Transferring and Learning Representations for Image Generation and Translation

Yaxing Wang

Universitat Autònoma
de Barcelona

Transferring and Learning Representations for Image Generation and Translation

A dissertation submitted by **Yaxing Wang** at Universitat Autònoma de Barcelonato fulfil the degree of **Doctor of Philosophy**.

Bellaterra, January 10, 2020

Director	**Dr. Joost van de Weijer** Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB) **Dr. Luis Herranz** Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB) **Dr. Abel González García** Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB)
Thesis committee	**Dr. Adria Ruiz** Institut national de recherche en informatique et en automatique (INRIA) **Dr. Antonio López Peña** Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB) **Dr. Antonio Agudo Martinez** Institut de Robòtica i Informàtica Industrial Universitat Politècnica de Catalunya (UPC)

This document was typeset by the author using LaTeX 2$_\varepsilon$.

ISBN: 978-84-121011-5-7

Printed by Ediciones Gráficas Rey, S.L.

Acknowledgements

Joost said, 'You never know what happens the next time,' when I suffered from the failure of my research project in the second year of the Ph.D. I think his expression is not valid here; it is very clear to me what to write in this acknowledgement.

Nevertheless, I would like to thank my supervisor Dr. Joost van de Weijer, for his guidance, inspiration, enthusiasm, patience and support throughout my years at CVC and UAB. I think of Joost as the optimizer and I am a network trained from scratch. He aims to obtain global coverage. First, he recommends various resources (scientific articles, blogs, courses, etc.) and researchers as training data. Second, during each updating, he tries to search the optimal solution and avoids me to suffer from local coverage. Third, his supervision is always patient, encouraging and constructive, preventing me from mode crashing due to gradient explosion or vanishing gradients. Perhaps this example sounds a bit complicated, but I really want to express my deep appreciation to Joost van de Weijer.

For my co-supervisor, Dr. Luis Herranz my most sincere thanks, too. He always is responsible, helpful and patient. His constructive suggestions and responsible attitude considerably helped me to improve my research. He always is focused no matter how important the problem. He is able to give advice or recommend relevant papers when I get stuck in my projects. What is important, he can speak Chinese and explain hard questions with me in Chinese since my English is poor.

I also want to deeply thank my co-supervisor, Dr. Abel González García, who is responsible, constructive and encouraging. In my heart, he is patient, smart and knowledgeable whenever I explain my topic and trouble to him with my hard-to-understand English. He always is positive and encouraging when we conduct our project. In my heart, he is not only my supervisor but a good friend who is willing to listen to my trouble of my work and life, and give helpful advice to me.

I have had the privilege of conducting and discussing my research with many LAMP members, including Bogdan, Andy, Mikhail, David and Bartlomiej. My sincere thanks to Marc, Xialei, Lu, Lichao, Laura, Aymen, Oguz, Fei, Chenshen, Kai, Shun, Minghan, Shiqi, Mikel, Carola, Aitor, Javad, and Albin. Many thanks to all the CVC people: Yi, Yixiong, Dena, Cori, Zhijie, Lei, Xavier, Pau, Bojana, Sounak, Gemma, Arash, Montse, Gigi and all the people in administration.... Thanks to my dearest friends in Barcelona, Bin, Yi, Xiao GuanGuan, Liu, Dailu, Lang, Jinqiang,

Rong, Yu, Tingting, Fangchang, Junpeng, Qiaoming, Jie, and Yu, we have spent much good times together. Thanks to my ZUMBA teachers: Monica, Naty, and Montse.

Over the summers, I have had the fortune to work as an intern in the Inception Institute of Artificial Intelligence (IIAI) in Abu Dhabi. I would like to thank my supervisors , Dr. Fahad Shahbaz Khan and Dr. Salman Khan. You were active and helpful. You always provided valuable guidance and discussions to my project. Although you were busy, you tried your best to help me. I was extremely happy to study with you. I also thank my friends from IIAI, including Haoliang, Tianyang, Jinpeng, Aditya Arora and Akshita Gupta. With your help, I had a wonderful life in Abu Dhabi.

Finally, I am grateful to my parents and my wife Xiaoqing for their love and support during this wonderful journey. To my love, Xiaoqing, who always supports and encourages me. I am lucky we can enjoy many happy moment together and find support in each other to overcome difficulties. The unbroken bonds between us made me the person who I am today.

Abstract

Image generation is arguably one of the most attractive, compelling, and challenging tasks in computer vision. Among the methods which perform image generation, generative adversarial networks (GANs) play a key role. The most common image generation models based on GANs can be divided into two main approaches. The first one, called simply image generation takes random noise as an input and synthesizes an image which follows the same distribution as the images in the training set. The second class, which is called image-to-image translation, aims to map an image from a source domain to one that is indistinguishable from those in the target domain. Image-to-image translation methods can further be divided into paired and unpaired image-to-image translation based on whether they require paired data or not. In this thesis, we aim to address some challenges of both image generation and image-to-image generation.

GANs highly rely upon having access to vast quantities of data, and fail to generate realistic images from random noise when applied to domains with few images. To address this problem, we aim to transfer knowledge from a model trained on a large dataset (source domain) to the one learned on limited data (target domain). We find that both GANs and conditional GANs can benefit from models trained on large datasets. Our experiments show that transferring the discriminator is more important than the generator. Using both the generator and discriminator results in the best performance. We found, however, that this method suffers from overfitting, since we update all parameters to adapt to the target data. We propose a novel architecture, which is tailored to address knowledge transfer to very small target domains. Our approach effectively explores which part of the latent space is more related to the target domain. Additionally, the proposed method is able to transfer knowledge from multiple pretrained GANs.

Although image-to-image translation has achieved outstanding performance, it still faces several problems. First, for translation between complex domains (such as translations between different modalities) image-to-image translation methods require paired data. We show that when only some of the pairwise translations have been seen (i.e. during training), we can infer the remaining unseen translations (where training pairs are not available). We propose a new approach where we align multiple encoders and decoders in such a way that the desired translation can be obtained by simply cascading the source encoder and the target decoder, even when they have not interacted during the training stage (i.e. unseen). Second, we address the issue of bias in image-to-image translation. Biased datasets unavoid-

ably contain undesired changes, which are due to the fact that the target dataset has a particular underlying visual distribution. We use carefully designed semantic constraints to reduce the effects of the bias. The semantic constraint aims to enforce the preservation of desired image properties. Finally, current approaches fail to generate diverse outputs or perform scalable image transfer in a single model. To alleviate this problem, we propose a scalable and diverse image-to-image translation. We employ random noise to control the diversity. The scalabitlity is determined by conditioning the domain label.

Key words: *computer vision, deep learning, imitation learning, adversarial generative networks, image generation, image-to-image translation.*

Resumen

La generación de imágenes es una de las tareas más atractivas, fascinantes y complejas en la visión por computador. De los diferentes métodos para la generación de imágenes, las redes generativas adversarias (o también llamadas "GANs") juegan un papel crucial. Los modelos generativos más comunes basados en GANs se pueden dividir en dos apartados. El primero, simplemente llamado generativo, utiliza como entrada ruido aleatorio y sintetiza una imagen que sigue la misma distribución que las imágenes de entrenamiento. En el segundo apartado encontramos la traducción de imagen a imagen, cuyo objetivo consiste en transferir la imagen de un dominio origen a uno que es indistinguible del dominio objetivo. Los métodos de esta categoria de traducción de imagen a imagen se pueden subdividir en emparejados o no emparejados, dependiendo de si se requiere que los datos sean emparejados o no. En esta tesis, el objetivo consiste en resolver algunos de los retos tanto en la generación de imágenes como en la traducción de imagen a imagen.

Las GANs dependen en gran parte del acceso a gran cantidad de datos, y fallan al generar imágenes realistas a partir de ruido aleatorio cuando se aplican a dominios con pocas imágenes. Para solucionar este problema, proponemos transferir el conocimiento de un modelo entrenado a partir de un conjunto de datos con muchas imágenes (dominio origen) a uno entrenado con datos limitados (dominio objetivo). Encontramos que tanto las GANs como las GANs condicionales pueden beneficiarse de los modelos entrenados con grandes conjuntos de datos. Nuestros experimentos muestran que transferir el discriminador es más importante que hacerlo para el generador. Usar tanto el generador como el discriminador resulta en un mayor rendimiento. Sin embargo, este método sufre de overfitting, dado que actualizamos todos los parámetros para adaptar el modelo a los datos del objetivo. Para ello proponemos una arquitectura nueva, hecha a medida para resolver la transferencia de conocimiento en el caso de dominios objetivo con muy pocas imágenes. Nuestro método explora eficientemente qué parte del espacio latente está más relacionado con el dominio objetivo. Adicionalmente, el método propuesto es capaz de transferir el conocimiento a partir de múltiples GANs pre-entrenadas.

Aunque la traducción de imagen a imagen ha conseguido rendimientos extraordinarios, tiene que enfrentarse a diferentes problemas. Primero, para el caso de la traducción entre dominios complejos (cuyas traducciones son entre diferentes modalidades) se ha observado que los métodos de traducción de imagen a imagen requieren datos emparejados. Demostramos que únicamente cuando algunas de las traducciones disponen de esta información (i.e. durante el entrenamien-

to), podemos inferir las traducciones restantes (cuyos pares no están disponibles). Proponemos un método nuevo en el cual alineamos diferentes codificadores y decodificadores de imagen de una manera que nos permite obtener la traducción simplemente encadenando el codificador de origen con el decodificador objetivo, aún cuando estos no han interactuado durante la fase de entrenamiento (i.e. sin disponer de dicha información). Segundo, existe el problema del sesgo en la traducción de imagen a imagen. Los conjuntos de datos sesgados inevitablemente contienen cambios no deseados, eso se debe a que el dataset objetivo tiene una distribución visual subyacente. Proponemos el uso de restricciones semánticas cuidadosamente diseñadas para reducir los efectos del sesgo. El uso de la restricción semántica implica la preservación de las propiedades de imagen deseada. Finalmente, los métodos actuales fallan en generar resultados diversos o en realizar transferencia de conocimiento escalables a un único modelo. Para aliviar este problema, proponemos una manera escalable y diversa para la traducción de imagen a imagen. Para ello utilizamos ruido aleatorio para el control de la diversidad. La escalabilidad es determinada a partir del condicionamiento de la etiqueta del dominio.

Palabras clave: *visión por computador, aprendizaje profundo, aprendizaje por imitación, redes generativas adversarias, generación de imágenes, traducción de imagen a imagen*

Resum

La generació d'imatges és una de les tasques més atractives, fascinants i complexes de la visió per computador. Dels diferents mètodes per la generació d'imatges, les xarxes generatives adversaries (o també anomenades "GANs") juguen un paper crucial. Els mètodes generatius més comuns basats en GANs es poden dividir en dos apartats. El primer, simplement anomenat generatiu, utilitza soroll aleatori i sintetitza una imatge per tal de seguir la mateixa distribució que les imatges d'entrenament. En el segon apartat trobem la traducció d'imatge a imatge, on el seu objectiu consiteix en transferir la imatge d'un domini origen a un que és indistingible d'un domini objectiu. Els mètodes d'aquesta categoria de traducció d'imatge a imatge es poden subdividir en emparellats o no emparellats, depenent de si requereixen que les dades siguin emparellades o no. En aquesta tesi, l'objectiu consisteix en resoldre alguns dels reptes tant en la generació d'imatges com en la traducció d'imatge a imatge.

Les GANs depenen en gran part de l'accés a una gran quantitat de dades, i fallen al generar imatges realistes a partir del soroll aleatori quan s'apliquen a dominis amb poques imatges. Per solucionar aquest problema, la solució proposada consisteix en transferir el coneixement d'un model entrenat a partir d'un conjunt de dades amb moltes imatges (domini origen) a un entrenat amb dades limitades (domini objectiu). Hem trobat que tant les GANs com les GANs condicionals poden beneficiar-se dels models entrenats amb grans conjunts de dades. Els nostres experiments mostren que transferir el discriminador és més important que fer-ho per el cas del generador. Utilitzar tant el generador com el discriminador resulta en un millor rendiment. No obstant, aquest mètode sufreix d'overfitting, donat que actualitzem tots els paràmetres per adaptar el mètode a les dades de l'objectiu. Proposem una arquitectura nova, feta a mesura per tal de resoldre la transferència de coneixement per el cas de dominis objectius amb molt poques imatges. El nostre mètode explora eficientment quina part de l'espai latent està més relacionat amb el domini objectiu. Adicionalment, el mètode proposat és capaç de transferir el coneixement a partir de múltiples GANs pre-entrenades.

Tot i que la traducció de imatge a imatge ha conseguit rendiments extraordinaris, ha d'enfrentarse a diferents problemes. Primer, per el cas de la traducció entre dominis complexes (on les traduccions són entre diferents modalitats) s'ha vist que els mètodes de traducció de imatge a imatge requereixen dades emparellades. Demostrem que únicament quan algunes de les traduccions disposen de la informació (i.e. durant l'entrenament), podem inferir les traduccions restants (on les

parelles no estan disponibles). Proposem un mètode nou en el cual alineem diferents codificadors y decodificadors d'imatge d'una manera que ens permet obtenir la traducció simplement encadenant el codificador d'origen amb el decodificador objectiu, encara que aquests no hagin interactuat durant la fase d'entrenament (i.e. sense disposar d'aquesta informació). Segon, existeix el esbiaixament en la traducció de imatge a imatge. Els datasets esbiaixats inevitablement contenen canvis no desitjats, això es deu a que el dataset objectiu té una distribució visual subjacent. Proposem l'ús de restriccions semàntiques curosament dissenyades per reduir els efectes de l'esbiaixament. L'ús de la restricció semàntica implica la preservació de les propietats de les imatges desitjades. Finalment, els mètodes actuals fallen en generar resultats diversos o en realitzar transferència de coneixement escalable a un únic model. Per aliviar aquest problema, proposem una manera escalable i diversa per a la traducció de imatge a imatge. Utilitzem el soroll aleatori per el control de la diversitat. La escalabilitat és determinada a partir del condicionament de la etiqueta del domini.

Paraules clau: *visió per computador, aprenentatge profund, aprenentatge per imitació, xarxes generatives adversaries, generació d'imatges, traducció d'imatge a imatge*

Contents

1 Introduction

Images play a crucial role in our life, as we take photos or videos, watch movies or sport matches, conduct video chat and play video games. Besides, the companies which provide social media services connecting people face huge amounts of image data on a daily basis. For example, more than 95 million photos and videos are uploaded to Instagram every day, and photo uploads total over 300 million per day, according to Mary Meeker's annual Internet Trends report [2]. That is 657 billion photos per year.

The desire to automatically extract information from these large amounts of image data results in the research field of Computer Vision, since the raw image data contains limited information. Researchers proposed different methods to understand images and videos. For example, image classification and semantic segmentation assign labels to images and pixels of images respectively; an object detection model can identify which of a known set of objects might be present and provide information about their positions within the image; video tracking is the process of locating a moving object (or multiple objects) over time in a video.

In recent years, researchers have made significant advances in the field of image generation. These methods require the machine to synthesize visual objects which are indistinguishable from real objects. The generation could be according to a text description [192], a list of attributes [31, 117, 130] or based on an input image. In case of an input image these algorithms are known as image-to-image translation methods. The field of image generation is important for many applications, including transformations between different modalities (e.g., from RGB to depth [99]), between domains (e.g., gray to color images [196], horses to zebras [206]) or editing operations (e.g., artistic style transfer [49]). In this dissertation, we focus on image generation.

Several different approaches have been developed for image generation. Some of these maximize the log likelihood, like the Boltzmann machines [148]. These contain likelihood functions and employ numerous approximations to the likelihood gradient. Generative stochastic networks [14] provide a model that is trained with backpropagation [91]. Kingma et al. [83] proposed variational autoencoders

Figure 1.1 – A GAN is composed of two players who play a minimax game [54].

(VAEs), allowing one to backpropagate through Gaussian distributions with finite variance. Generative Adversarial Networks (GANs) [54] play a minimax game with two players that aim to find a saddle point, which is a minimum with respect to one player's strategy and a maximum with respect to the otherplayer's strategy (see figure 1.1). In this dissertation we focus on GANs to conduct image generation.

1.1 Basics of Image Generation and Image Translation

In this thesis, we propose several improvements for GANs and image-to-image translation methods. We first shortly introduce the basics of both. The thesis is divided in two parts, where the first part focuses on improvements to image generation with GANs, and the second part focuses on image-to-image methods.

1.1.1 Image Generation with Generative Adversarial Networks

As shown in Figure 1.2, a GAN is a framework consisting of a deep generative model and a discriminative model, both of which play a minimax game. The generator takes random noise as an input, and synthesizes the fake sample using several network layers (e.g. fully connected layers and convolutional layers). The aim of the generator is to generate samples that are similar to the real samples. The discriminator, which also contains several layers, takes both fake and real samples as input, and aims to distinguish between the real and fake sample generated by

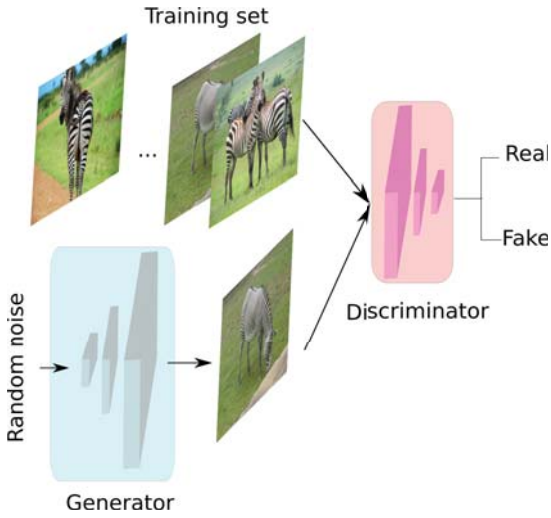


Figure 1.2 – Framework of a GAN [54]. A GAN contains of a generator and a discriminator. The generator takes random noise as input and outputs a fake image. The discriminator takes both a fake and real image as input, and aims to distinguish between them.

the generator.

After its initial introduction many works have proposed improvements. Optimizing GANs is hard, since they often face mode collapse and unstable training problems. Several methods focus on fixing these issues [10, 56, 110, 116, 149]. Besides, current methods design new architectures to synthesize high resolution images [19, 36, 76, 77, 132]. For example, DCGAN [132] proposed a new architecture which resulted in more stable training. Their generator architecture was based on blocks which consist of a fully connected layer, convolutional layer, batch normalization layer and Relu layer. Their discriminator architecture used blocks consisting of a convolutional, a batch normalization, a relu and fully connected layer. The architecture was further improved by Progressive GAN [76] which generates high-quality images by means of synthesizing images progressively from low to high-resolution. Finally, BigGAN [19] successfully performs conditional high-realistic generation from ImageNet [35] by introducing orthogonal regularization to the generator.

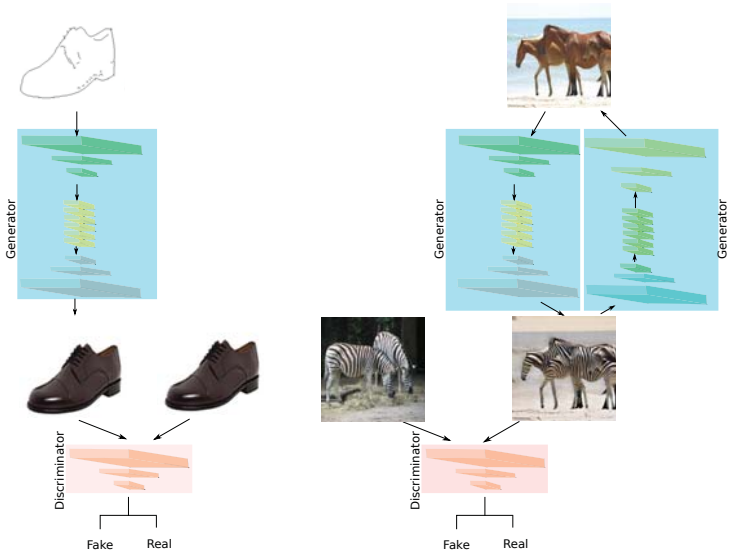


Figure 1.3 – Paired (left) and unpaired (right) image-to-image translation framework. The paired image-to-image translation maps the input image from source domain to target domain, and requires the corresponding ground truth in target domain. The unpaired image-to-image translation relaxes this limitation, and preforms image translation without the need of paired data.

1.1.2 Image-to-Image Translation with Generative Adversarial Networks

The goal of image-to-image translation is to learn a mapping between images of a source domain and images of a target domain. In the left of figure 1.3 we show the basic steps of image-to-image translation. In this example the model aims to map an edge image to a photo-realistic shoe image. Specifically, the generator is composed of both an encoder and a decoder. The encoder takes the edge image as input, and maps it into a latent representation. Next, the decoder maps the latent representation into the generated image (also called fake image). The discriminator takes both fake and real images as input, and aims to distinguish them.

The initial work on image-to-image translation was based on the assumption of paired image data [70]. Paired image data refers to the fact that we have access to pairs of data, which represent the same instance in both domains. For many applications, however, we do not have access to paired image data. For example, when we would like to transfer horses to zebras, we do not have images where

both animals are in the exact same pose and surroundings. In the left of figure 1.3, we presented the standard framework of paired image-to-image translation. This model relies on paired data, and is able to be optimized by both mean square error (MSE) and adversarial loss. The right of figure 1.3 shows the framework of unpaired image-to-image translation, which maps the input sample from source domain into target domain without requirement of paired data. This model uses the cycle consistency loss [206] as well as adversarial loss. The consistency loss ensures that when the system maps the input image from the source domain into the target domain and back, we obtain the input image again.

1.2 Challenges of transferring and learning representations for image generation and translation

1.2.1 Limitations of image generation methods

In this section, we highlight several shortcomings of the state-of-the-art for image generation with GANs. This is the motivation for Part I of the thesis.

Part of the impressive results which have been recently obtained by GANs is due to the fact that they are based on very large neural networks, trained on large datasets. For instance, Spectral Normalization GAN (SNGAN) with projection discriminator [117] for 128×128 pixel images has 90M trainable parameters of both the generator and the discriminator. BigGAN [19] trained on ImageNet [35] for 128×128 pixel images needs about 300M parameters. Progressive Growing GANs [76] (PGAN) requires about 487M parameters when it synthesizes high resolution images (1024×1024 pixels). These models benefit from the large model to improve performance of image generation. Besides, in order to optimize such large network and generate highly realistic images, a large dataset is required to train generative models [19, 76, 77, 116, 117]. For example, PGAN generates human faces and is trained on the Celeba-HQ dataset [105], which contains 30M celebrity faces. BigGAN [19] obtains highly realistic results based on the Imagenet dataset, which has a total of about 1M. Besides, they also train on JFT-300M dataset. However, labeling large-scale datasets is costly and time consuming, and a conventional generative model cannot be applied to a domain in which collecting sufficient data is difficult. These problems make those methods less applicable in practice. Therefore, exploiting pretrained generative models for training on small domains is an urgent research problem.

Transfer learning plays a key role in computer vision. It allows us to learn a large and deep model with limited data. For example, a common practice to improve performance is to use a feature extractor model initialized with one trained on a

large dataset (source model) [101, 102, 125, 186, 187, 195]. Then the model is trained with a smaller learning rate with the small target dataset. This operation is called fine-tuning [125]. Even when the source dataset is from a completely different domain, the final performance after fine tuning is normally still better than if the model was trained from scratch with only the small target dataset. This is due to the fact that the representations learned on the large dataset are generally usefull for visual tasks, and therefore give a good initialization from which to start training on a new dataset, even if only few samples are available for the target domain.

In this thesis, we focus on several specific challenges and limitations regarding transfer learning for generative models.

Transfer learning by fine-tuning generative models While discriminative models for tasks with small datasets are customary initialized with a pre-train model (e.g. ImageNet), GANs are almost always trained from scratch. To the best of our knowledge, training GANs with limited data using a pre-trained model from some large source datates (e.g. Imagenet, Places [203]) had never been studied. As we discussed above, the current GAN models require a large amount of parameters to generate realistic images, which makes training very challenging. This is especially severe when training data is limited, resulting in unrealistic generated images and severe overfitting.

Efficient sampling and training. Finetuning of pretrained GANs might not be optimal. Actually some more recent works observe several problems [51, 122, 173]. They note that this method suffers from overfitting and mode collapse, since all parameters of the models are updated with limited data. Some recent works aim to address this problem by only updating a part of the network. Giacomello *et al.* [51] observed that the two networks (generator and discriminator) in a GAN are trained towards opposing objectives, requiring back-propagating information through both networks, which is computationally expensive. They develop a method to directly train the extreme layers in the generator and discriminator against each other, by-passing all the intermediate layers. Noguchi and Harada [122] address the limitations by only updating the batch normalization parameters (scale and shift) of the generator. Although less susceptible to mode collapse, this method suffers from another problem that largely limits the capacity of transferring knowledge, since updating only the parameters of the batch normalization allows to change low level texture and style changes but fails to learn more structural changes, such as shape changes. Therefore, exploring strategies which only require few parameters to be learned to transfer knowledge to a new domain are expected to improve over a simple fine-tuning strategy (which allows all parameters in the network to change).

Multi-source transfer learning. Current methods only consider transferring knowledge from one pre-trained generative model instead of multiple models.

However, for many applications it might be beneficial to transfer knowledge from multiple domains. Both Imagenet [35] and Places [203] contain different objects, and the pre-trained models acquired from both datasets include different information. When we train generative models on a dataset that has some overlap with both datasets, transferring the information from both pre-trained models is expected to be beneficial.

1.2.2 Limitations of image-to-image translation methods

In this section, we highlight several shortcomings of the state-of-the-art for image-to-image translation methods. This is the motivation for Part II of the thesis.

Recently, GANs have shown remarkable performance on a wide variety of image-to-image translation tasks [68, 70, 131, 181, 206], super-resolution [92], image compression [138], and conditional image generation such as text to image [191, 192], tracking [194], segmentation to image [76, 167]. However, there are still some major challenges which hamper the use of these methods for some real-world applications.

Transferring translations and inferring unseen translations. Image-to-image translation usually distinguishes between two main directions: paired and unpaired image-to-image translation. However, for many real-world cases we have access to multiple domains some of which are paired and some of which are unpaired. Consider the case where we would like to perform image-to-image translation between multiple modalities but only have access to paired data for some modalities. This results in the research question: can we exploit the paired modalities to learn the translation model between the unpaired modalities? As an example consider the three domains (RGB, semantic segmentation, and depth) as a particular case. Here, we might want to map depth to semantic segmentation, with the constraint that we do not have explicit depth-segmentation pairs during training time. We only have access to RGB-sementation and RGB-depth pairs, i.e. RGB images and the corresponding semantic segmentation, and RGB images and the corresponding depth images. In this case, we need to explore the principle that using knowledge acquired between paired images makes it possible to map between unpaired images (depth image and semantic segmentation). We call this *unseen* image-to-image translation, and we name this setting *zero-pair translation*.

Biases in image-to-image translation. Data-driven models tend to replicate the biases underlying the dataset, and image-to-image translation is no exception. Bias in image-to-image translation often manifests as unwanted changes of some visual properties. For example, if the image-to-image translation task is mapping faces without make-up (source domain) to faces with make-up (target domain), we expect

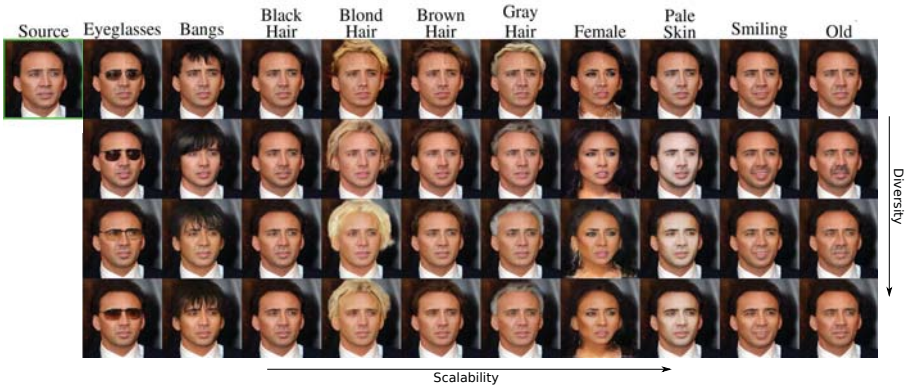


Figure 1.4 – Example of scalable and diverse image translations for various attributes [140].

the model to only focus on mapping make-up instead of other unwanted properties (e.g. gender). The training samples, however, contain males and females in the domain without make-up, but there is a strong bias towards females in make-up images. The translation not only adds make-up to the generated face (wanted property), but it also changes the gender (the unwanted properties). The internal biases in the input and output training sets determine what particular visual and semantic properties of the input image are changed. With such biases, the translator learns to generate female faces with make-up even when the input is a male face. While the change in the make-up attribute is desired, the change in gender is not. Therefore, addressing biases in image-to-image translation is a relevant research topic.

Scalable and Diverse Cross-domain Image Translation. Recent research trends address two limitations of earlier image-to-image translation approaches, namely diversity and scalability. Diversity is the property that a single input image can be mapped into multiple plausible outputs in a specific target domain. Scalability is the property that a single model can project an input image into varying target domains. In figure 1.4, we show an example of diverse and scalable image translations. Current methods [68, 131] improve over the single-sample limitation of deterministic models by generating diverse translations given an input image. The scalability problem has also been successfully alleviated [31, 130], enabling translations across several domains using a single model. However, current methods fail to perform image-to-image translation with scalability and diversity using a simple and compact network.

1.3 Objectives and approach

Above we have indicated several limitations of computer vision in image generation. In this dissertation, we propose methods to address these limitations. In Part 1 we aim to transfer knowledge to efficiently train generative models on small dataset. Then we introduce the proposed methods to tackle several problems for image-to-image translation in Part 2.

1.3.1 Objectives and approach for image generation

The above discussion motivates our approach to transferring knowledge for generative models. We firstly explore fine-tuning the generative model, from which we learn that using pre-trained models for GANs is effective to generate more realistic images. Then, we propose a new method to overcome the issue that directly performing fine-tuning leads to overfitting.

Pretrained generative models for domains with limited data Using the pre-trained networks to initialize the target domain has been widely accepted for visual problems, such as image classification, image detection, semantic segmentation and so on. However, knowledge transfer has not been studied within the context of generative deep networks. Besides, current generative models, which generate high-quality images, benefit from large-scale models and datasets. When given small target dataset, GANs fail to synthesize realistic images. Therefore, we study knowledge transfer for GANs.

In chapter 2, we propose the usage of pre-trained networks to transfer knowledge in generative models. We present the following contributions: (1) we explicitly explore several different conditions to conduct fine-tuning, and clearly show that using pre-trained models largely reduce the requirement of training time and obtain improvements when the target domain lacks data. (2) We also evaluate knowledge transfer from unconditional GANs to conditional GANs.

MineGAN: effective knowledge transfer from GANs to target domains with few images. As we discussed above, the finetuning method for transferring knowledge for generative model suffers from several drawbacks. This is mainly caused because all parameters are allowed to change, which can lead to overfitting in case the target domain is small. Besides, current methods are not able to use information of multiple pre-trained models. In this thesis, we not only explore how to transfer effectively knowledge for generative model, but also propose a novel method to use knowledge from multiple pre-trained models.

In chapter 3, we introduce the process of *mining* of GANs, that is performed by a *mining network*, that transfers a multivariate normal distribution into a distribution

on the input space of the pretrained GAN in such a way that the generated images resemble those of a target domain. We consider a variety of different relations between the source data, for which we have pretrained models, and the target data. Including the case where for the target domain only part of the knowledge of the source domain is advantageous, and the case where the knowledge from multiple domains should be combined for optimal transfer. We will also consider the case of transferring knowledge from conditional GANs to unconditional GANs.

1.3.2 Objectives and approach for image-to-image translation

Mix and match networks: encoder-decoder alignment for zero-pair image translation. As discussed in the previous section, we introduce zero-pair image translation, which is a new setting for testing image translations that involves evaluating on *unseen* translations, i.e. translations for which no paired data is available during training. In order to tackle this problem, in Chapter 4 we propose *Mix and Match Networks (M&MNets)*, an approach that addresses zero-pair image translation by seeking alignment between encoders and decoders via their latent spaces. An unseen translation between two domains is performed by simply concatenating the input domain encoder and the output domain decoder. We study several techniques that can improve this alignment, including the usage of autoencoders, latent space consistency losses and pooling indices as side information to guide the reconstruction of spatial structure. We evaluate this approach in a challenging cross-modal task, where we perform zero-pair depth to semantic segmentation translation, using only RGB to depth and RGB to semantic segmentation pairs during training.

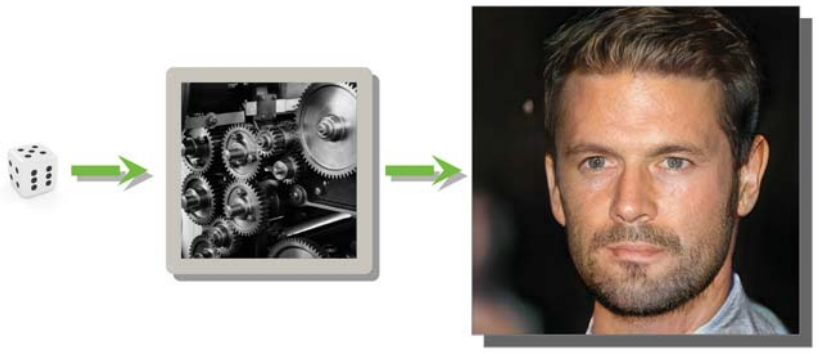
Finally, we show that aligned encoder-decoder networks also have advantages in domains with unpaired data. In this case, we show that mix and match networks scale better with the number of domains, since they are not required to learn all pairwise image translation networks.

Controlling biases and diversity in diverse image-to-image translation. We describe the problem of unbiased image-to-image translation by introducing the concept of wanted and unwanted changes. To address the problem of biases in image-to-image translation we propose an approach that uses *semantic constraints* to counter undesired biases. We apply these constraints using additional pretrained neural networks to extract relevant semantic features. Modeling an adequate semantic constraint is often not trivial, since naive implementations may result in irrelevant information. This fails to fix undesired side effects, it is ineffective for bias compensation, and reduces the ability of image translation (e.g limiting the diversity in the output). In Chapter 5 we present an efficient approach to model

an effective semantic constraint that obtains two benefits: alleviating bias while preserving the desired properties. Meanwhile, our method still is able to preserve diversity.

Scalable and Diverse cross-domain Image Translation Current methods are not able to perform diverse and scalable image translations in a single model. They either conduct diverse translation [68, 93, 131] or perform scalable translation [31, 130]. In Chapter 6, we address this issue. We propose a compact and general architecture model that achieves the desired goal: diversity and scalability in a single model. The Conditional Instance Normalization (CIN) [68] introduces two conditional new factors (the shift and scale parameters) to the normalization layer. We employ CIN layers to obtain diverse outputs. Besides, we explore the reasons behind its success, and find that the shift parameter plays a more important role than the scale parameter. Scalability is enabled by using the domain labels as inputs to the encoder.

Image generation Part I



How do we use transfer learning for image generation with limited data [76]

2 Transferring GANs: generating images from limited data[1]

2.1 Introduction

Generative Adversarial Networks (GANs) can generate samples from complex image distributions [54]. They consist of two networks: a discriminator which aims to separate real images from fake (or generated) images, and a generator which is simultaneously optimized to generate images which are classified as real by the discriminator. The theory was later extended to the case of conditional GANs where the generative process is constrained using a conditioning prior [115] which is provided as an additional input. GANs have further been widely applied in applications, including super-resolution [92], 3D object generation and reconstruction [154], human pose estimation [107], and age estimation [198].

Deep neural networks have obtained excellent results for discriminative classification problems for which large datasets exist; for example on the ImageNet dataset which consists of over 1M images [84]. However, for many problems the amount of labeled data is not sufficient to train the millions of parameters typically present in these networks. Fortunately, it was found that the knowledge contained in a network trained on a large dataset (such as ImageNet) can easily be transferred to other computer vision tasks. Either by using these networks as off-the-shelf feature extractors [11], or by adapting them to a new domain by a process called fine tuning [125]. In the latter case, the pre-trained network is used to initialize the weights for a new task (effectively transferring the knowledge learned from the source domain), which are then fine tuned with the training images from the new domain. It has been shown that much fewer images were required to train networks which were initialized with a pre-trained network.

GANs are in general trained from scratch. The procedure of using a pre-trained network for initialization – which is very popular for discriminative networks – is to the best of our knowledge not used for GANs. However, like in the case of discriminative networks, the number of parameters in a GAN is vast; for example the

[1] This chapter is based on a publication in the European Conference on Computer Vision (ECCV 2018) [173].

popular DC-GAN architecture [132] requires 36M parameters to generate an image of 64x64. Especially in the case of domains which lack many training images, the usage of pre-trained GANs could significantly improve the quality of the generated images.

Therefore, in this work, we set out to evaluate the usage of pre-trained networks for GANs. The chapter has the following contributions:

1. We evaluate several transfer configurations, and show that pre-trained networks can effectively accelerate the learning process and provide useful prior knowledge when data is limited.

2. We study how the relation between source and target domains impacts the results, and discuss the problem of choosing a suitable pre-trained model, which seems more difficult than in the case of discriminative tasks.

3. We evaluate the transfer from unconditional GANs to conditional GANs for two commonly used methods to condition GANs.

2.2 Related Work

Transfer learning/domain transfer: Learning how to transfer knowledge from a source domain to target domain is a well studied problem in computer vision [126]. In the deep learning era, complex knowledge is extracted during the training stage on large datasets [146, 204]. Domain adaptation by means of fine tuning a pre-trained network has become the default approach for many applications with limited training data or slow convergence [37, 125].

Several works have investigated transferring knowledge to unsupervised or sparsely labeled domains. Tzeng et al. [164] optimized for domain invariance, while transferring task information that is present in the correlation between the classes of the source domain. Ganin et al. [48] proposed to learn domain invariant features by means of a gradient reversal layer. A network simultaneously trained on these invariant features can be transfered to the target domain. Finally, domain transfer has also been studied for networks that learn metrics [65]. In contrast to these methods, we do not focus on transferring discriminative features, but transferring knowledge for image generation.

GAN: Goodfellow et al. [54] introduced the first GAN model for image generation. Their architecture uses a series of fully connected layers and thus is limited to simple datasets. When approaching the generation of real images of higher complexity, convolutional architectures have shown to be a more suitable option. Shortly afterwards, Deep Convolutional GANs (DC-GAN) quickly became the standard GAN

architecture for image generation problems [132]. In DC-GAN, the generator sequentially up-samples the input features by using fractionally-strided convolutions, whereas the discriminator uses normal convolutions to classify the input images. Recent multi-scale architectures [36, 67, 76] can effectively generate high resolution images. It was also found that ensembles can be used to improve the quality of the generated distribution [174].

Independently of the type of architecture used, GANs present multiple challenges regarding their training, such as convergence properties, stability issues, or mode collapse. Arjovksy et al. [9] showed that the original GAN loss [54] are unable to properly deal with ill-suited distributions such as those with disjoint supports, often found during GAN training. Addressing these limitations the Wassertein GAN [10] uses the Wasserstein distance as a robust loss, yet requiring the generator to be 1-Lipschitz. This constrain is originally enforced by clipping the weights. Alternatively, an even more stable solution is adding a gradient penalty term to the loss (known as WGAN-GP) [56].

cGAN: Conditional GANs (cGANs) [115] are a class of GANs that use a particular attribute as a prior to build conditional generative models. Examples of conditions are class labels [55, 124, 130], text [134, 191], another image (image translation [81, 206] and style transfer [39]).

Most cGAN models [38, 115, 157, 191] apply their condition in both generator and discriminator by concatenating it to the input of the layers, i.e. the noise vector for the first layer or the learned features for the internal layers. Instead, in [39], they include the conditioning in the batch normalization layer. The AC-GAN framework [124] extends the discriminator with an auxiliary decoder to reconstruct class-conditional information. Similarly, InfoGAN [25] reconstructs a subset of the latent variables from which the samples were generated. Miyato et al. [117] propose another modification of the discriminator based on a 'projection layer' that uses the inner product between the conditional information and the intermediate output to compute its loss.

