Unsupervised image-to-image translation using generative models

by

Lei Luo

B.S., China University of Geosciences (Beijing), 2014

M.A., Kansas State University, 2017

M.S., Kansas State University, 2020

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

In recent years deep learning has achieved great success in various computer vision tasks, such as image classification and segmentation. Unsupervised image-to-image (I2I) translation, which models how to translate images from one domain to another without paired data, lacks systematic and thorough study. In this dissertation I illustrate the significance of studying unsupervised I2I translation, relevant theories, and propose potential approaches to addressing drawbacks and shortcomings in existing works. This dissertation introduces four new contributions in unsupervised I2I translation. The first contribution is the proposal of a unified framework for unsupervised I2I translation. The second contribution is to provide fine-grained control on I2I translation where current approaches fall short. The third contribution of this dissertation is cooperating a module for controlling shapes when translating certain type of images, which require preserving shapes after I2I translation. Lastly, this dissertation proposes a new I2I translation framework that learns to, in an unsupervised manner, only translate objects of interest and leave others unaltered.

The first contribution of this work is to address the open problem of multimodal unsupervised I2I translation using a generative adversarial network. Previous works, such as MUNIT and DRIT, are able to translate images among multiple domains, but they generate images of inferior quality and less diverse. Moreover, they require training $n(n-1)$ generators and n discriminators for learning to translate images among n domains. Therefore, I propose a simpler yet more effective framework for multimodal unsupervised I2I translation. The new approach only consists of a mapping network, a encode-decoder pair (generator), and a discriminator. The methods assume that the latent space can be decomposed into content and style sub-spaces by the encoder, where content space is deemed domain-invariant and

style space is domain-dependent. Unlike MUNIT and DRIT that simply sample style codes from a standard normal distribution when translating, I employ a mapping network to learn style codes of different domains. Translation is done through the decoder by keeping content codes and exchanging the style codes. To encourage diversity in translated images, I employ style regularizations and inject Guassian noise in the decoder. Extensive experiments show that the new framework is superior to or comparable to state-of-the-art baselines.

The second contribution of this dissertation is to add fine-grained control when performing I2I translation. The new framework first assumes that the latent space can be decomposed into content and style sub-spaces. Instead of naively exchanging style codes when translating, the framework uses an interpolator that guides the transformation and produces sequences of intermediate results under different strengths of transformation. Domain specific information, which might still exist in content code and generate inferior images if they are simply treated as domain-invariant, are excluded in our framework. We prove the key assumptions of our framework by establishing some theoretical foundations. Extensive experiments show that the translated images using the new framework are superior or comparable to state-of-the-field baselines.

This dissertation also proposes a new I2I translation framework that is shape-aware. Attribute transfer is more challenging when the source and target domain share different shapes, and this new model is able to preserve shape when transferring attributes. Compared to other state-of-art GANs-based image-to-image translation models, the new model is able to generate more visually appealing results while maintaining the quality of results from transfer learning.

The last part of this work tries to learn to only translate objects of interest and keep the background unaltered, which produces more visually pleasing results than other approaches. Previous works, such as CycleGAN, MUNIT, and StarGAN2 are able to translate images among multiple domains and generate diverse images, but they often introduce unwanted changes to the background. To improve this, I propose a simple yet effective attention-based

framework for unsupervised I2I translation. The framework not only translates solely objects of interests and leave the background unaltered, but also generates images for multiple domains simultaneously. Unlike recent studies on unsupervised I2I with attention mechanism that require ground truth for learning attention maps, the new approach learns attention maps in an unsupervised manner. Extensive experiments show that the new framework is superior to the state-of-the-art baselines.

Unsupervised image-to-image translation using generative models

by

Lei Luo

B.S., China University of Geosciences (Beijing), 2014

M.A., Kansas State University, 2017

M.S., Kansas State University, 2020

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor

William H. Hsu

Copyright

Abstract

In recent years deep learning has achieved great success in various computer vision tasks, such as image classification and segmentation. Unsupervised image-to-image (I2I) translation, which models how to translate images from one domain to another without paired data, lacks systematic and thorough study. In this dissertation I illustrate the significance of studying unsupervised I2I translation, relevant theories, and propose potential approaches to addressing drawbacks and shortcomings in existing works. This dissertation introduces four new contributions in unsupervised I2I translation. The first contribution is the proposal of a unified framework for unsupervised I2I translation. The second contribution is to provide fine-grained control on I2I translation where current approaches fall short. The third contribution of this dissertation is cooperating a module for controlling shapes when translating certain type of images, which require preserving shapes after I2I translation. Lastly, this dissertation proposes a new I2I translation framework that learns to, in an unsupervised manner, only translate objects of interest and leave others unaltered.

The first contribution of this work is to address the open problem of multimodal unsupervised I2I translation using a generative adversarial network. Previous works, such as MUNIT and DRIT, are able to translate images among multiple domains, but they generate images of inferior quality and less diverse. Moreover, they require training $n(n-1)$ generators and n discriminators for learning to translate images among n domains. Therefore, I propose a simpler yet more effective framework for multimodal unsupervised I2I translation. The new approach only consists of a mapping network, a encode-decoder pair (generator), and a discriminator. The methods assume that the latent space can be decomposed into content and style sub-spaces by the encoder, where content space is deemed domain-invariant and

style space is domain-dependent. Unlike MUNIT and DRIT that simply sample style codes from a standard normal distribution when translating, I employ a mapping network to learn style codes of different domains. Translation is done through the decoder by keeping content codes and exchanging the style codes. To encourage diversity in translated images, I employ style regularizations and inject Guassian noise in the decoder. Extensive experiments show that the new framework is superior to or comparable to state-of-the-art baselines.

The second contribution of this dissertation is to add fine-grained control when performing I2I translation. The new framework first assumes that the latent space can be decomposed into content and style sub-spaces. Instead of naively exchanging style codes when translating, the framework uses an interpolator that guides the transformation and produces sequences of intermediate results under different strengths of transformation. Domain specific information, which might still exist in content code and generate inferior images if they are simply treated as domain-invariant, are excluded in our framework. We prove the key assumptions of our framework by establishing some theoretical foundations. Extensive experiments show that the translated images using the new framework are superior or comparable to state-of-the-field baselines.

This dissertation also proposes a new I2I translation framework that is shape-aware. Attribute transfer is more challenging when the source and target domain share different shapes, and this new model is able to preserve shape when transferring attributes. Compared to other state-of-art GANs-based image-to-image translation models, the new model is able to generate more visually appealing results while maintaining the quality of results from transfer learning.

The last part of this work tries to learn to only translate objects of interest and keep the background unaltered, which produces more visually pleasing results than other approaches. Previous works, such as CycleGAN, MUNIT, and StarGAN2 are able to translate images among multiple domains and generate diverse images, but they often introduce unwanted changes to the background. To improve this, I propose a simple yet effective attention-based

framework for unsupervised I2I translation. The framework not only translates solely objects of interests and leave the background unaltered, but also generates images for multiple domains simultaneously. Unlike recent studies on unsupervised I2I with attention mechanism that require ground truth for learning attention maps, the new approach learns attention maps in an unsupervised manner. Extensive experiments show that the new framework is superior to the state-of-the-art baselines.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my family for supporting my study at Kansas State University. Being away from home for so long has not only been difficult for me, but for them as well.

I would also like to thank my major professor, Dr. William Hsu, for his tremendous support in not merely giving me the fish, more importantly teaching me how to fish and conduct research independently. He exemplified himself to me the high standards, academic rigor, research skills and mindset, which will certainly benefit me for years to come. He is selfless and always putting students' needs first, which made this long journey much more smooth. I couldn't thank him enough in such a short paragraph.

I would also like to express my thankfulness to Dr. Douglas Goodin for being an amazing advisor during my master's study in the Geography department. It wasn't easy during the early days when I came to the States, and Dr. Goodin has made the transition much easier. Furthermore, he introduced me to Dr. Hsu when I approached him and expressed my intention to do a Ph.D. study in Computer Science.

I would like to thank all my committee members, Dr. Mitchell Neilsen, Dr. Pascal Hitzler, and Dr. Douglas Goodin for their valuable time and comments on my work.

Last but not the least, I would like to thank all the people who have helped me along this journey, whose names aren't listed but will not be forgotten. I would also like to thank myself on always thinking positively and not giving up, and always want to be a better person in all aspects.

Chapter 1

Introduction

Chapter 1 introduces the problem of unsupervised image-to-image (I2I) translation and briefly summarizes techniques for solving it. I begin by defining the problem and addressing its significance. Then, classic and current approaches to unsupervised I2I translation will be reviewed.

1.1 Problem Statement

In recent years, computer vision (CV) has seen great success using deep learning due to improvements in parallel computing devices and deep learning libraries. Deep learning models outperform humans in various CV tasks like image classification[4–6] and object detection[7–11]. I2I translation, which has received less attention than other CV tasks, focuses on how images translate from one domain to another, so the translated images preserve the content of the source domain image but appear as an image of target domain. For example, Fig 1.1 shows examples of translating edge maps to shoes and handbags. Image-to-image translation is a broad body of sub-problems that can be categorized into multi-domain and multimodal

Figure 1.1: *Examples of translating edge maps to shoes and handbags.*

problems. In multi-domain I2I translation, models can translate images to multiple domains simultaneously as with a model that translates cat images to dog and lion images. Multimodal I2I translation shows that translation results are not deterministic and have variations. For example, a winter scene translated to summer scenes can differ because of the lighting, or the degree of a smile might differ when translating neutral faces to smiling ones.

I2I translation has great significance, and various computer vision tasks can benefit from it. Class imbalance issues in image classification is one such example. Class imbalance issues occur when one class has far fewer images than other classes. Deep learning models trained on imbalanced data sets tend to be biased and sensitive. One can use a I2I translation model to synthesize data for the smaller classes so that data sets are more balanced. Image super-resolution is another example of I2I translation, referring to making images with low-resolution into high-resolution images. Fig. 1.2 is an example of image super-resolution. This means that a low resolution image can be transmitted over the internet at lower cost than a high-resolution image; the receiver can then use super-resolution to make the image a high resolution image using a trained image super-resolution model.

I2I translation not only has wide practical uses but also constantly encourages new theoretical breakthroughs. I2I translation is closely related to generative models, which are

Figure 1.2: *An example of image super-resolution.*

roughly categorized into three types. The first type is autoencoders (AE) for studying data compression and reconstruction. Subsequent improvement on AE has introduced various regularizations on loss functions to learn disentangled representations of input data, which directly inspires the study of I2I translation. For example, varying the smile dimension in learned disentangled representation can turn neutral faces into smiling ones. Another generative model is flow-based, which uses the variable theorem and learns invertable mapping from image to image, a natural option for modelling I2I. Another type of generative model, generative adversarial networks (GANs)[12], shows impressive results for unsupervised learning and is vital to this research[13–18]. GANs consist of a generator and a discriminator, with the former producing fake data samples from a latent vector and the later attempting to distinguish fake samples from real ones. The generator can be an AE, a flow-based model, or a mix of the two, thus creating many options in designing generative models. No thorough study, however, provides a systematic guideline for choosing the right design. Therefore, in this research, I bridge this gap theoretically and empirically.

1.2 Unsupervised Image-to-Image Translation

Image-to-image (I2I) translation refers to translating images from one domain to another, one with different properties. An example is the task of turning images of cartoon sketches into real-life graphs. Many tasks in computer vision can be considered I2I translation: image inpainting[19], style and attribute transfer[20;21], and super-resolution[22], among others. Paired I2I transfer tasks require paired data sets that are costly to acquire; paired data sets make I2I relatively easy to solve, unlike unpaired data sets. Chen and Koltun translated paired images of semantic maps to photographic images using regression[23]. Isola et al. framed paired I2I translation tasks using conditional generative models[24].

Unsupervised I2I translation converts images from one domain to another without paired data supervision. Much success in unpaired I2I translation is due to the cycle consistency constraint, proposed in three earlier works: CycleGANs[25], DiscoGANs[26], and DualGANs[27]. Recent systems like MUNIT[28] and DRIT[29] were developed to perform multimodal I2I translation, which refers to producing images with the same content but in different contexts. For example, a winter scene could be translated into many different summer scenes depending on weather or lighting. StarGAN-V2[30] and ModularGANs[31] have been proposed to translate more than two domains.

Previous research shows that unsupervised I2I is feasible, but several shortcomings in the process must be addressed.

- Recent SOTA studies like MUNIT[28] and DRIT[29] can translate images into several domains, but they generate less diverse images with inferior quality. Moreover, they require training $n(n-1)$ generators and n discriminators for learning to translate images among n domains, which is computationally expensive.
- Most frameworks for unsupervised I2I translation shift images by simply keeping content latent codes and exchanging style latent codes, which generate inferior images.

Moreover, they often cannot control translation strength because they naively adopt the cycle consistency loss. As a result, only one translation can be produced, and generating possible intermediate translation results is not feasible.

- Existing studies mostly focus on I2I at the image level, meaning that the entire input image is translated. This introduces unwanted changes to the background and produces sub-optimal translation results.

To remedy the drawbacks of current methods of unsupervised I2I translation, I propose several approaches based on GANs and autoencoders, which I hypothesize will improve the quality of generated images according to current state-of-the-field metrics.

1.3 A Unified Framework for Unsupervised I2I Translation

MUNIT and DRIT have been recently suggested as frameworks for unsupervised I2I translation. MUNIT, or Multimodal UNsupervised Image-to-image Translation, and DRIT, Diverse Image-to-Image Translation via Disentangled Representations were used in two co-current studies that addressed the multimodal problem of unsupervised I2I translation. They use encoders to extract content and style codes from images and translate images by replacing the style codes with codes of different domains while keeping the content codes of the source domain. They so need a pair of encoders and decoders for each domain. As a result, they require training $n(n-1)$ generators and n discriminators for learning to translate images among n domains, which is computationally expensive. Moreover, they sample style codes from a standard normal distribution when translating, which leads to inferior translation.

Therefore, I propose a simplified, yet more effective, framework for unsupervised I2I translation. The framework consists of only one paired encoder-decoder and one discrim-

inator for learning to translate among all domains, while achieving better generalization than previous attempts at unsupervised translation. Instead of sampling style codes from a standard normal distribution, I used a mapping network to learn style information. With more generalization capability added by the mapping network, the translation results improved. MUNIT and DRIT also do not regularize the style and content codes. Both studies using MUNIT and DRIT assume that style codes are domain-specific and content codes are domain-invariant, an assumption that requires imposing regularization. Therefore, I added loss functions to impose constraints that show content codes are domain-invariant and style codes are diverse.

To summarize, the novel contributions for this research are the following:

- A new I2I translation framework that is simplified yet more effective than existing frameworks and reduces network size and training time;
- A novel framework with a mapping network to learn the styles of different domains, leading to better translation results;
- Several regularization techniques to show that content codes are domain-invariant and style codes are diverse;
- Results from extensive experiments that show the framework is superior or comparable to SOTA baselines.

1.4 Unsupervised I2I with Fine-grained Control on Latent Space

Previous studies using such translators as self-supervised CycleGAN, MUNIT, and contrastive GAN translate images by keeping content latent codes from one domain and ex-

changing the style latent codes with codes from another domain, thus generating images of inferior quality. Moreover, they depend on the cycle consistency constraint, so they cannot generate translation images in intermediate stages. Therefore, I included a module, the interpolator, which is placed between style codes of different domains. The interpolator takes two style codes and mixes them non-linearly. This allows control of the strength of translation by changing how much style code is taken from different domains. Unlike naively mixing two style codes linearly, the interpolator can produce smoother and more natural looking results. MUNIT and DRIT both operate assuming content codes of each domain are domian-invariant, but domain-specific information may still occur in content code. Without special care, generated images could still show traits of the source domain. Therefore, to keep content code domain-independent, I removed style-specific information in content code before translating images.

To sum up, fine-grained control over latent space manifests three ways: 1) latent codes can be deconstructed into content and style, much like DRIT[29] and MUNIT[28]; 2) instead of simply exchanging style codes, an interpolator, which is a neural network, guides their transformation; and 3) domain-specific information in content code is removed before translation for better results. This research offers the following novel contributions:

- A new framework with an embedded interpolator controls translation strength and generates smooth-looking translation results of intermediate stages;
- New techniques remove domain-specific information in content code;
- A simplified framework architecture based on MUNIT reduces the size and training time but achieves better translation.
- The framework appears superior or comparable to SOTA baselines after extensive experiments on publicly available data sets.

1.5 Shape-aware Generative Adversarial Networks for Attribute Transfer

Generative adversarial networks (GANs) have been successfully applied to transfer visual attributes in many domains, including that of human face images. This success is partly attributable to the facts that human faces have similar shapes and the positions of eyes, noses, and mouths are fixed among different people. Attribute transfer is more challenging when the source and target domain share different shapes. In this paper, we introduce a shape-aware GAN model that is able to preserve shape when transferring attributes, and propose its application to some real-world domains. Compared to other state-of-art GANs-based image-to-image translation models, the model I propose is able to generate more visually appealing results while maintaining the quality of results from transfer learning. The novel contributions are as follows:

- A novel shape-aware GANs model that is capable of multi-domain, multi-modal attribute transfer while maintaining the shape of source domain;
- Several strategies for stabilizing training of the proposed model.
- Experiments showed that the model is able to generate more visually satisfying results than recently proposed state-of-art baseline model while maintaining the quality of translated results.

1.6 Instance-level Unsupervised I2I with Self-supervised Learning

Most studies on I2I translation emphasize translating at the image-level, not the instance level. As a result, if the task is to translate horse images to zebra images, an image with a horse eating green grass can be translated into a fake zebra, but the grass would also change to yellow because most zebra images show yellow grass. In other words, the translated horse image shows unwanted changes to the background. In another scenario, images have multiple objects of interest like two or more horses, so frameworks based on image-level translation fail to accurately translate all objects at the same time. To achieve instance-level translation, then, a segmentation mask must post-process the translated results. Segmentation annotations are costly because they require pixel-level annotation. Therefore, I applied self-supervised methods to learn the segmentation mask while also learning to translate. Self-supervised approaches exploit labels and annotations from the input data itself, so it can be considered an unsupervised approach, unlike a supervised approach that requires annotations provided by humans. This part of the research has the following novel contributions:

- A novel framework for unsupervised I2I translation with an attention mechanism, which allows image translation at the instance level;
- A framework that allows unsupervised learning of attention maps, which does not require segmentation annotations;
- Our framework was superior the SOTA baselines, as demonstrated by extensive experiments using publicly available data sets.

1.7 Road Map

This dissertation is organized as follows:

Chapter 2 introduces theoretical background that serves as a foundation for the theory behind I2I translation. This includes autoencoders and their variants, flow-based models, and finally GANs.

Chapter 3 lays out a novel, simplified, yet more effective model for image translation based on GANs and AEs. This model improves on MUNIT and DRIT but overcomes several shortcomings. The proposed model simplifies the network architecture without compromising translation performance. It also provides several regularization techniques to solidify the assumption that content code is domain-invariant and style code is diverse. To further improve translation performance, style information from different domains was learned via a mapping network instead of samples from a standard Gaussian distribution. Test beds and evaluation metrics for evaluating the new model are also discussed, and the model is compared to SOTA baselines.

Chapter 4 offers a new model that addresses the effect of fine-grained control on translation results. To be specific, existing models perform translation by exchanging style codes from different domains. We could not, however, generate translation results of intermediate stages using existing models. Moreover, naively interpolating two translated images to make a pseudo intermediate image does not guarantee a smooth result. Therefore, our model was equipped with an interpolator, which also operates as a neural network, to guide the creation of intermediate stages. A commonly used assumption underlying I2I translation is that content codes are domain-invariant across multiple domains. However, content codes are not guaranteed to contain domain-specific information, which leads to results that look artificial. To address this problem, before translation begins, the new model removed domain-related information contained in content codes. Test beds and evaluation metrics were selected to

evaluate the new model, comparing it against previous SOTA baselines.

Chapter 5 proposes a shape-aware GAN model that is able to preserve the shape before and after translation. I employ the same idea of interpolator proposed in Chapter 4, so the model can translate images of intermediate results. Using an example of tomato leaves, the model shows promising results in successfully preserving shape at little cost of classification accuracy.

Chapter 6 offers a novel model to achieve instance-level translation. Previous methods mostly translate at image-level, which introduces unwanted changes to the background and is limited if multiple objects of interest appear in the original images. The new model achieved instance-level translation by using a self-supervised, attention mechanism. Some studies have attempted to use an attention mechanism, but the mechanism is often supervised and requires segmentation masks in the model, a costly technique. Moreover, previous research often used oversimplified models for dual-domain translation, which requires a large amount of training to model multiple domains. Therefore, our model used a unified framework that achieved instance-level I2I translation while also using multiple domains and producing diverse translation results. Test beds and evaluation metrics evaluated the new model, comparing it against previous SOTA baselines.

In Chapter 7, I review the claims made throughout this dissertation, summarize key finding, and provide ideas for future study.

Chapter 2

Literature Review

This chapter provides a review of recent literature on generative models with a focus on autoencoders and generative adversarial neural networks.

2.1 Autoencoders

The ability to learn useful representations of data with little or no supervision is challenging. Autoencoders (AEs[32]) could accomplish this goal. AEs learn mapping from high-dimensional observations to lower-dimensional representation space so that the original observations can be reconstructed (approximately) from lower-dimensional representations. Bengio et al.[33] proposed a set of meta-priors, which are general assumptions about or goals of AE such as disentanglement of explanatory factors and the concentration of data on low-dimensional manifolds. AEs can be grouped by meta-priors into four categories:

- Disentanglement: If data consists of independent factors of variations, disentanglement AEs can capture these factors in learned representations. Therefore, changing one

factor leads to only one variation in reconstructed data.

- Hierarchical organization of explanatory factors: This assumes data can be described by a hierarchy of increasingly abstract concepts. For example, images can be abstractly described using objects at different levels of granularity.

- Semi-supervised learning: Supervised and unsupervised learning share representations. and shared representations can then be used to synthesize data.

- Clustering: Assuming data has multi-category structures, structures can be captured with a latent mixture model where each mixture corresponds to one category, and its distribution models the factors of variations within that category, which leads to a representation within that cluster and enables controlled data synthesis.

