. Springer, 2003, vol. 53. 18
- [38] A. E. Eiben and J. Smith, “From evolutionary computation to the evolution of things,” *Nature*, vol. 521, no. 7553, p. 476, 2015. 18
- [39] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658. 44
- [40] D. Eigen, D. Krishnan, and R. Fergus, “Restoring an image taken through a window covered with dirt or rain,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 633–640. 44
- [41] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in neural information processing systems*, 2014, pp. 2366–2374. 44
- [42] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006. 40
- [43] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013. 44
- [44] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 64–72. 15

REFERENCES

- [45] T. C. Fu, Y. C. Liu, W. C. Chiu, S. D. Wang, and Y. C. F. Wang, “Learning cross-domain disentangled deep representation with supervision from a single domain,” 2017. 7
- [46] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017. 40, 44
- [47] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin, “Triangle generative adversarial networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 15
- [48] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by back-propagation,” pp. 1180–1189, 2014. 45
- [49] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *arXiv preprint arXiv:1508.06576*, 2015. 46
- [50] J. Gauthier, “Conditional generative adversarial nets for convolutional face generation,” *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, vol. 2014, 2014. 73
- [51] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. 7
- [52] J. Glover, “Modeling documents with generative adversarial networks,” 2016. 73
- [53] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” *arXiv preprint arXiv:1701.00160*, 2016. 4, 7, 17
- [54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680. 3, 4, 8, 15, 17, 18, 21, 22, 40, 44, 73, 76, 92

REFERENCES

- [55] P. Grnarova, K. Y. Levy, A. Lucchi, T. Hofmann, and A. Krause, “An online learning approach to generative adversarial networks,” 2017. 73
- [56] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-pie,” *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010. 79, 86
- [57] G. L. Guimaraes, B. Sanchezlengeling, C. Outeiral, P. L. C. Farias, and A. Aspurguzik, “Objective-reinforced generative adversarial networks (organ) for sequence generation models,” 2017. 73
- [58] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 16, 17, 23, 29, 35, 93
- [59] K. Hausman, Y. Chebotar, S. Schaal, G. Sukhatme, and J. Lim, “Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets,” 2017. 70
- [60] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” 2017. 9
- [61] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012. 18
- [62] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming auto-encoders,” in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 44–51. 72
- [63] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006. 7
- [64] R. D. Hjelm, A. P. Jacob, T. Che, K. Cho, and Y. Bengio, “Boundary-seeking generative adversarial networks,” 2017. 9

REFERENCES

- [65] J. Hoffman, E. Tzeng, T. Park, J. Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, “Cycada: Cycle-consistent adversarial domain adaptation,” 2017. 44
- [66] F.-H. Hsu, *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press, 2002. 15
- [67] Z. Hu, Z. Yang, R. Salakhutdinov, and E. P. Xing, “On unifying deep generative models,” 2017. 46
- [68] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arXiv preprint arXiv:1611.07004*, 2016. 9, 41, 45, 52, 54, 55, 57, 64, 67
- [69] —, “Image-to-image translation with conditional adversarial networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 15
- [70] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711. 2, 41, 44, 46, 57
- [71] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, “Category-specific object reconstruction from a single image,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1966–1974. 70
- [72] M. W. Khan, “A survey: Image segmentation techniques,” *International Journal of Future Computer and Communication*, vol. 3, no. 2, p. 89, 2014. 40
- [73] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 06–11 Aug 2017, pp. 1857–1865. 4, 41, 45
- [74] T. Kim, B. Kim, M. Cha, and J. Kim, “Unsupervised visual attribute transfer with reconfigurable generative adversarial networks,” 2017. 9