2.3 Generative Adversarial Networks

2.3.1 Loss functions

A GAN consists of a generator G and a discriminator D [54]. The aim is to train a generator G which generates samples that are indistinguishable from the real data distribution. The discriminator is optimized to distinguish samples from the real data distribution p_{data} from those of the fake (generated) data distribution p_g. The generator takes noise $z \sim p_z$ as input, and generates samples $G(z)$ with a distribu-

tion p_g. The networks are trained with an adversarial objective. The generator is optimized to generate samples which would be classified by the discriminator as belonging to the real data distribution. The minimax game objective is given by:

$$G^* = \arg\min_G \max_D \mathscr{L}_{GAN}(G, D) \tag{2.1}$$

$$\mathscr{L}_{GAN}(G, D) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{2.2}$$

In the case of WGAN-GP [56] the two loss functions are:

$$\begin{aligned} \mathscr{L}_{WGAN-GP}(D) &= -\mathbb{E}_{x \sim p_{data}}[D(x)] + \mathbb{E}_{z \sim p_z}[D(G(z))] \\ &+ \lambda \mathbb{E}_{x \sim p_{data}, z \sim p_z, \alpha \sim (0,1)} \left[(\|\nabla D(\alpha x + (1-\alpha) G(z))\|_2 - 1)^2 \right] \end{aligned} \tag{2.3}$$

$$\mathscr{L}_{WGAN-GP}(G) = -\mathbb{E}_{z \sim p_z}[D(G(z))] \tag{2.4}$$

2.3.2 Evaluation Metrics

Evaluating GANs is notoriously difficult [160] and there is no clear agreed reference metric yet. In general, a good metric should measure the quality and the diversity in the generated data. Likelihood has been shown to not correlate well with these requirements [160]. Better correlation with human perception has been found in the widely used Inception Score [149], but recent works have also shown its limitations [205]. In our experiments we use two recent metrics that show better correlation in recent studies [15, 69]. While not perfect, we believe they are satisfactory enough to help us to compare the models in our experiments.

Fréchet Inception Distance [61] The similarity between two sets is measured as their Fréchet distance (also known as Wasserstein-2 distance) in an embedded space. The embedding is computed using a fixed convolutional network (an Inception model) up to a specific layer. The embedded data is assumed to follow a multivariate normal distribution, which is estimated by computing their mean and covariance. In particular, the FID is computed as

$$\mathrm{FID}(\mathscr{X}_1, \mathscr{X}_2) = \left\| \mu_1 - \mu_2 \right\|_2^2 + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2(\Sigma_1 \Sigma_2)^{\frac{1}{2}} \right) \tag{2.5}$$

Typically, $\mathscr{X}_1$ is the full dataset with real images, while $\mathscr{X}_2$ is a set of generated samples. We use FID as our primary metric, since it is efficient to compute and correlates well with human perception [61].

Independent Wasserstein (IW) critic [33] This metric uses an independent critic $\hat{D}$ only for evaluation. This independent critic will approximate the Wasserstein

Table 2.1 – FID/IW (the lower the better / the higher the better) for different transfer configurations. ImageNet was used as source dataset and LSUN Bedrooms as target (100K images).

Generator	Scratch		Pre-trained	
Discriminator	Scratch	Pre-trained	Scratch	Pre-trained
FID $\left(\mathscr{X}^{tgt}_{data}, \mathscr{X}^{tgt}_{gen}\right)$	32.87	30.57	56.16	**24.35**
IW $\left(\mathscr{X}^{tgt}_{val}, \mathscr{X}^{tgt}_{gen}\right)$	-4.27	-4.02	-6.35	**-3.88**

distance [9] between two datasets $\mathscr{X}_1$ and $\mathscr{X}_2$ as

$$\mathrm{IW}(\mathscr{X}_1, \mathscr{X}_2) = \mathbb{E}_{x\sim\mathscr{X}_1}\left(\hat{D}(x)\right) - \mathbb{E}_{x\sim\mathscr{X}_2}\left(\hat{D}(x)\right) \tag{2.6}$$

In this case, $\mathscr{X}_1$ is typically a validation set, used to train the independent critic. We report IW only in some experiments, due to the larger computational cost that requires training a network for each measurement.

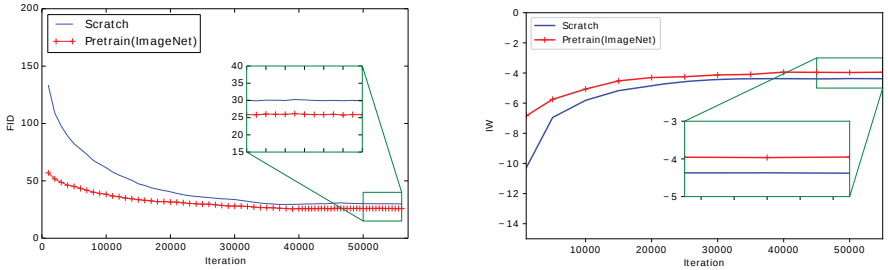
2.3.3 GAN adaptation

To study the effect of domain transfer for GANs we will use the WGAN-GP [56] architecture which uses ResNet in both generator and discriminator. This architecture has been experimentally demonstrated to be stable and robust against mode collapse [56]. The generator consists of one fully connected layer, four Residual Blocks and one convolution layer, and the Discriminator has same setting. The same architecture is used for conditional GAN.

Implementation details

We generate images of 64×64 pixels, using standard values for hyperparameters. The source models[2] are trained with a batch of 128 images during 50K iterations (except 10K iterations for CelebA) using Adam [82] and a learning rate of 1e-4. For fine tuning we use a batch size of 64 and a learning rate of 1e-4 (except 1e-5 for 1K target samples). Batch normalization and layer normalization are used in the generator and discriminator respectively.

[2]The pre-trained models are available at https://github.com/yaxingwang/Transferring-GANs.



(a) Unconditional GAN(FID) (b) Unconditional GAN(IW)

Figure 2.1 – Evolution of evaluation metrics when trained from scratch or using a pre-trained model for unconditional GAN measured with (a) FID and (b) IW (source: ImageNet, target: LSUN Bedrooms, metrics: FID and IW). The curves are smoothed for easier visualization by averaging in a window of a few iterations.

2.3.4 Generator/discriminator transfer configuration

The two networks of the GAN (generator and discriminator) can be initialized with either random or pre-trained weights (from the source networks). In a first experiment we consider the four possible combinations using a GAN pre-trained with ImageNet and 100K samples of LSUN bedrooms as target dataset. The source GAN was trained for 50K iterations. The target GAN was trained for (additional) 40K iterations.

Table 2.1 shows the results. Interestingly, we found that transferring the discriminator is more critical than transferring the generator. The former helps to improve the results in both FID and IW metrics, while the latter only helps if the discriminator was already transferred, otherwise harming the performance. Transferring both obtains the best result. We also found that training is more stable in this setting. Therefore, in the rest of the experiments we evaluated either training both networks from scratch or pre-training both (henceforth simply referred to as *pre-trained*).

Figure 2.1 shows the evolution of FID and IW during the training process with and without transfer. Networks adapted from a pre-trained model can generate images of given scores in significantly fewer iterations. Training from scratch for a long time manages to reduce this gap significantly, but pre-trained GANs can generate images with good quality already with much fewer iterations. Figures 2.2 and 2.4 show specific examples illustrating visually these conclusions.

The number of training images is critical to obtain realistic images, in particular as the resolution increases. Our experimental settings involve generating images of 64×64 pixels, where GANs typically require hundreds of thousands of training

Table 2.2 – FID/IW for different sizes of the target set (LSUN Bedrooms) using ImageNet as source dataset.

Target samples	1K	5K	10K	50K	100K	500K	1M
From scratch	256.1/-33.3	86.0/-18.5	73.7/-15.3	45.5/-7.4	32.9/-4.3	24.9/-3.6	21.0/-2.9
Pre-trained	93.4/-22.5	74.3/-16.3	47.0/-7.0	29.6/-4.56	24.4/-4.0	21.6/-3.2	18.5/-2.8

images to obtain convincing results. We evaluate our approach in a challenging setting where we use as few as 1000 images from the LSUN Bedrooms dataset, and using ImageNet as source dataset. Note that, in general, GANs evaluated on LSUN Bedrooms use the full set of 3M million images.

2.3.5 Size of the target dataset

Table 2.2 shows FID and IW measured for different amounts of training samples of the target domain. As the training data becomes scarce, the training set implicitly becomes less representative of the full dataset (i.e. less diverse). In this experiment, a GAN adapted from the pre-trained model requires roughly between two and five times fewer images to obtain a similar score than a GAN trained from scratch. FID and IW are sensitive to this factor, so in order to have a lower bound we also measured the FID between the specific subset used as training data and the full dataset. With 1K images this value is even higher than the value for generated samples after training with 100K and 1M images.

Intializing with the pre-trained GAN helps to improve the results in all cases, being more significant as the target data is more limited. The difference with the lower bound is still large, which suggests that there is still field for improvement in settings with limited data.

Figure 2.2 shows images generated at different iterations. As in the previous case, pre-trained networks can generate high quality images already in earlier iterations, in particular with sharper and more defined shapes and more realistic fine details. Visually, the difference is also more evident with limited data, where learning to generate fine details is difficult, so adapting pre-trained networks can transfer relevant prior information.

2.3.6 Source and target domains

The domain of the source model and its relation with the target domain are also a critical factor. We evaluate different combinations of source domains and target domains (see Table 2.3 for details). As source datasets we used ImageNet, Places, LSUN

Figure 2.2 – Images generated at different iterations (from 0 and 10000, step 2000) for LSUN bedrooms training from scratch and from a pre-trained network. Better viewed in electronic version.

Bedrooms and CelebA. Note that both ImageNet and Places cover wide domains, with great diversity in objects and scenes, respectively, while LSUN Bedrooms and CelebA cover more densely a narrow domain. As target we used smaller datasets, including Oxford Flowers, LSUN Kitchens (a subset of 50K out of 2M images), Label Faces in the Wild (LFW) and CityScapes.

We pre-trained GANs for the four source datasets and then trained five GANs for each of the four target datasets (from scratch and initialized with each of the source GANs). The FID and IW after fine tuning are shown in Table 2.4. Pre-trained GANs achieve significantly better results. Both metrics generally agree but there are some interesting exceptions. The best source model for Flowers as target is ImageNet, which is not surprising since it contains also flowers, plants and objects in general. It is more surprising that Bedrooms is also competitive according to FID (but not so much according to IW). The most interesting case is perhaps Kitchens, since Places has several thousands of kitchens in the dataset, yet also many more

Table 2.3 – Datasets used in the experiments.

Source datasets	ImageNet [146]	Places [204]	Bedrooms [185]	CelebA [105]
Number of images	1M	2.4M	3M	200K
Number of classes	1000	205	1	1
Target datasets	Flower [121]	Kitchens [185]	LFW [66]	Cityscapes [32]
Number of images	8K	50K	13K	3.5K
Number of classes	102	1	1	1

Table 2.4 – Distance between target real data and target generated data FID/IW.

Source → Target ↓	Scratch	ImageNet	Places	Bedrooms	CelebA
Flowers	71.98/-13.62	**54.04**/**-3.09**	66.25/-5.97	56.12/-5.90	67.96/-12.64
Kitchens	42.43/-7.79	34.35/-4.45	34.59/**-2.92**	**28.54**/-3.06	38.41/-4.98
LFW	19.36/-8.62	9.65/-5.17	15.02/-6.61	7.45/-3.61	**7.16**/**-3.45**
Cityscapes	155.68/-9.32	**122.46**/-9.00	151.34/-8.94	123.21/-8.44	130.64/**-6.40**

classes that are less related. In contrast, bedrooms and kitchens are not the same class yet still very related visually and structurally, so the much larger set of related images in Bedrooms may be a better choice. Here FID and IW do not agree, with FID clearly favoring Bedrooms, and even the less related ImageNet, over Places, while IW preferring Places by a small margin. As expected, CelebA is the best source for LFW, since both contain faces (with different scales though), but Bedroom is surprisingly very close to the performance in both metrics. For Cityscapes all methods have similar results (within a similar range), with both high FID and IW, perhaps due to the large distance to all source domains.

2.3.7 Selecting the pre-trained model

Selecting a pre-trained model for a discriminative task (e.g. classification) is reduced to simply selecting either ImageNet, for object-centric domains, or Places, for scene-centric ones. The target classifier or fine tuning will simply learn to ignore non-related features and filters of the source network.

However, this simple rule of thumb does not seem to apply so clearly in our GAN transfer setting due to generation being a much more complex task than

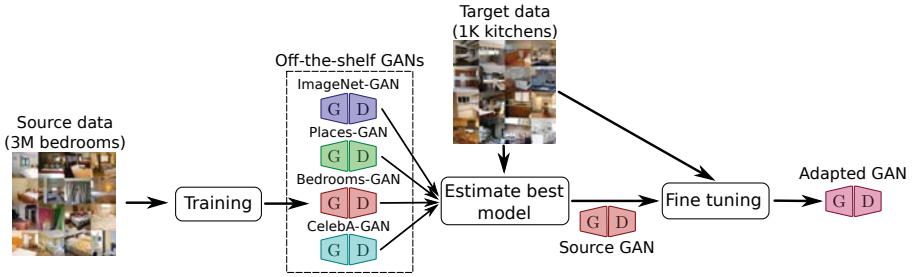


Figure 2.3 – Transferring GANs: training source GANs, estimation of the most suitable pre-trained model and adaptation to the target domain.

Table 2.5 – Distance between source generated data $\mathscr{X}_{gen}^{src}$ and target real data $\mathscr{X}_{data}^{tgt}$, and distance between source real $\mathscr{X}_{data}^{src}$ and generated data $\mathscr{X}_{gen}^{src}$.

	Source → Target ↓	ImageNet	Places	Bedrooms	CelebA
FID$\left(\mathscr{X}_{gen}^{src}, \mathscr{X}_{data}^{tgt}\right)$	Flowers	**237.04**	251.93	278.80	284.74
	Kitchens	183.27	180.63	**70.06**	254.12
	LFW	333.54	333.38	329.92	**151.46**
	Cityscapes	233.45	**181.72**	227.53	292.66
FID$\left(\mathscr{X}_{gen}^{src}, \mathscr{X}_{data}^{src}\right)$	Source	63.46	55.66	17.30	75.84

discrimination. Results in Table 2.4 show that sometimes unrelated datasets may perform better than other apparently more related. The large number of unrelated classes may be an important factor, since narrow yet dense domains also seem to perform better even when they are not so related (e.g. Bedrooms). There are also non-trivial biases in the datasets that may explain this behavior. Therefore, a way to estimate the most suitable model for a given target dataset is desirable, given a collection of pre-trained GANs.

Perhaps the most simple way is to measure the distance between the source and target domains. We evaluated the FID between the (real) images in the target and the source datasets (results included in Appendix A.1.1). While showing some correlation with the FID of the target generated data, it has the limitation of not considering whether the actual pre-trained model is able or not to accurately sample from the real distribution. A more helpful metric is the distance between the target

data and the *generated* samples by the pre-trained model. In this way, the quality of the model is taken into account. We estimate this distance also using FID. In general, there seem to roughly correlate with the final FID results with target generated data (compare Tables 2.4 and 2.5). Nevertheless, it is surprising that Places is estimated as a good source dataset but does not live up to the expectation. The opposite occurs for Bedrooms, which seems to deliver better results than expected. This may suggest that density is more important than diversity for a good transferable model, even for apparently unrelated target domains.

In our opinion, the FID between source generated and target real data is a rough indicator of suitability rather than accurate metric. It should taken into account jointly with others factors (e.g. quality of the source model) to decide which model is best for a given target dataset.

2.3.8 Visualizing the adaptation process

One advantage of the image generation setting is that the process of shifting from the source domain towards the target domain can be visualized by sampling images at different iterations, in particular during the initial ones. Figure 2.4 shows some examples of the target domain Kitchens and different source domains (iterations are sampled in a logarithmic scale).

Trained from scratch, the generated images simply start with noisy patterns that evolve slowly, and after 4000 iterations the model manages to reproduce the global layout and color, but still fails to generate convincing details. Both the GANs pre-trained with Places and ImageNet fail to generate realistic enough source images and often sample from unrelated source classes (see iteration 0). During the initial adaptation steps, the GAN tries to generate kitchen-like patterns by matching and slightly modifying the source pattern, therefore preserving global features such as colors and global layout, at least during a significant number of iterations, then slowly changing them to more realistic ones. Nevertheless, the textures and edges are sharper and more realistic than from scratch. The GAN pre-trained with Bedrooms can already generate very convincing bedrooms, which share a lot of features with kitchens. The larger number of training images in Bedrooms helps to learn transferable fine grained details that other datasets cannot. The adaptation mostly preserves the layout, colors and perspective of the source generated bedroom, and slowly transforms it into kitchens by changing fine grained details, resulting in more convincing images than with the other source datasets. Despite being a completely unrelated domain, CelebA also manages to help in speeding up the learning process by providing useful priors. Different parts such as face, hair and eyes are transformed into different parts of the kitchen. Rather than the face itself, the most predominant feature remaining from the source generated

image is the background color and shape, that influences in the layout and colors that the generated kitchens will have.

2.4 Transferring to conditional GANs

Here we study the transferring the representation learned by a pre-trained unconditional GAN to a cGAN [115]. cGANs allow us to condition the generative model on particular information such as classes, attributes, or even other images. Let y be a conditioning variable. The discriminator $D(x, y)$ aims to distinguish pairs of real data x and y sampled from the joint distribution $p_{data}(x, y)$ from pairs of generated outputs $G(z, y')$ conditioned on samples y' from y's marginal $p_{data}(y)$.

2.4.1 Conditional GAN adaptation

For the current study, we adopt the Auxiliary Classifier GAN (AC-GAN) framework of [124]. In this formulation, the discriminator has an 'auxiliary classifier' that outputs a probability distribution over classes $P(C = y|x)$ conditioned on the input x. The objective function is then composed of the conditional version of the GAN loss $\mathscr{L}_{GAN}$ (eq. (2.2)) and the log-likelihood of the correct class. The final loss functions for generator and discriminator are:

$$\mathscr{L}_{AC-GAN}(G) = \mathscr{L}_{GAN}(G) - \alpha_G \mathbb{E}\left[\log\left(P\left(C = y'|G(z, y')\right)\right)\right], \tag{2.7}$$

$$\mathscr{L}_{AC-GAN}(D) = \mathscr{L}_{GAN}(D) - \alpha_D \mathbb{E}\left[\log\left(P\left(C = y|x\right)\right)\right], \tag{2.8}$$

respectively. The parameters α_G and α_D weight the contribution of the auxiliary classifier loss with respect to the GAN loss for the generator and discriminator. In our implementation, we use Resnet-18 [58] for both G and D, and the WGAN-GP loss from the equations (2.3) and (2.4) as the GAN loss. Overall, the implementation details (batch size, learning rate) are the same as introduced in section 2.3.3.

In AC-GAN, the conditioning is performed only on the generator by appending the class label to the input noise vector. We call this variant 'Cond Concat'. We randomly initialize the weights which are connected to the conditioning prior. We also used another variant following [39], in which the conditioning prior is embedded in the batch normalization layers of the generator (referred to as 'Cond BNorm'). In this case, there are different batch normalization parameters for each class. We initialize these parameters by copying the values from the unconditional GAN to all classes.

Table 2.6 – Per-class and overall FID for AC-GAN. Source: Places, target: LSUN

Init	Iter	Bedr	Bridge	Church	Classr	Confer	Dining	Kitchen	Living	Rest	Tower	Avg.	All
Scratch	250	298.4	310.3	314.4	376.6	339.1	294.9	314.2	316.5	324.4	301.0	319.0	352.4
	2500	195.9	135.0	133.0	218.6	185.3	173.9	167.9	189.3	159.5	125.6	168.4	137.3
	25000	72.9	78.0	52.4	106.7	76.9	40.1	53.9	56.1	74.7	59.8	67.2	49.6
Pre-trained	250	**168.3**	**122.1**	**148.1**	**145.0**	**151.6**	**144.2**	**156.9**	**150.1**	**113.3**	**129.7**	**142.9**	**107.2**
	2500	**140.8**	**96.8**	**77.4**	**136.0**	**136.8**	**84.6**	**85.5**	**94.9**	**77.0**	**69.4**	**99.9**	**74.8**
	25000	**59.9**	**68.6**	**48.2**	**79.0**	**68.7**	**35.2**	**48.2**	**47.9**	**44.4**	**49.9**	**55.0**	**42.7**

2.4.2 Results

We use Places [204] as the source domain and consider all the ten classes of the LSUN dataset [185] as target domain. We train the AC-GAN with 10K images per class for 25K iterations. The weights of the conditional GAN can be transferred from the pre-trained unconditional GAN (see section 2.3.1) or initialized at random. The performance is assessed in terms of the FID score between target domain and generated images. The FID is computed class-wise, averaging over all classes and also considering the dataset as a whole (class-agnostic case). The classes in the target domain have been generated uniformly. The results are presented in table 2.6, where we show the performance of the AC-GAN whose weights have been transferred from pre-trained network vs. an AC-GAN initialized randomly. We computed the FID for 250, 2500 and 25000 iterations. At the beginning of the learning process, there is a significant difference between the two cases. The gap is reduced towards the end of the learning process but a significant performance gain still remains for pre-trained networks. We also consider the case with fewer images per class. The results after 25000 iterations for 100 and 1K images per class are provided in the last column of table 2.7. We can observe how the difference between networks trained from scratch or from pre-trained weights is more significant for smaller sample sizes. This confirms the trend observed in section 2.3.5: transferring the pre-trained weights is especially advantageous when only limited data is available.

The same behavior can be observed in figure 2.5 (left) where we compare the performance of the AC-GAN with two unconditional GANs, one pre-trained on the source domain and one trained from scratch, as in section 2.3.4. The curves correspond to the class-agnostic case (column 'All' in the table 2.6). From this plot, we can observe three aspects: (i) the two variants of AC-GAN perform similarly (for this reason, for the remaining of the experiments we consider only 'Cond BNorm'); (ii) the network initialized with pre-trained weights converges faster than the network trained from scratch, and the overall performance is better; and (iii) AC-GAN performs slightly better than the unconditional GAN.

Table 2.7 – Accuracy of AC-GAN for the classification task and overall FID for different sizes of the target set (LSUN).

#images	Method	Accuracy (%)											FID
		Bedr	Bridge	Church	Classr	Confer	Dining	Kitchen	Living	Rest	Tower	Avg.	
100/class	scratch	23.0	**88.2**	**55.1**	29.2	3.6	24.9	20.8	8.4	**89.3**	**61.6**	40.4	162.9
	pre-trained	**35.7**	72.7	45.7	**59.4**	**7.9**	**38.2**	**36.3**	**20.1**	81.0	56.6	**45.4**	**119.1**
1K/class	scratch	49.9	78.1	**75.1**	51.8	14.6	51.2	31.2	23.2	**90.7**	61.5	52.7	117.3
	pre-trained	**76.4**	**82.5**	69.1	**80.6**	**34.2**	**52.6**	**62.4**	**52.9**	80.5	**67.5**	**65.9**	**77.5**
10K/class	scratch	**94.9**	94.3	89.6	85.0	82.4	**91.2**	88.0	86.9	91.3	83.5	88.7	49.6
	pre-trained	87.1	**95.7**	**90.8**	**95.1**	**86.8**	90.2	**88.9**	**90.1**	**93.0**	**88.9**	**90.8**	**42.7**

Next, we evaluate the AC-GAN performance on a classification experiment. We train a reference classifier on the 10 classes of LSUN (10K real images per class). Then, we evaluate the quality of each model trained for 25K iterations by generating 10K images per class and measuring the accuracy of the reference classifier for 100, 1K and 10K images per class. The results show an improvement when using pre-trained models, with higher accuracy and lower FID in all settings, suggesting that it captures better the real data distribution of the dataset compared to training from scratch.

Finally, we perform a psychophysical experiment with generated images by AC-GAN with LSUN as target. Human subjects are presented with two images: pre-trained vs. from scratch (generated from the same condition `<class>`), and asked 'Which of these two images of `<class>` is more realistic?' Subjects were also given the option to skip a particular pair should they find very hard to decide for one of them. We require each subject to provide 100 valid assessments. We use 10 human subjects which evaluate image pairs for different settings (100, 1K, 10K images per class). The results (Fig. 2.5 right) clearly show that the images based on pre-trained GANs are considered to be more realistic in the case of 100 and 1K images per class (e.g. pre-trained is preferred in 67% of cases with 1K images). As expected the difference is smaller for the 10K case.

2.5 Conclusions

We show how the principles of transfer learning can be applied to generative features for image generation with GANs. GANs, and conditional GANs, benefit from transferring pre-trained models, resulting in lower FID scores and more recognizable images with less training data. Somewhat contrary to intuition, our experiments show that transferring the discriminator is much more critical than the generator (yet transferring both networks is best). However, there are also other important

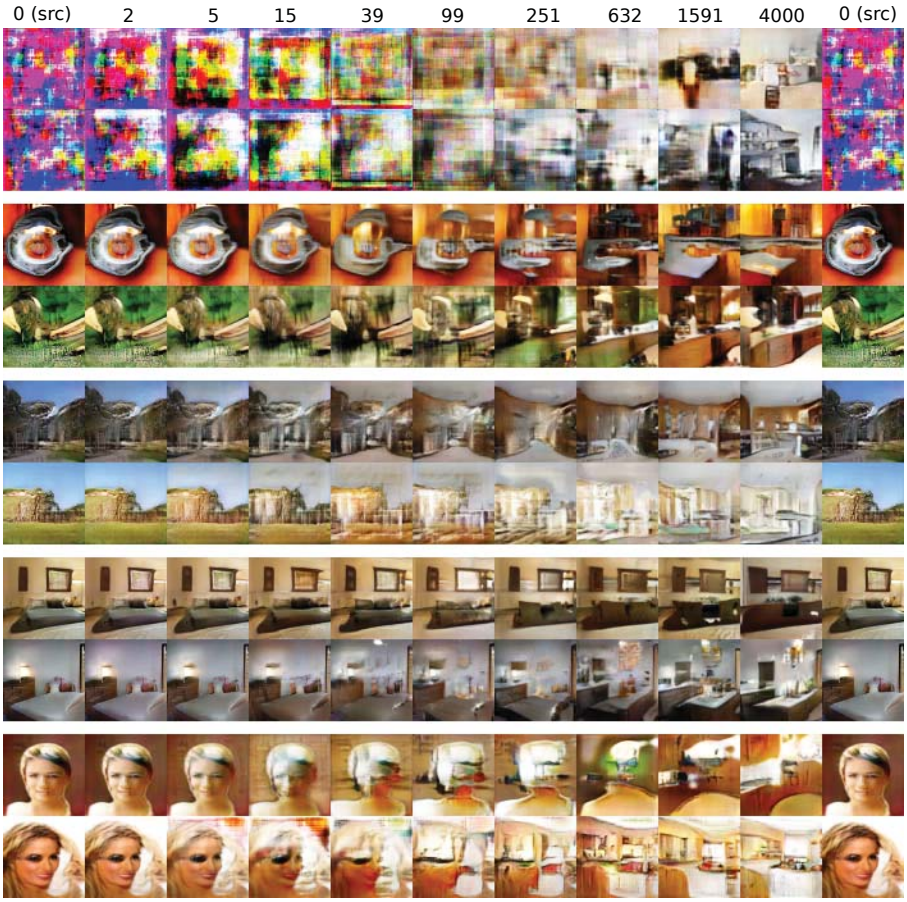


Figure 2.4 – Evolution of generated images (in logarithmic scale) for LSUN kitchens with different source datasets (from top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA). Better viewed in electronic version.

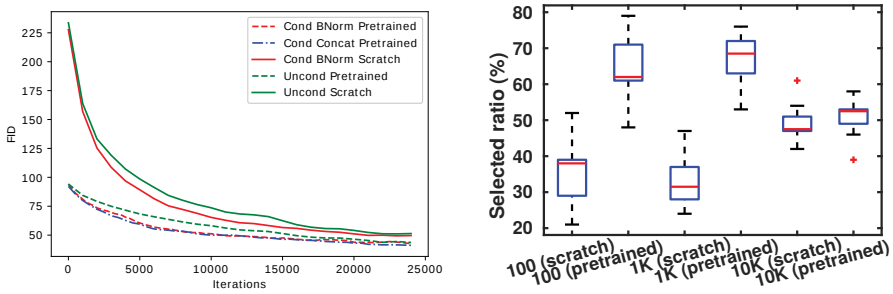


Figure 2.5 – (Left) FID score for Conditional and Unconditional GAN (source: Places, target: LSUN 10 classes). (Right) Human evaluation of image quality.

differences with the discriminative scenario. Notably, it seems that a much higher density (images per class) is required to learn good transferable features for image generation, than for image discrimination (where diversity seems more critical). As a consequence, ImageNet and Places, while producing excellent transferable features for discrimination, seem not dense enough for generation, and LSUN data seems to be a better choice despite its limited diversity. Nevertheless, poor transferability may be also related to the limitations of current GAN techniques, and better ones could also lead to better transferability.

Our experiments evaluate GANs in settings rarely explored in previous works and show that there are many open problems. These settings include GANs and evaluation metrics in the very limited data regime, better mechanisms to estimate the most suitable pre-trained model for a given target dataset, and the design of better pre-trained GAN models.

3 MineGAN: effective knowledge transfer from GANs to target domains with few images[1]

3.1 Introduction

Generative adversarial networks (GANs) can learn the complex underlying distribution of image collections [54]. They have been shown to generate high-quality realistic images [19, 76, 77] and are used in many applications including image manipulation [70, 206], style transfer [49], compression [163], and colorization [196].

It is known that high-quality GANs require a significant amount of training data and time. For example, Progressive GANs [76] are trained on 30K images and are reported to require a month of training on one NVIDIA Tesla V100. Being able to exploit these high-quality pre-trained models, not just to generate the distribution on which they are trained, but also to combine them with other models and adjust them to a target distribution is a desirable objective. For instance, it might be desirable to only generate women using a GAN trained to generate men and women alike. Alternatively, one may want to generate smiling people from two pre-trained generative models, one for men and one for women. The focus of this chapter is on performing these operations using only a small target set of images, and without access to the large datasets used to pretrain the models.

Transferring knowledge to domains with limited data has been extensively studied for discriminative models [37, 125, 126, 164], enabling the re-use of high-quality networks. However, knowledge transfer for generative models has received significantly less attention, possibly due to its great difficulty, especially when transferring to target domains with few images. single pre-trained generative model and showed that it is beneficial for domains with scarce data. However, Noguchi and Harada [122] observed that this technique leads to mode collapse. Instead, they proposed to reduce the number of trainable parameters, and only finetune the learnable parameters for the batch normalization (scale and shift) of the generator. Despite being less prone to overfitting, their approach severely limits the flexibility of the knowledge transfer.

[1] This chapter is under review in the Conference on Computer Vision and Pattern Recognition (CVPR 2020).

In this work, we address knowledge transfer by adapting a trained generative model for targeted image generation given a small sample of the target distribution. We introduce the process of *mining* of GANs. This is performed by a *miner network* that transforms a multivariate normal distribution into a distribution on the input space of the pre-trained GAN in such a way that the generated images resemble those of the target domain. The miner network has considerably fewer parameters than the pre-trained GAN and is therefore less prone to overfitting. The mining step predisposes the pre-trained GAN to sample from a narrower region of the latent distribution that is closer to the target domain, which in turn eases the subsequent finetuning step by providing a cleaner training signal with lower variance (in contrast to sampling from the whole source latent space as in [173]). Consequently, our method preserves the adaptation capabilities of finetuning while preventing overfitting. Importantly, our mining approach enables transferring from multiple pre-trained GANs, which allows us to aggregate information from multiple sources simultaneously to generate samples akin to the target domain. We show that these networks can be trained by a selective backpropagation procedure. Our main contributions are:

- We introduce a novel miner network to steer the sampling of the latent distribution of a pre-trained GAN to a target distribution determined by few images. The miner network has relatively few parameters and is therefore less prone to overfitting.
- We propose the first method to transfer knowledge from multiple GANs to a single generative model.
- We evaluate the proposed approach on a variety of settings, including transferring knowledge from unconditional, conditional, and multiple GANs. Experiments are performed on high-resolution datasets with high complexity such as LSUN [183], CelebA [105] and ImageNet [84]. We outperform existing competitors, including TransferGAN [173] and BSA [122].

3.2 Related work

Generative adversarial networks. GANs consists of two modules: a generator and a discriminator [54]. The generator aims to generate images to fool the discriminator, while the discriminator aims to distinguish generated from real images. Training GANs was initially difficult, as they often suffer from mode collapse and unstable training issues. Several previous methods focus on addressing these problems [10, 56, 110, 116, 149]. Another major line of research aims to improve the model architectures to generate higher quality images [19, 36, 76, 77, 132]. Progressive GAN [76] generates high-quality images by means of synthesizing images

progressively from low to high-resolution. Finally, BigGAN [19] successfully performs conditional high-realistic generation from ImageNet [35].

Transfer learning for GANs. While knowledge transfer has been widely studied for discriminative models in computer vision [37, 125, 126, 164], only a few works have explored transferring knowledge for generative models [122, 173]. Chapter 2 investigated finetuning of pre-trained GANs, leading to improved performance for target domains with limited samples. This method, however, suffers from mode collapse and overfitting, as it updates all parameters of the generator to adapt to the target domain. Recently, Noguchi and Harada [122] proposed to only update the batch normalization parameters. Although less susceptible to mode collapse, this approach significantly reduces the adaptation flexibility of the model since changing only the parameters of the batch normalization permits for style changes but is not expected to function when shape needs to be changed. They also replaced the GAN loss with a mean square error loss. As a result, their model only learns the relationship between latent vectors and sparse training samples, requiring the input noise distribution to be truncated during inference to generate realistic samples. The proposed MineGAN does not suffer from this drawback, as it learns how to automatically adapt the input distribution. In addition, we are the first to consider transferring knowledge from multiple GANs to a single target domain.

Iterative image generation. Nguyen et al. [120] have investigated training networks to generate images that maximize the activation of neurons in a pre-trained classification network. In a follow-up approach [119] that improves the diversity of the generated images, they use this technique to generate images of a particular class from a pre-trained classifier network. In principle, these works do not aim at transferring knowledge to a new domain, and can instead only be applied to generate a distribution that is exactly described by one of the class labels of the pre-trained classifier network. Another major difference is that the generation at inference time of each image is an iterative process of successive backpropagation updates until convergence, whereas our method is feedforward during inference.

3.3 Mining operations on GANs

Assume we have access to one or more pre-trained GANs and wish to use their knowledge to train a new GAN for a target domain with few images. For clarity's sake, we first introduce mining from a single GAN in Section 3.3.2, but our method is general for an arbitrary number of pre-trained GANs, as explained in Section 3.3.3. Then, we show how the miners can be used to train new GANs (Section 3.3.4).

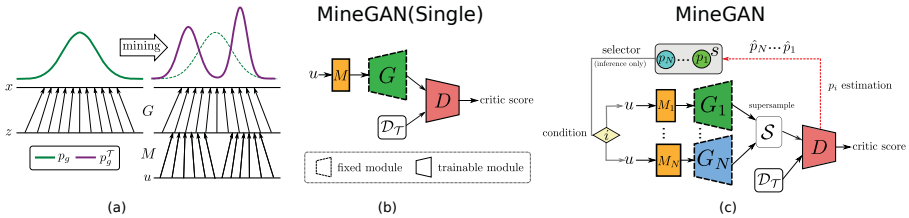


Figure 3.1 – (a) Intuition behind our approach for a simple case. Mining shifts the prior input distribution towards the most promising regions with respect to given target data $\mathscr{D}_{\mathscr{T}}$. In practice, the input distribution is much more complex. (b) Architecture implementing the proposed mining operation on a single GAN. Miner M identifies the relevant regions of the prior distribution so that generated samples are close to the target data $\mathscr{D}_{\mathscr{T}}$. Note that when training the miner the generator remains fixed. (c) Training setup for multiple generators. Miners M_1,...,M_N identify subregions of the pre-trained generators while selector $\mathscr{S}$ learns the sampling frequencies of the various generators.

3.3.1 GAN formulation

Let $p_{data}(x)$ be a probability distribution over real data x determined by a set of real images $\mathscr{D}$, and let $p_z(z)$ be a prior distribution over an input noise variable z. The generator G is trained to synthesize images given $z \sim p_z(z)$ as input, inducing a generative distribution $p_g(x)$ that should approximate the real data distribution $p_{data}(x)$. This is achieved through an adversarial game [54], in which a discriminator D aims to distinguish between real images and images generated by G, while the generator tries to generate images that fool D.

In this work, we follow the WGAN-GP [56] approach, which provides better convergence properties by using the Wasserstein loss [10] and a gradient penalty term (omitted from our formulation for simplicity).

The discriminator (or *critic*) and generator losses are defined as follows:

$$\mathscr{L}_D = \mathbb{E}_{z\sim p_z(z)}[D(G(z))] - \mathbb{E}_{x\sim p_{data}(x)}[D(x)], \tag{3.1}$$

$$\mathscr{L}_G = -\mathbb{E}_{z\sim p_z(z)}[D(G(z))]. \tag{3.2}$$

We also consider families of pre-trained generators $\{G_i\}$. Each G_i has the ability to synthesize images given input noise $z \sim p_z^i(z)$. For simplicity and without loss of generality, we assume the prior distributions are Gaussian, i.e. $p_z^i(z) = \mathscr{N}(z|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$. Each generator $G_i(z)$ induces a learned generative distribution $p_g^i(x)$, which ap-

proximates the corresponding real data distribution $p^i_{data}(x)$ over real data x given by the set of source domain images $\mathscr{D}_i$.

3.3.2 Mining from a single GAN

We would like to approximate a target real data distribution $p^{\mathscr{T}}_{data}(x)$ induced by a set of real images $\mathscr{D}_{\mathscr{T}}$, given a critic D and a generator G, which have been trained to approximate a source data distribution $p_{data}(x)$ via the generative distribution $p_g(x)$. The mining operation learns a new generative distribution $p^{\mathscr{T}}_g(x)$ by finding those regions in $p_g(x)$ that better approximate the target data distribution $p^{\mathscr{T}}_{data}(x)$ while keeping G fixed. In order to find such regions, mining actually finds a new prior distribution $p^{\mathscr{T}}_z(z)$ such that samples $G(z)$ with $z \sim p^{\mathscr{T}}_z(z)$ are similar to samples from $p^{\mathscr{T}}_{data}(x)$ (see Fig. 3.1a). For this purpose, we propose a new GAN component called *miner* which is a small network M, implemented by a multilayer perceptron. Its goal is to transform the original input noise variable $u \sim p_z(u)$ to follow a new, more suitable prior that identifies the regions in $p_g(x)$ that most closely align with the target distribution.

Fig. 3.1b presents the proposed mining architecture, called *MineGAN*. Miner M acts as an interface between the input noise variable and the generator, which remains fixed during training. To generate an image, we first sample $u \sim p_z(u)$, transform it with M and then input the transformed variable to the generator, i.e. $G(M(u))$. We train the model adversarially: the critic D aims to distinguish between fake images output by the generator $G(M(u))$ and real images x from the target data distribution $p^{\mathscr{T}}_{data}(x)$. We implement this with the following modification on the WGAN-GP loss

$$\mathscr{L}^M_D = \mathbb{E}_{u\sim p_z(u)}[D(G(M(u)))] - \mathbb{E}_{x\sim p^{\mathscr{T}}_{data}(x)}[D(x)], \tag{3.3}$$

$$\mathscr{L}^M_G = -\mathbb{E}_{u\sim p_z(u)}[D(G(M(u)))]. \tag{3.4}$$

The parameters of G are kept unchanged but the gradients are backgropagated all the way to M to learn its parameters. This training strategy will gear the miner towards the most promising regions of the input space, i.e. those that generate images close to $\mathscr{D}_{\mathscr{T}}$. Therefore, M is effectively mining the relevant input regions of prior $p_z(u)$ and giving rise to a targeted prior $p^{\mathscr{T}}_z(z)$, which will focus on these regions while ignoring other ones that lead to samples far off the target distribution $p^{\mathscr{T}}_{data}(x)$.