2.1.1 Preliminaries

Variational autoencoders[34] are intended to learn disentangled, semantically meaningful, statistically independent and causal factors of variation in data. VAEs consist of two coupled but independent models: the encoder and decoder. The encoder models a mapping ($p_\theta(z|x)$) from data space to latent variables, and the decoder learns a mapping from latent variables to original data space. To turn the intractable posterior inference into tractable problems, we can introduce a parametric inference model $q_\phi(z|x)$, where ϕ are the parameters of this inference model. We optimize ϕ such that $p_\theta(z|x) \approx q_\phi(z|x)$. The distribution $q_\phi(z|x)$ can be parameterized using deep neural networks. The parameters ϕ can be the the weights and biases of the neural network.

$$(\mu, log\sigma) = \text{Encoder}_\phi(x) \tag{2.1}$$

$$q_\phi(z|x) = \mathcal{N}(z; \mu, \text{diag}(\sigma)) \tag{2.2}$$

The optimization objective of the variational autoencoder is the evidence of the lower bound,

abbreviated as ELBO. For any choice of inference model $q_\phi(z|x)$, we have

$$\begin{aligned} logp_\theta(x) &= \mathbb{E}_{q_\phi(z|x)}[logp_\theta(x)] \\ &= \mathbb{E}_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{p_\theta(z|x)}]] \\ &= \mathbb{E}_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}]] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}]]}_{=\mathcal{L}_{\theta,\phi}(x)} + \underbrace{\mathbb{E}_{q_\phi(z|x)}[log[\frac{q_\phi(z|x)}{p_\theta(z|x)}]]}_{=\mathcal{D}_{\mathcal{KL}}(q_\phi(z|x)||p_\theta(z|x))} \end{aligned} \tag{2.3}$$

The first term is the ELBO, and the second term is the Kullback-Leibler (KL) divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$, which is equal to or greater than 0. So we have the following inequality:

$$\begin{aligned} \mathcal{L}_{\theta,\phi}(x) &= logp_\theta(z|x) - \mathcal{D}_{\mathcal{KL}}(q_\phi(z|x)||p_\theta(z|x)) \\ &\leq logp_\theta(z|x). \end{aligned} \tag{2.4}$$

When we maximize the ELBO $\mathcal{L}_{\theta,\phi}(x)$ w.r.t θ and ϕ, we will maximize the marginal likelihood $p_\theta(x)$ and minimize the KL divergence so that $q_\phi(z|x)$ is closer to $logp_\theta(z|x)$.

Intuitively, we would attempt to optimize the ELBO using gradient descent. The back-propagation, however, through z is infeasible because it is randomly sampled from some distribution. We resolve this problem by the reparameterization trick. assuming we have continuous latent variables and a differentiable encoder and generative model, so the ELBO can be straightforwardly differentiated w.r.t. both θ and ϕ through a change of variables. This process is called the reparameterization trick. We first express the random variable $z \sim q_\phi(z|x)$ as some differentiable (and invertible) transformation of another random variable ϵ, given z and ϕ: $z = g(\epsilon, \phi, x)$, where the distribution of random variable ϵ is independent of x and ϕ. Then, we can rewrite the expectations in terms of ϵ:

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(z)]. \tag{2.5}$$

Under the reparameterization, we can replace the expectation for $q_\phi(z|x)$ with one for

$p(\epsilon)$:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_{q_\phi(z|x)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}]] \\ &= \mathbb{E}_{p(\epsilon)}[log[\frac{p_\theta(x,z)}{q_\phi(z|x)}]].\end{aligned} \tag{2.6}$$

After we replace z with $g(\epsilon, \phi, x)$, we can calculate the gradient for z and ϕ because they are not produced in a random sampling process.

Computation of ELBO

Computing ELBO requires computing the density, $log(q_\phi(z|x))$, which can be easy to compute with suitable $g(\cdot)$. If $g(\cdot)$ is an invertible function, the densities of ϵ and z are related by $logq_\phi(z|x) = logp(\epsilon) - logd_\phi(x, \epsilon)$, where the the second term is $log|\det\frac{\partial z}{\partial \epsilon}|$.

We usually use Gaussian distribution for factorizing the encoder, which means $q_\phi(z|x) = \mathcal{N}(z; \mu, \text{diag}(\sigma^2))$, where μ and σ are outputs of the encoder. After reparameterization, we can write:

$$\epsilon \sim \mathcal{N}(0, I), \tag{2.7}$$

$$(\mu, log\sigma) = \text{Encoder}_\phi(x), \tag{2.8}$$

$$z = \mu + \phi \odot \epsilon, \tag{2.9}$$

where $\odot$ is element-wise product. The Jacobian of the transformation from ϵ to z is $\frac{\partial z}{\partial \epsilon} = \text{diag}(\sigma)$, which is a diagonal matrix with the elements of σ on the diagonal. Then, the log determinant of the Jacobian is $\sum_i log\sigma_i$, so the posterior density can be written as

$$\begin{aligned}logq_\phi(z|x) &= logp(\epsilon) - logd_\phi(x, \epsilon) \\ &= \sum_i log\mathcal{N}(\epsilon_i; 0, 1) - log\sigma_i.\end{aligned} \tag{2.10}$$

2.1.2 VAE for Learning Disentangled Representations

Assuming the data are generated from independent factors of variations, VAEs with suitable regularization can learn disentangled representations. A classic approach to enforce some meta-priors on the latent representations is to regularize the posterior $q_\phi(z|x)$ or the aggregated posterior $q_\phi(z)$. Most recent work can be summarized into an objective in the form

$$\mathcal{L}_{VAE}(\theta, \phi) + \lambda_1 \mathbb{E}_{p(\hat{x})}[R_1(q_\phi(z|x))] + \lambda_2 R_1(q_\phi(z)), \tag{2.11}$$

where R_1 and R_2 are regularizers, and λ_1, $\lambda_2 > 0$ are the corresponding weights. The main idea behind several recent research on disentanglement augmented the $\mathcal{L}_{VAE}$ loss with regularizers that encourage disentanglement of the latent variable z.

$\boldsymbol{\beta - VAE}$[35] proposes to weigh the second term in Equation 2.11 with a coefficient $\beta > 1$, thus adding $\lambda_1 = \beta - 1 > 0$ to $\mathcal{L}_{VAE}(\theta, \phi)$:

$$\mathcal{L}_{\beta-VAE}(\theta, \phi) = \mathcal{L}_{VAE}(\theta, \phi) + \lambda_1 \mathbb{E}_{p(\hat{x})}[R_1(q_\phi(z|x))]. \tag{2.12}$$

By minimizing Equation 2.12, we encourage $q_\phi(z|x)$ to better match the factorized prior $q_\phi(x)$, which in turn constrains the implicit capacity of the latent representation $z \sim q_\phi(z|x)$ and encourages it to be factorized.

In **FactorVAE**[36], we observed that Equation 2.11 encouraged $q_\phi(z)$ to be factorized by penalizing the second term but simultaneously discouraged the latent code to be informative by penalizing the first term. To reinforce only the former effect, FactorVAE proposes to regularize $\mathcal{L}_{VAE}$ with the total correlation $\text{TC}(q_\phi(z) = D_{KL}(q_\phi(z)||\prod_j q_\phi(z_j))$ of $q_\phi(z)$, which is a popular measure of dependence for multiple random variables. The objective of FactorVAE is

$$\mathcal{L}_{FactorVAE}(\theta, \phi) = \mathcal{L}_{VAE}(\theta, \phi) + \lambda_2 \text{TC}(q_\phi(z), \tag{2.13}$$

where the last term is the total correlation.

$\boldsymbol{\beta - TCVAE}$[37] splits $\mathcal{D}_{\mathcal{KL}}(q_\phi(z)||p_\theta(z))$ into $D_{KL}(q_\phi(z)||\prod_j q_\phi(z_j))+\sum_{j=1}^m D_{KL}(q_\phi(z_j)||p(z_j))$, penalizing each term individually.

$$\mathcal{L}_{\beta-TCVAE}(\theta,\phi) = \mathcal{L}_{VAE}(\theta,\phi) + \lambda_1 \text{TC}(q_\phi(z) + \lambda_2 \sum_j D_{KL}(q_\phi(z_j)||p(z_j)), \tag{2.14}$$

where λ_1 and λ_2 are set to 0 by default, leading to the same objective as FactorVAE. Unlike FactorVAE, however, the TC term is estimated using importance sampling. **InfoVAE**[38] rewrites $\mathcal{L}_{VAE}(\theta,\phi)$ as

$$D_{KL}(q_\phi(z)||p(z)) + \mathbb{E}_{\hat{p}(x)} D_{KL}(q_\phi(x|z)||p_\theta(x|z)), \tag{2.15}$$

which encourages disentanglement. We can further reweight the first term in 2.15 and encourage mutual information between $z \sim q(z|x)$ and x by adding a regularizer proportional to $I_{q_\phi}(x;z)$ to 2.15. We can further rearrange the terms in the resulting objective, arriving at

$$\mathcal{L}_{VAE}(\theta,\phi) + \lambda_1 D_{KL}(q_\phi(z)||p(z)) + \lambda_2 \mathbb{E}_{\hat{p}(x)} D_{KL}(q_\phi(z|x)||p_\theta(z)). \tag{2.16}$$

For tractability, the last term in Equation 2.16 is replaced by other divergences, such as the Jensen-Shannon divergence, which is implemented as a GAN model.

DIP-VAE[39] It suggests matching the moments of the aggregated posterior $q_\phi(z)$ to a multivariate standard normal before $p(z)$ during optimization of $\mathcal{L}_{VAE}(\theta,\phi)$ to encourage disentanglement of the latent variables $z \sim q(z)$. It proposes to match the covariance of $q_\phi(z)$ and $N(0,I)$ by penalizing their L2 distance, leading to a disentangled inferred prior objective:

$$\mathcal{L}_{VAE}(\theta,\phi) + \lambda_1 \sum_{k\neq l} (Cov_{q_\phi(z)}[z])_{k,l}^2 + \lambda_2 \sum_k ((Cov_{q_\phi(z)}[z])_{k,k} - 1)^2. \tag{2.17}$$

For standard parametrization, $q_\phi(z|x) = N(\mu_\phi(x), \text{diag}(\sigma_\phi(x)))$, $Cov_{q_\phi(z)}[z] = \sum_{i=1}^N \text{diag}(\sigma_\phi(x_i)) + Cov_{\hat{p}(x)}[\mu_\phi(x)]$, $\sigma_\phi(x)$ only contributes to the diagonal of $Cov_{q_\phi(z)}[z]$.

HSIC-VAE[40] It leverage the Hilbert-Schmidt independence criterion (HSIC) to encourage

independence between groups of latent variables, as

$$\mathcal{L}_{VAE}(\theta,\phi) + \lambda \text{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2})), \tag{2.18}$$

where $z_G = \{z_k\}_{k\in G}$ is an estimator of HSIC. In addition to controlling independent relations of the latent variables, HSIC can be used to remove sensitive information, provided as labels with the training data, from latent representation by using the regularizer HSIC.

HFVAE[41] It hierarchically decomposes the D_{KL} term into a regularization term of the dependencies between groups of latent variables $\{G_k\}_{k=1}^{n_G}$ and regularization of the dependencies between the random variables in each group G_k. Re-weighting different regularization terms allows encouraging different degrees of intra- and inter-group disentanglement, leading to the following:

$$\begin{aligned}
\mathcal{L}_{\text{HFVAE}}(\theta,\phi) &= \mathcal{L}_{VAE} - \lambda_1 I_{q_\phi}(x;z) \\
&= +\lambda_2(-\mathbb{E}_{q_\phi(z)}[\log \frac{p(z)}{\prod_{G\in\boldsymbol{G}} p(z_G)}] + D_{KL}(q_\phi(z)||\prod_{G\in\boldsymbol{G}} q_\phi(z_G))) \\
&= +\lambda_3 \sum_{G\in\boldsymbol{G}} (-\mathbb{E}_{q_\phi(z_G)}[\log \frac{p(z_G)}{\prod_{k\in\boldsymbol{G}} p(z_k)}] + D_{KL}(q_\phi(z_G)||\prod_{k\in\boldsymbol{G}} q_\phi(z_k))),
\end{aligned} \tag{2.19}$$

where λ_1 controls the mutual information between the data and latent variables, and λ_2 and λ_3 determine the regularization of dependencies within groups by penalizing the corresponding total correlation.

VIB[42] It aims to derive a variational approximation of the information bottleneck objective, which targets learning a compact representation z of some random variable x that is maximally informative about some random variable y. The objective of VIB is

$$\mathcal{L}_{\text{VIB}}(\theta,\phi) = \mathcal{L}_{VAE}(\theta,\phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[D_{KL}(q_\phi(z|x)||p(z))] + \lambda_2 \text{TC}(q_\phi(z)). \tag{2.20}$$

We can derive a more tractable expression for 2.20 and establish a connection to dropout for particular choices of $p(z)$ and $q_\phi(z|x)$.

AAE[43] Adversarial Autoencoders (AAEs) turn a standard autoencoder into a generative

model by imposing a prior distribution $p(z)$ on the latent variables by penalizing some statistical divergence D_f between $p(z)$ and $q_\phi(z)$ using a GAN. Using the negative log-likelihood as reconstruction loss, the AAE objective can be written as

$$\mathcal{L}_{\text{AAE}}(\theta, \phi) = \mathbb{E}_{\hat{p}(x)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)]] + \lambda_2 D_f(q(z)||p(z)). \tag{2.21}$$

The encoder and decoder are deterministic, which means that $p(x|z)$ and $q(z|x)$ are replaced by decoder (D_θ) and encoder (E_ϕ), and the negative log-likelihood in 2.24 is replaced with the standard autoencoder loss. The advantage of implementing the regularizer $\lambda_2 D_f$ using a GAN is that any $p(z)$ we sample can be matched.

VFAE[44] (Variational Fair Autoencoders) assumes a likelihood of the form $p_\theta(x|z, s)$, where s models redundant latent factors like sensitive information. z models the remaining latent factors. Using an approximate posterior of the form $q_\phi(z|x, s)$ and imposing factorized prior $p(z)p(s)$ encourages independence of $z \sim q_\phi(z|x, s)$ from s. However, z might still contain information about s, in particular in supervised settings where z encodes label information y that might correlate with s and additional factors of variation z'. To mitigate this issue, VFAE proposes an MMD-based regularizer to $\mathcal{L}_{\text{VAE}}$, encouraging independence between $q(z|s = k)$ and $q(z|s = k')$.

$$\mathcal{L}_{\text{VFAE}}(\theta, \phi) = \mathcal{L}_{\text{VAE}} + \lambda_1 \sum_{l=2}^{K} \text{MMD}(q_\phi(z|s = l), q_\phi(z|s = 1)), \tag{2.22}$$

where $q_\phi(z|s = l) = \sum_{i:s^{(i)}=l} \frac{1}{|\{i:s^{(i)}=l\}|} q_\phi(z|x^{(i)}, s^{(i)})$. To reduce the computational complexity of the MMD, VFAE can use random Fourier features.

2.1.3 VAE for Learning Hierarchical Representations

PixelVAE[45] It uses a VAE with feed-forward convolutional encoder and decoder, combining the decoder with a (shallow) conditional PixelCNN[46] to predict output probabilities. It has a hierarchical encoder and decoder structure with several levels of latent variables, meaning

$p_\theta(x, z_1, z_2, ..., z_L) = p_\theta(x|z_1)p_\theta(z_1|z_2)...p_\theta(z_{L-1}|z_L)p(z_L)$. The variables are realized by a feed-forward convolutional network. This approach extracts high- and low-level features on one hand, allowing controlled generation of local and global structure but, on the other hand, results in better clustering of the codes according to classes for multi-class data.

VLAE[47] The second term in $\mathcal{L}_{\text{VAE}}$ of Equation 2.11 encourages the latent code $z \sim q_\phi(z|x)$ to store only the information that cannot be modeled locally by decoding distribution $p_\theta(x|z)$. To enforce latent codes $z \sim q_\phi(z|x)$ in storing meaningful information, VLAE adapts the structure of the decoding distribution $p_\theta(x|z)$, so it cannot model information we want z to store. For example, to encourage z to capture global high-level information while allowing $p_\theta(x|z)$ to model local information like texture, we used an autoregressive decoding distribution with a limited local receptive field that cannot model long-range spatial dependencies.

VQ-VAE[48] It realizes a VAE with discrete latent space structure using vector quantization (VQ-VAE). Each latent variable z_j is assumed to be a categorical random variable with K categories, and the approximate posterior $q_\phi(z_j|x)$ is assumed to be deterministic. Each category is associated with an embedding vector $e_k \in R^D$. The embedding operation induces an additional latent space dimension of size D. Let us say the latent representation z is an $M \times M \times 1$ feature map, the embedded latent representation $\tilde{z}$ is an $M \times M \times D$ feature map. The distribution $q_\phi(\tilde{z_j}|x)$ is implemented using a deterministic encoder network $E_\phi(x)$ with D-dimensional output, quantized w.r.t the embedding vectors $\{e_k\}_{k=1}^K$. In summary, we have

$$q_\phi(\tilde{z_j} = e_k|x) = \begin{cases} 1 & \text{if } k = \ argmin_l \left\| \mathbb{E}_\phi(x) - e_l \right\| \\ 0 & \text{otherwise} \end{cases}, \tag{2.23}$$

The embeddings e_k can be learned individually for each latent variable z_j or shared for the entire latent space. Assuming a uniform prior $p(z)$, the second term in $\mathcal{L}_{\text{VAE}}$ evaluates to $\log K$ because $q_\phi(z|x)$ is deterministic and can be discarded during optimization. The embedding vectors e_k, which do not receive gradients as a consequence of using a straight-through estimator, are updated as the mean of the encoded points $\mathbb{E}_\phi(x^{(i)})$ assigned to the

corresponding category k as in mini-batch k-means.

2.1.4 VAE for Semi-supervised Representations

For **Semi-supervised VAE**[49], latent codes can be divided into two parts: z and y, where y is label information. The inference model takes the form $q_\phi(z|y,x) = q_\phi(z|y,x)q_\phi(y|x)$, meaning there is a hierarchy between y and z. When labels are available, the inference model is conditioned on y, meaning $q_\phi(z|y,x)$. Without a label, the label is inferred from $q_\phi(z,y|x)$. This model thus effectively disentangles the latent code into two parts y and z and allows semi-supervised classification and controlled generation by holding one of the factors fixed and generating the other. This model can optionally be combined with an additional, unsupervised learning model to gain an additional level of hierarchy.

AAE[43] (Adversarial Autoencoders (AAEs)) turns a standard autoencoder into a generative model by imposing a prior distribution $p(z)$ on the latent variables by penalizing some statistical divergence D_f between $p(z)$ and $q_\phi(z)$ using a GAN. Using the negative log-likelihood as reconstruction loss, the AAE objective can be written as

$$\mathcal{L}_{\text{AAE}}(\theta,\phi) = \mathbb{E}_{\hat{p}(x)}[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)]] + \lambda_2 D_f(q(z)||p(z)). \tag{2.24}$$

The encoder and decoder are assumed to be deterministic, which means that $p(x|z)$ and $q(z|x)$ are replaced by decoder (D_θ) and encoder (E_ϕ), and the negative log-likelihood in Equation 2.24 is replaced with the standard autoencoder loss. The advantage of implementing the regularizer $\lambda_2 D_f$ using a GAN is that any $p(z)$ we can sample can be matched.

2.1.5 VAE for Unsupervised Learning

In **PixelGAN-AE**[50], if $p_\theta(x|z)$ is not too powerful (in the sense that it cannot model the data distribution unconditionally without using the latent code z), the term $p_\theta(x|z)$

and the reconstruction term have competing effects: a small amount of mutual information $I_{q_\phi}(x;z)$ makes reconstructing x^i from $q_\phi(z|x^{(i)})$ challenging for $p_\theta(x|z)$, leading to a large reconstruction error. Conversely, a small reconstruction error requires the code z to be informative and hence $I_{q_\phi}(x;z)$ to be large. In contrast, if the decoder is powerful, the mutual information and reconstruction terms can be minimized and are largely independent, which prevents the latent code from being informative and hence providing a useful representation. To prevent this, PixelGAN-AE drops the $I_{q_\phi}(x;z)$.

$$\mathcal{L}_{\text{PixelGAN-AE}}(\theta,\phi) = \mathcal{L}_{VAE}(\theta,\phi) - I_{q_\phi}(x;z). \tag{2.25}$$

JointVAE[51] equips the $\beta-$VAE framework with heterogeneous latent variable distributions by concatenating continuous latent variables z with discrete ones c for improved disentanglement of different types of latent factors. The corresponding approximate posterior is factorized as $q_\phi(c|x)q_\phi(z|x)$. The regularization strength λ_1 in the $\beta-$VAE objective is gradually increased during training, assigning different weights to the regularization term corresponding to the discrete and continuous random variables. Numerical results (based on visual inspection) show that the discrete latent variables naturally model discrete factors of variation like digit class in MNIST or garment type in Fashion-MNIST and hence disentangle such factors better than models with continuous latent variables only.

2.2 GANs for Unpaired Image-to-Image Translation

Ideally, generative models learn how data is distributed, thus allowing data synthesis from the learned distribution. Since the advent of GANs[12], generative models have achieved impressive results in tasks like image editing[52;53] and style transfer[21]. GANs learn the data distribution by approximating the similarity of distributions between the training data and the fake data produced by the learned model. GANs usually comprise a generator and a discriminator. The entire model learns by playing a minimax game: the generator tries to fool the discriminator by gradually generating realistic data samples, and the discriminator,

in turn, tries to distinguish real samples from fake ones. GANs have been improved in various ways. To produce more realistic samples, an architecture of stacked GANs has been proposed: the laplacian pyramid of GANs[54]; layered, recursive GANs[55]; progressive growing GANs[56]; and style-based GANs[20;21]. Several studies have attempted to solve the instability training of GANs using energy-based GANs[57], Wasserstein GANs[58], and boundary equilibrium GANs[59].

Generative models are designed to model and reproduce the statistical distribution of the training data, allowing the synthesis of data from the learned distribution. The key incentive behind GANs is estimating the underlying probability density or probability mass function of the observed data. GANs learn the probability distribution implicitly by computing the similarity of the distribution between the real training examples and the fake data generated by the learned model. After the model is trained well, it can be used to generate additional data with distribution similar to the real data.