REFERENCES

- [75] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 55, 80
- [76] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013. 57, 72
- [77] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, “On convergence and stability of gans,” 2017. 5
- [78] S. Kohl, D. Bonekamp, H. P. Schlemmer, K. Yaqubi, M. Hohenfellner, B. Hadaschik, J. P. Radtke, and K. Maierhein, “Adversarial networks for the detection of aggressive prostate cancer,” 2017. 2
- [79] J. Kos, I. Fischer, and D. Song, “Adversarial examples for generative models,” 2017. 9
- [80] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, “Gagan: Geometry-aware generative adversarial networks,” 2017. 72
- [81] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009. 28
- [82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 44
- [83] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, “Deep convolutional inverse graphics network,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2539–2547. 70, 72, 82, 83
- [84] A. Lamb, V. Dumoulin, and A. Courville, “Discriminative regularization for generative models,” 2016. 5
- [85] S. Lander and Y. Shang, “Evoae—a new evolutionary method for training autoencoders for deep learning networks,” in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 2. IEEE, 2015, pp. 790–795. 18

REFERENCES

- [86] C. Lassner, G. Pons-Moll, and P. V. Gehler, “A generative model of people in clothing,” 2017. 2
- [87] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. 7
- [88] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” *arXiv preprint arXiv:1609.04802*, 2016. 9, 41, 44, 45, 46, 57
- [89] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, “Alice: Towards understanding adversarial learning for joint distribution matching,” 2017. 46
- [90] J. Li, A. Madry, J. Peebles, and L. Schmidt, “Towards understanding the dynamics of generative adversarial networks,” 2017. 5
- [91] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, “Perceptual generative adversarial networks for small object detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 45
- [92] Y. Li, S. Liu, J. Yang, and M.-H. Yang, “Generative face completion,” *arXiv preprint arXiv:1704.05838*, 2017. 44
- [93] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, “Video generation from text,” 2017. 15
- [94] X. Liang, H. Zhang, and E. P. Xing, “Generative semantic manipulation with contrasting gan,” 2017. 40
- [95] K. Lin, D. Li, X. He, Z. Zhang, and M. T. Sun, “Adversarial ranking for language generation,” 2017. 73
- [96] T. Liu, D. Tao, M. Song, and S. J. Maybank, “Algorithm-dependent generalization bounds for multi-task learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 227–241, February 2017. 72

REFERENCES

- [97] Y. X. Liu, A. Gupta, P. Abbeel, and S. Levine, “Imitation from observation: Learning to imitate behaviors from raw video via context translation,” 2017. 70
- [98] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3730–3738. 28
- [99] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. 44
- [100] W. Lotter, G. Kreiman, and D. Cox, “Unsupervised learning of visual structure using predictive generative networks,” *arXiv preprint arXiv:1511.06380*, 2015. 9, 45, 47
- [101] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra, “Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 15
- [102] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, “Natural image colorization,” in *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 2007, pp. 309–320. 40
- [103] L. Ma, Q. Sun, X. Jia, B. Schiele, T. Tuytelaars, and L. Van Gool, “Pose guided person image generation,” 2017. 70
- [104] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 6, 15, 17, 23, 24, 92
- [105] M. Mardani, E. Gong, J. Y. Cheng, S. Vasanawala, G. Zaharchuk, M. Alley, N. Thakur, S. Han, W. Dally, and J. M. Pauly, “Deep generative adversarial networks for compressed sensing automates mri,” 2017. 9

REFERENCES

- [106] L. Mescheder, S. Nowozin, and A. Geiger, “The numerics of gans,” 2017. 5
- [107] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 29
- [108] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, A. Navruzian, N. Duffy, and B. Hodjat, “Evolving deep neural networks,” *arXiv preprint arXiv:1703.00548*, 2017. 18
- [109] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014. 40
- [110] T. M. Moerland, J. Broekens, and C. M. Jonker, “Learning multimodal transition dynamics for model-based reinforcement learning,” 2017. 41
- [111] V. Nagarajan and J. Z. Kolter, “Gradient descent gan optimization is locally stable,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 21, 26
- [112] K. Nasrollahi and T. B. Moeslund, “Super-resolution: a comprehensive survey,” *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014. 2, 40
- [113] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, “Plug & play generative networks: Conditional iterative generation of images in latent space,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 15
- [114] T. D. Nguyen, T. Le, H. Vu, and D. Phung, “Dual discriminator generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. 6, 15
- [115] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528. 44

