UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Exploring Multi-Domain and Multi-Modal Representations for Unsupervised Image-to-Image Translation

Yahui Liu

Advisor

Dr. Bruno Lepri

Fondazione Bruno Kessler

Co-Advisor

Prof. Dr. Nicu Sebe

Università degli Studi di Trento

May 2022

Abstract

Unsupervised image-to-image translation (UNIT) is a challenging task in the image manipulation field, where input images in a visual domain are mapped into another domain with desired visual patterns (also called styles). An ideal direction in this field is to build a model that can map an input image in a domain to multiple target domains and generate diverse outputs in each target domain, which is termed as multi-domain and multi-modal unsupervised image-to-image translation (MMUIT). Recent studies have shown remarkable results in UNIT but they suffer from four main limitations: (1) State-of-the-art UNIT methods are either built from several two-domain mappings that are required to be learned independently or they generate low-diversity results, a phenomenon also known as model collapse. (2) Most of the manipulation is with the assistance of visual maps or digital labels without exploring natural languages, which could be more scalable and flexible in practice. (3) In an MMUIT system, the style latent space is usually disentangled between every two image domains. While interpolations within domains are smooth, interpolations between two different domains often result in unrealistic images with artifacts when interpolating between two randomly sampled style representations from two different domains. Improving the smoothness of the style latent space can lead to gradual interpolations between any two style latent representations even between any two domains. (4) It is expensive to train MMUIT models from scratch at high resolution. Interpreting the latent space of pre-trained unconditional

GANs can achieve pretty good image translations, especially high-quality synthesized images (e.g., 1024x1024 resolution). However, few works explore building an MMUIT system with such pre-trained GANs.

In this thesis, we focus on these vital issues and propose several techniques for building better MMUIT systems. First, we base on the content-style disentangled framework and propose to fit the style latent space with Gaussian Mixture Models (GMMs). It allows a well-trained network using a shared disentangled style latent space to model multi-domain translations. Meanwhile, we can randomly sample different style representations from a Gaussian component or use a reference image for style transfer. Second, we show how the GMM-modeled latent style space can be combined with a language model (e.g., a simple LSTM network) to manipulate multiple styles by using textual commands. Then, we not only propose easy-to-use constraints to improve the smoothness of the style latent space in MMUIT models, but also design a novel metric to quantitatively evaluate the smoothness of the style latent space. Finally, we build a new model to use pretrained unconditional GANs to do MMUIT tasks.

Keywords

Generative adversarial networks (GANs), image-to-image translation, multi-domain image translation, multi-modal image translation, unsupervised learning, image manipulation

Contents

List of Tables

List of Figures

Publications

This thesis consists of the following publications:

- Chapter 2:
  - **Yahui Liu**, Marco De Nadai, Jian Yao, Nicu Sebe, Bruno Lepri, Xavier Alameda-Pineda. GMM-UNIT: Unsupervised Multi-Domain and Multi-Modal Image-to-Image Translation via Attribute Gaussian Mixture Modeling, *arXiv:2003.06788*, 2020.

- Chapter 3:
  - **Yahui Liu**, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe What to Change: A Text-guided Unsupervised Image-to-Image Translation Approach. *Proceedings of ACM International Conference on Multimedia (ACM MM)*, 2020.

- Chapter 4:
  - **Yahui Liu**, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the Disentangled Latent Style Space for Unsupervised Image-to-Image Translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

- Chapter 5:
  - **Yahui Liu**, Yajing Chen, Linchao Bao, Nicu Sebe, Bruno Lepri, Marco De Nadai. ISF-GAN: An Implicit Style Function for High-Resolution Image-to-Image Translation, *IEEE Transactions on Multimedia (TMM)*, 2022.

The following papers are published during the course of the Ph.D but not included in this thesis:

1. Yixuan Su, Tian Lan, **Yahui Liu**, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong and Nigel Collier. Language Models Can See: Plugging Visual Controls in Text Generation. arXiv 2022.

2. **Yahui Liu**, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient Training of Visual Transformers with Small Datasets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

3. Jiannan Xiang*, **Yahui Liu*** (equal contributions), Deng Cai, Huayang Li, Defu Lian and Lemao Liu. Assessing Dialogue Systems with Distribution Distances. In *Findings of the Association for Computational Linguistics*, 2021.

4. Lei Ding, Hao Tang, **Yahui Liu**, Yilei Shi, Xiao Xiang Zhu, Lorenzo Bruzzone. Adversarial Shape Learning for Building Extraction in VHR Remote Sensing Images. *IEEE Transactions on Image Processing (TIP)*, 2021.

5. Pierfrancesco Ardino, **Yahui Liu**, Elisa Ricci, Bruno Lepri, and Marco De Nadai. Semantic-Guided Inpainting Network for Complex Urban Scenes Manipulation. *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2020.

6. **Yahui Liu***, Raul Gomez* (equal contributions), Marco De Nadai, Dimosthenis Karatzas, Nicu Sebe, and Bruno Lepri. Retrieval Guided Unsupervised Multi-domain Image-to-Image Translation. *Proceedings of ACM International Conference on Multimedia (ACM MM)*, 2020.

7. **Yahui Liu**, Marco De Nadai, Gloria Zen, Nicu Sebe, and Bruno Lepri. Gesture-to-Gesture Translation in the Wild via Category-Independent Conditional Maps. *Proceedings of ACM International Conference on Multimedia (ACM MM)*, 2019.

Chapter 1

Introduction

1.1 Unsupervised Image-to-image Translation

Unsupervised image-to-image translation (UNIT) aims at learning a mapping between several visual domains from unpaired training images, which has significant influence on many real-world applications where data are expensive, or impossible to obtain and to annotate. Such techniques can be potentially widely used in various scenarios, such as increase the resolution of images [29], image inpainting [100, 141, 142], style transfer [50, 150], video generation [114], domain adaptation [139, 107] and so on. Recently, we have witnessed the booming development of learning such translations in an unsupervised manner without using any annotated data (i.e., unpaired data) [149, 49, 24, 25, 78].

In image-to-image translation, an image domain refers to a set of images sharing some distinctive visual pattern, also called *style*. For example, we can group face images based on the gender of people, or we can group animal images by the categories (e.g., cat, dog, horse, zebra). Meanwhile, the styles can be with multiple appearance modes within each domain. Researchers in this community have been looking forward to seeing a model that can achieve multi-modal and multi-domain unsupervised image-to-image translation (MMUIT). To achieve this goal, there are at least four

necessary and challenging issues that should be solved as follows:

- The generated images should look as similar as possible to the realistic images. For example, there should be fewer artifacts in the generated images, and both the image quality and resolution should be higher.
- The translated images are well lie in the target image domain, which can be distinguished with correct styles by human and automatic evaluators.
- We can train a single model once on an unpaired dataset to achieve the translations among multiple domains. Especially, it is better to avoid training a model between every two domains for computation and memory efficiency reasons. In this manner, translating multiple styles in one step is feasible and convenient.
- The model can generate multi-modal (or diverse) outputs. In another word, it can map an image to multiple stochastic results in a target target domain. To our supervise, a multi-modal model sometimes allows generating smooth interpolations between two sampled outputs.

To achieve these goals in a single model, using a latent variable to represent the specific style is necessary. Thus, we need to well model the style representations of each image domain in a shared latent space. Following the path of previous work, the idea of disentangling content and style representations from the input images shows a promising direction for building an ideal MMUIT model. Intuitively, when we use shared latent space to model all style representations of different image domains. A cluster of style representations of one image domain must be separatable from another cluster of a different image domain. Otherwise, in a mass latent space, it is unstable for the MMUIT model to transfer the input images to the target domain with the desired styles.

Therefore, we focus on these aforementioned issues and propose several interesting methods to extend both the theories and applications in the MMUIT field. In general, there are four main topics that are deeply discussed in this thesis:

- Sparse priors (e.g., Gaussian Mixture Models (GMMs)) are feasible to apply to model a disentangled style latent space. Once it works, there are at least two advantages of using GMMs to model the style latent space: (1) it is easy to keep the disentanglement among different style clusters of each image domain with parameterized GMMs; (2) it is possible to have a smooth style latent pace. When we interplate between two style codes (even sampled from different image domains), the model can output images with gradual and smooth changes.

- Such MMUIT models can be applied to some interesting applications. For example, we can use ambiguous textual commands to manipulate images (e.g., "change the hair color to blond" for a human face image). In this challenging task, the model should understand both vision and language inputs during the manipulation.

- Given an arbitrary MMUIT model with a style latent space, we propose three easy-to-use constraints for the latent space to be smooth. With a smooth latent space, the model can generate gradual and realistic interpolations between any two style latent codes, even between different domains.

- Usually, it is expensive to train an MMUIT model from scratch at high resolution. Interpreting the latent space of pretrained unconditional GANs shows a promising direction to alleviate this problem. With the state-of-the-art GANs (e.g., StyleGAN v1/v2 [56, 57]), we proposed a neural implicit style function to directly map the latent codes of GANs

to obtain high-quality image-to-image translation (e.g., 1024×1024 resolution).

1.2 Contributions and Outlines

In Chapter 2, we observe that recent studies have shown remarkable success for multiple domains but they suffer from two main limitations: they are either built from several two-domain mappings that are required to be learned independently, or they generate low-diversity results, a problem known as *mode collapse*. To overcome these limitations, we propose a method named GMM-UNIT, which is based on a content-attribute disentangled representation where the attribute space is fitted with a GMM. Each GMM component represents a domain, and this simple assumption has two prominent advantages. First, it can be easily extended to most MMUIT tasks. Second, the continuous domain encoding allows for interpolation between domains and for extrapolation to unseen domains and translations. Additionally, we show how GMM-UNIT can be constrained down to different methods in the literature, meaning that GMM-UNIT is a unifying framework for unsupervised image-to-image translation.

In Chapter 3, we extend the GMM-UNIT to a challenging scenario to interpret the multi-media input (i.e., text and images). Here the model not only has to learn the manipulation without the ground truth of the desired output but also has to deal with the inherent ambiguity of natural language. To achieve these goals, we propose a novel unsupervised approach, based on image-to-image translation, that alters the attributes of a given image through a command-like sentence such as "change the hair color to black". Contrarily to state-of-the-art approaches, our model does not require a human-annotated dataset nor a textual description of all the attributes of the desired image, but only those that have to be modified. Our proposed

model disentangles the image content from the visual attributes, and it learns to modify the latter using the textual description, before generating a new image from the content and the modified attribute representation. Considering text might be inherently ambiguous (blond hair may refer to different shadows of blond, e.g., golden, icy, sandy), our method generates multiple stochastic versions of the same translation.

MMUIT models are usually evaluated also using the quality of their semantic interpolation results. However, state-of-the-art models frequently show abrupt changes in the image appearance during interpolation, and especially perform poorly in interpolations across domains. In Chapter 4, we analyze the reasons behind this phenomenon and propose a new training protocol based on three specific losses which help a translation network to learn a smooth and disentangled latent style space in which: 1) Both intra- and inter-domain interpolations correspond to gradual changes in the generated images, and 2) The content of the source image is well preserved during the translation. The proposed method can be plugged into existing translation approaches, and our extensive experiments on different datasets show that it can significantly boost the quality of the generated images and the smoothness of the interpolations. Moreover, we propose a novel evaluation metric to properly measure the smoothness of latent style space of MMUIT models.

Recently, there has been increasing interests in image editing methods that employ pre-trained unconditional image generators (e.g., StyleGAN [56, 57]). However, it is a challenging task to apply these methods to translate images to multiple visual domains. Existing works often do not preserve the domain-invariant part of the image (e.g., the identity in human face translations), and they usually do not handle multiple domains or do not allow for multi-modal translations. In Chapter 5, we propose a neural implicit style function (ISF) to straightforwardly achieve multi-modal and

multi-domain image-to-image translation from pre-trained unconditional generators. The ISF manipulates the semantics of an input latent code to make the image generated from it lying in the desired visual domain. Our results in human face and animal manipulations show significantly improved results over the baselines. Our model enables cost-effective MMUIT at high resolution using pre-trained unconditional GANs.

Finally, we conclude in Chapter 6 and foresee potential new directions for future work.

Chapter 2

Gaussian Mixture Models

2.1 Introduction

Translating images from one domain into another is a challenging task that has significant influence on many real-world applications where data are expensive, or impossible to obtain and to annotate. Image-to-Image translation models have indeed been used to increase the resolution of images [29], fill missing parts [100], transfer styles [36], synthesize new images from labels [72], and help domain adaptation [16, 95]. In many of these scenarios, it is desirable to have a model mapping one image to multiple domains, while providing visual diversity (e.g., a day scene $\leftrightarrow$ night scene in different seasons). However, most of the existing models can either map an image to *multiple* stochastic results in a single domain, or model *multiple* domains in a deterministic fashion. In other words, the majority of the methods in the literature are either multi-domain or multi-modal.

Several reasons have hampered a stochastic translation of images to multiple domains. On the one hand, most of the Generative Adversarial Network (GAN) models assume a deterministic mapping [24, 101, 149], thus failing at modeling the correct distribution of the data [49]. On the other hand, approaches based on Variational Auto-Encoders (VAEs) usually as-

sume a shared and common zero-mean unit-variance normally distributed space [49, 150], limiting to two-domain translations.

We propose a novel UNsupervised Image-to-image Translation (UNIT) model that disentangles the visual content from the domain attributes. The attribute latent space is assumed to follow a Gaussian Mixture Model (GMM), thus naming the method: GMM-UNIT (see Section 2.1). This simple assumption allows three key properties: *mode-diversity* thanks to the stochastic nature of the probabilistic latent model, *multi-domain translation* since the domains are represented as clusters in the same attribute spaces and *few/zero-shot generation* since the continuity of the attribute representation allows interpolating between domains and extrapolating to unseen domains with very few or almost no observed data from these domains. The code and models will be made publicly available.

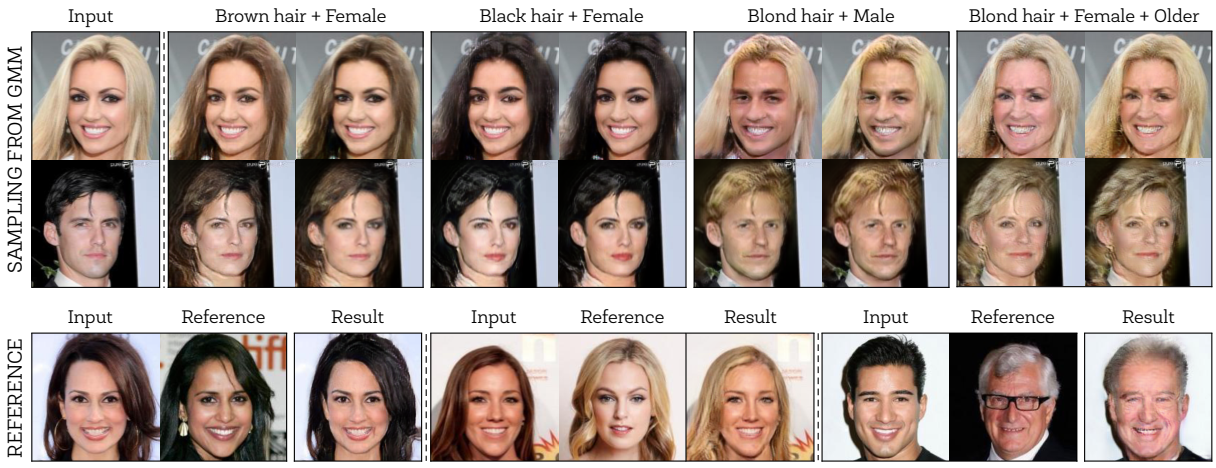


Figure 2.1: GMM-UNIT is a multi-domain and multi-modal image-to-image translation model where the target domain can either be sampled from a distribution, or extracted from a reference image. The first two rows show diverse images generated for each domain translation. The last row shows translations from a reference image.

2.2 Related work

Our work is best placed in the literature of image-to-image translation, where the challenge is to translate one image from a visual domain (e.g., summer) to another one (e.g., winter). This problem is inherently ill-posed, as there could be many mappings between two images. Thus, researchers tried to tackle the problem from different perspectives. The most impressive results on this task are undoubtedly related to GANs, which aim to synthesize new images as similar as possible to the real data through an adversarial approach between a Discriminator and a Generator. The former continuously learns to recognize real and fake images, while the latter tries to generate new images that are indistinguishable from the real data, and thus to fool the Discriminator. These networks can be effectively conditioned and thus generate new samples from a specific class [21] and a latent vector extracted from the images. For example, [50] and [128] trained a conditional GAN to encode the latent features that are shared between images of the same domain and thus decode the features to images of the target domain in a one-to-one mapping. However, this approach is limited to supervised settings, where pairs of corresponding images in different domains are available (e.g., a photos-sketch image pair). In many cases, it is too expensive and unrealistic to collect a large amount of paired data.

Unsupervised Domain Translation. Translating images from one domain to another without a paired supervision is particularly difficult, as the model has to learn how to represent both the content and the domain. Thus, constraints are needed to narrow down the space of feasible mappings between images. [119] proposed to minimize the feature-level distance between the generated and input images. [72] created a shared latent space between the domains, which encourages different images to be mapped in the same latent space. [149] proposed CycleGAN, which uses a cycle consistency loss that requires a generated image to be translated

back to the original domain. Similarly, [58] used a reconstruction loss applying the same approach to both the target and input domains. [93] later expanded the previous approach to the problem of translating multiple instances of objects in the same image. All these methods, however, are limited to a one-to-one domain mapping, thus requiring training multiple models for cross-domain translation. Recently, [24] proposed StarGAN, a unified framework to translate images in a multi-domain setting through a single GAN model. To do so, they used a conditional label and a domain classifier ensuring network consistency when translating between domains. However, StarGAN is limited to a deterministic mapping between domains.

Style transfer. A related problem is style transfer, which aims to transform the style of an image but not its content (e.g., from a photo to a Monet painting) to another image [28, 35, 48, 120]. Differently from domain translation, usually the style is extracted from a single reference image. We will show that our model could be applied to style transfer as well.

Multi-modal Domain Translation. Most existing image-to-image translation methods are deterministic, thus limiting the diversity of the translated outputs. However, even in a one-to-one domain translation such as when we want to translate people's hair from blond to black, there could be multiple hair color shades that are not modeled in a deterministic mapping. The straightforward solution would be injecting noise in the model, but it turned out to be worthless as GANs tend to ignore it [50, 88, 150]. To address this problem, [150] proposed BicycleGAN, which encourages the multi-modality in a paired setting through GANs and Variational Auto-Encoders (VAEs). [6] have instead augmented CycleGAN with two latent variables for the input and target domains and showed that it is possible to increase diversity by marginalizing over these latent spaces. [49] proposed MUNIT, which assumes that domains share a common content space but different style spaces. Then, they showed that by sampling from the style

space and using Adaptive Instance Normalization (AdaIN) [48], it is possible to have diverse and multimodal outputs. Similarly, [83] focused on the semantic consistency during the translation, and applied AdaIN to the feature-level space. Recently, [85] proposed a mode seeking loss to encourage GANs to better explore the modes and help the network avoiding the mode collapse.

Altogether, the models in the literature are either multi-modal or multi-domain. Thus, one has to choose between generating diverse results and training one single model for multiple domains. Here, we propose a unified model to overcome this limitation. Concurrent to our work, DRIT++ [66] also proposed a multi-modal and multi-domain model using a discrete domain encoding and assuming, however, a zero-mean unit-variance Gaussian shared space for multiple modes. We instead propose a content-attribute disentangled representation, where the attribute space fits a GMM distribution. A variational loss forces the latent representation to follow this GMM, where each component is associated to a domain. This is the key to provide for both multi-modal and multi-domain translation. In addition, GMM-UNIT is the first method proposing a continuous encoding of the domains, as opposed to the discrete encoding used in the literature. This is important because it allows for domain interpolation and extrapolation with very few or no data (few/zero-shot generation). The main properties of GMM-UNIT compared to the literature are shown in Table 2.1.

2.3 GMM-UNIT

GMM-UNIT is an image-to-image translation model that translates an image from one domain to multiple domains in a stochastic fashion, which means that it generates multiple outputs with visual diversity for the same translation.

Table 2.1: A comparison of the state of the art for image-to-image translation.

Method	Unpaired	Multi-Domain	Multi-Modal	Domain encoding
CycleGAN [149]	✓			None
BicycleGAN [150]			✓	None
MUNIT [49]	✓		✓	None
StarGAN [24]	✓	✓		Discrete
DRIT++ [66]	✓	✓	✓	Discrete
GMM-UNIT	✓	✓	✓	Continuous

Following recent seminal works [49, 65], our model assumes that each image can be decomposed in a domain-invariant content space and a domain-specific attribute space. Given Z attributes of a set of images, we model the attribute latent space through Gaussian Mixture Models (GMMs). Formally the probability density of the latent space $\mathbf{z}$ is defined as:

$$p(\mathbf{z}) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k) \tag{2.1}$$

where $\mathbf{z} \in \mathbb{R}^Z$ denotes a random attribute vector sample, $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$ denote respectively the mean vector and covariance matrix of the k-th GMM component, which is a Z-dimensional Gaussian ($\boldsymbol{\mu}^k \in \mathbb{R}^Z$ and $\boldsymbol{\Sigma}^k \in \mathbb{R}^{Z \times Z}$ is symmetric and positive definite). ϕ_k denotes the weight associated to the k-th component, where $\phi_k \geq 0$, $\sum_{k=1}^{K} \phi_k = 1$. As later explained, in this paper we set $K = |\text{domains in the data}|$, which means that each Gaussian component represents a domain. In other words, for an image $\mathbf{x}^n$ from domain $\boldsymbol{\mathcal{X}}^n$ (i.e., $\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^n}$), then its latent attribute is assumed to follow $\mathbf{z}^n \sim \mathcal{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)$, which is the n-th Gaussian component of the GMM that describes the domain $\boldsymbol{\mathcal{X}}^n$.

In the proposed representation, the domains are Gaussian components in a mixture. This simple yet effective model has one prominent advantage. Differently from previous works, where each domain is a category with a binary vector representation, we model the distribution of attribute space.

The continuous encoding of the domains we here introduce allows us to navigate in the attribute latent space, thus generating images corresponding to domains that have never (or very little) been observed and allowing to interpolate between two domains.

We note that the state of the art models can be traced back particular case of GMMs. Existing multi-domain models such as StarGAN [24] or GANimation [101] can be modeled with $K = |\text{domains in the data}|$ and $\forall k\ \boldsymbol{\Sigma}^k = 0$, thus only allowing the generation of a single result per domain translation. Then, when $K = 1$, $\boldsymbol{\mu} = \mathbf{0}$, and $\boldsymbol{\Sigma} = \mathbf{I}$ it is possible to model the state of the art approaches in multi-modal translation [49, 150], which share a unique latent space where every domain is overlapped, and it is thus necessary to train $K(K-1)$ models to achieve the multi-domain translation. Finally, we can obtain DRIT++ [66] by separating the attribute latent space into what they call an attribute space and a domain code. The former is a GMM with $K = 1$, $\boldsymbol{\mu} = \mathbf{0}$, and $\boldsymbol{\Sigma} = \mathbf{I}$, while the latter is another GMM with $K = |\text{domain in the data}|$ and $\forall k\ \boldsymbol{\Sigma}^k = 0$, which in [66] is a one-hot encoding of the domain. Thus, our GMM-UNIT is a generalization of the existing state of the art. In the next sections, we formalize our model and show that the use of GMMs for the latent space allows learning multi-modal and multi-domain mappings, and also few/zero-shot image generation.

2.3.1 The Generative-discriminative Approach

GMM-UNIT follows the generative-discriminative philosophy. The generator inputs a *content* latent code $\mathbf{c} \in \boldsymbol{\mathcal{C}} \subset \mathbb{R}^C$ and an *attribute* latent code $\mathbf{z} \in \boldsymbol{\mathcal{Z}} \subset \mathbb{R}^Z$, and outputs a generated image $G(\mathbf{c}, \mathbf{z})$. This image is then fed to a discriminator that must discern between "real" or "fake" images ($D_{\text{r/f}}$), and must also recognize the domain of the generated image (D_{dom}).

The attribute and content latent representations need to be learned,

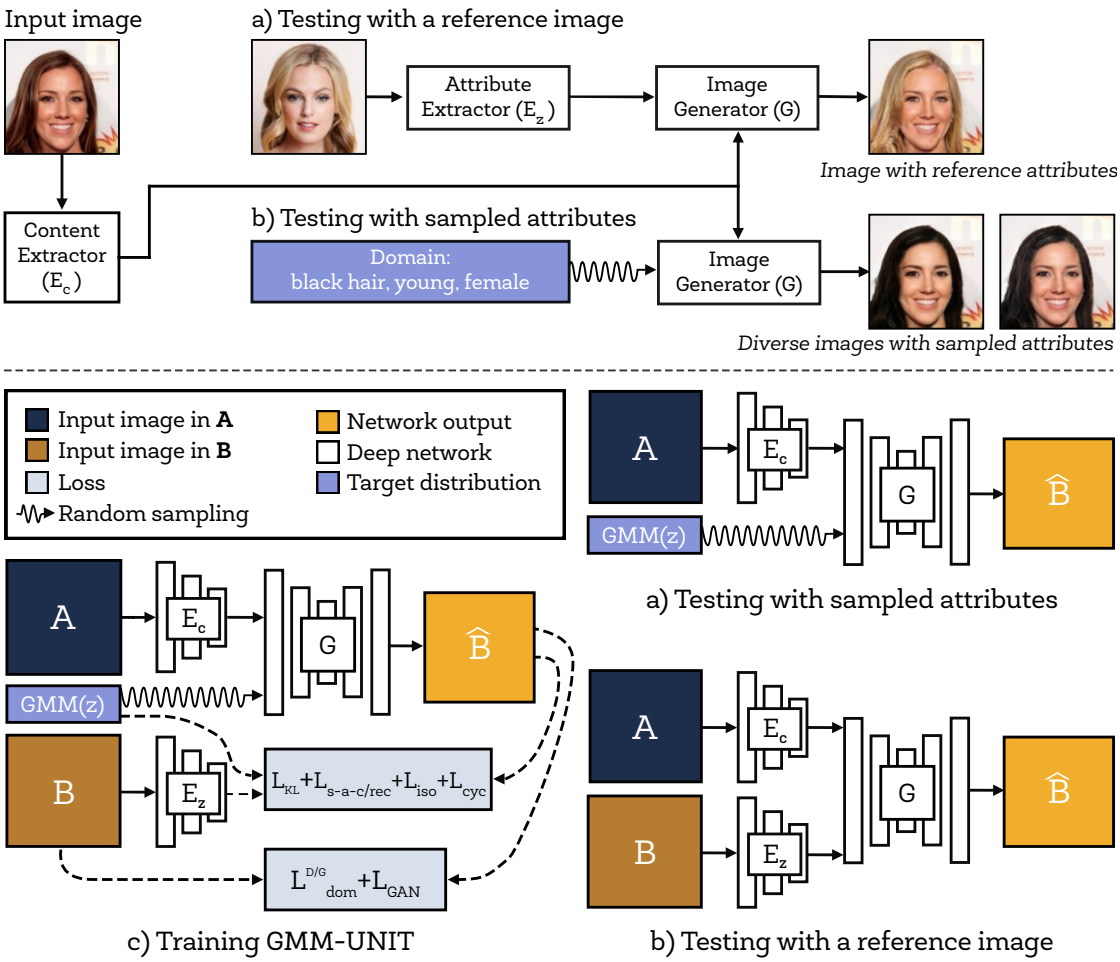


Figure 2.2: GMM-UNIT translates an input image from one domain to a target domain. The content is extracted from the input image, while the attribute can be either sampled (a) or extracted from a reference image (b). In detail: c) Training phase to translate an image from domain $\boldsymbol{A}$ to $\boldsymbol{B}$. The generator uses the content of the input image (extracted by E_c) and the attribute of the target image (extracted by E_z) to generate an image in $\boldsymbol{B}$. This image has the content of $\boldsymbol{A}$ (e.g., Scarlett Johansson) but the attributes of $\boldsymbol{B}$ (e.g., black hair). The attributes are modeled through a GMM. b) Testing phase where we use the content of an image in $\boldsymbol{A}$ and the target attributes sampled from the GMM distribution of the attributes of domain $\boldsymbol{B}$; c) Testing phase where we extract the content from an image in $\boldsymbol{A}$ and the attributes from an image belonging to the target domain $\boldsymbol{B}$. The style of this Figure is inspired from [150].

and they are modeled by two architectures, namely a *content extractor* E_c and an *attribute extractor* E_z. See 2.2 for a graphical representation of

GMM-UNIT for an $\boldsymbol{A} \leftrightarrow \boldsymbol{B}$ domain translation.

In addition to tackling the problem of multi-domain and multi-modal translation, we would like these two extractors, content and attribute, to be *disentangled* [49]. This would constrain the learning and hopefully yield better domain translation, since the content would be as independent as possible from the attributes. We expect the attributes features to be related to the considered attributes, while the content features are supposed to be related to the rest of the image. Formally, the following two properties must hold:

Sampled attribute translation

$$G(E_c(\mathbf{x}^m), \mathbf{z}^n) \sim p_{\boldsymbol{\mathcal{X}}^n} \forall\ \mathbf{z}^n \sim \mathcal{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n),\ \mathbf{x}^m \sim p_{\boldsymbol{\mathcal{X}}^m},\ n, m \in \{1, \ldots, K\}. \tag{2.2}$$

Extracted attribute translation

$$G(E_c(\mathbf{x}^m), E_z(\mathbf{x}^n)) \sim p_{\boldsymbol{\mathcal{X}}^n} \quad \forall\ \mathbf{x}^n \sim p_{\boldsymbol{\mathcal{X}}^n},\ \mathbf{x}^m \sim p_{\boldsymbol{\mathcal{X}}^m},\ n, m \in \{1, \ldots, K\}. \tag{2.3}$$

2.3.2 Training Detail

The encoders E_c and E_z, and the generator G need to be learned to satisfy three main properties. **Consistency**: An image and its generated/extracted codes have to be consistent even after a translation from a domain $\boldsymbol{A}$ to a domain $\boldsymbol{B}$. **Fit**: The distribution of the attribute latent space must follow a GMM. **Realism**: The generated images must be indistinguishable from the real images. In the following, we discuss different losses used to force the overall pipeline to satisfy these properties.