Figure 2.1 shows the general architecture of GANs. The learning process is guided by a minmax game (See Equation 2.1) where the discriminator (D) increases the probability of classifying images as real when the images (x) are sampled from the real distribution ($p_{data}(x)$). The generator (G), however, tries to decrease the likelihood of identifying generated images as real when images come from the fake distribution ($p_z(z)$). As learning progresses, the discriminator gets better at classifying the data as real or not, and the generator becomes better at producing realistic data. Naturally, the generator can then be used to generate data when training examples are not sufficiently available.

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{x \in p_{data}(x)} [logD(x)] + \mathop{\mathbb{E}}_{z \in p_z(z)} [log(1 - D(G(z)))] \tag{2.26}$$

Image-to-image (I2I) translation refers to translating images from one domain with one set of properties to another with different properties. An example is the task of turning images of cartoon sketches into real-life graphs. Many tasks in computer vision can be posed as I2I translation, such as image inpainting[19], style and attribute transfer[20;21], and super-

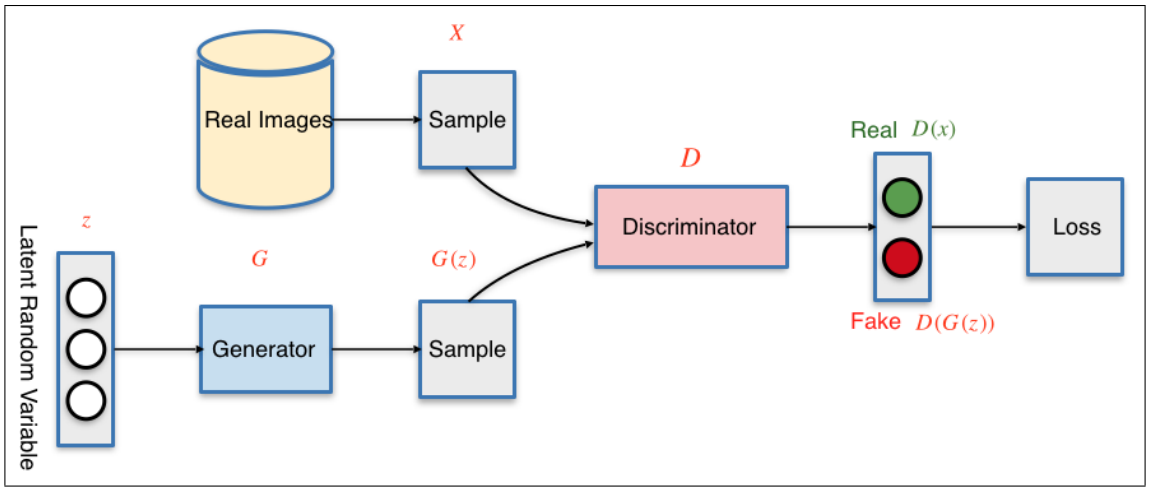


Figure 2.1: *Architecture of GANs*

resolution[22]. Paired I2I transfer tasks require costly paired data sets; translation using paired data sets are relatively easier to solve than unpaired data sets. Chen and Koltun translated paired images of semantic maps to photographic images using regression[23]. Isola et al. framed paired I2I translation tasks using conditional generative models[24]. We will review the literature on the more challenging unpaired I2I task where no paired data sets are available. Unpaired I2I translation can be categorized into two-domain I2I and multi-domain I2I, and each category can be divided into two sub-groups: single-modal output and multi-modal output. Single-modal output means that a framework can produce only a single output, and multi-modal output refers to variations on a single output. For example, a winter scene can be translated into different summer scenes with different lighting or weather conditions.

Two-domain single-modal I2I

Two-domain single-modal I2I refers to frameworks that can translate images between two domains resulting in only one translated image. We begin by reviewing some pioneering research that inspired unpaired I2I.

CycleGANs[16] learn a mapping G: $X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution from Y; CycleGANs use an adversarial loss expressed as

$$\mathcal{L}_{\text{CycleGANs}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))], \quad (2.27)$$

where G generates images $G(x)$ that look similar to images from domain Y , while D_Y distinguishes between translated samples $G(x)$ and real samples y. G minimizes this objective against an adversary D that maximizes it. CycleGANs introduce a similar adversarial loss for the mapping function G': $Y \rightarrow X$ as well as its discriminator D_X. CycleGANs introduce a cycle consistency loss to enforce $G'(G(X)) \approx X$, meaning for each image x from domain X, the image translation cycle should bring x back to the original image. The cycle consistency loss is calculated as

$$\mathcal{L}_{\text{CycleGANs}}(G, G') = \mathbb{E}_{y \sim p_{data}(y)}[||G(G'(y)) - y||_1] + \mathbb{E}_{x \sim p_{data}(x)}[||G'(G(x)) - x||_1]. \quad (2.28)$$

The full objective is

$$\mathcal{L}(G, G', X, Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(G', D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F), \quad (2.29)$$

where λ controls the relative importance of the two objectives. We solve for

$$G^*, G'^* = \arg \min_{G, G'} \max_{D_x, D_y} \mathcal{L}(G, G', X, Y). \quad (2.30)$$

U-GAT-IT[60] stands for unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. U-GAT-IT has an attention module that guides a model to focus on more important regions, distinguishing between source and target domains based on the attention map obtained by the auxiliary classifier. Moreover, the AdaLIN (adaptive layer-instance normalization) function helps our attention-guided model to flexibly control the amount of change in shape and texture by learned parameters depending on datasets. The translation model consists of an encoder E_s and an auxiliary classifier $\eta_s(x)$ trained to learn the weight of the k-th feature map for the source domain, w_s^k, by using global average pooling and global max pooling; i.e., $\eta_s(x) = \sigma(\sum_k w_s^k \sum_{ij} E_s^{k_{ij}}(x))$.

By exploiting w_s^k, we can calculate a set of domain specific attention feature maps $a_s(x) = w_s * E_s(x) = \{w_s^k * E_s^k(x) | 1 \leq k \leq n\}$, where n is the number of encoded feature maps. Residual blocks are equipped with AdaLIN whose parameters, γ and β, are dynamically computed by a fully connected layer from the attention map.

$$\text{AdaLIN}(a, \gamma, \beta) = \gamma(\rho * \hat{a_I} + (1 - \rho) * \hat{a_L}) + \beta \tag{2.31}$$

$$\hat{a_I} = \frac{a - \mu_I}{\sqrt{\sigma_I^2 + \epsilon}}, \ \hat{a_L} = \frac{a - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}}, \tag{2.32}$$

$$\rho \leftarrow clip_{[0,1]}(\rho - \tau \Delta \rho) \tag{2.33}$$

μ_I, μ_L are channel-wise, layer-wise mean, and σ_I, σ_L are standard deviations, γ and β are parameters generated by the fully connected layer, τ is the learning rate, and $\Delta\rho$ indicates the parameter update vector (e.g., the gradient) determined by the optimizer. A special loss function used in this research is CAM loss, which is the classification loss. Imposing this loss function shows the generator and discriminator where they must improve or what makes the most difference between the two domains in the current state.

UAIT[61] stands for unsupervised attention-guided image-to-image translation, where unsupervised attention mechanisms are introduced to jointly train with generators and discriminators. The proposed approach has two attention networks. One A_S extracts attention maps of images from the source domain, and another A_T extracts attention maps of images from the target domain. The translated image is evaluated by

$$s' = s_a \odot F_{S \to T}(s) + (1 - s_a) \odot s, \tag{2.34}$$

where $F_{S \to T}(s)$ translates source image s to the target domain T, and s_a is the attention map.

UAITs also embed attention mechanisms into the discriminator, which only considers attended regions. Simply using $s_a \odot s$ is problematic because real samples fed to the discriminator now depend on the initially-untrained attention map s_a. This leads to mode collapse if all networks in the GAN are trained jointly. To overcome this issue, UAITs first train

the discriminators on full images for 30 epochs, and then switch to masked images once the attention networks A_S and A_T have developed. Further, with a continuous attention map, the discriminator may receive fractional pixel values, which may be close to zero early in training. While the generator benefits from the ability to blend pixels at object boundaries, multiplying real images by these fractional values causes the discriminator to learn that mid gray is real. Thus, they provide a threshold for the learned attention map for the discriminator. The loss functions are common, which include adversarial loss and cycle consistency loss.

TIIT[62] stands for towards instance-level image-to-image translation. TIIT is based on MUNIT[28] and equipped with instance-level translation. MaskRCNN[63] is used to extract (multiple) instance(s) from images. Translation is done by swapping style codes but keeping content codes. The loss functions are similar to MUNIT, which include cycle consistency loss, adversarial loss, and latent codes reconstruction loss, but TIIT uses the global level and MUNIT the local instance level.

AGOT[64] stands for attention-GAN for object transfiguration in wild images. This framework also introduces an attention mechanism into an image-to-image translation framework. AGOT, however, is supervised; i.e. they train the attention network with segmentation annotations. Like UAIT, they express translated images as

$$s' = s_a \odot F_{S \to T}(s) + (1 - s_a) \odot s, \tag{2.35}$$

where $F_{S \to T}(s)$ translates source image s to the target domain T, and s_a is the attention map. The loss functions used to train the network are standard for other image-to-image translation frameworks. The loss functions are adversarial loss, cycle consistency loss, and a sparse loss to encourage the attention network to focus on a small region related to the object instead of the whole image. The sparse loss is the L1 loss of attention maps.

AGGAN[65] stands for attention-guided generative adversarial networks for unsupervised image-to-image translation. AGGANs embed an attention module inside the generator. Like

other image-to-image translation frameworks, AGGAN learns two attention maps, one from the source domain to the target domain and another in the reverse direction. For example, the generator $(G_{X\to Y}) : x \to [M_y, R_y, G_y]$ takes in source image x and outputs an attention map (M_y), content map (R_y), and translation results G_y. The attention maps define a per pixel intensity specifying to what extent each pixel in the content maps will contribute to the final rendered image. Translation results G_y can be obtained by $R_y * M_y + x * (1 - M_y)$. The generator only considers the attended area, but the discriminator looks at the entire region. Therefore, an attention mechanism is added to the discriminator, so it only focuses on the area of interest. The final loss function comprises attention-guided adversarial loss and cycle consistency loss. To reduce changes and constrain generators, we adopted pixel loss between the input images and the generated images. To prevent attention maps being saturated to 1, at which point, the attention-guided generator has no effect, generators perform a total variation regularization over attention maps.

MUNIT & DRIT[28;29] share similar ideas: the generator consists of a style encoder and a content encoder. The content encoder extracts content codes, which are assumed to be domain-invariant, and the style encoder outputs style codes that are domain-variant. When MUNIT and DRIT perform image translation, the style codes of different domains are exchanged, and the content codes are kept. Like other GANs-based models, they use a discriminator to distinguish images of different domains. The loss functions for training the model are adversarial losses for translated results, cycle-consistency loss, image reconstruction loss when decoded back from the latent codes, and latent code reconstruction loss. Image reconstruction loss and latent code reconstruction loss have not been discussed, so they will be illustrated here. We can reconstruct images using the decoder on the latent codes obtained from the encoders, so the image reconstruction loss is expressed as

$$L^x_{recon} = \|D(E_c(x), E_s(x)) - x\|_1 , \tag{2.36}$$

where E_c is the content encoder, and E_s is the style encoder. D is the decoder. Latent codes can be reconstructed from the translated images, and the latent code reconstruction loss is

expressed as

$$L^{c}_{recon} = \|c'_m - c_m\|_1 \,; \tag{2.37}$$

$$L^{s}_{recon} = \|s'_n - s_n\|_1 \,, \tag{2.38}$$

where c_m and s_n are content codes of x_m and style codes of x_n. Encoding the translated image x_{mn} produces (c'_m, s'_n).

StyleGANs[13] are based on the progressively growing gan[56]; both the generator and discriminator grow progressively, starting from a low resolution of 4×4 and adding new layers that model increasingly fine details as training progresses before achieving high resolution of 1024×1024. Several techniques can increase the quality of the final images. The first uses minibatch standard deviation, which solves one problem with gans; they have a tendency to capture only a subset of the variation found in training data. Their solution is to compute the standard deviation for each feature in each spatial location over the minibatch. Then, these estimates are averaged over all features and spatial locations to arrive at a single value. They replicate the value and concatenate it to all spatial locations and over the minibatch, yielding one additional feature map, which is inserted towards the end of the discriminator.

The second technique solves the problem of GANs being prone to escalating signal magnitudes as a result of unhealthy competition between the discriminator and the generator. Using a trivial $N(0, 1)$ initialization and then explicitly scaling the weights at runtime replaces carefully designed weight initialization. Another technique is to normalize the feature vector in each pixel to unit length in the generator after each convolution layer.

StyleGAN is based on a progressively growing gan, but it makes several improvements. The biggest change is removing the progressive grow training scheme. First, an 8-layer multilayer perceptron is used on the initial latent code z that produces an intermediate style code w. Then, w is injected into the generator using adaptive instance normalization (AdaIN)

operations after each convolution layer. The AdaIN operation is defined as

$$\text{AdaIN}(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}, \tag{2.39}$$

where each feature map x_i is normalized separately and then scaled and biased using the corresponding scalar components from style y. Finally, the generator is provided a direct means to generate stochastic detail by introducing explicit noise inputs. These are single-channel images consisting of uncorrelated Gaussian noise. We fed a dedicated noise image to each layer of the synthesis network.

2.3 Flow-Based Model for I2I Translation

Autoencoders and generative adversarial networks do not explicitly model the probability density function of data. Flow-based models try to do so using normalizing flows, which are a powerful statistical tool for density estimation. Normalizing flows[66] transform a simple distribution into a complex one by applying a sequence of invertible transformation functions. Flowing through a chain of transformations, such models repeatedly substitute one variable for a new one according to the change of variables theorem, eventually obtaining a probability distribution of the final target variable. Mathematics requires in normalizing flows that all transformation functions be easily invertible and their Jacobian determinant easy to determine. There are three important models with normalizing flows. Dinh et al.[67] used RealNVP (real-valued non-volume preserving) to implement a normalizing flow by stacking a sequence of invertible bijective transformation functions. Dinh et al.[68] also used a predecessor model of RealNVP called NICE (non-linear independent component estimation), whose transformation is the affine coupling layer without the scale term, known as an additive coupling layer. Kingma and Dharimal[69] extended previous reversible generative models, NICE and RealNVP, and simplified the architecture by replacing the reverse permutation operation on the channel ordering with invertible 1x1 convolutions. The details of these models are reviewed in detail in Chapter 2.

Image-to-image translation would be easy if we could find the hidden factors behind the images that control whether the image is, for instance, a cat or dog. When we discuss GANs and VAEs, latent codes z are important intermediaries between source images and translated images. However, neither of them explicitly learns the probability density function because it is too difficult. Flow-based models can find such disentangled factors, so we can manipulate them to turn, for example, cat images into dog images.

Flow-based generative models are based on the change of variable theorem, which is briefly illustrated here. Given a random variable z and its known probability density function $z \sim \pi(z)$, we construct a new random variable using a 1-1 mapping function $x = f(z)$, whose inverse function $z = f^{-1}(z)$ exists. We can infer $p(x)$ using

$$\int p(x)dx = \int \pi(z)dz = 1 \tag{2.40}$$

$$p(x) = \pi(z)|\frac{dz}{dx}| = \pi(f^{-1}(x))|\frac{df^{-1}}{dx}| = \pi(f^{-1}(x))|(f^{-1})'(x)|. \tag{2.41}$$

The multivariable version has a similar format:

$$z \sim \pi(x), x = f(z), z = f^{-1}(x) \tag{2.42}$$

$$p(x) = \pi(z)|\det\frac{dz}{dx}| = \pi(f^{-1}(x))|\det\frac{df^{-1}}{dx}| \tag{2.43}$$

where $\det\frac{dz}{dx}$ is the Jacobian determinant of the function f. A normalizing flow transforms a simple distribution into a complex one by applying a sequence of invertible transformation functions. Flowing through a chain of transformations, we repeatedly substitute each variable for a new one following the change of variables theorem and eventually obtaining a probability distribution of the final target variable. To illustrate, we have $x = z_i = f_i(z_{i-1}), z_{i-1} =$

$f_{i-1}(z_{i-2}), ..., z_1 = f_1(z_0)$; then we have

$$z_{i-1} \sim p_{i-1}(z_{z-1}) \tag{2.44}$$

$$z_i = f_i(z_{i-1}) \rightarrow z_{i-1} = f_i^{-1}(z_i) \tag{2.45}$$

$$p_i(z_i) = p_{i-1}(f_i^{-1}(z_i))|\det\frac{df_i^{-1}}{dz_i}| \tag{2.46}$$

$$= p_{i-1}(z_{i-1})|\det(\frac{df_i}{dz_{i-1}})^{-1}| \quad \text{according to inverse function theorem.} \tag{2.47}$$

$$= p_{i-1}(z_{i-1})|\det\frac{df_i}{dz_{i-1}}|^{-1} \quad \text{according to property of the Jacobian invertible function, then} \tag{2.48}$$

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log|\det\frac{df_i}{dz_{i-1}}| \tag{2.49}$$

$$= \log p_{i-2}(z_{i-2}) - \log|\det\frac{df_{i-1}}{dz_{i-2}}| - \log|\det\frac{df_i}{dz_{i-1}}| \tag{2.50}$$

$$= ... = \log p_0(z_0) - \sum_{i=1}^{K}\log|\det\frac{df_i}{dz_{i-1}}| \quad p_0(z_0) \text{ is known to us.} \tag{2.51}$$

The path traversed by the random variables $z_i = f_i(z_{i-1})$ is the flow, and the full chain formed by successive distributions p_i is called a normalizing flow. Required by the computation in the equation, a transformation function f_i should be easily invertible and its Jacobian determinant easy to compute.

RealNVP[67] stands for real-valued non-volume preserving, which implements a normalizing flow by stacking a sequence of invertible bijective transformation functions. In each bijection $f : x \rightarrow y$, known as affine coupling layer, the input dimensions are split into two parts. The first d dimensions remain the same, and the rest undergo an affine transformation. Both the scale and shift parameters are functions of the first d dimensions, so we have

$$y_{1:d} = x_{1:d} \tag{2.52}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}), \tag{2.53}$$

where $s(.)$ and $t(.)$ are scale and translation functions, and both map. The $\odot$ operation is the element-wise product. The mapping function is easily invertible, and its Jacobian

determinant is easy to compute: $\exp(\sum_{j=1}^{D-d} s(x_{1:d})_j)$. Furthermore, RealNVP can work in a multi-scale architecture to build a more efficient model for larger inputs. The multi-scale architecture applies several sampling operations to normal affine layers, including spatial checkerboard pattern masking, squeezing, and channel-wise masking.

NICE[68] stands for non-linear independent component estimation, a predecessor of RealNVP. The transformation in NICE is the affine coupling layer without the scale term, known as an additive coupling layer.

$$y_{1:d} = x_{1:d} \tag{2.54}$$

$$y_{d+1:D} = x_{d+1:D} + m(x_{1:d}). \tag{2.55}$$

GLOW[69] stands for generative flow with invertible 1x1 convolutions. The GLOW model extends the previous reversible generative models, NICE and RealNVP, by replacing the reverse permutation operation on the channel ordering with invertible 1×1 convolutions. The flow module in GLOW consists of three sub-modules. It begins with activation normalization, which performs an affine transformation using one scale and bias parameter per channel. The parameters are trainable but initialized, so the first minibatch of data has a mean of 0 and standard deviation of 1 after normalization. The second step in the flow module is invertible 1×1 convolution, which is a generalization of any permutation of the channel ordering. For instance, we have an invertible 1×1 convolution of an input $hw \times c$ tensor h with a weight matrix W of size $c \times c$. The output is a $h \times w \times c$ tensor, labeled as $f = \text{conv2d(h;W)}$. We must compute the Jacobian determinant, $\det\frac{df}{dh}$, to apply the change of variable rule. The input and output after 1×1 convolution can be viewed as a matrix of $h \times w$. Each entry x_{ij} in h is a vector of c channels, and each entry is multiplied by the weight matrix W to obtain the corresponding entry y_{ij} in the output matrix. The derivative of each entry is $\frac{dx_{ij}W}{dx_{ij}} = W$, and there are $h \times w$ such entries in total, so we have

$$\log|\det\frac{d\text{conv2d}(h;W)}{dh}| = \log(|\text{detW}|^{hw}) = hw\log|\text{det}W|. \tag{2.56}$$

The inverse 1×1 convolution depends on the inverse matrix of W^{-1}. The weight matrix is relatively small, so the amount of computation for the determinant and inversion is not huge. The last step in the flow module is the affine coupling layer, which shares the same design as RealNVP.

2.4 Summary

I2I translation is of great significance, and various computer vision tasks can be posed as and benefit from it. Class imbalance issues in image classification is one such example. Class imbalance issues occur when one class of images have far fewer entries than the other classes of images. Deep learning models trained on imbalanced data sets tend to be biased and sensitive. One can use a I2I translation model to synthesis fake samples, so data sets become more balanced. Image super-resolution is another example of I2I translation, which refers to turning low-resolution images to high-resolution ones. Image super-resolution is of great value. One application is that one only needs to transmit a low resolution image over the internet at low cost, and then the recipient restores the image to high resolution using a trained model.

I2I translation not only has wide practical uses but is also a study that constantly encourages new theoretical breakthroughs. I2I translation can benefit from the information in Chapter 2: autoencoders, GANs, and flow-based models. We can impose various regularizations on loss functions to learn disentangled representations of input data, which directly inspires the study of I2I translation. For example, varying the smile dimension in learned disentangled representation can turn neutral faces into smiling ones. Another kind of generative models are flow-based models, which are based on the change of variable theorem and learning an invertable mapping from image to image, which is a natural option for modelling I2I. GANs can generate images with higher quality than autoencoders via adversarial training. The generator can be even be an AE, a flow-based model, or a mix of the two,

which provides many options in designing generative models. Flow-based models can directly model data distribution instead of sampling through a prior latent distribution as in GANs, and it can also help reveal disentangled latent factors, which can be used directly for I2I translation.