We distinguish two types of targeted generation: on-manifold and off-manifold. In the *on-manifold* case, there is a significant overlap between the original distribu-

tion $p_{data}(x)$ and the target distribution $p^{\mathcal{T}}_{data}(x)$. For example, $p_{data}(x)$ could be the distribution of human faces (both male and female) while $p^{\mathcal{T}}_{data}(x)$ includes female faces only. On the other hand, in *off-manifold* generation, the overlap between the two distributions is negligible, e.g. $p^{\mathcal{T}}_{data}(x)$ contains cat faces. The off-manifold task is evidently more challenging as the miner needs to find samples out of the original distribution (see Fig. 3.4). Specifically, we can consider the images in $\mathscr{D}$ to lie on a high-dimensional image manifold that contains the support of the real data distribution $p_{data}(x)$ [9]. For a target distribution farther away from $p_{data}(x)$, its support will be more disjoint from the original distribution's support, and thus its samples might be off the manifold that contains $\mathscr{D}$.

3.3.3 Mining from multiple GANs

In the general case, the mining operation is applied on multiple pre-trained generators. Given target data $\mathscr{D}_{\mathcal{T}}$, the task consists in mining relevant regions from the induced generative distributions learned by a family of N generators $\{G_i\}$. In this task, we do not have access to the original data used to train $\{G_i\}$ and can only use target data $\mathscr{D}_{\mathcal{T}}$. Fig. 3.1c presents the architecture of our model, which extends the mining architecture for a single pre-trained GAN by including multiple miners and an additional component called *selector*. In the following, we present this component and describe the training process in detail.

Supersample. In traditional GAN training, a fake minibatch is composed of fake images $G(z)$ generated with different samples $z \sim p_z(z)$. To construct fake minibatches for training a set of miners, we introduce the concept of *supersample*. A supersample $\mathscr{S}$ is a set of samples composed of exactly one sample per generator of the family, i.e. $\mathscr{S} = \{G_i(z)|z \sim p^i_z(z); i = 1,...,N\}$. Each minibatch contains K supersamples, which amounts to a total of $K \times N$ fake images per minibatch.

Selector. The selector's task is choosing which pre-trained model to use for generating samples during inference. For instance, imagine that $\mathscr{D}_1$ is a set of 'kitchen' images and $\mathscr{D}_2$ are 'bedroom' images, and let $\mathscr{D}_{\mathcal{T}}$ be 'white kitchens'. The selector should prioritize sampling from G_1, as the learned generative distribution $p^1_g(x)$ will contain kitchen images and thus will naturally be closer to $p^{\mathcal{T}}_{data}(x)$, the target distribution of white kitchens. Should $\mathscr{D}_{\mathcal{T}}$ comprise both white kitchens and dark bedrooms, sampling should be proportional to the distribution in the data.

We model the selector as a random variable s following a categorical distribution parametrized by $p_1,...,p_N$ with $p_i > 0$ and $\sum p_i = 1$. We estimate the parameters of this distribution as follows. The quality of each sample $G_i(z)$ is evaluated by a single critic D based on its critic value $D(G_i(z))$. Higher critic values indicate that the generated sample from G_i is closer to the real distribution.

For each supersample $\mathscr{S}$ in the minibatch, we record which generator obtains the maximum critic value, i.e. $\text{argmax}_i D(G_i(z))$. By accumulating over all K supersamples and normalizing, we obtain an empirical probability value $\hat{p}_i$ that reflects how often generator G_i obtained the maximum critic value among all generators for the current minibatch. We estimate each parameter p_i as the empirical average $\hat{p}_i$ estimated in the last 1000 minibatches. Note that p_i are learned during training and stay fixed during inference.

Critic and miner training. We now define the training behavior of the remaining learnable components, namely the critic D and miners $\{M_i\}$, when minibatches are composed of supersamples. The critic aims to distinguish real images from fake images. This is done by looking for artifacts in the fake images which distinguish them from the real ones. Another, less discussed but equally important task of the critic, is to observe the frequency of occurrence of images: if some (potentially high-quality) image occurs more often among fake images than real ones, the critic will lower its score, and thereby motivate the generator to lower the frequency of occurrence of this image. Training the critic by backpropagating from all images in the supersample prevents it from assessing the frequency of occurrence of the generated images (and we empirically observed this to yield unsatisfactory results). Therefore, the training loss for multiple GAN mining is:

$$\begin{aligned}\mathscr{L}_D^M = \mathbb{E}_{\{u^i \sim p_z^i(u)\}}[\max_i\{D(G_i(M_i(u^i)))\}] \\ -\mathbb{E}_{x\sim p_{data}^{\mathscr{T}}(x)}[D(x)]\end{aligned} \tag{3.5}$$

$$\mathscr{L}_G^M = -\mathbb{E}_{\{u^i \sim p_z^i(u)\}}[\max_i\{D(G_i(M_i(u^i)))\}]. \tag{3.6}$$

As a result of the max operator we only backpropagate from the generated image that obtained the highest critic score. Training with Eq. 3.6 allows the critic to assess the frequency of occurrence correctly. Using this strategy, the critic can perform both its tasks: boosting the quality of the images as well as driving the miner to closely follow the distribution of the target set. Note that in this case we initialize the single critic D with the pre-trained weights from one of the pre-trained critics[2].

Conditional GANs. So far, we have only considered unconditional GAN models. However, conditional GANs are used by the most successful approaches [19, 190]. cGANs introduce an additional input variable to condition the generation to the class label. Here we extend our proposed MineGAN to cGANs that condition on

[2]We empirically found that starting from any pre-trained critic leads to similar results (see Appendix A.2.4)

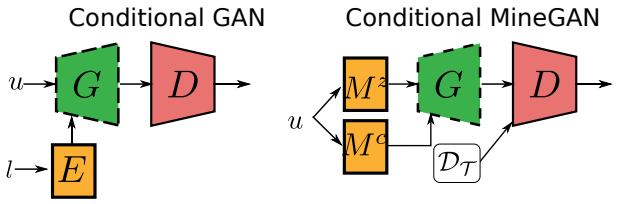


Figure 3.2 – Application of mining in conditional setting (on BigGAN [19]). We apply an additional miner network to estimate the class embedding. $\mathcal{D}_{\mathcal{T}}$: target data, E: class embedding, l: label.

the batch normalization layer [19, 39][3]More concretely, we experiment with BigGAN [19], as shown in Fig. 3.2 (left). First, a label l is mapped to an embedding vector by means of a class embedding E, and then this vector is mapped to layer-specific batch normalization parameters. The discriminator is further conditioned via label projection [117]. Fig. 3.2 (right) shows how to mine BigGANs. Alongside the standard miner M^z, we introduce a second miner network M^c, which maps from u to the embedding space, resulting in a generator $G(M^c(u), M^z(u))$. The training is equal to that of a single GAN and follows Eqs. 3.3 and 3.4.

3.3.4 Knowledge transfer with MineGAN

The underlying idea of mining is to predispose the pre-trained model to the target distribution by reducing the divergence between source and target distributions. The miner network contains relatively few parameters and is therefore less prone to overfitting, which is known to occur when directly finetuning the generator G [122]. We finalize the knowledge transfer to the new domain by finetuning both the miner M and generator G (by releasing its weights). The risk of overfitting is now diminished as the generative distribution is closer to the target, thus requiring thus a lower degree of parameter adaptation. Moreover, the training is substantially more efficient than directly finetuning the pre-trained GAN [173], where synthesized images are not necessarily similar to the target samples. A mined pre-trained model makes the sampling more effective, leading to less noisy gradients and a cleaner training signal.

[3]See Appendix A.2.2.

3.4 Experiments

In this section, we first introduce the used evaluation measures and architectures. Then, we evaluate our method for knowledge transfer from unconditional GANs, considering both a single and multiple pre-trained generators. Finally, we assess transfer learning from conditional GANs. Our experiments focus on transferring knowledge to target domains with few images.

Evaluation measures. We employ the widely used Fréchet Inception Distance (FID) [61] for evaluation. FID measures the similarity between two sets in the embedding space given by the features of a convolutional neural network. More specifically, it computes the differences between the estimated means and covariances assuming a multivariate normal distribution on the features. FID measures both the quality and diversity of the generated images and has been shown to correlate well with human perception [61]. However, it suffers from instability on small datasets. For this reason, we also employ Kernel Maximum Mean Discrepancy (KMMD) with a Gaussian kernel and Mean Variance (MV) for some experiments [122]. Low KMMD values indicate high quality images, while high values of MV indicate more image diversity.

Baselines. We compare our method with the following baselines. *TransferGAN [173]* directly updates both the generator and the discriminator for the target domain. *VAE [83]* is a variational autoencoder trained following [122], i.e. fully supervised by pairs of latent vectors and training images. *BSA [122]* updates only the batch normalization parameters of the generator instead of all the parameters. *DGN-AM [120]* generates images that maximize the activation of neurons in a pre-trained classification network. *PPGN [119]* improves the diversity of DGN-AM by of adding a prior to the latent code via denoising autoencoder. Note that both of DGN-AM and PPGN require the target domain label, and thus we only include them in the conditional setting.

Architectures. We introduce mining to several architectures, including Progressive GAN [76], SNGAN [116], and BigGAN [19]. The training details for all models are included in Appendix A.2.1. For the miner, we use four fully connected layers for all experiments except those on MNIST, where we use only two.

3.4.1 Knowledge transfer from unconditional GANs

MNIST dataset. To illustrate the functioning of the miner we show some results MNIST [90] dataset[4]. We use 1000 images of size 28×28 as target data. We test mining for off-manifold targeted image generation. In off-manifold targeted gen-

[4]We add quantitative results on MNIST in Appendix A.2.2

Figure 3.3 – Results for off-manifold generation of MineGAN(w/o FT). We generate 20 samples of digits '5', '8' or '9'.

eration, G is pre-trained to synthesize all MNIST digits except for the target one, e.g. G generates 0-8 but not 9. Here we illustrate the results after only training the miner, without an additional finetuning step. The results are depicted in Fig. 3.3. Interestingly, the miner manages to steer the generator to output samples that resemble the target digits, mostly by using and merging patterns from other digits in the source set. For example, digit '9' frequently resembles a modified 4 while '8' heavily borrows from 0s and 3s. We can also observe that some digits can be more challenging to generate. For example, '5' is generally more distinct from other digits and thus in more cases the resulting sample is confused with other digits such as '3'. In conclusion, even though target classes are not in the training set of the pre-trained GAN, still similar examples might be found on the manifold of the generator.

Single pre-trained model. We start by transferring knowledge from a Progressive GAN trained on *CelebA [105]*. We evaluate the performance on target datasets of varying size with 1024×1024 images. We consider two target domains: on-manifold, *FFHQ women* [77] and off-manifold, *FFHQ children face* [77]. We consider two versions of our model: *MineGAN* refers to the mining method combined with finetuning to the target domain, whereas *MineGAN(w/o FT)* only applies mining. We compare our results to training from *Scratch*, and the *TransferGAN* method of [173]. In the plots in Fig. 3.5, we show the performance in terms of FID and KMMD as a function of the number of images in the target domain. The proposed MineGAN framework outperforms all baselines. For the on-manifold experiment, MineGAN already outperforms the other baselines, and results are further improved with additional finetuning. Interestingly, for the off-manifold experiment, MineGAN without finetuning obtains only slightly worse results than TransferGAN, showing that the miner already manages to generate images close to the target domain.

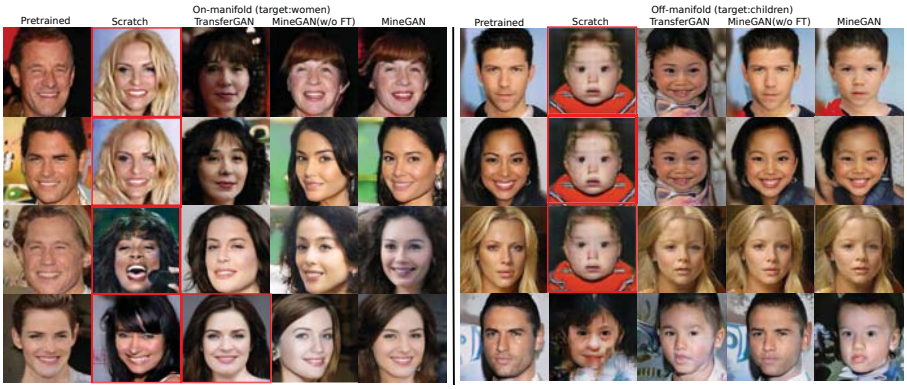


Figure 3.4 – Results: (Left) On-manifold (CelebA→FFHQ women), (Right) Off-manifold (CelebA→FFHQ children). Based on pre-trained Progressive GAN. The images with red box are suffering from overfitting. We show this in Appendix A.2.3. More examples are shown in Appendix A.2.3.

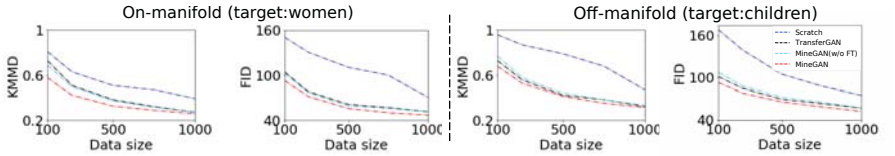
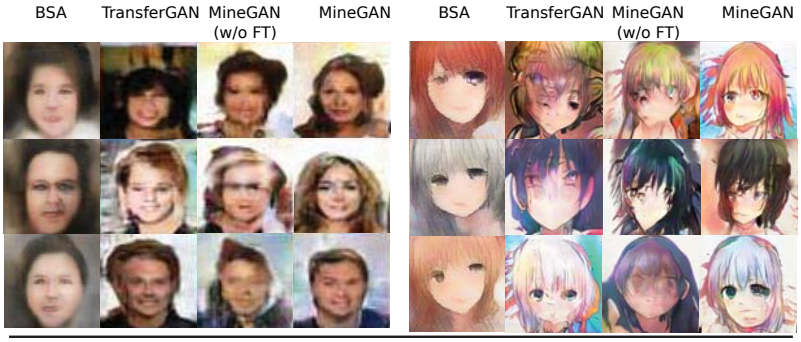


Figure 3.5 – KMMD and FID on CelebA→FFHQ women (left) and CelebA→FFHQ children (right).

Fig. 3.4 shows images generated when the target data contains 100 training images. Training the model from scratch results in overfitting. Also TransferGAN sometimes suffers from overfitting. MineGAN, in contrast, generates high-quality images without overfitting to the target domain. The generated images are sharper, more diverse, and have more realistic fine details.

We also compare here with Batch Statistics Adaptation (BSA) [122] using the same settings and architecture, namely SNGAN [116]. They performed knowledge transfer from a pre-trained SNGAN on ImageNet [84] to FFHQ [77] and to Anime Face [7]. Target domains have only 25 images of size 128×128. We added our results to those reported in [122] in Fig. 3.6 (bottom). Compared to BSA, MineGAN (w/o FT) obtains similar KMMD scores, showing that generated images obtain comparable quality. MineGAN outperforms BSA both in KMMD score and Mean Variance. The qualitative results (shown in Fig. 3.6 (top)) clearly show that MineGAN outperforms



Method	FFHQ		Anime Face	
	KMMD	MV	KMMD	MV
From scratch	0.890	-	0.753	-
TransferGAN [173]	0.346	0.506	0.347	0.785
VAE [83]	0.744	-	0.790	-
BSA [122]	0.345	0.785	0.342	0.908
MineGAN (w/o FT)	0.349	0.774	0.347	0.891
MineGAN	**0.337**	**0.812**	**0.334**	**0.934**

Figure 3.6 – Results for various knowledge transfer methods. (Top) Generated images. (Bottom) KMMD and MV.

the baselines. BSA presents blur artifacts, which are probably caused by the mean square error used to optimize their model.

Multiple pre-trained models. We now evaluate the general case for MineGAN, where there is more than one pre-trained model to mine from. We start with two pre-trained Progressive GANs: one on *Cars* and one on *Buses*, both from the LSUN dataset [183]. These pre-trained networks generate cars and buses of a variety of different colors. We collect a target dataset of 200 images (of 256×256 resolution) of *red vehicles*, which contains both red cars and red buses. We consider three target sets with different car-bus ratios (0.3:0.7, 0.5:0.5, and 0.7:0.3) which allows us to evaluate the estimated probabilities p_i of the selector. To successfully generate *all types* of red vehicle, knowledge needs to be transferred from both pre-trained models.

Fig. 3.7 shows the synthesized images. As expected, the limited amount of data makes training from scratch result in overfitting. TransferGAN [173] produces only high-quality output samples for one of the two classes (the class that coincides with the pre-trained model) and it cannot extract knowledge from both pre-trained GANs. On the other hand, MineGAN generates high-quality images by successfully transferring the knowledge from both source domains simultaneously. Table 3.1 (top

Method	→ Red vehicle	→ Tower	→ Bedroom
Scratch	190 / 185 / 196	176	181
TransferGAN (car)	76.9 / 72.4 / 75.6	-	-
TransferGAN (bus)	72.8 / 71.3 / 73.5	-	-
TransferGAN (livingroom)	-	78.9	65.4
TransferGAN (church)	-	73.8	71.5
MineGAN (w/o FT)	67.3 / 65.9 / 65.8	69.2	58.9
MineGAN	61.2 / 59.4 / 61.5	62.4	54.7
Estimated p_i			
Car	0.34 / 0.48 / 0.64	-	-
Bus	0.66 / 0.52 / 0.36	-	-
Living room	-	0.07	0.45
Kitchen	-	0.06	0.40
Bridge	-	0.42	0.08
Church	-	0.45	0.07

Table 3.1 – Results for {Car, Bus} → Red vehicles with three different target data distributions (ratios cars:buses are 0.3:0.7, 0.5:0.5 and 0.7:0.3) and {Living room, Bridge, Church, Kitchen} → Tower/Bedroom. (Top) FID scores between real and generated samples. (Bottom) Estimated probabilities p_i for each model.

rows) quantitatively validates that our method outperfroms TransferGAN with a significantly lower FID score. Furthermore, the probability distribution predicted by the selector, reported in Table 3.1 (bottom rows), matches the class distribution of the target data.

To demonstrate the scalability of MineGAN with multiple pre-trained models, we conduct experiments using four different generators, each trained on a different LSUN category including *Livingroom*, *Kitchen*, *Church*, and *Bridge*. We consider two different off-manifold target datasets, one with *Bedroom* images and one with *Tower* images, both containing 200 images. Table 3.1 (left-bottom rows) again shows that our method obtains significantly better FID scores even when we choose the most relevant pre-trained GAN to initialize training for TransferGAN. Table 3.1 (right-bottom rows) shows that the miner identifies the relevant pre-trained models, e.g. transferring knowledge from *Bridge* and *Church* for the target domain *Tower*. Finally, Fig. 3.7 (right) provides visual examples.

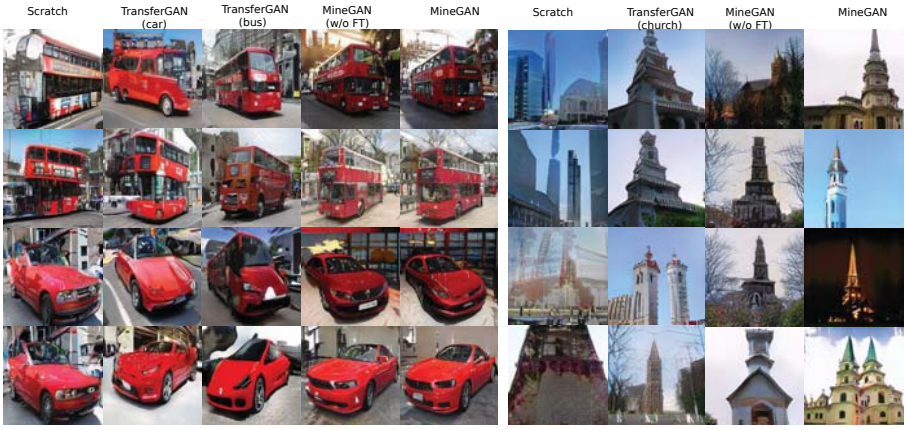


Figure 3.7 – Results: {car, bus} → red vehicles (left) and {Living room, Bridge, Church, Kitchen} → Tower (right). Based on pre-trained Progressive GAN. For TransferGAN we show the pre-trained model between parentheses. More examples in Appendix A.2.4.

3.4.2 Knowledge transfer from conditional GANs

Here we transfer knowledge from a pre-trained *conditional GAN* (see Section 3.3.3). We use BigGAN [19], which is trained using ImageNet [146], and evaluate on two target datasets: on-manifold (ImageNet: *cock*, *tape player*, *broccoli*, *fire engine*, *harvester*) and off-manifold (Places365 [202]: *alley*, *arch*, *art gallery*, *auditorium*, *ballroom*). We use 500 images per category. We compare MineGAN with training from scratch, TransferGAN [173], and two iterative methods: DGN-AM [120] and PPGN [119] [5]. It should be noted that both DGN-AM [120] and PPGN [119] are based on a less complex GAN (equivalent to DCGAN [132]). Therefore, we expect these methods to exhibit results of inferior quality, and so the comparison here should be interpreted in the context of GAN quality progress. However, we would like to stress that both DGN-AM and PPGN do not aim to transfer knowledge to new domains. They can only generate samples of a particular class of a pre-trained classifier network, and they have no explicit loss ensuring that the generated images follow a target distribution.

Fig. 3.8 shows qualitative results for the different methods. As in the unconditional case, MineGAN produces very realistic results, even for the challenging off-manifold case.

[5] We were unable to obtain satisfactory results with BSA [122] in this setting (images suffered from blur artifacts) and have excluded it here.

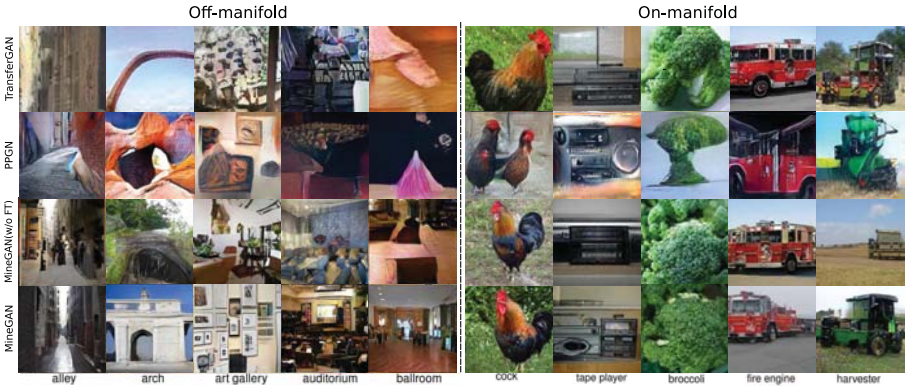


Figure 3.8 – Results for conditional GAN. (Left) Off-manifold (ImageNet→Places365). (Right) On-manifold (ImageNet→ImageNet).

Table 3.2 presents quantitative results in terms of FID and KMMD. We also indicate whether each method uses the label of the target domain class. Our method obtains the best scores for both metrics, despite not using target label information. PPGN performs significantly worse than our method. TransferGAN has a large performance drop for the off-manifold case, for which it cannot use the target label as it is not in the pre-trained GAN (see [173] for details).

Another important point regarding DGN-AM and PPGN is that each image generation during inference is an iterative process of successive backpropagation updates until convergence, whereas our method is feedforward. For this reason, we include in Table 3.2 the inference running time of each method, using the default 200 iterations for DGN-AM and PPGN. All timings have been computed with a CPU Intel Xeon E5-1620 v3 @ 3.50GHz and GPU NVIDIA RTX 2080 Ti. We can clearly observe that the feedforward methods (TransferGAN and ours) are three orders of magnitude faster despite being applied on a more complex GAN [19].

3.5 Conclusions

We presented a model for knowledge transfer for generative models. It is based on a mining operation that identifies the regions on the learned GAN manifold that are closer to a given target domain. Mining leads to more effective and efficient fine tuning, even with few target domain images. Our method can be applied to single and multiple pre-trained GANs. Experiments with various GAN architectures (BigGAN, Progressive GAN, and SNGAN) on multiple datasets demonstrated its

Method	Off-manifold		On-manifold		Time (ms)
	Label	FID/KMMD	Label	FID/KMMD	
Scratch	No	190 / 0.96	No	187 / 0.93	5.1
TransferGAN	No	89.2 / 0.53	Yes	58.4 / 0.39	5.1
DGN-AM	Yes	214 / 0.98	Yes	180 / 0.95	3020
PPGN	Yes	139 / 0.56	Yes	127 / 0.47	3830
MineGAN (w/o FT)	No	82.3 / 0.47	No	61.8 / 0.32	5.2
MineGAN	No	58.4 / 0.41	No	52.3 / 0.25	5.2

Table 3.2 – Distance between real data and generated samples as measured by FID score and KMMD value. The off-manifold results correspond to ImageNet → Places365, and the on-manifold results correspond to ImageNet → ImageNet. We also indicate whether the method requires the target label. Finally, we show the inference time for the various methods in milliseconds.

effectiveness. Results showed that we outperform previous approaches, including TranferGAN [173] and BSA [122]. Finally, we demonstrated that MineGAN can be used to transfer knowledge from multiple domains.

Image-to-image translation Part II

How do we address problems of image-to-image translation [1]

4 Mix and match networks: cross-modal alignment for zero-pair image-to-image translation [1]

4.1 Introduction

For many computer vision applications, the task is to estimate a mapping between an input image and an output image. This family of methods is often known as image-to-image translations (image translations hereinafter). They include transformations between different modalities, such as from RGB to depth [99], or domains, such as luminance to color images [196], or editing operations such as artistic style changes [49]. These mappings can also include other 2D representations such as semantic segmentations [106] or surface normals [41]. One drawback of the initial research on image translations is that the methods required paired data to train the mapping between the domains [41, 70, 106]. Another class of algorithms, based on cycle consistency, address the problem of mapping between unpaired domains [81, 181, 206]. These methods are based on the observation that translating from one domain to another and translating back to the original domain should result in recovering the original input image.

The discussed approaches consider translations between two domains which are either paired or unpaired. However, for many real-world applications there exist both paired and unpaired domains simultaneously. Consider the case of image translation between multiple modalities, where for some of them we have access to aligned data pairs but not for all modalities. The aim would then be to exploit the knowledge from the paired modalities to obtain an improved mapping for the unpaired modalities. An example of such a translation setting is the following: you have access to a set of RGB images and their semantic segmentation, and a (different) set of RGB images and their corresponding depth maps, but you are interested in obtaining a mapping from depth to semantic segmentation (see Figure 4.1). We call this the *unseen* translation because we do not have pairs for this translation, and we refer to this setting as *zero-pair translation*. Zero-pair translation is typically

[1] This chapter is under review on the international journal of computer vision (minor revision) [171], which is a extended version of the published paper at the Computer Vision and Pattern Recognition Conference (CVPR 2018).

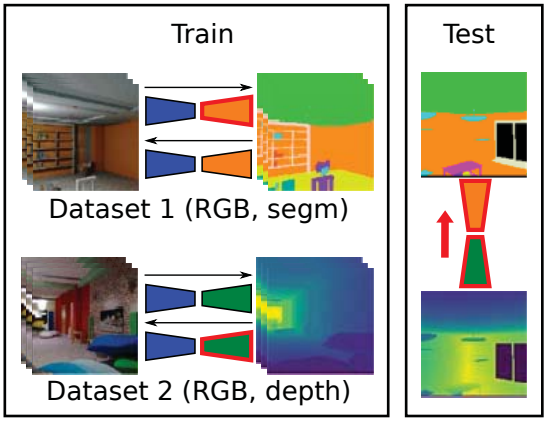


Figure 4.1 – Overview of mix and match networks (M&MNets) and zero-pair translation. Two disjoint datasets are used to train seen translations between RGB and segmentation and between RGB and depth (and vice versa). We want to infer the unseen depth-to-segmentation translation (i.e. *Zero-pair translation*). The M&MNets approach builds the unseen translator by simply cascading the source encoder and target decoder (i.e. depth and segmentation, respectively). Best viewed in color.

desired when we extend an experimental setup with an additional camera in another modality. We now would like to immediately exploit this new sensor without the cost of labelling new data. In this work, we provide a new approach to address the zero-pair translation problem.

We propose a new method, which we call *mix and match networks*, which addresses the problem of learning a mapping between unpaired modalities by seeking alignment between encoders and decoders via their latent spaces[2]. The translation between unseen modalities is performed by simply concatenating the source modality encoder and the target modality decoder (see Figure 4.1). The success of the method depends on the alignment of the encoder and decoder for the unseen translation. We study several techniques that contribute to achieve alignment, including the usage of autoencoders, latent space consistency losses and the usage of robust side information to guide the reconstruction of spatial structure.

We evaluate our approach in a challenging cross-modal task, where we perform zero-pair depth to semantic segmentation translation (or semantic segmentation to depth translation), using only RGB-depth and RGB-semantic segmentation pairs

[2]The code is available online at http://github.com/yaxingwang/Mix-and-match-networks.

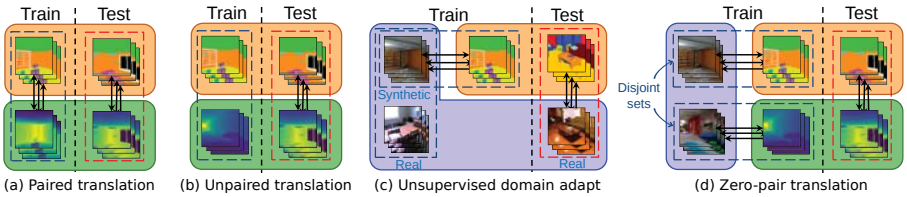


Figure 4.2 – cross-modal translation train and test settings: (a) paired translation, (b) unpaired translation, (c) unsupervised domain adaptation for segmentation (two modalities and two domains in the RGB modality), (d) zero-paired translation (three modalities). Best viewed in color.

during training. Furthermore, we show that the results can be further improved by using pseudo-pairs between the unseen modalities that allow the network to exploit unseen shared information. We also show that our approach can be used for cross-modal translation and with unpaired data. In particular, we show that mix and match networks scale better with the number of modalities, since they are not required to learn all pairwise image translation networks (i.e. scales linearly instead of quadratically).

This chapter is an extended version of a previous conference publication [172]. We have included more analysis and insight about how mix and match networks exploit the information shared between modalities, and propose an improved mix and match networks framework with pseudo-pairs which allows us to access previously unexploited shared information between unseen modalities (see Section 4.5). This was found to significantly improve performance. In addition, [172] only report results on a synthetic dataset. Here we also provide results on real images (SUN RGB-D dataset [155]) and four modalities (Freiburg Forest dataset [165]). Furthermore, we have added more insights on how the alignments between encoders and decoders evolve during training.

4.2 Related work

In this section we discuss the literature of related research areas.

4.2.1 Image-to-image translation

Paired translations Generic encoder-decoder architectures have achieved impressive results in a wide range of transformations between images. [70] proposed *pix2pix*, which is a conditional generative adversarial network (conditional

GAN) [54, 115] trained with pairs of input and output images to learn a variety of image translations. Those translations include cross-domain image translations such as colorization and style transfer. [53] disentangle the information of the domains in the latent space, which allows to do cross-domain retrieval as well as perform one-to-many translations. The ability of GANs to generate realistic images also enables *pix2pix* to address effectively challenging cross-modal translations, such as semantic segmentation to RGB image. In this case, recent multi-scale architectures [24, 167] achieve better results in higher resolution images.

Unpaired translations Various works extended image translation to the case where no explicit input-output image pairs are available (*unpaired* image translation), using the idea of cyclic consistency [81, 98, 181, 206] or consistency between certain extracted features [158]. To avoid accidental artifacts and improve learning, [114] integrate an attention mechanism to help translations focus on semantically meaningful regions. [100] show that unsupervised mappings can be learned by imposing a joint latent space between the encoder and the decoder. Both TransGaGa [176] and TraVeLGAN [6] address the issues of image translation across large geometry variations. The former disentangles image space in a Cartesian product of the appearance and the geometry latent spaces, and the latter considers a Siamese network to replace the cycle-consistency constraint.

In this work, we consider the case where paired data is available between some modalities and not available between others (i.e. zero-pair), and how the knowledge can be transferred to those unseen translations. Whereas previous work has focused on unpaired domains of the same modality, we show results for unpaired domains of different modalities.

Diversity in translations Given an input image (e.g. an edge image or a grayscale image) there are often multiple possible solutions (e.g. different plausible colorizations). The paired translation framework was extended to one-to-many translations in the work of [207]. DRIT [93], MUNIT [68] and Augmented CycleGAN [5] can learn one-to-many translations in unpaired settings. In general, disentangled representations allow achieving diversity by keeping the content component and sampling the style component of the latent representation [53, 93, 112]. [30] propose a novel group-wise deep whitening-and-coloring method to improve computational efficiency. [4] scale the latent filter to avoid a complicated network framework to perform one-to-many translations.

Multi-domain translations We also consider the case of multiple domains (and modalities). In concurrent work, [31] also address scaling to multiple domains by using a single encoder-decoder model, which was previously explored by [130]. [27] effectively disentangle the intermediate states between source and target domains. [170] perform diverse and scalable image transfer by a single model. These works

focus on faces and changing relatively superficial and localized attributes such as make-up, hair color, gender, etc., always within the RGB modality. In contrast, our approach uses multiple cross-aligned modality-specific encoders and decoders, which are necessary to address the deeper structural changes required by our cross-modal setting. [8] also use multiple encoders-decoders but focus on the easier cross-domain task of style transfer.

4.2.2 Semantic segmentation and depth estimation

Semantic image segmentation aims at assigning each pixel to an object class. [106] propose fully convolutional networks (FCN), following an encoder-decoder structure. Since the FCN shows outstanding performance, this paradigm has been adopted in many current methods for semantic segmentation [12, 23, 141, 182, 199]. Of particular interest is Segnet [12], which we adapt in our method. Segnet introduces the use of pooling indices instead of copying encoder features (i.e. skip connections, as in U-Net [141]). We also consider pooling indices in our architecture for zero-pair image translation because we found them to be more robust and invariant under unseen translations.

Depth estimation aims at estimating the depth structure of a RGB image, usually represented as a depth map encoding the distance of each pixel to the camera. Most depth estimation methods are formalized as regression problems, where the aim is to minimize the mean squared error (MSE) with respect to a ground truth depth map. In general, an encoder-decoder architecture is used, often incorporating multiscale networks and skip connections [41, 80, 86, 88, 99, 143, 166].

Multi-modal encoder-decoders With the development of multi-sensor cameras and datasets [87, 152, 155], encoder-decoder architectures have been adapted to multi-modal inputs [118], where different modalities (e.g. RGB, depth, infrared, surface normals) are encoded and combined prior to the decoding. The network is trained to perform tasks such as multi-modal object recognition [29, 42, 155], scene recognition [155, 156], object detection [57] (with simple classifiers or regressors as decoders in these cases) and semantic segmentation [78, 152, 168]. Similarly, multi-task learning can be applied to reconstruct multiple modalities [41, 78]. For instance [41] estimate depth, surface normals and semantic segmentation from a single RGB image, which can be seen as cross-modal translation.

Training a multi-task multimodal encoder-decoder network was recently studied in [85]. They use a joint latent representation space for the various modalities. In our work we consider the alignment and transferability of pairwise image translations to unseen translations, rather than joint encoder-decoder architectures. Another multimodal encoder-decoder network was studied in [21]. They show that

multi-modal autoencoders can address the depth estimation and semantic segmentation tasks simultaneously, even in the absence of some of the input modalities. All these works do not consider the zero-pair image translation problem addressed in this work.

4.2.3 Zero-shot recognition

In conventional supervised image recognition, the objective is to predict the class label that is provided during training. However, this poses limitations in scalability to new classes, since new training data and annotations are required. In zero-shot learning [3, 46, 89, 178, 179], the objective is to predict an unknown class for which there is no image available, but a description of the class (i.e. *class prototype*) or any other source of semantic similarity with seen classes. This description can be a set of attributes(e.g. has wings, blue, four legs, indoor) [72, 89], concept ontologies [44, 139] or textual descriptions [133]. In general, an intermediate semantic space is leveraged as a bridge between the visual features from seen classes and class description from unseen ones. In contrast to zero-shot recognition, we focus on unseen translations (unseen input-output pairs rather than simply unseen class labels).

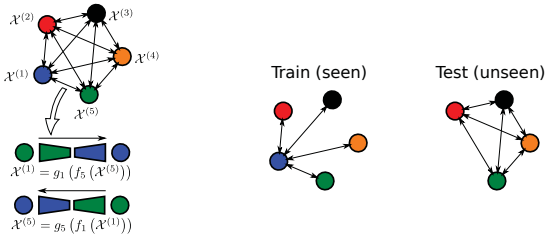
4.2.4 Zero-pair language translation

Evaluating models on unseen language pairs has been studied recently in machine translation [28, 45, 75, 201]. Johnson et al. [75] proposed a neural language model that can translate between multiple languages, even pairs of language where no explicit paired sentences where provided[3]. In their method, the encoder, decoder and attention are shared. In our method we focus on images, which are essentially a radically different type of data, with two dimensional structure in contrast to the sequential structure of language.

4.2.5 Domain adaptation

A related line of research is unsupervised domain adaptation. In that case the task is to transfer knowledge from a supervised source domain to an unsupervised target domain (see Figure 4.2c). This problem has been addressed by finding domain invariant feature spaces [47, 52, 162], using image translation models to map between source and target domain [177], and exploiting pseudo-labels [147, 209].

[3]Note that [75] refers to this as *zero-shot* translation. In this work we refer to this setting as zero-pair to emphasize that what is unseen is paired data and avoid ambiguities with traditional zero-shot recognition which typically refers to unseen samples.



(a) All seen (b) Seen and unseen

Figure 4.3 – Multi-domain image translation using pairwise translations: (a) all translations are seen during training, and (b) our setting: some translations are seen, then test on unseen. Best viewed in color.

Knowledge can also be transferred across modalities [22, 57, 62, 63]. For instance, Gupta *et al.* [57] use cross-modal distillation to learn depth models for classification by distilling RGB features (from pretrained model trained on a much larger RGB dataset), through a large set of unlabeled RGB-D pairs. Modality adaptation can also be achieved using cross-modal translation [180, 194].

When comparing this line of research with the setting we consider in this work (i.e. zero-pair translation) there are some important differences. The unsupervised domain adaptation setting (see Figure 4.2c) typically involves two modalities (i.e. RGB and segmentation), and two domains within the RGB modality (e.g. synthetic and real). Paired data is available only for the synthetic-segmentation while the synthetic-real translation is unpaired, and the unseen translation is real-segmentation (with test paired data). In contrast, our setting (see Figure 4.2d) is more challenging involving three modalities, with one disjoint paired training set for each seen translation. In comparison, using paired data to tackle domain shift allows us to reach much larger and challenging domain shifts and even modality shifts, a setting which, to the best of our knowledge, is not considered in domain adaptation literature.

4.3 Multi-modal image translations

We consider the problem of image translation between multiple modalities In particular, a translation from a source modality $\mathcal{X}^{(i)}$ to a target modality $\mathcal{X}^{(j)}$ is a mapping $T_{ij}: x^{(i)} \mapsto x^{(j)}$. This mapping is implemented as an encoder-decoder chain $x^{(j)} = T_{ij}\left(x^{(i)}\right) = g_j\left(f_i\left(x^{(i)}\right)\right)$ with source encoder f_i and target decoder g_j.

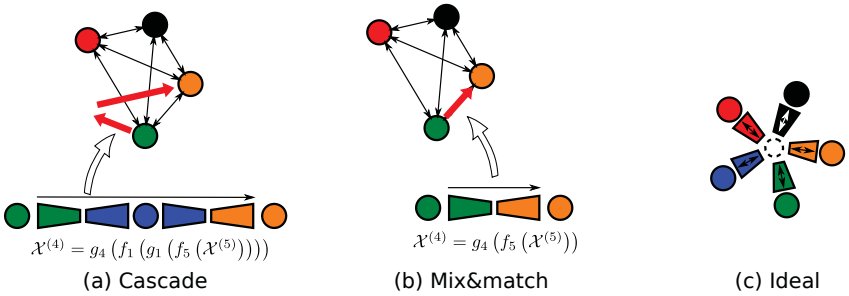


Figure 4.4 – Inferring unseen translations: (a) cascading translators, (b) mix and match networks (M&MNets), and (c) ideal case of encoders-decoders with aligned representations. Best viewed in color.