In the **consistency** term, we include image, attribute and content reconstruction, as well as cycle consistency. More formally, we use the following losses:

- *Self-reconstruction* of any input image from its extracted content and attribute vectors:

$$\mathcal{L}_{\text{s/rec}} = \sum_{n=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}} \left[\| G(E_c(\mathbf{x}), E_z(\mathbf{x})) - \mathbf{x} \|_1 \right] \tag{2.4}$$

- *Content reconstruction* from an image, translated into any domain:

$$\mathcal{L}_{\text{c/rec}} = \sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, \mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m)} \left[\| E_c(G(E_c(\mathbf{x}), \mathbf{z})) - E_c(\mathbf{x}) \|_1 \right] \tag{2.5}$$

- *Attribute reconstruction* from an image translated with any content:

$$\mathcal{L}_{\text{a/rec}} = \sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, \mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m)} \left[\| E_z(G(E_c(\mathbf{x}), \mathbf{z})) - \mathbf{z} \|_1 \right] \tag{2.6}$$

- *Cycle consistency* when translating an image back to the original domain:

$$\mathcal{L}_{\text{cyc}} = \sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, \mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m)} \left[\| G(E_c(G(E_c(\mathbf{x}), \mathbf{z})), E_z(\mathbf{x})) - \mathbf{x} \|_1 \right] \tag{2.7}$$

We note that all these losses are used in prior work [24, 49, 149, 150] to constraint the infinite number of mappings that exist between an image in one domain and an image into another one. The $\mathcal{L}_1$ loss is used as it generates sharper results than the $\mathcal{L}_2$ loss [50]. We also propose to complement the Attribute reconstruction with an isometry loss, to encourage the attribute extractor to be as similar as possible to the sampled attributes. Formally:

$$\begin{aligned} \mathcal{L}_{\text{iso}} = \sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, \mathbf{z}, \mathbf{z}'\sim\mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m)} \big[\big| \| E_z(G(E_c(\mathbf{x}), \mathbf{z})) - \\ E_z(G(E_c(\mathbf{x}), \mathbf{z}')) \|_1 - \| \mathbf{z} - \mathbf{z}' \|_1 \big| \big] \end{aligned} \tag{2.8}$$

In the **fit** term we encourage both the attribute latent variable to follow the Gaussian mixture distribution and the generated images to follow the domain's distribution. We set two loss functions:

- *Kullback-Leibler divergence* between the extracted latent code and the model. Since the KL divergence between two GMMs is not analytically

tractable, we resort on the fact that we know from which domain are we sampling and define:

$$\mathcal{L}_{\text{KL}} = \sum_{n=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}}[\mathcal{D}_{KL}(E_z(\mathbf{x})\|\mathcal{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n))] \tag{2.9}$$

where $\mathcal{D}_{KL}(p\|q) = -\int p(t)\log\frac{p(t)}{q(t)}dt$ is the Kullback-Leibler divergence.

- *Domain classification* of generated and original images. For any given input image $\mathbf{x}$, we would like the method to classify it as its original domain, and to be able to generate from its content an image in any domain. Therefore, we need two different losses, one directly applied to the original images, and a second one applied to the generated images:

$$\begin{aligned}\mathcal{L}^D_{\text{dom}} &= \sum_{n=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, d_{\boldsymbol{\mathcal{X}}^n}}[-\log D_{\text{dom}}(d_{\boldsymbol{\mathcal{X}}^n}|\mathbf{x})],\\ \mathcal{L}^G_{\text{dom}} &= \sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}, d_{\boldsymbol{\mathcal{X}}^m}, \mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}^m,\boldsymbol{\Sigma}^m)}[-\log D_{\text{dom}}(d_{\boldsymbol{\mathcal{X}}^m}|G(E_c(\mathbf{x}),\mathbf{z}))]\end{aligned} \tag{2.10}$$

where $d_{\boldsymbol{\mathcal{X}}^n}$ is the label of domain n. Importantly, while the generator is trained using the second loss only, the discriminator D_{dom} is trained using both.

The **realism** term tries to making the generated images indistinguishable from real images; we adopt the adversarial loss to optimize both the real/fake discriminator $D_{\text{r/f}}$ and the generator G:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = &\sum_{n,m=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^n}}[-\log D_{\text{r/f}}(\mathbf{x})] + \\ &\mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^m}, \mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}^n,\boldsymbol{\Sigma}^n)}[-\log(1 - D_{\text{r/f}}(G(E_c(\mathbf{x}),\mathbf{z})))]\end{aligned} \tag{2.11}$$

The full objective function of our network is:

$$\begin{aligned}\mathcal{L}_D =& \mathcal{L}_{\text{GAN}} + \mathcal{L}^D_{\text{dom}}\\ \mathcal{L}_G =& \mathcal{L}_{\text{GAN}} + \lambda_{\text{s/rec}}\mathcal{L}_{\text{s/rec}} + \mathcal{L}_{\text{c/rec}} + \mathcal{L}_{\text{a/rec}}\\ &+ \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}} + \mathcal{L}^G_{\text{dom}}\end{aligned} \tag{2.12}$$

where $\{\lambda_{\text{s/rec}}, \lambda_{\text{cyc}}, \lambda_{\text{KL}}, \lambda_{\text{iso}}\}$ are hyper-parameters of weights for corresponding loss terms. The values of most of these parameters come from the literature.

2.4 Experiments

We perform extensive quantitative and qualitative analysis in three real-world tasks, namely: edges-shoes, digits and faces. First, we test GMM-UNIT on a simple task such as a one-to-one domain translation. Then, we move to the problem of multi-domain translation where each domain is independent from each other. Finally, we test our model on multi-domain translation where each domain is built upon different combinations of lower level attributes. Specifically, for this task, we test GMM-UNIT in a dataset containing over 40 labels related to facial attributes such as hair color, gender, and age. Each domain is then composed by combinations of these attributes, which might be mutually exclusive (e.g., either male or female) or mutually inclusive (e.g., blond and black hair).

Additionally, we show how the learned GMM latent space can be used to interpolate attributes and generate images in previously unseen domains. Finally, we apply GMM-UNIT to the Style transfer task.

We compare our model to the state of the art of both multi-modal and multi-domain image translation problems. In the former, we select BicycleGAN [150], MUNIT [149] and MSGAN [85]. In the latter, we compare with StarGAN [24] and DRIT++ [66], which is the only multi-modal and multi-domain method in the literature. However, StarGAN is not multi-modal. Thus, similarly to what done previously [150], we modify StarGAN to be conditioned on Gaussian noise $(-0.2\mathcal{N}(0,1) + 0.1)$ in the input domain vector. We call this version of the model StarGAN*.

Our deep neural model architecture is built upon the state-of-the-art methods MUNIT [49], BicycleGAN [150] and StarGAN [24]. As shown in Table 2.2, we apply Instance Normalization (IN) [125] to the content encoder E_c, while we apply Adaptive Instance Normalization (AdaIN) [48] and Layer Normalization (LN) [9] for the decoder G. For the discriminator

network, we use Leaky ReLU [136] with a negative slope of 0.2. We note that we reduce the number of layers of the discriminator on the Digits dataset.

We use the Adam optimizer [59] with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and an initial learning rate of 0.0001. The learning rate is decreased by half every 2e5 iterations. In all experiments, we use a batch size of 1 for Edges2shoes and Faces and batch size of 32 for Digits. And we set the loss weights to $\lambda_{\text{s/rec}} = 10$, $\lambda_{\text{cyc}} = 10$, $\lambda_{\text{KL}} = 0.1$, and $\lambda_{\text{iso}} = 0.1$. We use the domain-invariant perceptual loss with weight 0.1 in all experiments. Random mirroring is applied during training.

While the GMM supports a full covariance matrix, simplify the problem as typically done in the literature. The simplified version satisfies the following properties:

- The mean vectors are placed on the vertices of Z-dimensional regular simplex, so that the mean vectors are equidistant.
- The covariance matrices are diagonal, with the same on all the components. In other words, each Gaussian component is *spherical*, formally: $\boldsymbol{\Sigma}_k = \boldsymbol{\sigma}_k \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

2.4.1 Metrics

We quantitatively evaluate our method through image quality and diversity of generated images. The former is evaluated through the Fréchet Inception Distance (FID) [46], while we evaluate the latter through the LPIPS [145].

FID We use FID to measure the distance between the generated and real distributions. Lower FID values indicate better quality of the generated images. We estimate the FID using 1000 input images and 10 samples per input v.s. randomly selected 10000 images from the target domain.

Table 2.2: GMM-UNIT network architecture. We use the following notations: Z: the dimension of attribute vector, n: the number of attributes, N: the number of output channels, K: kernel size, S: stride size, P: padding size, CONV: a convolutional layer, GAP: a global average pooling layer, UPCONV: a 2×bilinear upsampling layer followed by a convolutional layer, FC: fully connected layer. We set $C = 1$ in Edges2shoes and Digits, $C = 8$ in Faces. † refers to be optional.

Part	**Input → Output Shape**	**Layer Information**
	$(h, w, 3) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), IN, ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), IN, ReLU
E_c	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(h, w, 3) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), ReLU
E_z	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 256)$	CONV-(N256, K4x4, S2, P1), ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (1, 1, 256)$	GAP
	$(256,) \rightarrow (CZ,)$	FC-(NCZ)
	$(256,) \rightarrow (CZ,)$	FC-(NCZ)
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), AdaIN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), AdaIN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), AdaIN, ReLU
G	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), AdaIN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	UPCONV-(N128, K5x5, S1, P2), LN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 64)$	UPCONV-(N64, K5x5, S1, P2), LN, ReLU
	$(h, w, 64) \rightarrow (h, w, 3)$	CONV-(N3, K7x7, S1, P3), Tanh
	$\dagger(h, w, 64(+1)) \rightarrow (h, w, 1)$	CONV-(N3, K7x7, S1, P3), Sigmoid
	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
D	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{16}, \frac{w}{16}, 1)$	CONV-(N1, K1x1, S1, P0)
	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (1, 1, n)$	CONV-(Nn, K$\frac{h}{16}$x$\frac{w}{16}$, S1, P0)

LPIPS The LPIPS distance is defined as the $\mathcal{L}_2$ distance between the features extracted by a deep learning model of two images. This distance has been demonstrated to match well the human perceptual similarity [145]. Thus, following [49, 65, 150], we randomly select 100 input images and translate them to different domains. For each domain translation, we generate 10 images for each input image and evaluate the average LPIPS distance between the 10 generated images. Finally, we get the average of all distances. Higher LPIPS distance indicates better diversity among the generated images.

2.4.2 Edges ↔ Shoes: Two-domains Translation

We first evaluate our model on a simpler task than multi-domain translation: two-domain translation (e.g., edges to shoes). We use the dataset provided by [50, 149] containing images of shoes and their edge maps generated by the Holistically-nested Edge Detection (HED) [135]. We resize all images to 256×256 and train a single model for edges ↔ shoes without using paired information. Figure 2.3 displays examples of shoes generated from the same sketch by all the state of the art models. GMM-UNIT and MUNIT generate high-quality and diverse results that are almost indistinguishable from the ground truth and the results of BicycleGAN, which is a paired (supervised) method. Although, MSGAN and DRIT++ generate diverse images, they suffer from low quality results. The results of StarGAN* confirm the findings of previous studies that only adding noise does not increase diversity [50, 88, 150]. These results are confirmed in the quantitative evaluation displayed in Table 2.3. Our model generates images with high diversity and quality using half the parameters of the state of the art (MUNIT), which needs to be re-trained for each transformation. Particularly, the diversity is comparable to the paired model performance. These results show that this multi-modal and multi-domain model can be

efficiently applied also to simpler tasks than multi-domain problems without much loss in performance, while other multi-domain models suffer in this setting.

Figure 2.3: Qualitative evaluation on the Edges → Shoes.

Table 2.3: Quantitative evaluation on the Edges → Shoes dataset. The best performance for unpaired (unsupervised) models is in **green**. † refers to supervised method. MM and MD stands for Multi-Modal and Multi-Domain respectively.

Model	Unpaired	MM	MD	FID↑	LPIPS↑	Params↓
StarGAN* [24]	✓		✓	140.41	$.002 \pm .000$	$53.23M \times 1$
MUNIT [49]	✓	✓		**54.52**	**$.227 \pm .001$**	$23.52M \times 2$
MSGAN [85]	✓	✓		111.19	$.221 \pm .003$	$65.03M \times 2$
DRIT++ [66]	✓	✓	✓	123.87	$.233 \pm .002$	$54.06M \times 1$
GMM-UNIT	✓	✓	✓	58.46	$.200 \pm .002$	**$23.52M \times 1$**
BicycleGAN† [150]		✓		47.43	$.199 \pm .001$	$64.30M \times 2$

2.4.3 Digits: Single-attribute Multi-domain Translation

We then increase the complexity of the task by evaluating our model in a multi-domain translation setting, where each domain is composed by digits

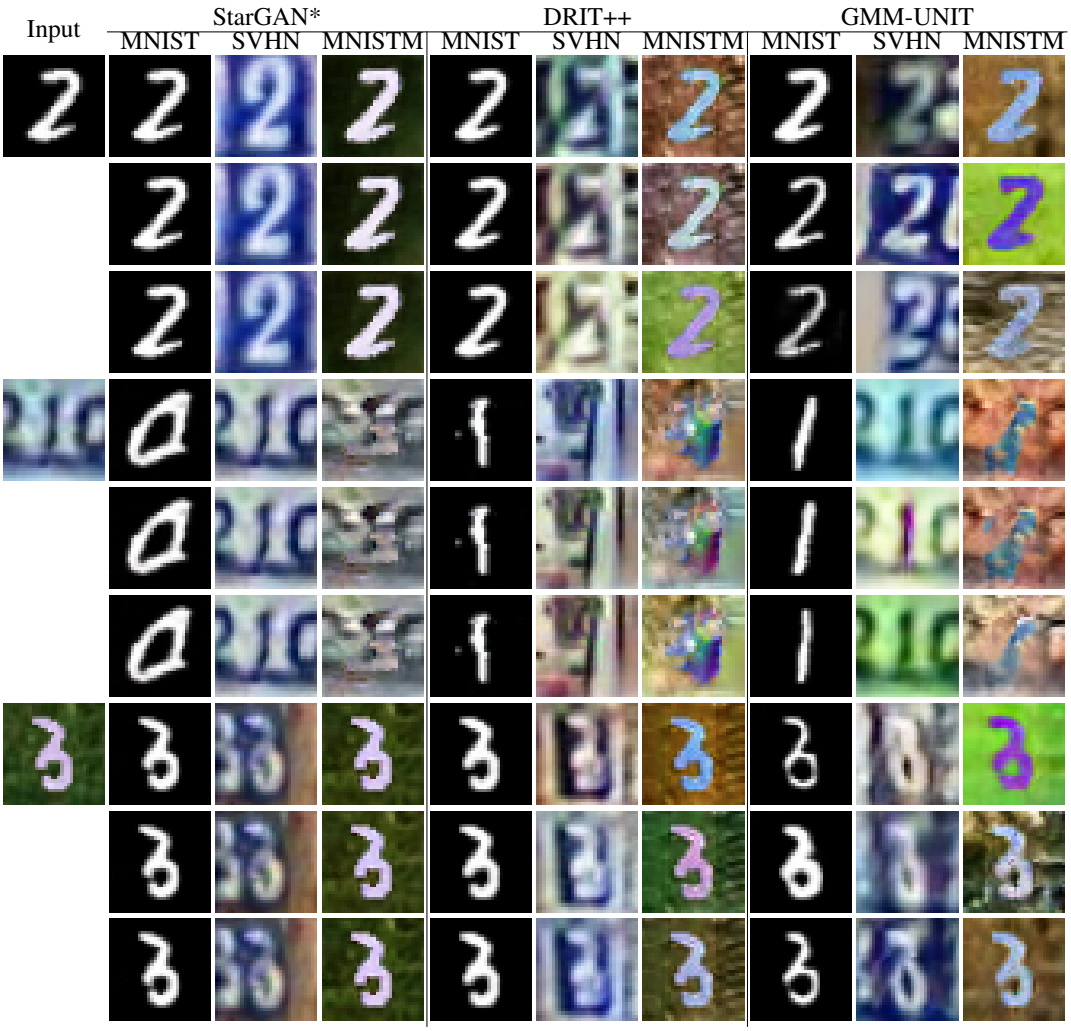


Figure 2.4: Visual comparisons of state of the art methods on the digits dataset. We note that StarGAN* [24] is a multi-domain (deterministic) model, while DRIT++ [66] and GMM-UNIT are multi-modal and multi-domain methods.

collected in different scenes. We use the Digits-Five dataset introduced in [137], from which we select three different domains, namely MNIST [62], MNIST-M [34], and Street View House Numbers (SVHN) [97]. During the training, given that all images are resized to 32×32, we reduce the depth of our model and compared models. Figure 2.4 shows the qualitative comparison with the state of the art, while we compare our model with the state-of-the-art on multi-domain translation, and we show in Table 2.4 the

quantitative results.

From these results we conclude that StarGAN* fails at generating diversity, while GMM-UNIT generates images with higher quality and diversity than all the state-of-the-art models. Additional experiments carried out implementing a StarGAN*-like GMM-UNIT (i.e., setting $\mathbf{\Sigma}^k = 0, \ \forall k$) indeed produced similar results. Specifically, the StarGAN*-like GMM-UNIT tends to generate for each input image one single (deterministic) output and thus the corresponding LPIPS scores are zero.

Table 2.4: Quantitative evaluation on the Digits and Faces datasets. The best performance is in **green**. For Faces, we also evaluate the diversity on the background.

Model	MM	MD	Digits		Faces		
			FID↓	LPIPS↑	FID↓	LPIPS↑	LPIPS_b↓
StarGAN* [24]		✓	69.11	$.006 \pm .000$	51.68	$.002 \pm .000$	$.035 \pm .010$
DRIT++ [66]	✓	✓	88.94	$.058 \pm .001$	55.64	$.017 \pm .001$	$.055 \pm .001$
GMM-UNIT	✓	✓	**60.43**	$\mathbf{.124 \pm .002}$	**46.21**	$\mathbf{.048 \pm .002}$	$\mathbf{.022 \pm .004}$

2.4.4 Faces: Multi-attribute Multi-domain Translation

We also evaluate GMM-UNIT in the complex setting of multi-domain translation in a dataset of facial attributes. We use the Celebfaces Attributes (CelebA) dataset [80], which contains 202,599 face images of celebrities where each face is annotated with 40 binary attributes. We apply central cropping to the initial 178×218 size images to 178×178, then resize the cropped images to 128×128. We randomly select 2,000 images for testing and use all remaining images for training. This dataset is composed of some attributes that are mutually exclusive (e.g., either male or female) and those that are mutually inclusive (e.g., people could have both blond and black hair). Thus, we model each attribute as a different GMM component. For this reason, we can generate new images for all the combinations

of attributes by sampling from the GMM. As aforementioned, this is not possible for state-of-the-art models such as StarGAN and DRIT++, as they use one-hot domain codes to represent the domains. To be consistent with the state of the art (StarGAN) we show five binary attributes: hair color (*black, blond, brown*), gender (*male/female*), and age (*young/old*). These five attributes allow GMM-UNIT to generate 32 domains.

We observed that image-to-image translation is sensitive to complex background information. In fact, models are inclined to manipulate the intensity and details of pixels that are not related to the desired attribute transformation. Hence, we add a convolutional layer at the end the of decoder G to learn a one-channel attention mask $\mathbf{M}$ in an unsupervised manner. Hence, the final prediction $\widehat{\mathbf{B}}$ is obtained through combining the input image $\mathbf{A}$ and its initial prediction $\widetilde{\mathbf{B}}$ through: $\widehat{\mathbf{B}} = \widetilde{\mathbf{B}} \cdot \mathbf{M} + \mathbf{A} \cdot (1 - \mathbf{M})$. We also apply the attention layer to Edges $\leftrightarrow$ Shoes and Digits, but find that it provides no noticeable improvements in the results.

Figure 2.5 shows some generated results of our model. We can see that GMM-UNIT learns to translate images to simple attributes such as blond hair, but also to translate images with combinations of them (e.g., blond hair and male). Moreover, we can see that the rows show different realizations of the model thus demonstrating the stochastic approach of GMM-UNIT. These results are corroborated by Table 2.4 that shows that our model is superior to StarGAN* and DRIT++ in both quality and diversity of generated images. Particularly, the use of an attention mechanism allows our model to achieve diversity only on the part of the image that is involved in the transformation (e.g., hair and face for gender and hair translation). To demonstrate this, we compute the LPIPS distance between the background of the input image and the generated images (LPIPS_b). Table 2.4 that our model is the best at preserving the original background information. In Figure 2.10 we show the difference between

the diversity we achieve and DRIT++ diversity. GMM-UNIT preserves the background while it changes the face and create diverse hair styles, while DRIT++ just changes the overall color intensity and affects parts of the image not related to the attributes, which is not desirable.

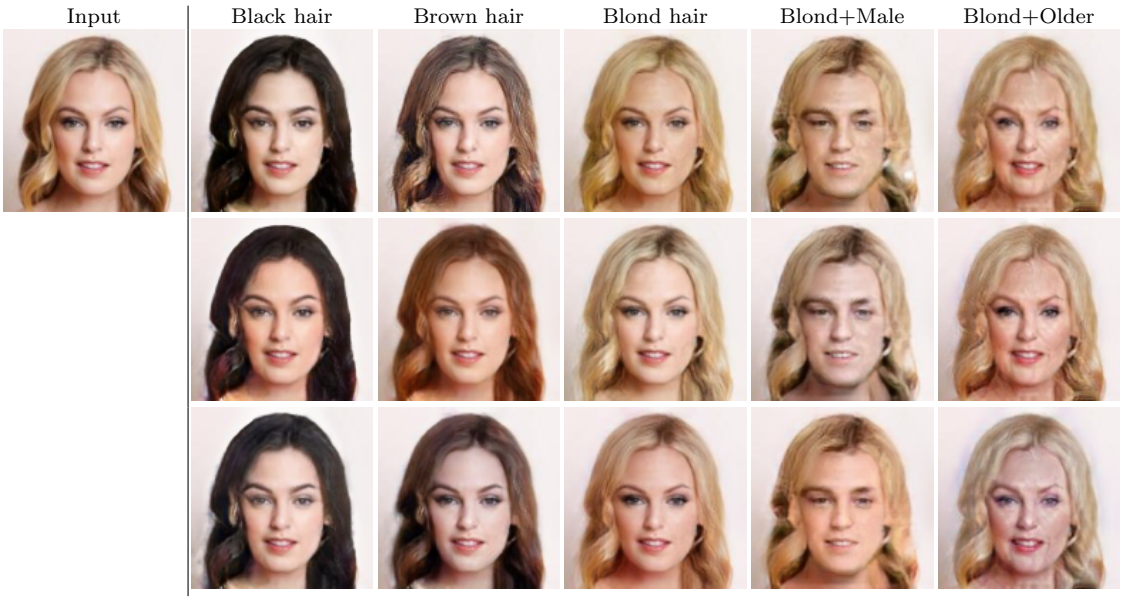


Figure 2.5: Facial expression synthesis results on the CelebA dataset with different attribute combinations. Each row represents a different output sampled from the model.

2.4.5 Style transfer

We evaluate our model on style transfer, which is a specific task where the style is usually extracted from a single reference image. Thus, we randomly select two input images and synthesize new images where, instead of sampling from the GMM distribution, we extract the style (through E_z) from some reference images. Figure 2.6 shows that the generated images are sharp and realistic, showing that our method can also be effectively applied to Style transfer.

Figure 2.6: Examples of GMM-UNIT applied on the Style transfer task. The style is here extracted from a single reference images provided by the user.

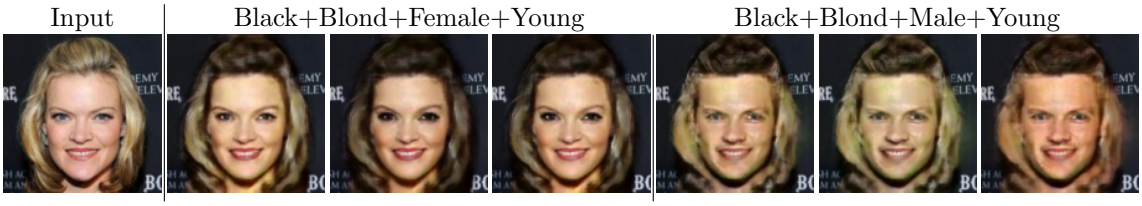


Figure 2.7: Generated images in previously unseen combinations of attributes.

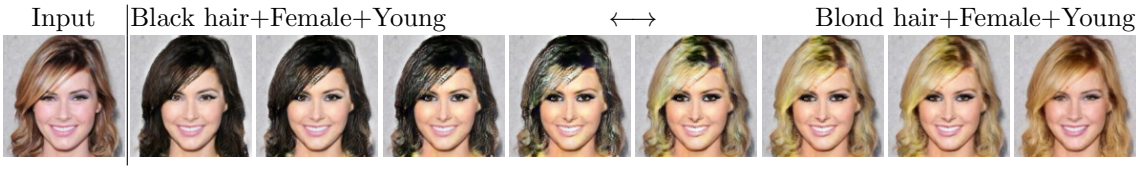


Figure 2.8: An example of domain interpolation given an input image.

2.4.6 Domain interpolation and extrapolation

In addition, we evaluate the ability of GMM-UNIT to synthesize new images with attributes that are extremely scarce or non present in the training dataset. To do so, we select three combinations of attributes consisting of less than two images in the CelebA dataset: *Black hair+Blond hair+Male+Young* and *Black hair+Blond hair+Female+Young*.

Figure 2.7 shows that learning the continuous and multi-modal latent distribution of attributes allow to effectively generate images as zero- or few-shot generation. At the best of our knowledge, we are the first ones being able to translate images in previously unseen domains at no additional cost. Recent literature on zero-pair translation learning indeed scale linearly with the number of domains [130]. This ability can be of vital importance in tasks where labels are extremely imbalanced.

Finally, we show that by learning the full latent distribution of the attributes we can do attribute interpolation both intra- and inter-domains. In contrast, state of the art methods such as [66] can only do intra-domain interpolations due to their discrete domain encoding. Other works such as Chen *et al.* [22] are focused on explicitly learning an interpolation and use a reference image to do the same task, while we can either interpolate between two reference images or between any two points in the attribute latent space (by sampling these points/vectors), even for multiple attributes. Figure 3.6 shows some generated images through a linear interpolation between two given attributes.

2.4.7 Visualization of the Attribute Latent Space

In Figure 2.9 we illustrate how three exemplar attributes (black, blond and brown hair) sampled from the GMM distribution are similarly projected in the latent space as those same attributes extracted by the encoder E_z. To project the attributes to a 2D space we use the t-SNE [84] algorithm with perplexity $= 30, \mathrm{lr} = 1.0$ and 300 iterations. We can observe from the figure that the attributes are well separated in the space, while the extracted attributes are very close to those sampled. In other words, for example the extracted black hair attribute is most similar to the sampled black hair attribute and most dissimilar to the extracted/sampled attribute of brown hair.

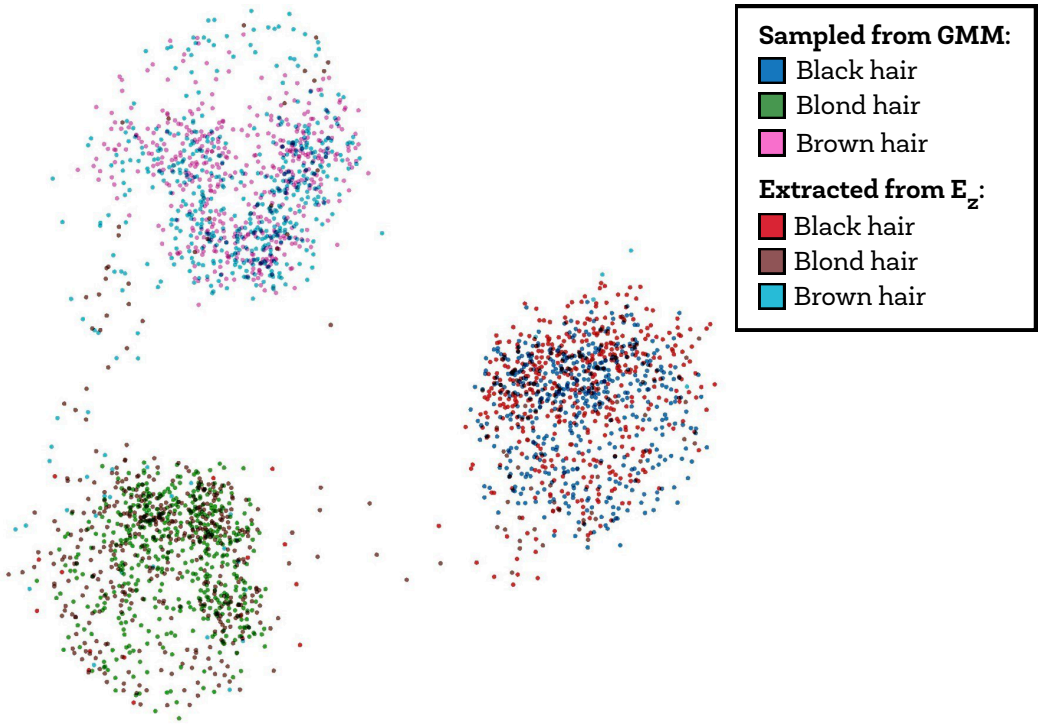


Figure 2.9: t-SNE projection of the attribute vectors in a 2D space. The points cloud refer to both extracted and sampled attributes, namely black, blond and brown hair, from the GMM-UNIT. The attributes are well separated, while for each attribute the extracted vectors are similar to the sampled ones.