In the next four chapters, I will outline potential approaches to I2I translation with methods mostly focused on GANs and AEs.

Chapter 3

A Unified Framework for Unsupervised I2I Translation

Image-to-image translation refers to translating images from one domain to another featuring different styles, styles that are visually distinct. An example is the task of turning images of cartoon sketches into real-life graphs. Many tasks in computer vision can be viewed as image-to-image translation, such as image inpainting[19], style transfer[20;21], and super-resolution[22]. Supervised image-to-image translation tasks need paired data sets that are costly to obtain, and such tasks are relatively easier to solve than their unsupervised counterparts. Under paired data supervision, image-to-image translation can be done by taking a regression approach[23] or using conditional generative models[24]. Our work addresses the more challenging unsupervised image-to-image translation task without access to paired data sets. Most research into unsupervised image-to-image translation draws inspiration from CycleGANs[25] using the cycle consistency constraint, research that has achieved impressive results. More recent studies have improved upon CycleGANs and can translate images among multiple domains. The research often operates under the assumption that latent codes can be decomposed as content codes and style codes[28;29]. Translation thus exchanges style codes from

different domains while keeping the original content codes. Style codes, however, are simply sampled from a standard normal distribution, which leads to inferior translation results. Moreover, these translations require training $n(n-1)$ generators and n discriminators to learn to translate images among n domains, which is costly. In our study, I suggest a simpler yet more effective approach. Our framework shares the assumption that style codes are domain-dependent and content codes are domain-invariant. However, our approach uses only one generator-discriminator pair and a mapping network, which learns the style codes of different domains. I also used several effective techniques for encouraging more diverse translated results. Extensive experiments showed that our framework is superior or comparable to state-of-the-art (SOAT) baselines. To summarize, the novel contributions for this part of work include

- A new I2I translation framework that is simplified yet more effective than existing approaches and reduces network size and training time.
- A novel framework that consists of a mapping network to learn styles of different domains, which leads to better translation results.
- Several regularization techniques to impose the assumption that content codes are domain-invariant and style codes are diverse.
- Extensive experiments on publicly available data sets show that this framework is superior or comparable to SOTA baselines.

3.1 Related Work

Generative adversarial networks. Ideally, generative models learn how data is distributed, thus allowing data synthesis from the learned distribution. Since the advent of GANs[70], generative models have achieved impressive results in tasks like image editing[52;53]

and style transfer[21]. GANs try to learn the data distribution by approximating the similarity of distributions between the training data and the fake data produced by the learned model. GANs usually comprise a generator and a discriminator. The entire model learns by playing a min-max game: the generator tries to fool the discriminator by gradually generating realistic data samples, and the discriminator, in turn, tries to distinguish real samples from fake ones. GANs have been improved in various ways. To produce more realistic samples, an architecture of stacked GANs has been proposed: the laplacian pyramid of GANs[54]; layered, recursive GANs[55]; progressive growing GANs[56]; and style-based GANs[20;21]. Several studies have attempted to solve the instability training of GANs using energy-based GANs[57], Wasserstein GANs[58], and boundary equilibrium GANs[59]. In this study, I used GANs with improved techniques to learn the distribution of data and how to translate among different domains.

Unsupervised image-to-image translation. Unsupervised image-to-image translation takes images from one domain and translates them to another without paired data supervision. Much success in unsupervised image-to-image translation is due to the cycle consistency constraint, proposed in three earlier research reports: CycleGANs[25], DiscoGANs[26], and DualGANs[27]. To translate more than two domains, MUNIT[28] and DRIT[29] can be used. These methods, however, naively sample style codes from a standard normal distribution, which leads to inferior translation results. Moreover, they require training $n(n-1)$ generators and n discriminators for translating images among n domains, which is computationally expensive and time-consuming. Our method involved a simpler yet more effective approach that required only one set of generator-discriminator. Recent systems like StarGAN2[30] and ModularGANs[31] were developed for multimodal image-to-image translation to produce images with the same content but different contexts. Inspired by StyleGANs[20], I used a mapping network to model style codes of different domains. Furthermore, I added several regularization techniques to encourage diversity in translated results.

3.2 Methods

3.2.1 Preliminaries

Let x be an image that belongs to one of many domains. Graph (a) in Figure 6.1 shows an overview of our model. I began with a latent vector z sampled from a standard normal distribution. z goes through a mapping network, which learns style codes s of a specific domain, where m is a domain label and $s = M(z, m)$. I furthermore used a content encoder E_c to extract content codes c from image inputs. The decoder D takes content and style codes to generate reconstructed images x', which are then used by style encoder E_s to produce reconstructed style codes s'. I computed two L1 losses using the reconstructed images and style codes. Finally, I used a multi-task discriminator to distinguish real images from generated ones. During the translation phase, I kept the original content codes but used the style codes of the target domains. Graph (b) of Figure 6.1 illustrates image translation between two domains.

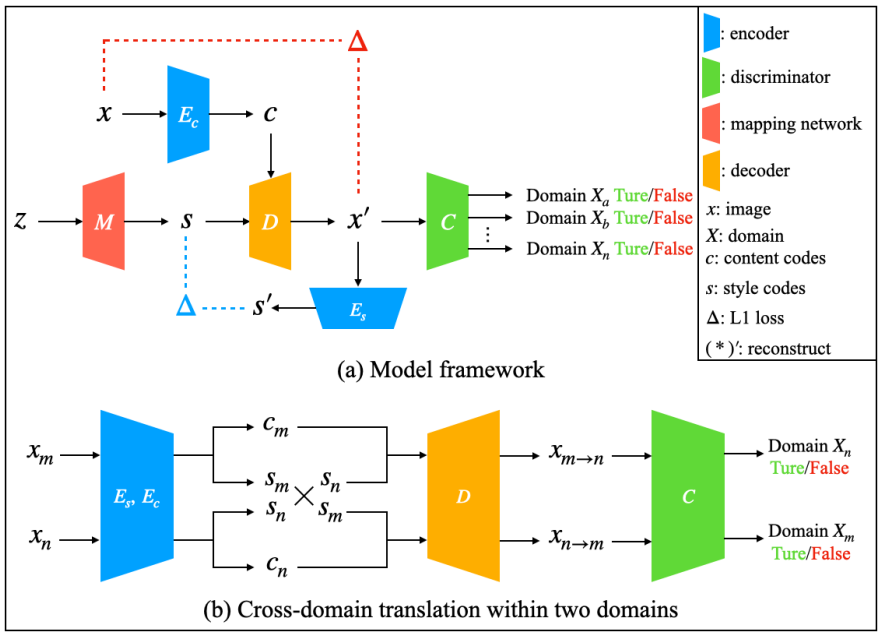


Figure 3.1: *The structure of our framework. (a) shows how our framework learns and (b) shows cross-domain translation within two domains.*

3.2.2 Training Objectives

In this section, I discuss the loss functions for learning in our framework.

Image reconstruction loss. After images are encoded to style and content codes, the decoder maps the latent space back to the image space and reconstructs the image. Image reconstruction loss is formulated as

$$L^x_{recon} = \|D(E_c(x), M(z, m)) - x\|_1 , \tag{3.1}$$

where m is the domain to which image x belongs.

Style code reconstruction loss. After encoding reconstructed images using the style encoder, I obtained reconstructed style codes. I constructed the style code reconstruction loss as follows:

$$L^s_{recon} = \|s - E_s(x')\|_1 , \tag{3.2}$$

where $x' = D(E_c(x), M(z, m))$ and $x \in X_m$.

Regularization of style and content codes. To further encourage domain-variant style codes and domain-invariant content codes, I added regularizers to style and content encoders. The style regularizer forces style codes of different domains to differ by minimizing L^s_{regu}, which is calculated as

$$\begin{aligned} L^s_{regu} = & - \|D(c_m, s_m) - D(c_m, s_n)\|_1 \\ & - \|D(c_n, s_m) - D(c_n, s_n)\|_1 , \end{aligned} \tag{3.3}$$

where $(c_m, s_m) = (E_c(x_m), E_s(x_m))$ and $(c_n, s_n) = (E_c(x_n), E_s(x_n))$. c_m and s_m are content and style codes of image $x_m \in X_m$. c_n and s_n are content and style codes of image $x_n \in X_n$.

The content regularizer encourages content codes of different domains to remain similar by

minimizing L^c_{regu}, which is formulated as

$$\begin{aligned} L^c_{regu} = & \left\| D(c_m, s_m) - D(c_n, s_m) \right\|_1 \\ & + \left\| D(c_m, s_n) - D(c_n, s_n) \right\|_1 . \end{aligned} \tag{3.4}$$

Inspired by StarGAN2[30], I calculated style diversity as

$$L_{ds} = \left\| E_s(x_1) - E_s(x_2) \right\|_1 , \tag{3.5}$$

where z_1 and z_2 are two random latent vectors; $x_1 = D(E_c(x), M(z_1, m))$, and $x_2 = D(E_c(x), M(z_2, m))$.

Adversarial loss. GANs are used to match the distribution of translated results to real image samples, to the point that the discriminator finds real and fake samples indistinguishable. I used two adversarial losses with one for learning latent-guided translation and the other for reference-guided translation. Latent-guided translation refers to using the mapping network to obtain target style codes, and reference-guided translation uses the style encoder to extract style codes of target domains. The adversarial loss for learning the discriminator C_m with latent-guided translation is formulated as

$$\begin{aligned} L^l_{adv} = & \mathop{\mathbb{E}}_{z \sim N(0,I), x_n \sim p(X_n)} [log C_m(D(E_c(x_n), M(z, m)))] \\ & + \mathop{\mathbb{E}}_{x_m \sim p(X_m)} [log(1 - C_m(x_m))], \end{aligned} \tag{3.6}$$

and the adversarial loss for learning the discriminator C_m with reference-guided translation is constructed as

$$\begin{aligned} L^r_{adv} = & \mathop{\mathbb{E}}_{x_m \sim p(X_m), x_n \sim p(X_n)} [log C_m(D(E_c(x_n), E_s(x_m)))] \\ & + \mathop{\mathbb{E}}_{x_m \sim p(X_m)} [log(1 - C_m(x_m))], \end{aligned} \tag{3.7}$$

where the discriminator C_m attempts to identify if images are from the domain m.

Full objective. Our full objective is formulated as follows:

$$\begin{aligned} \min_{M,E,D} \max_{C} \quad & \lambda_1 L^x_{recon} + \lambda_2 L^s_{recon} + \lambda_3 (L^s_{regu} + L^c_{regu}) \\ & + \lambda_4 (L^l_{adv} + L^r_{adv}) - \lambda_5 L_{ds}, \end{aligned} \tag{3.8}$$

where λ_1 to λ_5 are hyperparameters for each loss term.

3.3 Theoretical Analysis

I established a theoretical foundation for our framework. Specifically, minimizing the proposed loss functions led to 1) reconstructed images with the same distribution as the original image. Further, the encoder and decoder had reverse functions. 2) The content codes matched across domains with the interpolators mapping the source style codes to target style codes. And 3) the conditional distributions $p(x_n|x_m)$ and $p(x_m|x_n)$ were estimated with the learned distribution $p(x_{m\to n}|x_m)$ and $p(x_{n\to m}|x_n)$. For ease of notation, I denoted the two conditionals $p(x_{m\to n}|x_m)$ and $p(x_{n\to m}|x_n)$ as $p(x_{mn}|x_m)$ and $p(x_{nm}|x_n)$.

3.3.1 Proposition 1

When the image reconstruction loss is minimized, reconstructed images have the same distribution as the original image, and the encoder and decoder are reverse functions.

Proof: The image reconstruction loss functions are described in (4.1) and (4.2). When the reconstruction loss is minimized, the reconstructed images look similar to the original images, which is to say $p(x_m) = p(x^{recon}_m) = p(x_{mnm})$.

If I have $p(x_m) = p(x_m^{recon})$, then

$$p(x_m^{recon}) = p(D(E_c(x_m), E_s(x_m))) = p(x_m).$$

Thus, the encoder and decoder are reverse functions, which is to say $E = (D)^{-1}$.

3.3.2 Proposition 2

When the adversarial loss is minimized, the content codes match across domains, and the interpolator can map the style codes from the source domain to the target domain.

Proof: When (4.8) is minimized, the translated image is indistinguishable from images in the target domain, meaning $p(x_{mn}) = p(x_n)$ and $p(x_{nm}) = p(x_m)$. Thus, we have

$$\begin{aligned} p(D(c_n, s_n)) &= p(x_n) \\ &= p(x_{mn}) \\ &= p(D(c_m, s_n)). \end{aligned}$$

Thus, $p(c_n) = p(c_m)$, or the content codes match across domains. When translating style codes across domains, an interpolator guides the transition. Therefore, the proof can be written

$$\begin{aligned} p(D(c_n, s_n)) &= p(x_n) \\ &= p(x_{mn}) \\ &= p(D(c_m, s_{m\to n})) \\ &= p(D(c_m, I_{mn}(s_m, s_n))) \end{aligned}$$

since $p(c_n) = p(c_m)$, then $p(I_{mn}(s_m, s_n))) = p(s_n)$. Similarly, $p(I_{nm}(s_m, s_n))) = p(s_m)$, which proves that the interpolator can effectively map style codes from the source domain to the target domain.

3.3.3 Proposition 3

When optimality is achieved, the conditional distributions $p(x_m|x_n)$ and $p(x_n|x_m)$ can be estimated with $p(x_{nm}|x_n)$ and $p(x_{mn}|x_m)$.

Proof: If $p(x_{mn}|x_m) = p(D(c_m, s_n)|x_m)$, and since $p(c_n) = p(c_m)$, we can write

$$\begin{aligned} p(x_{mn}|x_m) &= p(D(c_m, s_n)|x_m) \\ &= p(D(c_n, s_n)|x_m) \\ &= p(x_n|x_m). \end{aligned}$$

Similarly, $p(x_{nm}|x_n) = p(x_m|x_n)$, meaning the conditional distributions $p(x_m|x_n)$ and $p(x_n|x_m)$ can be estimated with the learned distribution $p(x_{nm}|x_n)$ and $p(x_{mn}|x_m)$ without having to know the joint distribution, $p(x_m, x_n)$.

Data sets. I evaluated our framework on CelebA-HQ[56] and AFHQ[30] data sets. As with StarGAN2[30], I separated CelebA-HQ as domains of male and female and AFHQ as domains of cat, dog, and wild. For a fair comparison, all images were trained with size 256×256, the largest resolution supported by the baselines.

Evaluation metrics. I evaluated visual quality using Frechét inception distance (FID)[71] and the diversity of translated images with learned perceptual image patch similarity (LPIPS)[72]. FID measures the discrepancy between two sets of images. I translated each test image in

the source domain into a target domain using 10 reference images randomly sampled from the test set of the target domain. I then calculated FID between the translated images and test images in the target domain. I calculated FID for every pair of image domains (e.g., cat $\leftrightarrow$ dog) and reported the average value. LPIPS measures the diversity of generated images using the L_1 distance between features extracted from the pretrained AlexNet[4]. For each test image from a source domain, I generated 10 outputs of a target domain using 10 reference images randomly sampled from the test set of the target domain. I then computed the average of the pairwise distances among all outputs produced from the same input, which are 45 image pairs. Finally, I reported the average LPIPS values over all test images. Lower FID values indicated the two sets of images have similar distributions. Higher LPIPS values indicated higher diversity of generated images.

To evaluate visual quality of translation results, I used the Amazon Mechanical Turk (AMT) to compare the results against the baselines based on user preferences. Given a source image and a reference image, AMT workers selected the best transfer result among all models.

3.4 Experiments

3.4.1 Data Sets

I used data sets similar to those used in previous research on unsupervised I2I translation. As in previous research[24;28;52], I used images of shoes and their edge map images generated by[73]. With 100,000 images of shoes $\leftrightarrow$ edges, 400 images of them were used for testing; the rest were used as the training data set. Fig. 3.2 provides an example of such edge maps for shoes and handbags.

The cats $\leftrightarrow$ dogs data set comes from Huang[28], which contains approximately 2,300

Figure 3.2: *Examples of shoe and handbag edges*

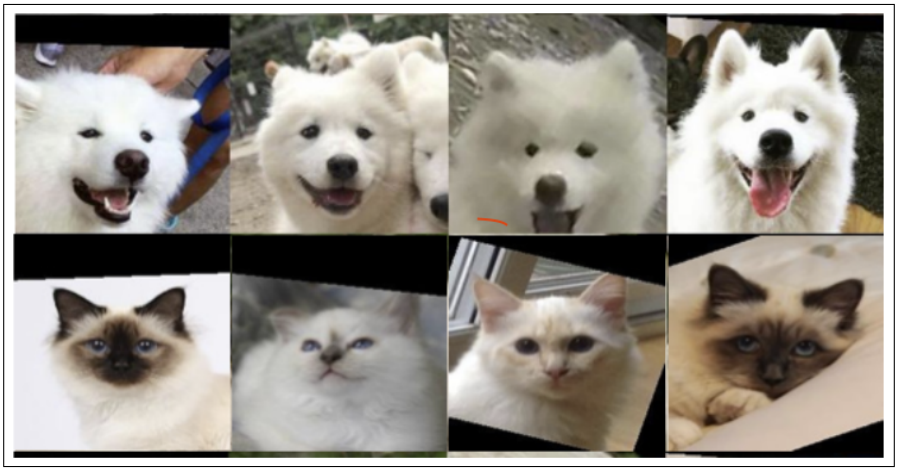


Figure 3.3: *Examples of cat and dog images*

images of cats and dogs. We used 100 images of cats and 100 images of dogs for testing; the rest were used for training. Fig. 3.3 shows an example of cat and dog images.

3.4.2 Baselines

I compared our framework to three baseline models developed in recent years. Our framework is closely related to MUNIT[28] and DRIT[29], which I also used as baseline models. More recent

Figure 3.4: *Examples of animal and human images*

research on unsupervised image-to-image translation used StarGAN2[30;74–76], which achieves impressive results and was another baseline model in our study.

Data sets and framework details. We evaluated our framework on CelebA-HQ[56] and AFHQ[30] data sets. As with StarGAN2[30], we also separated CelebA-HQ as domains of male and female, and AFHQ as domains of cat, dog, and wild. For a fair comparison, all images were trained with images 256×256, which is the largest resolution supported by the baselines. The detailed architecture of our framework and training details are in Supplementary information, and code will available upon publication.

Evaluation metrics. We evaluated the visual quality using Frech´t inception distance (FID)[71] and the diversity of translated images with learned perceptual image patch similarity (LPIPS)[72]. Images generated by our framework are compared with the testing data set to calculate FID and LPIPS. Lower FID values indicate that the two sets of images have more similar distributions. Higher values of LPIPS indicate higher diversity of generated images.

Models	Performance (↑)
MUNIT	2.820 %
DRIT	9.050 %
StarGAN2	43.50 %
Ours	**44.63 %**

Table 3.1: *Votes from ATM workers for most preferred style transfer results.*

Models	CelebA-HQ		AFHQ	
	FID(↓)	LPIPS (↑)	FID(↓)	LPIPS (↑)
MUNIT	31.4	0.363	41.5	**0.511**
DRIT	52.1	0.178	95.6	0.326
StarGAN2	**13.7**	0.452	**16.2**	0.450
Ours	17.5	**0.459**	19.9	0.476
Test data	*14.8*	–	*12.9*	–

Table 3.2: *Quantitative comparison on latent-guided translation.*

3.5 Results

In this section, we show the qualitative and quantitative results of the experiments. An ablation study was used to evaluate the effectiveness of several key design choices.

Qualitative results. We used the AMT to compare our results against the baselines. Given a source image and a reference image, AMT workers selected the best transfer result among all models. All ten workers answered 60 questions about images. Table 3.1 shows our method slightly outperformed StarGAN2[30] and exceeded MUNIT[28] and DRIT[29] by a large margin.

Quantitative results. The latent-guided and reference-guided translations were similar to StarGAN2[30]. Figure 3.6 provides examples. We used FID to evaluate the similarity of distributions and LPIPS to evaluate the diversity of generated images. As tables 3.2 and 3.3 show, our method and StarGAN2[30] performed similarly, and both outperformed MUNIT[28] and DRIT[29] by a large margin other than latent-guided LPIPS results of MUNIT[28] on AFHQ. StarGAN2[30] achieved the lowest FID for the both data sets, and our method achieved the highest LPIPS among all models.

Models	CelebA-HQ		AFHQ	
	FID(↓)	LPIPS (↑)	FID(↓)	LPIPS (↑)
MUNIT	107.1	0.176	223.9	0.199
DRIT	53.3	0.311	114.8	0.156
StarGAN2	**23.8**	0.388	**19.8**	0.432
Ours	25.3	**0.391**	22.3	**0.439**
Test data	*14.8*	–	*12.9*	–

Table 3.3: *Quantitative comparison on reference-guided translation.*

Ablation studies To further validate effects of key design choices in our framework, we carried out ablation studies on the AFHQ data set. The results are shown in Figure 3.5 and Table 3.4. The model without style and content regularizer but with noise injection was used as the vanilla model. We can see that a style regularizer effectively increases diversity in generated images.