Translations between modalities connected during training are all learned jointly, and in both directions. Note that the encoder and decoder of translation T_{ij} are different from those of T_{ji}. In order to perform any arbitrary translation between modalities, all pairwise translations must be trained (i.e. seen) during the training stage (see Figure 4.3).

In this article we address the case where only a subset of the translations are seen during training, while the rest remain unseen (see Figure 4.3(b)). Our objective is to be able to infer these unseen translations during test time.

4.3.1 Inferring unseen translations

In the case where some of the translations are unseen during training, we could still try to infer them by reusing the available networks. Here we discuss two possible ways: *cascading translators*, which we use as baseline, and the proposed *mix and match networks* approach.

Cascaded translators Assuming there is a path of seen translations between the source modality and the target modality via intermediate modalities (see Figure 4.3(b)), a possible solution is simply concatenating the seen translators across this path. This will result in a mapping from the source to the target modality by reconstructing images in the intermediate modalities (see Figure 4.4(a)). However, the success of this approach depends on the effectiveness of the intermediate translators.

Unpaired translators An alternative is to frame the problem as unpaired translation between the source and target modalities and disregard the other modalities, learning a translation using methods based on cycle consistency [81, 100, 181, 206].

This approach requires training an unpaired translator per unseen translation. In general, unpaired translation can be effective when the translation is within the same modality and involves a relatively small shift between source and target domains (e.g. body texture in horse-to-zebra), but struggles in the more challenging case of cross-modal translations.

Mix and match networks (M&MNets) We propose to obtain the unseen translator by simply concatenating the encoder of the source modality and the decoder of the target modality (see Figure 4.4(b)). The problem is that these two networks have not directly interacted during training, and therefore, for this approach to be successful, the two latent spaces must be aligned.

4.3.2 Aligning for unseen translations

The key challenge in M&MNets is to ensure that the latent representation from the encoders can be decoded by all decoders, including those unseen (see Figure 4.4(c)). In order to address this challenge, encoders and decoders must be aligned in their latent representations. In addition, the encoder-decoder pair should be able to preserve the spatial structure, even in unseen translations.

In the following we describe the different techniques we use to enforce feature alignment between unseen encoder-decoder pairs.

Shared encoders and decoders Sharing encoders and decoders is a basic requirement to reuse latent representations and reduce the number of networks.

Autoencoders We jointly train modality-specific autoencoders with the image translation networks. By sharing the weights between the auto-encoders and the image translation encoder-decoder pairs the latent space is forced to align.

Robust side information In general, image translation tasks result in output images with similar spatial structure as the input ones, such as scene layouts, shapes and contours that are preserved across the translation. In fact, this spatial structure available in the input image is critical to simplify the problem and achieve good results, especially in cross-modal translations. Successful image translation methods often use multi-scale intermediate representations from the encoder as side information to guide the decoder in the upsampling process. Examples of side information are skip connections [58, 141] and pooling indices [12, 97]. We exploit side information in cross-modal translation (see discussion in Section 4.4.4).

Latent space consistency (only in paired settings) When paired data between some modalities is available, we can enforce consistency in the latent representations of each direction of the translations. [158] use L2 distance between a latent representation and the reconstructed after another decoding and encoding cycle. Here we enforce the representations $f_i\left(x^{(i)}\right)$ and $f_j\left(x^{(j)}\right)$ of two paired samples

$\left(x^{(i)}, x^{(j)}\right)$, to be aligned, since both images represent the same content (just in two different modalities). This is done by introducing a latent space consistency loss which is defined as $\left\| f_i\left(x^{(i)}\right) - f_j\left(x^{(j)}\right)\right\|_2$. We exploit this constraint in zero-pair image translation (see Section 4.4).

Adding noise to latent space The latent space consistency we apply is based on reducing the difference between the $f_i\left(x^{(i)}\right)$ and $f_j\left(x^{(j)}\right)$. The network can minimize this loss by aligning the representations of $f_i\left(x^{(i)}\right)$ and $f_j\left(x^{(j)}\right)$, but it could also minimize it by just reducing the signal $\left\| f_i\left(x^{(i)}\right)\right\|$ and $\left\| f_j\left(x^{(j)}\right)\right\|$. This would reduce the latent space consistency loss but not improve the alignment. Adding noise to the output of each encoder prevents this problem, since reducing the signal would then hurt the translation and auto-encoder losses. In practice, we found that adding noise helps to train the networks and improves the results during test.

4.3.3 Scalable image translation with M&MNets

As the number of modalities increases, the number of pairwise translations grows quadratically. Training encoder-decoder pairs for all pairwise translations in N modalities would require $N \times (N-1)/2$ encoders and $N \times (N-1)/2$ decoders (see Figure 4.3). One of the advantages of M&MNets is their better *scalability*, since many of those translations can be inferred without explicitly training them (see Figure 4.3(b)). It requires that each encoder and decoder should be involved in at least one translation pair during training in order to be aligned with the others, thereby reducing complexity from quadratic to linear with the number of modalities (i.e. N encoders and N decoders).

4.3.4 Translating domains instead of modalities

Although we described the proposed framework for cross-modal translation, the same framework can be easily adapted to cross-domain image translation. In that case, the modality is the same (typically RGB) and the translation is arguably less complex since the network does not need to learn to change the modality, just the domain. It can be learned sometimes with unpaired data (e.g. style transfer, face attributes and expressions).

Here we use cross-domain image translation to illustrate the scalability of M&MNets.The datasets (color and artworks) and the network architecture are provided in Appendix A.3.2. Figure 4.5 shows two examples involving multi-domain unpaired image translation. Figure 4.5a-b shows an image recoloring application with eleven domains ($N = 11$). Images are objects in the colored objects dataset [188], where we use colors as domains. A naive solution is training all pairwise recoloring

Figure 4.5 – Two examples of scalable inference of multi-domain translations with M&MNets. Color transfer (a-b): only transformations from blue or to blue (anchor domain) are seen. Style transfer (c-d): trained on four styles + photo (anchor) with data from [206]). From left to right: photo, Monet, van Gogh, Ukiyo-e and Cezanne. Input images are highlighted in red and seen translations in blue. Best viewed in color.

combinations with CycleGANs, which requires training a total of $N(N-1)/2 = 55$ encoders (and decoders). In contrast, M&MNets only require to train eleven encoders and eleven decoders, while still successfully addressing the recoloring task. In particular all translations from or to the blue domain are trained, while the remaining translations not involving blue are unseen. The input images (framed in red) and the resulting seen translations (framed in blue) are shown in Figure 4.5a. The additional images in Figure 4.5b correspond to the remaining unseen translations.

We also illustrate M&MNets in a style transfer setting with five domains. They include photo (used as anchor domain) and four artistic styles with data from [206]). M&MNets can reasonably infer unseen translations between styles (see Figure 4.5d) using only five encoders and five decoders (for a total of twenty possible translations). Note that the purpose of these examples is to illustrate the scalability aspect of M&MNets in multiple domains, not to compete with state-of-the-art recoloring or style transfer methods.

4.4 Zero-pair cross-modal translation

Well aligned M&MNets can be applied to a variety of problems. Here, we apply them to a challenging setting we call *zero-pair cross-modal translation*, which involves three modalities[4]. Note that cross-modal translations usually require modality-

[4]For simplicity, we will refer to the output semantic segmentation maps and depth as modalities rather than tasks, as done in some works.

specific architectures and losses.

4.4.1 Problem definition

We consider the problem of jointly learning two seen cross-modal translations: RGB-to-segmentation translation $y = T_{RS}(x)$ (and $x = T_{SR}(y)$) and RGB-to-depth translation $z = T_{RD}(x)$ (and $x = T_{DR}(z)$) and $x = T_{DR}(z)$) and evaluating on the unseen depth-to-segmentation transformations $y = T_{DS}(z)$ and $z = T_{SD}(y)$ (see Figures 4.1 and 4.2c). In contrast to the conventional unpaired translation setting, here seen translations have paired data (cross-modal translation is difficult to learn without paired samples). In particular, we consider the case where the former translations are learned from a semantic segmentation dataset $\mathscr{D}_{RS}$ with pairs $(x, y) \in \mathscr{D}_{RS}$ of RGB images and segmentation maps, and the second from a disjoint RGB-D dataset $\mathscr{D}_{RD}$ with pairs of RGB and depth images $(x, z) \in \mathscr{D}_{RD}$. Therefore no pairs with matching depth images and segmentation maps are available to the system. The system is evaluated on a third dataset $\mathscr{D}_{DS}$ with paired depth images and segmentation maps.

4.4.2 Mix and match networks architecture

The overview of the framework is shown in Figure 4.6. As basic building blocks we use three modality-specific encoders $f_R(x)$, $f_D(z)$ and $f_S(y)$ (RGB, depth and semantic segmentation, respectively), and the corresponding three modality-specific decoders $g_R(h)$, $g_D(h)$ and $g_S(h)$, where h is the latent representation in the shared space. The required translations are implemented as $y = T_{RS}(x) = g_S(f_R(x))$, $z = T_{RD}(x) = g_D(f_R(x))$ and $y = T_{DS}(z) = g_S(f_D(z))$.

Encoder and decoder weights are shared across the different translations involving same modalities (same color in Figure 4.6). To enforce better alignment between encoders and decoders of the same modality, we also include self-translations using the corresponding three autoencoders $T_{RR}(x) = g_R(f_R(x))$, $T_{DD}(y) = g_D(f_D(y))$ and $T_{SS}(z) = g_S(f_S(z))$.

We base our encoders and decoders on the SegNet architecture [12]. The encoder of SegNet itself is based on the 13 convolutional layers of the VGG-16 architecture [153]. The decoder mirrors the encoder architecture with 13 deconvolutional layers. Weights in encoders and decoders are randomly initialized following a standard Gaussian distribution except for the RGB encoder which is pretrained on ImageNet [35].

As in SegNet, pooling indices at each downsampling layer of the encoder are provided to the corresponding upsampling layer of the (seen or unseen) decoder[5].

[5]The RGB decoder does not use pooling indices, since in our experiments we observed undesired

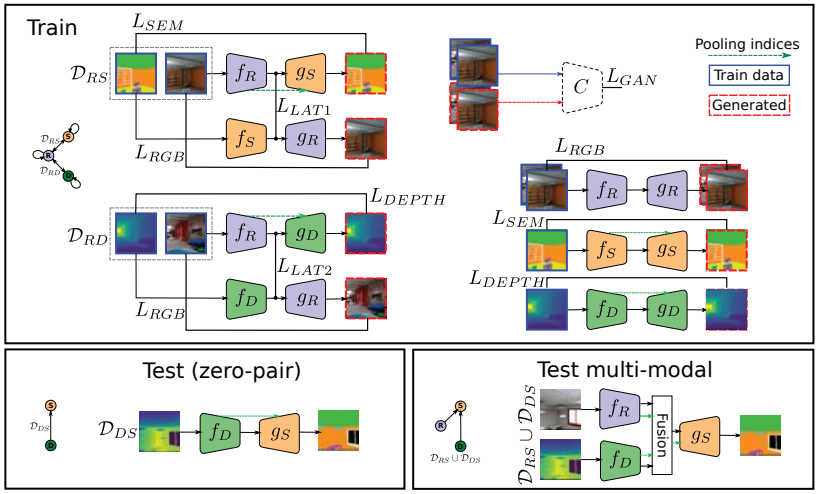


Figure 4.6 – Zero-pair cross-modal and multi-modalimage translation with M&MNets. Two disjoint sets $\mathscr{D}_{RS}$ and $\mathscr{D}_{RD}$ are seen during training, containing (RGB,depth) pairs and (RGB,segmentation) pairs, respectively. The system is tested on the unseen translation depth-to-segmentation (zero-pair) and (RGB+depth)-to-segmentation (multimodal), using a third unseen set $\mathscr{D}_{DS}$. Encoders and decoders with the same color share weights. Note that we do not apply pooling indices for RGB decoders. Best viewed in color.

These pooling indices seem to be relatively similar across the three modalities and effective to transfer spatial structure information that help to obtain better depth and segmentation boundaries in higher resolutions. Thus, they provide relatively modality-independent side information. We also experimented with skip connections and no side information at all.

4.4.3 Loss functions

As we mentioned before, for a correct cross-alignment between encoders and decoders, training is critical for zero-pair translation. The final loss combines a number of modality-specific losses for both cross-modal translation and self-translation (i.e. autoencoders) and alignment constraints in the latent space

$$L = \lambda_R L_{RGB} + \lambda_S L_{SEG} + \lambda_D L_{DEPTH} + \lambda_A L_{LAT}$$

grid-like artifacts in the RGB output when we use them.

where λ_R, λ_S, λ_D and λ_A are weights which balance the losses.

RGB We use a combination of pixelwise L2 distance and adversarial loss $L_{RGB} = \lambda_{L2} L_{L2} + L_{GAN}$. L2 distance is used to compare the ground truth RGB image and the output RGB image of the translation from a corresponding depth or segmentation image. It is also used in the RGB autoencoder

$$L_{L2} = \mathbb{E}_{(x,y)\sim\mathscr{D}_{RS}} \left[\left\| T_{SR}\left(y\right) - x \right\|_2 \right] \tag{4.1}$$
$$+ \mathbb{E}_{(x,z)\sim\mathscr{D}_{RD}} \left[\| T_{DR}\left(z\right) - x \|_2 \right] \tag{4.2}$$
$$+ \mathbb{E}_{x\sim\mathscr{D}_{RS}\cup\mathscr{D}_{RD}} \left[\| T_{RR}\left(x\right) - x \|_2 \right] \tag{4.3}$$

In addition, we also include the least squares adversarial loss [70, 109] on the output of the RGB decoder

$$L_{GAN} = \mathbb{E}_{x\sim\mathscr{D}_{RS}\cup\mathscr{D}_{RD}} \left[\left(C\left(x\right) - 1\right)^2 \right] + \mathbb{E}_{\hat{x}\sim\hat{p}(x)} \left[\left(C\left(\hat{x}\right)\right)^2 \right]$$

where $\hat{p}(x)$ is the resulting distribution of the combined images $\hat{x}$ generated by $\hat{x} = T_{SR}\left(y\right)$, $\hat{x} = T_{DR}\left(z\right)$ and $\hat{x} = T_{RR}\left(x\right)$. Note that the RGB autoencoder and the discriminator $C\left(x\right)$ are both trained with the combined RGB data $\mathscr{X}$.

Depth For depth we use the Berhu loss [88] in both RGB-to-depth translation and in the depth autoencoder

$$L_{DEPTH} = \mathbb{E}_{(x,z)\sim\mathscr{D}_{RD}} \left[\mathscr{B}\left(T_{RD}\left(x\right) - z\right) \right] \tag{4.4}$$
$$+ \mathbb{E}_{(x,z)\sim\mathscr{D}_{RD}} \left[\mathscr{B}\left(T_{DD}\left(z\right) - z\right) \right] \tag{4.5}$$

where $\mathscr{B}\left(z\right)$ is the average Berhu loss, which is given by

$$\mathscr{B}\left(z' - z\right) = \begin{cases} \left|\left(z' - z\right)\right| & \left|z' - z\right| \leqslant c \\ \frac{\left(z' - z\right)^2 + c^2}{2c} & \left|z' - z\right| > c \end{cases} \tag{4.6}$$

where $z' = T_{RD}\left(x\right)$, and $c = \frac{1}{5}\max_i\left(\left|z'_i - z_i\right|\right)$, where i indexes the pixels of each image.

Semantic segmentation For segmentation we use the average cross-entropy loss $\mathscr{CE}\left(\hat{y}, y\right)$ in both RGB-to-segmentation translation and the segmentation autoencoder

$$L_{SEM} = \mathbb{E}_{(x,y)\sim\mathscr{D}_{RS}} \left[\mathscr{CE}\left(T_{RS}\left(x\right), y\right) \right] \tag{4.7}$$
$$+ \mathbb{E}_{(x,y)\sim\mathscr{D}_{RS}} \left[\mathscr{CE}\left(T_{SS}\left(y\right), y\right) \right]. \tag{4.8}$$

Latent space consistency We enforce latent representations to remain close, in-

dependently of the encoder that generated them. In our case we have two latent space consistency losses

$$L_{LAT} = L_{LAT1} + L_{LAT2} \tag{4.9}$$

$$L_{LAT1} = \mathbb{E}_{(x,y)\sim\mathscr{D}_{RS}} \left[\left\| f_R(x) - f_S(y) \right\|_2 \right] \tag{4.10}$$

$$L_{LAT2} = \mathbb{E}_{(x,z)\sim\mathscr{D}_{RD}} \left[\left\| f_R(x) - f_D(z) \right\|_2 \right] \tag{4.11}$$

4.4.4 The role of side information

Spatial side information plays an important role in image translation, especially in cross-modal translation (e.g. semantic segmentation). Reconstructing images requires reconstructing spatial details. Side information from a particular encoder layer can provide helpful hints to the decoder about how to reconstruct the spatial structure at a specific scale and level of abstraction.

Skip connections Perhaps the most common type of side information connecting encoders and decoders comes in the form of *skip connections*, where the feature from a particular layer is copied and concatenated with another feature further in the processing chain. U-Net [141] introduced a widely used architecture in image segmentation and image translation where convolutional layers in encoder and decoder are mirrored and the feature of a particular encoding layer is concatenated with the feature with the corresponding layer at the decoder. It is important to observe that skip connections make the decoder heavily condition on the particular features of the encoder. This is not a problem in general because translations are usually seen during training and therefore latent representations are aligned. However, in our setting with unseen translations that conditioning is simply catastrophic, because the target decoder is only aware of the features in encoders from modalities seen during training. Otherwise, as in the case of an unseen encoder, the result is largely unpredictable.

Pooling indices The SegNet architecture [12] includes unpooling layers that leverage pooling indices from the mirror layers of the encoder. Pooling indices capture the locations of the maximum values in the input feature map of a max pooling layer. These locations are then used to guide the corresponding unpooling operation in the decoder, helping to preserve finer details. Note that pooling indices are more compact descriptors than encoder features from skip connections, and since the unpooling operation is not learned, pooling indices are less dependent on the particular encoder and therefore more robust for unseen translations.

4.5 Shared information between unseen modalities

4.5.1 Shared and modality-specific information

The information conveyed by the latent representation is key to perform image translation. Encoders extract this information from the input image and decoders use it to reconstruct the output image. In general, this latent representation can contain information shared between the source and target modalities, and information specific to each modality. In a setting where the same latent representation is used across multiple encoders and decoders, the latent representation must capture information about all input and output modalities.

We can represent modalities as circles, whose intersections represent shared information between them. Figure 4.7a represents the particular case of zero-pair cross-modal translation with three modalities (described in the previous section). Note that translators and autoencoders force the latent representation to capture both shared and modality-specific information. However, the better the information shared between modalities is captured in the latent representation, the more effective cross-modal translations are.

The framework described in Section 4.4.2 enables the inference of unseen translations via the anchor modality RGB, whose encoder and decoder are shared across the two seen translations. That is the only component that indirectly enforces alignment of depth and segmentation encoders and decoders. Therefore, the latent information used in the unseen translation is the one shared by the three modalities.

In contrast, the information shared between depth and segmentation that is not shared with RGB (the dashed region in Figure 4.7a) is not exploited during training by depth and segmentation encoders and decoders, because it is of no use to solve any of the seen translations. This makes inferred translations less effective because depth and segmentation encoders are ignoring potentially useful information that could improve translation to segmentation and depth, respectively. In this section we propose an extension of our basic framework that aims at explicitly enforcing alignment between unseen modalities in order to exploit all shared information between unseen modalities (see the highlighted region in Figure 4.7b). Since no training pairs between those modalities are available, that alignment requires to be between unpaired samples.

4.5.2 Exploiting shared information between unseen modalities

We adapt the idea of pseudo-labels, used previously in unsupervised domain adaptation [147, 209], to our zero-pair cross-modal setting. The main idea is that we would also like to train directly the encoder-decoder between the unseen modalities.

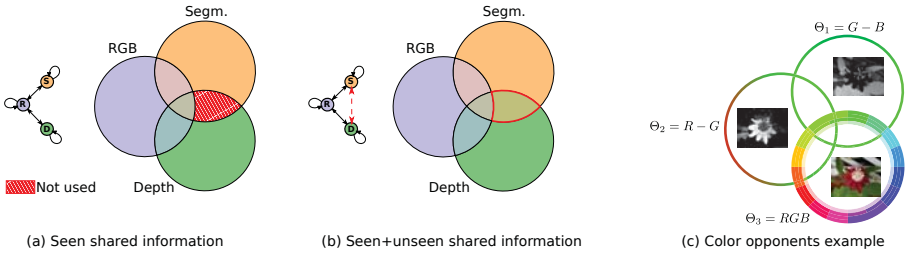


Figure 4.7 – Specific and shared information: (a) basic mix and match nets (see Fig 4.6) ignore depth-segmentation shared information, (b) extended mix and match net exploiting depth-segmentation shared information (unpaired information in our case), and (c) illustration using color opponents (trained on (Θ_1,Θ_2) and (Θ_1,Θ_3), and evaluated on unseen translation (Θ_2,Θ_3)). Best viewed in color.

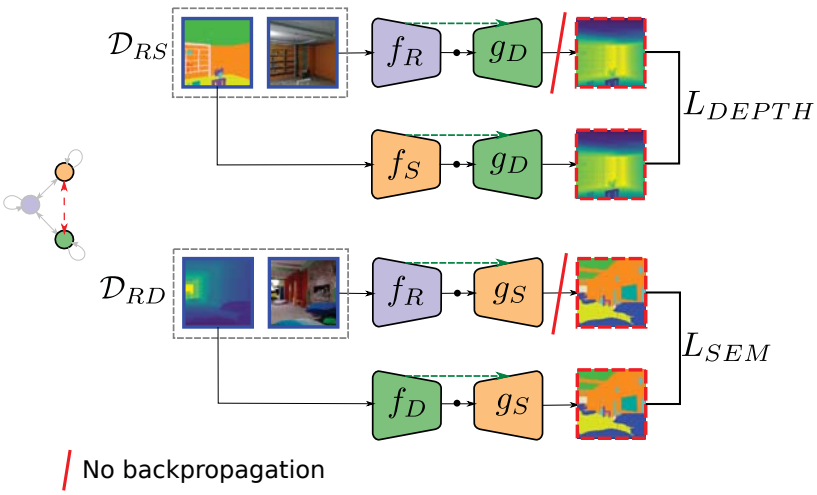


Figure 4.8 – Pseudo-pairs pipeline on the unseen translation. This pipeline is combined with the basic cross-modal M&MNets of Fig 4.6.

However, since we have no paired data between these modalities, we propose to use pseudo-pairs.

In our specific zero-pair cross-modal setting, recall we use x, y, and z to respectively indicate data from the the RGB, semantic segmentation and depth modality. We use the encoder-decoder networks between the seen modalities to form the pseudo-pairs $(T_{RD}(x), y)$ and $(T_{RS}(x), z)$. Now we can also train encoders and decoders between the unseen modalities depth and segmentation (see Figure 4.8) using the following loss:

$$L_{PP} = \mathbb{E}_{(x,y)\sim\mathscr{D}_{RS}}\left[\mathscr{B}\left(T_{RD}(x) - T_{SD}\left(y\right)\right)\right] \tag{4.12}$$

$$+ \mathbb{E}_{(x,z)\sim\mathscr{D}_{RD}}\left[\mathscr{CE}\left(T_{RS}(x), T_{DS}(z)\right)\right] \tag{4.13}$$

where $\mathscr{B}$ is the average Berhu loss [88], and $\mathscr{CE}$ is the cross-entropy loss. The direct training of the encoder-decoder between the unseen modality allows us to exploit correlation between features in these modalities for which no evidence exists in the RGB modality (dashed region in Figure 4.7a). In practice we first train the network without the pseudo-labels. After convergence we add L_{PP} and train further with all losses until final convergence.

Note that this additional term encourages the segmentation-to-depth and depth-to-segmentation translators to exploit this shared information between the unseen modalities, including the previously ignored one, in order to improve the translation to match the one obtained from RGB. The latter is more accurate since it has been trained with paired samples. A problem with this approach is that this new loss can harm the training of seen translations from RGB, since pseudo-labels are less reliable than true labels. For this reason we do not update the weights of the translators involving RGB with the pseudo-pairs (this is indicated with the red line in Figure 4.8).

4.5.3 Pseudo-pair example

To illustrate the potential of pseudo-pairs we consider a cross-domain image translation example where the not-used part between the unpaired domains (striped region in Figure 4.7) is expected to be substantial. We consider the task of estimating an RGB image from a single channel. In particular, we consider the following three domains[6]

$$\begin{aligned}\Theta_1 &= R - G\\ \Theta_2 &= G - B\\ \Theta_3 &= (R, G, B)\end{aligned} \tag{4.14}$$

[6]We choose the opponent channels because they are less correlated than the R,G and B channels [50].

Type	Method	Accuracy (%)
Seen	**Paired**	
	M&MNets $\Theta_1 \rightarrow \Theta_3$	75.0
Unseen	**Zero-pair**	
	M&MNets $\Theta_2 \rightarrow \Theta_3$	36.5
	M&MNets+PP $\Theta_2 \rightarrow \Theta_3$	**57.5**
Seen/unseen	**Multi-modal**	
	M&MNets $(\Theta_1, \Theta_2) \rightarrow \Theta_3$	77.5
	M&MNets + PP $(\Theta_1, \Theta_2) \rightarrow \Theta_3$	**80.5**

Table 4.1 – Flower classification accuracy obtained on Θ_3 computed for various image translation models. The importance of pseudo-pairs can be clearly seen.

where Θ_1 and Θ_2 are scalar images and Θ_3 is a three channel RGB image (see Figure 4.7(c)). Both domains Θ_1 and Θ_2 contain relevant and complementary information on estimating the RGB image.

For this experiment we use the ten most frequent classes of the Flower dataset [121] which are *passionflower*, *petunia*, *rose*, *wallflower*, *watercress*, *waterlily*, *cyclamen*, *foxglove*, *frangipani*, *hibiscus*. For training we have pairs (Θ_1, Θ_2) and (Θ_1, Θ_3) of non-overlapping images. For testing we use a separate test set. To evaluate the quality of the computed RGB images, we apply a flower classification algorithm on them and report the classification accuracy (See Appendix A.3.3).

The results are presented in Table 4.1. In the first two rows the result of M&MNets with and without pseudo-pairs are compared. The usage of pseudo-pairs results in a huge absolute performance gain of 21%. This shows that, for domains which have considerable amounts of complementary information, pseudo-pairs can significantly improve performance. In the next two rows, we have also included the multi-modal results. In this case the pseudo-pairs double the performance gain with respect to the paired domain (last row) from $77.5-75 = 2.5\%$ to $80.5-75 = 5.5\%$.

The qualitative results are provided in Figure 4.9. The results show the effectiveness of the pseudo-pairs. The method without the pseudo-pairs can only exploit information which is shared between the three domains. The domain Θ_1 contains information about the red-green color axes, and the mix and match nets (without pseudo-pairs) approach does partially manage to reconstruct that part (see first row Figure 4.9). However, Θ_1 has no access to the blue-yellow information which is encoded in the Θ_2. Adding the pseudo-pairs allows to exploit this information and the reconstructed RGB images are closer to the ground truth image (see second and third row Figure 4.9).

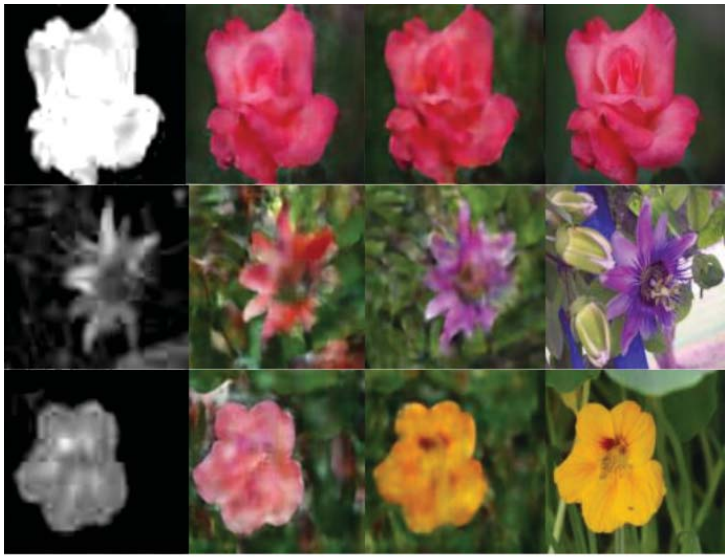


Figure 4.9 – Visualization of RGB image estimation in Flowers dataset. (a) input image from Θ_2 (via seen translation), (b) zero pair translation without pseudo-pairs [172], (c) zero pair with the pseudo-pairs (PP), (d) ground truth.

4.6 Experiments

In this section we demonstrate the effectiveness of M&MNets and their variants to address unseen translations in the challenging cross-modal translation setting involving the modalities RGB, depth and segmentation.

4.6.1 Datasets and experimental settings

We use two RGB-D datasets annotated with segmentation maps, one with synthetic images and the other with real captured images. A third dataset also includes near infrared (NIR) as a fourth modality.

SceneNet RGB-D The SceneNet RGB-D dataset [113] consists of 16865 synthesized train videos and 1000 test videos. Each of them contains 300 frames representing the same scene in a multi-modal triplet (i.e. RGB, depth and segmentation), with a size of 320x240 pixels. We collected 150K triplets for our train set, 10K triplets for our

validation set and 10K triplets for our test set. The triplets are sampled uniformly from the first frame to the last frame every 30 frames. The triplets for the validation set are collected from the remaining train videos and the test set is taken from the test dataset.

In order to evaluate zero-pair translation, we divided the train set (and validation set) into two equal non-overlapping splits from different videos (to avoid covering the same scenes). We discard depth images in one set and segmentation maps in the other, thus creating two disjoint training sets with paired instances, $\mathscr{D}_{RS}$ and $\mathscr{D}_{RD}$ respectively, to train our model.

SUN RGB-D The SUN RGBD dataset [155] contains 10335 real RGB-D images of room scenes. Each RGB image has a corresponding depth and segmentation map. We collected two sets: 10K triplets for the train set and 335 triplets for test set. For the train set, we split it into two disjoint subsets, one containing (RGB, segmentation) pairs, and the other containing (RGB, depth) pairs, each of them consisting of 5K pairs.

Freiburg Forest The Freiburg Forest dataset [165] consists of images of 1024×768. We crop images (RGB, depth, NIR and semantic segmentation) to 256×256. We consider five different semantic classes: *Sky*, *Trail*, *Grass*, *Vegetation* and *Obstacle*. Note we combine the *tree* and *vegatation* into an single class(*Vegetation*) as suggested in [165]. We use the train and test datasets splits provided by the authors.

Network training We use Adam [82] with a batch size of 6, using a learning rate of 0.0002. We set $\lambda_R = 1$, $\lambda_S = 100$, $\lambda_D = 10$, $\lambda_A = 1$, $\lambda_{L2} = 1$. We initially train the mix and match framework without autoencoders, without latent consistency losses, and without adding noise during the first 200K iterations. Then we freeze the RGB encoder, add the autoencoders, latent consistency losses and noise to the latent space, and for the following 200K iterations we use $\lambda_R = 10$, $\lambda_A = 10$, $\lambda_{L2} = 100$. We found that the network converges faster using a larger λ_A for the second stage. The noise is sampled from a Gaussian distribution with zero mean and a standard deviation of 0.5. For the variant with pseudo-pairs, in a third stage we include the pseudo-pair pipeline and the corresponding loss and train for another additional 100K iterations, using $\lambda_{PP} = 100$ and learning rate 0.00002. We experimentally found that the above setting also achieves outstanding performance on Freiburg Forest dataset. The network information is displayed in Appendix A.3.1.

Evaluation metrics Following common practice, for the segmentation modality we compute the intersection over union (IoU) and per-class average (mIoU), and the global scores, which gives the percentage of correctly classified pixels. For the depth modality we also include quantitative evaluation, following the standard

error metrics for depth estimation [41]:

$$\begin{aligned} \delta < \nu &= \frac{1}{|y|} \sum_{y_i \in y} \left[\delta(y_i, y_i') < \nu \right] \\ \text{RMSE (linear)} &= \sqrt{\frac{1}{|y|} \sum_{y_i \in y} \left\| y_i - y_i' \right\|^2} \\ \text{RMSE (log)} &= \sqrt{\frac{1}{|y|} \sum_{y_i \in y} \left\| \log y_i - \log y_i' \right\|^2} \end{aligned} \tag{4.15}$$

where y and y' are the predicted and ground truth depth images, $\delta(u, v) = \max(\frac{u}{v}, \frac{v}{u})$ and $[P]$ is the Iverson bracket which is 1 when P is true and 0 otherwise.

4.6.2 Experiments on SceneNet RGB-D

Ablation study

We first performed an ablation study on the impact of several design elements on the overall performance of the system. We use a smaller subset of SceneNet RGB-D based on 51K triplets from the first 1000 videos (selecting 50 frames from the first 1000 videos for train, and the first frame from another 1000 videos for test).

Side information We first evaluate the usage of side information from the encoder to guide the upsampling process in the decoder. We consider three variants: no side information, skip connections [141] and pooling indices [12]. The results in Table 4.2 show that skip connections obtain worse results than no side information at all. This is caused by the fact that side information makes the decoder(s) conditioned on the *seen* encoder(s). This is problematic for *unseen* translations because the features passed through skip connections are different from those seen by the decoder during training, resulting in a drop in performance. In contrast, pooling indices provide a significant boost over no side information. Although the decoder is still conditioned to the particular seen encoders, pooling indices seem to provide helpful spatial hints to recover finer details, while being more invariant to the particular input-output combination, and even generalizing to unseen ones.

Figure 4.10 illustrates the differences between these three variants in depth-to-segmentation translation. Without side information the network is able to reconstruct a coarse segmentation, but without further guidance it is not able to refine it properly. Skip connections completely confuse the decoder by providing unseen encoding features. Pooling indices are able to provide helpful hints about spatial structure that allows the unseen decoder to recover finer segmentation maps.

RGB pretraining We also compare training the RGB encoder from scratch and initializing with pretrained weights from ImageNet. Table 4.2 shows an additional gain of around 4% in mIoU when using the pretrained weights.

Given these results we perform all the remaining experiments initializing the RGB encoder with pretrained weights and use pooling indices as side information.

Side information	Pretrained	mIoU	Global
-	N	29.8%	61.6%
Skip connections	N	12.7%	50.1%
Pooling indices	N	43.2%	73.5%
Pooling indices	Y	**46.7%**	**78.4%**

Table 4.2 – Influence of side information and RGB encoder pretraining on the final results. The task is zero-shot depth-to-semantic segmentation in SceneNet RGB-D (51K).

AutoEnc	Latent	Noise	PP	mIoU	Global
N	N	N	N	6.48%	15.7%
Y	N	N	N	20.3%	49.4%
Y	Y	N	N	45.8%	76.9%
Y	Y	Y	N	46.7%	78.4%
Y	Y	Y	Y	**49.2%**	**80.5%**

Table 4.3 – Impact of several components (autoencoder, latent space consistency loss, noise and pseudo-pairs) in the performance. The task is zero-pair depth-to-segmentation in SceneNet RGB-D (51K). PP: pseudo-pairs.

Latent space consistency, noise and autoencoders We evaluate these three factors in Table 4.3. The results show that latent space consistency and the usage of autoencoders lead to significant performance gains; for both, the performance (in mIoU) is more than doubled. Adding noise to the output of the encoder results in a small performance gain. The results in Table 4.3 do not apply pooling indices for the RGB decoder (as also shown in Fig. 4.6). When we add pooling indices to our approach without pseudo-pairs, results drop from 46.7% to 42.4% in mIoU. This could be because we focus on unseen translations to depth or segmentation modalities, which do not include reconstructing the RGB modality. We believe that forcing the RGB decoder to use pooling indices to reconstruct RGB images lowers the efficiency of the latent representation to reconstruct depth or segmentation. Hence, we sacrifice some of the performance translating to the RGB modality to improve the results for depth and semantic segmentation.

Pseudo-pairs We also evaluate the impact of using pseudo-pairs to exploit shared information between unseen modalities. Table 4.3 shows a significant gain of almost 3% in mIoU and a more moderate gain in global accuracy.

In the following sections we use the SceneNet RGB-D dataset with 170K triplets.

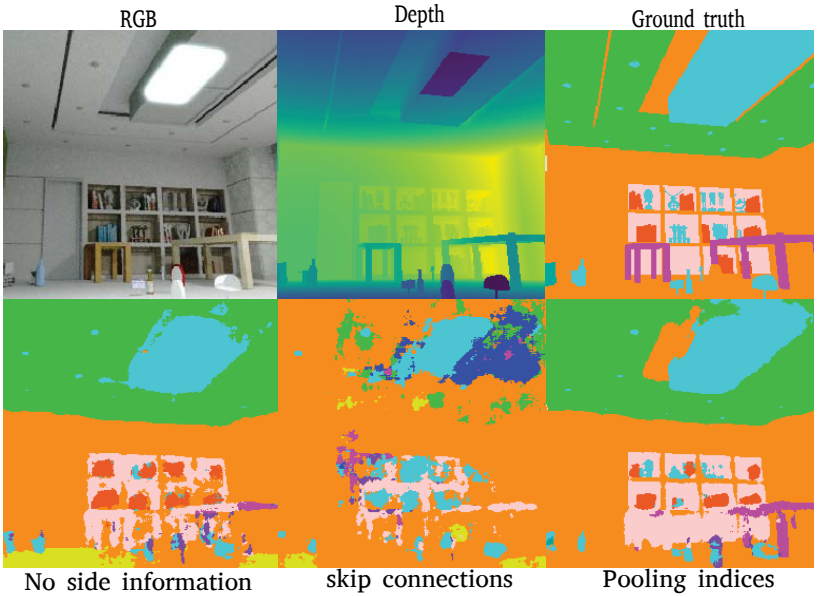


Figure 4.10 – Role of side information in unseen depth-to-segmentation translation in SceneNet RGB-D.

Monitoring alignment

The main challenge for M&MNets is to align the different modality-specific bottleneck features, in particular for unseen translations. We measure the alignment between the features extracted from the triplets in the test set $\mathscr{D}_{DS}$. For each triplet (x, y, z) (i.e. RGB, segmentation and depth images) we extract the corresponding triplet of latent features $(f_R(x), f_S(y), f_D(z))$ and measure their average pairwise cross-modal alignment. The alignment between RGB and segmentation features is measured using the following alignment factor

$$\mathrm{AF}_{RS} = E_{(x,y)\sim\mathscr{D}_{RS}}\left[\frac{f_R(x)^\intercal f_S(y)}{\|f_R(x)\|\,\|f_S(y)\|}\right] \tag{4.16}$$

The other alignment factors AF_{RD} and AF_{DS} between RGB and depth features and between depth and segmentation features are defined analogously. Figure 4.11 shows the evolution of this alignment during training and across the different stages. The three curves follow a similar trend, with the alignment increasing in

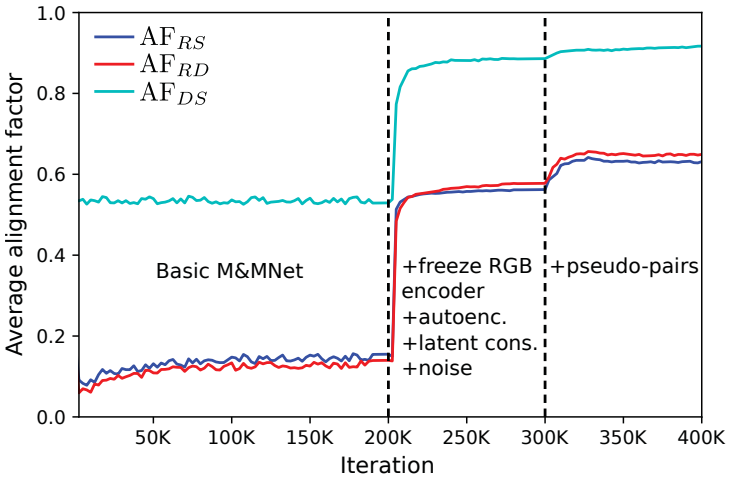


Figure 4.11 – Monitoring alignment between latent features on SceneNet RGB-D.

the first iterations of each stage and then stabilizing. The beginning of stage two shows a dramatic increase in the alignment, with a more moderate increase at stage three. These results are consistent with those of the ablation study of the previous section, showing that better alignment typically leads to better results in unseen translations. Overall, they show that latent space consistency, autoencoders, pseudo-pairs and pooling indices contribute to the effectiveness of M&MNets to address unseen image translation in the zero-pair setting.