2.4.8 Ablation study

Given that the importance of $\mathcal{L}_{s/rec}$ and $\mathcal{L}_{dom}$ was verified in previous works (i.e., CycleGAN and StarGAN), and that $\mathcal{L}_{c/rec}, \mathcal{L}_{KL}$ are necessary to the model convergence, we compare GMM-UNIT with three variants of the model that ablate $\mathcal{L}_{\text{cyc}}$, $\mathcal{L}_{\text{a/rec}}$ and $\mathcal{L}_{\text{iso}}$ in the Digits dataset. Table 2.5 shows the results of the ablation. As expected, $\mathcal{L}_{\text{cyc}}$ is needed to have higher image quality, and we observe that it increases the diversity because of noisy results. When $\mathcal{L}_{\text{a/rec}}$ is removed image quality decreases, but $\mathcal{L}_{\text{iso}}$ still helps to learn the attributes space. Finally, without $\mathcal{L}_{\text{iso}}$ we observe that both diversity and quality decrease, thus confirming the need of all these losses. For the first time from its introduction in [49], we also test for the *disentangled* assumption of visual content and attributes. Although we

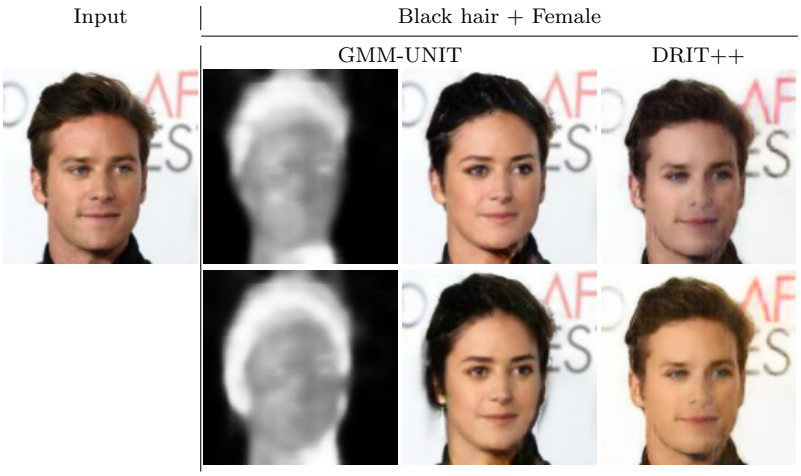


Figure 2.10: GMM-UNIT diversity is only on the subject thanks to the attention, while DRIT++ changes also the background.

cannot test the network removing the attribute extractor E_z, we remove the content extractor E_c and change the generator G to have $\mathbf{x}$ and $\mathbf{z}$ as input. We observe that the results are similar, although the diversity decreases substantially. This means that the disentanglement approach needs to be further studied in the multiple architectures and tasks that propose it [40, 49, 133] to understand its necessity and contribution.

Table 2.5: Ablation study performance on the Digits dataset.

Model	**FID↓**	**LPIPS↑**
GMM-UNIT (A)	**60.43**	.124 ± .002
(A) w/o $\mathcal{L}_{\text{cyc}}$	84.06	**.138 ± .003**
(A) w/o $\mathcal{L}_{\text{a/rec}}$	62.20	.121 ± .002
(A) w/o $\mathcal{L}_{\text{iso}}$	63.70	.115 ± .002
(A) w/o *disent.*	60.72	.097 ± .003

2.5 Conclusion

In this paper, we present a novel image-to-image translation model that maps images to multiple domains and provides a stochastic translation.

GMM-UNIT disentangles the content of an image from its attributes and represents the attribute space with a GMM, which allows us to have a continuous encoding of domains. This has two main advantages: first, it can easily be extended to most multi-domain and multi-modal image-to-image translation tasks. Second, GMM-UNIT allows for interpolation across-domains and the translation of images into previously unseen domains.

We conduct extensive experiments in three different tasks, namely two-domain translation, multi-domain translation and multi-attribute multi-domain translation. We show that GMM-UNIT achieves quality and diversity superior to state of the art, most of the times with fewer parameters. Future work includes the possibility to thoroughly learn the mean vectors of the GMM from the data and extending the experiments to a higher number of GMM components per domain.

In the next Chapter 3, we show that the proposed GMM-UNIT model can be extended for text-guided image-to-image translation.

Chapter 3

Text-guided UNIT

3.1 Introduction

Editing image attributes on portable devices is an uncomfortable operation for most people, but especially for impaired users. On smartphones, changing the hair color on a picture requires to manually select the pixels to be altered and color them properly.

Inspired by the emergence of voice assistants, which provide us a more convenient way to interact with machines through human language, we make pilot attempts to manipulate images through textual commands. Meeting this goal requires a deep integration of vision and language techniques, as previously achieved for other tasks, namely: image captioning [54, 140], text-to-image generation [67, 68, 96, 103, 143], text-based video editing [33] and drawing-based image editing [52, 100]. However, modifying parts of the image through natural language is still a challenging research problem.

Existing approaches, based on Generative Adversarial Networks (GANs) [7, 41], require either to describe all the characteristics of the desired image [103, 151] (e.g., "*young oval-faced lady with blond and black hair, smiling*"), or collecting a richly detailed dataset of supervised actions [31, 32] or captions [96, 103, 104, 138, 68]. For example, TAGAN [96] uses text input

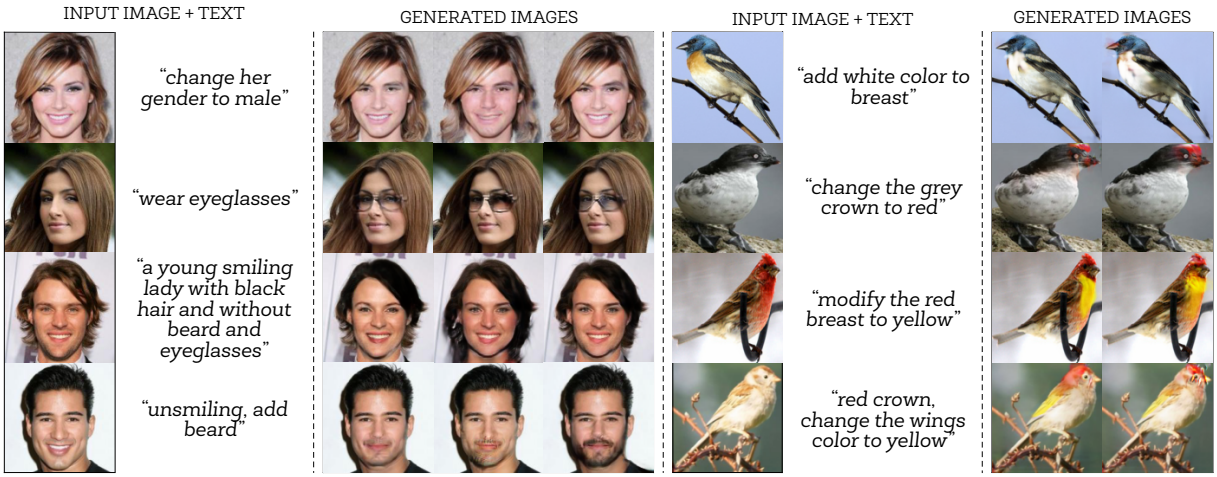


Figure 3.1: Our model allows to manipulate visual attributes through human-written text. To deal with the inherent ambiguity of textual commands, our model generates multiple versions of the same translation being as such multi-modal. Here, we see some examples of generated images from the CelebA [80] and CUB [126] datasets.

to change an image in order to have the attributes explained in the provided text description, but this method relies on detailed human-written captions. Similarly, El-Nouby *et al.* [31, 32] rely on annotated datasets to interact with the user and to generate semantically-consistent images from textual instructions. Two main issues have hindered the possibility to have an effective and unsupervised approach that follows human-written commands (e.g., "*make him smile*"). On the one hand, the literature on text-based image manipulation does not explicitly model image attributes, which would indicate if a specific part of the image is going to be changed or not. On the other hand, most GAN models assume a deterministic mapping, strongly limiting the capacity to handle the inherent ambiguity of human textual commands (e.g., "*blond hair*" might mean different shades of blond colors).

In this paper, we propose *Describe What to Change* (DWC), a novel method based on GANs to manipulate visual attributes from a human-written textual description of the required changes, thus named DWC-

Table 3.1: Example of text describing a translation from an image of a young smiling woman with blond hair and eyeglasses to an older smiling woman with black hair and eyeglasses. Differently from captioning text, users are not required to know and mention all the modeled attributes.

Input text	**Features**
Captioning text: `This woman is old and smiling, with black hair, eyeglasses and no beard.`	- Enumerates all the attributes; - No need to understand attributes from the input image; - Often trained with positive and negative annotation for each translation pair [96, 68].
Textual command: `Change the hair to be black, increase her age.`	- Focuses only on the differences; - Requires the extraction of attributes from the input image; - Trained without ground truth for each translation pair.

GAN. To explicitly model visual attributes and cope with the text ambiguity, our model translates images from one visual domain to other visual domains in a multi-modal (stochastic) fashion. Our contribution is threefold:

- We propose the use of textual commands instead of image captions. As shown in Table 3.1, this leads to three advantages: (1) the commands can be more flexible than image captions and we can compose them for progressive manipulation; (2) the commands can be automatically generated, and (3) users are not required to know and mention all the modeled visual attributes in an input image, only the desired changes. DWC-GAN thus does not rely on human-annotated captioning text compared with state-of-the-art approaches [96, 103, 104, 68].

- Our model disentangles the content features and the attribute representation from the input image making the manipulation more controllable and robust. The attribute space follows a Gaussian Mixture

Model (GMM), where the attributes of the source and the modified image follow the Gaussian distribution associated to their respective domains.

- To the best of our knowledge, we are the first to manipulate images through textual commands in a multi-modal fashion, allowing multiple versions of the same translation.

3.2 Related Work

Our work is best placed in the literature of image-to-image translation and text-to-image generation. The former aims to transform an input image belonging to a visual domain (e.g., young people) to another domain (e.g., elderly people), while the latter tries to generate images from a textual description provided as input. These two fields have witnessed many improvements in quality and realism, thanks to the advent of GANs [41, 92] and, in particular, conditional GANs (cGANs) [92], which are conditioned by extra information (e.g., the hair colour in face generation). GANs aim to synthesize images through a min-max game between a discriminator, trying to discriminate between real and fake data, and a generator, seeking to generate data that resemble the real ones.

Image-to-image translation: Conditional GANs were first employed for image-to-image translation in pix2pix [50] for learning a mapping from one domain to another by minimizing the L_1 loss between the generated and the target image. However, pix2pix requires a large amount of paired data (e.g., colored-greyscale images), which is often unrealistic to collect. Thus, the interest had shifted to unsupervised models that learn to translate images with unpaired data, for which the image content might vary (e.g., daylight-night images can have different people and cars). These models require additional constraints to better differentiate the domain-

dependent parts of the image from the domain-independent ones. For example, Liu *et al.* [80] assume that the two domains share a common latent space. CycleGAN [149] requires an image translated from a domain A to a domain B to be translated back to A, and applies a consistency loss, which was later employed by many other works [24, 49, 76, 93]. MUNIT [49] assumes that images share a domain-invariant content latent space and a style latent space, which is specific to a domain. This choice allows for generating multiple samples of the translation by drawing from the style distribution. Most of these approaches are, however, limited to one-to-one domain translation, which requires training a large number of models in the case of multiple domains. StarGAN [24] proposed a unified approach for multi-domain translation by employing a binary domain vector as input, which specifies the target domain, and a domain classification loss that helps the translation. Recently, GMM-UNIT [75] proposed a unified approach for multi-domain and multi-modal translation by modeling attributes through a Gaussian mixture, in which each Gaussian component represents a domain. In this paper, inspired by GMM-UNIT, we build a translation system where both the original and the manipulated attributes follow a GMM.

Text-to-image generation: cGANs can be conditioned with complex information, such as human-written descriptions, and then can generate accurate images, as shown by Reed *et al.* [103, 104]. For example, StackGAN [143] uses two stages of GANs in order to generate high-resolution images. The first stage generates a low-resolution image with the primitive shape and colour of the desired object, while the second one further refines the image and synthesize 256x256 images. The same authors improved the model with multiple generators through StackGAN++ [144]. Qiao *et al.* [102] focused instead on semantic consistency, enforcing the generated images to have the same semantics with the input text description.

In other words, the caption synthesized from the generated image should have the same meaning as the input text. Recently, Li *et al.* proposed StoryGAN [69], which generates a series of images that are contextually coherent with previously generated images and with the sequence of text descriptions provided by the user.

These approaches, however, require to accurately describe the picture to be generated, without allowing to start from an existing image and modify it through text.

Image manipulation: Modifying images through some user-defined conditions is a challenging task. Most of the previous approaches rely on conditional inpaiting [12, 52, 147], in which the network fills a user-selected area with pixel values coherent with the user preferences and the context. However, image editing does not necessarily require to select the exact pixels that have to be changed. Wang *et al.* [127] learned to change global image characteristics such as brightness and white balance from textual information. Zou *et al.* [151] proposed a network that colourizes sketches following the instructions provided by the input text specifications. El-Nouby *et al.* [31, 32] introduced a network that generates images from a dialogue with the user. A first image is generated from text, then objects are added according to the user preference provided as free text. Chen *et al.* [20] learned with a recurrent model and an attention module to colourize greyscale images from a description of the colours or the scene (e.g., *"the afternoon light flooded the little room from the window"*). Recently, Cheng *et al.* [23] proposed a network, tailored for dialogue systems, that gets as inputs an image and a series of textual descriptions that are apt to modify the image. The network is encouraged to synthesize images that are similar to the input and the previously generated images. However, examples show that the content of the image might vary substantially. Nam *et al.* [96] and Li *et al.* [68] introduced instead TAGAN and ManiGAN that modify fine-

grained attributes such as the colours of birds' wings through text. The key idea of their methods is reconstructing the images with positive and negative sentences, where the positive/negative refers to captioning text matching/mismatching the corresponding image. The recurrent network is trained to learn the words that refer to the attributes in the images and it allows to change words to manipulate multiple attributes at once.

Current image manipulation methods based on text often rely on human-annotated captioning text [96, 103, 104, 138, 68], which describes the scene in the image. Moreover, they do not explicitly model the concept of attributes (i.e., domains), sometimes failing at balancing the trade-off between keeping the old content and changing it [96], which is a well-known issue in image-to-image translation. Last but not least, existing solutions are deterministic, thus limiting the diversity of the translated outputs, even if the provided text might have different meanings. For example, "*wear a beard*" might mean to have a goatee, short or long beard.

Therefore, we build on state-of-the-art approaches for image-to-image translation and on the image manipulation literature and propose to explicitly model the visual attributes of an image and learn how to translate them through automatically generated commands. We are, up to the best of our knowledge, the first proposing a multi-domain and multi-modal method that translates images from one domain to another one by means of the user descriptions of the attributes to be changed.

3.3 Method

Our model is designed to modify the attributes of an image $\mathbf{x}$ through a human-written text $\mathbf{t}$, which describes the changes to be made in the image. In other words, we want to generate a new image $\hat{\mathbf{x}}$ that has the visual content of $\mathbf{x}$ and all its attributes but those attributes that should

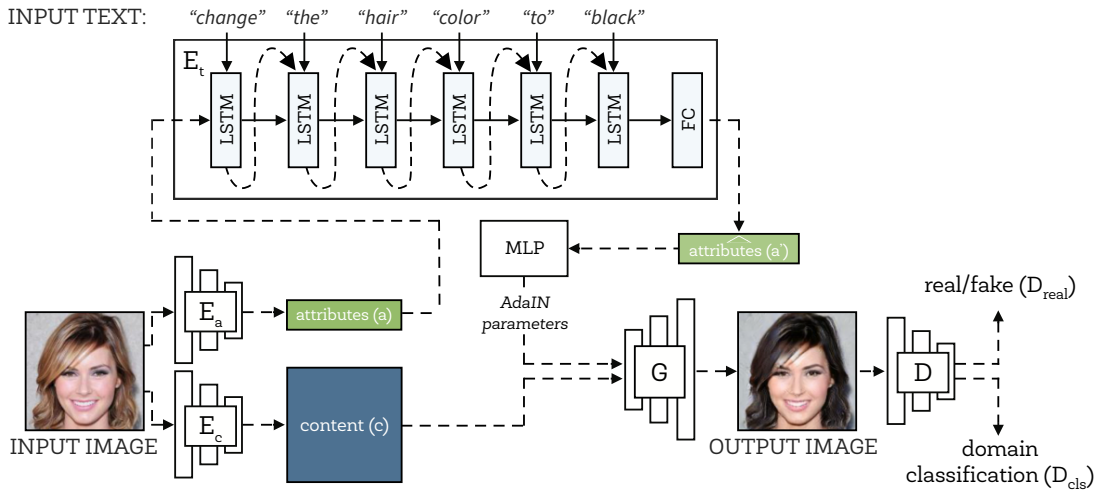


Figure 3.2: Architecture of our model. First, we disentangle the attributes and the content of the input image. Then, we modify the visual attributes of the original image using a text encoder. The generator uses a MLP to produce a set of AdaIN [48] parameters from the attribute representation. The content features is then processed by the parameterized generator to predict a realistic image with required attributes. Finally, the discriminator classifies whether the generated image is real or fake.

be altered accordingly to the text $\mathbf{t}$. Different from previous works, $\mathbf{t}$ is not a description of all the attributes of the generated image, but it is a direct command describing the modifications to be made (e.g., *change the hair color to black*). Thus, the proposed method should be able to model the attributes of the original image, the modifications of (some of) the attributes described in the text, and the attributes of the generated image. Moreover, it has to deal with the ambiguity of input text, which does not describe all the attributes of the target image, nor it is direct and clear as one-hot vector inputs.

The architecture, as shown in Figure 3.2, processes the input as follows. Given N attributes of a set of images, we first disentangle the visual content $\mathbf{c} \in \boldsymbol{\mathcal{C}} \subset \mathbb{R}^C$ and the attributes $\mathbf{a} \in \boldsymbol{\mathcal{A}} \subset \mathbb{R}^N$ of the input image $\mathbf{x}$ through the encoders E_c and E_a, respectively. Then, E_t, which is a Recurrent Neural Network (RNN) module, uses the input sentence $\mathbf{t}$ and the extracted

attributes $\mathbf{a}$ to infer the desired (target) attributes $\mathbf{a}' = E_t(\mathbf{t}, \mathbf{a})$. The inferred target attributes are then used along with the extracted content $\mathbf{c}$, by the generator G, to output the image $G(\mathbf{c}, \mathbf{a}')$. Finally, the discriminator D discerns between "real" or "fake" images (D_{real}) and recognizes the domain of the generated image (D_{cls}). Here, a domain is a combination of visual attributes. The model (content and attribute extractors, generator and discriminators) is learned in an end-to-end manner.

3.3.1 Assumptions

The basic assumption of DWC-GAN is that each image can be decomposed in a domain-invariant content space and a domain-specific attribute space [40, 49, 75]. We let the network model the high-dimension content features, while we represent the attributes of the image through Gaussian components in a mixture. This representation allows to model the combinations of attributes in a continuous space. We can then exploit this space to work with attribute combinations that have never (or little) been observed in the data. Formally and similarly to GMM-UNIT [75], we model the attributes with a K-component d-dimensional GMM: $p(\mathbf{a}) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\mathbf{a}; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)$ where ϕ_k, $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$ denote respectively the weight, mean vector and covariance matrix of the k-th GMM component ($\phi_k \geq 0$, $\sum_{k=1}^{K} \phi_k = 1$, $\boldsymbol{\mu}^k \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}^k \in \mathbb{R}^{d \times d}$ is symmetric and positive definite). Therefore, we model the attributes in a domain through a corresponding GMM component. Thus, for an image $\mathbf{x}$ from domain $\boldsymbol{\mathcal{X}}^k$ (i.e. $\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}$), its latent attribute $\mathbf{a}$ is assumed to follow the k-th Gaussian component: $\mathbf{a}^k \sim \mathcal{N}(\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)$. We aim to manipulate a source image $\mathbf{x}$ from $\boldsymbol{\mathcal{X}}^k$ into domain $\boldsymbol{\mathcal{X}}^\ell$ through a textual command $\mathbf{t}^\ell \sim p_{\boldsymbol{\mathcal{T}}^\ell}$, where $k, \ell \in \{1, \ldots, K\}$.

3.3.2 Multi-modal image generation

In order to have a multi-modal (non-deterministic) translation, we enforce both the sampled and extracted attributes to follow the image domain distribution, that is $\forall \ell, k$:

$$G(E_c(\mathbf{x}), \mathbf{a}) \sim p_{\boldsymbol{\mathcal{X}}^k}, \quad \forall\, \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k), \mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^\ell}.$$

We ensure these two properties with the following losses.

Reconstruction losses. These losses, introduced at first in [49, 149], force the output to be consistent with the original content and attributes. Specifically, the *self-reconstruction loss* ensures that the original image is recovered if its attribute and content is used in the generator. If $\mathbf{c}_\mathbf{x} = E_c(\mathbf{x})$ and $\mathbf{a}_\mathbf{x} = E_a(\mathbf{x})$, then for all k:

$$\mathcal{L}_{rec\_s} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}} \left[\| G(\mathbf{c}_\mathbf{x}, \mathbf{a}_\mathbf{x}) - \mathbf{x} \|_1 \right].$$

The *content and attribute reconstruction losses* [49] are used to constrain and learn the content and attribute extractors:

$$\mathcal{L}_{rec\_c} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)} \left[\| E_c(G(\mathbf{c}_\mathbf{x}, \mathbf{a})) - \mathbf{c}_\mathbf{x} \|_1 \right],$$

$$\mathcal{L}_{rec\_a} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}} \left[\| E_a(G(\mathbf{c}_\mathbf{x}, \mathbf{a}_\mathbf{x})) - \mathbf{a}_\mathbf{x} \|_1 \right].$$

The *cycle reconstruction* loss [149] enforces consistency when translating an image into a new domain and then back to the original one:

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)} \left[\| G(E_c(G(\mathbf{c}_\mathbf{x}, \mathbf{a})), \mathbf{a}_\mathbf{x}) - \mathbf{x} \|_1 \right],$$

where the use of the $\mathcal{L}_1$ loss inside the expectation is motivated by previous works [50] showing that $\mathcal{L}_1$ produces sharper results than $\mathcal{L}_2$. To encourage the generator to produce diverse images, we explicitly regularize G with the *diversity sensitive loss* [86, 25]:

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{a}_1, \mathbf{a}_2 \sim \mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)} \left[\| G(\mathbf{c}_\mathbf{x}, \mathbf{a}_1) - G(\mathbf{c}_\mathbf{x}, \mathbf{a}_2) \|_1 \right]$$

Domain losses. Similarly to StarGAN [24], for any given input image $\mathbf{x}$, we would like the method to classify it as its original domain, and to be able to generate an image in any domain from its content. Therefore, we need two different losses, one directly applied to the original images, and a second one applied to the generated images:

$$\mathcal{L}_{cls}^{D} = \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}, d_{\boldsymbol{\mathcal{X}}^k}}[-\log D_{cls}(d_{\boldsymbol{\mathcal{X}}^k}|\mathbf{x})] \quad \text{and}$$

$$\mathcal{L}_{cls}^{G} = \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}, d_{\boldsymbol{\mathcal{X}}^\ell}, \mathbf{a}\sim\mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)}[-\log D_{cls}(d_{\boldsymbol{\mathcal{X}}^\ell}|G(\mathbf{c}_\mathbf{x}, \mathbf{a}))],$$

where $d_{\boldsymbol{\mathcal{X}}^\ell}$ is the label of ℓ-th domain. Importantly, the discriminator D is trained using the first loss, while the generator G is trained using the second loss.

Adversarial losses. These terms enforce the generated images to be indistinguishable from the real images by following the formulation of LSGAN [87]:

$$\begin{aligned}\mathcal{L}_{GAN}^{D} = &\mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}}[D_{\text{real}}(\mathbf{x})^2] + \\ &\mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{a}\sim\mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)}[(1 - D_{\text{real}}(G(\mathbf{c}_\mathbf{x}, \mathbf{a})))^2]\end{aligned} \tag{3.1}$$

$$\mathcal{L}_{GAN}^{G} = \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{a}\sim\mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)}[D_{\text{real}}(G(\mathbf{c}_\mathbf{x}, \mathbf{a}))^2] \tag{3.2}$$

3.3.3 Attribute manipulation losses

As mentioned above, the underlying statistical assumption for the attributes is to follow a GMM – one component per domain. Very importantly, both the attribute representations extracted from image by E_a and the attribute representations obtained from E_t should follow the assumption and correspond to the correct component. We do that by imposing that the Kullback–Leibler (KL) divergence of the extracted and manipulated attributes correspond to the Gaussian component of the original and

targetted domain respectively. Recalling that $\mathbf{a}_\mathbf{x} = E_a(\mathbf{x})$, we write:

$$\begin{aligned}\mathcal{L}_{KL} = \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{t}\sim p_{\boldsymbol{\mathcal{T}}^\ell}} \big[& \mathcal{D}_{KL}(\mathbf{a}_\mathbf{x} \| \mathcal{N}(\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)) + \\ & \mathcal{D}_{KL}(E_t(\mathbf{t}, \mathbf{a}_\mathbf{x}) \| \mathcal{N}(\boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell)) \big]\end{aligned} \tag{3.3}$$

where $\mathcal{D}_{KL}(p\|q) = -\int p(t) \log \frac{p(t)}{q(t)} dt$ is the KL.

Intuitively, the second KL divergence enforces the images generated from the manipulated attributes to follow the distribution of the target domain:

$$G(\mathbf{c}_\mathbf{x}, E_t(\mathbf{t}, \mathbf{a}_\mathbf{x})) \sim p_{\boldsymbol{\mathcal{X}}^\ell}, \quad \forall\, \mathbf{x} \sim p_{\boldsymbol{\mathcal{X}}^k}, \mathbf{t} \sim p_{\boldsymbol{\mathcal{T}}^\ell} \tag{3.4}$$

Finally, the full objective function of our network is:

$$\mathcal{L}_D = \mathcal{L}^D_{GAN} + \lambda_{cls}\mathcal{L}^D_{cls} \tag{3.5}$$

$$\begin{aligned}\mathcal{L}_G = & \mathcal{L}^G_{GAN} + \lambda_{rec\_s}\mathcal{L}_{rec\_s} + \mathcal{L}_{rec\_c} + \mathcal{L}_{rec\_a} + \\ & \lambda_{cyc}\mathcal{L}_{cyc} - \mathcal{L}_{ds} + \lambda_{KL}\mathcal{L}_{KL} + \mathcal{L}^G_{cls}\end{aligned} \tag{3.6}$$

where $\{\lambda_{rec\_s}, \lambda_{cyc}, \lambda_{KL}\}$ are hyper-parameters of weights for corresponding loss terms. The value of most of these parameters come from the literature. Note that these losses are required to constrain down the difficult problem of image to image translation, disentangle content and style, and achieve stochastic results by sampling from the GMM distribution of attributes. Without these losses, the problem would be much less constrained and difficult.

3.3.4 Domain Sampling

We propose a strategy to obtain diverse manipulated results during testing by following the assumption introduced in Section 3.3.1. After extracting the target attributes $\mathbf{a}'$ from the original image $\mathbf{x}$ and input text $\mathbf{t}$, we assign it to a closest component $(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ of the GMM by following: $k^* = \arg\max_{k\in\{1,\dots,K\}}\{\Phi(\mathbf{a}'; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)\}$, where Φ is the Gaussian probability density function. Then, we can randomly sample several $\mathbf{a}^*$ from the

component $\mathcal{N}(\boldsymbol{\mu}^{k^*}, \boldsymbol{\Sigma}^{k^*})$. With such different sampled $\mathbf{a}^*$'s and the trained non-linear generator G, our model achieves multi-modality in any domain, which is different from state-of-the-art approaches [96, 144] that use random noise sampled from a standard normal distribution $\mathcal{N}(0, 1)$ to generate multiple diverse results independently of the domain.

Sampling all the target attributes might hurt translation performance. Thus, we propose an additional constraint that the ideal attribute representation is a mixture of domain sampling $\mathbf{a}^*$ and extracted representation $E_a(\mathbf{x})$, where $\mathbf{a}^*$ copies the attribute representation from $E_a(\mathbf{x})$ when a target attribute is the same as the attribute in the input image.

3.3.5 Unsupervised Attention

Attention mechanisms have proven successful in preserving the visual information that should not be modified in an image-to-image translation [89, 101, 75, 76]. For example, GMM-UNIT [75] claims that models are inclined to manipulate the intensity and details of pixels that are not related to the desired attribute transformation. Therefore, we follow the state-of-the-art approaches and add a convolutional layer followed by a sigmoid layer at the end of G to learn a single channel attention mask $\mathbf{M}$ in an unsupervised manner. Then, the final prediction $\tilde{\mathbf{x}}$ is obtained from a convex combination of the input and the initial translation: $\tilde{\mathbf{x}} = \hat{\mathbf{x}} \cdot \mathbf{M} + \mathbf{x} \cdot (1 - \mathbf{M})$.

3.4 Experiments

3.4.1 Datasets

CelebA. The CelebFaces Attributes (CelebA) dataset [80] contains 202,599 face images of celebrities where each face is annotated with 40 binary attributes. This dataset is composed of some attributes that are mutually

exclusive (e.g., either male or female) and those that are mutually inclusive (e.g., people could have both blond and black hair). To be consistent with previous papers [24, 75], we select a subset of attributes, namely *black hair*, *blond hair*, *brown hair*, *male/female*, *young/old*, *smile*, *eyeglasses*, *beard*. As we model these attributes as different GMM components, our model allows the generation of images with up to 2^8 different combinations of the attributes. Note that in the dataset only 100 combinations exist. As preprocessing, we center crop the initial 178×218 images to 178×178 and resize them to 128×128. We randomly select 2,000 images for testing and use all remaining images for training.

CUB Birds. The CUB dataset [126] contains 11,788 images of 200 bird species, and provides hundreds of annotated attributes for each bird (e.g., color, shape). We select some color attributes {*grey*, *brown*, *red*, *yellow*, *white*, *buff*} of three body parts {*crown*, *wing*, *breast*} of the birds, yielding a total of $K = 18$ attributes used to represent around one thousand domains appearing in the dataset. Differently from the CelebA dataset, the selected colors could appear in all selected body parts. Hence, the colors and body parts follow a hierarchical structure, requiring the model to modify the color of a specific body part. We crop all images by using the annotated bounding boxes of the birds and resize images to 128×128.