Modules	FID (↓)	LPIPS (↑)
vanilla model	29.1	–
+ noise injection	27.6	0.407
+ content regularizer	23.8	0.414
+ style regularizer	**22.3**	**0.439**

Table 3.4: *FID and LPIPS results of incrementally adding modules to our framework for reference-guided translation on the AFHQ data set. The vanilla model does not report LPIPS result as it is a deterministic model.*

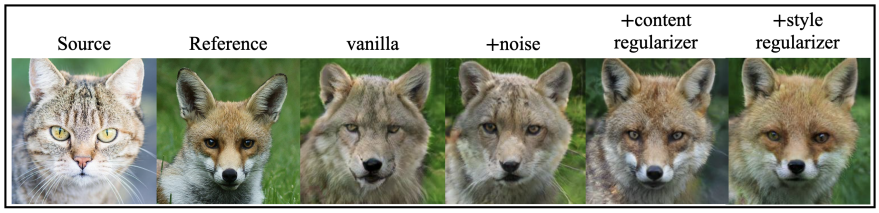
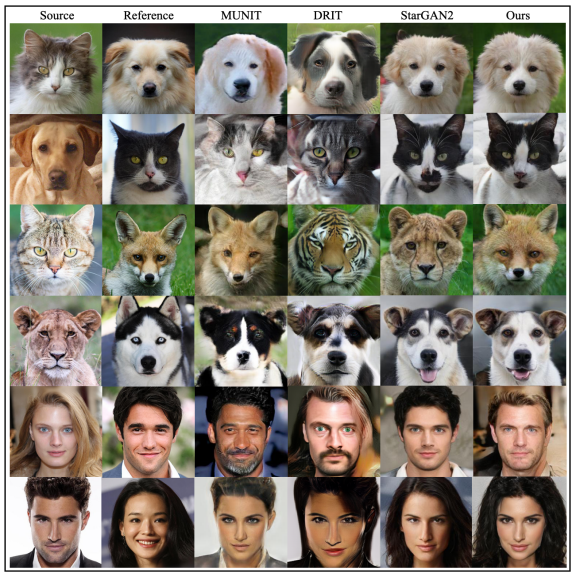


Figure 3.5: *An example of reference-guided translation by incrementally adding modules*

3.6 Conclusions

In this report, I presented a simpler yet more effective framework for multimodal unsupervised image-to-image translation. Our model consisted of a mapping network and a generator-discriminator pair only. Unlike MUNIT[28] and DRIT[29] that simply sample style

codes from a standard normal distribution when translating, we used a mapping network to learn style codes of different domains. To further encourage diversity in translated images, we used style regularizations and injected Guassian noise in the decoder. The qualitative and quantitative results show that our framework is superior or comparable to the SOAT baselines in multimodal unsupervised image-to-image translation.



(a) Examples of reference-guided translation.

(b) Examples of latent-guided translation.

Figure 3.6: *Examples of image-to-image translation guided by reference images and latent codes*

Chapter 4

Improve Unsupervised I2I with Fine-grained Control on Latent Space

Image-to-image (I2I) translation refers to translating images from one domain to another domain with different properties. An example is the task of turning images of cartoon sketches into real-life graphs. Many tasks in computer vision can be posed as I2I translation, such as image inpainting[19], style and attribute transfer[20;21], and super-resolution[22]. Paired I2I transfer tasks require paired data sets that are costly to acquire, making such tasks relatively easier to solve than unpaired I2I transfer tasks. Chen and Koltun translated paired images of semantic map to photographic images using a regression approach[23]. Isola et al. framed paired I2I translation tasks with conditional generative models[24]. Our work addresses the more challenging unpaired I2I task, where no paired data sets are available. Most research on unpaired I2I translation draws inspiration from CycleGANs using the cycle consistency constraint[25], which have achieved impressive results. These models, however, often have little control over translation strength and can only provide a single translated image as output. Furthermore, they often disentangle latent space into domain-invariant (content codes) and domain-specific parts (style codes). When translating, content codes

are kept while style codes are exchanged. Domain-specific information, however, might still exist in content codes, which leads to unnatural translation results if that information is not removed[77].

In this research, the need for fine-grained control over latent space demonstrated the inferior translation capability of previous research that depended solely on the cycle consistency constraint or translated images by simply exchanging style codes. Fine-grained control over latent space manifests in three aspects: 1) latent codes can be decomposed into content and style, much like DRIT[29] and MUNIT[28]; 2) an interpolator, which is a neural network, can guide the transformation of style codes instead of simply exchanging one style code to another; and 3) domain-specific information in content code is removed before translation for better translation results. Much like DRIT and MUNIT, our framework assumed that latent space can be decomposed into content space by the content encoder and style space by the style encoder. Before decoding the latent codes to obtain translated results, redundant domain-specific information in content codes is removed. Furthermore, another set of modules, which we called the interpolator, smoothly guide the transition of style codes, allowing us to generate intermediate images with different degrees of transformation. In the end, our framework differentiated translated images using a discriminator. Extensive experiments demonstrated that our method is superior or comparable to state-of-the-art (SOTA) baselines in unpaired I2I translation. The novel contributions of this research are

- A new framework with an embedded interpolator can control translation strength and generate smooth-looking translation results of intermediate states.
- New techniques can remove domain-specific information in content codes.
- A simplified framework architecture was based on MUNIT, reduced the size and training time, but achieved better translation performance.
- Extensive experiments on publicly available data sets showed that the framework is superior or comparable to SOTA baselines.

4.1 Related Work

Generative adversarial networks. Ideally, generative models learn how data is distributed, thus allowing data synthesis from the learned distribution. Since the advent of GANs[70], generative models have achieved impressive results in both image editing[52] and style transfer[21]. GANs try to learn the data distribution by approximating the similarity of distributions between the training data and the fake data produced by the learned model. GANs usually comprise a generator and a discriminator. The entire model learns by playing a min-max game: the generator tries to fool the discriminator by gradually generating realistic data samples, and the discriminator, in turn, tries to distinguish real samples from fake ones. GANs have improved in various ways. To produce more realistic samples, an architecture of stacked GANs has been proposed: the laplacian pyramid of GANs[54]; layered, recursive GANs[55]; and style-based GANs[20;21]. Several studies have attempted to solve the instability training of GANs using energy-based GANs[57], Wasserstein GANs[58], and boundary equilibrium GANs[59]. In this study, I used GANs with improved techniques to learn the distribution of image data and translate them to different domains.

Unpaired I2I translation. Unpaired I2I translation translates images from one domain to another without paired data supervision. Much success in unpaired I2I translation is due to the cycle consistency constraint, proposed in earlier research: CycleGANs[25], DiscoGANs[26], and DualGANs[27]. Recent systems like MUNIT[28] and DRIT[29] were developed to perform multimodal I2I translation, which refers to producing images with the same content but different contexts. For example, a winter scene could be translated into many different summer scenes depending on weather or lighting. To translate into more than two domains, StarGAN-V2[30] and ModularGANs[31] were proposed. I2I translation methods using GANs that rely only on cycle consistency constraints usually suffer from the issue of discreteness, which refers to inability to continuously control the transformation strength. In this study, I used an interpolator to guide the translation, which allowed us to generate visually appealing intermediate translation results.

Our framework is closely related to MUNIT in that latent space can be decomposed into a style sub-space and a content sub-space. Our framework, however, differs from MUNIT in four aspects: 1. Instead of having to train $n(n-1)$ sets of encoder-decoder for translating images between n domains, our framework consists of only one such set that works for multi-domains; 2. Our framework does not impose a Gaussian prior distribution for style codes, instead learning distributions during training; 3. Our framework removes redundant domain-specific information in content codes before translation, thus generating more natural-looking results; 4. Most unpaired I2I translation models that depend on cycle-consistency loss cannot generate sequences of intermediate translation results. I used an interpolator module that helped smoothly translate the latent codes of different domains and produced visually satisfying intermediate translation results.

4.2 Methods

4.2.1 Preliminaries

Let $x_m \in X_m$ and $x_n \in X_n$ be two images from domain X_m and domain X_n. Our goal was to estimate the conditional distributions $p(x_m|x_n)$ and $p(x_n|x_m)$ using the learned distribution $p(x_{n\to m}|x_n)$ and $p(x_{m\to n}|x_m)$, given the marginal distribution of $p(x_m)$ and $p(x_n)$ but without requiring access to the joint distribution of $p(x_m, x_n)$. Figure 4.1 shows an overview of our model. Our framework starts with an encoder $E = (E_s, E_c)$ that maps images from image space to latent space, where E_s is the style encoder and E_c is the content encoder. The latent codes consist of style latent codes (s_m, s_n) and content latent codes (c_m, c_n), where $(c_m, s_m) = (E_c(x_m), E_s(x_m))$ and $(c_n, s_n) = (E_c(x_n), E_s(x_n))$. After style codes are obtained, an interpolator I helps transform the style codes across different domains. The translated style codes $s_{m\to n}$ and $s_{n\to m}$ are obtained by calculating $s_m + \alpha * I_{mn}(s_n - s_m)$ and $s_n + \alpha * I_{nm}(s_m - s_n)$, where α is the transformation strength. Style is injected into

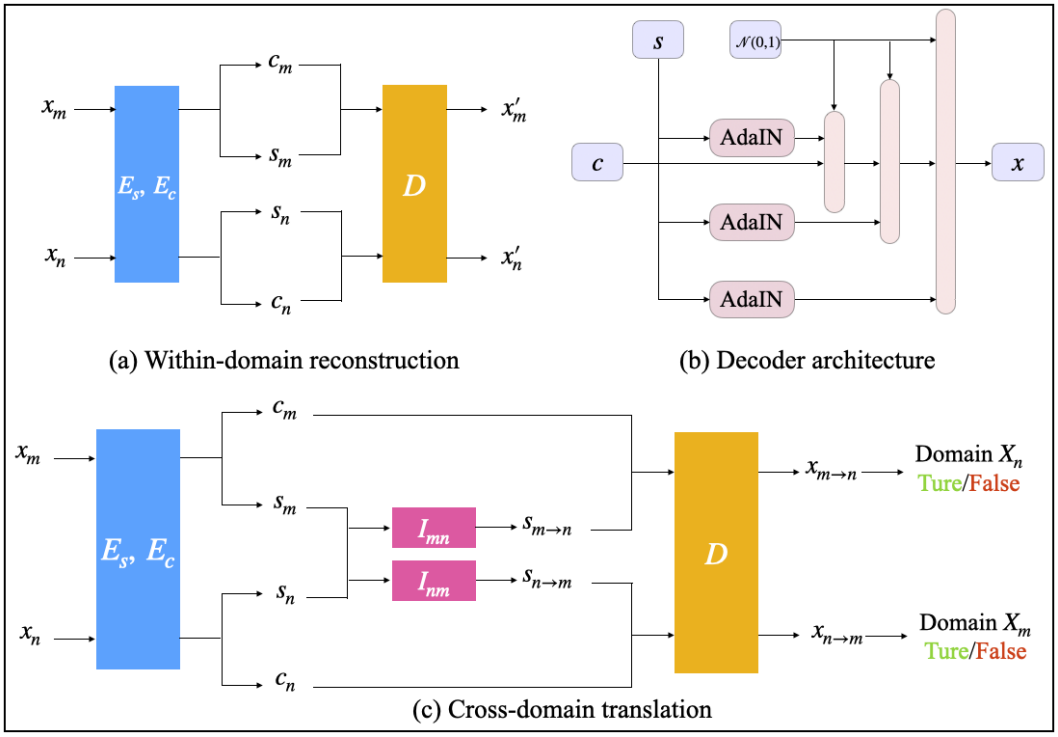


Figure 4.1: *The structure of our framework: (a) shows within-domain image reconstruction, and (b) shows key components of the decoder. The number of convolutional layers are more than what the graph shows; (c) shows cross-domain translation.*

the decoder by AdaIN[78] operations. Before injecting the style of the target domain, I removed domain-specific information by injecting the negative style of the same domain; the strength of the negative style is learned during training. Inspired by StyleGAN[20], I introduced stochastic variation into our model by injecting noise into the decoder. After the transformed style codes are obtained, the decoder D decodes the style and content codes back to image space, thus generating translated images x_{mn} and x_{nm}, where $x_{mn} = D(c_m, s_{m\to n})$ and $x_{nm} = D(c_n, s_{n\to m})$. Finally, the discriminator C tries to differentiate real images from fake ones.

4.2.2 Loss Functions

In this section, I discuss the loss functions and the training algorithm of our framework.

Image reconstruction loss. After images are encoded to style and content codes, the decoder can map them back to the image space and reconstruct the image. Therefore, the image reconstruction loss of x_m is formulated as

$$L_{recon}^{x_m} = \|D(E_c(x_m), E_s(x_m)) - x_m\|_1 , \tag{4.1}$$

and $L_{recon}^{x_n}$ is expressed similarly. After images are translated from one domain to another, the images in the source domain can be reconstructed by inverting the process. For example, x_{mn} has the content of image x_m and the style from domain X_n. x_{mn} is obtained by evaluating $D(c_m, s_n)$. Encoding x_{mn} again produces (c'_m, s'_n), and by decoding $D(c'_m, s_m)$, x_m, which is now denoted by x_{mnm}, can be reconstructed. Thus, $L_{recon}^{x_{mnm}}$ is calculated as

$$L_{recon}^{x_{mnm}} = \|x_{mnm} - x_m\|_1 = \|D(E_c(x_{mn}), E_s(x_m)) - x_m\|_1 . \tag{4.2}$$

Similarly, $L_{recon}^{x_{nmn}} = \|x_{nmn} - x_n\|_1$. The reconstructed images should be consistent with the semantics of the original images, so perceptual loss was penalized to minimize the semantic difference:

$$L_{perc}^{x_m} = \|\Phi_3(D(E_c(x_m), E_s(x_m))) - \Phi_3(x_m)\|^2 , \tag{4.3}$$

where Φ_3 denotes the ReLU3_1 layer of a pretrained VGG network[79]. Perceptual loss was similarly calculated for $L_{perc}^{x_n}$, $L_{perc}^{x_{mnm}}$, and $L_{perc}^{x_{nmn}}$.

Latent code reconstruction loss. By encoding the translated images, I obtained a new set of content and style codes. For example, encoding the translated image x_{mn} produces

(c'_m, s'_n). The latent code reconstruction loss was calculated as

$$L^c_{recon} = \|c'_m - c_m\|_1 ; L^s_{recon} = \|s'_n - s_n\|_1 . \tag{4.4}$$

Interpolation loss. Given latent codes of two domains means latent codes can be interpolated linearly. For example, $s_m + \alpha * (s_n - s_m)$ translates s_m to s_n under translation strength α. This approach, however, does not guarantee smooth-looking results because the translation path might not be linear. I used an interpolator to smoothly transition style codes of different domains, calculated as $s_m + \alpha * I_{mn}(s_n - s_m)$. α, a random value that is uniformly sampled from 0 to 1, controls the translation strength. For domain labels, however, I adopted a linear interpolation strategy. That is to say, I linearly interpolated the domain labels using the same α and used the interpolated domain label as ground truth. The intuition behind this is that linearly interpolated images should have linearly interpolated labels, but linearly interpolated images are not guaranteed to be smooth-looking. Therefore, an interpolator network is trained to guide the translation. The discriminator C is trained to produce realistic fake images and also to predict domains of images. I used the binary cross entropy (BCE) loss and adversarial loss jointly to train the interpolator. The BCE loss function for the interpolator I_{mn} is calculated as

$$L_{I_{mn}} = \text{BCE}(C(x_{mn}), gt\_domain), \tag{4.5}$$

where x_{mn} is a translated image via $D(c_m, s_m + \alpha * I_{mn}(s_n - s_m))$ and gt_domain is the ground truth domain label, which is linearly interpolated via $\text{label}_m + \alpha * (\text{label}_n - \text{label}_m)$. $L_{I_{nm}}$ can be calculated similarly.

Regularizers on style and content codes. To further encourage style codes to remain

domain-variant and content codes domain-invariant, I added regularizers on the style and content codes. The style regularizer forces style codes of different domains to differ by minimizing L^s_{regu}, which is calculated as

$$L^s_{regu} = - \left\| D(c_m, s_m) - D(c_m, s_n) \right\|_1 - \left\| D(c_n, s_m) - D(c_n, s_n) \right\|_1 . \tag{4.6}$$

The content regularizer encourages content codes of different domains to be similar by minimizing L^c_{regu}, which is calculated as

$$L^c_{regu} = \left\| D(c_m, s_m) - D(c_n, s_m) \right\|_1 + \left\| D(c_m, s_n) - D(c_n, s_n) \right\|_1 . \tag{4.7}$$

Adversarial loss. GANs match the distribution of translated results to real image samples, so the discriminator finds real and fake samples indistinguishable. The loss for learning the discriminator C is calculated as

$$L_C^{x_{mn}} = \mathop{\mathbb{E}}_{c_m \sim p(c_m), s_{m\to n} \sim p(s_n)} [log(1 - C(D(c_m, s_{m\to n})))] + \mathop{\mathbb{E}}_{x_n \sim p(X_n)} [logC(x_n)], \tag{4.8}$$

where the discriminator C tries to differentiate real images from X_n and translated images x_{mn}. $L_C^{x_{nm}}$ is obtained similarly.

Model training. We alternately trained our discriminator and the rest of the modules: encoders, decoders, mapping networks, and the interpolator. The training procedure of our framework is illustrated in Algorithm 1 using a convergence bound B that is empirically

calibrated at 1,000,000.

Algorithm 1: Model Training

Result: style encoder $\boldsymbol{E_s}$, content encoder $\boldsymbol{E_c}$, interpolators $\boldsymbol{I_{mn}}$, $\boldsymbol{I_{nm}}$, decoder $\boldsymbol{D}$, and $\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$ that control the strength of negative style injected for removing domain dependent information.

$n = 0$;

while $n < B$ **do**

Calculate $L_C^{x_{mn}}, L_C^{x_{nm}}$ according to (4.8);

Update the discriminator $\boldsymbol{C}$;

Calculate $L_{recon}^{x_m}, L_{recon}^{x_n}, L_{recon}^{x_{mnm}}, L_{recon}^{x_{nmn}}$ according to (4.1), (4.2);

Calculate $L_{perc}^{x_m}, L_{perc}^{x_n}, L_{perc}^{x_{mnm}}, L_{perc}^{x_{nmn}}$ according to (4.3);

Calculate $L_{recon}^{c}, L_{recon}^{s}$ according to (4.4);

Calculate $L_{I_{mn}}, L_{I_{nm}}$ according to (4.5);

Calculate $L_{regu}^{s}, L_{regu}^{c}$ according to (6.4), (6.9);

Update the decoder $\boldsymbol{D}$, the style encoder $\boldsymbol{E_s}$, the content encoder $\boldsymbol{E_c}$, $\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$, and the interpolator $\boldsymbol{I_{mn}}$, $\boldsymbol{I_{nm}}$;

$n++$;

end

4.3 Experiments

This section provides information about the data sets, baselines, and evaluation metrics that we used to test our framework.

Data sets. As in previous research[24;28;52], we used images of shoes and their edge map images generated by[73]. There are 100,000 images of shoes ↔ edges, and of these images, 400 were used for testing; the rest were used for training. The cats ↔ dogs data set came from[28], which has approximately 2,300 images of cats and dogs. We retained 100 images of

cats and 100 images of dogs for testing, using the rest for training.

Baselines. We compared our framework to three baseline models developed in recent years. Our framework is closely related to DRIT and MUNIT, which we used as baseline models. StarGAN-V2[30] was recently proposed and achieved SOTA results on unpaired I2I translation. Therefore, we used StarGAN-V2 as another baseline in our study.

Evaluation metrics. We evaluated the visual quality using Frechét inception distance (FID)[71] and the diversity of translated images with learned perceptual image patch similarity (LPIPS)[72]. FID measures the discrepancy between two sets of images. We translated each test image in the source domain into a target domain using 10 reference images randomly sampled from the test set of the target domain. We then calculated FID between the translated images and test images in the target domain. We calculated FID for every pair of image domains (e.g., cat $\leftrightarrow$ dog) and reported the average value. LPIPS measured the diversity of generated images using the L_1 distance between features extracted from the pretrained AlexNet[4]. For each test image from a source domain, we generated 10 outputs of a target domain using 10 reference images randomly sampled from the test set of the target domain. Then, we computed the average of the pairwise distances among all outputs produced from the same input, or 45 image pairs. Finally, we reported the average of the LPIPS values over all test images. Lower FID values indicate that the two sets of images have similar distributions. Higher values of LPIPS indicate higher diversity of generated images.

FID[71] and the diversity of translated images using LPIPS[72] are commonly used tp evaluate I2I translation performance. FID measures the distribution similarity between translation results and test set. LPIPS measures the diversity of generated images. Lower FID values indicate that the two sets of images have similar distributions. Higher LPIPS values indicate higher diversity of generated images.

Table 4.1: *Votes from ATM workers for most preferred style transfer results.*

Models	Performance (↑)
MUNIT	13.22 %
DRIT	15.06 %
StarGAN-V2	35.11 %
Ours	**36.61 %**

4.4 Results

In this section, we provide the qualitative and quantitative results of the experiments. An ablation study is also included for evaluating the effectiveness of several key design choices.

Qualitative results. Graph (a) of Figure 4.2 shows several example translation results of different models. To evaluate visual quality of translation results, user preferences from Amazon Mechanical Turk (AMT) compared our results against the baselines. Given a source image and a reference image, we instructed AMT respondents to select the best transfer result among all models. We asked 60 questions of all ten workers. As shown in Table 4.1, our translation method slightly outperforms StarGAN-V2[30] and exceeded MUNIT[28] and DRIT[29] by a large margin. Unlike the baselines, which suffered from the issue of discreteness and could produce only one final translation image, our framework generated sequences of intermediate translation results by interpolating style codes using different translation strengths. Graph (b) in Figure 4.2 shows results of translating between the cat and dog domains under different strengths of translation. Our framework used $s_m + \alpha * I_{mn}(s_n - s_m)$ during interpolation, which generated smooth-looking intermediate results. By default, other baselines cannot produce intermediate translation results. We interpolated the style codes linearly using $s_m + \alpha * (s_n - s_m)$, so we can see that StarGAN-V2 and MUNIT translation results contained artifacts while the DRIT results differed only in lighting.