Comparison with other models

In this section we compare M&MNets, and its variant with pseudo-pairs with several baselines:

- **CycleGAN**. We adapt CycleGAN [206] to learn a mapping from depth to semantic segmentation (and vice versa) in a purely unpaired setting. In contrast to M&MNets, this method only leverages depth and semantic segmentation, ignoring the available RGB data and the corresponding pairs (as shown in Figure 4.2a).
- **2×pix2pix**. We adapt pix2pix [70] to learn two cross-modal translations from paired data (i.e. $D \rightarrow R$ and $R \rightarrow S$). The architecture uses skip connections (which are effective in this case since both translations are seen) and the corresponding modality-specific losses. We adapt the code from [70]. In contrast to ours, it requires explicit decoding to RGB, which may degrade the

Method	Conn.	L_{SEM}	Bed	Book	Ceiling	Chair	Floor	Furniture	Object	Picture	Sofa	Table	TV	Wall	Window	mIoU	Global
Baselines																	
CycleGAN	SC	CE	2.79	0.00	16.9	6.81	4.48	0.92	7.43	0.57	9.48	0.92	0.31	17.4	15.1	6.34	14.2
2×pix2pix	SC	CE	34.6	1.88	70.9	20.9	63.6	17.6	14.1	0.03	**38.4**	10.0	4.33	67.7	20.5	25.4	57.6
StarGAN(unpaired)	PI	CE	6.71	1.42	17.6	6.21	13.2	1.25	8.51	0.52	12.8	3.24	4.28	9.52	8.57	7.21	10.7
StarGAN(paired)	PI	CE	9.70	2.56	18.4	5.70	15.7	0.41	9.20	1.56	14.2	5.02	3.56	14.7	11.4	8.62	14.1
M&MNets $D \rightarrow R \rightarrow S$	PI	CE	0.02	0.00	8.76	0.10	2.91	2.06	1.65	0.19	0.02	0.28	0.02	58.2	3.30	5.96	32.3
M&MNets $D \rightarrow R \rightarrow S$	SC	CE	25.4	0.26	82.7	0.44	56.6	6.30	23.6	5.42	0.54	21.9	10.0	68.6	19.6	24.7	59.7
Zero-pair																	
M&MNets $D \rightarrow S$	PI	CE	50.8	18.9	89.8	31.6	**88.7**	48.3	44.9	62.1	17.8	**49.9**	51.9	86.2	79.2	55.4	80.4
M&MNets+PP $D \rightarrow S$	PI	CE	52.1	29.0	88.6	32.7	86.9	**66.9**	48.4	76.6	25.1	45.5	58.8	88.5	**82.0**	60.1	82.2
Multi-modal																	
M&MNets $(R,D) \rightarrow S$	PI	CE	49.9	25.5	88.2	31.8	86.8	56.0	45.4	70.5	17.4	46.2	57.3	87.9	79.8	57.1	81.2
M&MNets+PP $(R,D) \rightarrow S$	PI	CE	**53.3**	**35.7**	**89.9**	**37.0**	88.6	59.3	**55.8**	**76.9**	25.7	46.6	**69.6**	**89.5**	80.0	**62.2**	**83.5**
Oracle																	
$D \rightarrow S$	PI	CE	53.7	31.0	89.1	31.4	88.2	66.8	52.7	78.4	25.7	47.4	59.3	89.7	82.2	61.2	83.4
$(R,D) \rightarrow S$	PI	CE	58.4	40.8	91.3	41.6	90.7	61.5	57.6	80.9	36.8	51.6	72.6	88.4	83.1	65.7	84.0

Table 4.4 – Zero-pair depth-to-segmentation translation on SceneNet RGB-D. **SC**: skip connections, **PI**: pooling indexes, **CE**: cross-entropy, **PP**: pseudo-pairs. x

quality of the prediction.

- **StarGAN**. We consider two adaptations of the StarGAN [31]. Both versions share the layers of the network for all modalities except for the first layer of the encoder and the last layer of of decoder which are modality-specific layers. This is required since modalities vary in the number of channels. The first version, called *StarGAN(unpaired)*, uses the losses originally proposed in [31]. We also implement a version which exploits the paired data, which we call *StarGAN(paired)*. For this version, we removed the cycle consistency (which is not required for paired modalities). We found this to slightly improve results.
- $D \rightarrow R \rightarrow S$ is similar to 2×pix2pix but with the architecture used in M&MNets. We train a translation model from depth to RGB and from RGB to segmentation, and obtain the transformation depth-to-segmentation by concatenating them. Note that it also requires translating to intermediate RGB images.
- $S \rightarrow R \rightarrow D$ is analogous to the previous baseline.
- **M&MNets** is the original mix and match networks [172].
- **M&MNets+PP** is the variant of M&MNets using pseudo-pairs.
- **Oracle** is the upper bound obtained by training a translation fully supervised with paired data.

Table 4.4 shows results for the different methods for depth-to-segmentation translation. CycleGAN is not able to learn a good mapping, showing the difficulty of unpaired translation to solve this complex cross-modal task. 2×pix2pix manages

(a) Input: depth (b) 2×Pix2pix (c) CycleGAN

(d) M&MNet (e) M&MNet(pseudo label) (f) Ground truth

Figure 4.12 – Zero-pair depth-to-segmentation translation on SceneNet RGB-D.

to improve the results by resorting to the anchor modality RGB, although still not satisfactory since this sequence of translations does not enforce explicit alignment between depth and segmentation, and the first translation network may also discard information not relevant for the RGB task, but necessary for reconstructing the segmentation image (like in the "Chinese whispers"/telephone game). Also, both results for *StarGAN* show that this approach is unable to learn a good mapping between the unseen modalities.

M&MNets evaluated on ($D \rightarrow R \rightarrow S$) achieve a similar result as CycleGAN, but significantly worse than 2×pix2pix. However, when we run our architecture with skip connections we obtain results similar to 2×pix2pix. Note that in this setting translations only involve seen encoders and decoders, so skip connections function well. The direct combination ($D \rightarrow S$) with M&MNets outperforms all baselines significantly. The performance more than doubles in terms of mIoU. Results improve another 5% in mIoU when adding the pseudo-pairs during training.

Method	$\delta <$ 1.25	1.25^2	1.25^3	RMSE (lin)	RMSE (log)
Baselines					
CycleGAN	0.05	0.12	0.20	4.63	1.98
2×pix2pix	0.14	0.31	0.46	3.14	1.28
StarGAN(unpaired)	0.05	0.14	0.23	4.60	1.96
StarGAN(paired)	0.07	0.15	0.26	4.58	1.94
M&MNets $S \rightarrow R \rightarrow D$	0.15	0.30	0.44	3.24	1.24
Zero-pair					
M&MNets $S \rightarrow D$	0.33	0.42	0.59	2.80	0.67
M&MNets+PP $S \rightarrow D$	0.42	0.61	0.79	2.24	0.60
Multi-modal					
M&MNets $(R,S) \rightarrow D$	0.36	0.48	0.65	2.48	0.64
M&MNets+PP $(R,S) \rightarrow D$	**0.47**	**0.69**	**0.81**	**1.98**	**0.49**
Oracle					
$S \rightarrow D$	0.49	0.72	0.85	1.94	0.43
$(R,S) \rightarrow D$	0.51	0.76	0.90	1.79	0.29

Table 4.5 – Zero-pair segmentation-to-depth on SceneNet RGB-D.

Figure 4.12 shows a representative example of the differences between the evaluated methods. CycleGAN fails to recover any meaningful segmentation of the scene, revealing the difficulty to learn cross-modal translations without paired data. 2×pix2pix manages to recover the layout and coarse segmentation, but fails to segment medium and small size objects. M&MNets are able to obtain finer and more accurate segmentations.

Table 4.5 shows results when we test in the opposite direction from semantic segmentation to depth. The conclusions are similar as in previous experiment: M&MNets outperform both baseline methods on all five evaluation metrics. Figure 4.13 illustrates this case, showing how pooling indices are also key to obtain good depth images, compared with no side information at all. The variant with pseudo-pairs obtains the best results.

Multi-modal translation

Since features from different modalities are aligned, we can also use M&MNets for multi-modal translation. For instance, in the previous multi-modal setting, given the RGB and depth images of the same scene we can translate to segmentation. We simply combine both modality-specific latent features x and z using a weighted average $y = (1-\alpha)\,x + \alpha z$, where α controls the weight of each modality. We set $\alpha = 0.2$ and use the pooling indices from the RGB encoder (instead of those from depth

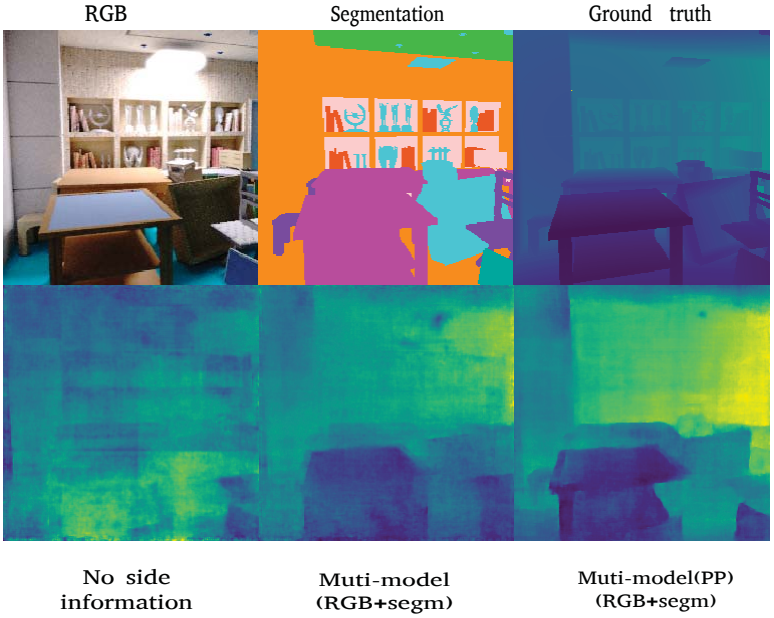


Figure 4.13 – Zero-pair and multimodal segmentation-to-depth on SceneNet RGB-D.

encoder). The resulting feature y is then decoded using the segmentation decoder. We proceed analogously to translation from RGB and segmentation to depth. The results in Table 4.4 and Table 4.5 show that this multi-modal combination further improves the performance of zero-pair translation, as the example in Figure 4.13 illustrates.

4.6.3 Experiments on SUN RGB-D

The previous results were obtained on the SceneNet RGB-D dataset which consists of synthetic images. Here we also show that M&MNets can be effective for the more challenging dataset SUN RGB-D, which involves real images and more limited data. The results in Table 4.6 and Table 4.7 show that M&MNets consistently outperform the other baselines in both unseen translation directions, with the new variant with pseudo-pairs obtaining the best performance. Similarly, multi-modal translation further improves the performance. Figures 4.15 and 4.16 illustrate how the proposed methods can reconstruct more reliably the target modality, especially the finer

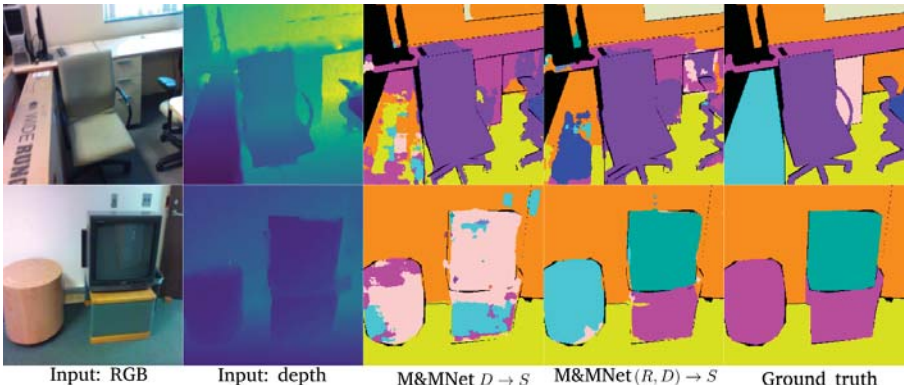


Figure 4.14 – Failure cases of the proposed framework on SUN RGB-D. See text for discussion.

details.

The results also show that the depth cue is insufficient to detect some of the classes such as *Book* and *TV*. The oracle results show that this is also the case when you have access to depth-semantic segmentation pairs. The results also show that our multi-modal results are biased towards RGB: this is reflected in the bad results which are obtained for the class *bed* which is well detected in the depth modality but not in the RGB modality, and also not by our multi-modal system. Examples of these cases are provided in Fig. 4.14.

4.6.4 Experiments on four modalities Freiburg Forest

As an example of zero-pair translation for an application with more than three modalities we perform experiments on the Freiburg Forest dataset which contains the RGB, depth, NIR and semantic segmentation modalities. For the training we use the settings used in the previous experiments, and add a $Berhu$ loss (see Eq. 4.5) for NIR in this experiment.

In the provided dataset all modalities are recorded for all scenes, however we consider that we have pairs for RGB and semantic segmentation, and we have a non-overlapping dataset of triplets for RGB, Depth, and NIR (see Figure 4.17). This scenario could be considered realistic. It reflects a situation where initially the robot only has a RGB camera, and labellers have provided semantic segmentation maps for these images. Then two additional sensors are added later to the robot, but no segmentation maps are available for this newly obtained multi-modal data.

Method	Conn.	L_{SEM}	Bed	Book	Ceiling	Chair	Floor	Furniture	Object	Picture	Sofa	Table	TV	Wall	Window	mIoU	Global
Baselines																	
CycleGAN	SC	CE	0.00	0.00	0.00	17.9	46.9	1.67	4.59	0.00	0.00	18.9	0.00	29.6	25.4	11.1	26.3
2×pix2pix	SC	CE	3.88	0.00	12.4	29.6	57.1	17.2	13.0	35.4	8.07	35.1	0.00	47.0	7.73	20.5	38.6
StarGAN(unpaired)	PI	CE	0.00	0.00	2.45	15.8	33.6	5.73	6.28	0.57	0.00	6.25	0.00	28.4	26.9	9.69	20.6
StarGAN(paired)	PI	CE	0.00	0.00	2.01	20.2	38.9	4.12	5.78	0.31	0.00	7.30	0.00	31.5	30.7	10.8	23.8
M&MNets $D \rightarrow R \rightarrow S$	PI	CE	0.00	0.00	0.00	17.0	39.4	0.52	0.01	0.00	0.01	12.2	0.00	31.0	5.19	8.12	22.8
M&MNets $D \rightarrow R \rightarrow S$	SC	CE	**39.9**	0.25	15.2	37.6	58.0	19.0	11.7	2.45	4.82	36.9	0.00	46.8	12.3	21.9	40.6
Zero-pair																	
M&MNets $D \rightarrow S$	PI	CE	28.4	2.90	22.6	41.9	71.6	14.1	25.1	17.8	**11.8**	49.7	0.08	64.2	15.5	28.1	51.8
M&MNets+PP $D \rightarrow S$	PI	CE	29.8	4.52	**28.5**	44.1	73.3	17.2	27.5	20.1	9.81	53.4	0.14	67.5	17.9	30.2	54.2
Multi-modal																	
M&MNets $(R, D) \rightarrow S$	PI	CE	0.00	16.6	21.4	**56.0**	72.1	24.2	28.3	38.1	**21.7**	57.0	64.6	68.0	43.7	39.4	58.8
M&MNets+PP $(R, D) \rightarrow S$	PI	CE	0.10	**19.3**	25.5	54.6	**74.6**	**25.6**	**30.1**	**42.4**	21.0	**58.1**	**65.2**	**69.0**	**49.7**	**41.1**	**59.8**
Oracle																	
$D \rightarrow S$	PI	CE	32.6	8.01	36.5	56.8	84.7	20.4	31.4	19.7	8.75	61.7	1.60	72.1	21.2	35.1	62.3
$(R, D) \rightarrow S$	PI	CE	0.13	21.2	26.4	56.2	78.9	26.9	35.2	44.4	23.2	60.2	67.3	71.2	52.3	43.3	62.5

Table 4.6 – Zero-pair depth-to-semantic segmentation on SUN RGB-D. **SC**: skip connections, **PI**: pooling indexes, **CE**: cross-entropy, **PP**: pseudo-pairs.

As we can see in Table 4.8, our method achieves the best scores. In the case of zero-pair setting (M&MNets $D \rightarrow S$, M&MNets $N \rightarrow S$, M&MNets+PP $D \rightarrow S$ and M&MNets+PP $N \rightarrow S$) the results obtain a large gap when compared to the baselines, clearly demonstrating the superiority of our method. For example, for $N \rightarrow S$ we obtain an increase of 22% over 2×pix2pix. The multi-modal results show that adding more modalities further increases results. Mainly, the performance on the category *obstacle* increases. Figure 4.18 shows representative examples of the different methods. The conclusions are similar to previous experiments: we effectively conduct cross-modal translation with zero-pair data and pseudo-labeling further improves the results.

4.7 Conclusions

We have introduced mix and match networks as a framework to perform image translations between unseen modalities by leveraging the knowledge learned from seen translations with explicit training data. The key challenge lies in aligning the latent representations in the bottlenecks in such a way that any encoder-decoder combination is able to perform effectively its corresponding translation. M&MNets have advantages in terms of scalability since only seen translations need to be trained. We also introduced zero-pair cross-modal translation, a challenging scenario involving three modalities and paired seen and unseen translations. In order

Method	$\delta <$ 1.25	1.25^2	1.25^3	RMSE (lin)	RMSE (log)
Baselines					
CycleGAN	0.06	0.13	0.24	4.80	1.57
2×pix2pix	0.13	0.34	0.59	3.80	1.30
StarGAN(unpaired)	0.06	0.12	0.22	5.04	1.59
StarGAN(paired)	0.07	0.15	0.27	4.60	1.55
M&MNets $S \rightarrow R \rightarrow D$	0.12	0.35	0.62	3.90	1.36
Zero-pair					
M&MNets $S \rightarrow D$	0.45	0.66	0.78	1.75	0.53
M&MNets+PP $S \rightarrow D$	0.49	0.77	0.90	1.42	0.37
Multi-modal					
M&MNets $(R,S) \rightarrow D$	0.53	0.80	0.92	1.63	0.35
M&MNets+PP $(R,S) \rightarrow D$	**0.56**	**0.83**	**0.93**	**1.33**	**0.34**
Oracle					
$S \rightarrow D$	0.61	0.88	0.97	1.20	0.30
$(R,S) \rightarrow D$	0.64	0.92	0.98	0.98	0.27

Table 4.7 – Zero-pair semantic segmentation-to-depth on SUN RGB-D.

to effectively address this problem, we described several tools to enforce the alignment of latent representations, including autoencoders, latent consistency losses, and robust side information. In particular, our results show that side information is critical to perform satisfactory cross-modal translations, but conventional side information such as skip connections may not work properly with unseen translations. We found that pooling indices are more robust and invariant, and provide helpful hints to guide the reconstruction of spatial structure.

We also analyzed a specific limitation of the original M&MNets [172] in the zero-pair setting, which is that a significant part of the shared features between unseen modalities is not exploited. We proposed a variant that generates pseudo-pairs to enforce the networks to use more information between unseen modalities, even when that information is not shared by seen translations. The effectiveness of M&MNets with pseudo-pairs has been evaluated in several multi-modal datasets.

A potential limitation of our system is that we work with separate encoder and decoders for each modality. Some recent cross-domain image translators such as StarGAN [31] and SDIT [170] use a single shared encoder and a single shared decoder. In that spirit, it could be possible to have partially shared encoders and decoders between different modalities. However, modality-specific layers would be still required in more challenging cross-modal translation.

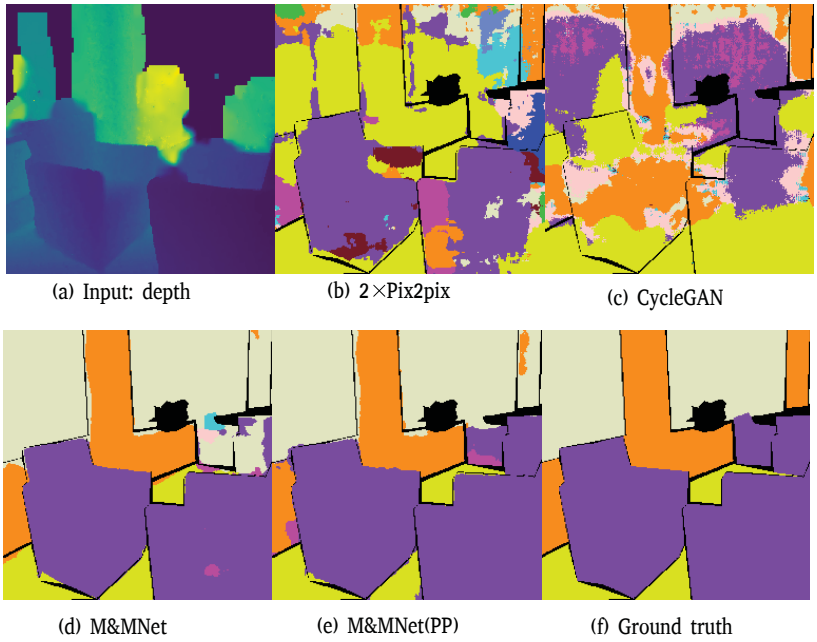
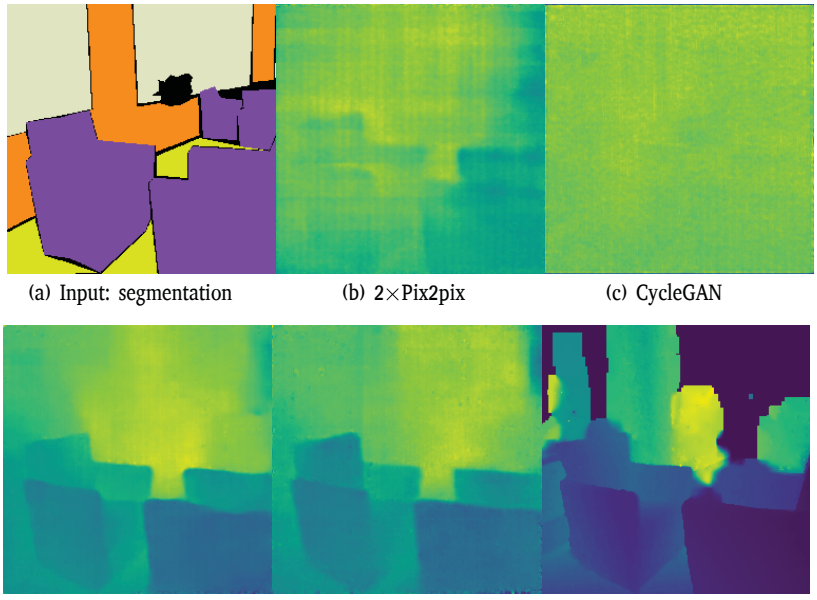


Figure 4.15 – Example of zero-pair depth-to-segmentation on SUN RGB-D.



(a) Input: segmentation (b) 2×Pix2pix (c) CycleGAN

(d) M&MNet (e) M&MNet(PP) (f) Ground truth

Figure 4.16 – Example of zero-pair segmentation-to-depth on SUN RGB-D.

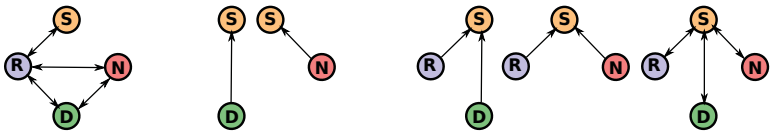


Figure 4.17 – Cross-modal translations in the Freiburg Forest dataset experiment: (a) train, (b) test (zero-shot) and (c) test (multimodal). We show only translations to semantic segmentation for simplicity.

Method	Conn.	Sky	Trail	Grass	Vegetation	Obstacle	mIoU	Global
Baselines								
CycleGAN $D \rightarrow S$	SC	36.3	31.7	19.2	24.5	5.40	23.4	26.2
CycleGAN $N \rightarrow S$	SC	37.2	34.1	18.4	29.5	0.41	23.9	28.5
2×pix2pix$D \rightarrow S$	SC	72.9	32.2	45.7	67.9	30.9	49.9	59.9
2×pix2pix$N \rightarrow S$	SC	78.6	43.2	53.4	74.4	18.6	53.6	66.8
StarGAN$D \rightarrow S$	PI	45.2	28.1	24.4	21.5	1.36	24.1	28.1
StarGAN $N \rightarrow S$	PI	31.2	15.1	29.4	23.2	10.7	21.9	25.8
M&MNets $D \rightarrow R \rightarrow S$	PI	45.3	19.6	25.4	35.5	25.3	30.0	33.5
M&MNets $N \rightarrow R \rightarrow S$	PI	58.1	34.1	32.4	42.4	12.3	35.8	42.4
Zero-pair								
M&MNets $D \rightarrow S$	PI	89.0	71.8	71.3	82.7	43.7	71.6	80.0
M&MNets $N \rightarrow S$	PI	88.1	78.1	73.4	83.1	41.0	72.7	81.0
M&MNets+PP $D \rightarrow S$	PI	89.7	75.4	72.4	83.6	45.7	73.4	81.1
M&MNets+PP $N \rightarrow S$	PI	89.9	80.1	76.9	85.5	44.2	75.3	83.5
Multi-modal								
M&MNets $(R, D) \rightarrow S$	PI	91.2	84.5	85.4	89.1	50.3	80.1	88.0
M&MNets $(R, N) \rightarrow S$	PI	91.0	83.5	85.3	90.0	52.9	80.5	88.3
M&MNets $(R, D, N) \rightarrow S$	PI	91.2	84.2	85.8	90.1	58.2	81.8	88.5
M&MNets+PP $(R, D) \rightarrow S$	PI	90.9	83.9	85.0	88.7	59.5	81.6	88.1
M&MNets+PP $(R, N) \rightarrow S$	PI	**91.7**	85.4	86.1	89.9	58.2	82.2	88.6
M&MNets+PP $(R, D, N) \rightarrow S$	PI	91.5	**85.8**	**86.6**	**90.6**	**60.3**	**83.0**	**89.3**
Oracle								
$D \rightarrow S$	PI	89.5	75.4	80.4	81.2	54.7	76.2	82.2
$N \rightarrow S$	PI	90.2	81.5	83.6	85.2	50.4	78.2	85.4
$(R, D, N) \rightarrow S$	PI	91.9	85.7	87.9	90.1	64.9	84.1	89.4

Table 4.8 – Zero-pair (NIR, depth)-to-semantic segmentation on Freiburg Forest. **SC**: skip connections, **PI**: pooling indexes, **PP**: pseudo-pairs.

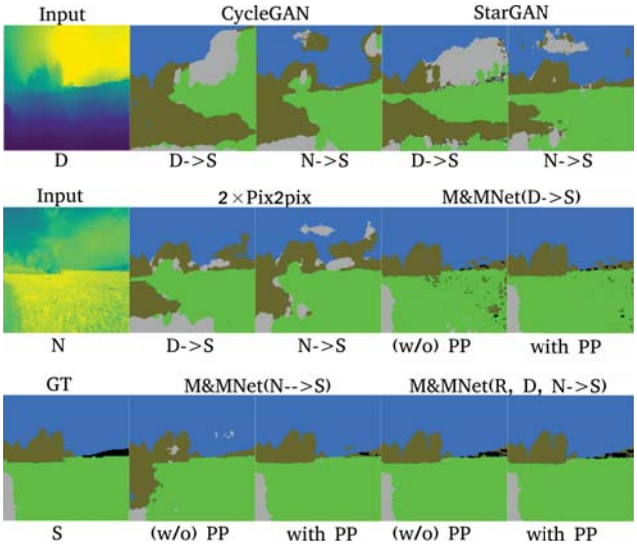


Figure 4.18 – Example of zero-pair translations. R: RGB, D: depth, N: NIR, S: semantic segmentation, and PP: pseudo-pairs.

5 Controlling biases and diversity in diverse image-to-image translation [1]

5.1 Introduction

Image-to-image translation (simply image translation hereinafter) is a powerful framework to apply complex data-driven transformations to images [53, 70, 81, 93, 172, 181]. The transformation is determined by the data collected from the input and output domains, which can be arranged as explicit input-output instance pairs [70] or just the looser pairing at set level [81, 100, 181, 206], known as paired and unpaired image translation, respectively.

Early image translation methods were deterministic in the sense that same input image is always translated to the same output image. However, a single input image often can have multiple plausible output images, allowing for variations in color, texture, illumination, etc. Recent approaches allow for diversity[2] in the output [68, 93, 207] by formulating image translation as a mapping from an input image to a (conditional) output distribution (see Fig. 5.1a), where a particular output is sampled from that distribution. In practice, the sampling is performed in the latent representation that is the input of the generator, which is explicitly disentangled into content representation and style representation [93, 207]. Concretely, the style code is sampled to achieve diversity in the output while preserving the image content. A concern with image translation models, and machine learning models in general, is that they capture the inherent biases in the training datasets. The problem of undesired bias in data is paramount in deep learning, raising concerns in multiple communities as automation and artificial intelligence become pervasive in their interaction with humans, such as systems involving analyzing face or person images, or communication in natural language. For example, it is known that most face recognition systems suffer from gender and racial bias [20]. Similar gender bias is observed in image captioning [59]. Here we focus on the kind of biases that may affect image translation systems. Although bias is inherent to data collection,

[1] This chapter is submitted for journal review.

[2] In some papers this is referred to as *multimodal*, in the sense that the output distribution can have multiple modes.

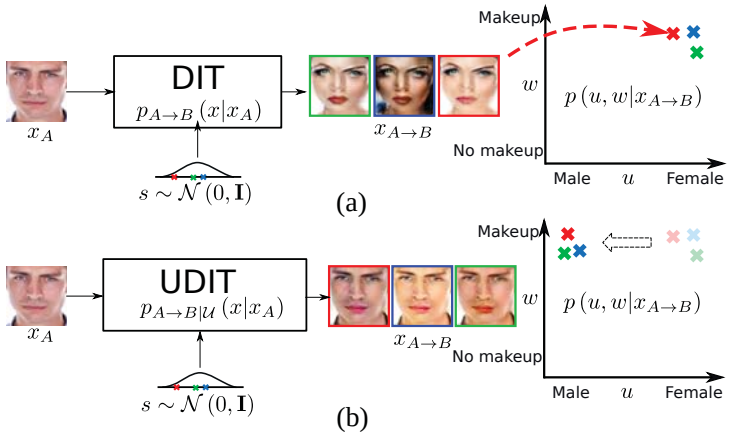


Figure 5.1 – Diverse image-to-image translation in a very biased setting (domain A: mostly white males without makeup, domain B: white females with makeup): (a) biased translations, (b) with semantic constraint to alleviate bias while keeping relevant diversity.

it is certainly possible to design better and more balanced datasets, or at least understand the related biases, their nature and try to incorporate tools to alleviate them [64, 73, 200, 208].

What particular visual and semantic properties of the input image are changed during the translation is determined by the internal and relative biases between the input and output training sets. These biases have significant impact on the diversity and potential unwanted changes, such as changing the gender, race or identity of a particular input face image. As an example we can consider the input domain *faces without makeup* and the output domain *faces with makeup*, so we expect that the image translator learns to add makeup to a face. However, the input training set may be heavily biased towards males without makeup, and the output training set towards females with makeup[3]. With such biases, the translator learns to generate female faces with makeup even when the input is a male face (see Fig. 5.1a). While the change in the makeup attribute is desired, the change in identity and gender are not.

In this work we propose to make the image translator counter undesired biases, by incorporating *semantic constraints* that enforce minimizing the undesired

[3] In addition to biases towards white and young people, we do not consider other specific biases in this example for the sake of simplicity.

changes (e.g. see Fig. 5.1b when constraining the identity, which implicitly constrains gender). These constraints are implemented as neural networks that extract relevant semantic features. Designing an adequate semantic constraint is often not trivial, and naive implementations may carry irrelevant information.

This often leads to undesired side effects such as ineffective bias compensation and limiting the desired diversity in the output. Here we address these issues and propose an approach to design an effective semantic constraint that both alleviates bias and preserves desired diversity.

5.2 Related Work

Image-to-image translation has recently received exceptional attention due to its excellent results and its great versatility to solve multiple computer vision problems [18, 68, 70, 94, 103, 194, 206, 207]. Most image translation approaches employ conditional Generative Adversarial Networks (GANs) [54], which consist of two networks, the generator and the discriminator, that compete against each other. The generator attempts to generate samples that resemble the original input distribution, while the discriminator tries to detect whether samples are real or originate from the generator. In the case of image translation, this generative process is conditioned on an input image. The seminal work of [70] was the first GAN-based image translation approach that was not specialized to a particular task. In spite of the exceptional results on multiple translation tasks such as grayscale to color images or edges to real images, this approach is limited by the requirement of pairs of corresponding images in both domains, which are expensive to obtain and might not even exist for particular tasks. Several methods [81, 100, 158, 181, 206] have extended pix2pix to the unpaired setting by introducing a cycle consistency loss, which assumes that mapping an image to the target domain and then translating it back to the source should leave it unaltered.

Diversity in image-to-image translation. A limitation of the above image translation models is that they do not model the inherent diversity of the target distribution (e.g. same shoe can come in different colors). For example, pix2pix [70] tries to generate diverse outputs by including noise alongside the input image, but this noise is largely ignored by the model and the output is effectively deterministic. BicycleGAN [207] proposed to overcome this limitation by adding the reconstruction of the latent input code as a side task, thus forcing the generator to take noise into account and create diverse outputs. BicycleGAN still requires paired data. In the unpaired setting, several recent works [5, 68, 93] address unpaired diverse image translation. Our approach falls into this category as it does not need paired data and it outputs diverse translations. Our work is closest to MUNIT [68], which divides the

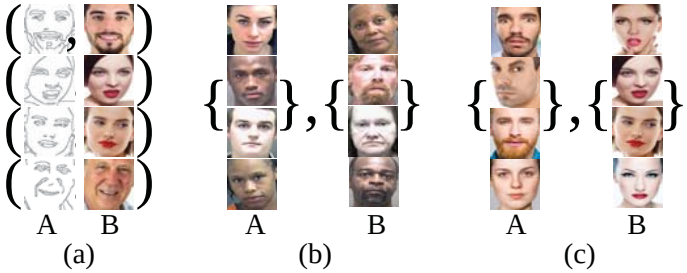


Figure 5.2 – Examples of training sets for image translation: (a) paired edge-photo, (b) unpaired young-old (well-aligned biases), and (c) unpaired without-with makeup (misaligned in gender).

latent space into a shared part across domains and a part specific to each domain. However, these methods output too much diversity in some cases, which results in the undesired change of image content that should be preserved by the model (e.g. identity, race). Moreover, such changes are often determined by the underlying bias in the dataset, which MUNIT captures and amplifies during translation.

Disentangled representations. While DIT methods explicitly disentangle content and style to enable diversity, other methods attempt to obtain disentangled representations to isolate different factors of variation in images [13], which is beneficial for tasks such as cross-domain classification [16, 17, 47, 104] or retrieval [53]. In the context of generative models, [112] combined a GAN with a Variational Autoencoder (VAE) to obtain an internal representation that is disentangled across specified (e.g. labels) and unspecified factors. InfoGAN [25] achieves some control over the variation factors in images by optimizing a lower bound on the mutual information between images and their representations. Some approaches impose a particular structure in the learned image manifold, either by representing each factor of variation as a different sub-manifold [135] or by solving analogical relationships through representation arithmetic [136]. The work of [53] achieves cross-domain disentanglement by separating the internal representation into a shared part across domains and domain-exclusive parts, which contain the factors of variation of each domain. In our case we assume we do not have access to disentangled representations beyond content and style, and especially between wanted and unwanted changes.

Bias in machine learning datasets. Since machine learning is mostly fitting predictive models to data, the problem of biased training data is of great relevance. Dataset bias in general refers to the observation that models trained in one dataset may lead to poor generalization when evaluated on other datasets, due to the specific

bias in each of them [161]. Bias is multifaceted, and datasets can be biased in many ways (e.g. illumination conditions, capture devices, class imbalance, scale [60]). Dataset bias can be addressed and improve cross-dataset generalization [43, 79]. A related problem is domain adaptation [34, 129] where models trained on a source domain are adapted to a target domain, trying to overcome the difference in biases. Biased datasets lead to biased models, which have severe implications as data-driven artificial intelligence becomes pervasive. For instance, most commercial face recognition and image captioning systems exhibit gender and ethnicity biases [20, 59]. Therefore, tackling bias is an increasingly important topic in machine learning [64, 73, 200, 208]. Here we focus on the specific problem of understanding bias in image translation.

5.3 Diverse image translation

5.3.1 Definition and Setup

Our goal is to translate samples from a source domain A to a target domain B in an unpaired setting, i.e. without corresponding images across domains. Let $x_A \in X_A$ be a sample from the marginal distribution of images in the source domain, $p_A(x)$. We want to obtain a translation $x_{A\to B}$ to B, sampled from a conditional distribution $p_{A\to B}(x|x_A)$ that approximates the true conditional $p_B(x|x_A)$. The difficulty of this task resides in the impossibility to observe the joint distribution $p_{A,B}(x_A, x_B)$ in the unpaired setting, and the complexity of the conditional distribution $p_B(x|x_A)$, which is generally multi-modal. Simultaneously, we want to obtain the inverse translation $x_{B\to A}$.

Current unpaired diverse image translation methods [68, 93] use an encoder-decoder architecture, where the input image is first encoded into a latent code and then later decoded to generate the translated target image. These methods resort to the assumption that part of the latent space, the *content*, is shared by both domains, whereas the *style* contains only the domain-specific characteristics. Concretely, let us consider content encoders E_i^c and style encoders E_i^s, where $i \in \{A, B\}$ indexes over domains. Then, the latent representation of an input image x_i can be decomposed into content $c_i = E_i^c(x_i)$ and style $s_i = E_i^s(x_i)$. Given that style is purely domain-specific, we only need the particular content code c_i for translation, combined with a randomly sampled style code $s' \sim \mathcal{N}(0, \mathbf{I})$, to generate the output image through the decoder G_j as $x_{i\to j} = G_j(c_i, s')$.

Note that the decoders are deterministic functions that act as inverses of the encoders ($x_i = G_i(E_i^c(x_i), E_i^s(x_i))$, the stochasticity of the output translations is introduced through the sampling of the style code, which is the source of diversity on the generated translations (Fig. 5.4a).

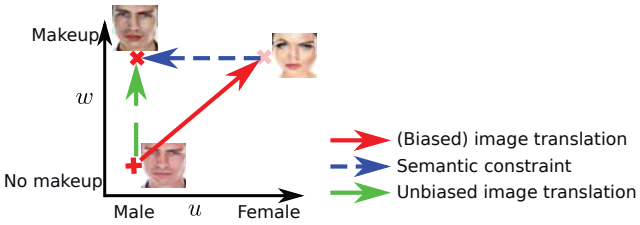


Figure 5.3 – Geometric interpretation of the semantic constraint unbiasing the translation.

5.3.2 Biases in diverse image translation

Wanted and unwanted properties. Images are complex and diverse in nature, reflected at many levels, such as visual appearance, structure and semantics. Therefore, the dataset bias is also complex and multifaceted, and it may be convenient to analyze separately specific biases depending on specific semantic properties. Let $a(w,u)$ represent the relevant semantic properties associated with an image x that are subject to change during translation, with w being those we want to change (i.e. *wanted*), and u being those we do not want to be changed (i.e. *unwanted*). We assume that they can be obtained via the mappings $w = g(x)$ and $u = h(x)$. For instance, in the example of Fig. 5.1, w is makeup and u is gender (for simplicity, but more generally u could also include identity, race, etc.). The distributions of images of the source domain i and the target domain j induce the corresponding distributions of properties $p_i(w,u|x_i)$ and $p_j\left(w,u|x_j\right)$, respectively.