3.4.2 Automatic Text Description

The text encoder E_t has to be trained with a dataset describing the changes to be made in the visual attributes of the input image. Therefore, we here propose a protocol to generate the corpus automatically for the datasets provided multiple annotated attributes or labels. Given an image $\mathbf{x}$, we can collect a set of attributes $\{p_1, \ldots, p_N\}$ and obtain the corresponding real-valued vector $\mathbf{p} \in \{0, 1\}^N$ for generating the description text. We denote $\mathbf{p}'$ as the randomly assigned real-valued vector of a target image. After

observing the human-used commands of manipulating images, we propose an automatic description text generation method through three different strategies:

Step-by-step actions. We can compare the differences between $\mathbf{p}$ and $\mathbf{p}'$ element-by-element. For example, if the hair colors in the source image and target image are "*brown*" and "*black*", respectively. We can describe the action like "*change the brown hair to black*", "*change the hair color to black*", "*black hair*". Similarly, we can describe all actions that change the image one by one. For the same attributes in $\mathbf{p}$ and $\mathbf{p}'$, we can provide empty description or the no-change commands (e.g., "*do not change the hair color*", "*keep his hair color unchanged*"). Then, the sequence of step-by-step actions are disrupted randomly. A full example for this strategy is "*make the face older, translate the face to be smiling, remove the eyeglasses, do nothing on the gender, change the hair color to black, wear a beard*".

Overall description. In practice, it's feasible to describe the attributes in the target images to manipulate images, which is similar to the text-to-image task [96, 103, 104, 68]. Here, we completely describe the appeared attributes in $\mathbf{p}'$. For example, "*a smiling young man with black hair, wearing a beard and without eyeglasses*". In addition, the untouched attributes are not regarded as existing attributes in the target image by default.

Mixed description. To increase the generalization, we mix the two previous strategies by randomly mentioning some attributes in $\mathbf{p}$ and specifying the exact change of attributes in $\mathbf{p}'$ For example, *translate the young miss to be an old smiling man with black hair and wearing beards and eyeglasses.*". In this case, the attributes of both the input image and input text are mentioned, which is more complex for the language model.

During training, we propose an automatic text protocol to collect the description text, which takes use of the attribute annotations only. Hence,

we launch our experiments based on two multi-label recognition or classification [80, 126] datasets. In CelebA [80] dataset, we select 8 attributes from annotated attributes { "*Black_Hair*", "*Blond_Hair*", "*Brown_Hair*", "*Male*", "*Smiling*", "*Young*", "*Eyeglasses*", "*No_Beard*"}, which including both the style (e.g., hair color) and content (e.g., gender, age, beard and eyeglasses). In CUB [126] dataset, we select three body parts of the birds {"*wing*", "*crown*", "*breast*"} and each part can be consist of a set of colors {"*grey*", "*brown*", "*red*", "*yellow*", "*white*", "*buff*"}.

To automatically collect manipulation text, we provide some common templates and phrases to collect the text, such as "*change ... to ...*", "*with ... hair*", "*make the face/bird ...*", "*a man/woman/bird with/has ...*". We randomly permute the description sequence of the attributes and replace some words with synonymy (e.g., "*change*", "*translate*", "*modify*" and "*make*") to improve the generalization. Similarly, we randomly select the synonymy named entities during generate text (e.g., "*man*", "*male*", "*boy*", "*gentleman*" and "*sir*" for the attribute *Male*). We also test the sensitivity of the strategy, as the choice of the strategy might influence the results of the model. Table 3.2 shows that our model is robust to the choice of the textual strategy.

Table 3.2: Quantitative comparison on the different automatic text strategies on the CelebA dataset.

Strategy Name	**Metrics**		
	IS↑	FID↓	F1↑
Step-by-step	3.070	32.25	94.14
Overall	3.081	32.07	94.78
Mixed	3.078	32.16	94.80
Random	3.069	32.14	94.80

3.4.3 Metrics and Baseline Models

We quantitatively evaluate our model through the image quality, diversity and the accuracy of generated images. We evaluate the image quality through the Fréchet Inception Distance (FID) [46] and the Inception Score (IS) [109], the diversity is measured by the Learned Perceptual Image Patch Similarity (LPIPS) [145], the mean accuracy (Mean Acc) of multi-label classification, while we also measure the accuracy through a user study.

FID and IS. We randomly select 10000 source images and 10000 target images. The models transfer all source images with the same attributes of target images through textual commands, as shown in Section 3.4.2. Hence, the FID is estimated using 10000 generated images vs the selected target 10000 images. The IS is evaluated using Inception-v3 [118] and 10000 generated images for FID.

LPIPS. The LPIPS distance is defined as the L_2 distance between the features extracted by a deep learning model of two images. This distance has been demonstrated to match well the human perceptual similarity [145]. Thus, following [49, 150], we randomly select 100 input images and translate them to different domains. For each domain translation, we generate 10 images for each input image and evaluate the average LPIPS distance between the 10 generated images. Finally, we get the average of all distances. Higher LPIPS distance indicates better diversity among the generated images. We show in the results both the mean and standard deviation.

Accuracy and Realism. Through these two metrics we evaluate how humans perceive generated images. *Accuracy* measures whether the attributes of the generated images are coherent with the target attributes (0: incorrect; 1: correct). *Realism* quantifies how realistic a generated image is perceived (0: bad, unrecognizable; 1: neutral, recognizable but with obvious defects; 2: good, recognizable but with slight defects; 3: perfect,

recognizable and without any defect). We test Accuracy and Realism by randomly sampling 50 images with the same translation condition and collect more than 20 surveys from different people with various background.

To verify the priorities of our proposed method, we mainly compare with several state-of-the-art methods in both image-to-image translation and text-to-image translation fields:

StackGAN++ [144] is a text-to-image model that takes text as input and encodes the text into embedding for the decoding network. Here, we train/test this model by using the embedding of the manipulation text as input.

TAGAN [96] and **ManiGAN** [68] are two image-conditioned text-to-image models that take the original image and detailed description text of the target image as input. We follow the original method and train/test the model on our automatic description text.

StarGAN [24] is a unified multi-domain translation model that takes the original image and target attribute vector as input. Here, we add the same RNN module (i.e., the text understanding module) used in our framework to StarGAN to extract attributes from the input text. After that, we feed the embedding of the input text and original image as input. We call this modified version as StarGAN*.

For all the baseline models, we use the same pre-trained word embeddings in 300 dimensions for the input description text. The embeddings are trained by the skip-gram model described in [14] on Wikipedia using fastText[1].

Regarding the baseline models, we modify the baseline models based on their corresponding released codes, including StackGAN++[2], StarGAN[3],

[1] https://github.com/facebookresearch/fastText
[2] https://github.com/hanzhanggit/StackGAN-v2
[3] https://github.com/yunjey/stargan

TAGAN[4] and ManiGAN[5]. To train StarGAN on this task, we add a module E_t same as ours and learn to predict target attributes guided by binary cross-entropy loss. For fairness, all models take use the same corpus and images during the training. To test the training speed in Table 1 of the main paper we set the batch size 1 and use a single GTX TITAN Xp with 12GB Memory for all models.

Table 3.3: Quantitative evaluation on the CelebA dataset. There is no captioning text on CelebA.

Method	**IS↑**	**FID↓**	**LPIPS↑**	**Accuracy↑**	**Realism↑**	**Params ↓**
StackGAN++ [144]	1.444	285.48	**.292**±.053	.000	.00	50.30M
TAGAN [96]	1.178	421.84	.024±.012	.000	.00	44.20M
StarGAN* [24]	2.541	50.66	.000±.000	.256	1.17	54.10M
DWC-GAN (proposed)	**3.069**	**32.14**	.152±.003	**.885**	**2.25**	**32.75**M

Figure 3.3: Qualitative evaluation for different textual input on CelebA dataset. Our model generates high-quality images that are consistent with the textual commands.

[4] https://github.com/woozzu/tagan
[5] https://github.com/mrlibw/ManiGAN

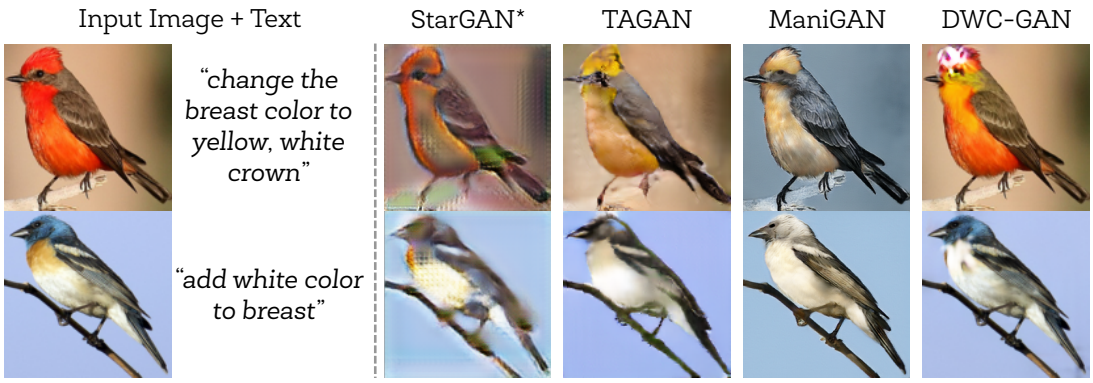


Figure 3.4: Qualitative comparisons for different textual input on CUB datasets. For reference, we show also the results of StarGAN* [24], TAGAN [96] and ManiGAN [68].

3.4.4 Results

We begin by quantitatively comparing our model with state of the art. Table 3.3 shows that our model generates better quality images than all competing methods, using around half the parameters and achieving comparable training speed. Specifically, DWC-GAN outperforms our most similar competitor, TAGAN, attaining higher IS (3.069 vs 1.178) and lower FID (32.14 vs 421.84) with less required parameters (32.75M vs 44.20M). Surprisingly, the diversity of generated images is lower than StackGAN++. However, the qualitative results show that StackGAN++ generates diverse but very noisy images, in which the attributes are not easily understandable. Contrarily, Figure 3.3 shows that DWC-GAN generates high-quality images that are also consistent with the manipulation described by the text. Our approach outperforms state of the art in all the metrics in the CUB dataset as well, as shown in both Figure 3.3 and Table 3.4. Additional results can be seen in Figure 3.1.

Differently from state of the art, DWC-GAN explicitly models the attributes of images in a smooth and continuous space through a GMM. This allows to translate images in multiple domains, but also to have multimodal results. As Figure 3.1 shows, our model generates different realiza-

Table 3.4: Quantitative evaluation on the CUB dataset.

Method	IS↑	FID↓	LPIPS↑	Params ↓
StackGAN++ [144]	1.029	278.60	.028±.009	50.30M
TAGAN [96]	4.451	50.51	.060±.024	44.20M
ManiGAN [68]	4.136	11.74	.001±.000	163.34M
StarGAN* [24]	4.343	109.89	.000±.000	54.10M
DWC-GAN (proposed)	**4.599**	**2.96**	**.081±.001**	**33.53M**

tions of the same manipulation. For example, in *"add beards"* the model synthesizes various versions of the subjects' beards, as the command is open to more than one interpretation. State-of-the-art models such as TAGAN and StarGAN* do instead generate results with no, or low, diversity.

Our model uses the source attributes and the human-written sentence to generate the desired attributes. The results of StackGAN++ show that it is not feasible to only encode textual commands to generate meaningful target images, while StarGAN* shows that taking into account only the text attributes as an additional condition is not enough. In particular, StarGAN* exhibits a visible mode collapse in this setting. These results show that applying and modifying the existing baselines is not feasible to solve our task.

We trained our model with automatically generated text, thus without relying on human-annotated datasets as state of the art does [96, 103, 104]. We observe that while TAGAN does not perform well with our corpus, we have consistent results with different strategies of automatic text generation, and even with a text strategy that is similar to their dataset (the Overall strategy). This result proves that annotating the dataset with automatically generated textual commands is a feasible strategy to train a model that manipulates images through text.

Human evaluation We evaluated StarGAN* and DWC-GAN by a user study where more than 20 people where asked to judge the Accuracy and

Realism of generated results. On average, 82.0% of the images generated by DWC-GAN were judged as correct, while StarGAN* generated only 25.6% of corrected results. Regarding Realism, DWC-GAN had an average score of 2.25, while StarGAN* had 1.17. Specifically, in 50.50% of the people said DWC-GAN were perfect, recognizable and without defects while only 2.82% of the people said the same for StarGAN*. StarGAN* results are indeed often judged "neutral, recognizable but with obvious defects" (52.91% of times). StackGAN++ and TAGAN do not achieve accuracy and realism higher than zero because they do not generate recognizable images.

These results show that DWC-GAN generates correct images, which can be mistaken with the real images at higher rates than competitors.

Progressive Manipulation A few state of the art approaches for text-to-image generation and image manipulation are expressly designed for conversational systems [23, 31, 32, 69, 94]. In Figure 3.5 we show that we can repeatedly apply our method to generated images to have a manipulation in multiples steps. This result highlights the consistency of our generated results, which can be used as input of our model, and that DWC-GAN might be applied in an interactive environment.

Domain Interpolation Our method generates interpolated images from two sentences. Figure 3.6 shows that by manipulating an image in two different ways, it is possible to generate the intermediate steps that go from one textual manipulation to the other.

User Study on Textual Commands Our training corpus is automatically generated, thus it might suffer limited generalizability. We conduct an experiment to verify these possible issues by asking 15 different people to describe in natural language how to edit an image A to make it similar to another image B. We asked people to do this experiment for 15 randomly chosen images. The resulting collected textual commands and their corresponding images are fed into our model to evaluate the results. We found

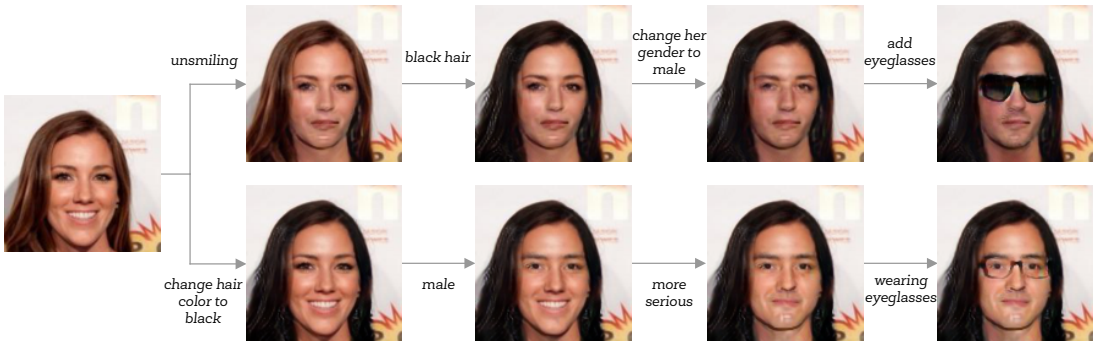


Figure 3.5: An example of progressive manipulation. Our method can used in an interactive environment.

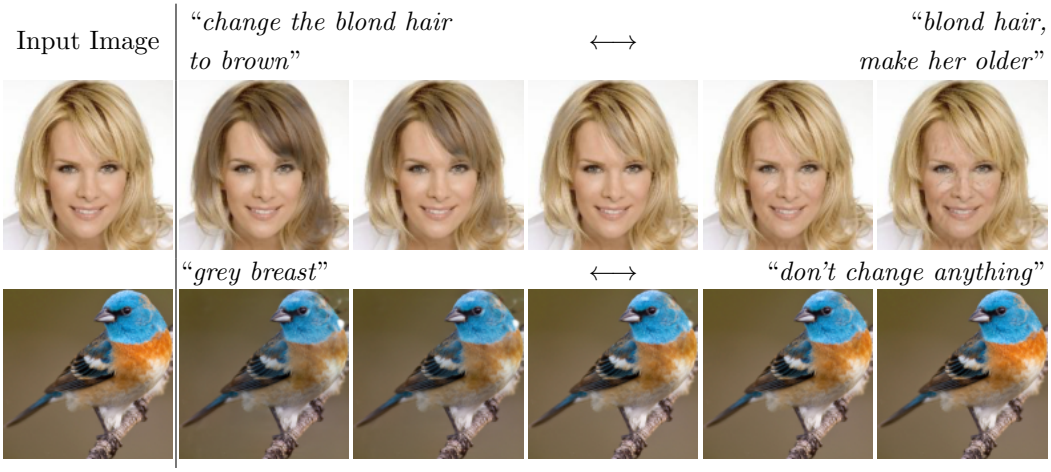


Figure 3.6: Domain interpolation given an input image.

that our model can be generalized to real human-written descriptions well.

Attention Visualization We visualize the unsupervised attention mask in Figure 3.7. It shows the localization area of different manipulations, which indeed helps to focus on the attribute-related pixels and preserves the background pixels.

Ablation Study We here investigate the effect of removing some components from our architecture. Results are showed in Table 3.5. First, we remove the *Cycle consistency*, which is widely employed by the literature of image-to-image translation [49, 149, 150]. We observe that both the image quality and diversity significantly performs worse without the cycle consistency, which indicates that the Cycle consistency loss indeed constraints

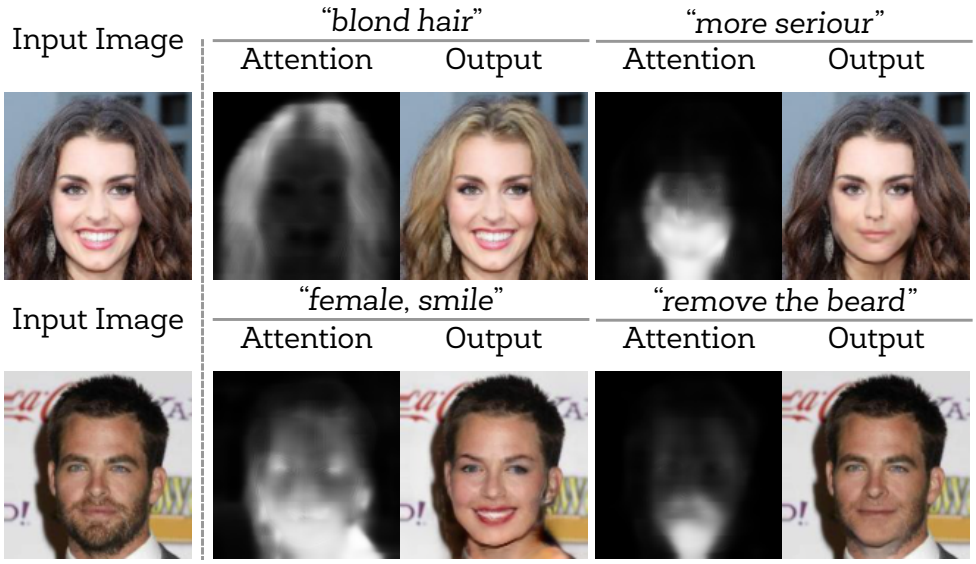


Figure 3.7: Unsupervised learned attention in DWC-GAN.

the network on the translation. Then, when we remove the *Attribute reconstruction*, which is applied to learn the attribute extractor E_a. Here, we use an attribute classifier to measure the F1 score. It shows that the constraint on the attribute latent variable slightly decreases the classification performance. However, both image quality and diversity are slightly improved. We also evaluate the contribution of pre-trained word embeddings by using our model without them. We observe that the use of pre-trained word embeddings improves the quality of generated images and converge the training faster. We observe that diversity sensitive loss indeed increases the diversities during the sampling, while slightly reduces the performances on FID and IS. In additional, $\mathcal{L}_{cyc}, \mathcal{L}_{rec\_a}$ and using pre-trained embeddings can also have effects on the diversities. We investigate whether the contribution of our model relies only on the use of the GMM to model the attributes. Thus, we constrain down our model to a StarGAN*-like model by setting the covariance matrix of the GMM to zero ($\forall k\ \mathbf{\Sigma}^k = 0$). As expected, we observe that our model behaves similarly but with no diversity (LPIPS: .000±.000). Moreover, this proves that our model outperforms

StarGAN* even in this case (see Table 3.3 for a comparison).

Table 3.5: Ablation study performance on the CelebA dataset.

Model	**IS**↑	**FID**↓	**LPIPS**↑	**F1 (%)**↑
DWC-GAN w/o $\mathcal{L}_{cyc}$	2.589	97.94	.042±.001	96.32
DWC-GAN w/o embedd.	2.782	73.90	.033±.001	98.06
DWC-GAN w/o $\mathcal{L}_{rec\_a}$	2.961	32.35	.072±.001	98.49
DWC-GAN $\forall k\ \mathbf{\Sigma}^k = 0$	2.872	33.17	.000±.000	**98.64**
DWC-GAN w/o $\mathcal{L}_{ds}$	**3.148**	**31.29**	.061±.001	94.69
DWC-GAN	3.069	32.14	**.152±.003**	94.80

3.5 Conclusion

In this paper, we presented a novel method to manipulate visual attributes through text. Our model builds upon recent literature of image to image translation, using unpaired images and modeling visual attributes through a GMM. Users interact with the model by feeding an image and a text that describes the desired alteration to the original image. The model extracts from the input image the visual attributes, understand the text and the desired transformation, and sample multiple images having the content of the input image and the style of the desired output. By sampling multiple synthesized images, our model deals with the inherent ambiguity of natural language and shows the user multiple transformations (e.g., multiple shadows and styles of blonde hair) from which she/he might select the best output.

To the best of our knowledge, we are the first at exploiting automatically generated sentences and unpaired images to manipulate image attributes with textual commands. We showed that our method can model multiple attributes, and that can be applied in a wide range of situations such as interpolation and progressive manipulation. We foresee that our work

will stimulate further research on attribute-based image manipulation and generation and will stimulate software integration in virtual assistants such as Alexa or image editors such as Photoshop.

In the next Chapter 4, we discuss how to constraint the smoothness of the style latent space of a general MMUIT model in details.

Chapter 4

Smoothing Style Latent Space

4.1 Introduction

Translating images from one domain to another is a challenging image manipulation task that has recently drawn increasing attention in the computer vision community [24, 25, 49, 50, 66, 74, 113, 149]. A "domain" refers to a set of images sharing some distinctive visual pattern, usually called "style" (e.g., the gender or the hair color in face datasets) [25, 49, 149]. The Image-to-Image (I2I) translation task aims to change the domain-specific aspects of an image while preserving its "content" (e.g., the identity of a person or the image background) [49]. Since paired data (e.g., images of the same person with different gender) are usually not available, an important aspect of I2I translation models is the unsupervised training [149]. Moreover, it is usually desirable to synthesize the multiple appearances *modes* within the same style domain, in such a way to be able to generate *diverse* images for the same input image.

Recent work addresses the I2I translation using multiple domains [24, 66, 25] and generating multi-modal outputs [66, 25]. These Multi-domain and Multi-modal Unsupervised Image-to-Image Translation (MMUIT) models are commonly evaluated based on the quality and the diversity of the generated images, including the results obtained by interpolating between

Figure 4.1: Our method generates smooth interpolations within and across domains in various image-to-image translation tasks. Here, we show gender, age and smile translations from CelebA-HQ [55] and animal translations from AFHQ [25].

two endpoints in their latent representations (e.g., see Figure 4.1). However, interpolations are usually computed using only points belonging to the same domain, and most of the state-of-the-art MMUIT methods are inclined to produce artifacts or unrealistic images when tested using across-domain interpolations. This is shown in Figure 4.2 (c), where, using the state-of-the-art StarGAN v2 [25], the inter-domain area in the style space frequently generates artifacts. Another common and related problem is the lack of graduality in both intra and inter domain interpolations, i.e., the generation of abrupt appearance changes corresponding to two close points in the latent space.

In this paper, we address the problem of learning a smoothed and disentangled style space for MMUIT models, which can be used for gradual and realistic image interpolations within and across domains. With "disentangled" we mean that the representations of different domains are well separated and clustered (Figure 4.2), so that intra-domain interpolations

correspond to only intra-domain images. With "smoothed" we mean that the semantics of the style space changes gradually and these changes correspond to small changes in the human perceptual similarity.

The main idea of our proposal is based on the hypothesis that the interpolation problems are related to the exploration of latent space areas which correspond to sparse training data. We again refer to Figure 4.2 to illustrate the intuition behind this observation. Many MMUIT methods use adversarial discriminators to separate the distributions of different domains [25]. However, a side-effect of this disentanglement process is that some areas of the latent space do not correspond to real data observed during training. Consequently, when interpolating in those areas, the decoding process may lead to generating unrealistic images. We propose to solve this problem jointly using a triplet loss [110, 11] and a simplified version of the Kullback-Leibler (KL) divergence regularization [60]. The former separates the domains using a small *margin* on their relative distance, while the latter encourages the style codes to lie in a compact space. The proposed simplified KL regularization does not involve the estimation of parametric distributions [60] and it can be easily plugged in Generative Adversarial Networks (GANs) [25, 10]. On the other hand, differently from adversarial discrimination, the triplet-loss margin can *control* the inter-domain distances and help to preserve the domain disentanglement in the compact space, Finally, we also encourage the content *preservation* during the translation using a perceptual-distance based loss. Figure 4.1 shows some interpolation results obtained using our method. In Section 5.4 we qualitatively and quantitatively evaluate our approach and we show that it can be plugged in different existing MMUIT methods improving their results. The last contribution of this paper concerns the proposal of the Perceptual Smoothness (PS) metric based on the perceptual similarity of the interpolated images, to quantitatively evaluate the style smoothness in

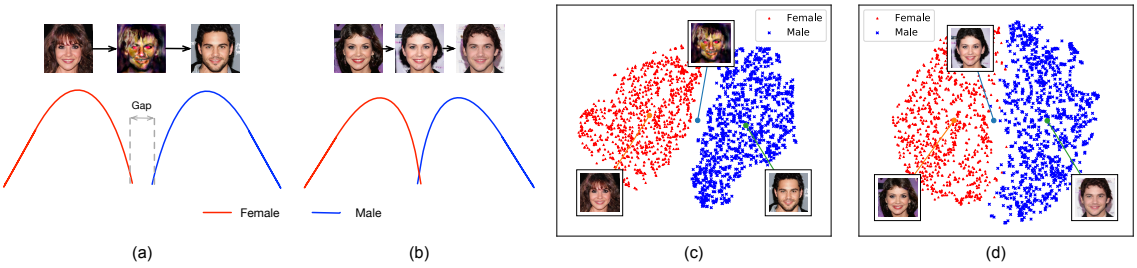


Figure 4.2: An illustration of the relation between smoothness and disentanglement of the style space. (a) Two well-separated distributions with a large margin in between. The intermediate area can lead to the generation of artifacts because it has not been sufficiently explored during training. (b) When the margin is reduced, the corresponding image appearance changes are smoother. (c) A t-SNE visualization of randomly sampled style codes using StarGAN v2 [25], which shows a disentangled style space but also that the inter-domain area generates images with artifacts. (d) The same visualization shows that, using our method, despite the disentanglement is preserved, the inter-domain area generates realistic images.

MMUIT models.

The **contributions** of this paper can be summarized as follows. First, we propose a new training strategy based on three specific losses which improve the interpolation smoothness and the content preservation of different MMUIT models. Second, we propose a novel metric to fill-in the gap of previous MMUIT evaluation protocols and quantitatively measure the smoothness of the style space.

4.2 Related Work

Unsupervised Domain Translation. Translating images from one domain to another without paired-image supervision is a challenging task. Different constraints have been proposed to narrow down the space of feasible mappings between images. Taigman *et al.* [119] minimize the feature-level distance between the generated and the source image. Liu *et al.* [72] create a shared latent space between the domains, which encourages dif-

ferent images to be mapped into the same space. CycleGAN [149] uses a cycle consistency loss in which the generated image is translated back to the original domain (an approach proved to be pivotal in the field [58, 5, 93]). However, all these approaches are limited to one-to-one domain translations, thus requiring $m(m-1)$ trained models for translations with m domains. StarGAN [24] was the first single-model for *multi-domain* translation settings. The generation process is conditioned by a target domain label, input to the generator, and by a domain classifier in the discriminator. However, the I2I translation of StarGAN is deterministic, since, for a given source image and target domain, only one target image can be generated (no multi-modality).

Multi-modal and Multi-domain Translation. After the pioneering works in supervised and one-to-one image translations [150, 49, 85], the recent literature is mainly focused in multiple-domains and multi-modal translations. Both DRIT++ [66] and SMIT [106] use a noise input vector and a domain label to increase the output diversity. StarGAN v2 [25] relies on a multitask discriminator [73] to model multiple domains, a noise-to-style mapping network, and a diversity sensitive loss [85] to explore the image space better. However, qualitative results show changes of subtle "content" details (e.g., the color of the eyes, the shape of the chin or the background) while translating the image with respect to the style (e.g., the hair colour or the gender).

Although MMUIT models do not require any image-level supervision, they still require set-level supervision (i.e., domain labels for each image). Very recently, TUNIT [10] proposed a "truly unsupervised" task where the network does not need any supervision. TUNIT learns the set-level characteristics of the images (i.e., the domains), and then it learns to map the images to all the domains. We will empirically show that our method can be used with both StarGAN v2 and TUNIT, and significantly improve

the interpolation smoothness with both models.

Latent-space interpolations. There is a quickly growing interest in the recent I2I translation literature with respect to latent space interpolations as a byproduct of the translation task. However, most previous works are only qualitatively evaluated, they use only intra-domain interpolations [65, 66, 106], or they require specific architectural choices. For example, DLOW [39] is a one-to-one domain translation, and RelGAN [132] uses a linear interpolation loss at training time, but it is not multi-modal. In StarGAN v2 [25], the style codes of different domains are very well disentangled, but the inter-domain interpolations show low-quality results (e.g., see Figure 4.2). HomoGAN [22] learns an explicit linear interpolator between images, but the generated images have very limited diversity.