Quantitative results. The qualitative observations are confirmed by quantitative evaluations. As Table 4.2 shows, StarGAN-V2 achieved the lowest FID and highest LPIPS on the `cat2dog` data set among all models, but our model results were comparable. Translated images by our model on the `edges2shoes` have lower FID and higher LPIPS values than all

Table 4.2: *Quantitative evaluation of image translation using FID and LPIPS. Cat images are translated to dog images, and edges are translated to shoe images.*

Metric	Data set	DRIT	MUNIT	StarGAN-V2	Ours
FID (↓)	cat2dog	148.87	122.04	**18.81**	21.53
FID (↓)	edge2shoes	273.93	274.11	63.78	**61.33**
LPIPS (↑)	cat2dog	0.251	0.263	**0.355**	0.341
LPIPS (↑)	edge2shoes	0.108	0.110	0.114	**0.126**

Table 4.3: *FID and LPIPS results of incrementally adding modules to our framework. LPIPS values for the naive model are not reported as it is a deterministic model.*

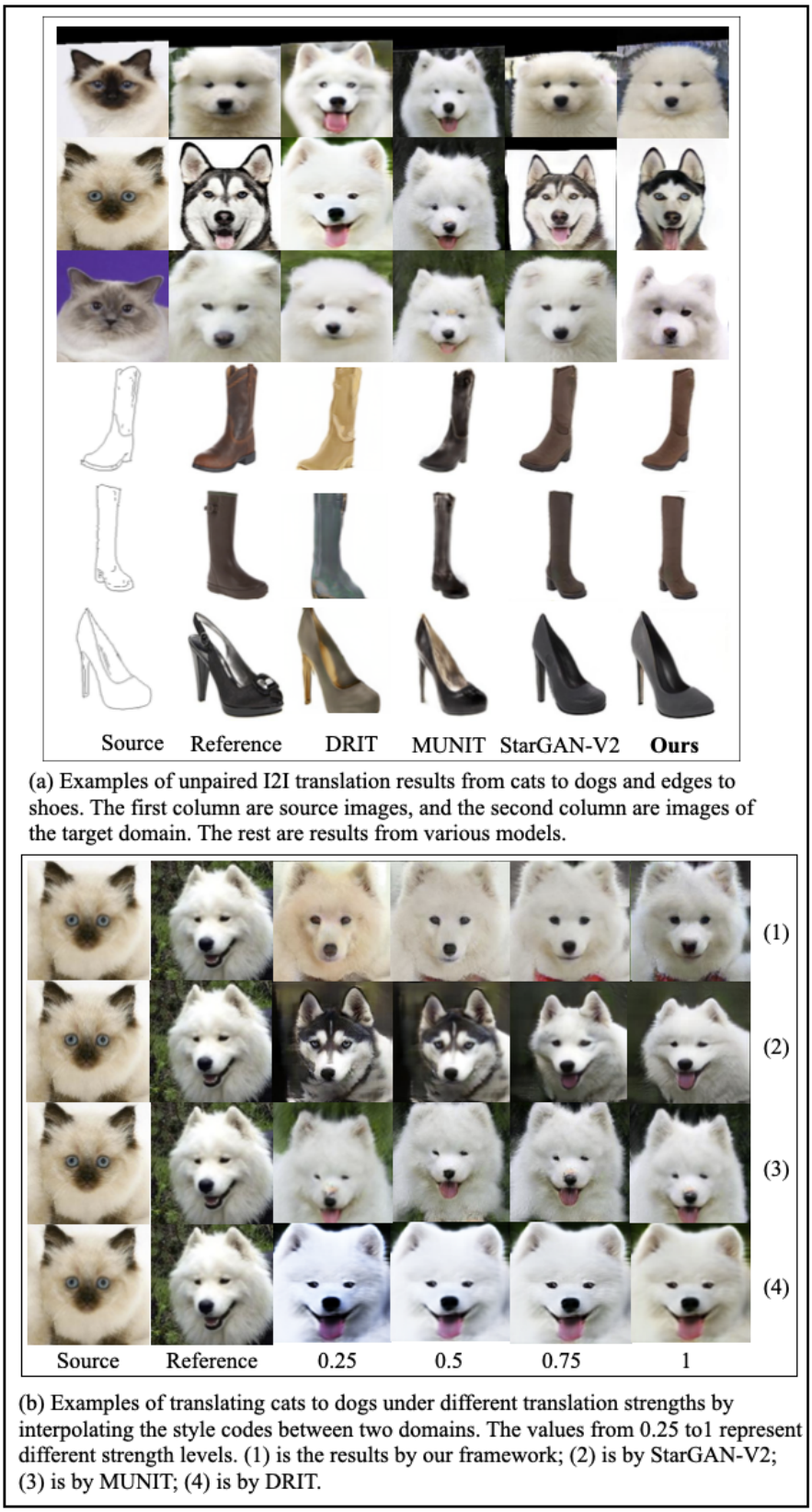
Modules	FID (↓)	LPIPS (↑)
naive model	103.30	—
+ noise injection	76.88	0.326
+ style regularization	59.21	0.329
+ content regularization	47.70	0.331
+ interpolators	30.45	0.333
+ domain-specific information elimination	**21.53**	**0.341**

baselines.

Ablation studies To further validate effects of key loss functions and design choices in our framework, we carried out ablation studies on the `cat2dog` data set. Let the model without domain-specific information removal ($\beta_{\boldsymbol{m}}$, $\beta_{\boldsymbol{n}}$), interpolators, latent codes regularizers, and noise injection be the naive model. We incrementally added modules to the naive model and calculated FID and LPIPS values. Table 4.3 shows the quantitative evaluations; qualitative results are in Figure 4.3.

4.5 Conclusions

In this research report, I presented a new framework for unpaired I2I translation. Our framework used fine-grained control over latent codes to achieve better translation results. We show that removing redundant domain-specific information during cross-domain translation



(a) Examples of unpaired I2I translation results from cats to dogs and edges to shoes. The first column are source images, and the second column are images of the target domain. The rest are results from various models.

(b) Examples of translating cats to dogs under different translation strengths by interpolating the style codes between two domains. The values from 0.25 to1 represent different strength levels. (1) is the results by our framework; (2) is by StarGAN-V2; (3) is by MUNIT; (4) is by DRIT.

Figure 4.2: *Examples of translating results by our framework. (a) compares translation results by different baselines; (b) shows examples of interpolation by all models.*

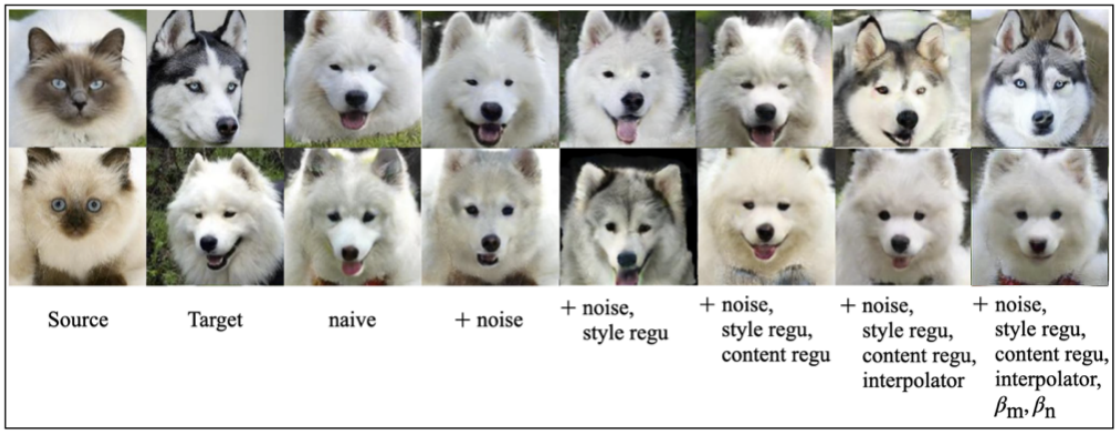


Figure 4.3: *Ablation study of our framework, which shows examples of translating cats to dogs by incrementally adding modules.*

helped produce better results. We also show that rather than simply exchanging style codes, an interpolator can guide the transformation to generate more visually appealing images, which also allows us to produce intermediate translation results. The qualitative results and quantitative evaluations show that our framework is superior or comparable to the SOTA baselines in unpaired I2I translation.

Chapter 5

Shape-aware Generative Adversarial Networks for Attribute Transfer

Attribute transfer in vision refers to transferring some abstract elements of source images to images in a target domain. For example, in human face attribute transfer, the task often refers to transforming smiling to neutral faces, while the identity of each face is maintained. Many approaches have been used for this task. However, many domains have only limited access to paired data (that is the same person with different facial expressions), attribute transfer mainly depends on cycle-consistency loss[16]. Several studies exploited mappings among different domains under the constraint of cycle-consistency and achieved satisfying results[1;2;30]. Fig 5.1 shows that StarGAN and ELEGANT can transfer human face attributes, but their approach does not provide intermediate states from the source to target domain.

Another branch of research studies latent space interpolation, which interpolates between two domains and thus generates a sequence of intermediate states between the source and just one target state. This assumes attribute space is flat and linear. Convolutional Neural Networks (ConvNets) have achieved great success in image classification in recent years[4–6;80]. The last layer of ConvNets is usually a fully connected layer without activation functions,

Figure 5.1: *Examples of transferring human face attributes by StarGAN[1] and ELEGANT[2]. Figure is excerpted from HomoInterpGAN[3].*

which linearly maps learned features to class labels. Because the features are linearly separable, one can transfer features simply. For example, let x and y be learned features of two instances from two different domains. The transferred x can be obtained by moving it in the direction of y. Intermediate states can also be obtained along the transition.

Although some successful attribute transfers have been demonstrated using cycle-consistency loss and latent space interpolation, the process remains a challenge, especially in successfully keeping the identity of source domain after transferring. Taking attribute transfer between human faces as an example, many GANs-based models can add glasses to face pictures or make non-smiling faces into smiling ones. This success partly depends on the fact that human faces have similar shapes and features have similar positions on the face (nose and eyes, for instance, have a relatively fixed position). Moreover, data sets like CelebA, are usually preprocessed so key point positions are aligned and generate natural-looking results for transfer images. In domains without this feature or domains with vastly different images, the resulting transfer images can look fake and unnatural if the techniques mentioned above are naively applied.

Thus, in this study, we used an example domain (a tomato leaf data set) that contains healthy leaves and diseased leaves. We treated different types of leaves as different domains; the goal was to transfer exemplars of healthy leaves to known categories of unhealthy leaves while maintaining their original identity, which we deemed to be the shape (especially outer contour) of the leaves. This is the novel contribution of this study, which generalized to

other domains with varying shapes. Our novel contributions follow.

- We proposed a novel, shape-aware GANs model that can process multi-domain, multi-modal attribute transfer while maintaining the image shape from the source domain.
- We proposed several strategies for stabilizing training of this model.
- Experiments showed that our model could generate more visually satisfying results than recently proposed SOTA baseline models while maintaining the quality of translated results.

5.1 Related Work

Image-to-Image Translation. Impressive results have been achieved in recent years in image-to-image translation. Pix2pix[24] produced high quality results with conditional GANs, using adversarial loss and L1 loss to guide the learning of the model. To increase the diversity of produced images, a noise term was added to an improved version of the pix2pix model. Since L1 loss depends on paired data that is difficult to obtain, research has also focused on unpaired image-to-image translation, for which CycleGAN[16] and DiscoGAN[26] were later developed. They use the cycle-consistency loss to map between two domains while maintaining some key attributes. However, they can, at most, train two domains at a time and are often applied to facial expressions. Moreover, naively applying cycle-consistency loss does not guarantee production of natural-looking results when domains differ greatly from one another and are internally diverse.

Latent Space Interpolation. One drawback of GANs that solely depends on cycle-consistency loss is an inability to produce a sequence of intermediate images from the source to target domain. Latent space interpolation builds on the fact that there is a flat feature space[81]. Once the original image space is mapped onto a feature space, interpolation can be

done by gradually moving the latent space of the source domain towards the target domain. However, there are infinitely many ways of connecting two points in the latent space, so finding the one that can produce smooth and natural-looking results is of great value. Instead of naively interpolating using a straight line connecting the two points in latent space, Chen et al.[3] used an artificial neural network (ANN) to learn the path and achieved visually satisfying results.

Image Segmentation. One of first deep learning models for image segmentation used a fully convolutional network (FCN)[82], with only convolutional layers that can produce segmentation maps that are the same size as input images. One of the drawbacks of FCN is that it pays little attention to useful scene-level semantic context. To remedy this problem, deep learning models that incorporated graphical models were created. Several learning frameworks incorporate conditional random fields[83–85]. Markov Random Fields are representations that are also commonly used to apply deep learning models to image segmentation [16]. Another branch of research on image segmentation used an encoder-decoder architecture. Some early research was done by[86], where the encoder used architecture similar to VGG-16 and the decoder consisted of deconvolution and unpooling layers. The UNet[87], initially developed for medical image segmentation, is also commonly used. A UNet consists of down-sampling and up-sampling steps, where the former extracts features by using 3×3 convolutions and the latter reduces the number of feature maps while increasing the dimension. Features from down-sampling are concatenated with those from up-sampling and finally 1×1 convolution generates the segmentation map.

5.2 Methods

Transferring attributes from one domain to a different one serves, without loss of generality, as an example illustrating the proposed model. We will transfer attributes from domain X to an example image in domain Y.

Learning Encoder and Decoder. Images x are passed into an encoder (E) first, resulting in an interpolated latent vector (V) of fixed length, so $V_x = E(x)$. The image from the target domain Y also goes through the encoder, so V_y. Next, an ANN (ann) helps guide the transition from V_x to V_y, and the interpolated latent vector is obtained by $V_I = ann(V_x, V_y)$. Eventually, the decoder (D) generates the interpolated image by $D(V_I)$. The loss for the encoder is

$$L_E = -E_{x\sim X, y\sim Y}(ann(E(x), E(y))). \tag{5.1}$$

To guide the learning of the decoder, real images are compared to reconstructed ones, and reconstruction loss is formulated as

$$L_{recons} = MSE(x, D(E(x))), L_{E,ann} = E_{F\sim P_r}[\mathfrak{D}(F)] - E_{\widetilde{F}\sim P_r}[\mathfrak{D}(F)]. \tag{5.2}$$

Learning Critic. Much like WPGAN[58], the loss function for learning the critic (D) is formulated as

$$L_{\mathfrak{D}} = E_{F\sim P_r}[\mathfrak{D}(F)] - E_{\widetilde{F}\sim P_r}[\mathfrak{D}(F)] + \lambda_{gp} GP, \tag{5.3}$$

where $\widetilde{F} = ann(F_x, F_y)$ is the interpolated feature, $F_x = E(x)$ and $F_y = E(y)$ are extracted features from the encoder (E). GP is the gradient penalty term and λ is defined as[58] did. P_f and P_r are distributions from fake and real feature samples. λ_{gp} is set to 10 for all experiments in this study.

Learning Interpolator. After the encoder projects images into latent space, which is

flat, interpolation can be done linearly as

$$f(F_x, F_y) = F_x + \alpha(F_y - F_x), \tag{5.4}$$

where α controls the interpolation strength.

Because many paths connect the two points in the latent space, naively interpolating linearly might produce blurry images with many artifacts, so an ANN model had the best transition from the two domains. Our interpolation method is formulated as

$$f(F_x, F_y) = F_x + a \times \alpha(F_y - F_x), \tag{5.5}$$

where *ann* ia a learnable CNN.

Learning Shape. To preserve the identity of the source domain, our model incorporated UNet, which outputs a binary map delineating the boundary of leaves, for calculating the shape loss (L_{shape}), which is formulated as

$$L_{shape} = Dice(S_{interpolated}, S_x), \tag{5.6}$$

where $Dice$ is the Dice loss[88]. $S_{interpolated}$ is the UNet output of interpolated image, and S_x is the UNet output of source domain image.

Model Architecture and Training Algorithm. Figure 5.2 shows the proposed model. Input images and target domain images were fed into E first, which produced respective

features. The features were then piped into the interpolator, which guided the transition from source to target domain and produced the interpolated feature. Features from target domain and interpolated feature were processed by the critic, which calculated the $L_{\mathfrak{D}}$ and L_E. The decoder mapped features from source domain images to the original image space, and then L_{recons} was calculated by comparing the reconstructed image and original image. A UNet was also trained to preserve the shape of interpolated images by calculating the Dice loss (L_{shape}). The training procedure is shown in Algorithm **??**.

Algorithm 2: Model Training, $n_{critic} = 5$

Data: Source domain images x_i and target domian images y_i, where $i = 1, 2, ..., N$

Result: encoder $\boldsymbol{E}$, decoder $\boldsymbol{D}$, interpolator $\boldsymbol{I}$, and UNet $\boldsymbol{U}$

initialization;

while *not converged* **do**

    m = 0;

    while $m < n_{critic}$ **do**

        Calculate $L_{\mathfrak{D}}$ and update $\mathfrak{D}$;

        Calculate L_{recon} and update D;

        $m++$;

    Calculate L_E and update E and I;

    Calculate L_{shape} and update U;

5.3 Experiments and Results

We experimented using the PlantVillage tomato leaf data set. This data set contains healthy leaves and diseased leaves, with diseases categorized as bacterial spot, early blight, late blight, mold, septoria spot, spider mites, target spot, and yellow leaf. All images were resized to 128×128. We annotated the segmentation map of leaves for training the UNet module. The baseline model used for comparison was HomoInterpGAN, which achieved impressive results

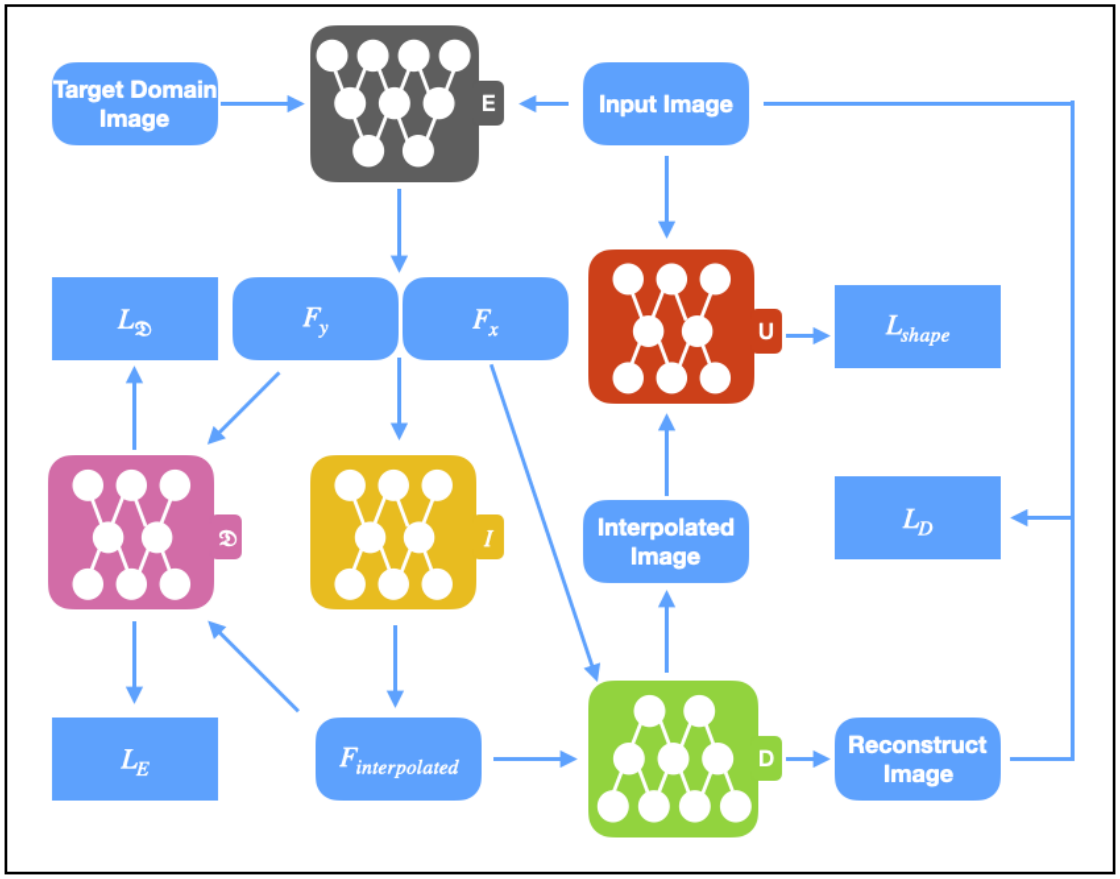


Figure 5.2: *Framework structure: the encoder E maps images of source domain and target domain to their feature space. The interpolator I learns the path from source to target latent space. The decoder D reconstructs the source image from feature space F_x and produces interpolated images from interpolated features. The critic $\mathfrak{D}$ learns how real the interpolated features are. UNet U forces similarity in the shapes of our interpolated images and source images.*

in transferring human-face attributes.

Both our model and the baseline model can interpolate intermediate images from source to target domains. Figure 5.3 shows the results, illustrating the interpolation from source (healthy leaf) to three different target domains (three types of unhealthy leaves) under incremental transition strength. Results show that both models can interpolate in latent space and produce intermediate images from source to target domain. However, interpolation results from HomoInterpGAN could not preserve the shape of source images. We observed some ghosting effects, where the interpolation tries to copy the shape of target domain with

Table 5.1: *Test accuracy on data sets produced by different models*

Model	Accuracy (↑)
baseline(orginal test data)	97.33 %
MUNIT	73.32 %
HomoInterpGAN	95.02 %
StarGAN-V2	94.11 %
Ours	93.87 %

implausible results, as shown in row (a) with transition strengths 0.75 and 1. Row (c) shows another example produced by HomoInterpGAN with transition strength 1 developing another pointy tip, which seems to be inherited from the target leaf. HomoInterpGAN could not preserve shape, which is most obvious in row (e) with transition strengths 0.75 and 1, where the shape of source image was totally transformed into that of the target domain. Image translation results showed that our model can better preserve the shape from the source domain while constraining the results to attributes of the target domain, producing natural-looking interpolation images.

Like the research in[16;24;89], we trained a neural network (ResNet34) as a classifier to test the quality of produced images after image-to-image translation. The trained classifier was tested on the original test data set, and the test set accuracy is treated as the baseline. All models were used to transform healthy leaves into unhealthy ones, and the produced images were then classified. Table 5.1 shows the test set accuracy of the classifier trained on the original data set and on images produced after image-to-image translation. Compared to the baseline testing accuracy (approximately 97%), 95% of the HomoInterpGAN translated images were correctly classified into corresponding categories, and nearly 94% of images translated by our model were successfully identified. StarGAN-V2 performed slightly better than ours, but MUNIT showed the worst results among all models. Although slightly lower in test accuracy than HomoInterpGAN, our model produced images that were more visually satisfying; that is to say, qualitative attributes from the target domain were successfully transferred to the source domain and the shape of source images was preserved.

sectionConclusion We created a new framework for unpaired image-to-image translation, transferring attributes, and producing natural-looking intermediate results. Our model features a UNet module that preserves the shape of translation results. In addition, our model incorporates a neural network especially designed for learning the best path to transit in the latent space. Results showed that both our model and the baseline model can transfer attributes. However, in transforming images, the baseline model naively attempts to copy the shape of target domains, and thus generates ghosting artifacts in the translated results. By contrast, our approach can smoothly transfer attributes and produce more visually appealing results by preserving the shape of source domain without too much trade-off (about 1%) in the quality of translated results.