Translations in the space of properties. During training, the image translator learns the mapping between both domains, and consequently what properties to modify. An input image x_i has the properties $w_i = g(x_i)$ and $u_i = g(x_i)$, and the corresponding translation $x_{i\to j}$ will have $w_{i\to j} = g\left(x_{i\to j}\right)$ and $u_{i\to j} = h\left(x_{i\to j}\right)$. The image translation is *successful* if and $w_{i\to j} \neq w_i$ is effectively the wanted property of the target domain. Similarly, a translation is *unbiased* when $u_{i\to j} = u_i$. In general, DIT results in biased translations when $u_{i\to j} \neq u_i$ (see Fig. 5.3), which stems from the original bias in the training dataset.

5.4 Unbiased diverse image translation

5.4.1 Unbiasing the generated images

For simplicity, let us consider the paired image translation case where a ground truth translation x_j is available for each x_i, with the corresponding properties

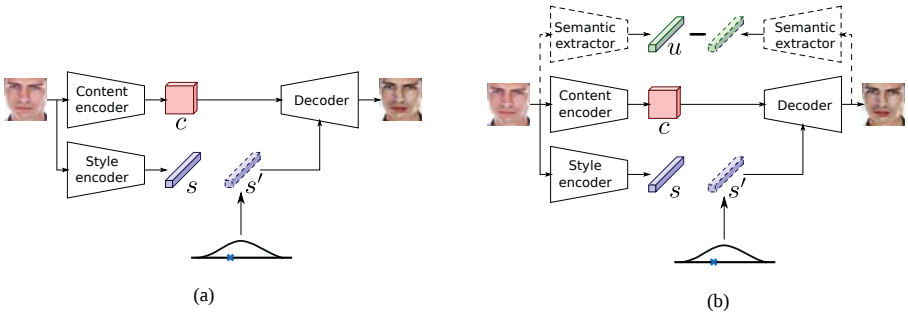


Figure 5.4 – Diverse image-to-image translation (DIT): (a) biased, (b) unbiased (i.e. UDIT) via a semantic constraint implemented with a semantic extractor.

$(w_j, u_j) = g(x_j)$. In order to learn a successful and unbiased translation we would like to enforce the constraints $w_{i\to j} = w_j$ and $u_{i\to j} = u_i$, respectively.

However, we focus on the the more complex case of diverse image translation, where the output is stochastic, i.e. a distribution rather than a single image. In this case the constraints may not be enforced at the sample level but at the distribution level. In the case of u we aim at enforcing

$$u_{i\to j} = h(x_{i\to j}) = h(x_i) = u_i \tag{5.1}$$
$$\forall x_{i\to j} \sim p_{i\to j}(x|x_i), \forall x_i \sim p_i(x)$$

which ensures that the unwanted properties remain unchanged throughout the translation. Similarly,

$$w_{i\to j} = g(x_{i\to j}) = g(x_j) = w_j \tag{5.2}$$
$$\forall x_{i\to j} \sim p_{i\to j}(x|x_i), \forall x_j \sim p_j(x|x_i), \forall x_i \sim p_i(x)$$

which ensures that the wanted properties change properly, according to the desired translation. Note that for convenience we assume that the true conditional distribution of the translation $p_j(x|x_i)$ is known.

In this way, the biases in the distribution of generated images would be aligned properly, achieving our goal of removing unwanted biases in the translation (see Fig. 5.3). In the previous example we would like the translated images to preserve the statistics of gender distribution of A while adapting to the statistics of makeup distribution of B. Similarly in the direction from B to A.

Note that the different settings in image translation implicitly or explicitly enforce this sort of alignments via pairing or the design of the dataset. For instance,

Fig. 5.2a shows an example of a dataset for paired translation, where the instance-level pairing already prevents unwanted gender bias (50% males and females). Gender bias can also be prevented in unpaired translation by designing well-balanced and statistically aligned training sets for domains A and B (see Fig. 5.2b). However, Fig. 5.2c shows a dataset clearly biased and misaligned on gender. In this case, it is desirable that the model can be forced to correct this unwanted misalignment, to prevent biased translations.

In practice, directly enforcing constraints (5.1) and (5.2) is not possible since w and u are not disentangled in our setting. Besides, we do not have access to $p_j(x|x_i)$.

For this reason we propose to implement (5.1) via the addition of a semantic regularization constraint that enforces the preservation of u properties during translation, while constraint (5.2) is indirectly enforced via the image translation loss. A bad implementation of the semantic constraint can hamper the effectiveness of image translation in practice (e.g. limiting diversity), so the appropriate design of the semantic constraint and its implementation is related to both constraints.

5.4.2 Semantic regularization constraint

Here we propose an Unbiased DIT model (UDIT) that enforces constraint (5.1) via a *semantic extractor* h that estimates the representative semantic properties we want to preserve in the image as $u_i = h(x_i)$.

Constraint (5.2) on the wanted changes is implicitly enforced by the DIT model, including the unpaired setting. Fig. 5.4b illustrates how a proper semantic constraint regularizes the initial DIT model to alleviate the unwanted bias.

In particular, we include a *semantic constraint loss*

$$\mathscr{L}_{\mathscr{U}}^{u_i} = \mathbb{E}_{x_i \sim p_i(x)}\big[||u_{i\to j} - u_i||\big], \tag{5.3}$$

where $\mathscr{U}$ represents the semantic properties we want to keep unchanged throughout the translation. By including $\mathscr{L}_{\mathscr{U}}^{u_i}$ in our training objective (sec. 5.5.2), we are effectively conditioning the output conditional distribution to $\mathscr{U}$, i.e. $p_{i\to j|\mathscr{U}}(x|x_i)$, and hence alleviating the unwanted bias in the output samples $x_{i\to j} \sim p_{i\to j|\mathscr{U}}(x|x_i)$, when $\mathscr{U}$ is properly designed. Fig. 5.4b shows the architecture of this UDIT. Note how this constraint is only enforced during training, we do not use u_i during translation at inference time.

5.5 UDIT implementation

5.5.1 Semantic extractor

Crucial for the success of our method is the proper design of the semantic extractor $h(x)$, which in general will be implemented as a neural network. We must guarantee that the extracted feature contains enough relevant information regarding the specific semantic property that we want to preserve (i.e. captures u properly). On the other hand, we want to prevent it from containing additional information that could potentially introduce undesired side effect such as limiting the translation ability of the model or the diversity on the output. We now develop a procedure to design effective semantic extractors that satisfy both requirements.

Capturing the semantic property. As feature extractors, we consider convolutional neural networks (CNNs) implementing classification tasks related with u (e.g. gender classification and facial behavior analysis [145]), which we train on a suitable external dataset. The CNN may also be initialized with models pretrained in large datasets (e.g. ImageNet [146], DeepFace [159]). In principle we are interested in a suitable intermediate feature that captures u well. In particular, the convolutional features that are input into the first fully connected layer are often good candidates, as they contain semantically meaningful information while still being spatially localized.

Reducing undesired information. Deep features from generic feature extractors such as models trained in ImageNet capture rich and varied properties in a relatively high dimensional feature. This can be harmful in our case, since they can also capture properties unrelated with u. The classifier can learn to ignore them and still solve the task, but they remain as noise in the semantic feature, being enforced through the constraint and therefore limiting the flexibility of the image translator to generate the wanted change and diversity. In order to address this problem, we propose to add an additional convolutional layer with a kernel $1 \times 1 \times D$ with the purpose of reducing the dimensionality of the feature. We experimentally find the minimum value of D that keeps a satisfactory accuracy. The output of this additional layer is used as semantic feature.

In summary, the designed features will ideally contain the right amount of information relevant for the task, and no irrelevant information that could interfere with the wanted translation.

5.5.2 Full model

The proposed unbiasing methodology is generic enough to be applicable in most image-to-image translation methods. The UDIT models in our experiments are

based on MUNIT [68] extended with particular semantic constraints. The model is composed of within-domain autoencoders and cross-domains translators with reconstruction of translated features. We also consider a variant that uses pooling indices as side information [12].

In the following, we detail the remaining losses and present the full model.

Adversarial loss. The translator attempts to generate realistic images that fool the discriminator D_j, whose task is to distinguish fake images from real images. The discriminator is trained adversarially with

$$\begin{aligned}\mathscr{L}_{GAN}^{x_j} = &\frac{1}{2}\mathbb{E}_{x_i\sim p_i(x),s'\sim\mathcal{N}(0,\mathbf{I})}\left[(D_j(G_j(c_i,s'))^2\right]\\ &+\mathbb{E}_{x_j\sim p_j(x)}\left[(D_j(x_j)-1)^2\right].\end{aligned} \tag{5.4}$$

Reconstruction loss. The autoencoders ensure that the model is able to reconstruct the input image through the image reconstruction loss

$$\mathscr{L}_{recon}^{x_i} = \mathbb{E}_{x_i\sim p_i(x)}\left[||G_i(c_i,s_i)-x_i||_1\right]. \tag{5.5}$$

Moreover, the translated image is further encoded in both content and style, and the following feature reconstruction losses are applied

$$\mathscr{L}_{recon}^{c_i} = \mathbb{E}_{x_i\sim p_i(c),s'\sim\mathcal{N}(0,\mathbf{I})}\left[||E_j^c(G_j(c_i,s'))-c_i||\right], \tag{5.6}$$

$$\mathscr{L}_{recon}^{s_i} = \mathbb{E}_{x_i\sim p_i(c),s'\sim\mathcal{N}(0,\mathbf{I})}\left[||E_j^s(G_j(c_i,s'))-s'||\right]. \tag{5.7}$$

The loss on c_i enforces the preservation of the content code across domains, whereas the loss on the style encourages diversity on the outputs.

The loss used to trained UDIT follows MUNIT's loss combined with the semantic constraint loss (5.3) as follows

$$\begin{aligned}\mathscr{L} =&\mathscr{L}_{GAN}^{x_A}+\mathscr{L}_{GAN}^{x_B}+\lambda_x(\mathscr{L}_{recon}^{x_A}+\mathscr{L}_{recon}^{x_B})\\ &\lambda_c(\mathscr{L}_{recon}^{c_A}+\mathscr{L}_{recon}^{c_B})+\lambda_s(\mathscr{L}_{recon}^{s_A}+\mathscr{L}_{recon}^{s_B})\\ &\lambda_{\mathscr{U}}(\mathscr{L}_{\mathscr{U}}^{u_A}+\mathscr{L}_{\mathscr{U}}^{u_B}),\end{aligned} \tag{5.8}$$

where the $\lambda_x,\lambda_c,\lambda_s,\lambda_{\mathscr{U}}$ weights control the influence of each individual loss in the final objective. When $\lambda_{\mathscr{U}}=0$ we recover the baseline MUNIT model.

5.6 Experimental results

5.6.1 Datasets

We conduct experiments on four datasets that suffer from different types of biases.

Biased makeup is our heavily biased dataset, where the female gender predominates in the target domain. We collected images of people with and without makeup from the web. We retrieved 1,400 images of women with makeup by searching for "woman makeup face" and manually verifying them. For the no-makeup domain, we selected another 1400 images with 95% males faces and 5% female faces, so we purposely biased this domain towards males. All images were preprocessed by cropping the face, localized by a face detector.

MORPH [137] is also a face dataset for age translation (young ↔ old) with both ethnicity and gender biases. It contains 55,134 images of 13,000 subjects, and each image is annotated with gender, ethnicity, and age. There are five ethnic groups represented in the dataset: Black (African ancestry), White (European ancestry), Hispanic, Asian, and 'Other', which we discarded.

MORPH is a face image dataset for adult age progression, where the images depict people of different ages at different points in time, spanning up to 30 years for some subjects. MORPH is heavily biased towards men (>85%), and towards individuals with African ancestry (>78%), followed by European (≈17%), Hispanic (≈3.5%) and Asian (<0.3%) ancestries. We perform experiments using the identity constraint (sec. 5.6.4) with the purpose of preserving both gender and ethnicity.

Cityscapes [32]→Synthia [142] contains real and synthesized urban scenes that are biased towards a particular time of the day (day/night). Cityscapes [32] contains real street scenes captured from a moving vehicle during day-time (3000 images). Synthia [142], instead, is synthetically generated by a simulated car driving in a virtual world, both during day-time and night-time. We artificially bias the day-time/night-time distribution of Sytnhia by selecting 3000 images captured during night and only 300 images during day.

Biased handbags [206] contains images of handbags with two defining attributes: color (*red/black*) and texture (*flat/textured*). We select red and black as possible colors. Texture is also a binary attribute indicating the absence or presence of a non-flat texture on the handbags, i.e. flat or textured.

We create two datasets by selecting samples from the photo images of the handbags dataset used by [68, 70]. The input domain only contains one mode (e.g. flat black handbags for Handbags-color), while the target domain contains two modes but is heavily biased towards one, e.g. 1000 textured red and 100 flat red.

We note that we require the textured handbags to only have the right color (e.g. no stripes of another color), which limits the attribute to subtle variations mostly given by differences in the material.

Tables 5.1 and 5.2 specify the exact number of images used in our biased datasets for training and testing, respectively. Table 5.3 reports the setting to train the metric network.

Experiment	Domain A	Domain B
Biased makeup	1400 f-makeup	1330 m-nomakeup, 70 f-nomakeup
MORPH	10000 m-y, 1000 f-y	10000 m-o, 1000 f-o
Cityscapes-Synthia	3000 citys-day	3000 syn-night, 300 syn-day
Handbags-color	755 flat-black	1000 txt-red, 100 flat-red
Handbags-texture	1256 flat-red	1100 txt-black, 100 txt-red

Table 5.1 – Details of datasets used for *training* the image translation models. Note f=female, m=male, y=young, MORPHo=old, citys=cityscapes, syn=synthia, txt=textured.

Experiment	Domain A	Domain B
Biased makeup	100 f-makeup	100 m-nomakeup
MORPH	200 m-y, 200 f-y	200 m-o, 200 f-o
Cityscapes-Synthia	475 citys-day	-
Handbags-color	100 flat-black	-
Handbags-texture	100 flat-red	-

Table 5.2 – Details of datasets used for *testing* the image translation models. Note f=female, m=male, y=young, o=old, citys=cityscapes, syn=synthia, txt=textured.

Note for the biased makeup dataset, the used gender classifier is externally trained on Adience dataset [95].

Experiment	Domain A	Domain B
MORPH-gender	2000 m-y, 2000 m-o	2000 f-y, 2000 f-o
MORPH-ethnicity	1200 afri-y, 1200 afri-o	1200 euro-y, 1200 euro-o
Cityscapes-Synthia	3000 BDD-day, 3000 syn-day	3000 BDD-night, 3000 syn-night
Handbags-MORPHcolor	500 flat-red, 500 flat-black	500 txt-red, 500 txt-black
Handbags-texture	500 flat-red, 500 txt-red	500 flat-black, 500 txt-black

Table 5.3 – Details of datasets used training the classifier to evaluate quantitatively the results. Abbreviations used: f=female, m=male, y=young, o=old, afri=african, euro=european, BDD=BDD100K, syn=synthia, txt=textured. Note the used subsets are disjoint with the ones used to perform image translation.

5.6.2 Baselines and variants

We compare our method with the following approaches:

MUNIT [68] disentangles the latent distribution into the content space which is

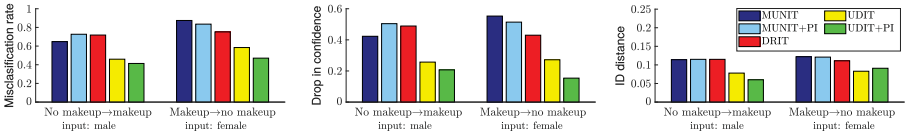


Figure 5.5 – Robustness to bias on Biased makeup: (left) misclassification rate, (middle) drop in confidence, (right) ID distance.

shared between two domains, and the style space which is domain-specific and aligned with a Gaussian distribution. At test time, MUNIT takes as an input the source image and different style codes to achieve diverse outputs.

DRIT [93] similarly explores the distribution of latent representation. Different from MUNIT by means of adaptive instance normalization to control diversity, DRIT directly insert noise into latent feature to achieve diverse output.

We compare the previous baselines with different configurations of the proposed UNIT approach. In particular we study variants with and without Pooling Index(PI).

5.6.3 Robustness to specific biases.

Evaluating the generated images is challenge [15], here we introduce a new method to measure whether translating an image across domains with misaligned biases changes particular properties of the image. For simplicity, we explain here these evaluation measures for the Biased makeup dataset (other datasets are similar). In particular, we want to evaluate whether applying or removing makeup on subjects changes their perceived gender. In order to do this, we train a gender classifier $f(x)$ and evaluate the gender prediction over the translated image, i.e. $f\left(x_{i\to j}\right)$. Since we have the ground-truth label for the original image, we can determine whether gender has been changed with respect to the original image. We call this measure *misclassification rate*. The problem with this measure is that the classifier might output erroneous estimates in the first place for some challenging cases.

For this reason, we also compute the *drop in confidence* of the classifier during translation as $\delta(x_i) = f(x_i) - f\left(x_{i\to j}\right)$.

This score will indicate the effect of the translation on the classifier estimation of the correct label, somewhat accounting for the classifier's failure cases.

We can use the above measures with general properties such as gender or race. However, our face experiments also include a setting in which we want to preserve the *identity* of the input. Evaluating changes in identity is more complex since the set of categories is specific to the dataset.

In this case, we measure the change in identity by directly computing the distance between identity features given an off-the-shelf face recognition net-

Figure 5.6 – Example translations for Biased makeup when applying makeup to a male. UDIT uses identity as semantic constraint.

Input	Direction	MUNIT	+PI	DRIT	UDIT	UDIT+PI
M	Makeup	0.268	0.267	0.263	0.192	0.151
F	Makeup	0.212	0.199	0.193	0.154	0.133
F	Demakeup	0.297	0.293	0.253	0.208	0.203

Table 5.4 – LPIPS distance on Biased makeup.

work [127]. We call this measure *ID distance* and only compute it for the face datasets.

Diversity. Several image translation approaches [68, 93, 207] measure the diversity of the outputs by using the perceptual similarity metric LPIPS [197], which is based on differences between deep features

We follow the protocol introduced in [207] and average the LPIPS distance between 19 random pairs of outputs for 100 different input images.

5.6.4 Biased makeup dataset

Semantic constraint. In this dataset, we focus on the misalignment between biases at two levels: gender and identity. Preserving identity is a more restrictive constrain than preserving gender, and implicitly also preserves it. For this reason, we use a semantic constraint based on identity (ID). We consider an off-the shelf

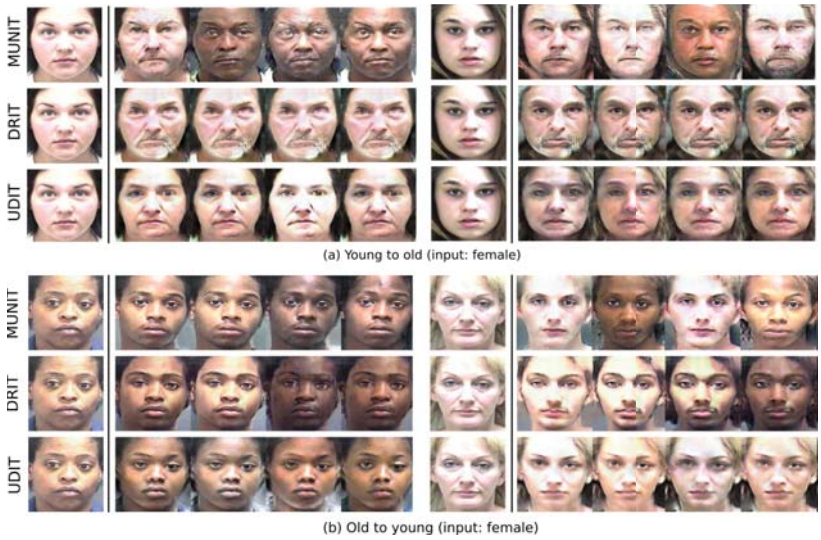


Figure 5.7 – Example translations on MORPH by biased DIT methods (MUNIT/-DRIT) and our UDIT with semantic constraint on identity.

network for face recognition [127] and select its highest level convolutional features as semantic feature. The model has been trained with VGG-Face [127], which contains over 2000 different identities.

Qualitative evaluation. Fig. 5.6 compares image translations obtained with MUNIT [68], MUNIT with pooling indices (PI), DRIT [93], and two variants of our model. The basic UDIT variant only uses a semantic constraint on ID, whereas UDIT+PI uses also pooling indices. We can observe that both MUNIT and DRIT change the gender (i.e. undesired change) when applying the desired translation (i.e. adding makeup).

This demonstrates the heavy influence of bias misalignment on DIT methods, which leads to the inevitable change of unwanted properties. Moreover, the generated images lack realism and quality, resembling cartoonish versions of human faces. Adding PI to MUNIT does not seem to bring any noticeable benefit.

Instead, our UDIT model trained with the ID semantic constraint is very effective to prevent both unwanted gender and identity changes, as show in the figure. Furthermore, the incorporation of pooling indices results in an even more successful change on wanted properties (e.g. adding makeup to males), while generating images of high quality and realism.

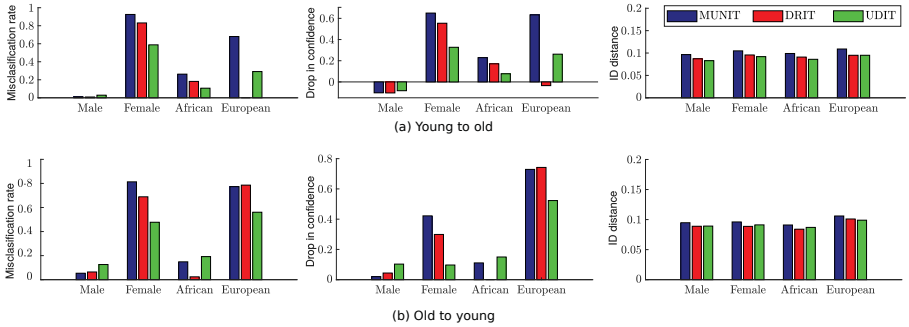


Figure 5.8 – Robustness to bias on MORPH: (a) *young* to *old* and (b) *old* to *young*: (left) misclassification rate, (middle) drop in confidence, and (right) ID distance.

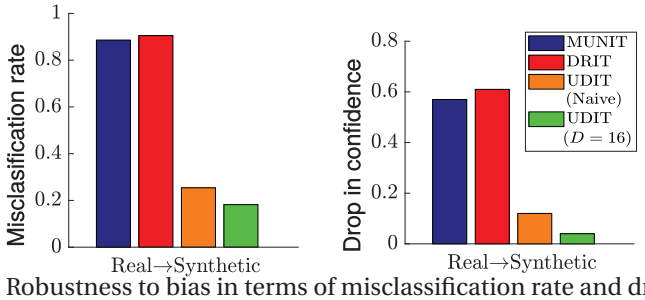


Figure 5.9 – Robustness to bias in terms of misclassification rate and drop in confidence.

Robustness to unwanted changes. Fig. 5.5 shows quantitative results of the three metrics evaluated on the different methods and both directions. We only evaluate over the gender that is underrepresented in the target domain. These results confirm the trends observed qualitatively in Fig. 5.6. DIT baselines perform poorly at maintaining gender and identity, including MUNIT with PI. Interestingly, the identity constraint clearly enhances the preservation of both wanted properties, as reflected by the substantial drop on all three robustness measures. Moreover, UDIT+PI further increases robustness to bias. This could be due to the improved quality of the output images with respect to the input, which leads to more reliable classifier predictions and pushes together the identity features. In the remainder of this chapter we only employ the UDIT+PI variant and refer to it simply as UDIT, unless stated otherwise.

Diversity. Table 5.4 shows the LPIPS distance of the different evaluated methods. UDIT models seem to be notably decreasing the LPIPS distance in comparison

D	2	8	16	32	64	128	256
Scenes-daytime	85	87	**91**	92	92	95	95
Handbags-color	96.3	**99.1**	99.0	99.3	98.3	98.9	98.4
Handbags-texture	64.2	65.2	66.4	**87.0**	91.3	92.8	95.4

Table 5.5 – Classifier accuracy for different D values. Boldface indicates the selected value for the semantic constraint.

to MUNIT and DRIT. This makes sense since the identity constraint not only prevents unwanted bias, but it also constrains the diversity in those directions that compromise the preservation of identity.

In this case, LPIPS distance may not be able to capture the more subtle variations that conform the diversity that should be expected in that setting. For example, the values for both UDIT variants are significantly lower than those of MUNIT or MUNIT+PI, but the examples in Fig. 5.6 show that it is able to generate very diverse images, within the narrow space that allows preserving gender and identity (e.g. lip color, skin tone and shading, beard thickness).

5.6.5 MORPH

Qualitative evaluation. Fig. 5.7a and b show examples of young female and old female, respectively, and their corresponding translations to the other domain (old and young). As we can observe, the translations are realistic in general. DRIT tends to output uni-modal samples / generate only one distribution mode, while the other two methods also generate rich variations, including skin tones, hair color, beard/moustache variations, etc. However, MUNIT tends to generate diversity that includes changes in ethnicity and gender.

In the case of the young female, gender is almost always changed due to the extreme bias towards males. UDIT, on the other hand, preserves the wanted semantic properties and outputs diversity without unwanted changes.

Robustness to unwanted changes. Here we evaluate how the identity constraint impacts gender and ethnicity changes compared to MUNIT and DRIT. Fig. 5.8 shows the misclassification rate and drop in confidence of two classifiers, gender and ethnicity, trained on a disjoint subset of MORPH not used for translation. We restrict our analysis to African and European, due to the very limited data in the other two ethnicities. The results show a drop in misclassification rate and a lower confidence drop when using UDIT, which are effective to alleviate gender bias (especially in females) and ethnicity bias (especially in Europeans). We also show ID distance, which achieves lower values for UDIT, indicating that identity is also

Figure 5.10 – Results on Cityscapes → Synthia-night. Example translations by MUNIT and UDIT with two variants of the semantic constraint.

better preserved. These results are in line with the observations in Fig. 5.7.

5.6.6 Cityscapes → Synthia-night

Semantic constraint. We train a binary classifier for daytime classification based on VGG16 [153] using both real and synthetic images. We use 6000 realistic images from BDD-100K [184] with a 50/50 daytime distribution. As synthetic images we use 6000 images from a disjoint subset of Synthia [142], also with a balanced class distribution. We consider two semantic constraints. The *naive* variant employs features of the last convolutional layer, which have dimension $8 \times 8 \times 512$. Given the high dimensionality of these semantic features, the undesired information contained in them could potentially limit the model's translation ability or the output diversity. For this reason, we also employ the *reduced* semantic constraint

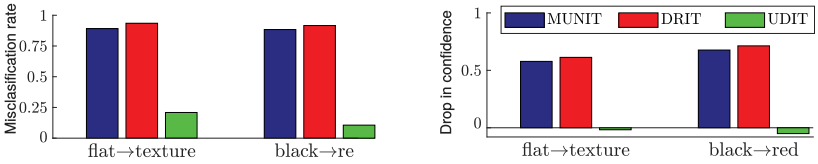


Figure 5.11 – Robustness to bias on Biased handbags.

variant presented in sec. 5.5.1, whose channel dimensions are reduced to D by an additional $1 \times 1 \times D$ layer. In order to select a suitable dimensionality we train several classifiers with different D values (Table 5.5). We select $D = 16$ as it offers a good trade-off between small size and accuracy.

Results. Figs. 5.10 and 5.9 present qualitative results and robustness measures respectively. MUNIT translations mostly depict night scenes, as can be confirmed by the high misclassification rate and drop in confidence. UDIT with naive constraint improves on this by preserving in the translations the input day-time. However, the outputs have clearly limited diversity and lower quality. UDIT with the reduced constraint achieves the overall best translations, both in terms of quality and wanted diversity. This leads to remarkably low values on both robustness measures.

5.6.7 Biased handbags

Semantic constraint. We consider two different semantic constraints depending on the experiment. For Handbags-texture we train a color classifier selecting 500 images per color from [186]. For Handbags-color, we gather images from the web searching for e.g. "textured red handbag" and verifying the downloaded images. We use 1000 flat and 1000 textured handbags to train the classifier. We only consider here the reduced variant of the semantic constraint. Table 5.5 shows the accuracy results for the different D values. We select $D = 8$ for color and $D = 32$ for texture. The overall lower accuracy of the texture classifier indicates that this is indeed a more subtle attribute, which in turn makes its recognition more challenging and increases the required dimensionality on the semantic features.

Results. Fig. 5.12 shows example results for these two experiments, evidencing how MUNIT succumbs to both types of biases. UDIT, on the other hand, manages to perform the desired translation without introducing unwanted changes. In general, the effects are more obvious for the color attribute as texture changes are harder to perceive. We confirm the benefits of UDIT quantitatively in Fig. 5.11. MUNIT and DRIT present a notably high misclassification rate and drop in confidence for both experiments. UDIT, instead, significantly increases the robustness to biases using a properly designed semantic constraint.

Figure 5.12 – Example translations for Handbags-texture (left) and Handbags-color (right). Better viewed electronically, zoom might be necessary to appreciate the changes in texture.

5.7 Conclusion

In this work we tackle the problem of learning image translation models from biased datasets, which leads to unwanted changes in the output images. In order to address tdirection of MORPH.his problem, we propose the use of semantic constraints, which can effectively alleviate the effects of biases. A properly designed semantic constraint allows for wanted diversity in the translations while preserving the desired semantic properties of the input image. We evaluated the effectiveness of our UDIT model on faces, objects, and scenes.

6 SDIT: Scalable and Diverse Cross-domain Image Translation [1]

6.1 Introduction

Image-to-image translation aims to build a model to map images from one domain to another. Many computer vision tasks can be interpreted as image-to-image translation, e.g. style transfer [49], image dehazing [193], colorization [196], surface normal estimation [41], and semantic segmentation [106]. Face translation has always been of great interest in the context of image translation, and several methods [31, 130, 131] have shown outstanding performance. Image-to-image translation can be formulated in a supervised manner when corresponding image pairs from both domains are provided, and unsupervised otherwise. In this chapter, we focus on unsupervised image-to-image translation with the two-fold goal of learning a model that has both scalability and diversity (see Figure 6.1(a)).

Recently, Isola *et al.* [70] consider a conditional generative adversarial network to perform image mapping from input to output with paired training samples. One of the drawbacks, however, is that this method produces a deterministic output for a given input image. BicycleGAN [207] extended image-to-image translation to one-to-many mappings between images by training the model to reconstruct the noise used in the latent space, effectively forcing it to use it in the translations. To address the same concern, Gonzalez-Garcia *et al.* [53] explicitly exploit the feature representation, disentangling the latent feature into shared and exclusive representations, the latter being aligned with the input noise.

The above methods, however, need paired images during the training process. For many image-to-image translation cases, obtaining abundant annotated data remains very expensive or, in some cases, even impossible. To relax the requirement of paired training images, recent approaches have made efforts to address this issue. The cyclic consistency constraint [81, 181, 206] was initially proposed for unpaired image-to-image translation. Liu *et al.* [100] assumes a shared joint latent distribution between the encoder and the decoder, then learns the unsupervised translation.

[1] This chapter is based on Conference of ACM Multimedia (ACM-MM 2019).

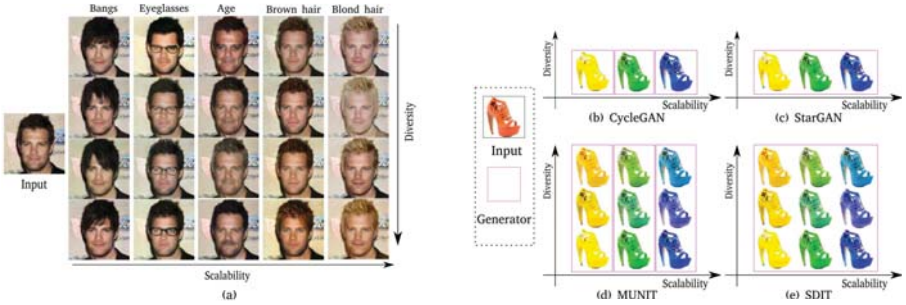


Figure 6.1 – (**a**) Example of diverse image translations for various attributes of our method generated by a single model. (**b-e**) Comparison to current unpaired image-to-image translation methods. Given four color subsets (*orange, yellow, green, blue*), the task is to translate images between the domains. (**b**) CycleGAN requires three independent generators (indicated by pink lines) which produce deterministic results. (**c**) StarGAN only requires a single generator but produces deterministic results. (**d**) MUNIT requires separate generators but is able to produce diverse results. (**e**) SDIT produces diverse results from a single generator.

Nonetheless, previous methods perform a deterministic one-to-one translation and lack diversity on its outputs, as shown in Figure 6.1(b). For example, given the task from orange (domain A) to yellow (domain B) the generator taking the orange shoes as input only synthesizes shows with a single shade of yellow. Recently, the idea of non-deterministic outputs was extended to unpaired methods [68, 93] by disentangling the latent feature space into content and style and aligning the style code with a known distribution (typically Gaussian or uniform). During inference, the model is able to generate diverse outputs by sampling different style codes from the distribution. The main drawback of these methods is that they lack scalability. As shown in Figure 6.1(d) the orange shoes can be translated into many possible green shoes with varying green shades. As the number of colors increases, however, the number of required domain-specific encoder-decoder pairs rises quadratically.

IcGAN [130] initially performs face editing by combining cGAN [115] with an attribute-independent encoder, and at the inference stage conducts face mapping for given face attributes. Recently, Yunjey *et al.* [31] proposed StarGAN, a domain-independent encoder-decoder architecture for face translation that concatenates the domain label to the input image. Unlike the aforementioned non-scalable approaches [68, 93], StarGAN is able to perform scalable image-to-image translation between multi-domains (Figure 6.1(b)). StarGAN, however, fails to synthesize diverse translation outputs.

In this chapter, we propose a compact and general architecture that allows for diversity and scalability in a single model, as shown in Figure 6.1(e). Our motivation is that scalability and diversity are orthogonal properties that can be independently controlled. Scalability is obtained by using the domain label to train a single multi-domain image translator, preventing the need to train a encoder-decoder for each domain. Inspired by [40], we employ Conditional Instance Normalization (CIN) layers in the generator to introduce the latent code and generate diverse outputs. We explore the reasons behind CIN's success (Fig. 6.6) and discover the following limitation: CIN affects the entirety of the latent features and could possibly modify areas that do not correspond to the specific target domain. To prevent this from happening, we include an attention mechanism that helps the model focus on domain-specific areas of the input image.

Our contributions are as follows:

- We propose a compact and effective framework that combines both scalability and diversity in a single model. Note that current models only possess one of these desirable properties, whereas our model achieves both simultaneously.
- We empirically demonstrate the effectiveness of the attention technique for multi-domain image-to-image translation.
- We conduct extensive qualitative and quantitative experiments. The results show that our method is able to synthesize diverse outputs while being scalable to multiple domains.

6.2 Related work

Generative adversarial networks. Typical GANs [54] are composed of two modules: a generator and a discriminator. The aim of the generator is to synthesize images to fool the discriminator, while the discriminator distinguishes between fake images and real images. There have been many variants of GANs [54] and they show remarkable performance on a wide variety of image-to-image translation tasks [68, 70, 93, 131, 181, 206], super-resolution [92], image compression [138], and conditional image generation such as text to image[108, 191, 192], segmentation to image[76, 167] and domain adaptation [47, 52, 147, 162, 177, 194, 209].

Conditional GANs. Exploiting conditional image generation is an active topic in GAN research. Early methods considered incorporating into the model category information [31, 115, 123, 124] or text description [74, 134, 191] for image synthesis. More recently, a wide variety of ideas have been proposed and used in several tasks such as image super-resolution [92], video prediction [111], and photo editing [151]. Similarly, we consider image-to-image translation conditioned on an input image

and the label of the target domain.

Image-to image-translation. The goal of image-to-image translation is to learn a mapping between images of the source domain and images of the target domain. Given pairs of data samples, pix2pix [70] initially performed this mapping by using conditional GANs and relying on the real images. This model, however, fails to conduct one-to-many mappings, namely, it cannot generate diverse outputs from a single input. BicycleGAN [207] explicitly modeled the mapping between output and latent space, and aligned the latent distribution with a known distribution. Finally, the diverse outputs are performed by sampling from the latent distribution. Gonzalez-Garcia *et al.* [53] disentangle the latent space into disjoint elements, which allows them to successfully perform cross-domain retrieval as well as one-to-many translation. The related work also be shown in [144]. Although these methods allow to synthesize diverse results, the requirement of paired data limits their application. Recently, the cycle consistency loss [81, 181, 206] is enforced into models to explicitly reconstruct the source sample, which is translated into the target domain and back, thus enabling translation using unpaired data. In addition, UNIT [100] aligns the latent space in two domains by assuming the similar domains share the same content. Although this approach shows remarkable results without paired data, they fail to perform diverse outputs. More recently, several image-to-image translation methods [5, 31, 131, 172] enable diverse results with the usage of noise or labels.

Diversity of image-to-image translation. Most recently, several approaches [25, 53, 68, 71, 93, 96, 189] consider to disentangle factors in feature space by enforcing a latent structure or regulating the structure distribution. Exploiting this disentangled representation enables the generator to synthesize diverse outputs by controlling style distribution. The key difference with the proposed method is that our method additionally performs *scalable* image-to-image translation while still having diversity.

Scalability of image-to-image translation. The scalability aim is to conduct image-to-image translation across multiple domains by a single generator. MM-Net [172] uses a shared encoder and a domain-independent decoder, not only allowing to perform style learning but zero-pair image-to-image translation. Anoosheh *et al.* [8] additionally consider encoder-decoder pairs for each domain as well as the used techniques in CycleGAN [206]. IcGAN [130] and StarGAN [31] condition the domain label on the latent space and input, respectively. Our approach also works by imposing domain labels in a single generator, but simultaneously enabling the model to synthesize diverse outputs.

Attention learning. Attention mechanisms have been successfully employed for image-to-image translation. Current approaches [26, 114] learn an attention mask

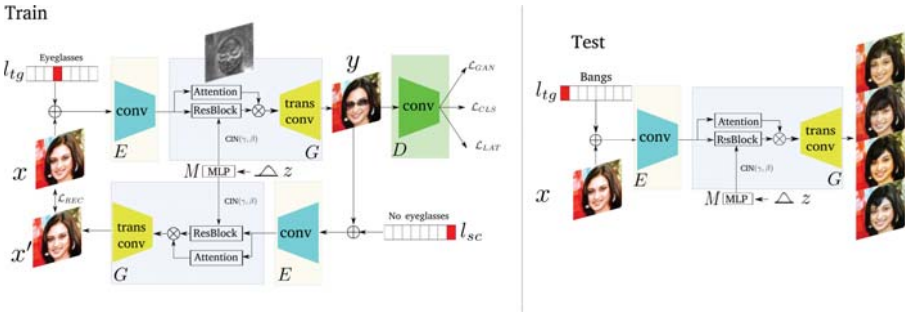


Figure 6.2 – **Model architecture**. (*Left*) The proposed approach is composed of two main parts: a discriminator D to distinguish the generated images and the real images; and the set of the encoder E, multilayer perceptron M and the generator G, containing the attention block, residual blocks with CIN, and the transposed convolutional layers. (*Right*) At test time, we can generate multiple plausible translations in the desired domain using a single model.

to enforce the translation to focus only on the objects of interest and preserve the background area. GANimation [131] uses *action units* to choose regions from the input images that are relevant for facial animation. These methods exploit attention mechanisms at the image level. Our method, on the other hand, learns feature-wise attention maps, which enables us to control which features are modified during translation. Therefore, our attention maps are highly effective at restricting the translation to change only domain-specific areas (e.g. forehead region when modifying the 'bangs' attribute).

6.3 Scalable and Diverse Image Translation

Our method must be able to perform multi-domain image-to-image translation. We aim to learn a model with both scalability and diversity. By scalability we refer to the property that a single model can be used to perform translations between multiple domains. By diversity we refer to the property that given a single input image, we can obtain multiple plausible output translations by sampling from a random variable.

6.3.1 Method Overview

Here we consider two domains: source domain $\mathscr{X} \subset \mathbb{R}^{H \times W \times 3}$ and target domain $\mathscr{Y} \subset \mathbb{R}^{H \times W \times 3}$ (it can trivially be extended to multiple domains). As illustrated in Figure 6.2, our framework is composed of four neural networks: encoder E, generator G, multilayer perceptron M, and discriminator D. Let $x \in \mathscr{X}$ be the input source image and $y \in \mathscr{Y}$ the target output, with corresponding labels $l_{sc} \in \{1,\dots,C\}$ for the source and $l_{tg} \in \{1,\dots,C\}$ for the target. In addition, let $z \in \mathbb{R}^Z$ be the latent code, which is sampled from a Gaussian distribution.