Interestingly, image interpolations are not limited to the I2I translation field. The problem is well studied in Auto-Encoders [60, 15, 13] and in GANs [8, 56, 57], where the image is encoded into the latent space without an explicit separation between content and style. For example, StyleGAN [56] and StyleGANv2 [57] show high-quality interpolations of the latent space, where the latter has been further studied to identify the emerging semantics (e.g. linear subspaces) without retraining the network [111, 51, 148]. Richardson *et al.* [105] propose to find the latent code of a real image in the pre-trained StyleGAN space. This two-stage inversion problem allows multi-modal one-to-one domain mappings and interpolations. However, these methods are not designed to keep the source-image content while changing the domain-specific appearance. Thus, they are not suitable for a typical MMUIT task.

4.3 Problem Formulation and Notation

Let $\boldsymbol{\mathcal{X}} = \bigcup_{k=1}^{m} \boldsymbol{\mathcal{X}}_k$ be the image set composed of m disjoint domains ($\boldsymbol{\mathcal{X}}_i \cap \boldsymbol{\mathcal{X}}_j = \emptyset, i \neq j$), where each domain $\boldsymbol{\mathcal{X}}_k$ contains images sharing the same style. The goal of a multi-domain I2I translation model is to learn a single functional $G(i, j) = \boldsymbol{\mathcal{X}}_i \rightarrow \boldsymbol{\mathcal{X}}_j$ for all possible $i, j \in \{1, 2, \cdots, m\}$. The domain identity can be represented either using a discrete domain label (e.g., i) or by means of a style code $\boldsymbol{s}$, where $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}$ is a continuous vector and the set $\boldsymbol{\mathcal{S}}$ of all the styles may be either shared among all the domains or it can be partitioned in different domain-specific subsets (i.e., $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{S}}_1, \cdots, \boldsymbol{\mathcal{S}}_m\}$). In our case, we use the second solution and we denote with $\hat{\boldsymbol{x}} = G(\boldsymbol{x}, \boldsymbol{s})$ the translation operation, where $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_i$ is the *source* image (and its domain implicitly indicates the source domain i), $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}_j$ is the *target* style code and $\hat{\boldsymbol{x}} \in \boldsymbol{\mathcal{X}}_j$ is the generated image.

The MMUIT task is an extension of the above description in which:

a. *Training is unsupervised.* This is crucial when collecting paired images is time consuming or impossible.

b. *The source content is preserved.* A translated image $\hat{\boldsymbol{x}} = G(\boldsymbol{x}, \boldsymbol{s})$ should preserve domain-invariant characteristics (commonly called "content") and change only the domain-specific properties of the source image $\boldsymbol{x}$. For example, in male $\leftrightarrow$ female translations, $\hat{\boldsymbol{x}}$ should keep the pose and the identity of $\boldsymbol{x}$, while changing other aspects to look like a female or a male.

c. *The output is multi-modal.* Most I2I translations methods are deterministic, since, at inference time, they can produce only *one* translated image $\hat{\boldsymbol{x}}$ given a source image $\boldsymbol{x}$ and a target domain j. However, in many practical applications, it is desirable that the appearance of $\hat{\boldsymbol{x}}$ depends also on some random factor, in such a way to be able to produce different plausible translations.

There are mainly two mechanisms that can be used to obtain a specific style code $\boldsymbol{s} \in \boldsymbol{\mathcal{S}}_j$. The first option is to sample a random vector (e.g., $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$) and then use an MLP to transform $\boldsymbol{z}$ into a style code: $\boldsymbol{s} = M(\boldsymbol{z}, j)$ [56], where j is the domain label. The second option is based on extracting the code from a reference image ($\boldsymbol{x}' \in \boldsymbol{\mathcal{X}}_j$) by means of an encoder: $\boldsymbol{s} = E(\boldsymbol{x}')$. In our case, we use both of them.

4.4 Method

Figure 4.2 shows the main intuition behind our method. A style space in which different domains are well separated (i.e., disentangled) may not be sufficient to guarantee smooth inter-domain interpolations. When the domain-specific distributions are too far apart from each other, this may lead to what we call "training gaps", i.e., portions of the space that are not populated with training samples. Consequently, at training time, the network has not observed samples in those regions, and, at inference time, it may misbehave when sampling in those regions (e.g., producing image artifacts). Moreover, a non-compact style space may create intra-domain "training gaps", leading to the generation of non-realistic images when drawing style codes in these areas. Thus, we argue that smoothness is related to reducing these training gaps and compacting the latent space.

Note that the commonly adopted domain loss [24] or the multitask adversarial discriminators [25, 73] might result in domain distributions far apart from each other to facilitate the discriminative task. In order to reduce these training gaps, the domain distributions are expected to be pulled closer while keeping the disentanglement. To achieve these goals, we propose two training losses, described below. First, we use a triplet loss [110] to guarantee the separability of the style codes in different domains. The advantage of the triplet loss is that, using a small margin, the

disentanglement of different domains in the latent space can be preserved. Meanwhile, it is convenient to control the inter-domain distance by adjusting the margin. However, our empirical results show that the triplet loss alone is insufficient to reduce the training gaps. For this reason, we propose to compact style space using a second loss.

We propose to use the Kullback-Leibler (KL) divergence with respect to an a priori Gaussian distribution to make the style space compact. This choice is inspired by the regularization adopted in Variational AutoEncoders (VAEs) [60]. In VAEs, an encoder network is trained to estimate the parameters of a multivariate Gaussian given a single (real) input example. However, in our case, a style code $\boldsymbol{s}$ can be either real (using the encoder E, see Section 4.3) or randomly sampled (using M, Section 4.3), and training an additional encoder to estimate the distribution parameters may be hard and not necessary. For this reason, we propose to simplify the KL divergence using a sample-based ℓ_2 regularization.

Finally, as mentioned in Section 4.3, another important aspect of the MMUIT task is content preservation. To this aim, we propose to use a third loss, based on the idea that the content of an image should be domain-independent (see Section 4.3) and that the similarity of two images with respect to the content can be estimated using a "perceptual distance". The latter is computed using a network pre-trained to simulate the human perceptual similarity [145].

In Section 4.4.1 we provide the details of these three losses. Note that our proposed losses can be applied to different I2I translation architectures which have an explicit style space (e.g., a style encoder E, see Section 4.3), possibly jointly with other losses. In Section 4.4.2 we show a specific implementation case, which we used in our experiments and which is inspired to StarGAN v2 [25].

4.4.1 Modeling the Style Space

Smoothing and disentangling the style space. We propose to use a triplet loss, which is largely used in metric learning [110, 117, 45, 17], to preserve the domain disentanglement:

$$\mathcal{L}_{tri} = \mathbb{E}_{(\boldsymbol{s}_a,\boldsymbol{s}_p,\boldsymbol{s}_n)\sim\boldsymbol{\mathcal{S}}}[\max\left(||\boldsymbol{s}_a - \boldsymbol{s}_p)|| - ||\boldsymbol{s}_a - \boldsymbol{s}_n|| + \alpha, 0\right)], \tag{4.1}$$

where α is a constant margin and $\boldsymbol{s}_a$ and $\boldsymbol{s}_p$ (i. e., the *anchor* and the *positive*, adopting the common terminology of the triplet loss [110]) are style codes extracted from the same domain (e.g., $\boldsymbol{s}_a, \boldsymbol{s}_p \in \boldsymbol{\mathcal{S}}_i$), while the *negative* $\boldsymbol{s}_n$ is extracted from a different domain ($\boldsymbol{s}_n \in \boldsymbol{\mathcal{S}}_j, j \neq i$). These style codes are obtained by sampling real images and using the encoder. In more detail, we randomly pick two images from the same domain i ($\boldsymbol{x}_a, \boldsymbol{x}_p \in \boldsymbol{\mathcal{X}}_i$), a third image from another, randomly chosen, domain j ($\boldsymbol{x}_n \in \boldsymbol{\mathcal{X}}_j, j \neq i$), and then we get the style codes using $\boldsymbol{s}_k = E(\boldsymbol{x}_k), k \in \{a, p, n\}$. Using Eq. (4.1), the network learns to cluster style codes of the same domain. Meanwhile, when the style space is compact, the margin α can control and preserve the disentanglement among the resulting clusters.

Thus, we encourage a compact space forcing an a prior Gaussian distribution on the set of all the style codes $\boldsymbol{\mathcal{S}}$:

$$\mathcal{L}_{kl} = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}[\mathcal{D}_{\mathrm{KL}}(p(\boldsymbol{s})||\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))], \tag{4.2}$$

where $\boldsymbol{I}$ is the identity matrix, $\mathcal{D}_{\mathrm{KL}}(p||q)$ is the Kullback-Leibler (KL) divergence and $p(\boldsymbol{s})$ is the distribution corresponding to the style code $\boldsymbol{s}$. However, $p(\boldsymbol{s})$ is unknown. In VAEs, $p(\boldsymbol{s})$ is commonly estimated assuming a Gaussian shape and using an encoder to regress the mean and the covariance-matrix parameters of each single sample-based distribution [60]. Very recently, Ghosh et al. [37] showed that, assuming the variance to be constant for all the samples, the KL divergence regularization can be simplified (up to a constant) to $\mathcal{L}_{SR}^{CV}(\boldsymbol{x}) = ||\boldsymbol{\mu}(\boldsymbol{x})||_2^2$, where "CV" stands for

Constant-Variance, and $\boldsymbol{\mu}(\boldsymbol{x})$ is the mean estimated by the encoder using $\boldsymbol{x}$. In this paper we propose a further simplification based on the assumption that $\boldsymbol{\mu}(\boldsymbol{s}) = \boldsymbol{s}$ (which is reasonable if $\boldsymbol{\mu}$ is estimated using only one sample) and we eventually get the proposed *Style Regularization* (SR) loss:

$$\mathcal{L}_{SR} = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}[||\boldsymbol{s}||_2^2]. \tag{4.3}$$

Eq. (4.3) penalizes samples $\boldsymbol{s}$ with a large ℓ_2 norm, so encouraging the distribution of $\boldsymbol{\mathcal{S}}$ to be a shrunk Gaussian centered on the origin. Intuitively, while the SR loss compacts the space, the triplet loss avoids a domain entanglement in the compacted region. Finally, we describe below how the style-code samples are drawn in Eq. (4.3) ($\boldsymbol{s} \sim \boldsymbol{\mathcal{S}}$). We use a mixed strategy, including both real and randomly generated codes. More in detail, with probability 0.5, we use a real sample $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ and we get: $\boldsymbol{s} = E(\boldsymbol{x})$, and, with probability 0.5, we use $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{s} = M(\boldsymbol{z}, j)$. In practice, we alternate mini-batch iterations in which we use only real samples with iterations in which we use only generated samples.

Preserving the source content. The third loss we propose aims at preserving the content in the I2I translation:

$$\mathcal{L}_{cont} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}},\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}[\psi(\boldsymbol{x}, G(\boldsymbol{x}, \boldsymbol{s}))], \tag{4.4}$$

where $\psi(\boldsymbol{x}_1, \boldsymbol{x}_2)$ estimates the perceptual distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ using an externally pre-trained network. The rationale behind Eq. (4.4) is that, given a source image $\boldsymbol{x}$ belonging to domain $\boldsymbol{\mathcal{X}}_i$, for each style code $\boldsymbol{s}$, extracted from the set of *all* the domains $\boldsymbol{\mathcal{S}}$, we want to minimize the perceptual distance between $\boldsymbol{x}$ and the transformed image $G(\boldsymbol{x}, \boldsymbol{s})$. By minimizing Eq. (4.4), the perceptual content (extracted through $\psi(\cdot)$) is encouraged to be independent of the domain (see the definition of content preservation in Section 4.3). Although different perceptual distances can be used (e.g., the Euclidean distance on VGG features [53]), we implement

$\psi(\boldsymbol{x}_1, \boldsymbol{x}_2)$ using the Learned Perceptual Image Patch Similarity (LPIPS) metric [145], which was shown to be well aligned with the human perceptual similarity [145] and it is obtained using a multi-layer representation of the two input images $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ in a pre-trained network.

The sampling procedure in the *content preserving* loss ($\mathcal{L}_{cont}$) is similar to the SR loss. First, we randomly sample $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$. Then, we either sample a different reference image $\boldsymbol{x}' \in \boldsymbol{\mathcal{X}}$ and get $\boldsymbol{s} = E(\boldsymbol{x}')$, or we use $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\boldsymbol{s} = M(\boldsymbol{z}, j)$.

We sum together the three proposed losses and we get:

$$\mathcal{L}_{smooth} = \mathcal{L}_{cont} + \lambda_{sr}\mathcal{L}_{SR} + \mathcal{L}_{tri}, \tag{4.5}$$

where λ_{sr} is the SR loss-specific weight.

4.4.2 Smoothing the Style Space of an Existing Model

The proposed $\mathcal{L}_{smooth}$ can be plugged in existing MMUIT methods which have an explicit style space, by summing it with their original objective function ($\mathcal{L}_{orig}$):

$$\mathcal{L}_{new} = \mathcal{L}_{smooth} + \mathcal{L}_{orig}. \tag{4.6}$$

In this subsection, we show an example in which $\mathcal{L}_{orig}$ is the original loss of the MMUIT state-of-the-art StarGAN v2 [25].

In StarGAN v2, the original loss is:

$$\mathcal{L}_{orig} = \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc} + \mathcal{L}_{adv} \tag{4.7}$$

where λ_{sty}, λ_{ds} and λ_{cyc} control the contribution of the *style reconstruction*, the *diversity sensitive*, and the *cycle consistency* loss, respectively.

The *style reconstruction* loss [49, 150, 25] pushes the target code ($\boldsymbol{s}$) and the code extracted from the generated image ($E(G(\boldsymbol{x}, \boldsymbol{s}))$) to be as close as possible:

$$\mathcal{L}_{sty} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}},\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}\left[\|\boldsymbol{s} - E(G(\boldsymbol{x}, \boldsymbol{s}))\|_1\right]. \tag{4.8}$$

The *diversity sensitive* loss [25, 86] encourages G to produce diverse images:

$$\mathcal{L}_{ds} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}_i,(\boldsymbol{s}_1,\boldsymbol{s}_2)\sim\boldsymbol{\mathcal{S}}_j}\left[\|G(\boldsymbol{x},\boldsymbol{s}_1) - G(\boldsymbol{x},\boldsymbol{s}_2)\|_1\right]. \tag{4.9}$$

The *cycle consistency* [149, 24, 25] loss is used to preserve the content of the source image $\boldsymbol{x}$:

$$\mathcal{L}_{cyc} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}},\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}\left[\|\boldsymbol{x} - G(G(\boldsymbol{x},\boldsymbol{s}), E(\boldsymbol{x}))\|_1\right]. \tag{4.10}$$

Finally, StarGAN v2 uses a multitask discriminator [73] D, which consists of multiple output branches. Each branch D_j learns a binary classification determining whether an image $\boldsymbol{x}$ is a real image of its dedicated domain j or a fake image. Thus, the *adversarial* loss can be formulated as:

$$\mathcal{L}_{adv} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}}_i,\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}_j}[\log D_i(\boldsymbol{x}) + \log(1 - D_j(G(\boldsymbol{x},\boldsymbol{s})))] \tag{4.11}$$

Note that this loss encourages the separation of the domain-specific distributions without controlling the relative inter-domain distance (Section 5.3). We use it jointly with our $\mathcal{L}_{tri}$.

We refer the reader to [25] for additional details. In Section 5.4 we evaluate the combination of our $\mathcal{L}_{smooth}$ with StarGAN v2 (Eq. (4.7)).

4.5 Evaluation Protocols

FID. For each translation $\boldsymbol{\mathcal{X}}_i \rightarrow \boldsymbol{\mathcal{X}}_j$, we use 1,000 test images and estimate the Fréchet Inception Distance (FID) [46] using interpolation results. In more detail, for each image, we randomly sample two style codes ($\boldsymbol{s}_1 \in \boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{s}_2 \in \boldsymbol{\mathcal{S}}_j$), which are linearly interpolated using 20 points. Each point (included $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$) is used to generate a translated image. The FID values are computed using the $20 \times 1,000$ outputs. A lower FID score indicates a lower discrepancy between the image quality of the real and generated images.

LPIPS. For a given domain $\boldsymbol{\mathcal{X}}_i$, we use 1,000 test images $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_i$, and, for each $\boldsymbol{x}$, we randomly generate 10 image translations in the target domain $\boldsymbol{\mathcal{X}}_j$. Then, the LPIPS [145] distances among the 10 generated images are computed. Finally, all distances are averaged. A higher LPIPS distance indicates a greater diversity among the generated images. Note that the LPIPS distance ($\psi(\boldsymbol{x}_1, \boldsymbol{x}_2)$) is computed using an *externally pre-trained* network [145], which is the same we use in Eq. (4.4) at training time.

FRD. For the specific case of face translations, we use a metric based on a pretrained VGGFace2 network (ϕ) [110, 17], which estimates the visual distance between two faces. Note that the identity of a person may be considered as a specific case of "content" (Section 4.3). We call this metric the Face Recognition Distance (FRD):

$$\text{FRD} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}},\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}\left[\|\phi(\boldsymbol{x}) - \phi(G(\boldsymbol{x}, \boldsymbol{s}))\|_2^2\right]. \tag{4.12}$$

PS. Karras *et al.* [56] recently proposed the Perceptual Path Length (PPL) to evaluate the smoothness and the disentanglement of a semantic latent space. PPL is based on measuring the LPIPS distance between close points in the style space. However, one issue with the PPL is that it can be minimized by a collapsed generator. For this reason, we alternatively propose the Perceptual Smoothness (PS) metric, which returns a normalized score in $[0, 1]$, indicating the smoothness of the style space.

In more detail, let $\boldsymbol{s}_0$ and $\boldsymbol{s}_T$ be two codes randomly sampled from the style space, $P = (\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T)$ the sequence of the linearly interpolated points between $\boldsymbol{s}_0$ and $\boldsymbol{s}_T$, and $A = (G(\boldsymbol{x}, \boldsymbol{s}_0), \ldots, G(\boldsymbol{x}, \boldsymbol{s}_T))$ the corresponding sequence of images generated starting from a source image $\boldsymbol{x}$. We measure the degree of linear *alignment* of the generated images using:

$$\ell_{\texttt{alig}} = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\mathcal{X}},\boldsymbol{s}_0,\boldsymbol{s}_T\sim\boldsymbol{\mathcal{S}}}\left[\frac{\delta(\boldsymbol{x}, \boldsymbol{s}_0, \boldsymbol{s}_T)}{\sum_{t=1}^{T}\delta(\boldsymbol{x}, \boldsymbol{s}_{t-1}, \boldsymbol{s}_t)}\right] \tag{4.13}$$

where $\delta(\boldsymbol{x}, \mathbf{s}_1, \mathbf{s}_2) = \psi(G(\boldsymbol{x}, \mathbf{s}_1), G(\boldsymbol{x}, \mathbf{s}_2))$ and $\psi(\cdot, \cdot)$ is the LPIPS distance.

When $\ell_{\texttt{alig}} = 1$, then the perceptual distance between $G(\boldsymbol{x}, \boldsymbol{s}_0)$ and $G(\boldsymbol{x}, \boldsymbol{s}_T)$ is equal to the sum of the perceptual distances between consecutive elements in A, thus, the images in A lie along a line in the space of $\psi(\cdot, \cdot)$ (which represents the human perceptual similarity [145]). Conversely, when $\ell_{\texttt{alig}} < 1$, then the images in A contain some visual attribute not contained in any of the endpoints. For example, transforming a short-hair male person to a short-hair girl, we may have $\ell_{\texttt{alig}} < 1$ when the images in A contain people with long hair. However, although aligned, the images in A may have a non-uniform distance, in which $\delta(\boldsymbol{x}, \boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$ varies depending on t. In order to measure the *uniformity* of these distances, we use the opposite of the Gini inequality coefficient [38]:

$$\ell_{\texttt{uni}} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}} \boldsymbol{s}_0, \boldsymbol{s}_T \sim \boldsymbol{\mathcal{S}}} \left[1 - \frac{\sum_{i,j=1}^{T} |\delta(\boldsymbol{x}, \boldsymbol{s}_{i-1}, \boldsymbol{s}_i) - \delta(\boldsymbol{x}, \boldsymbol{s}_{j-1}, \boldsymbol{s}_j)|}{2T^2 \mu_P} \right] \tag{4.14}$$

where μ_P is the average value of $\delta(\cdot)$ computed over all the pairs of elements in $P = (\boldsymbol{s}_0, \ldots, \boldsymbol{s}_T)$. Intuitively, $\ell_{\texttt{uni}} = 1$ when an evenly-spaced linear interpolation of the style codes corresponds to constant changes in the perceived difference of the generated images, while $\ell_{\texttt{uni}} = 0$ when there is only one abrupt change in a single step. Finally, we define PS as the harmonic mean of $\ell_{\texttt{alig}}$ and $\ell_{\texttt{uni}}$:

$$\text{PS} = 2 \cdot \frac{\ell_{\texttt{alig}} \cdot \ell_{\texttt{uni}}}{\ell_{\texttt{alig}} + \ell_{\texttt{uni}}} \in [0, 1]. \tag{4.15}$$

4.6 Experiments

Baselines. We compare our method with three state-of-the-art approaches: (1) StarGAN v2 [25], the state of the art for the MMUIT task; (2) HomoGAN [22]; and (3) TUNIT [10]. Moreover, as a reference for a high image quality, we also use InterFaceGAN [111], a StyleGAN-based method (trained with 1024×1024 images) which interpolates the pre-trained se-

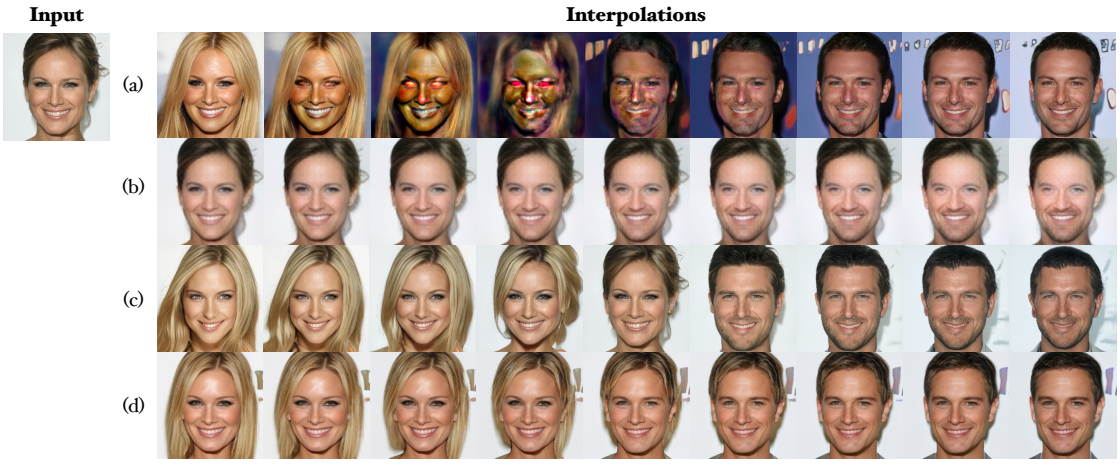


Figure 4.3: Inter-domain interpolation results: (a) StarGAN v2 [25], (b) HomoGAN [22], (c) InterFaceGAN [111], (d) ours. The domains correspond to genders. Our method generates smoother results while better preserving the source-person identity.

mantic space of StyleGAN [56] (see Section 4.2). InterFaceGAN is not designed for domain translation and for preserving the source content, but it can linearly interpolate a fixed latent space, massively trained with high-resolution images. All the baselines are tested using the original publicly available codes.

Datasets. We follow the experimental protocol of StarGAN v2 [25] and we use the CelebA-HQ [55] and the AFHQ dataset [25]. The domains are: *male-female*, *smile-no smile*, *young-non young* in CelebA-HQ; *cat*, *dog*, and *wildlife* in AFHQ. For a fair comparison, all models (except InterFaceGAN) are trained with 256×256 images.

Settings. We test our method in two experimental settings, respectively called "unsupervised" (with only set-level annotations) and "truly unsupervised" (no annotations [10]). Correspondingly, we plug our training losses ($\mathcal{L}_{smooth}$) in the state-of-the art StarGAN v2 [25] and TUNIT [10] (see Section 4.4.1). In each setting, we plug our method in the original architecture without adding additional modules and adopting the original hyper-parameter values without tuning.

4.6.1 Model Architecture

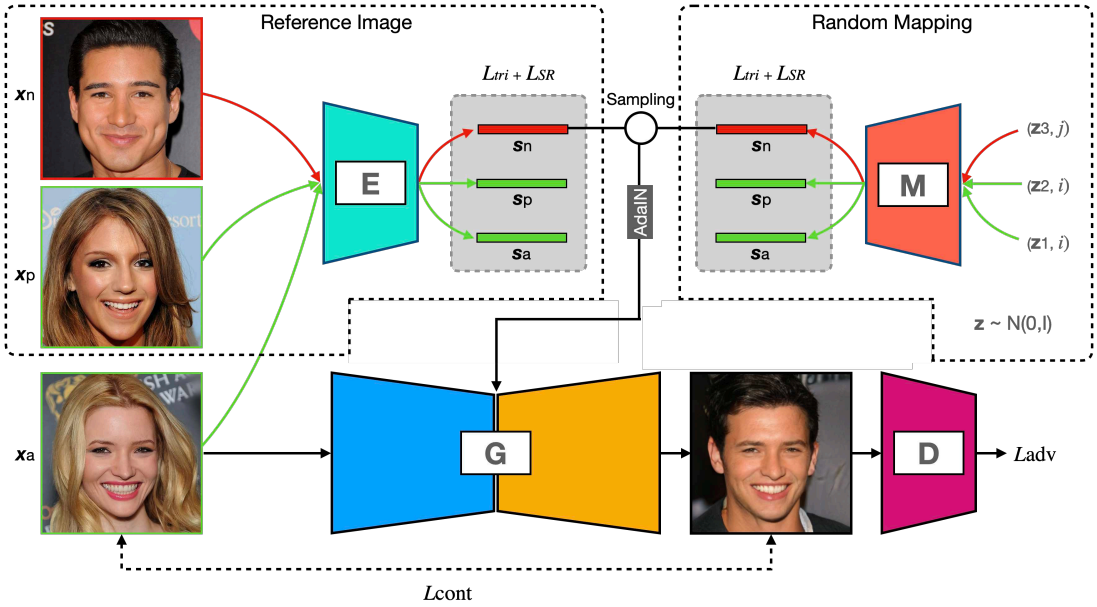


Figure 4.4: Our MMUIT generative framework and the style-code sampling strategies.

Figgure 4.4 shows the framework of our proposed method for MMUIT tasks. The model is composed of an image generator G, a discriminator D, an encoder E and an MLP M. G generates a new image from a source image $\boldsymbol{x}_a$ and a style code $\boldsymbol{s}$, which can either be extracted from a reference image (i.e., $\boldsymbol{s}_p = E(\boldsymbol{x}_p)$, or from a randomly sampled vector $\boldsymbol{z} \sim N(0, 1)$ through $\boldsymbol{s}_p = M(\boldsymbol{z})$. The discriminator D learns to classify an image as either a real image in its associated domain, or a fake image.

As explained in the main paper, we use $\mathcal{L}_{tri}$, $\mathcal{L}_{SR}$ and $\mathcal{L}_{cont}$ to compact and disentangle the style space and to help preserving the source content. In Figure 4.4, $\boldsymbol{s}_n$ is a style code of a domain different from the domain shared by $\boldsymbol{s}_p$ and $\boldsymbol{s}_a$.

4.6.2 Smoothness of the Style Space

Figure 4.3 shows a qualitative evaluation using the style-space interpolation between a source image and a reference style. As mentioned in Section 5.1

Table 4.1: Smoothness degree and identity preservation on the CelebA-HQ dataset.

Model	PS↑			FRD↓		
	Gender	Smile	Age	Gender	Smile	Age
HomoGAN [22]	.401	.351	.389	.903	.820	.842
StarGAN v2 [25]	.272	.282	.283	1.082	.894	.882
Ours	**.504**	**.513**	**.601**	**.837**	**.625**	**.650**
InterFaceGAN [111]§	.328	.436	.409	.884	.560	.722

§ Trained on 1024×1024 images.

Table 4.2: Image quality and translation diversity on the CelebA-HQ dataset.

Model	FID↓			LPIPS↑		
	Gender	Smile	Age	Gender	Smile	Age
HomoGAN [22]	55.23	58.02	57.50	.010	.005	.008
StarGAN v2 [25]	48.35	29.65	26.60	**.442**	**.413**	**.407**
Ours	**23.37**	**22.21**	**23.57**	.337	.095	.128
InterFaceGAN [111]§	13.75	12.81	12.25	.211	.115	.146

§ Trained on 1024×1024 images.

and 5.3, StarGAN v2 frequently generates artifacts in inter-domain interpolations (see Figure 4.3 (a)). HomoGAN results are very smooth, but they change very little the one from the other, and the model synthetizes lower quality images (Figure 4.3 (b)). InterFaceGAN (Figure 4.3 (c)) was trained at a higher image resolution with respect to the other models (ours included). However, compared to our method (Figure 4.3 (d)), the interpolation results are less smooth, especially in the middle, while the image quality of both methods is very similar. Moreover, comparing our approach to StarGAN v2, our method better preserves the background content in all the generated images.

These results are quantitatively confirmed in Table 4.1. The PS scores show that our proposal improves the state of the art significantly, which

means that it increases the smoothness of the style space in all the CelebA-HQ experiments. Note that our results are also better than InterFaceGAN, whose latent space is based on the pretrained StyleGAN [56], a very large capacity and training-intensive model. Table 4.4 and Figure 4.11 show similar results also in the challenging AFHQ dataset, where there is a large inter-domain shift. In this dataset, we tested both the unsupervised and the truly unsupervised setting, observing a clear improvement of both the semantic-space smoothness and the image quality using our method.