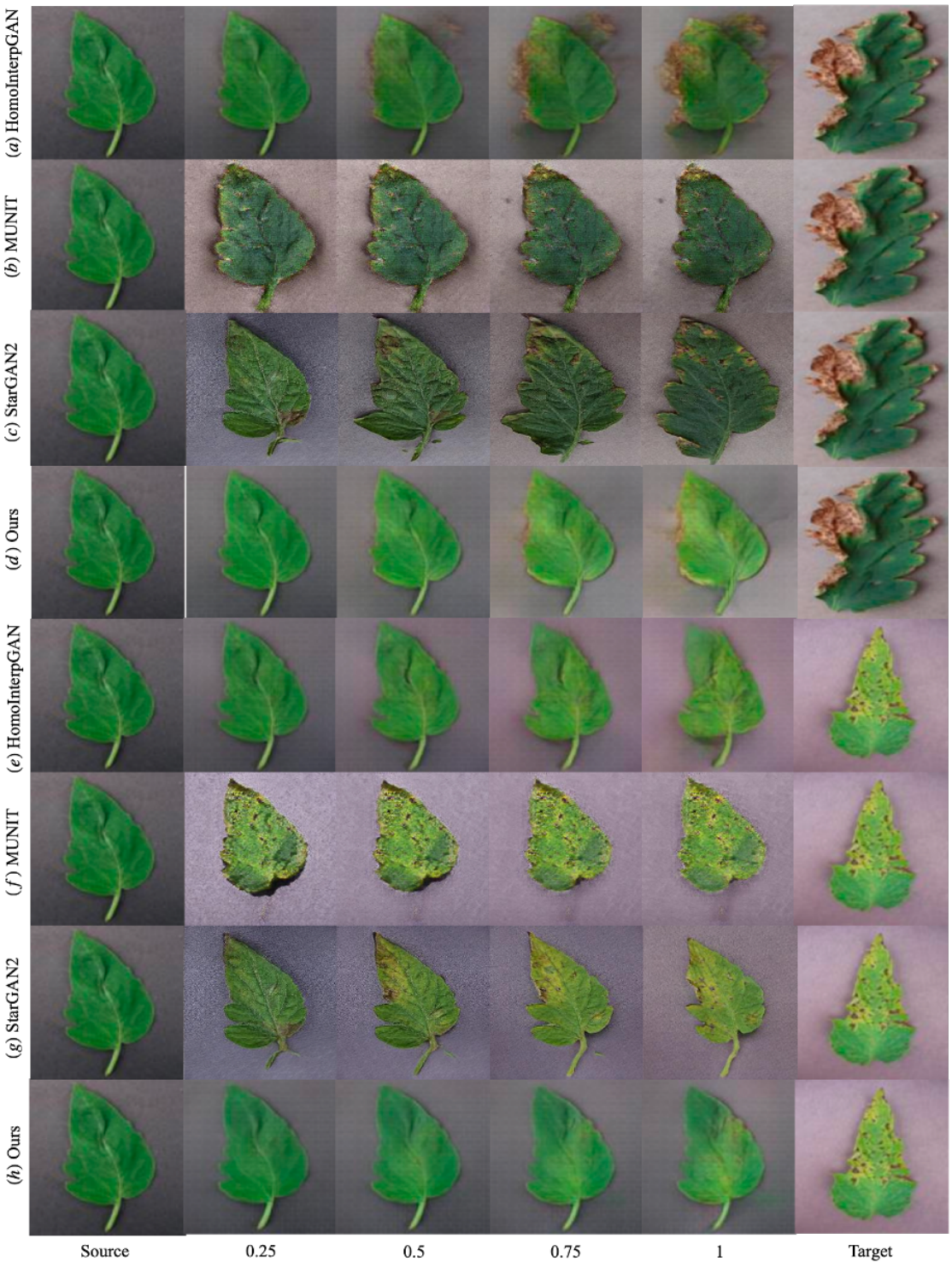


Figure 5.3: *Interpolation results from all models. The source image is the healthy leaf. The target of (a) through (d) is an example with bacterial disease. The target in (e) through (f) has septoria. Interpolation strength ranges from 0.25, 0.5, 0.75, to 1, as shown at the bottom of the figure.*

Chapter 6

Achieve Instance-level Unsupervised I2I with Self-supervised Learning

Image-to-image (I2I) translation refers to translating images from one domain to another featuring different styles that are visually distinctive. An example is the task of turning images of cartoon sketches into real-life photographs. Many tasks in computer vision can be viewed as I2I translation, such as image inpainting[19], style transfer as in StyleGAN2[21], and super-resolution[22]. Supervised I2I translation tasks need paired data sets that are costly to obtain, but such tasks are relatively easier to solve using supervised learning instead of unsupervised. Under paired data supervised learning, I2I translation can be done using regression[23] or conditional generative models[24]. Our work addresses the unsupervised I2I translation task, which is more challenging without access to paired data sets. Most research on unsupervised I2I translation draws inspiration from CycleGAN[25] using the cycle consistency constraint, which has achieved impressive results. More recent studies on such models as MUNIT[28] and StarGAN2[30] have improved upon CycleGAN and can translate images among multiple domains. This research, however, can introduce unwanted changes to both objects of interest and the background. Our research used a simpler yet effective approach.

Our framework consists of only one generator-discriminator pair and a mapping network, enabling multimodal and multi-domain translation. Moreover, our framework learns attention maps through an attention module, which allows translating objects of interest while leaving the background intact. Extensive experiments show that our framework is superior or comparable to SOTA baselines. The contributions of our research can be summarized as follows:

- A novel framework for unsupervised I2I translation with an attention mechanism allows image translation at instance level.
- Our framework uses unsupervised learning for attention maps, which requires no segmentation annotations. Our attention module could be used as a plug-and-play add-on for existing pre-trained I2I translation frameworks, making them capable of learning attention maps at lower cost than training an attention module and its generator from scratch.
- Unlike previous research into such models as MUNIT and DRIT[29] that require training $n(n-1)$ generators for translating images for n domains, our novel framework architecture requires training only one generator-discriminator pair and achieves multimodal, multi-domain I2I translation.
- Extensive experiments using publicly available data sets show that our framework is superior to SOTA baselines.

6.1 Related Work

Generative adversarial networks. Ideally, generative models learn how data is distributed, thus allowing data synthesis from the learned distribution. Since the advent of GANs[70], generative models like StyleGAN2 have achieved impressive results in tasks like

image editing[52] and style transfer. GANs try to learn the data distribution by approximating the similarity of distributions between the training data and the fake data produced by the learned model. GANs usually comprise a generator and a discriminator. The entire model learns by playing a min-max game: the generator tries to fool the discriminator by gradually generating realistic data samples, and the discriminator, in turn, tries to distinguish real samples from fake ones. GANs have improved in many ways. To produce more realistic samples, an architecture of stacked GANs has been proposed: the laplacian pyramid of GANs[54]; layered, recursive GANs[55]; and style-based GANs (StyleGAN and StyleGAN2). Several studies have attempted to solve the instability training of GANs using energy-based GANs[57] and Wasserstein GANs[58]. In this study, we used GANs with their improved techniques to learn the distribution of data and how to translate among different domains.

Unsupervised I2I translation. Unsupervised I2I translation takes images from one domain, translating them to another without paired data supervision. Much success in unsupervised I2I translation is due to the cycle consistency constraint, proposed in earlier research: CycleGAN, DiscoGAN[26], and DualGAN[27]. To translate among more than two domains, MUNIT and DRIT have been used. These methods, however, sample style codes from a standard normal distribution, which leads to inferior translation results. Moreover, they require training $n(n-1)$ generators and n discriminators for translating images among n domains, which is computationally expensive and time-consuming. Our method proposes a simpler yet more effective approach that requires only one generator-discriminator set. Recent systems such as StarGAN2 and ModularGAN[31] were developed to perform multimodal, image-to-image translation to produce images with the same content but different contexts. All the aforementioned methods, however, introduce undesired changes to the background while translating images.

Attention learning. With human attention mechanism as inspiration, attention mechanisms have been successfully applied to various computer vision and natural language processing tasks, such as machine translation[90], visual question answering[91], and image and

video captioning[92]. Attention mechanisms improve the performance of all these tasks by encouraging the model to focus on the most relevant parts of the input. To focus on the most discriminative semantic part and retain the background of images during translation, attention mechanism was been introduced into I2I. ConstrastGAN[93] takes a supervised approach and uses segmentation mask annotations as extra input data. Our approach also learns attention masks without using extra annotation, much like AttentionGAN[94], ATAGAN[95], and AGGAN[65], which add an attention module to each generator to locate the object of interest in image-to-image translation tasks. Thus, the background can be excluded from I2I translation. All these methods, however, can only translate between two domains at a time. To remedy the drawbacks, we created a unified I2I translation framework with an attention mechanism. Instead of having to train $n(n-1)$ generator-discriminator pairs for learning to translate among n domains, our method requires training only one such pair. Thus, our framework reduces training time and memory footprint with better or comparable translations.

6.2 Methods

6.2.1 Preliminaries

Let x be an image belonging to one of many domains. Diagram (a) in Figure 6.1 shows an overview of our model. We start from a latent vector z that is sampled from a standard normal distribution. z goes through a mapping network, which learns style codes s of a specific domain, where m is a domain label and $s = M(z, m)$. Meanwhile, we use a content encoder E_c to extract content codes c from image inputs. The decoder D takes content and style codes to generate reconstructed images x', which are then used by style encoder E_s to produce reconstructed style codes s'. We compute two L1 losses using the reconstructed images and style codes. Finally, we use a multi-task discriminator to distinguish real images

from generated ones. During the translation phase, we keep the same content codes but use the style codes of a target domain. Attention maps are learned using the attention module. Take translating a horse image x_m to a zebra image as an example, as shown in diagram (b) of Figure 6.1. The horse image is processed by the encoder, resulting in style codes s_m and content codes c_m. In the meantime, the attention module extracts attention maps att from the horse image. The style codes of the zebra image s_n are exchanged for those of the horse image. Then, the decoder uses the content codes c_m and style codes s_n to generate an intermediate fake zebra image whose background contains unwanted changes. We incorporate the attention map with the intermediate fake zebra image by $att \times D(c_m, s_n) + (1 - att) \times x_m$, which results in the final fake zebra image. Note that we only show the attention branch for translating horse to zebra because of space limitation. The other direction of translation follows a similar process.

6.2.2 Framework Architecture

In this section, we outline the architecture of different modules in our framework.

Encoder. Our encoder has two sub-encoders: the style encoder and the content encoder. Both start with a convolution layer. The content encoder consists of six residual blocks[96]. All layers are downsampled by average pooling operation (except for the last two layers) and are followed by instance normalization (IN)[97]. The style encoder also comprises six residual blocks without any activation function except for the last residual block. Lastly, the style encoder consists of a convolution layer with leaky ReLU and a reshape operation before outputting style codes by way of the linear layer.

Mapping network. Style codes of domains are modelled by a mapping network, which consists of eight linear layers with ReLU activation functions except for the last layer.

Decoder. The decoder maps latent codes, which consist of style codes and content

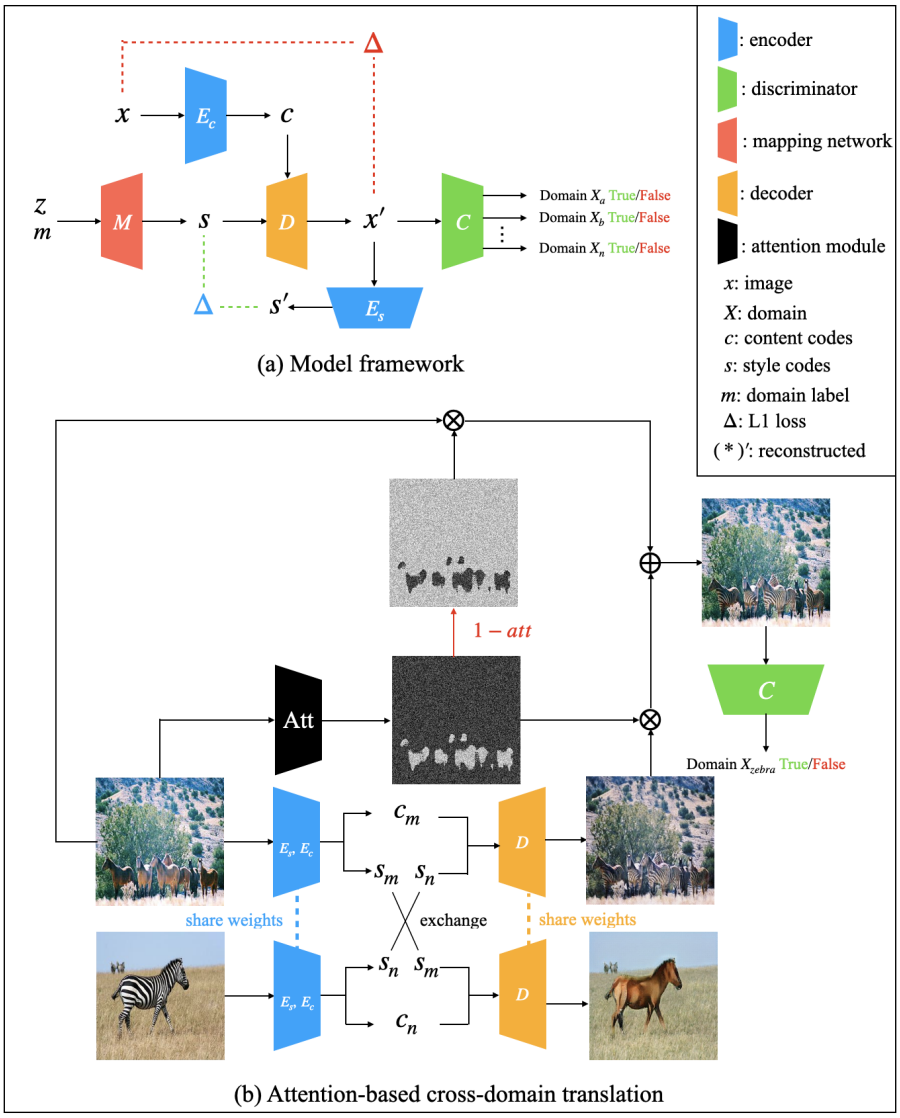


Figure 6.1: *The structure of our framework. (a) shows how our framework learns, and (b) shows cross-domain translation within the horse and zebra domain. The attention branch of translating zebra2horse is similar to horse2zebra, and thus is not shown.*

codes, to the original image space. To apply style to images of different domains, the style codes are injected into the decoder by AdaIN[78] coupled with residual blocks. The last layer is a convolution layer that generates images as outputs.

Attention module. The attention module has an encoder-decoder architecture. The encoder consists of three convolutional blocks, and the decoder has three convolutional layers with a sigmoid activation function at the end that outputs the attention probability map.

Discriminator. The architecture of the discriminator is similar to the style encoder except it has one more convolutional layer to predict domains.

6.2.3 Training Objectives

In this section, we discuss the loss functions for learning our framework.

Image reconstruction loss. After images are encoded to style and content codes, the decoder maps the latent space back to the image space and reconstructs the image. Image reconstruction loss is formulated as

$$L_{recon}^{x} = \|D(E_c(x), M(z, m)) - x\|_1 , \tag{6.1}$$

where m is the domain to which image x belongs.

Style code reconstruction loss. After encoding reconstructed images using the style encoder, we can obtain reconstructed style codes. We constructed the style code reconstruction loss as follows:

$$L^s_{recon} = \|s - E_s(x')\|_1 , \tag{6.2}$$

where $x' = D(E_c(x), M(z, m))$, and $x \in X_m$.

Attention consistency loss. Images before and after translation should have the same attention maps. Thus, the attention consistency loss is defined as

$$L_{att} = \|att(x_{mn}) - att(x_m)\|_1 , \tag{6.3}$$

where x_{mn} is the translated image, which is obtained by $att \times D(c_m, s_n) + (1 - att) \times x_m$. c_m is the content information of x_m, and s_n is the style information of image x_n.

Regularization on style and content codes. To further encourage style codes being domain-variant and content codes being domain-invariant, we added regularizers to style and content encoders. The style regularizer forces style codes of different domains to differ by minimizing L^s_{regu}, which is calculated as

$$L^s_{regu} = - \|D(c_m, s_m) - D(c_m, s_n)\|_1 - \|D(c_n, s_m) - D(c_n, s_n)\|_1 , \tag{6.4}$$

where $(c_m, s_m) = (E_c(x_m), E_s(x_m))$ and $(c_n, s_n) = (E_c(x_n), E_s(x_n))$. c_m and s_m are content and style codes of image $x_m \in X_m$. c_n and s_n are content and style codes of image $x_n \in X_n$.

The content regularizer encourages content codes of different domains to be similar by minimizing L^c_{regu}, which is formulated as

$$L^c_{regu} = \|D(c_m, s_m) - D(c_n, s_m)\|_1 + \|D(c_m, s_n) - D(c_n, s_n)\|_1 . \tag{6.5}$$

Inspired by StarGAN2, we calculated style diversity as

$$L_{ds} = \|E_s(x_1) - E_s(x_2)\|_1 , \tag{6.6}$$

where $x_1 = D(E_c(x), M(z_1, m))$, and $x_2 = D(E_c(x), M(z_2, m))$, and z_1 and z_2 are two random latent vectors.

Adversarial loss. GANs were used to match the distribution of translated results to real image samples, so the discriminator finds real and fake samples indistinguishable. We used two adversarial losses with one for learning latent-guided translation and the other for reference-guided translation. Latent-guided translation refers to using the mapping network to obtain target style codes, and reference-guided translation uses the style encoder to extract style codes of target domains. The adversarial loss for learning the discriminator C_m with latent-guided translation is formulated as

$$L^l_{adv} = \mathop{\mathbb{E}}_{z\sim N(0,I), x_n\sim p(X_n)}[logC_m(D(E_c(x_n), M(z, m)))] + \mathop{\mathbb{E}}_{x_m\sim p(X_m)}[log(1 - C_m(x_m))], \tag{6.7}$$

and the adversarial loss for learning the discriminator C_m with reference-guided translation is constructed as

$$L^r_{adv} = \mathop{\mathbb{E}}_{x_m\sim p(X_m), x_n\sim p(X_n)}[logC_m(x_{nm})] + \mathop{\mathbb{E}}_{x_m\sim p(X_m)}[log(1 - C_m(x_m))], \tag{6.8}$$

where the discriminator C_m tries to identify if images are from the domain m, and x_{nm} is obtained by $att \times D(c_n, s_m) + (1 - att) \times x_n$.

Full objective. Our full objective is formulated as follows:

$$\begin{aligned} \min_{M,E,D} \max_{C} \; & \lambda_1 L^x_{recon} + \lambda_2 L^s_{recon} + \lambda_3 (L^s_{regu} + L^c_{regu}) \\ & + \lambda_4 (L^l_{adv} + L^r_{adv}) - \lambda_5 L_{ds} + \lambda_6 L_{att}, \end{aligned} \tag{6.9}$$

where λ_1 to λ_6 are hyperparameters for each loss term.

Model training scheme. We found the model had difficulty in converging when the generator and the attention module were trained simultaneously. Therefore, we first trained the generator and the discriminator using $1e^{-4}$ as the learning rate for 100,000 iterations, which is empirically calibrated. Then, we froze the parameters of the generator when training the attention module for 30,000 iterations with the same learning rate. Lastly, we jointly trained the entire framework for another 10,000 iterations using a smaller learning rate $5e^{-5}$.

6.3 Experiments

This section presents the data sets, baselines, and evaluation metrics.

Baselines and data set. We compared our framework to four baseline models developed in recent years. CycleGAN is pioneering unsupervised I2I, and used as a baseline model. MUNIT and StarGAN2 achieve impressive results in unsupervised multimodal I2I translation, making it ideal to compare to our framework. We also compared our approach to AGGAN, a recent attention-based I2I translation framework.

We evaluated our framework on the *horse2zebra*, *AFHQ*, and *map2aerial* data sets. The

horse2zebra data set contains images of horses and zebras, downloaded from ImageNet using keywords wild horse and zebra. A total of 1,067 horse images and 1,334 zebra images were used for training, with 120 horse images and 140 zebra images reserved for testing. The *AFHQ* data set contains images of house cats, dogs, and wild animals (e.g., tigers, foxes, and lions). As with StarGAN2, we divided the *AFHQ* data set into domains of cats, dogs, and wild animals. The *map2aerial* data set was scraped from Google Maps, and images were sampled from in and around New York City. All images are 256×256.

Evaluation metrics. We evaluated the visual quality of translation using AMT, in which user preferences determine the results of different models. Quantitative measures without human participation, Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR), were used as in Chen et al. in AttentionGAN and AGGAN.

6.4 Results and Discussion

This section provides the qualitative and quantitative results of the experiments. An ablation study was also used to evaluate the effectiveness of several key design choices.

Qualitative results. We used AMT to compare our results against the baselines. Given a source image and a reference image, we instructed AMT participants to select the best translation among all models. We asked 50 questions of all ten participants. Table 6.1 shows our method outperformed all baseline models, especially MUNIT, CycleGAN, and StarGAN2, which are not attention-based I2I translation frameworks. Like MUNIT and StarGAN2, our model can perform latent-guided and reference-guided translation. Figure 6.2 (a) illustrates latent-guided translation, and Figure 6.3 shows I2I translations guided by reference images from all models. Our model and AGGAN can both preserve the background information, translating only the images of interest. CycleGAN and AGGAN can only perform reference-guided translation, so their latent-guided translation results are not shown.