An overview of our method is provided in Figure 6.2. To address the problem of scalability we introduce the target domain as a conditioning label to the encoder, $E(x, l_{tg})$. The diversity is introduced by the latent variable z, which is mapped to the input parameters of a Conditional Instance Normalization (CIN) layer [40] by means of the multilayer perceptron $M(z)$. The CIN learns an additive (β) and a multiplicative term (γ) for each feature layer. Both the output of the encoder E and the multilayer perceptron M are used as input to the generator $G(E(x, l_{tg}), M(z))$. The generator G outputs a sample y of the target domain. Sampling different z results into different output results y. The unpaired domain translation is enforced by a cycle consistency [81, 181, 206]: taking as input the output y and the source category l_{sc}, we reconstruct the input image x as $G(E(G(E(x, l_{tg}), M(z)), l_{sc}), M(z))$. The encoder E, the multilayer perceptron M, and the generator G are all shared.

The function of the discriminator D is threefold. It produces three outputs: $x \rightarrow \{D_{src}(x), D_{cls}(x), F_{rec}(x)\}$. Both $D_{src}(x)$ and $D_{cls}(x)$ represent probability distributions, while $F_{rec}(x)$ is a regressed code. The goal of $D_{src}(x)$ is to distinguish between real samples and generated images in the target domain. The auxiliary classifier $D_{cls}(x)$ predicts the target label and allows the generator to perform domain-specific output conditioned on it. This was found to improve the quality of the conditional GAN [124]. Similarly to previous methods [25, 68] we reconstruct the latent input code in the output $F_{rec}(x)$. This was found to lead to improved diversity. Note that F_{rec} is just used for generated samples, as F_{rec} aims to reconstruct the latent code, which is not defined for real images.

We shortly summarize here the differences of our method with respect to the most similar approaches. StarGAN [31] can also generate outputs on multiple domains, but: (1) it learns a scalable but deterministic model, while our method additionally obtains diversity via the latent code; (2) we explicitly exploit an attention mechanism to focus the generator on the object of interest. Comparing against both MUNIT [68] and DRIT [93], which perform diverse image-to-image translation but without being scalable, our method: (1) employs the domain label to control the target domain, allowing to conduct image-to-image translation among multiple domains with a single generator; (2) avoids the need for domain-specific

style encoders, effectively saving computational resources; (3) considers attention to avoid undesirable changes in the translation; and (4) experimentally proves that the bias of CIN is the key factor to make the generator achieve the diversity, whereas the multiplicative term was only found to play a minor role.

6.3.2 Training Losses

The full loss function consists of several losses: the *adversarial loss* that discriminates the distribution of synthesized data and the real distribution in target domain, *domain classification loss* which contributes to the model $\{E, G\}$ to learn the specific attribute for a given target label, the *latent code reconstruction loss* regularizes the latent code to improve diversity and avoids the problem of partial mode collapse, and the *image reconstruction loss* that guarantees that the translated image keeps the structure of the input images.

Adversarial loss. We employ GANs [54] to distinguish the generated images from the real images

$$\begin{aligned}\mathscr{L}_{GAN} &= \mathbb{E}_{x\sim\mathscr{X}}\left[\log D_{src}(x)\right] \\ &+ \mathbb{E}_{x\sim\mathscr{X},z\sim p(z)}\left[\log(1-D_{src}(G(E(x,l_{tg}),M(z))))\right],\end{aligned} \tag{6.1}$$

where the discriminator tries to differentiate between generated images from the generator and real images, while G tries to fool the discriminator taking the output of M and the output of E as input. The final loss function is optimized by the minimax game

$$\left\{E^*, G^*, D^*\right\} = \arg\min_{E,G}\max_{D}\mathscr{L}_{GAN}. \tag{6.2}$$

Domain classification loss. In this chapter, we consider Auxiliary Classifier GANs (AC-GAN) [124] to control domains. The discriminator aims to output a probability distribution over given input images y and domain label, in consequence E and G synthesize the domain-specific images. We share the discriminator model exc7pt for the last layer and optimize the triplet $\{E, G, D\}$ by the cross-entropy loss. The final domain classification loss for generated samples, real samples, and total are

$$\mathscr{L}_{FAKE}(E,G) = -\mathbb{E}_{x\sim\mathscr{X},z\sim p(z)}\left[\log\left(D_{cls}\left(l_{tg}|G(E(x,l_{tg}),M(z))\right)\right)\right], \tag{6.3}$$

$$\mathscr{L}_{REAL}(D) = -\mathbb{E}_{x\sim\mathscr{X}}\left[\log\left(D_{cls}\left(l_{sc}|x\right)\right)\right], \tag{6.4}$$

$$\mathscr{L}_{CLS} = \mathscr{L}_{REAL} + \mathscr{L}_{FAKE}, \tag{6.5}$$

respectively. Given domain labels l_{sc} and l_{tg} these objectives are able to minimize the classification loss so that the model explicitly generates domain-specific outputs.

Latent code reconstruction loss. The lack of constraints on the latent code results in the generated images suffering from partial mode collapse as the latent code is ignored. We use the discriminator to predict the latent code, which forces the network to use it for generation:

$$\mathscr{L}_{LAT}(E,G,D) = \mathbb{E}_{x\sim\mathscr{X},z\sim p(z)}\left[\|F_{rec}(x) - z\|_1\right] \tag{6.6}$$

Image reconstruction loss. Both adversarial loss and classification loss fail to keep the structure of the input. To avoid this, we formulate the image reconstruction loss as

$$\begin{aligned} y &= G\left(E\left(x, l_{tg}\right), M(z)\right), \\ x' &= G\left(E\left(y, l_{sc}\right), M(z)\right), \\ \mathscr{L}_{REC} &= \mathbb{E}_{x\sim\mathscr{X},x'\sim\mathscr{X}'}\left[\left\|x - x'\right\|_1\right]. \end{aligned} \tag{6.7}$$

Full Objective. The full objective function of our model is:

$$\begin{aligned} &\min_{E,G}\max_{D} \lambda_{GAN}\mathscr{L}_{GAN} + \lambda_{FAKE}\mathscr{L}_{FAKE} \\ &+ \lambda_{REAL}\mathscr{L}_{REAL} + \lambda_{LAT}\mathscr{L}_{LAT} + \lambda_{REC}\mathscr{L}_{REC} \end{aligned} \tag{6.8}$$

where λ_{GAN}, λ_{FAKE}, λ_{REAL}, λ_{LAT}, λ_{REC} are hyper-parameters that balance the importance of each iterm.

6.3.3 Attention-guided generator

The attention mechanism encourages the generator to locate the domain-specific area relevant to the target domain label. Let $e = E\left(x, l_{tg}\right)$ be the output of the encoder. We propose to localize the CIN operation by introducing an attention mechanism. Only part of the encoder output e should be changed to obtain the desired diversity. We separate the signal e into two parallel residual blocks T^c and T^a. The CIN is applied to the residual block according to $f = T^c(e, M(z))$. We estimate the attention with a separate residual block according to $a = T^a(e)$. We

then combine the original encoder output and the CIN output using attention:

$$h = (1 - a) \cdot e + a \cdot f. \tag{6.9}$$

In [131], an *attention loss* regularizes the attention maps, since they quickly saturate to 1. In contrast, we employ the attention in the bottleneck features, and experimentally prove that the attention masks can be easily learned. This makes the task easier due to lower resolution in the bottleneck, and avoids the need to tune the attention hyperparameter. Finally, our attention mechanism does not add any new terms to the overall optimization loss in (6.8).

6.4 Experimental setup

Training setting. Our model is composed of four sub-networks: encoder E, multilayer perceptron M, generator G, and discriminator D. The encoder contains 3 convolutional layers and 6 blocks. Each convolutional layer uses 4×4 filters with stride 2, except for the first one which uses 7×7 with stride 1, and each block contains two convolutional layers with 3×3 filters and stride of 1. M consists of two fully connected layers with 256 and 4096 units. The generator G comprises ResBlock layers, attention layers and two fractionally strided convolutional layers. The ResBlock consists of 6 residual blocks, as in the encoder E, but including CIN layers. The CIN layers take the output of E and the ouput of the M as input. Except for six blocks like the CIN layers, the attention layers also use additional convolutional layers with sigmoid activations on top. For the discriminator D, we use six convolutional layers with 4×4 and stride 2, followed by three parallel sub-networks, each of them containing one convolutional layer with 3×3 filters and stride 1, except for the branch to output F_{rec} which uses an additional fully connected layer from 32 units to 8. Note how M adds around 1M parameters to the architecture.

All models are implemented in PyTorch [128] and released[2]. We randomly initialize the weights following a Gaussian distribution, and optimize the model using Adam [82] with batch size 16 and 4 for face and non-face datasets, respectively. The learning rate is 0.0001, followed the exponential decay rates $\left(\beta_1, \beta_2\right) = (0.5, 0.999)$. In all experiments, we use the following hyper-parameters: $\lambda_{GAN} = 10$, $\lambda_{FAKE} = 1$, $\lambda_{REAL} = 1$, $\lambda_{LAT} = 10$ and $\lambda_{REC} = 800$. We use Gaussian noise to the latent code with zero mean and a standard deviation of 1.

[2]The codes are available at https://github.com/yaxingwang/SDIT

6.4.1 Datasets

We consider several datasets to evaluate our models. In order to verify the generality of our method, the datasets were chosen to cover a variety of cases, including faces (CelebA), object (Color), and scenes (Artworks).

CelebA [105]. The Celeb Faces Attributes is a face dataset of celebrities with 202,599 images and 40 attribute labels per face. To explicitly preserve the face ratio, we crop the face size of 178×218 and resize it to 128×128. We leave out 2000 random images for test and train with the rest.

Color dataset [186]. We use the dataset collected by Yu *et.al* [186], which consists of 11 color labels, each category containing 1000 images. In order to easily compare to the non-scalable baselines which need train one independent model for each domain pair, we use only four colors (*green*, *yellow*, *blue*, *orange*). We resize all images to 128×128. We collected 3200 images for the train set and 800 images for the test set.

Artworks [206]. We also illustrate SDIT in an artwork setting [206]. This includes real images (*photo*) and three artistic styles (*Monet*, *Ukiyo-e*, and *Cezanne*). The training set contains 3000 (photo), 700 (Ukiyo-e), 500 (Cezanne) and 1000 (Monet) images, while the test set are: 300 (photo), 100 (Ukiyo-e), 100 (Cezanne) and 200 (Monet) images. All image are resized to 256×256.

6.4.2 Evaluation Metrics

To validate our approach, we consider the three following metrics.

LPIPS. In this chapter, LPIPS [197] is used to compute the similarity of pairs of images from the same attribute. LPIPS takes larger values if the generator has more diversity. In our setting, we generate 10 samples given an input image via different random codes.

ID distance. The key point of face mapping is to preserve the *identity* of the input, since an identity change is unacceptable for this task. To measure whether two images depict the same identity, we consider *ID distance* [169], which represents the difference in identity between pairs of input and translated faces. More concretely, given a pair of input and output faces, we extract the identity features represented by the VGGFace [127] network, and compute the distance between these features. VGGFace is trained on a large face dataset and is robust to appearance changes (e.g. illumination, age, expression, etc.). Therefore, two images of the same person should have a very small value. We only use this evaluation metric for CelebA. We use all 2000 test images as input and generate 10 output images, which in total amounts to 20,000 pairs.

Reverse classification. One of the methods to evaluate conditional image-to-

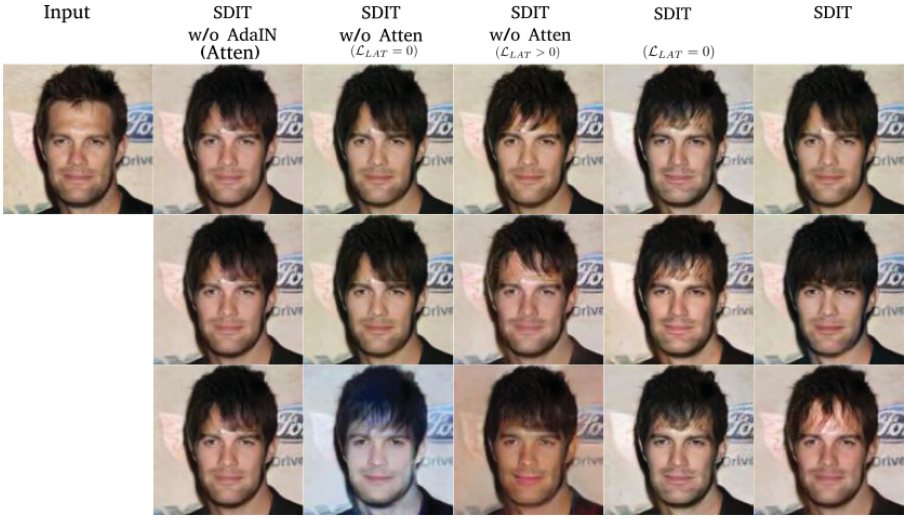


Figure 6.3 – Ablation study of different variants of our method. We show results for the face task of adding 'bangs'. We display three random outputs for each variant of the method.

image translation is to train a reference classifier on real images and test it on generated images [173, 175]. The reference classifier, however, fails to evaluate diversity, since it may still report a high accuracy even when the generator encounters mode-collapse for a specific domain, as shown on the third column of Figure 6.3. Following [150, 175], we use the *reverse classifier* which is trained using translated images for each target domain and evaluated on real images for which we know the label. Lower classification errors indicate more realistic and diverse translated images.

6.5 Experimental Results

In Section 6.5.1 we introduce several baselines against which we compare our model, as well as multiple variants of our model. Next, we evaluate the model on faces in Section 6.5.2. Finally, in Section 6.5.3 and Section 6.5.4, we analyze the generality of the model to color translation and scene translation.

Method	Atten	CIN	$\mathscr{L}_{LAT}$	ID Distance	LPIPS
SDIT w/o CIN (Atten)	Y	N	N	0.061	0.408
SDIT w/o Atten ($\mathscr{L}_{LAT} = 0$)	N	Y	N	0.063	0.409
SDIT w/o Atten ($\mathscr{L}_{LAT} > 0$)	N	Y	Y	0.070	**0.432**
SDIT ($\mathscr{L}_{LAT} = 0$)	Y	Y	N	0.063	0.412
SDIT	Y	Y	Y	**0.060**	0.424

Table 6.1 – ID distance (lower, better) / LPIPS (higher, better) for different variants of our method. Atten: attention, Y: yes, N: no.

Method	Bangs	Age	Gender	Smiling	Wearing hat	Pale skin	Brown hair	Blond hair	Eyeglasses	Mouth open	Mean
StarGAN [31]	**0.067**/0.427	**0.065**/0.428	**0.068**/0.428	**0.061**/0.427	**0.075**/0.427	**0.064**/0.421	0.060/0.418	0.067/0.426	**0.066**/0.435	0.059/0.429	**0.065**/0.427
IcGAN [130]	0.118/0.430	0.097/0.431	0.094/0.430	0.121/0.430	0.102/0.429	0.10/0.430	0.127/0.424	0.113/0.421	0.097/0.425	0.116/0.438	0.108/0.432
SDIT	0.068/**0.456**	**0.065**/**0.447**	0.069/**0.444**	**0.061**/**0.449**	0.076/**0.458**	0.065/**0.439**	**0.058**/**0.443**	**0.067**/**0.442**	**0.066**/**0.458**	**0.058**/**0.457**	**0.065**/**0.451**
Real data	-/0.486	-/0.483	-/0.484	-/0.480	-/0.489	-/0.479	-/0.492	-/0.490	-/0.492	-/0.489	-/0.486

Table 6.2 – ID distance (lower, better) / LPIPS (higher, better) on CelebA dataset.

6.5.1 Baselines and variants

We compare our method with the following baselines. For all baselines, we use the authors' original implementations and recommended hyperparameters. We also consider different configurations of our proposed SDIT approach. In particular, we study variants with and without CIN, attention, and latent code reconstruction.

CycleGAN [206]. CycleGAN is composed of two pairs of domain-specific encoders and decoders. The full objective is optimized with an adversarial loss and a cycle consistency loss.

MUNIT [68]. MUNIT disentangles the latent distribution into the content space which is shared between two domains, and the style space which is domain-specific and aligned with a Gaussian distribution. At test time, MUNIT takes as input the source image and different style codes to achieve diverse outputs.

IcGAN [130]. IcGAN explicitly maps the input face into a latent feature, followed by a decoder which is conditioned on the latent feature and a target face attribute. In addition, the face attribute can be explicitly reconstructed by an inverse encoder.

StarGAN [31]. StarGAN shares the encoders and decoders for all domains. The full model is trained by optimizing the adversarial loss, the reconstruction loss and the cross-entropy loss, which controls that the input image is translated into a target image.

6.5.2 Face translation

We firstly conduct an experiment on the CelebA [105] dataset to compare against ablations of our full model. Next, we compare SDIT to the baselines. For this case,

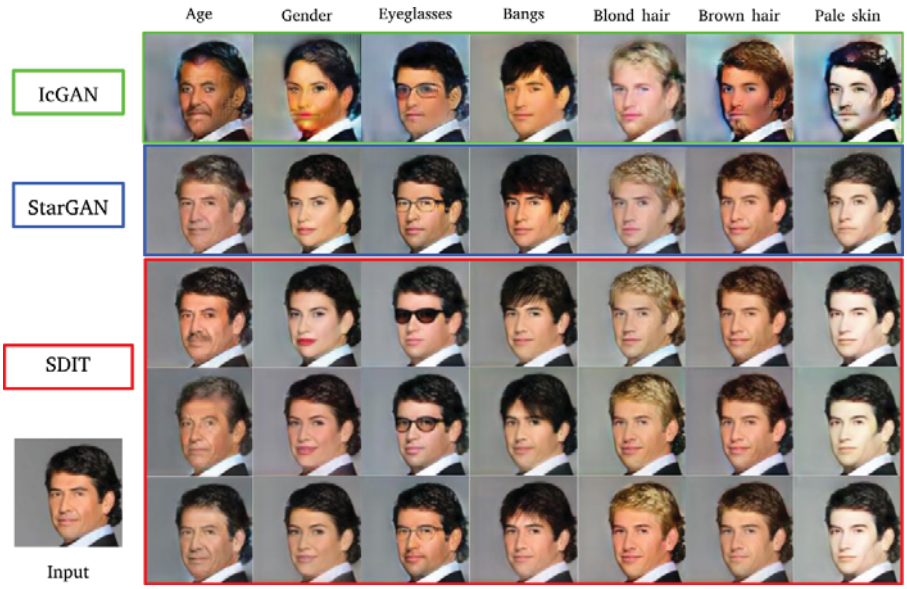


Figure 6.4 – Qualitative comparison to the baselines. The input face image is at the left bottom and the remaining columns show the attribute-specific mapped images. The first two lines show the translated results of the IcGAN [130] and StarGAN [31], respectively, while the remaining rows are from the proposed method.

we consider IcGAN and StarGAN, both of which show outstanding results for face synthesis.

Ablation study. We performed an ablation study comparing several variants of SDIT in terms of model diversity. We consider five attributes, namely *bangs*, *blond hair*, *brown hair*, *young*, and *male*. Figure 6.3 shows the translated images obtained with different variants of our method. As expected, SDIT with only *attention* (second column of Figure 6.3) fails to synthesize diverse outputs, since the model lacks the additional factors (e.g. noise) to control this. Both the third and fourth columns show that adding CIN to our method without attention generates diverse images. Their quality, however, is unsatisfactory and the model suffers from partial mode collapse, since CIN operates on the entire image, rather than being localized by the attention mechanism to the desired area (e.g. the bangs). Combining both CIN and attention but without the latent code reconstruction ($\mathscr{L}_{LAT} = 0$) leads to little diversity, as shown in the fifth column. Finally, our full model (last column) achieves the best results in terms of quality and diversity.

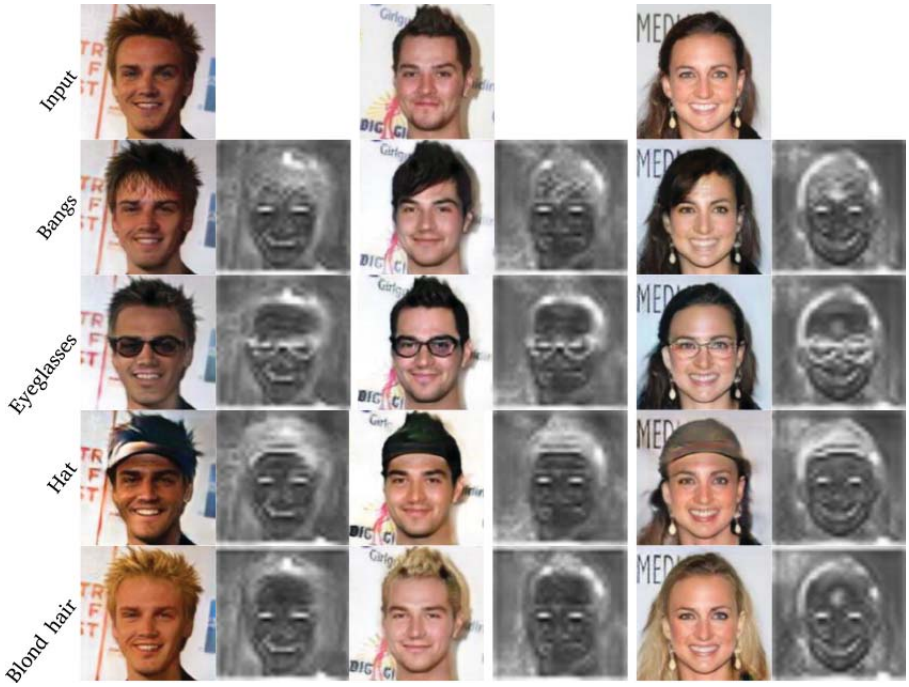


Figure 6.5 – Generated images and learned attention maps for three input images. For each of them we present multi-domain outputs and attribute-specific attention.

For quantitative evaluation, we report the results in terms of the ID distance and LPIPS. As shown in Table 6.1, the SDIT models without CIN or $\mathscr{L}_{LAT}$ generate less diverse outputs according to LPIPS scores. Using $\mathscr{L}_{LAT}$ without attention contributes to improve the diversity. It has a higher LPIPS, but this could be because it is adding unwanted diversity (e.g. the red lips in the fourth column of Figure 6.3). This may explain its higher ID distance. Combining both attention and $\mathscr{L}_{LAT} > 0$ (i.e. the full SDIT model) encourages the results to have better targeted diversity, as reported in the last row of Table 6.1. The preservation of identity is crucial for the facial attribute transfer task, and thus we keep both attention and the reconstruction loss in the following sections.

Attention. Figure 6.5 shows the attention maps for several translations from the face dataset. We note that our method explicitly learns the attribute-specific attention for a given face image (e.g. *eyeglasses*), and generates the corresponding

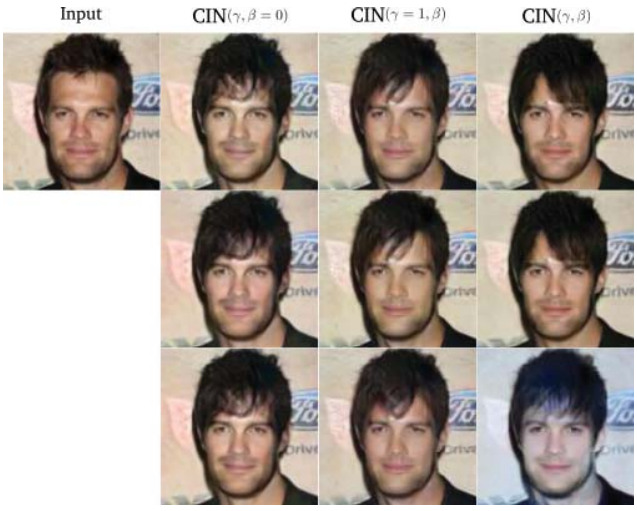


Figure 6.6 – Ablation study on CIN. We compare three cases: $(\gamma, \beta = 0)$, where γ is learnable; $(\gamma = 1, \beta)$, where β is learnable; and (γ, β), where both γ and β are learnable.

outputs. In this way, attention enables to modify only attribute-specific areas of the input image. This is a key factor to restrict the effect of the CIN, which otherwise would globally process the entire feature representation.

CIN learning. We explain here how CIN contributes to the diversity of the generator. In this experiment, we only consider CIN without attention nor latent code reconstruction. The operation performed by CIN on a feature e is given by:

$$\text{CIN}(e; z) = \gamma(z)\left(\frac{e-\mu(e)}{\delta(e)}\right) + \beta(z) \tag{6.10}$$

where e and z are the output of encoder E and latent code z, respectively; γ, β are affine parameters learned from M and $\mu(e)$, $\delta(e)$ are the mean and standard deviation. As shown in the second column of Figure 6.6, only learning γ fails to output diverse images, while only learning β already generates diverse results (third column of Figure 6.6), clearly indicating that β is the key factor to diversity. Updating the two parameters obtains a similar performance in this task. However, β could be ignored by the network. Therefore we introduced the latent code reconstruction loss, Eq. 6.6, which helps to avoid this.

Comparison against baselines. Figure 6.4 shows the comparison to the baselines

Figure 6.7 – Examples of scalable and diverse inference of multi-domain translations on (a) color dataset and (b) artworks dataset. In both cases, the first column is the input, the next three show results for CycleGAN [206], IcGAN [130], and StarGAN [31], respectively, followed by three samples from MUNIT [68] in next three columns and three samples from SDIT in the last three. Each row indicates a different domain.

on test data. We consider ten attributes: *bangs*, *blond hair*, *brown hair*, *young*, *male*, *mouth slightly open*, *smiling*, *pale skin*, *wearing hat*, and *eyeglasses*. Although both IcGAN and StarGAN are able to perform image-to-image translation to each domain, they fail to synthesize diverse outputs. Moreover, the performance of IcGAN is unsatisfactory and it fails to keep the personal identity. Our method not only enables the generation of realistic and diverse outputs, but also allows scalable image-to-image translation. Note that both StarGAN and our method use a single model. The visualization shows that scalability and diversity can be successfully integrated in a single model without conflict. Taking adding *bangs* as an example translation; the generated bangs with different directions do not impact the classification performance or the adversarial learning, in fact possibly contribute to the adversarial loss, since the CIN layer slightly reduces the compactness of the network, which increases the freedom of the generator.

As we can see in Table 6.2, our method obtains the best scores in both LPIPS and ID distance. In the case of LPIPS, the mean value of our method is 0.451, while IcGAN and StarGAN achieve 0.432 and 0.427 respectively. This clearly indicates that SDIT can successfully generate multimodal outputs using a single model. Moreover, the low ID distance indicates that SDIT effectively preserves the identity, achieving a competitive performance with StarGAN. Note that here we do not compare to CycleGAN and MUNIT because these methods require a single generator to be trained for each pair of domains. This is unfeasible for this task, because each attribute combination would require a different generator.

6.5.3 Object translation

The experiments in the previous section were conducted on a face dataset, in which all images have a relatively similar content and structure (a face on a background).

Method	Yellow	Blue	Green	Orange	Mean	Num E/G
CycleGAN	93.4/0.599	95.1/0.601	93.4/0.584	92.3/0.587	93.5/0.592	6/6
IcGAN	92.2/0.581	93.5/0.592	92.8/0.579	92.1/0.589	92.6/0.585	1/1
StarGAN	95.9/0.591	95.3/0.602	96.0/0.590	94.2/0.584	95.3/0.591	1/1
MUNIT	97.3/0.607	**97.1**/0.603	**97.2**/0.599	96.8/0.621	**97.2**/0.608	6/6
SDIT	**97.6**/**0.610**	96.6/**0.607**	**97.3**/**0.604**	97.1/**0.627**	**97.1**/**0.612**	1/1
Real image	98.5/0.652	98.6/0.652	97.8/0.653	98.8/0.652	98.4/0.652	-/-

Table 6.3 – Reverse classification accuracy (%) and LPIPS on the color dataset. For both metrics, the higher the better.

Here we consider the color object dataset to show that SDIT can be applied to datasets that lack a common structure. This dataset contains a wide range of different objects which greatly vary in shape, scale, and complexity. This makes the translation task more challenging.

Qualitative results. Figure 6.7(a) compares image-to-image translations obtained with CycleGAN [206], IcGAN [130], StarGAN [31], MUNIT [68] and the proposed method. We can see how SDIT clearly generates highly realistic and attribute-specific bags with different color shades, which is comparable to the results of MUNIT. Other baselines, however, only generate one color shade. The main advantage of SDIT is the *scalability*, as SDIT explicitly synthesizes the target color image (*yellow*, *green*, or *blue*) using a single generator.

Quantitative results. The qualitative observations above are validated here by quantitative evaluations. Table 6.3 compares the results of SDIT to the baseline methods. Our method outperforms both baseline methods on LPIPS despite only using a single model. For the classification accuracy, CycleGAN, IcGAN and StarGAN produce a lower score, since it is not able to generate diverse outputs for a given test samples. Both MUNIT and SDIT have a similar performance. However, for both CycleGAN and MUNIT training all pairwise translation would in case of N domains require $N \times (N-1)/2$ generators. Since we consider $N = 3$ here, we have trained a total of 6 generators for CycleGAN and MUNIT. The advantage of SDIT with respect to this non-scalable models would be even more evident for an increased number of domains.

6.5.4 Scene translation

Finally, we train our model on the photo and artworks dataset [206]. Differently from the model used for faces and color objects, here we consider the variant of our model without attention. This difference is due to the fact that previous datasets had a foreground that needed to be changed (object) and a fixed background, whereas in the scene case we need the generator to learn a global image translation instead

Method	Photo	Cezanne	Ukiyoe	Monet	Mean	Num E/G
CycleGAN	52.8/0.684	57.4/0.654	56.1/0.674	60.9/0.648	56.8/0.665	6/6
IcGAN	50.9/0.697	56.8/0.663	55.1/0.677	59.7/0.651	55.6/0.671	1/1
StarGAN	60.1/0.694	61.5/0.667	61.3/0.689	62.7/0.663	61.3/0.678	1/1
MUNIT	**66.2**/0.763	**67.9**/0.784	**67.2**/0.791	63.9/0.778	**66.3**/0.779	6/6
SDIT	65.6/**0.816**	63.4/**0.806**	65.3/**0.829**	**66.4**/**0.802**	65.1/**0.828**	1/1
Real image	70.2/0.856	72.4/0.874	69.9/0.884	71.7/0.864	71.1/0.869	-/-

Table 6.4 – Reverse classification accuracy (%) and LPIPS on the artworks dataset. For both metrics, the higher the better.

of a local one, and thus background must also be changed.

Figure 6.7(b) shows several representative examples of the different methods. The conclusions are similar to previous experiments: SDIT maps the input (*photo*) to other domains with diversity while using a single model. Table 6.4 also confirms this, showing how the proposed method achieves excellent scores with only one scalable model.

6.6 Conclusion

We have introduced SDIT to perform image-to-image translation with scalability and diversity using a simple and compact network. The key challenge lies in controlling the two functions separately without conflict. We achieve scalability by conditioning the encoder with the target domain label, and diversity by applying conditional instance normalization in the bottleneck. In addition, the use of attention on the latent represent further improves the performance of image translation, allowing the model to mainly focus on domain-specific areas instead of the unrelated ones. The model has limited applicability for domains with large variations (for example, faces and paintings in a single model) and works better when the domains have characteristics in common.

7 Conclusion

In this thesis, we proposed several improvements for image generation and image-to-image translation. In this chapter, we summarize the main conclusions of the methods proposed in this thesis. The chapter ends with some future research directions.

7.1 Conclusions

We have investigated visual synthesis from two directions: image generation and image-to-image translation. In the first part, we studied image generation in Chapter 2 and 3. Many visual tasks have benefited from transfer learning (e.g fine-tuning) when the amount of labelled data is not sufficient to optimize the millions of parameters required to perform these visual tasks. In Chapter 6, we explore the principles of transfer learning of both GANs and conditional GANs. However, directly conducting the fine-tuning for generative model easily results in overfitting when given limited target data, as we are updating all parameters to adapt to the small dataset. In Chapter 3, we propose an alternative approach, based on a miner network and a selector, to overcome this limitation. The miner network firstly explores the specific distribution that is helpful to generate the given target data. The selector is able to determinate the mixing coefficients of the various pre-trained generators in case we use multiple pre-trained models.

In the second part of the thesis, we focused on the problems of image-to-image translation in Chapter 4, 5, and 6. In Chapter 4, we study zero-pair image-to-image translation. The biases of image-to-image translation are investigated in Chapter 5. In Chapter 6, we proposed a novel method to conduct scalable and diverse image generation in a single model. The main conclusions for each chapter are summarized in the paragraphs below.

Chapter 2: Pretrained Generative Models for Domains with Limited Data. We explore the principles of transfer learning of GANs. We experimentally find that both GANs and conditional GANs profit from fine-tuning, resulting in a large improvement when given limited data. Interestingly, transferring the discriminator

is much more important than the generator. Using both pretrained generator and discriminator is to get the best performance. Notably, we find that a much higher density (images per class) may be more critical to lean good transferable features for fine-tuning of GANs, than the diversity (images per class).

Chapter 3: Effective Knowledge Transfer from GANs to Target Domains with Few Images. Directly performing fine-tuning for image generation is prone to result in overfitting, in this chapter we set out to avoid this problem and conduct effective knowledge transfer with few images. For this purpose, we proposed a mining operation that contributes to localize specific areas on the learned GAN manifold that are closer to a given target domain. Employing a mining network is helpful to conduct more effective and efficient fine-tuning, even with few target domain images. We also studied knowledge transfer from multiple pre-trained GANs as well as single GANs. We perform experiments on several complex datasets with various GAN architectures (BigGAN, Progressive GAN, and SNGAN), and demonstrated the generality of our method.

Chapter 4: Mix and Match Networks: Encoder-decoder Alignment for Zero-pair Image Translation. We proposed a new approach, called mix and match network (M &MNets), to conduct image-to-image translation between unseen domains (training stage) by employing the information learned from domains with paired data. We used several techniques to align the latent representation in the bottlenecks between unseen domains with the one between seen domains. We introduced autoencoders, latent consistency losses, and robust side information. In particular, we experimentally found that side information plays a key role to gain good cross-modal image translation, but standard side information (e.g skip connections) fails to work properly with unseen translations. Besides, the proposed method can also be applied to perform scalable image translation. Furthermore, we also investigated the specific limitation of the original M&MNets. We use a pseudo labeling technique to effectively use the shared features between unseen domains.

Chapter 5: Controlling Biases and Diversity in Diverse Image-to-image Translation. We studied the bias problem of learning image-to-image translation models, which occurs when the collected dataset is biased. The biases result in unwanted changes in generated images. In order to overcome this drawback, we introduced the use of semantic constraints, which are able to reduce the effects of biases. We experimentally find that the architecture of semantic constraint is important to preserve wanted properties of the input image. We validated the effectiveness of the proposed method on several complex datasets, including faces, objects and scenes.

Chapter 6: Scalable and Diverse Cross-domain Image Translation. None of the existing works on image-to-image translation is both scalable in the number of domains and able to generate diverse images. Using a single network, we presented a simple and compact framework which has both properties in a single model.

Specially, we obtain diversity by applying conditional instance normalization in the bottleneck. The scalability is achieved by using target domain labels in the encoder and decoder. Besides, we also investigated the attention in the bottleneck to improve the performance of image translation. The introduced attention helps our model to focus on domain-specific regions instead of the unrelated ones.

7.2 Future directions

We have identified several future work directions in the research lines on image generation and image-to-image translation. In Chapter 2 and 3, we proposed methods to steer information from pretrained models, which allows us to train models for domains with little labeled data. These methods, however, require to update all parameters to adapt to the target data. This operation often results in overfitting, even though the proposed method in Chapter 3 is able to reduce the problem. We are interested to identify specific parameters which are highly relevant for the target data domain, and only allow these to be changed during fine-tuning. This would reduce the problem of overfitting. Although some existing work studies this direction and only update the parameters of batch normalization [122], such an approach has limited ability to learn the distribution of target domain.

Recently, diverse and scalable image-to-image translation is attracting more and more attention. In Chapter 6, we improved this direction and achieved significant improvements in several datasets. The proposed method, however, has limited applicability for domains with large variations (for example, a single model which generates both faces and paintings). In the future, we aim to propose specific and effective image-to-image translation method to address this problem. For the implementation, we expect image-to-image translation network to focus on key-points of the input image, in order to allow for a wider range of structural changes between domains.

In addition, we addressed several issues of image-to-image translation in Chapter 4, 5 and 6. However, current methods still suffer from the problem that the model trained on a specific resolution fails to achieve interesting performance when we use a different image resolution at test time. We found that especially instance normalization is sensitive to the input resolution. We therefore are interested in new normalization methods that are invariant to the input resolution. These could then be used to translate between domains with different resolutions. Future research will focus on investigating this direction.

Summary of published works

Scientific Articles:

1. **Yaxing Wang**, Abel Gonzalez-Garcia, Joost van de Weijer and Luis Herranz. SDIT: Scalable and Diverse Cross-domain Image Translation. In proceedings of the ACM International Conference on Multimedia, pp. 1267-1276. 2019.
2. Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, **Yaxing Wang** and Bogdan Raducanu. Memory Replay GANs: learning to generate images from new categories without forgetting. In Advances in Neural Information Processing Systems, pp, 5962-5972. 2019.
3. **Yaxing Wang**, Joost van de Weijer and Luis Herranz. Mix and Match Networks: Encoder-Decoder Alignment for Zero-Pair Image Translation. In proceedings of the IEEE International Conference on Computer Vision and Pattern Recolonization, pp. 5467-5476. 2018.
4. **Yaxing Wang**, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia and Bogdan Raducanu. Transferring GANs: Generating Images from Limited Data. In proceedings of the European Conference on Computer Vision. pp. 220-236. 2018.
5. Caglayan Ozan, Aransa Walid, **Yaxing, Wang**, Marc Masana, Mercedes, García-Martínez, Fethi Bougares, Loïc Barrault, and Joost Van de Weijer. Does multimodality help human and machine for translation and image captioning? In the first Conference on Machine Translation. 2016.
6. **Yaxing Wang**, Lichao Zhang and Joost van de Weijer. Ensembles of Generative Adversarial Networks. In workshop on adversarial Training, NeurIPS. 2016.

Contributed Code and Datasets:

1. **SDIT: Scalable and Diverse Cross-domain Image Translation**
   https://github.com/yaxingwang/SDIT
2. **Mix and Match Networks: Encoder-Decoder Alignment for Zero-Pair Image Translation**
   https://github.com/yaxingwang/Mix-and-match-networks
3. **Transferring GANs: Generating Images from Limited Data**
   https://github.com/yaxingwang/Transferring-GANs

A Appendix

A.1 Transferring GANs: generating images from limited data

A.1.1 Distances between source and target data

Table A.1 shows the FID between the real images in the source and target datasets, which could be used as an estimation of which pre-trained GAN (on a source dataset) may be a good choice to adapt to a particular target dataset. In most of the cases, the lowest value in Table A.1 also corresponds to the lowest value in Table 1.

A.1.2 Model capacity

In order to check how important the capacity of the network is for transferring GAN features, we performed an additional experiment where we reduced the capacity of the network to half. We trained a source GAN with ImageNet, but in this case we reduced the number of filters in each layer to half its original value (with respect to the architecture used throughout our work, from WGAN-GP [25]). The model is then fine tuned with 10K images from LSUN Bedrooms. The results shown in Fig. A.1 suggest that also a lower capacity GAN adapting pre-trained features obtains

Table A.1 – Distance between source real data and target real data.

Distance	Source → Target ↓	ImageNet	Places	Bedrooms	CelebA
$\text{FID}\left(\mathscr{X}^{src}_{data}, \mathscr{X}^{tgt}_{data}\right)$	Flowers	**187.52**	292.36	270.09	317.21
	Kitchens	139.81	99.88	**66.54**	311.06
	LFW	266.50	326.76	318.98	**44.12**
	Cityscapes	205.04	**143.55**	221.65	349.28

significantly better results.