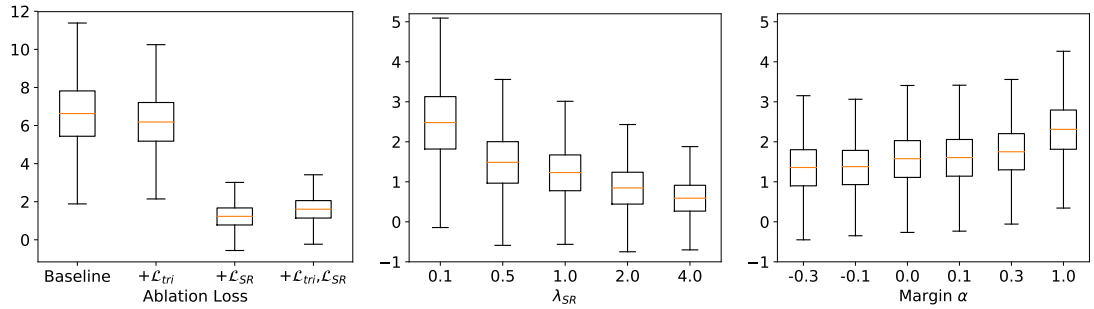
The comparison of the qualitative results in Figure 4.3 and Figure 4.11 with the PS scores in Table 4.1 and Table 4.4, respectively, show that the proposed PS metric can be reliably used to evaluate MMUIT models with respect to the style-space smoothness. In Figure 4.6, we show additional evidence on the quality of the PS metrics and how domain separation can be controlled by tuning the margin value of the triplet loss.

Table 4.2 and 4.4 show that the improvements on the style-space smoothness and the corresponding interpolation results do not come at the expense of the image quality. Conversely, these tables show that the FID values significantly improve with our method. The LPIPS results in Table 4.2 also show that HomoGAN generates images with little diversity. However, the LPIPS scores of StarGAN v2 are higher than our method. Nevertheless, the LPIPS metric is influenced by the presence of possible artifacts in the generated images, and, thus, an increased LPIPS value is not necessarily a strength of the model.

Finally, we performed a user study where we asked 40 users to choose between the face translations generated by StarGAN v2 and our method, providing 30 random image pairs to each user. In 75.8% of cases, the image generated by our model was selected as the better one, compared to StarGAN v2 (25.2%).

4.6.3 Analysing the Style-Space Compactness

Inter-domain Distance Distributions. In order to estimate the inter-domain distances and the degree of compactness of a high-dimensional semantic space, we compute the distribution of the distances $(d_s(\boldsymbol{s}_a, \boldsymbol{s}_n) - d_s(\boldsymbol{s}_a, \boldsymbol{s}_p))$. Specifically, we use the CelebA-HQ dataset [55] and we randomly sample 10,000 triplets $(\boldsymbol{s}_a, \boldsymbol{s}_p, \boldsymbol{s}_n)$ where $\boldsymbol{s}_a \sim \boldsymbol{\mathcal{S}}_i$, $\boldsymbol{s}_p \sim \boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{s}_n \sim \boldsymbol{\mathcal{S}}_j$ with $i \neq j$. Figure 4.5 shows the distribution of $(d_s(\boldsymbol{s}_a, \boldsymbol{s}_n) - d_s(\boldsymbol{s}_a, \boldsymbol{s}_p))$ under different experimental settings.



(a) (b) (c)

Figure 4.5: Distribution of $(d_s(\mathbf{s}_a, \mathbf{s}_n) - d_s(\mathbf{s}_a, \mathbf{s}_p))$ on different experimental settings on the CelebA-HQ dataset. (a) shows that $\mathcal{L}_{SR}$ helps to compact the style space, while $\mathcal{L}_{tri}$ can adjust the distance between the style clusters. (b) shows that the weight of the $\mathcal{L}_{SR}$ can control the compactness of the style space. (c) shows that increasing the margin α in $\mathcal{L}_{tri}$ has an effect on the distances between clusters.

Figure 4.5 (a) shows that the distance distribution of the baseline system (without using $\mathcal{L}_{tri}$ and $\mathcal{L}_{SR}$) is relatively wide and corresponds to the largest median. Our $\mathcal{L}_{tri}$ loss with a small margin can slightly reduce both the range between the lower quartile to upper quartile and the range between the minimum to the maximum score. Conversely, $\mathcal{L}_{SR}$ ($\lambda_{SR} = 1.0$) compacts the space significantly. Jointly using $\mathcal{L}_{SR}$ and $\mathcal{L}_{tri}$ ($\alpha = 0.1$), the $\mathcal{L}_{SR}$-only distribution is slightly shifted up. Figure 4.5 (b) shows the impact of λ_{SR} when we use $\mathcal{L}_{SR}$ without $\mathcal{L}_{tri}$. Conversely, Figure 4.5 (c) analyses the case of jointly using $\mathcal{L}_{SR}$ (with $\lambda_{SR} = 1.0$) and $\mathcal{L}_{tri}$ while changing the

margin α. The latter experiment shows that the Triplet Margin loss can adjust the distance between style clusters, since the ranges between the minimum and the maximum score are shifted when using a larger α.

The corresponding PS scores are presented in Figure 4.6, which shows that increasing λ_{SR} helps smoothing the space, but when $\lambda_{SR} > 0.5$, only limited improvements are obtained (see Figure 4.6 (a)).

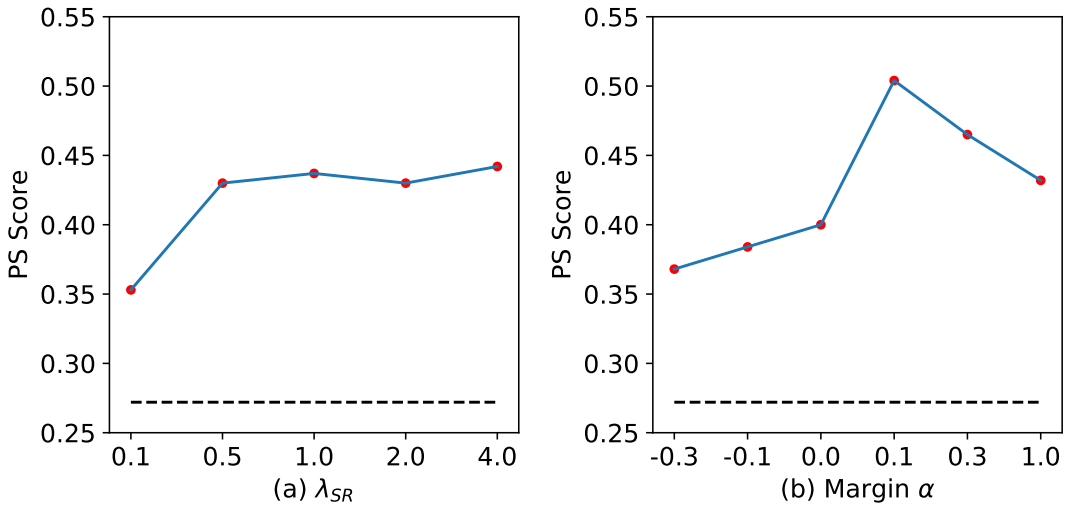


Figure 4.6: An ablation study on the influence of both (a) the SR loss weigh λ_{SR} and (b) the triplet loss margin α ($\lambda_{SR} = 1.0$) in the PS scores. The black dashed line refers to StarGAN v2 [25].

As shown in the main paper, the Triplet loss significantly influences the image quality and smoothness of I2I translations. Interestingly, the margin α also plays an important role. Using a small positive margin (e.g., 0.1) is enough to keep the disentanglement and achieve the best PS score, as shown in Figure 4.6 (b). Meanwhile, a large margin can push the style clusters far away from each other, which may be harmful for the smoothness degree of the space.

An Alternative Style Regularization. A possible alternative to the style-regularization loss ($\mathcal{L}_{SR}$), is based on the following formulation, whose goal is to compact the style codes close to the surface of the zero-centered,

n-dimensional unit sphere:

$$\mathcal{L}_{sph} = \mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\mathcal{S}}}\left[\left|\|\boldsymbol{s}\|_2 - 1\right|\right] \tag{4.16}$$

where $\|\cdot\|_2$ is the L_2 norm. Note that, since the volume of the whole n-sphere is larger than the volume of its surface, $\mathcal{L}_{sph}$ leads to a much more compact space compared to $\mathcal{L}_{SR}$. Table 4.3 quantitatively compares $\mathcal{L}_{sph}$ with $\mathcal{L}_{SR}$ and shows that a very compact space ($\mathcal{L}_{sph}$) leads to a higher smoothness but with a low diversity. This finding is qualitatively confirmed in Figure 4.7. This comparison indicates that there exists a trade-off between the smoothness of the space and the diversity of generated images.

Table 4.3: A comparisons between $\mathcal{L}_{SR}$ and $\mathcal{L}_{sph}$ on a gender translation task using the CelebA-HQ dataset.

Model	FID↓	LPIPS↑	PS↑	FRD↓
$\mathcal{L}_{SR}$	**23.37**	**.337**	.504	.837
$\mathcal{L}_{sph}$	23.66	.103	**.897**	**.808**

Figure 4.7: Visual comparisons between (a) $\mathcal{L}_{SR}$ and (b) $\mathcal{L}_{sph}$.

A Space Visualization Experiment. We perform an additional experiment on the MNIST dataset [61] to interpret the results of our model and directly visualize the distributions of style codes. In this experiment, we consider the categories of handwritten digits as "styles" and we set the dimension of style codes to 2, such that they can be easily plotted in a two-dimensional coordinate system without reducing the representation

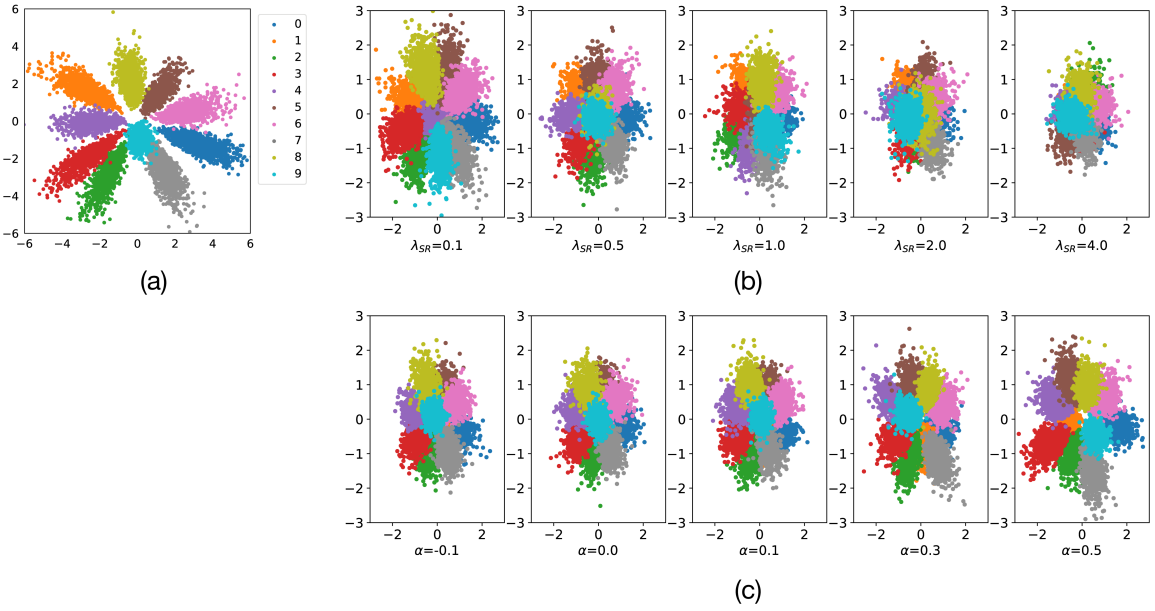


Figure 4.8: The distributions of style codes on a MNIST-based toy experiment. The original latent style space (a), using only $\mathcal{L}_{SR}$ with different loss weights λ_{SR} (b), and using $\mathcal{L}_{SR}$ ($\lambda_{SR} = 1.0$) and $\mathcal{L}_{tri}$ with different margin values α (c).

dimensionality with non-linear projections (e.g. t-SNE). As shown in Figure 4.8 (a), the original style codes without using our proposed losses, is scattered in a non-compact space, where there are many "training gaps". Once we increase the weight of λ_{SR}, the style codes are pushed in a more compact space. However, the clusters (i.e., the domains) are highly entangled, as shown in Figure 4.8 (b). Conversely, the triplet loss alleviates this issue by separating the compacted clusters, as shown in Figure 4.8 (c).

Moreover, we select two clusters with large "training gaps" (i.e., "2" (green color) and "7" (grey color)) in the original space Figure 4.8 (a). Figure 4.9 (a) shows an example of interpolation results between "2" and "7" with large "training gaps", showing, as expected, that the generated images contain artifacts. Figure 4.9 (b) refers to the same interpolation between "2" and "7" in the setting with $\lambda_{SR} = 1.0$. It seems that, due to the cluster overlapping, the interpolation traverses another cluster (i.e., "4") while moving from "2" to "7". Finally, the triplet loss is able to disen-

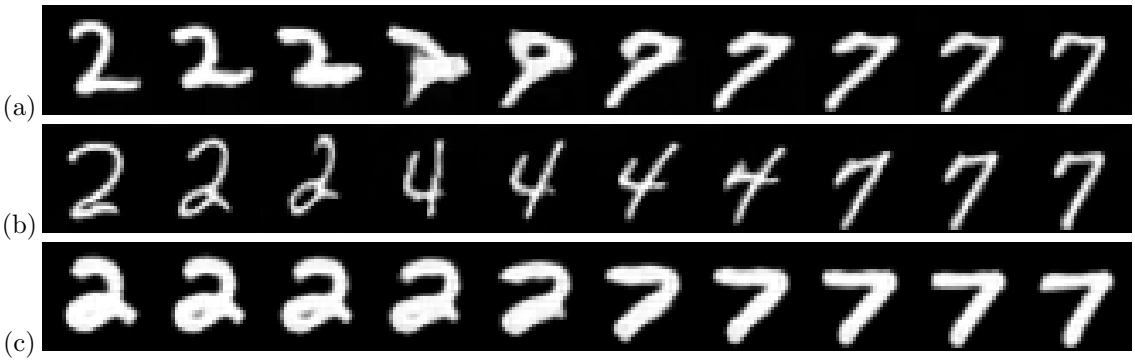


Figure 4.9: Interpolations results on MNIST between domain "2" and domain "7". (a) Original space, (b) Using only $\mathcal{L}_{SR}$ ($\lambda_{SR} = 1.0$). (c) Using $\mathcal{L}_{SR}$ ($\lambda_{SR} = 1.0$) and $\mathcal{L}_{tri}$ ($\alpha = 0.5$).

Figure 4.10: Content preservation using the CelebA-HQ dataset. Our method better preserves the ethnicity and identity of the source images compared to StarGAN v2.

tangle the compact space, as shown in Figure 4.9 (c), where no "intruder" is generated when interpolating between the two domains.

4.6.4 Identity Preservation

MMUIT models aim at translating images from one domain to another while keeping the content unchanged. While this goal is clear, the degree of content preservation is usually evaluated only qualitatively. Thus, we use the FRD (Section 4.5) and the most popular I2I translation task (face translation) to measure the content preservation of the compared models.

Table 4.1 shows that our FRD is the lowest over all the methods compared on the CelebA-HQ dataset, indicating that our method better maintains the person identity of source images. Qualitatively, Figure 4.10 shows that our method better preserves some distinct face characteristics (e.g., the eye color, the chin shape, or the ethnicity) of the source image while changing the style (i.e., the gender). This result also suggests that our model might be less influenced by the CelebA-HQ biases (e.g., Caucasian people).

4.6.5 Ablation Study

In this section, we evaluate the importance of each proposed component. Table 5.3 shows the FID, LPIPS, PS and FRD values for all the configurations, where each component is individually added to the baseline StarGAN v2, using CelebA-HQ. First, we observe that adding the $\mathcal{L}_{tri}$ loss to the baseline improves the quality, the diversity and the content preservation of the generated images. However the PS score decreases. This result suggests that better disentanglement might separate too much the styles between domains, thus decreasing the interpolation smoothness. The addition of $\mathcal{L}_{SR}$ helps improving most of the metrics but the diversity, showing that a more compact style space is a desirable property for MMUIT. As mentioned before, we note that higher diversity (LPIPS) might not be strictly related to high-quality images.

The combination of the two proposed smoothness losses dramatically improves the quality of generated images and the smoothness of the style space. This suggests that the style space should be compact and disentangled, while keeping the style clusters of different domains close to each other. Finally, $\mathcal{L}_{cont}$ further improves the FID, the PS and the FRD scores. The final configuration corresponds to our full-method and confirms that all the proposed components are helpful.

Table 4.4: Quantitative evaluation on the AFHQ dataset.

Model	Setting	FID↓	PS↑
StarGAN v2 [25]	Unsupervised	15.64	.226
Ours		**14.67**	**.301**
TUNIT [10]	Truly Unsupervised	29.45	.443
Ours		**16.59**	**.447**

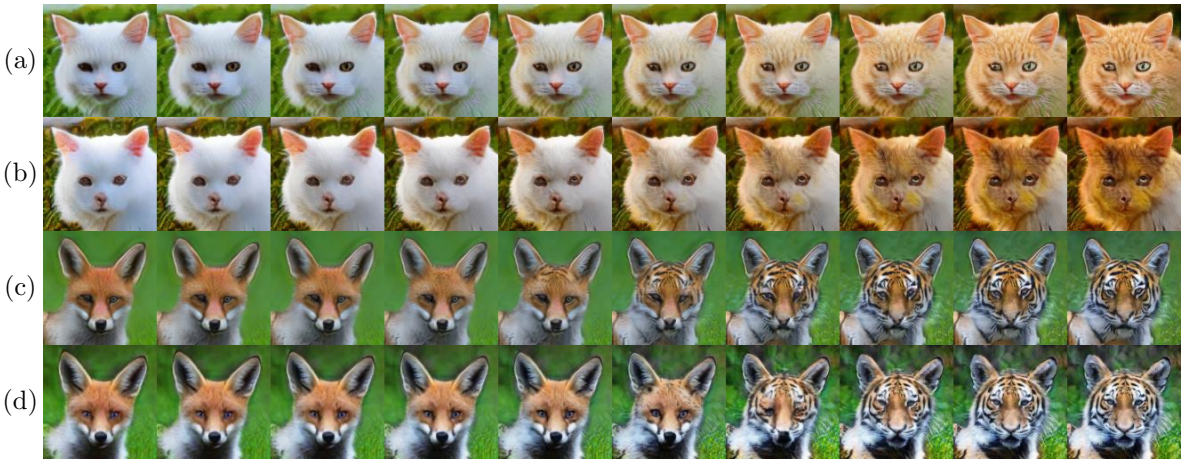


Figure 4.11: AFHQ dataset. (b,d) Generation results using TUNIT [10]. (a,c) TUNIT jointly with our losses.

Table 4.5: Ablation study on the CelebA-HQ dataset with a gender translation task.

Model	FID↓	LPIPS↑	PS↑	FRD↓
A: Baseline StarGAN v2 [25]	48.35	**.442**	.272	1.082
A + $\mathcal{L}_{tri}$	37.54	.403	.292	1.040
A + $\mathcal{L}_{SR}$	35.23	.368	.432	.912
A + $\mathcal{L}_{SR}$, $\mathcal{L}_{tri}$	24.29	.374	.501	.848
A + $\mathcal{L}_{SR}$, $\mathcal{L}_{tri}$, $\mathcal{L}_{cont}$	**23.37**	.337	**.504**	**.837**

4.7 Conclusion

In this paper, we proposed a new training strategy based on three specific losses which jointly improve both the smoothness of the style space and the content preservation of existing MMUIT models. We also proposed the PS metric, which specifically evaluates the style smoothness of I2I transla-

tion models. The experimental results show that our method significantly improves both the smoothness and the quality of the interpolation results and the translated images.

In the next Chapter 5, we show a novel method to use pretrained unconditional GANs for MMUIT tasks at high resolution (e.g., 1024×1024).

Chapter 5

Implicit Style Function

5.1 Introduction

Generative methods have become increasingly effective at synthesizing realistic images at high resolution, stimulating new practical applications in academia and industry. Many different tasks have been proposed, from super-resolution [63, 90] to image manipulation [64, 52], from image-to-image and text-to-image translations [24, 25, 67, 75] to video generation [114, 123]. Yet, training these task-specific models at high resolution (e.g., 1024×1024) is very computationally expensive. For this reason, recent works have interpreted and exploited the latent space of pre-trained high-resolution unconditional Generative Adversarial Networks (GANs) to solve several generative tasks without training a generator from scratch [1, 2, 3, 4, 98, 121, 146]. Most of these approaches are based on StyleGAN ([56, 57]), the state-of-the-art of unconditional image generation. StyleGAN maps a noise vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ to an intermediate and learned latent space $\boldsymbol{\mathcal{W}}^+$, which exhibits some intriguing disentangled semantic properties [26, 43, 134] that can be interpreted and exploited. For example, InterFaceGAN [111, 112] identifies the semantic attributes in $\boldsymbol{\mathcal{W}}^+$ to manipulate the semantics of a latent code and change the facial attributes of images. However, existing models manipulating StyleGAN latent codes

allow to edit just one attribute per time or fail to preserve the content of the image not involved in the manipulation, resulting, for example, in identity changes when gender is manipulated (see Figure 5.1). Moreover, most existing models are deterministic (i.e., not multi-modal). Altogether, these issues make existing approaches unsuitable for Multi-modal, and Multi-domain Unsupervised Image-to-image Translation (MMUIT) [25, 78].

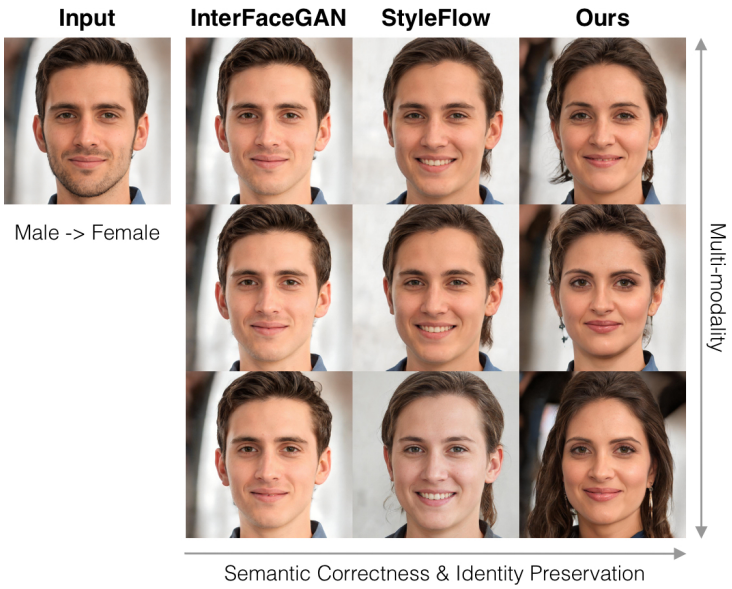


Figure 5.1: Our model focuses on Multi-modal and Multi-domain Unsupervised Image-to-image Translation. In this figure, we show an male→female translation, in which we wish to change the gender of the input image without changing some facial features that allows us to recognize that the input image and the output image depict the same person. Additionally, we want to generate multiple diverse images for each translation. We can observe that state-of-the-art models based on StyleGAN, do not usually maintain the content of the original image (e.g., background and people's identity), do not generate images with correct semantics, and have limited diversity. Our model better adheres to these properties.

An ideal MMUIT model should be able to change the *domain*-specific parts of the image while preserving the *content* of the image (e.g., the background and the identity of people) and synthesize images considering the multiple appearance *modes* within each domain [25, 78]. Here, *domain*

refers to a set of images having some distinctive visual pattern, usually called *style*. For example, we can group images based on the gender of people in pictures (see Figure 5.1). Since paired images (e.g., pictures of the same person with different gender) are usually not available, the model should be unsupervisedly trained [24, 25, 78, 49].

This paper proposes an implicit style function (ISF) that allows leveraging a pre-trained unconditional image generator to do MMUITs. The proposed method works directly on the latent space and enables the change of multiple semantic attributes at a time without affecting any content of the image that is not involved in the translation. Given a latent code in the latent space of pretrained GANs (e.g., $\boldsymbol{\mathcal{W}}^+$ in StyleGAN [56]), also called style code, $\boldsymbol{w}$, a semantic attribute label $\boldsymbol{d}$, and a noise vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, the ISF outputs a new style code $\boldsymbol{w}*$ that contains the desired semantic attributes. By randomly sampling multiple $\boldsymbol{z}$, several variants of the desired image translation (e.g., multiple hairstyles in gender manipulations) can be generated.

As a consequence, our proposed method can be used for multiple tasks, including Multi-domain Multi-modal Unsupervised Image-to-image Translation [25, 66] and smooth semantic interpolation [22]. When used in conjunction with GAN *inversion* techniques, which find the most similar style code given an input image (e.g., [2, 105]), our proposed method can also translate real images at high-resolution.

To enable better manipulations of StyleGAN's latent codes, we propose a simple extension of Adaptive Instance Normalization (AdaIN) [48], namely Adaptive Layer Normalization (AdaLN). AdaLN normalizes the latent code through Layer Normalization [9] instead of Instance Normalization [124] and goes beyond the assumption of independence between channels by computing the normalization for each sample across all the channels. Quantitative and qualitative results in face manipulations show

that our proposed method outperforms state-of-the-art methods.

Our contributions can be summarized as follows:

- We propose an implicit style function for editing multiple semantics of the input image at once by StyleGAN latent space manipulations. The images generated from the edited latent codes preserve the content/identity of input images while changing the user-specified semantics;

- We propose AdaLN, a simple extension of the popular normalization method AdaIN [48] to enable better arbitrary style transfer in the StyleGAN latent space;

- At the best of our knowledge, we are the first to address multi-domain and multi-modal unsupervised image-to-image translation by a *pre-trained* and *fixed* image generator, which allows having image translations at high resolution (i.e., 1024×1024).

5.2 Related Work

Unsupervised MMUIT models aim at learning a mapping between different visual domains unsupervisedly. They require that: 1) the domain-invariant part of the image, also called *content*, is preserved; 2) the model can map multiple domains; and 3) the output is multi-modal (i.e., diverse translations can be generated with the same input image).

To the best of our knowledge, there are no works, based on pre-trained and fixed unconditional GANs, that simultaneously meet all the requirements of unsupervised MMUIT models. However, we here review the literature of MMUIT models, StyleGAN-based latent space manipulations and StyleGAN-based inversion techniques.

Image-to-image Translation. Early attempts in Image-to-image Translation are based on paired images [50, 150, 115, 47] and one-to-one domain mappings [149, 49, 65, 101, 86, 131]. More recently, many studies have focused on training a single model that maps images into multiple domains and overcomes the deterministic (i.e., one-to-one) translation often assumed by previous works (e.g., [24, 72, 101]). We refer to these methods as MMUIT models. For example, DRIT++ [66] assumes the existence of a domain-independent ("content") and a domain-specific ("style") image representation and obtain this using two separate encoders. Then, DRIT++ allows multi-modal translations injecting random noise in the generation process. GMM-UNIT [75, 74] uses a Variational Auto-Encoder approach [60] where a style encoder maps the image into a Gaussian Mixture Model, from which it is possible to sample multiple different styles to be used in the generation process. StarGAN v2 [25, 78] proposes an architecture composed by a style encoder, which maps an existing image to a style code, and a mapping function, which starts from random noise and a domain code to generate a style code. In doing so, they achieve state-of-the-art performance in high-resolution (i.e., 256×256) and diversity of translated images.

The current state-of-the-art models in MMUIT field show results usually at 256×256 resolution and require training a generator from scratch. We here aim at enabling high-resolution MMUIT without training a generator.

StyleGAN Latent Space Manipulation. Most conditional GANs are trained at low and medium resolutions (i.e., up to 256×256 pixels). Scaling the image generators to higher resolutions to solve specific tasks requires a substantial computational budget. For this reason, scholars have recently started exploring and interpreting the latent code semantics of large unconditional networks for image synthesis such as PGGAN [55] and StyleGAN [56, 57]. The general idea is to adapt unconditional GANs to

solve specific tasks at high resolution without training task-specific generators [121]. For example, Härkönen *et al.* [43] identify important latent directions based on Principal Component Analysis (PCA) to control properties such as lighting, facial attributes, and landscape attributes. Shen *et al.* [112] use an off-the-shelf classifier to find the linear hyperplanes of semantic facial attributes. InterFaceGAN [111, 112] learns the latent semantics through linear hyper-planes, which rely on multiple SVMs that have to be specifically trained on each domain translation. Then, it edits face images moving the latent code with linear transformations. PSP [105] learns an encoder and a mapping function that enables multi-modal domain translations but does not support multi-domain translation. Chai *et al.* [19] propose to learn an *imperfect* masked encoder that finds the latent code that best resembles the input masked or coarsely-edited image. Then, it uses the fixed generator to synthesize photo-realistic results showing applications in image composition, image inpainting and multi-modal editing. Finally, StyleFlow [3] proposes to have attribute-controlled sampling and attributed-controlled editing through StyleGAN. They learn a function $\Phi(z, d)$ to sample StyleGAN latent codes conditioned to a semantic attribute vector, and they allow editing specific blocks of the StyleGAN latent space to edit pre-defined semantic categories.

However, existing latent space manipulation models based on StyleGAN are not suitable for MMUIT tasks. They are either limited on single-attribute at a time (e.g., [112, 111]) or one-domain translations (e.g., [105]). Moreover, they have very limited if no diversity in the manipulations (e.g., [112, 111, 105]) and, more importantly, manipulations along one attribute can easily result in unwanted changes along with other attributes (e.g., identity changes in hair color manipulations). We propose a simple and effective Implicit Style Function to manipulate the StyleGAN latent code and enable MMUIT at high resolution with a pre-trained and fixed

StyleGAN model.

GAN inversion in StyleGAN latent space. Solving the GAN inversion problem is essential to use pre-trained GANs with real images. GAN inversion techniques aim to find the generator's latent code that best corresponds to a given image.

Existing inversion approaches typically fall into two categories: encoder-based and optimization-based methods. The former [105] uses LBFGS [71] or similar optimizers to find the latent code $\boldsymbol{z}$ that best recovers the image $\boldsymbol{x}$ with $\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}}(\text{dist}(G(\boldsymbol{z}), \boldsymbol{x}))$ where dist is a metric distance function in the image space. The latter models (e.g., [1, 2, 42]) instead speeds-up the process at inference time by learning an encoder E. Then, the optimal $\boldsymbol{z}$ is simply the result of a feed-forward pass through E, more formally: $\boldsymbol{z}^* = E(x)$.