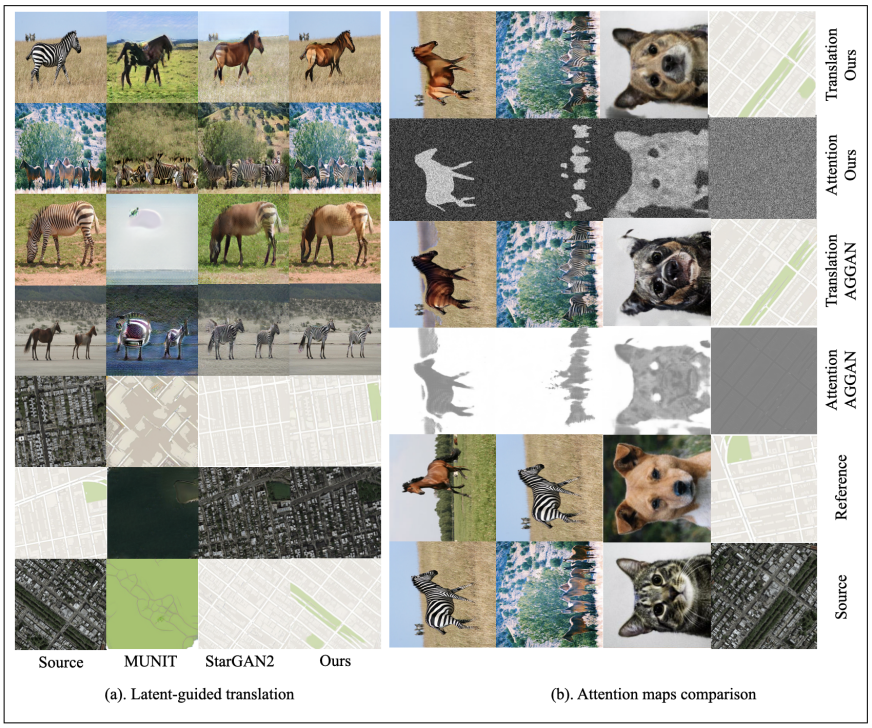


Figure 6.2: *(a) provides examples of latent-guided I2I translation results, and (b) compares attention maps generated by our framework and AGGAN.*

Models	User Preference (↑)
CycleGAN	8.31 %
MUNIT	2.55 %
StarGAN2	3.13 %
AGGAN	40.93 %
Ours	**45.08 %**

Table 6.1: *Votes from AMT participants for preferred translation results.*

Figure 6.2 (b) offers two examples of attention maps from our model and AGGAN, showing that our attention maps are more accurate than AGGAN. Given these results, we argue that "undesired changes" requires a clear definition. Clearly, in translations, such as transferring a map into an aerial photograph, we would assume the attention map is the entire image (See the attention map in figure (b) of Figure 6.2). It may be more appropriate to apply a separation of background from one domain from the background of another: *horse2zebra* instead of *map2aerial*.

Quantitative results. Like MUNIT and StarGAN2, our model can perform latent-

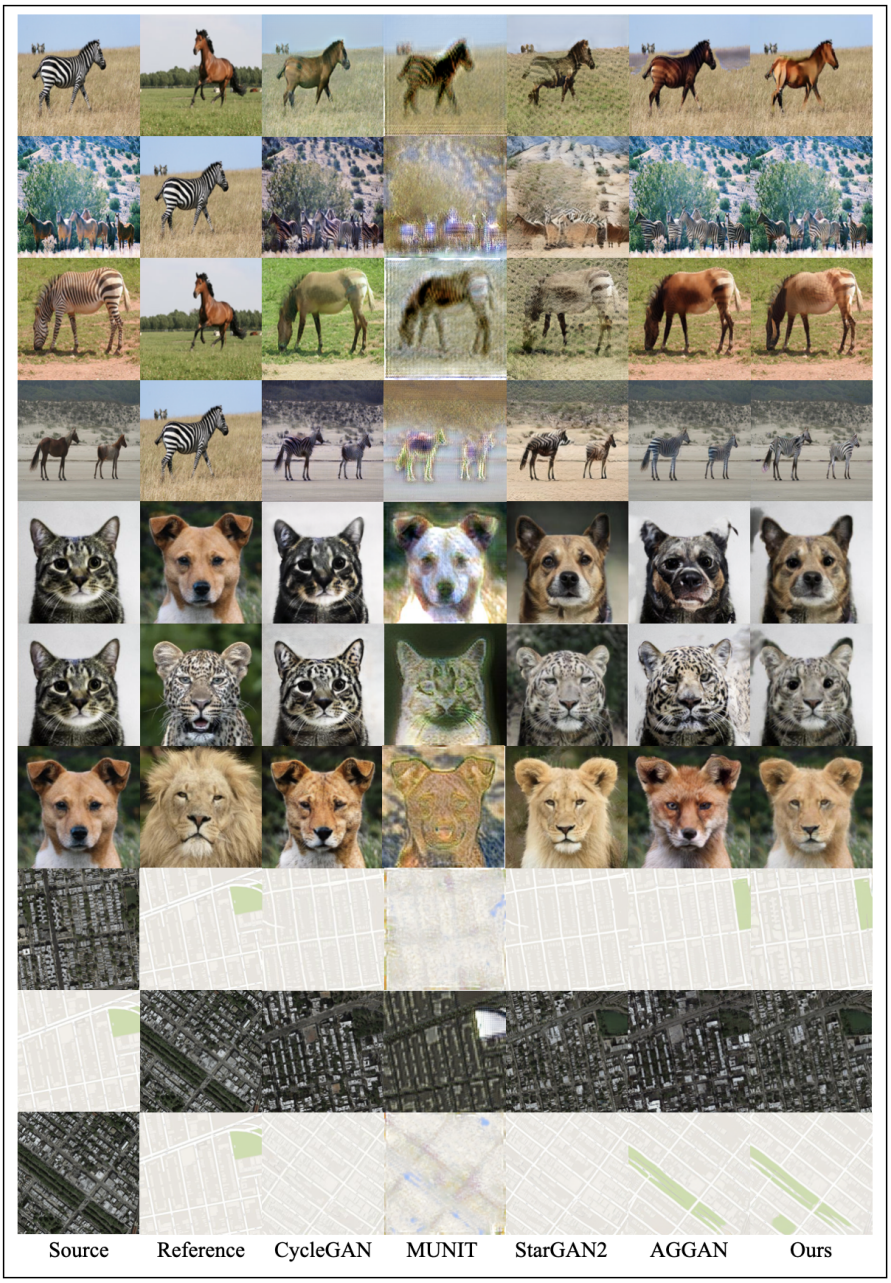


Figure 6.3: *Examples of reference-guided I2I translation by different models.*

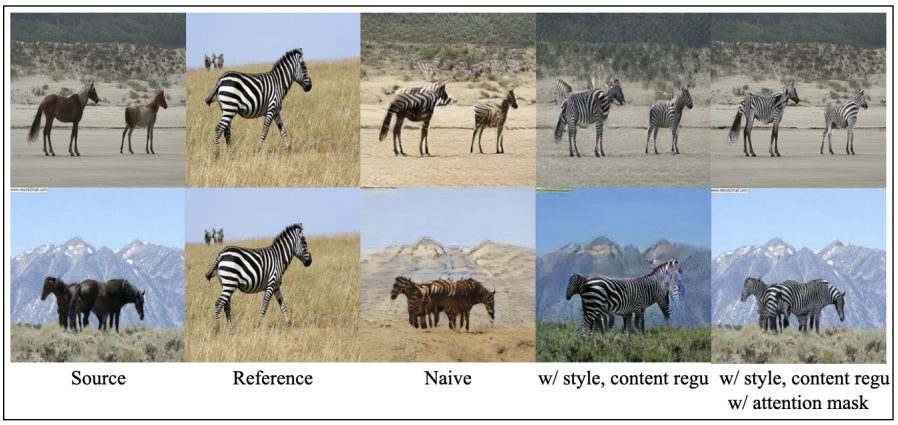


Figure 6.4: *An example of reference-guided translation by incrementally adding modules.*

Models	*horse2zebra*		*zebra2horse*	
	SSIM(↑)	PSNR (↑)	SSIM(↑)	PSNR (↑)
CycleGAN	0.7313	21.96	0.8453	26.31
MUNIT	0.1176	14.89	0.3664	15.29
StarGAN2	0.3281	16.86	0.4729	19.43
AGGAN	0.9686	33.16	0.9843	43.02
Ours	**0.9699**	**36.12**	**0.9851**	**44.11**

Table 6.2: *Quantitative comparison on reference-guided translation.*

guided and reference-guided translation. We evaluated all models using SSIM and PSNR, which require ground truth attention maps of images. As for AttentionGAN, we obtained attention maps using the DeepLab semantic image segmentation model[98] pretrained on MSCOCO[99] data set. Note that we provided only quantitative results on the *horse2zebra* data set because the DeepLab model was not trained on the *map2aerial* data set, and no ground truth attention maps were available for calculating SSIM and PSNR. Tables 6.2 and 6.3 show that our framework outperformed by a large margin all baseline models, especially CycleGAN, MUNIT, and StarGAN2. Again, CycleGAN and AGGAN cannot perform latent-guided translation. Therefore, quantitative results for these two models were not reported.

Models	*horse2zebra*		*zebra2horse*	
	SSIM(↑)	PSNR (↑)	SSIM(↑)	PSNR (↑)
MUNIT	0.1925	11.66	0.3901	13.88
StarGAN2	0.3353	18.87	0.4953	19.92
Ours	**0.9712**	**33.76**	**0.9857**	**43.14**

Table 6.3: *Quantitative comparison on latent-guided translation.*

Modules	SSIM (↑)	PSNR (↑)
naive model	0.3062	12.73
+ style, content regularizer	0.3511	19.04
+ attention masks	**0.9699**	**36.12**

Table 6.4: *SSIM and PSNR results of incrementally adding modules to our framework for reference-guided translation on the horse2zebra data set.*

Ablation studies. To further validate the effects of key design choices in our framework, we carried out ablation studies on the *horse2zebra* data set (see Table 6.4 and Figure 6.4). Let the model without style, content regularizer, and attention module be the naive model. We can see that adding attention greatly improved translation results.

6.5 Conclusions

In this research, we created a simple, yet effective, attention-based framework for unsupervised I2I translation. Our framework not only translates objects of interest while leaving the background unaltered, but also generates images for multiple domains simultaneously. Unlike similar research on unsupervised I2I with an attention mechanism that requires ground truth for learning attention maps, our approach uses unsupervised learning. The qualitative and quantitative results show that our framework is superior to the SOTA baseline models.

Chapter 7

Conclusions and Future Work

In this chapter, I present a review of claims in Section 7.1 and summarize my finding in Section 7.2. In the end, I present my suggestions for future work in Section 7.3.

7.1 Review of Claims

Through the methodology chapters (3 to 6), user studies were conducted for qualitative evaluation on results of models. The standard practice in the field of I2I is that survey participants are asked to choose the model that produced the best translation result. Their opinions are then summarized to show which model was superior. Further statistical analysis might be needed to evaluate the statistical significance of such survey results. This practice, which would improve the state of the field in terms of validation and reproducibility, would involve both cross-validation image datasets and interannotator agreement.

In the discussion of autoencoders in Chapter 2, I mentioned that VAEs are able to model internal attributes of data by means of learning series of normal distributions. The learned attributes are implicit, meaning that it might be difficult to map learned attributes

to the physical world. For example, hidden attribute a controls the strength of smiling. Another use case is applying VAEs to hyperspectral images, which are commonly used data in the field of remote sensing. There is little difference, conceptually, on applying VAEs on hyperspectral images than regular RGBA images. Because inside of ConvNets, there are potentially thousands of convolution layers, which can be thought of a hyperspectral image. In practice, however, it might be difficult to deal with hyperspectral images because they tend to be too large to fit in GPU VRAM, or they will be limited to train on a small batch size.

Some methods proposed in this dissertation are use-case dependent. For example, in Chapter 5, I proposed a shape-aware GAN framework to model what tomato leaves will look like if they are 25%, 50%, or 75% unhealthy. Being able to perform partial translation might be helpful for studying disease progression for the case of tomato leaves.

In Chapter 6, I employed an attention mechanism to separate foreground and background objects. Such separation might make more sense for cases like *horse2zebra* than the case of *map2aerial*, because we would like to apply the style of Google Maps to aerial photos in the entire region rather than part of it. Another thing to be noted is that learning to separate foreground and background objects might be harder for the *horse2zebra* case, because horses and zebras appear different sizes and ConvNets are not scale-invariant. That means the model needs to adapt to objects of different sizes. In contrast, the attention module simply focuses on the entire region for the case of *map2aerial*.

7.2 Summary

This dissertation studies the problem of unsupervised I2I translation by using generative models. I present the problem and related literature reviews in Chapter 1 and 2. In each of Chapters 3 to 6, I show some drawbacks in existing methods and present ways to improve the

shortcoming. I show by extensive experiments that my approaches surpass or achieve comparable performance than SOTA methods. This dissertation was organized in the following logical order:

- Chapter 1 defines the problem of I2I translation and why it is of significance to study it.
- Chapter 2 reviews related literature on I2I in detail, mainly focusing on VAEs and GANs. I also pointed out the limitations of drawbacks of existing works, upon which my methodology chapters are built.
- Studies in Chapter 3 is accepted for publication at International Journal of Machine Learning and Computing. Chapter 3 proposes a simplified framework for I2I translation, but achieves better translation results. There are two biggest contributions. First, there is no need to train $n(n-1)$ generators and discriminators in order to learn to translate among n domains. Second, adding a mapping network to predict general style information of a domain in addition to sampling from a standard distribution.
- Chapter 4 is published at the 30th International Conference on Artificial Neural Networks (ICANN) 2021. Chapter 4 improves upon Chapter 3 by adding fine-grained control over latent space. My findings suggest that domain-related information that still exists in content codes, and translation results can be improved by excluding them. Another contribution of Chapter 4 is adding a interpolator to guide the exchange of style codes because the process might not be linear.
- Chapter 5 is published at the 13th International Conference on Machine Vision (ICMV) 2020. Chapter 3 and Chapter 4 lay the ground for Chapter 5 and 6, in which Chapter 5 introduces the idea of foreground-background separation, and how to perform such separation in a semi-supervised fashion. I use an example of tomato leaves and add a shape-regularization module (a UNet) to constrain the shape of leaves after translation.

- Chapter 6 is under review for ICANN 2022. Chapter 6 proposes a framework that inherit the ideas proposed in previous chapters, and introduces a totally unsupervised framework for learning I2I translation and foreground-background separation.

7.3 Future Work

Given my findings as documented in Chapters 3 to 6, one direction of future work that I view as most promising and worth while is to develop a standalone, one-suits-all model that learns segmentation masks. This new model is able to utilize existing I2I translation frameworks and learn foreground and background without retrain the generator. Furthermore, it can be plugged into arbitrary I2I translation frameworks and learns segmentation at minimum costs. Visual transforms have achieved some promising results. Another direction of future research is to use visual transformers in the architecture of generators.

Bibliography

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, 2017.

[2] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–187, September 2018.

[3] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2416, 2019.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[7] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. URL http://arxiv.org/abs/1504.08083.

[8] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *CoRR*, abs/1807.05511, 2018. URL http://arxiv.org/abs/1807.05511.

[9] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. URL http://arxiv.org/abs/1506.02640.

[10] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. URL http://arxiv.org/abs/1612.08242.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. URL http://arxiv.org/abs/1812.04948.

[14] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv e-prints*, art. arXiv:1610.09585, Oct 2016.

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL http://arxiv.org/abs/1710.10196.

[16] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.

[17] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and A. Cristiano I. Malossi. BAGAN: data augmentation with balancing GAN. *CoRR*, abs/1803.09655, 2018. URL http://arxiv.org/abs/1803.09655.

[18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. URL http://arxiv.org/abs/1606.03657.

[19] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in neural information processing systems*, pages 331–340, 2018.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2019.

[22] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[23] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[25] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.

[26] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks, 2017.

[27] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation, 2017.

[28] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation, 2018.

[29] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations, 2018.

[30] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains, 2019.

[31] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks, 2018.

[32] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075. doi: 10.1126/science.1127647. URL https://science.sciencemag.org/content/313/5786/504.

[33] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[35] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[36] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2654–2663. PMLR, 2018. URL http://proceedings.mlr.press/v80/kim18b.html.

[37] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2615–2625, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/1ee3dfcd8a0645a25a35977997223d22-Abstract.html.

[38] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017. URL http://arxiv.org/abs/1706.02262.

[39] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

[40] Romain Lopez, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6117–6128, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/9a96a2c73c0d477ff2a6da3bf538f4f4-Abstract.html.

[41] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer G. Dy, and Jan-Willem van de Meent. Structured disentangled representations. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 2525–2534. PMLR, 2019. URL http://proceedings.mlr.press/v89/esmaeili19a.html.

[42] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

[43] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015. URL http://arxiv.org/abs/1511.05644.

[44] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. The variational fair autoencoder. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.00830.

[45] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taïga, Francesco Visin, David Vázquez, and Aaron C. Courville. Pixelvae: A latent variable model for natural images. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJKYvt5lg.

[46] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In

Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4790–4798, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/b1301141feffabac455e1f90a7de2054-Abstract.html.

[47] Yookoon S. Park, Chris Dongjoo Kim, and Gunhee Kim. Variational laplace autoencoders. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5032–5041. PMLR, 2019. URL http://proceedings.mlr.press/v97/park19a.html.

[48] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

[49] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/d523773c6b194f37b938d340d5d02232-Abstract.html.

[50] Alireza Makhzani and Brendan J. Frey. Pixelgan autoencoders. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: An-*

nual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 1975–1985, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7e7e69ea3384874304911625ac34321c-Abstract.html.

[51] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 708–718, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/b9228e0962a78b84f3d5d92f4faa000b-Abstract.html.

[52] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. 2016.

[53] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space?, 2019.

[54] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks, 2015.

[55] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation, 2017.

[56] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

[57] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network, 2016.

[58] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[59] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks, 2017.

[60] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BJlZ5ySKPH.

[61] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3697–3707, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/4e87337f366f72daa424dae11df0538c-Abstract.html.

[62] Zhiqiang Shen, Mingyang Huang, Jianping Shi, Xiangyang Xue, and Thomas S. Huang. Towards instance-level image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3683–3692. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00380. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Shen_Towards_Instance-Level_Image-To-Image_Translation_CVPR_2019_paper.html.

[63] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.322. URL https://doi.org/10.1109/ICCV.2017.322.

[64] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206

of *Lecture Notes in Computer Science*, pages 167–184. Springer, 2018. doi: 10.1007/978-3-030-01216-8_11. URL https://doi.org/10.1007/978-3-030-01216-8_11.

[65] Hao Tang, Dan Xu, Nicu Sebe, and Yan Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pages 1–8. IEEE, 2019. doi: 10.1109/IJCNN.2019.8851881. URL https://doi.org/10.1109/IJCNN.2019.8851881.

[66] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

[67] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

[68] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

[69] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018.

[70] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[71] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017.

[72] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[73] Saining Xie and Zhuowen Tu. Holistically-nested edge detection, 2015.

[74] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8012–8021. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00820. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Wu_TransGaGa_Geometry-Aware_Unsupervised_Image-To-Image_Translation_CVPR_2019_paper.html.

[75] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX*, volume 12354 of *Lecture Notes in Computer Science*, pages 319–345. Springer, 2020. doi: 10.1007/978-3-030-58545-7_19. URL https://doi.org/10.1007/978-3-030-58545-7_19.

[76] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 10550–10559. IEEE, 2019. doi: 10.1109/ICCV.2019.01065. URL https://doi.org/10.1109/ICCV.2019.01065.

[77] Hsin-Yu Chang, Zhixiang Wang, and Yung-Yu Chuang. Domain-specific mappings for generative adversarial style transfer, 2020.

[78] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[79] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[81] Yoshua Bengio, Grégoire Mesnil, Yann N. Dauphin, and Salah Rifai. Better mixing via deep representations. *CoRR*, abs/1207.4404, 2012. URL http://arxiv.org/abs/1207.4404.

[82] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. URL http://arxiv.org/abs/1411.4038.

[83] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. doi: 10.1109/TPAMI.2017.2699184.

[84] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. *CoRR*, abs/1502.03240, 2015. URL http://arxiv.org/abs/1502.03240.

[85] Alexander G. Schwing and Raquel Urtasun. Fully connected deep structured networks. *CoRR*, abs/1503.02351, 2015. URL http://arxiv.org/abs/1503.02351.

[86] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. *CoRR*, abs/1509.02634, 2015. URL http://arxiv.org/abs/1509.02634.

[87] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[88] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.

[89] Lei Luo, William Hsu, and Shangxian Wang. Data augmentation using generative adversarial networks for electrical insulator anomaly detection. In *Proceedings of the 2020 2nd International Conference on Management Science and Industrial Engineering*, MSIE 2020, page 231–236, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377065. doi: 10.1145/3396743.3396790. URL https://doi.org/10.1145/3396743.3396790.

[90] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

[91] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 21–29. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.10. URL https://doi.org/10.1109/CVPR.2016.10.

[92] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/xuc15.html.

[93] Xiaodan Liang, Hao Zhang, Liang Lin, and Eric P. Xing. Generative semantic manipulation with mask-contrasting GAN. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217

of *Lecture Notes in Computer Science*, pages 574–590. Springer, 2018. doi: 10.1007/978-3-030-01261-8_34. URL https://doi.org/10.1007/978-3-030-01261-8_34.

[94] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pages 167–184. Springer, 2018. doi: 10.1007/978-3-030-01216-8_11. URL https://doi.org/10.1007/978-3-030-01216-8_11.

[95] Dimitris Kastaniotis, Ioanna Ntinou, Dimitrios Tsourounis, George Economou, and Spiros Fotopoulos. Attention-aware generative adversarial networks (ata-gans). In *13th IEEE Image, Video, and Multidimensional Signal Processing Workshop, IVMSP 2018, Aristi Village, Zagorochoria, Greece, June 10-12, 2018*, pages 1–5. IEEE, 2018. doi: 10.1109/IVMSPW.2018.8448850. URL https://doi.org/10.1109/IVMSPW.2018.8448850.

[96] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[97] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, 2017.

[98] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4): 834–848, 2018. doi: 10.1109/TPAMI.2017.2699184. URL https://doi.org/10.1109/TPAMI.2017.2699184.

[99] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, edi-

tors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.