REFERENCES

- [116] A. Odena, “Semi-supervised learning with generative adversarial networks,” 2016. 11
- [117] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” 2016. 5
- [118] A. Osokin, A. Chessel, R. E. C. Salas, and F. Vaggi, “Gans for biological image synthesis,” 2017. 9
- [119] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, “Transformation-grounded image generation network for novel 3d view synthesis,” *arXiv preprint arXiv:1703.02921*, 2017. 46
- [120] S. Pascual, A. Bonafonte, and J. Serr, “Segan: Speech enhancement generative adversarial network,” 2017. 15
- [121] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544. 9, 40, 41, 44, 45, 48, 54, 57, 61, 62
- [122] X. Peng and K. Saenko, “Synthetic to real adaptation with generative correlation alignment networks,” 2017. 70
- [123] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, “Generative adversarial perturbations,” 2017. 45
- [124] Y. Pu, W. Wang, R. Henao, L. Chen, Z. Gan, C. Li, and L. Carin, “Adversarial symmetric variational autoencoder,” 2017. 46
- [125] C. Qin, C.-C. Chang, and Y.-P. Chiu, “A novel joint data-hiding and compression scheme based on smvq and image inpainting,” *IEEE transactions on image processing*, vol. 23, no. 3, pp. 969–978, 2014. 44
- [126] H. Quan, D. N. Tu, T. Le, and D. Phung, “Multi-generator generative adversarial nets,” 2017. 17

REFERENCES

- [127] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 6, 15, 16, 17, 28, 52, 73, 93
- [128] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. Le, and A. Kurakin, “Large-scale evolution of image classifiers,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 2902–2911. 18
- [129] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 3, 2016. 73
- [130] A. Royer, A. Kolesnikov, and C. H. Lampert, “Probabilistic image colorization,” 2017. 40
- [131] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988. 44
- [132] T. Ruzic and A. Pizurica, “Context-aware patch-based image inpainting using markov random field modeling,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 444–456, 2015. 44
- [133] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 2234–2242. 17, 32
- [134] T. Salimans, J. Ho, X. Chen, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017. 18
- [135] S. Santurkar, L. Schmidt, and A. Mdry, “A classification-based perspective on gan distributions,” 2017. 5

REFERENCES

- [136] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on image processing*, vol. 15, no. 2, pp. 430–444, 2006. 57
- [137] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, 2016. 40
- [138] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, “Style transfer from non-parallel text by cross-alignment,” 2017. 2
- [139] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014. 41, 48, 60
- [140] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, “Learning to generate images with perceptual similarity metrics,” *Computer Science*, 2016. 41
- [141] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, “Vegan: Reducing mode collapse in gans using implicit variational learning,” 2017. 5
- [142] M. Tatarchenko, A. Dosovitskiy, and T. Brox, “Multi-view 3d models from single images with a convolutional network,” in *European Conference on Computer Vision*. Springer, 2016, pp. 322–337. 44, 57, 72, 80, 82, 83
- [143] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” in *9th ISCA Speech Synthesis Workshop*, pp. 125–125. 15
- [144] R. Vedantam, I. Fischer, J. Huang, and K. Murphy, “Generative models of visually grounded imagination,” 2017. 73
- [145] S. Vicente, J. Carreira, L. Agapito, and J. Batista, “Reconstructing pascal voc,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 41–48. 70

REFERENCES

- [146] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in *Advances In Neural Information Processing Systems (NIPS)*, 2016, pp. 613–621. 15
- [147] C. Vondrick and A. Torralba, “Generating the future with adversarial transformers,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 15
- [148] ———, “Generating the future with adversarial transformers,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 15
- [149] J. Walker, K. Marino, A. Gupta, and M. Hebert, “The pose knows: Video forecasting by generating pose futures,” in *IEEE International Conference on Computer Vision*, 2017, pp. 3352–3361. 2
- [150] C. Wang, H. Zheng, Z. Yu, Z. Zheng, Z. Gu, and B. Zheng, “Discriminative region proposal adversarial networks for high-quality image-to-image translation,” 2017. 40
- [151] C. Wang, Y. Zeng, L. Simon, I. Kakadiaris, D. Samaras, and N. Paragios, “Viewpoint invariant 3d landmark model inference from monocular 2d images using higher-order priors,” in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 319–326. 70
- [152] C. Wang, C. Wang, C. Xu, and D. Tao, “Tag disentangled generative adversarial networks for object image re-rendering,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2901–2907. 4, 13, 15
- [153] C. Wang, C. Xu, C. Wang, and D. Tao, “Perceptual adversarial networks for image-to-image transformation,” *IEEE Transactions on Image Processing*, 2018. 13
- [154] C. Wang, C. Xu, X. Yao, and D. Tao, “Evolutionary generative adversarial networks,” *arXiv preprint arXiv:1803.00657*, 2018. 12