Fusion methods. A noteworthy application of Image-to-image Translation is image fusion, which aims to exploit images obtained by different sensors to generate better and more robust images. For example, FusionGAN [82] focuses on fusing the thermal radiation information in infrared images and the texture detail information in visible images to generate images through a GAN. Pan-GAN [81] instead focuses on fusing the thermal radiation information in infrared images and the texture detail information in visible images to generate images through a GAN. The latter focuses on remote sensing images fusing different layers of information to do pan-sharpening and better preserve the spectral and spatial information in images.

5.3 Method

An unconditional generator $G : \boldsymbol{\mathcal{Z}} \rightarrow \boldsymbol{\mathcal{X}}$ learns to synthesize an image $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ given a vector $\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}$, lying in a low-dimensional latent space. In

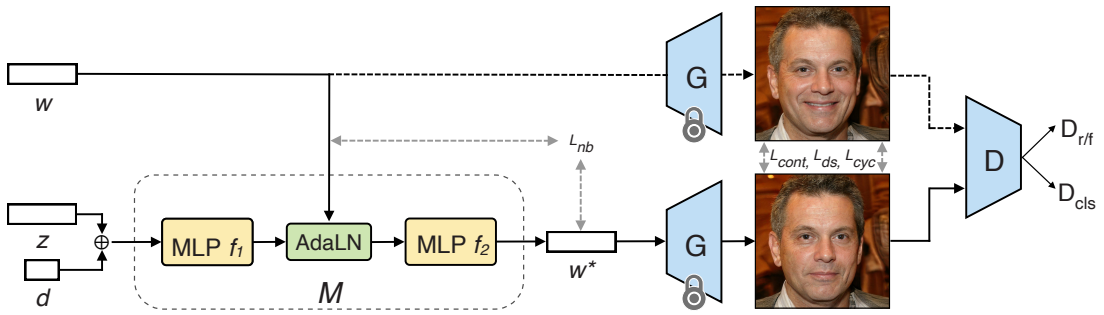


Figure 5.2: We train an Implicit Style Function $\mathcal{M}$ that manipulates a style code $\boldsymbol{w}$ into $\boldsymbol{w}^*$ given a randomly sampled noise $\boldsymbol{z}$ and a semantic attribute label $\boldsymbol{d}$. $\mathcal{M}$ is trained so that the image generated by G should have the semantics specified by $\boldsymbol{d}$ without changing anything along other attributes (e.g., face identity). G is a *pre-trained* and *fixed* uncodintional GANs (e.g., StyleGAN). We also train a discriminator D that discriminates between real/fake images and classifies the image attributes.

this work, we focus on StyleGAN v1 [56] and StyleGAN v2 [57], which first learn a non-linear mapping $f : \boldsymbol{\mathcal{Z}} \rightarrow \boldsymbol{\mathcal{W}}^+$ that outputs $\boldsymbol{w} = f(\boldsymbol{z}) \in \mathbb{R}^{18\times512}$, and then generates the image from $\boldsymbol{\mathcal{W}}^+$ space: $\boldsymbol{x} = G(\boldsymbol{w})$.

We aim at manipulating $\boldsymbol{w}$ such that it follows some user-specified semantic attributes $\boldsymbol{d} \in \boldsymbol{\mathcal{D}} \subset \mathbb{R}^m$ that refer to a m-dimensional semantic space, while best preserving the attribute-invariant content (e.g., face identity) of the source images. In other words, we want to learn a non-linear neural Implicit Style Function (ISF) $\mathcal{M} : \boldsymbol{\mathcal{W}}^+ \rightarrow \boldsymbol{\mathcal{W}}^+$ that outputs a manipulated code $\boldsymbol{w}^*$ in which:

1. *The output looks realistic.* The generated image $\boldsymbol{x}^* = G(\boldsymbol{w}^*)$ should look as realistic as the images generated from non-manipulated latent codes;

2. *Generated images exhibit the desired semantics.* The translation from $\boldsymbol{w}$ to $\boldsymbol{w}^*$ should be semantically correct;

3. *The source content is preserved.* A translated image $\boldsymbol{x}^* = G(\boldsymbol{w}^*)$ should maintain the attribute-invariant characteristics of the image

and change only the attribute-specific properties of the source image $\boldsymbol{x}$. For example, in smile $\leftrightarrow$ non-smile translations, $\boldsymbol{x}^*$ should maintain the pose and identity of the source image $\boldsymbol{x}$;

4. *The output is multi-modal.* Translations are inherently ambiguous, thus it is desirable to have a model that generates different, plausible translations. Note that we are not here interested in images whose appearance is almost identical but e.g. individual hairs are placed very differently (such in StyleGAN [56]);

5. *Training is multi-domain.* A single function $\mathcal{M}$ should be able to map each style code into multiple, different domains.

Simultaneously meeting these properties allows to overcome several limits of existing approaches and enable MMUITs using *pre-trained* and *frozen* unconditional GANs.

To fulfill the above properties, we propose a multilayer perceptron (MLP) as Implicit Style Function $\mathcal{M}$, which can be formulated as:

$$\boldsymbol{w}^* = \mathcal{M}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{d}), \tag{5.1}$$

where $\mathcal{M}(\cdot)$ predicts a target latent code $\boldsymbol{w}^*$ from a source latent code $\boldsymbol{w}$, a noise vector $\boldsymbol{z}$ randomly sampled from a standard Gaussian (i.e., $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$), and a target semantic vector $\boldsymbol{d}$. The source latent code $\boldsymbol{w}$ represents the latent code to be manipulated and $\boldsymbol{z}$ ensures the stochasticity of the manipulation process.

In order to learn such $\mathcal{M}$ in an unsupervised scenario, we assume to have a discriminator at training time which consists of two sub-tasks: (1) a binary classifier to distinguish the generated images from realistic images, and (2) a multi-label classifier $f : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{D}}$ to distinguish whether the images are with expected visual semantics. Thus, the $\mathcal{M}$ is learnt in an adversarial approach.

5.3.1 Learning the Implicit Style Function

We propose to train our ISF $\mathcal{M}$ in an adversarial fashion [41] with the help of a multi-task discriminator D, which is learned together with $\mathcal{M}$. At training time, we only optimize the parameters in the networks $\mathcal{M}$ and D, while we do not update the parameters of the pre-trained generator G. Such a strategy helps our method requiring fewer resources than traditional models that train a generator from scratch. The detailed architecture is depicted in Figure 5.2.

We use several objective functions to constraint the training process and learn the ISF $\mathcal{M}$.

Realism & Semantic Correctness. Inspired by recent literature (e.g., [24, 78]), we use a discriminator D with two branches namely D_{cls}, devoted to domain classification, and $D_{\text{r/f}}$, which learns a binary classification determining whether an image $\boldsymbol{x}$ is real or fake. More formally, we ensure the *realism* of the synthesized images through:

$$\begin{aligned}\mathcal{L}_{\text{r/f}} = \mathbb{E}_{\boldsymbol{w},\boldsymbol{z},\boldsymbol{d}}[\log D_{\text{r/f}}(G(\boldsymbol{w})) + \\ \log(1 - D_{\text{r/f}}(G(\mathcal{M}(\boldsymbol{w},\boldsymbol{z},\boldsymbol{d}))))].\end{aligned} \tag{5.2}$$

To ensure a manipulated code $\boldsymbol{w}^* = \mathcal{M}(\boldsymbol{w},\boldsymbol{z},\boldsymbol{d})$ generates an image with the specified semantics $\boldsymbol{d}$, we impose a domain classification loss when optimizing D and $\mathcal{M}$. More formally, D calculates the cross-entropy loss on the generated images over the non-manipulated latent codes by:

$$\mathcal{L}_{\text{cls}}^{D} = \mathbb{E}_{\boldsymbol{w},\boldsymbol{d}_0}\left[-\log D_{\text{cls}}(\boldsymbol{d}_0|G(\boldsymbol{w}))\right], \tag{5.3}$$

where $\boldsymbol{d}_0 = f(G(\boldsymbol{w}))$ refers to the attribute vector extracted from the image generated from $G(\boldsymbol{w})$ by utilizing the pre-trained attribute classifiers f. We will explain f in the Experiments Section. To learn classifying images, $\mathcal{M}$ tries to minimize the classification error of images generated

with manipulated latent codes with:

$$\mathcal{L}_{\text{cls}}^{\mathcal{M}} = \mathbb{E}_{\boldsymbol{s},\boldsymbol{z},\boldsymbol{d}} \left[-\log D_{\text{cls}}(\boldsymbol{d}|G(\mathcal{M}(\boldsymbol{s},\boldsymbol{z},\boldsymbol{d}))) \right]. \tag{5.4}$$

Content Preservation. We propose to preserve the *content* during the latent code manipulation with the following loss:

$$\mathcal{L}_{\text{cont}} = \mathbb{E}_{\boldsymbol{s},\boldsymbol{z},\boldsymbol{d}} \left[\psi(G(\boldsymbol{s}), G(\mathcal{M}(\boldsymbol{s},\boldsymbol{z},\boldsymbol{d}))) \right], \tag{5.5}$$

where $\psi(\boldsymbol{x}_1, \boldsymbol{x}_2)$ estimates the perceptual distance between image $\boldsymbol{x}_1$ and image $\boldsymbol{x}_2$ using an externally pre-trained network. Eq. (5.5) minimizes the perceptual distance between $G(\boldsymbol{w})$ and any image generated with a manipulated code (i.e., $G(\mathcal{M}(\boldsymbol{w},\boldsymbol{z},\boldsymbol{d}))$). Similarly to Liu *et al.* [78], Eq. (5.5) implies that some perceptual features should be maintained during the manipulation process. Although different perceptual distances can be used (e.g., the ℓ_2 distance on VGG features [53]), we implement $\psi(\boldsymbol{x}_1, \boldsymbol{x}_2)$ using the Learned Perceptual Image Patch Similarity (LPIPS) metric [145], which has been proved to be better aligned with the human perceptual similarity.

To further help the network to preserve the content of the input image, we also introduce a *neighbouring constraint*:

$$\mathcal{L}_{\text{nb}} = \mathbb{E}_{\boldsymbol{w},\boldsymbol{z},\boldsymbol{d}} \left[\|\boldsymbol{w} - \mathcal{M}(\boldsymbol{w},\boldsymbol{z},\boldsymbol{d})\|_2 \right]. \tag{5.6}$$

Eq. (5.6) is motivated by previous literature that has shown neighbouring latent codes in StyleGAN exhibit similar semantic properties [56, 57].

Finally, we also encourage the *cycle consistency* [149, 24, 25], which is typically used in most MMUIT models, to stabilize the training process. The *cycle consistency* loss encourages the image generated by the original latent code $\boldsymbol{w}$ and the image synthesized by the latent code mapped back into the original domain $\mathcal{M}(\boldsymbol{w}^*, \boldsymbol{z}, \boldsymbol{d}_0)$ to be as similar as possible. More formally, the cycle consistency loss is:

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{\boldsymbol{w},\boldsymbol{z},\boldsymbol{d},\boldsymbol{d}_0} \left[\|G(\boldsymbol{w}) - G(\mathcal{M}(\boldsymbol{w}^*, \boldsymbol{z}, \boldsymbol{d}_0))\|_1 \right]. \tag{5.7}$$

Multi-modality. We aim at having multi-modal outputs through the randomly sampled noise $\boldsymbol{z}$ injected in $\mathcal{M}$. However, to further encourage $\mathcal{M}$ to produce diverse outputs, we employ the *diversity sensitive* loss [25, 86]:

$$\mathcal{L}_{\text{ds}} = \mathbb{E}_{\boldsymbol{w},\boldsymbol{d},(\boldsymbol{z}_1,\boldsymbol{z}_2)\sim\mathcal{N}(\mathbf{0},\mathbf{I}_n)}[\quad \|G(\mathcal{M}(\boldsymbol{w},\boldsymbol{z}_1,\boldsymbol{d}) - G(\mathcal{M}(\boldsymbol{w},\boldsymbol{z}_2,\boldsymbol{d})\|_1], \tag{5.8}$$

where $\mathcal{M}$ is conditioned on two random noise vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. We maximize $\mathcal{L}_{\text{ds}}$ to have generated images that are as different as possible from each-other. To stabilize the learning process, we linearly decay the weight of the loss to zero during the training.

Overall Objective The full objective functions can be summarized as follow:

$$\begin{aligned}\min_{\mathcal{M}}\max_{D} =& \lambda_{\text{r/f}}\mathcal{L}_{\text{r/f}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{cont}}\mathcal{L}_{\text{cont}} + \lambda_{\text{nb}}\mathcal{L}_{nb} + \\ & \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} - \lambda_{\text{ds}}\mathcal{L}_{\text{ds}}\end{aligned} \tag{5.9}$$

where $\lambda_{\text{r/f}}$, λ_{cls}, λ_{cont}, λ_{nb}, λ_{cyc} and λ_{ds} are the hyper-parameters for each loss term. We note that the minus term before λ_{ds} allows to maximize Equation (5.8).

5.3.2 Injecting the domain and multi-modality

As shown in Figure 5.2, $\mathcal{M}$ is implemented through two MLP function f_1 and f_2 and an AdaLN layer between them, which we explain in this Section. The non-linear f_1 function maps the concatenation of a domain vector and a random noise to a hidden latent variable $\boldsymbol{h} = f_1(\boldsymbol{d} \oplus \boldsymbol{z})$. Then, $\boldsymbol{h}$ goes through AdaLN, which is based on the widely used Adaptive Instance Normalization (AdaIN) [48]. Several state of the art approaches in I2I translation use AdaIN to inject the style features into the generating process and thus transfer the desired style into the output image. AdaIN assumes that images with the same style have a common mean and variance. Thus, existing methods usually extract the channel-wise mean and

variance from images. However, previous literature has shown that the channels in the $\mathcal{W}^+$ latent codes of StyleGAN are correlated. For example, Karras *et al.* [56] has shown that bottom style layers in W (i.e., the first channels of $\mathcal{W}^+$) control high-level aspects such as pose, general hair style, face shape, and eyeglasses, middle-layers are about facial features, hair style, eyes open/closed, while higher layers are about color scheme and micro-structure.

Thus, we propose a simple modification to AdaIN by replacing Instance Normalization [124] with Layer Normalization:

$$\text{AdaLN}(\boldsymbol{w}, \boldsymbol{\gamma}, \boldsymbol{\beta}) = \boldsymbol{\gamma} \frac{\boldsymbol{w} - \mu(\boldsymbol{w})}{\sigma(\boldsymbol{w})} + \boldsymbol{\beta}, \tag{5.10}$$

where the $\mu(\cdot)$ and $\sigma(\cdot)$ are calculated by the Layer Normalization, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are parameters generated by a linear mapping function with the input $\boldsymbol{h}$. Layer Normalization [9] computes the normalization for each sample across all the channels, avoiding destroying valuable information in the latent code. The parameters in the AdaLN module are learnt during the training.

The output of AdaLN goes through a second MLP function f_2 to obtain the edited style code $\boldsymbol{w}^* = f_2(\text{AdaLN}(\boldsymbol{w}, \boldsymbol{\gamma}, \boldsymbol{\beta}))$.

5.4 Experiments

We focus the experiments of the proposed ISF-GAN on StyleGAN v1 [56] and StyleGAN v2 [57], the two most prominent state-of-the-art unconditional GANs. However, we note that our methodology can be applied to several GANs such as PG-GAN [55].

Baselines. We compare our proposal with two state-of-the-art methods employing pre-trained and fixed unconditional GAN to manipulate faces: InterFaceGAN [111] and StyleFlow [3]. Moreover, we compare the ISF-

GAN to StarGAN v2 [25] and its smooth extension [78] on gender translation, which are two state-of-the-art methods for MMUITs. We use the officially released code for all the baselines in all these comparisons.

Datasets. InterFaceGAN [111] and StyleFlow [3] were trained and tested on different conditions. To have a fair evaluation between the proposed method and the baselines, we collect the following datasets:

- Set_1: we randomly sample 90K latent codes and collect the corresponding images through StyleGAN v1 [56] pre-trained on FFHQ dataset [56]. Then, we use an *off-the-shelf* classifier [56] to label each latent code with the corresponding semantic attributes. We randomly split the dataset into 80K and 10K samples for the training and testing sets;

- Set_2: we collect the dataset released by StyleFlow [3], where 10K training images and 1K testing images are provided with annotated attributes and latent codes for StyleGAN v2 [57] pretrained on FFHQ dataset [56].

In each dataset, we model four key facial attributes commonly-used by InterFaceGAN [111] and StyleFlow [3] for analysis, including *gender*, *smile* (expression), *age*, and *eyeglasses*. We note that it is easily possible to plug in more semantic attributes as long as an attribute classifier is available.

Evaluation metrics. We evaluate both the visual quality and the diversity of generated images using Fréchet Inception Distance (FID) [46], the Learned Perceptual Image Patch Similarity (LPIPS) [145] and the Accuracy, evaluated through an *off-the-shelf* classifier provided by [56].

Moreover, we propose to use the state-of-the-art face recognition method ArcFace [27] to evaluate the content preservation in MMUITs. We define a new metric called Face Recognition Similarity (FRS) that estimates the similarity between features extracted from two facial images. More for-

mally:

$$\text{FRS} = \mathbb{E}_{\boldsymbol{s},\boldsymbol{z},\boldsymbol{d}}\left[\langle \omega(G(\boldsymbol{s})), \omega(G(\mathcal{M}(\boldsymbol{s},\boldsymbol{z},\boldsymbol{d})))\rangle\right] \tag{5.11}$$

where $\omega(\cdot)$ refers to the pre-trained ArcFace network, $\langle\cdot,\cdot\rangle$ refers to cosine similarity between two input vectors. The extracted face regions are resized to 112×112 before being fed into ω. An high FRS score shows the two faces have more similar identities, which indicates that a model better preserves the identities of faces during image manipulations.

Training Details. ISF-GAN is trained for 40K iterations with batch size 4. The training time takes about 16 hours on 4 Tesla V100 GPUs with our implementation in PyTorch [99]. Compared to the state-of-the-art MMUIT model StarGAN v2 [25], the training time is reduced by half while having a much higher resolution (i.e., from 256 × 256 to 1024 × 1024). We set $\lambda_{\text{r/f}} = 1$, $\lambda_{\text{cls}} = 1$ $\lambda_{\text{cont}} = 1$, $\lambda_{\text{nb}} = 0.1$, $\lambda_{\text{cyc}} = 1$ and $\lambda_{\text{ds}} = 2.0$. We adopt the non-saturating adversarial loss [41] with R1 regularization [91] using γ=1. We use the Adam [59] optimizer with β_1=0 and β_2=0.99. The learning rates for $\mathcal{M}$ and D are set to 10^{-5}.

5.4.1 Latent Codes Manipulation

We start by evaluating the performance of our framework on manipulating StyleGAN latent codes. Given a randomly sampled StyleGAN latent code $\boldsymbol{w}$ with its corresponding attribute vector $\boldsymbol{d}$, we choose a desire attribute vector $\boldsymbol{d}^*$ and use $\mathcal{M}$ to estimate a target $\boldsymbol{w}^*$. Intuitively, an ideal model should well manipulate the original latent code $\boldsymbol{w}$ to make the generated image $\boldsymbol{x}^* = G(\boldsymbol{w}^*)$ with correct semantics. Figure 5.3 suggests that the proposed ISF-GAN correctly learns the different semantics in StyleGAN v1 latent codes and allows to generate realistic translations. Compared to InterFaceGAN, our synthesized faces preserve the identity of the input image better, while correctly lying on the desired semantics. Similarly, as

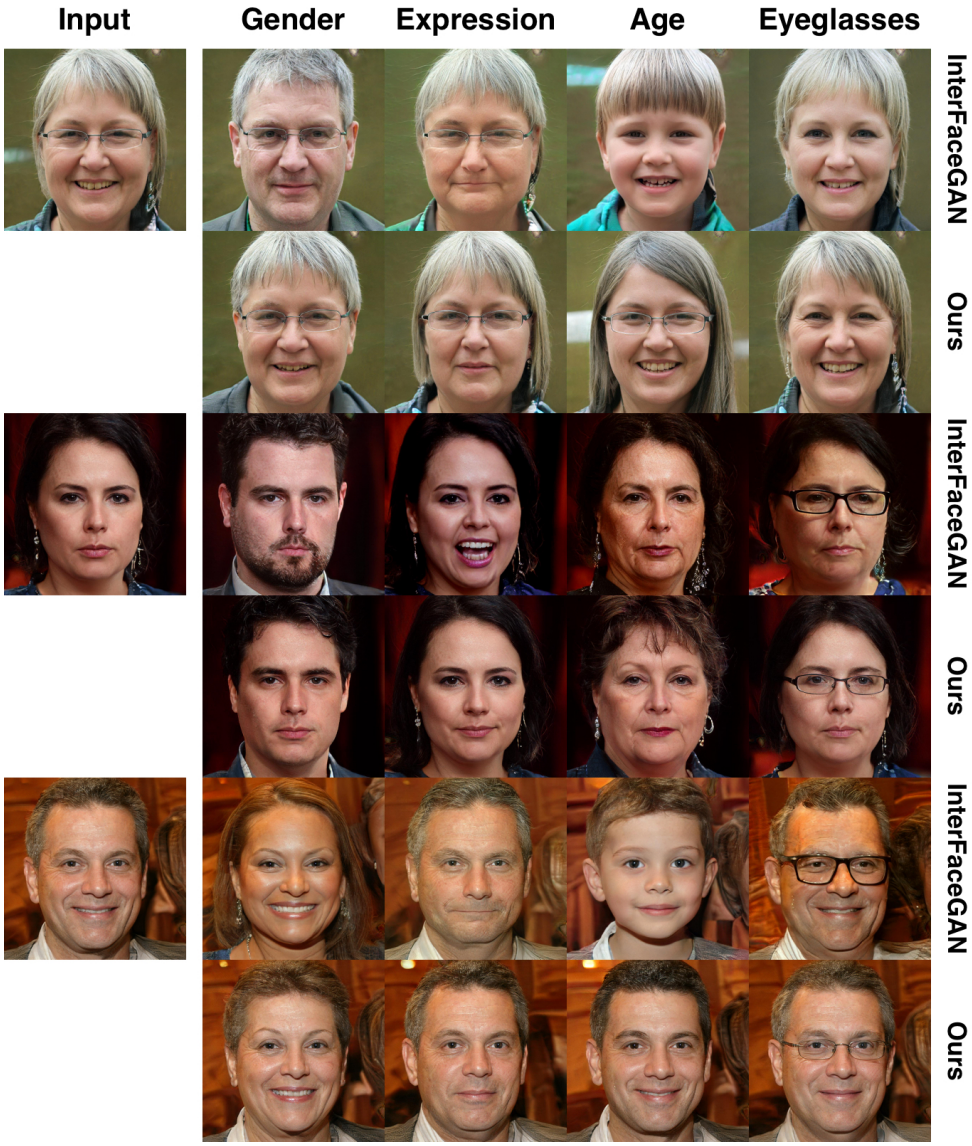


Figure 5.3: Visual comparisons between InterFaceGAN [111] and ISF-GAN on various attribute manipulations tested on the Set_1.

shown in Figure 5.4, the proposed method not only translate the input images with correct visual semantics but better preserves the face identity also in StyleGAN v2.

As shown in Table 5.1, the overall quantitative evaluation verifies the superiority of the proposed method. The proposed ISF-GAN achieves better performance on FID and LPIPS, which indicates ISF-GAN synthesizes

Figure 5.4: Visual comparisons between InterFaceGAN [111], StyleFlow [3] and our ISF-GAN on the Set_2.

more realistic images and enables the generation of diverse images. This observation is consistent with the visual comparisons presented in Figure 5.3 and Figure 5.4. Table 5.1 shows that InterFaceGAN scores 0.00 in the LPIPS as the model is deterministic and does not allow diverse translations for the same input code. The only way to have diversity is to generate style latent codes in very close range to the translated latent

code. This way scores 0.06 and 0.05 for Set_1 and Set_2, which is a result comparable to StyleFlow. However, changing the style codes in this way might affect the entire content of the generated image, including the identity of the person. Thus, it is not trivial to have multi-modal translations with InterFaceGAN.

We note that ISF-GAN can modify more than one attribute at a time, as shown in Figure 5.5 while InterFaceGAN [111, 112] relies on multiple learned hyper-planes and requires users to translate multiple attributes in multiple steps (e.g. female→male→smile). StyleFlow [3] can manipulate multiple attributes at a time, but achieves lower accuracy than our model on the domain label used as a target of the translation (see mAcc in Table 5.1).

We also note that we could not test StyleFlow on Set_1 as the authors did not release the model for StyleGAN v1. Thus, we could not have a fair comparison with them.

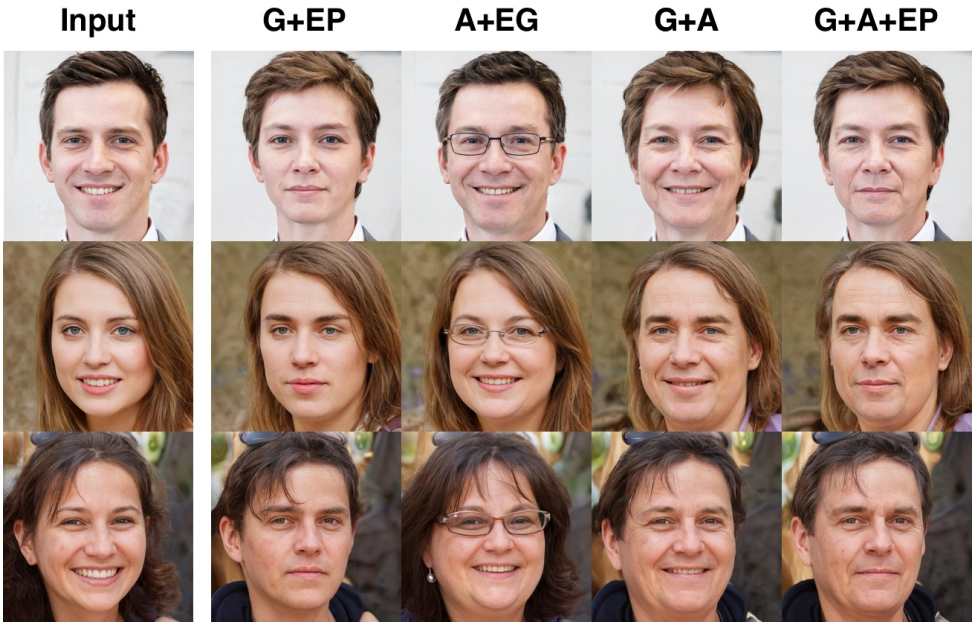


Figure 5.5: Visual results on multi-attributes manipulation at a time, where "G", "EP", "A" and "EG" refer to *gender*, *expression*, *age* and *eyeglasses* attributes, respectively.

Table 5.1: Quantitative comparisons on image quality, diversity, content preservation and accuracy of generated images based on pre-trained StyleGAN v1 [56]/v2[57]. The proposed ISF-GAN outperforms all state-of-the-art methods.

Model	Set_1				Set_2			
	FID↓	**LPIPS↑**	**FRS↑**	**mAcc↑**	**FID↓**	**LPIPS↑**	**FRS↑**	**mAcc↑**
StyleFlow [3]	-	-	-	-	50.42	0.04	0.62	72.23
InterFaceGAN [112]	50.89	0.00	0.32	94.17	29.85	0.00	**0.66**	78.95
Ours	**29.44**	**0.16**	**0.75**	**95.45**	**23.93**	**0.22**	**0.66**	**96.52**

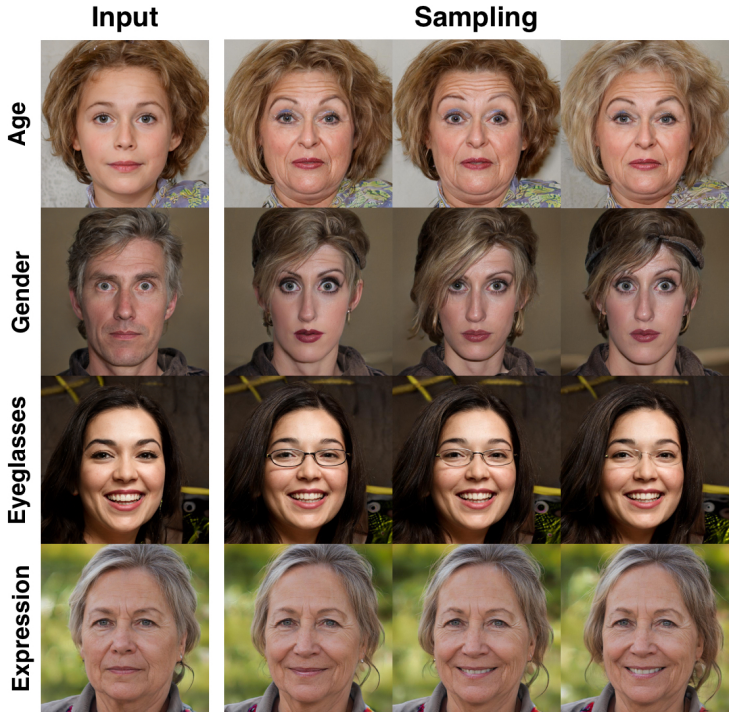


Figure 5.6: Multi-modal results obtained randomly sampling multiple $\boldsymbol{z}$ and collect diverse synthesized images.

Content preservation. Quantitatively evaluating content preservation is very challenging. However, in face translation, the content is mainly related to the identity of people. Thus, we here evaluate whether the identity is preserved during the manipulation tasks. Table 5.1 shows that FRS increases by 121% in InterFaceGAN with Set_1. In Set_2, improvements are only slight perhaps due to the smoother style latent space of StyleGANv2.

Figure 5.6 shows that the identity preservation is also stable when we use randomly sample different noise codes $\boldsymbol{z}$.