REFERENCES

- [155] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, “Irgan: A minimax game for unifying generative and discriminative information retrieval models,” 2017. 9
- [156] R. Wang, A. Cully, H. J. Chang, and Y. Demiris, “Magan: Margin adaptation for generative adversarial networks,” 2017. 9
- [157] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” *arXiv preprint arXiv:1711.11585*, 2017. 41
- [158] Y. Wang, C. Xu, J. Qiu, C. Xu, and D. Tao, “Towards evolutionary compression,” *arXiv preprint arXiv:1707.08005*, 2017. 18
- [159] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, March 2002. 57
- [160] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 57
- [161] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, “Texturegan: Controlling deep image synthesis with texture patches,” 2017. 73
- [162] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha, “Wasserstein learning of deep generative point process models,” 2017. 6
- [163] T. Xiao, J. Hong, and J. Ma, “Dna-gan: Learning disentangled representations from multi-attribute images,” 2017. 7
- [164] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403. 54
- [165] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo, “Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks,” 2017. 15

REFERENCES

- [166] C. Xu, D. Tao, and C. Xu, “Multi-view intact space learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2531–2544, 2015. 71
- [167] Q. Xu, Z. Qin, and T. Wan, “Generative cooperative net for image generation and data augmentation,” 2017. 1
- [168] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, “Weakly-supervised disentangling with recurrent transformations for 3d view synthesis,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1099–1107. 44, 72, 82
- [169] L. C. Yang, S. Y. Chou, and Y. H. Yang, “Midinet: A convolutional generative adversarial network for symbolic-domain music generation,” 2017. 6
- [170] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Joint rain detection and removal via iterative region dependent multi-task learning,” *arXiv preprint arXiv:1609.07769*, 2016. 44
- [171] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999. 18
- [172] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 41, 45
- [173] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, “Optimizing deep learning hyper-parameters through an evolutionary algorithm,” in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*. ACM, 2015, p. 4. 18
- [174] A. Yu and K. Grauman, “Fine-grained visual comparisons with local learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 192–199. 54
- [175] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop,” *arXiv preprint arXiv:1506.03365*, 2015. 28, 35

REFERENCES

- [176] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 15
- [177] H. Zhang, V. M. Patel, B. S. Riggan, and S. Hu, “Generative adversarial network-based synthesis of visible faces from polarimetric thermal faces,” 2017. 73
- [178] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” *arXiv preprint arXiv:1701.05957*, 2017. xiv, 9, 41, 44, 45, 46, 49, 54, 57, 60, 64
- [179] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *European Conference on Computer Vision*. Springer, 2016, pp. 649–666. 40, 48
- [180] Y. Zhang, Z. Gan, and L. Carin, “Generating text via adversarial training,” in *NIPS workshop on Adversarial Training*, 2016. 15
- [181] B. Zhao, X. Wu, Z. Q. Cheng, H. Liu, and J. Feng, “Multi-view image generation from a single-view,” 2017. 70
- [182] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 4, 6, 9, 15, 17, 45, 49
- [183] Z. Zhao, D. Dua, and S. Singh, “Generating natural adversarial examples,” 2017. 40
- [184] S. Zhou, T. Xiao, Y. Yang, D. Feng, Q. He, and W. He, “Genegan: Learning object transfiguration and attribute subspace from unpaired data,” 2017. 72
- [185] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613. 9, 45, 54

REFERENCES

- [186] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 15
- [187] —, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 41, 45
- [188] J. Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, “Toward multimodal image-to-image translation,” 2017. 41