Latent space separation. The FID, LPIPS and FRS results do not come at the expense of the semantic correctness of generated images. We calculate the classification accuracy of the generated images by using off-the-shelf classifiers, in which the test set is composed of 1,000 positive samples and 1,000 negative samples for each manipulated attribute. In Table 5.1, our proposed method achieves better performance in both the Set_1 and Set_2. We also compare the semantic accuracy of our method with the baselines Table 5.1. Our proposed method achieves the best performances on almost all of the four attributes. It indicates that ISF-GAN correctly disentangles the different semantics contained in the StyleGAN latent code better than the start-of-the-art methods, thus allowing to synthesize images into the desired semantic specified by attribute vector $\boldsymbol{d}$.

Figure 5.7: Smooth inter-domain interpolations of our proposed ISF-GAN on various face attributes.

5.4.2 Semantic Interpolation

Evaluating semantic interpolations is important for two main reasons. On the one hand, semantic interpolation allows to generate a smooth flow

between two semantics (e.g., *males* and *females*), enabling users to generate intermediate images (e.g., non-binary gender or people transitioning gender) that are rarely present in existing datasets. On the other hand, interpolating latent codes allows qualitatively and quantitatively evaluating the smoothness of the latent space.

To semantically interpolate in the StyleGAN space, we first randomly sample a style $\boldsymbol{s}$, then we manipulate its semantic with $\boldsymbol{s}^* = \mathcal{M}(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{d})$, where $\boldsymbol{d}$ changes exactly one attribute from the attribute vector of $\boldsymbol{s}$. Then, we linearly interpolate T equi-distant points along the interpolation line between $\boldsymbol{s}$ and $\boldsymbol{s}^*$. Finally, we generate the images in the path: $[G(\boldsymbol{s}_0), G(\boldsymbol{s}_1), \ldots, G(\boldsymbol{s}_T))]$, where $\boldsymbol{s}_0 = \boldsymbol{s}$ and $\boldsymbol{s}_T = G(\boldsymbol{s}^*)$.

Figure 5.7 shows multiple semantic interpolations where we change the age, gender, expression and eyeglasses of some randomly sampled StyleGAN latent codes. We observe that the interpolations are very smooth, having almost imperceptible perceptual differences between neighbouring images. More visual comparisons with state-of-the-art are presented in Figure 5.8 and Figure 5.9.

Figure 5.8: Comparisons on gender interpolations between InterFaceGAN [111] and our proposed method tested on Set_1. Compared to InterFaceGAN, our method preserves better the face identity along the interpolations.

Figure 5.9: Comparisons on gender interpolations between InterFaceGAN [111], StyleFlow [3] and our proposed method tested on Set_2. Compared to InterFaceGAN and StyleFlow, our method preserves better the face identity along the interpolations.

To quantitatively evaluate the smoothness of interpolated images we rely on the Perceptual Path Length (PPL) [56] and a metric we propose: the Perceptual Interpolation Range (PIR). PPL measures the perceptual variation between pairs of images generated under small perturbations ϵ in the latent space:

$$PPL = \mathbb{E}_{\boldsymbol{s},\boldsymbol{s}^* \sim \mathcal{S}, t \sim U(0,1)} \left[\frac{1}{\epsilon^2} \Phi(\text{lerp}(\boldsymbol{s}, \boldsymbol{s}^*, t), \text{lerp}(\boldsymbol{s}, \boldsymbol{s}^*, t+\epsilon)) \right] \tag{5.12}$$

where $\Phi(\boldsymbol{s}_i, \boldsymbol{s}_j) = d(G(\boldsymbol{s}_i), G(\boldsymbol{s}_j))$, and $d(\cdot,\cdot)$ refers to the L2 distance between VGG [116] features.

PIR instead measures the semantic smoothness along the entire interpolation path between two latent codes. Specifically, PIR computes the range between the maximum and the minimum perceptual distance between two consecutive style codes along the interpolation path as $\max_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t) -$

$\min_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$. Very smooth interpolations between $\boldsymbol{s}_0$ and $\boldsymbol{s}_T$ are expected to have $\max_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t) \approx \min_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$, thus resulting in $PIR \approx 0$. Interpolations with abrupt changes would have $\max_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t) \gg \min_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)$, resulting in a high PIR. Since the perceptual distance along the interpolation path might be different depending on the sampled $\boldsymbol{s}_0$ and $\boldsymbol{s}_T$ we normalize the computed range by $\phi(\boldsymbol{s}_1, \boldsymbol{s}_T)$. Thus:

$$PIR = \mathbb{E}_{\boldsymbol{s}_1, \boldsymbol{s}_T \sim \mathcal{S}} \left[\frac{\max_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t) - \min_{t=1}^{T} \phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)}{\phi(\boldsymbol{s}_1, \boldsymbol{s}_T) + \epsilon} \right] \tag{5.13}$$

where ϵ is used for numerical stability. PIR uses the LPIPS [145] as the perceptual distance d, as LPIPS has been shown to be well aligned with human perceptual similarity.

Table 5.2 shows the quantitative results of the tested models. Both the PPL and PIR results show that the latent codes of our model are less entangled and that the perceptual path is substantially shorter than InterFaceGAN in Set_1. Similarly, although the PPL results are comparable between StyleFlow, InterFaceGAN and the proposed ISF-GAN in Set_2, PIR results suggest that there are less drastic changes between images generated along the interpolation path in ISF-GAN.

Table 5.2: Comparisons on the smoothness of semantic interpolations on *gender* translation.

Method	**Dataset**	**PPL↓**	**PIR↓**
InterFaceGAN [111]	Set_1	28.76	0.06
Ours		**10.23**	**0.01**
StyleFlow [3]	Set_2	63.06	0.11
InterFaceGAN [111]		**60.47**	**0.07**
Ours		62.06	**0.07**

Figure 5.10: Visual comparisons between the proposed ISF-GAN and StarGAN v2 [25] on gender translation.

5.4.3 Comparisons with Traditional MMUIT Models

We evaluate ISF-GAN with the state-of-the-art model of MMUITs, namely StarGAN v2 [25]. We test all the models on gender translation, following the main results of StarGAN v2.

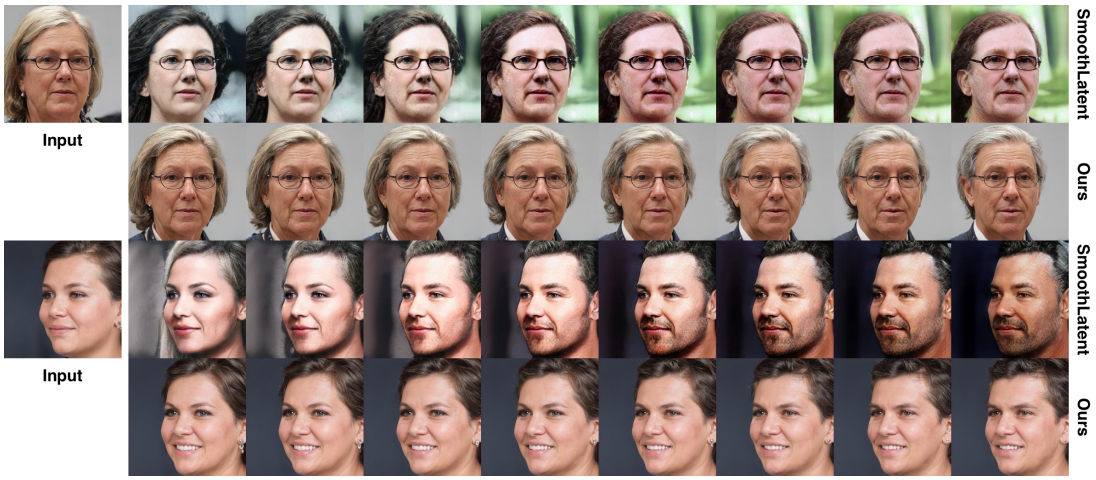


Figure 5.11: Visual comparisons between the proposed ISF-GAN and SmoothLatent [78] on gender interpolation. Although SmoothLatent can synthesize smooth interpolations, ISF-GAN generates more realistic images.

ISF-GAN achieves better performance compared to StarGAN v2 (20.67

vs 71.20 on FID score). Figure 5.10 shows that ISF-GAN not only generates more realistic images with correct semantics but also preserves better face identity and background. We also compare our model with SmoothLatent [78], which improves StarGAN v2 to have smooth inter-domain interpolations. ISF-GAN achieves better performance compared to SmoothLatent (0.050 vs 0.062 on PIR score). Figure 5.11 shows that the inter-domain interpolations are smoother in both visual semantics and background.

We note that training ISF-GAN requires fewer resources than training StarGAN v2 and its extension.

5.4.4 Manipulating Real Face Images

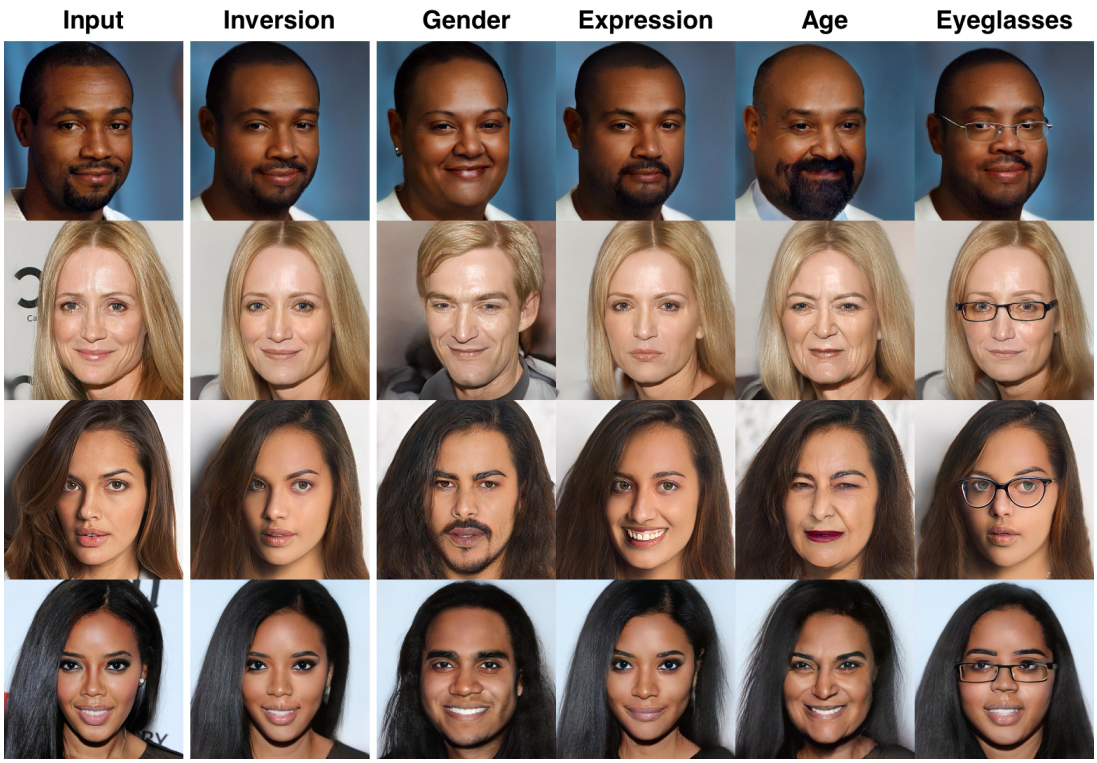


Figure 5.12: Manipulations of real images from CelebA-HQ [55] through ISF-GAN and an image embedding method [1].

Figure 5.12 shows an example of manipulation of real images. First,

we map the source images into the StyleGAN latent space through Abdal *et al.* [1], in which they search the latent code that best approximates the given image. Then, we use our model to translate images into multiple semantics. The results show that ISF-GAN can achieve high-quality results with such inverted latent codes of real images. Moreover, it indicates a promising application that ISF-GAN is used in conjunction with an *off-the-shelf* method to do GAN inversion in StyleGAN (either through an encoder or an optimization method) for unsupervised image-to-image translation.

5.4.5 Beyond Face Translations

To verify the generalization of the proposed method, we apply it to a two-domain image translation (e.g., Cat ↔ Dog) task. We collect a pre-trained StyleGAN v2 on the AFHQ dataset [25] (using only "cat" and "dog" images). The model is trained with images at 256×256 resolution, batch size 8 and 1e5 iterations on 4 Tesla V100 GPUs. Figure 5.13 shows several randomly sampled generated images based on the pre-trained StyleGAN v2.

Similar to the procedure in Section 5.4, we randomly sample 11K latent codes (10K for training and 1K for testing) in the $\mathcal{W}^+$ space and collect the corresponding images through the pre-trained GAN. Furthermore, in order to automatically distinguish the cat and dog faces, we also train a two-class classifier (i.e., a standard ResNet-50 [44]) on the AFHQ dataset. Finally, we train the proposed ISF-GAN on such pre-trained GAN and aim to do Cat↔Dog translation. As shown in Figure 5.14, ISF-GAN performs well on this task, which indicates the excellent generalization ability of the proposed method.

Figure 5.13: Random sampling in the latent space of pretrained StyleGAN v2 on Cat↔Dog dataset [25].

Figure 5.14: Visual results of ISF-GAN for Cat ↔ Dog translations based on a pre-trained StyleGAN v2.

5.4.6 Ablation Study

We here study the importance of each component of our proposed method. As shown in Table 5.3, the FID, LPIPS, FRS and PIR values for all the configurations where we removed or replaced components of our proposal Table 5.3-A. We repeated the experiments five times to compute the standard deviation of each metric.

First, we observe that FID and FRS significantly decrease by removing the neighbouring constraints $\mathcal{L}_{nb}$, but the diversity in the generated images (see Table 5.3-B) is evidently improved. This result suggests that constraining the latent codes to be similar in features to neighbouring latent codes helps disentangle the attributes from the attribute-invariant part of the image, thus improving the content preservation and quality of synthesized images.

Second, when we remove $\mathcal{L}_{cont}$, which encourages manipulated images to be perceptually similar to the original image, the LPIPS also improves but FID and FRS decrease. It shows a similar tendency as Table 5.3-B. However, $\mathcal{L}_{cont}$ has a more significant effect on the interpolation smoothness compared to $\mathcal{L}_{nb}$. This result suggests that perceptual loss is significant for better image quality and content preservation. From qualitative results, we see that $\mathcal{L}_{cont}$ helps the ISF-GAN better preserve some distinct face characteristics (e.g., the chin shape). The network without $\mathcal{L}_{cont}$ generates more diverse images because it also changes the identity of people.

We also ablate the proposed AdaLN by replacing it with AdaIN. Table 5.3-D shows that all metrics get worse. This suggests that StyleGAN latent codes do not have a channel-wise style distribution and that a channel-wise normalization destroys valuable information in it. Thus, the experiment verifies that the proposed AdaLN is more suitable for manipulating the latent codes of StyleGAN.

Table 5.3: Ablation study on our proposed losses and AdaLN on gender manipulations with Set_1.

Model	**FID↓**	**LPIPS↑**	**FRS↑**	**PIR↓**
A: Our proposal	**29.44**$_{\pm.08}$	.16$_{\pm.00}$	**.754**$_{\pm.005}$	**.009**$_{\pm.003}$
B: A w/o $\mathcal{L}_{nb}$	36.31$_{\pm.15}$	**.29**$_{\pm.00}$	.512$_{\pm.001}$	.029$_{\pm.009}$
C: A w/o $\mathcal{L}_{cont}$	31.20$_{\pm.13}$	.29$_{\pm.00}$	.646$_{\pm.003}$	.014$_{\pm.002}$
D: A w AdaIN	30.44$_{\pm.13}$	.13$_{\pm.00}$	.695$_{\pm.002}$	.012$_{\pm.003}$

Finally, we ablate the value of λ_{ds}, which controls how much the model should focus on diversity. As expected, Table 5.4 shows that the bigger is the value, the higher is the diversity of generated images. However, we also see that the changes on FID, FRS and PIR seem to be negatively correlated with the changes on LPIPS. This result is explained by the image quality and diversity trade-off previously discussed in [78]. It is easier for the model to collapse to images very similar to the target domain, thus obtaining a lower FID and LPIPS, than creating very diverse images that sometimes contain features (e.g., a lot of different hairstyles) that are not present in the source domain images.

In our paper, we chose to have a higher FID but having higher LPIPS. However, we note that our improvements in FID from the state of the art models are still substantial, as depicted in Table 5.1.

Table 5.4: Image quality - diversity trade-off on Set_2.

Model	**FID↓**	**LPIPS↑**	**FRS↑**	**PIR↓**
Our proposal w $\lambda_{ds} = 0.2$	22.60	0.05	0.76	0.05
Our proposal w $\lambda_{ds} = 1$	23.64	0.16	0.67	0.06
Our proposal w $\lambda_{ds} = 2$	23.93	0.22	0.66	0.07

5.5 Conclusion

In this paper, we propose an Implicit Style Function to manipulate the semantics of StyleGAN latent codes. We show that our approach enables multi-modal manipulations controlled by an attribute vector through qualitative and quantitative results. Moreover, these manipulations are smoother than the state-of-the-art approaches and better preserve the attribute-invariant visual parts of the original latent code. Altogether, our experimental results show that our model represents a competitive approach for applying *pre-trained* and *fixed* large-scale generators to Multi-modal and Multi-domain Unsupervised Image-to-image Translation tasks.

Chapter 6

Conclusion and Future Work

Unsupervised image-to-image translation (UNIT) aims at learning a mapping between several visual domains by using unpaired training images. Recent work has shown remarkable performance on various tasks (e.g., scene translation (segmentation map $\leftrightarrow$ street view image), attribute manipulation (gender translation for faces), style transfer (winter $\leftrightarrow$ summer)). In this thesis, we have explored four vital issues in the multi-modal and multi-domain unsupervised image-to-image translation (MMUIT) task.

In Chapter 2, we propose a method named GMM-UNIT to train only a single model to achieve multi-modal and multi-domain UNIT, which is based on a content-attribute disentangled representation where the attribute space is fitted with a GMM. Each GMM component represents a domain, and this simple assumption has two prominent advantages. First, it can be easily extended to most MMUIT tasks. Second, the continuous domain encoding allows for interpolation between domains and for extrapolation to unseen domains and translations.

In Chapter 3, we extend the GMM-UNIT to a challenging scenario to interpret the multi-media input (i.e., text and images). Here the model not only has to learn the manipulation without the ground truth of the desired output but also has to deal with the inherent ambiguity of nat-

ural language. To achieve these goals, we propose a novel unsupervised approach, based on image-to-image translation, that alters the attributes of a given image through a command-like sentence such as "change the hair color to black". Our proposed model disentangles the image content from the visual attributes, and it learns to modify the latter using the textual description, before generating a new image from the content and the modified attribute representation. Because text might be inherently ambiguous (blond hair may refer to different shadows of blond, e.g. golden, icy, sandy), our method generates multiple stochastic versions of the same translation.

In Chapter 4, we propose a new training protocol based on three specific losses which help a translation network to learn a smooth and disentangled latent style space in which: 1) Both intra- and inter-domain interpolations correspond to gradual changes in the generated images and 2) The content of the source image is better preserved during the translation. Moreover, we propose a novel evaluation metric to properly measure the smoothness of latent style space of I2I translation models. The proposed method can be plugged into existing translation approaches, and our extensive experiments on different datasets show that it can significantly boost the quality of the generated images and the graduality of the interpolations.

In Chapter 5, we propose an implicit style function (ISF) to straightforwardly achieve multi-modal and multi-domain image-to-image translation from pre-trained unconditional GANs. The ISF manipulates the semantics of an input latent code to make the image generated from it lie in the desired visual domain. Our results in the human face and animal manipulations show significantly improved results over the baselines. Our model enables cost-effective multi-modal unsupervised image-to-image translations at high resolution using pre-trained unconditional GANs.

Recently, Visual Transformers (VTs) have drawn booming attention in

the computer vision community. Particularly, they have been achieving significant improvements on image classification [30, 122, 79], object detection [18, 79], semantic segmentation [129, 79] and even some low-level vision tasks [70]. We have been also exploring this field in our present and future work:

- Efficient training VTs on small datasets without expensive pre-training. We empirically analyze different VTs, comparing their robustness in a small training set regime, and we show that, despite having a comparable accuracy when trained on ImageNet [108], their performance on smaller datasets can be largely different. Moreover, we propose an auxiliary self-supervised task which can extract additional information from images with only a negligible computational overhead. This task encourages the VTs to learn spatial relations within an image and makes the VT training much more robust when training data is scarce. Our task is used jointly with the standard (supervised) training and it does not depend on specific architectural choices, thus it can be easily plugged in the existing VTs. We have verified this direction in our latest publication [77].

- Interpreting the self-attention mechanism to have better content-aware self-attention heads. We propose a regularization method based on a spatial formulation of the information entropy. By minimizing the proposed spatial entropy, we explicitly ask the VT to produce "spatially ordered" attention maps, this way including an *object-based prior* during training. In extensive experiments, we show that the proposed regularization approach is beneficial with different training scenarios, datasets, downstream tasks and VT architectures.

- Exploring the super-resolution task with VTs as backbones to extract better features. Exploring the latent space of well-trained generative

models have achieved great success for face super-resolution (FSR). However, there is always a trade-off issue between the pixel-wise alignment and perceptual similarity in these methods. In the meanwhile, Reference-based Super-Resolution (Ref-SR) methods have shown another promising direction to enhance a low-resolution (LR) image by transferring the details from a high-resolution (HR) reference image. Although it alleviates the trade-off issue, it requires a manually selected reference image with similar content as the LR image. Instead of using GAN-inversion methods to generate the HR target images directly, we only generate neighboring reference images which are close to the target HR images in the latent space. With a coarse SR image recovered from a basic SR network and the neighboring high-quality synthesized references, our reference attention Transformer (RA-Transformer) can generate more realistic super-resolved images with better performance on both pixel-wise alignment and perceptual similarity.

Acknowledgement

I want to thank many important people who support and help me during my Ph.D. study. The thesis cannot be finished without them. My supervisors *Dr. Bruno Lepri* and *Prof. Nicu Sebe*, are perfect mentors I have ever met in my academic experience. They do not perform only as professional teachers, but also as close friends to me. They always care about both my research and life, and try their best to help me when I meet any difficulties. *Dr. Marco De Nadai* was my colleague and one of my best friends from the same research team, who worked with me during my whole Ph.D. study. I am always impressed by the means that he thinks about the research. During the thousands of discussions, he always discover valuable problems and taught me how to conduct deep research on them. Meanwhile, he always gives me very useful advice especially when I get stuck in the research. *Dr. Xavier Alameda-Pineda*, is a very close collaborator during the first two years of my Ph.D study. He is a master of math. I am very pleased to have worked with him during these years and I learned a lot from him on how to mathematically treat a research problem and how it could be formalized. *Prof. Enver Sangineto* and *Dr. Wei Wang* are very close collaborators during the last two years of my Ph.D. study. I'm impressed by them that they always think about the problem carefully and explain everything extremely clear. *Dr. Gianni Barlacchi* is a nice colleague from FBK, who helped me on the remote internship at Amazon Alexa AI, Berlin, Germany. *Dr. Wei Bi* and *Dr. Linchao Bao* are mentors during my internship at Tencent AI Lab, Shenzhen, China. They provided me with plenty of resources (e.g., computation resources and data) to explore large-scale industrial problems. Meanwhile, they provide much contrastive feedback during my research projects. All the beloved mobsers (e.g., Simone Centellegher, Lorenzo Lucchini, Gabriele Santin,

Pierfrancesco Ardino, Massimiliano Luca and Antonio Longa), mhuggers (e.g., Prof. Elisa Ricci, Hao Tang, Guanglei Yang, Jichao Zhang, Yue Song, Zhun Zhong, Subhankar Roy, Aliaksandr Siarohin, Elia Peruzzo and Weijie Wang), and friends from other research groups (e.g., Deng Cai, Huayang Li, Yan Wang, Yajing Chen) really bring me a lot of happiness. They are so friendly and helpful. I truly appreciate all the people mentioned above and wish everyone all the best. Finally, I would like to thank my parents and my girlfriend (Keqing Tan) for their support without reservation during my academic pursuit. I wish them all the happiness and good health.

Bibliography

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

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020.

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv preprint arXiv:2008.02401*, 2020.

[4] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan. *arXiv preprint arXiv:2103.14968*, 2021.

[5] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. *NeurIPS*, 2018.

[6] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *ICML*, 2018.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[8] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *ICLR*, 2018.

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[10] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. *arXiv preprint arXiv:2006.06500*, 2020.

[11] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016.

[12] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 38(4):59, 2019.

[13] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

[14] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146, 2017.

[15] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018.

[16] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[17] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2018.

[18] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[19] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. In *ICLR*, 2021.

[20] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018.

[21] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[22] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *CVPR*, 2019.

[23] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention gan for interactive image editing via dialogue. In *AAAI*, 2018.

[24] 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. In *CVPR*, 2018.

[25] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

[26] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, 2020.

[27] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[28] Chris Donahue, Akshay Balsubramani, Julian McAuley, and Zachary C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. In *ICLR*, 2018.

[29] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014.

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[31] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Keep drawing it: Iterative language-based image generation and editing. In *NeurIPS*, 2018.

[32] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *ICCV*, 2019.

[33] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[34] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2014.

[35] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

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

[37] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael J. Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. In *ICLR*, 2020.

[38] Corrado Gini. Variabilità e mutabilità (variability and mutability). *Memorie di metodologica statistica*, 1912.

[39] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019.

[40] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *NeurIPS*, 2018.

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[42] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020.

[43] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[45] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017.

[46] 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. In *NeurIPS*, 2017.

[47] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

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

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

[50] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[51] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2020.

[52] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *ICCV*, 2019.

[53] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[54] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *ArXiV*, 2017.

[55] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[56] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[57] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

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

[59] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[60] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[61] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[62] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[63] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[64] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NeurIPS*, 2018.

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

[66] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision (IJCV)*, 2020.

[67] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Controllable text-to-image generation. *arXiv preprint arXiv:1909.07083*, 2019.

[68] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. *arXiv preprint arXiv:1912.06203*, 2019.

[69] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *CVPR*, 2019.

[70] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV Workshops*, 2021.

[71] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[72] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017.

[73] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019.

[74] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe what to change: A text-guided unsupervised image-to-image translation approach. In *ACM MM*, 2020.

[75] Yahui Liu, Marco De Nadai, Jian Yao, Nicu Sebe, Bruno Lepri, and Xavier Alameda-Pineda. Gmm-unit: Unsupervised multi-domain and multi-modal image-to-image translation via attribute gaussian mixture modeling. *arXiv preprint arXiv:2003.06788*, 2020.

[76] Yahui Liu, Marco De Nadai, Gloria Zen, Nicu Sebe, and Bruno Lepri. Gesture-to-gesture translation in the wild via category-independent conditional maps. *ACM MM*, 2019.

[77] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets. In *NeurIPS*, 2021.

[78] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *CVPR*, 2021.

[79] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[80] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[81] Jiayi Ma, Wei Yu, Chen Chen, Pengwei Liang, Xiaojie Guo, and Junjun Jiang. Pan-gan: An unsupervised pan-sharpening method for remote sensing image fusion. *Information Fusion*, 62:110–120, 2020.

[82] Jiayi Ma, Wei Yu, Pengwei Liang, Chang Li, and Junjun Jiang. Fusiongan: A generative adversarial network for infrared and visible image fusion. *Information Fusion*, 48:11–26, 2019.

[83] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*, 2019.

[84] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[85] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019.

[86] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019.

[87] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[88] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[89] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, 2018.

[90] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, 2020.

[91] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018.

[92] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[93] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instance-aware image-to-image translation. In *ICLR*, 2019.

[94] Aditya Mogadala, Marimuthu Kalimuthu, and Dietrich Klakow. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *arXiv preprint arXiv:1907.09358*, 2019.

[95] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018.

[96] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *NeurIPS*, 2018.

[97] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[98] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020.

[99] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshops*, 2017.

[100] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[101] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 2018.

[102] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *CVPR*, 2019.

[103] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[104] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *NeurIPS*, 2016.

[105] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021.

[106] Andrés Romero, Pablo Arbeláez, Luc Van Gool, and Radu Timofte. Smit: Stochastic multi-label image-to-image translation. In *ICCV Workshops*, 2019.

[107] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *CVPR*, 2019.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[109] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[110] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[111] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[112] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[113] Aliaksandr Siarohin, Stéphane Lathuilière, Enver Sangineto, and Nicu Sebe. Appearance and Pose-Conditioned Human Image Generation using Deformable GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[114] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *NeurIPS*, 2019.

[115] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable GANs for pose-based human image generation. In *CVPR*, 2018.

[116] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[117] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

[118] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[119] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017.

[120] Joshua B Tenenbaum and William T Freeman. Separating style and content. In *NeurIPS*, 1997.

[121] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021.

[122] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[123] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018.

[124] Dmitry Ulyanov and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

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

[126] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Technical Report*, 2011.

[127] Hai Wang, Jason D Williams, and SingBing Kang. Learning to globally edit images with textual description. *arXiv preprint arXiv:1810.05786*, 2018.

[128] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[129] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

[130] Yaxing Wang, Joost van de Weijer, and Luis Herranz. Mix and match networks: encoder-decoder alignment for zero-pair image translation. In *CVPR*, 2018.

[131] Yaxing Wang, Lu Yu, and Joost van de Weijer. Deepi2i: Enabling deep hierarchical image-to-image translation by transferring from gans. In *NeurIPS*, 2020.

[132] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *ICCV*, 2019.

[133] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *CVPR*, 2019.

[134] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021.

[135] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[136] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[137] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *CVPR*, 2018.

[138] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[139] Guanglei Yang, Haifeng Xia, Mingli Ding, and Zhengming Ding. Bidirectional generation for unsupervised domain adaptation. In *AAAI*, 2020.

[140] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[141] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[142] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[143] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *CVPR*, 2017.

[144] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *TPAMI*, 41(8):1947–1962, 2018.

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

[146] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.

[147] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, 2019.

[148] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020.

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

[150] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017.

[151] Changqing Zou, Haoran Mo, Chengying Gao, Ruofei Du, and Hongbo Fu. Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG)*, 38:16, 2019.