A.2 Effective knowledge transfer from GANs to target domains with few images

A.2.1 Architecture and training details

MNIST dataset. Our model contains a *miner*, *generator* and *discriminator*. For both unconditional and conditional GANs, we use the same framework [56] to design the generator and discriminator. The miner is composed of two fully connected layers with the same dimensionality as the latent space $|z|$. The visual results are computed with $|z| = 16$; we found that the quantitative results improved for larger $|z|$ and choose $|z| = 128$.

In MNIST, we consider the case where label c is a one-hot vector. We use the selector to predict the conditioning label. We randomly initialize the weights of the miner following a Gaussian distribution, and optimize the model using Adam [82] with batch size of 64. The learning rate of our model is 0.0004, with an exponential decay rates of $(\beta_1, \beta_2) = (0.5, 0.999)$. Note the same configuration is also used for the unconditional case.

CelebA Women, FFHQ Children and LSUN (Tower and Bedroom) Datasets. We design the generator and discriminator based on Progressive GANs [76]. Both networks use a multi-scale technique to generate high-resolution images. The miner comprises out of four fully connected layers (8-64-128-256-512), each of which is followed with a *relu* and *pixel normalization* except for last layer. We use a Gaussian distribution to initialize the miner, and optimize the model using Adam [82] with batch size of 4. The learning rate of our model is 0.0015, with an

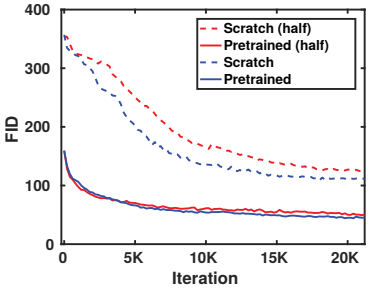


Figure A.1 – Model capacity.

exponential decay rates of $(\beta_1, \beta_2) = (0, 0.99)$.

FFHQ Face and Anime Face. We use the same network as [116], namely the SNGAN. The miner consists of three fully connected layers (8-32-64-128). We randomly initialize the weights following a Gaussian distribution. For this additional set of experiments, we use Adam [82] with a batch size of 8, following a hyper parameter learning rate of 0.0002 and exponential decay rate of $(\beta_1, \beta_2) = (0, 0.9)$.

Conditional GANs. For conditional GANs, we use the pretrained BigGAN [19]. We ignore the projection loss in the discriminator, since we do not have access to the label of the target data. The miner consists of two sub-networks: miner M^z and miner M^c. Both M^z and M^c are composed of four fully connected layers of sizes 128-128-128-128-120 and 128, respectively. We use Adam [82] with a batch size of 256, and learning rates of 0.0001 for miner and generator and 0.0004 for discriminator. The exponential decay rate is $(\beta_1, \beta_2) = (0, 0.999)$. We randomly initialize the weights following a Gaussian distribution.

A.2.2 MNIST experiment

We expand the MNIST experiments presented in Section 5.1 by providing a quantitative evaluation and including results on conditional GANs. As evaluation measures, we use FID (Section 5) and classifier error [150]. To compute classifier error, we first train a CNN classifier on real training data to distinguish between multiple classes (e.g. digit classifier). Then, we classify the generated images that should belong to a particular class and measure the error as the percentage of misclassified images. This gives us an estimation of how realistic and accurate the generated images are in the context of targeted generation.

The conditional architecture in this experiment (Section A.2.1) conditions by concatenating to the input noise z a one-hot vector c indicating the target class of the image. We extend MineGAN to this type of pretrained conditional models by considering each possible conditioning as an independently trained generator. Given a conditional generator $G(c, z)$, we consider $G(i, z)$ as G_i and apply the presented MineGAN approach on the family $\{G(i, z) |\ i = 1, ..., N\}$. The resulting selector now chooses among the N classes of the model rather than among N pretrained models, but the rest of the MineGAN training remains the same, including the training of N independent miners.

Table A.2 presents the results for both unconditional and conditional models, using a noise length of $|z| = 128$. The relatively low error values indicate that the miner manages to identify the correct regions for generating the target digits. The conditional model offers better results than the unconditional one by selecting the target class more often. We can also observe that the off-manifold task is more difficult than the on-manifold task, as indicated by the higher evaluation scores.

Table A.2 – Quantitative results of mining on MNIST, expressed as FID / classifier error.

d	On-manifold		Off-manifold	
	Unconditional	Conditional	Unconditional	Conditional
0	13.4 / 2.5	12.6 / 0.7	21.3 / 2.8	15.6 / 1.1
1	13.1 / 1.7	12.6 / 1.9	15.9 / 2.5	14.8 / 2.1
2	14.6 / 6.3	12.8 / 2.7	23.1 / 5.2	18.2 / 3.6
3	14.1 / 10.1	13.3 / 1.6	22.8 / 7.3	14.2 / 1.5
4	14.7 / 6.4	13.4 / 1.2	23.4 / 6.3	15.3 / 4.2
5	13.1 / 9.3	11.7 / 2.1	21.9 / 10.9	17.2 / 5.7
6	13.4 / 2.8	14.3 / 1.8	24 / 3.1	15.8 / 1.6
7	12.9 / 3.2	14.2 / 1.8	24.8 / 4.9	16.3 / 2.6
8	14.2 / 7.5	14.7 / 5.5	25.7 / 9.8	18.7 / 5.6
9	11.3 / 6.8	11.2 / 2.9	12.5 / 7.4	16.3 / 3.5
Average	13.5 / 5.7	13.1 / 2.2	21.5 / 6.0	16.2 / 3.2

However, the off-manifold scores are still reasonably low, indicating that the miner manages to find suitable regions from other digits by mining local patterns shared with the target. Overall, these results indicate the effectiveness of mining on MNIST for both types of targeted image generation. In addition, in Fig. A.2 we have added a visualization for the off-manifold MNIST classes which were not already shown in Fig. 2.

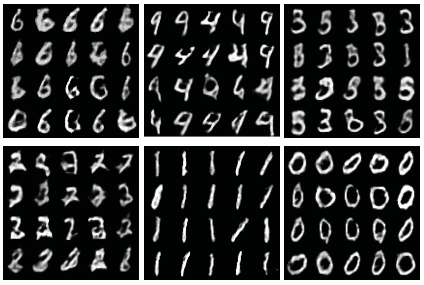


Figure A.2 – Results for unconditional off-manifold generation of digits ‘6’, ‘4’, ‘3’, ‘2’, ‘1’, ‘0’.

A.2.3 Further results on CelebA

We provide additional results for the on-manifold experiment CelebA→FFHQ women in Fig. A.3, and the off-manifold CelebA→FFHQ children in Fig. A.4. In

addition, we have also performed an on-manifold experiment with CelebA→CelebA women, whose results are provided in Fig. **??**.

A.2.4 Further results for LSUN

We provide additional results for the experiment ({bus, car}) → Red vehicles in Fig. A.7.

We also provide additional results for the experiment {Bedroom, Bridge, Church, Kitchen} → Tower/Bedroom in Fig. A.8.

When applying MineGAN to multiple pretrained GANs, we use one of the domains to initialize the weights of the critic. In Fig. A.8 we used *Church* to initialize the critic in case of the target set *Tower*, and *Kitchen* to initialize the critic for the target set *Bedroom*. We found this choice to be of little influence on the final results. When using *Kitchen* to initialize the critic for target set *Tower* results change from 62.4 to 61.7. When using *Church* to initialize the critic for target set *Bedroom* results change from 54.7 to 54.3.

A.3 Cross-modal alignment for zero-pair image-to-image translation

A.3.1 Appendix: Network architecture on RGB-D or RGB-D-NIR dataset

Table A.3 shows the architecture (convolutional and pooling layers) of the encoders used in the cross-modal experiment. Tables A.4 and A.5 show the corresponding decoders. Table A.6 shows the discriminator used for RGB. Every convolutional layer of the encoders, decoders and the discriminator is followed by a batch normalization layer and a ReLU layer (LeakyReLU for the discriminator). The only exception is the RGB encoder, which is initialized with weights from the VGG16 model pretrained on imageNet [153] and does not use batch normalization. The used abbreviations are shown in Table A.10.

A.3.2 Appendix: Network architectures

We use several datasets to verify the generality of our method, including object (Color) and scenes (Artworks).

Color dataset [188]. We consider the object dataset for color which is collected by [188], which includes 11 color labels, each category containing 1000 images. We resize all images to 128 × 128.

Layer	Input →Output	Kernel, stride
conv1 (RGB)	[6,256,256,3] → [6,256,256,64]	[3,3], 1
conv1 (Depth)	[6,256,256,1] → [6,256,256,64]	[3,3], 1
conv1 (NIR)	[6,256,256,1] → [6,256,256,64]	[3,3], 1
conv1 (Segm.)	[6,256,256,14] → [6,256,256,64]	[3,3], 1
conv2	[6,256,256,64] → [6,256,256,64]	[3,3], 1
pool2 (max)	[6,256,256,64] → [6,128,128,64]+indices2	[2,2], 2
conv3	[6,128,128,64] → [6,128,128,128]	[3,3], 1
conv4	[6,128,128,128] → [6,128,128,128]	[3,3], 1
pool4 (max)	[6,128,128,128] → [6,64,64,128]+indices4	[2,2], 2
conv5	[6,64,64,128] → [6,64,64,256]	[3,3], 1
conv6	[6,64,64,256] → [6,64,64,256]	[3,3], 1
conv7	[6,64,64,256] → [6,64,64,256]	[3,3], 1
pool7 (max)	[6,64,64,256] → [6,32,32,256]+indices7	[2,2], 2
conv8	[6,32,32,256] → [6,32,32,512]	[3,3], 1
conv9	[6,32,32,512] → [6,32,32,512]	[3,3], 1
con10	[6,32,32,512] → [6,32,32,512]	[3,3], 1
pool10 (max)	[6,32,32,512] → [6,16,16,512]+indices10	[2,2], 2
conv11	[6,16,16,512] → [6,16,16,512]	[3,3], 1
conv12	[6,16,16,512] → [6,16,16,512]	[3,3], 1
conv13	[6,16,16,512] → [6,16,16,512]	[3,3], 1
relu13	[6,16,16,512] → [6,16,16,512]	-, -
pool13 (max)	[6,16,16,512] → [6,8,8,512]+indices13	[2,2], 2

Table A.3 – The architecture of the encoder of RGB, depth, NIR and semantic segmentation.

Artworks [206]. We also illustrate M&MNet in an artwork setting. This includes real images (*photo*) and four artistic styles (*Monet*, *van Gogh*, *Ukiyo-e* and *Cezanne*). The the set contains 3000 (photo), 800 (Ukiyo-e), 500 (van Gogh), 600 (Cezanne) and 1200 (Monet) images. All images are resized to 256×256.

We consider Adam [82] with a batch size of 4, using a learning rate of 0.0002. The network is initialized using a Gaussian distribution with zero mean and a standard deviation of 0.5. We only use adversarial loss to train our model.

Tables A.7-A.9 show the architectures of the encoder, image decoder and discriminator used in the cross-modal experiment. The following tables only show the image size of 128×128, while for artworks dataset it is same architecture except for image resolution. The used abbreviations are shown in Table A.10.

A.3.3 Appendix: Network architecture for the Flower dataset

Flower dataset [121]. The Flower dataset consists of 102 categories. We consider 10 categories(*passionflower*, *petunia*, *rose*, *wallflower*, *watercress*, *waterlily*, *cyclamen*,

foxglove, *frangipani*, *hibiscus*). Each category includes between 100 and 258 images. we resize the image to 128×128.

Similarly, we optimize our model by means of using Adam [82], the batch size of 4 and a learning rate of 0.0002. We initialize hyperparameters using a Gaussian distribution with zero mean and a standard deviation of 0.5. We use adversarial loss and $L2$ to train Θ_3, and only $L2$ for Θ_1 and Θ_2.

Tables A.11 and A.12 detail the architecture of the encoder and decoder, respectively, of the two single channel modalities Θ_1 and Θ_2. The encoder and decoder for the third modality Θ_3 are analogous, just adapted to three input and output channels, respectively. For Θ_3 we also use the discriminator detailed in Table A.9.

A.4 Controlling biases and diversity in diverse image-to-image translation

Tables A.13-A.18 show the architectures of the content encoder, style encoder, image decoder and discriminator used in the cross-modal experiment. The used abbreviations are shown in Table A.19.

Figure A.3 – (CelebA→FFHQ women). Based on pretrained *Progressive GAN*.

Figure A.4 – (CelebA→ FFHQ children). Based on pretrained *Progressive GAN*.

Figure A.5 – (Top) 100 women faces from HHFQ dataset. (Bottom) training of model from scratch: the images start with low quality and iteratively overfit to a particular training image. Red boxes identify images which are remembered by the model trained from scratch or from TransferGAN (see Fig. 4). Based on pretrained *Progressive GAN*.

Figure A.6 – 100 children faces from HHFQ dataset. Red boxes identify images which are remembered by the model trained from scratch (see Fig. 4). Based on pretrained *Progressive GAN*.

Figure A.7 – ({bus, car}) →red vehicles. Based on pretrained *Progressive GAN*.

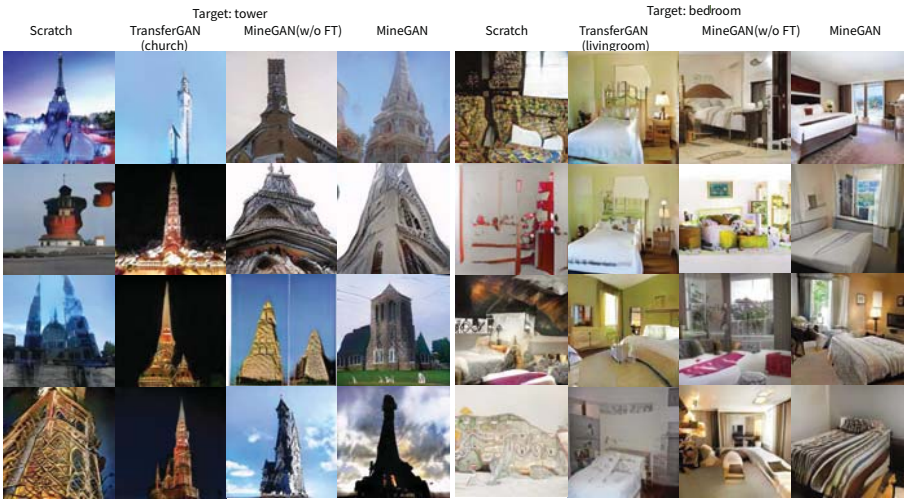


Figure A.8 – Results for unconditional GAN. (Top) (Livingroom, kitchen, bridge, church)→Tower. (Bottom) (Livingroom, kitchen, bridge, church)→Bedroom. Based on pretrained *Progressive GAN*.

layer	Input →Output	Kernel, stride
unpool1	indices13 + [6,8,8,512] → [6, 16, 16, 512]	[2, 2], 2
conv1	[6,16,16,512] → [6, 16, 16, 512]	[3,3], 1
BN1	[6,16,16,512] → [6, 16, 16, 512]	-, -
relu1	[6,16,16,512] → [6, 16, 16, 512]	-, -
conv2	[6,16,16,512] → [6, 16, 16, 512]	[3,3], 1
BN2	[6,16,16,512] → [6, 16, 16, 512]	-, -
relu2	[6,16,16,512] → [6, 16, 16, 512]	-, -
conv3	[6,16,16,512] → [6, 16, 16, 512]	[3,3], 1
BN3	[6,16,16,512] → [6, 16, 16, 512]	-, -
relu3	[6,16,16,512] → [6, 16, 16, 512]	-, -
unpool4	indices10 + [6,16,16,512] → [6, 32, 32, 512]	[2, 2], 2
conv4	[6,32,32,512] → [6, 32, 32, 512]	[3,3], 1
BN4	[6,32,32,512] → [6, 32, 32, 512]	-, -
relu4	[6,32,32,512] → [6, 32, 32, 512]	-, -
conv5	[6,32,32,512] → [6, 32, 32, 512]	[3,3], 1
BN5	[6,32,32,512] → [6, 32, 32, 512]	-, -
relu5	[6,32,32,512] → [6, 32, 32, 512]	-, -
conv6	[6,32,32,512] → [6, 32, 32, 256]	[3,3], 1
BN6	[6,32,32,512] → [6, 32, 32, 512]	-, -
relu6	[6,32,32,512] → [6, 32, 32, 512]	-, -
unpool7	indices7 + [6,32,32,256] → [6, 64, 64, 256]	[2, 2], 2
conv7	[6,64,64,256] → [6, 64, 64, 256]	[3,3], 1
BN7	[6,64,64,256] → [6, 64, 64, 256]	-, -
relu7	[6,64,64,256] → [6, 64, 64, 256]	-, -
conv8	[6,64,64,256]→ [6, 64, 64, 256]	[3,3], 1
BN8	[6,64,64,256] → [6, 64, 64, 256]	-, -
relu8	[6,64,64,256] → [6, 64, 64, 256]	-, -
conv9	[6,64,64,256]→ [6, 64, 64, 128]	[3,3], 1
BN9	[6,64,64,256] → [6, 64, 64, 256]	-, -
relu9	[6,64,64,256] → [6, 64, 64, 256]	-, -
unpool10	indices4 + [6,64,64,128] → [6, 128, 128, 128]	[2, 2], 2
conv10	[6,128,128,128] → [6, 128, 128, 128]	[3,3], 1
BN10	[6,128,128,128] → [6, 128, 128, 128]	-, -
relu10	[6,128,128,128] → [6, 128, 128, 128]	-, -
conv11	[6,128,128,128] → [6, 128, 128, 64]	[3,3], 1
BN11	[6,128,128,128] → [6, 128, 128, 128]	-, -
relu11	[6,128,128,128] → [6, 128, 128, 128]	-, -
unpool12	indices2 + [6,128,128,64] → [6, 256, 256, 64]	[2, 2], 2
conv12	[6,256,256,64] → [6, 256, 256, 64]	[3,3], 1
conv13 (Depth)	[6,256,256,64] → [6, 256, 256, 1]	[3,3], 1
conv13 (NIR)	[6,256,256,64] → [6, 256, 256, 5]	[3,3], 1
conv13 (Segm.)	[6,256,256,64] → [6, 256, 256, 14]	[3,3], 1

Table A.4 – The architecture of the decoder of depth, NIR and semantic segmentation.

layer	Input →Output	Kernel, stride
conv1	[6,8,8,512] → [6, 16, 16, 512]	[3, 3], 1
BN1	[6,16,16,512]→ [6, 16, 16, 512]	-, -
relu1	[6,16,16,512]→ [6, 16, 16, 512]	-, -
conv2	[6,16,16,512] → [6, 32, 32, 256]	[3, 3], 1
BN2	[6,32,32,256]→ [6, 32, 32, 256]	-, -
relu2	[6,32,32,256]→ [6, 32, 32, 256]	-, -
conv3	[6,32,32,256] → [6, 64, 64, 128]	[3, 3], 1
BN3	[6,64,64,128]→ [6, 64, 64, 128]	-, -
relu3	[6,64,64,128]→ [6, 64, 64, 128]	-, -
conv4	[6,64,64,128] → [6, 128, 128, 64]	[3, 3], 1
BN4	[6,128,128,64]→ [6, 128, 128, 64]	-, -
relu4	[6,128,128,64]→ [6, 128, 128, 64]	-, -
conv5	[6,128,128,64]→ [6, 256, 256, 3]	[3, 3], 1

Table A.5 – The architecture of the decoder of RGB

layer	Input →Output	Kernel, stride
deconv1	[6, 256, 256, 3] → [6, 128, 128, 64]	[5, 5], 2
lrelu1	[6, 128, 128, 64] → [6, 128, 128, 64]	-, -
deconv2	[6, 128, 128, 64] → [6, 64, 64, 128]	[5, 5], 2
lrelu2	[6, 64, 64, 128] → [6, 64, 64, 128]	-, -
deconv3	[6, 64, 64, 128] → [6, 32, 32, 256]	[5,5], 2
lrelu3	[6, 32, 32, 256] → [6, 32, 32, 256]	-, -
deconv4	[6, 32, 32, 256] → [6, 16, 16, 512]	[5,5], 2

Table A.6 – RGB discriminator.

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,3] → [4,128, 128, 64]	[7,7], 1, 3
IN1	[4,128, 128, 64] → [4,128, 128, 64]	-, -, -
pool1 (max)	[4,128, 128, 64] →[4,64, 64, 64]+indices1	[2,2], 2, -
conv2	[4,64, 64,64] → [4,64, 64,128]	[7,7], 1, 3
IN2	[4,64, 64,128] → [4,64, 64,128]	-, -, -
pool2 (max)	[4,64, 64,128] →[4,32, 32,128]+indices2	[2,2], 2, -
conv3	[4,32, 32,128] → [4,32, 32,256]	[7,7], 1, 3
IN3	[4,32, 32,256] → [4,32, 32,256]	-, -, -
pool3 (max)	[4,32, 32,256] → [4,16, 16,256]+indices3	[2,2], 2, -
RB(IN)4-9	[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3

Table A.7 – The architecture of the encoder for 128 × 128 input.

Layer	Input →Output	Kernel, stride, pad
RB(IN)1-6	[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3
unpool1	indices3 + [4,16, 16,256] → [4,32, 32,256]	[2, 2], 2, -
conv1	[4,32, 32,256] → [4,32, 32,128]	[7,7], 1, 3
IN1	[4,32, 32,128] → [4,32, 32,128]	-, -, -
unpool2	indices2 + [4,32, 32,128] → [4, 64, 64,128]	[2, 2], 2, -
conv2	[4, 64, 64,128] → [4, 64, 64,64]	[7,7], 1, 3
IN2	[4, 64, 64,64]→ [4, 64, 64,64]	-, -, -
unpool3	indices1 + [4, 64, 64,64] → [4, 128, 128,64]	[2, 2], 2, -
conv3	[4, 128, 128,64] → [4, 128, 128,3]	[7,7], 1, 3

Table A.8 – The architecture of the decoder for 128 × 128 output.

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,3] → [4,64, 64,64]	[4,4], 2, 1
lrelu1	[4,64, 64,64] → [4,64, 64,64]	-, -, -
conv2	[4,64, 64,64] → [4,32, 32,128]	[4,4], 2, 1
lrelu2	[4,32, 32,128] → [4,32, 32,128]	-, -, -
conv3	[4,32, 32,128] → [4,16, 16,256]	[4,4], 2, 1
lrelu3	[4,16, 16,256] → [4,16, 16,256]	-, -, -
conv4	[4,16, 16,256] → [4,8, 8,512]	[4,4], 2, 1
lrelu4	[4,8, 8,512] →[4,8, 8,512]	-, -, -
conv5	[4,8, 8,512] →[4,8, 8,1]	[1,1], 1, 0

Table A.9 – Architecture for the discriminator Loss specification for 128 × 128 input.

Abbreviation	Name
pool	pooling layer
unpool	unpooling layer
lrelu	leaky relu layer
conv	convolutional layer
linear	fully connection layer
BN	batch normalization layer
IN	instance normalization layer
RB(IN)	residual block layer using instance normalization

Table A.10 – Abbreviations used in other tables.

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,1] → [4,128, 128, 64]	[7,7], 1, 3
IN1	[4,128, 128, 64] → [4,128, 128, 64]	-, -, -
pool1 (max)	[4,128, 128, 64] →[4,64, 64, 64]+indices1	[2,2], 2, -
conv2	[4,64, 64,64] → [4,64, 64,128]	[7,7], 1, 3
IN2	[4,64, 64,128] → [4,64, 64,128]	-, -, -
pool2 (max)	[4,64, 64,128] →[4,32, 32,128]+indices2	[2,2], 2, -
conv3	[4,32, 32,128] → [4,32, 32,256]	[7,7], 1, 3
IN3	[4,32, 32,256] → [4,32, 32,256]	-, -, -
pool3 (max)	[4,32, 32,256] → [4,16, 16,256]+indices3	[2,2], 2, -
RB(IN)4-9	[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3

Table A.11 – The architecture of the encoder of Θ_1 and Θ_2.

Layer	Input →Output	Kernel, stride, pad
RB(IN)1-6	[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3
unpool1	indices3 + [4,16, 16,256] → [4,32, 32,256]	[2, 2], 2, -
conv1	[4,32, 32,256] → [4,32, 32,128]	[7,7], 1, 3
IN1	[4,32, 32,128] → [4,32, 32,128]	-, -, -
unpool2	indices2 + [4,32, 32,128] → [4, 64, 64,128]	[2, 2], 2, -
conv2	[4, 64, 64,128] → [4, 64, 64,64]	[7,7], 1, 3
IN2	[4, 64, 64,64]→ [4, 64, 64,64]	-, -, -
unpool3	indices1 + [4, 64, 64,64] → [4, 128, 128,64]	[2, 2], 2, -
conv3	[4, 128, 128,64] → [4, 128, 128,1]	[7,7], 1, 3

Table A.12 – The architecture of the decoder for Θ_1 and Θ_2.

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,3] → [4,128, 128, 64]	[7,7], 1, 3
IN1	[4,128, 128, 64] → [4,128, 128, 64]	-, -, -
pool1 (max)	[4,128, 128, 64] →[4,64, 64, 64]+indices1	[2,2], 2, -
conv2	[4,64, 64,64] → [4,64, 64,128]	[7,7], 1, 3
IN2	[4,64, 64,128] → [4,64, 64,128]	-, -, -
pool2 (max)	[4,64, 64,128] →[4,32, 32,128]+indices2	[2,2], 2, -
conv3	[4,32, 32,128] → [4,32, 32,256]	[7,7], 1, 3
IN3	[4,32, 32,256] → [4,32, 32,256]	-, -, -
pool3 (max)	[4,32, 32,256] → [4,16, 16,256]+indices3	[2,2], 2, -
RB(IN)4-9	[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3

Table A.13 – Content encoder.

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,3] → [4,128, 128, 64]	[7,7], 1, 3
relu1	[4,128, 128, 64] →[4,64, 64, 64]	-, -, -
conv2	[4,64, 64,64] → [4,32, 32,128]	[4, 4], 2, 1
relu2	[4,32, 32,128] →[4,32, 32,128]	-, -, -
conv3	[4,32, 32,128] → [4,16, 16,256]	[4,4], 2, 1
relu3	[4,16, 16,256] → [4,16, 16,256]	-, -, -
GAP	[4,16, 16,256] → [4,1, 1,256]	-, -,-
conv4	[4,1, 1,256] → [4,1, 1,8]	[1, 1],1,0

Table A.14 – Style encoder.

Layer	Input →Output
linear1	[4, 8] → [4, 256]
relu1	[4, 256] →[4, 256]
linear2	[4, 256] → [4, 256]
relu2	[4, 256] →[4, 256]
linear3	[4, 256] → [4, 256]
reshape	[4, 256] →[4,1,1, 256]

Table A.15 – Networks for the estimation of the affine parameters (μ) that are used in the AdaIN layer. The parameters μ and σ scale and shift the normalized content, respectively. Note that μ and σ share the first two layers.

Layer	Input →Output
linear1	[4, 8] → [4, 256]
relu1	[4, 256] →[4, 256]
linear2	[4, 256] → [4, 256]
relu2	[4, 256] →[4, 256]
linear3	[4, 256] → [4, 256]
reshape	[4, 256] →[4,1,1, 256]

Table A.16 – Networks for the estimation of the affine parameters (σ) that are used in the AdaIN layer. The parameters μ and σ scale and shift the normalized content, respectively. Note that μ and σ share the first two layers.

Layer	Input →Output	Kernel, stride, pad
RB(AdaIN)1-6	(μ,σ) +[4,16, 16,256] → [4,16, 16,256]	[7,7], 1, 3
unpool1	indices3 + [4,16, 16,256] → [4,32, 32,256]	[2, 2], 2, -
conv1	[4,32, 32,256] → [4,32, 32,128]	[7,7], 1, 3
IN1	[4,32, 32,128] → [4,32, 32,128]	-, -, -
unpool2	indices2 + [4,32, 32,128] → [4, 64, 64,128]	[2, 2], 2, -
conv2	[4, 64, 64,128] → [4, 64, 64,64]	[7,7], 1, 3
IN2	[4, 64, 64,64]→ [4, 64, 64,64]	-, -, -
unpool3	indices1 + [4, 64, 64,64] → [4, 128, 128,64]	[2, 2], 2, -
conv3	[4, 128, 128,64] → [4, 128, 128,3]	[7,7], 1, 3

Table A.17 – Decoder (Image generator).

Layer	Input →Output	Kernel, stride, pad
conv1	[4,128, 128,3] → [4,64, 64,64]	[4,4], 2, 1
lrelu1	[4,64, 64,64] → [4,64, 64,64]	-, -, -
conv2	[4,64, 64,64] → [4,32, 32,128]	[4,4], 2, 1
lrelu2	[4,32, 32,128] → [4,32, 32,128]	-, -, -
conv3	[4,32, 32,128] → [4,16, 16,256]	[4,4], 2, 1
lrelu3	[4,16, 16,256] → [4,16, 16,256]	-, -, -
conv4	[4,16, 16,256] → [4,8, 8,512]	[4,4], 2, 1
lrelu4	[4,8, 8,512] →[4,8, 8,512]	-, -, -
conv5	[4,8, 8,512] →[4,8, 8,1]	[1,1], 1, 0

Table A.18 – Architecture for the discrim Loss specificationinator for 128 × 128 input. The discriminators for 64 × 64, and 32 × 32 use the same convolutional architecture.

Abbreviation	Name
pool	pooling layer
unpool	unpooling layer
lrelu	leaky relu layer
concat	concatenate layer
conv	convolutional layer
linear	fully connection layer
IN	instance normalization layer
GAP	global average pooling layer
RB(IN)	residual block layer using instance normalization
RB(AdaIN)	residual block layer using adaptive instance normalization

Table A.19 – Abbreviations used in other tables.

Bibliography

[1] https://en.janbharattimes.com/hollywood/avengers-endgame-finally-beat-james-cameron-avatar-to-become-the-highest-grossing-movie-ever. 2019.

[2] https://techcrunch.com/2019/06/11/internet-trends-report-2019. 2019.

[3] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 38(7):1425–1438, 2016.

[4] Yazeed Alharbi, Neil Smith, and Peter Wonka. Latent filter scaling for multimodal unsupervised image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1458–1466, 2019.

[5] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *International Conference on Machine Learning (ICML)*, 2018.

[6] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] Anonymous, Danbooru community, Gwern Branwen, and Aaron Gokaslan. Danbooru2018: A large-scale crowdsourced and tagged anime illustration dataset. https://www.gwern.net/Danbooru2018, 2019.

[8] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun 2018.

[9] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning representations (ICLR)*, 2017.

[10] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223, 2017.

[11] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 38(9):1790–1802, 2016.

[12] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 2017.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 2013.

[14] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning (ICML)*, pages 226–234, 2014.

[15] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[16] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[18] Behzad Bozorgtabar, Mohammad Saeed Rad, Hazım Kemal Ekenel, and Jean-Philippe Thiran. Learn to synthesize and synthesize to learn. *Computer Vision and Image Understanding*, 2019.

[19] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning representations (ICLR)*, 2019.

[20] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

[21] Cesar Cadena, Anthony R Dick, and Ian D Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, 2016.

[22] Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2940–2949, 2016.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 40(4):834–848, 2018.

[24] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. *arXiv preprint arXiv:1707.09405*, 2017.

[25] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[26] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *European Conference on Computer Vision (ECCV)*, pages 164–180, 2018.

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

[28] Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. A teacher-student framework for zero-resource neural machine translation. *arXiv preprint arXiv:1705.00753*, 2017.

[29] Yanhua Cheng, Xin Zhao, Rui Cai, Zhiwei Li, Kaiqi Huang, Yong Rui, et al. Semi-supervised multimodal deep learning for rgb-d object recognition. *AAAI International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[30] Wonwoong Cho, Sungha Choi, David Keetae Park, Inkyu Shin, and Jaegul Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[33] Ivo Danihelka, Balaji Lakshminarayanan, Benigno Uria, Daan Wierstra, and Peter Dayan. Comparison of maximum likelihood and gan-based training of real nvps. *arXiv preprint arXiv:1705.05263*, 2017.

[34] Hal Daumé III. Frustratingly easy domain adaptation. *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, 2007.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[36] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1486–1494, 2015.

[37] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, pages 647–655, 2014.

[38] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning representations (ICLR)*, 2017.

[39] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *International Conference on Learning representations (ICLR)*, 2017.

[40] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[41] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[42] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[43] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1657–1664, 2013.

[44] Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic label sharing for learning with many categories. *European Conference on Computer Vision (ECCV)*, pages 762–775, 2010.

[45] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*, 2016.

[46] Yanwei Fu, Tao Xiang, Yu-Gang Jiang, Xiangyang Xue, Leonid Sigal, and Shaogang Gong. Recent advances in zero-shot recognition. *arXiv preprint arXiv:1710.04837*, 2017.

[47] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

[48] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[49] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[50] J-M Geusebroek, Rein Van den Boomgaard, Arnold W. M. Smeulders, and Hugo Geerts. Color invariance. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 23(12):1338–1350, 2001.

[51] Edoardo Giacomello, Daniele Loiacono, and Luca Mainardi. Transfer brain mri tumor segmentation models across modalities with adversarial networks. *arXiv preprint arXiv:1910.02717*, 2019.

[52] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

[53] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2018.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[55] Guillermo L Grinblat, Lucas C Uzal, and Pablo M Granitto. Class-splitting generative adversarial networks. *arXiv preprint arXiv:1709.07359*, 2017.

[56] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 5769–5779, 2017.

[57] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[59] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision (ECCV)*, volume 11207, pages 793–811. Springer, 2018.

[60] Luis Herranz, Shuqiang Jiang, and Xiangyang Li. Scene recognition with cnns: objects, scales and dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 571–579, 2016.

[61] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[62] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 826–834, 2016.

[63] Judy Hoffman, Saurabh Gupta, Jian Leong, Sergio Guadarrama, and Trevor Darrell. Cross-modal adaptation for rgb-d detection. In *International Conference on Robotics and Automation (ICRA)*, pages 5032–5039. IEEE, 2016.

[64] Ayanna Howard, Cha Zhang, and Eric Horvitz. Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems. In *2017 IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 1–7. IEEE, 2017.

[65] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Deep transfer metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 325–333. IEEE, 2015.

[66] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[67] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 4, 2017.

[68] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018.

[69] Daniel Jiwoong Im, He Ma, Graham Taylor, and Kristin Branson. Quantitatively evaluating gans with divergences proposed for training. In *International Conference on Learning representations (ICLR)*, 2018.

[70] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[71] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 4020–4031, 2018.

[72] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3464–3472, 2014.

[73] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. *arXiv preprint arXiv:1901.04966*, 2019.

[74] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1219–1228, 2018.

[75] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*, 2016.

[76] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning representations (ICLR)*, 2018.

[77] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[78] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 2017.

[79] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171. Springer, 2012.

[80] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *European Conference on Computer Vision (ECCV)*, pages 143–159. Springer, 2016.

[81] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 1857–1865, 2017.

[82] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning representations (ICLR)*, 2014.

[83] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning representations (ICLR)*, 2013.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[85] Ryohei Kuga, Asako Kanezaki, Masaki Samejima, Yusuke Sugano, and Yasuyuki Matsushita. Multi-task learning using multi-modal encoder-decoder networks with shared skip connections. In *The IEEE International Conference on Computer Vision Workshops*, Oct 2017.

[86] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6647–6655, 2017.

[87] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2011.

[88] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

[89] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 36(3):453–465, 2014.

[90] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[91] Yann Lecun. A theoretical framework for back-propagation. 2001.

[92] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690, 2017.

[93] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision (ECCV)*, 2018.

[94] Vladimir Lekic and Zdenka Babic. Automotive radar and camera fusion using generative adversarial networks. *Computer Vision and Image Understanding*, 04 2019.

[95] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015.

[96] Jerry Li. Twin-gan–unpaired cross-domain image translation with weight-sharing gans. *arXiv preprint arXiv:1809.00946*, 2018.

[97] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *European Conference on Computer Vision (ECCV)*, pages 453–468, 2018.

[98] Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5524–5532, 2018.

[99] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 38(10):2024–2039, 2016.

[100] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

[101] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[102] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[103] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 41(8):1862–1878, 2019.

[104] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[105] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[106] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[107] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 405–415, 2017.

[108] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5657–5666, 2018.

[109] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016.

[110] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017.

[111] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *International Conference on Learning representations (ICLR)*, 2016.

[112] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[113] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[114] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3697–3707, 2018.

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

[116] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning representations (ICLR)*, 2018.

[117] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *International Conference on Learning representations (ICLR)*, 2018.

[118] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, pages 689–696, 2011.

[119] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4467–4477, 2017.

[120] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3387–3395, 2016.

[121] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[122] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. *arXiv preprint arXiv:1904.01774*, 2019.

[123] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[124] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, pages 2642–2651. JMLR. org, 2017.

[125] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724. IEEE, 2014.

[126] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[127] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVA British Machine Vision Conference (BMVC)*, 2015.

[128] 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 *Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[129] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[130] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NIPS 2016 Workshop on Adversarial Training*, 2016.

[131] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *European Conference on Computer Vision (ECCV)*, pages 818–833, 2018.

[132] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning representations (ICLR)*, 2015.

[133] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–58, 2016.

[134] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning (ICML)*, pages 1060–1069, 2016.

[135] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning (ICML)*, 2014.

[136] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[137] Karl Ricanek and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 341–345. IEEE, 2006.

[138] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning (ICML)*, pages 2922–2930. JMLR. org, 2017.

[139] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1641–1648. IEEE, 2011.

[140] Andrés Romero, Pablo Arbeláez, Luc Van Gool, and Radu Timofte. Smit: Stochastic multi-label image-to-image translation. *arXiv preprint arXiv:1812.03704*, 2018.

[141] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[142] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[143] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5506–5514, 2016.

[144] Adrià Ruiz, Oriol Martinez, Xavier Binefa, and Jakob Verbeek. Learning disentangled representations with reference-based variational autoencoders. *arXiv preprint arXiv:1901.08534*, 2019.

[145] Adria Ruiz, Ognjen Rudovic, Xavier Binefa, and Maja Pantic. Multi-instance dynamic ordinal random fields for weakly supervised facial behavior analysis. *Transactions on Image Processing*, 27(8):3969–3982, 2018.

[146] 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.

[147] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. *International Conference on Machine Learning (ICML)*, 2017.

[148] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[149] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2234–2242, 2016.

[150] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *European Conference on Computer Vision (ECCV)*, pages 213–229, 2018.

[151] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5541–5550, 2017.

[152] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012.

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

[154] Edward Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. *arXiv preprint arXiv:1707.09557*, 2017.

[155] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[156] Xinhang Song, Luis Herranz, and Shuqiang Jiang. Depth cnns for rgb-d scene recognition: learning from scratch better than transferring from rgb-cnns. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[157] Kumar Sricharan, Raja Bala, Matthew Shreve, Hui Ding, Kumar Saketh, and Jin Sun. Semi-supervised conditional gans. *arXiv preprint arXiv:1708.05789*, 2017.

[158] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning representations (ICLR)*, 2017.

[159] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.

[160] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning representations (ICLR)*, 2015.

[161] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528. IEEE, 2011.

[162] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[163] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 5929–5940, 2018.

[164] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4068–4076, 2015.

[165] Abhinav Valada, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *International Symposium on Experimental Robotics*, pages 465–477. Springer, 2016.

[166] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015.

[167] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018.

[168] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[169] Yaxing Wang, Abel Gonzalez-Garcia, Joost van de Weijer, and Luis Herranz. Controlling biases and diversity in diverse image-to-image translation. *arXiv preprint arXiv:1907.09754*, 2019.

[170] Yaxing Wang, Abel Gonzalez-Garcia, Joost van de Weijer, and Luis Herranz. Sdit: Scalable and diverse cross-domain image translation. *ACM Multimedia*, 2019.

[171] Yaxing Wang, Luis Herranz, and Joost van de Weijer. Mix and match networks: multi-domain alignment for unpaired image-to-image translation. *arXiv preprint arXiv:1903.04294*, 2019.

[172] Yaxing Wang, Joost van de Weijer, and Luis Herranz. Mix and match networks: encoder-decoder alignment for zero-pair image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[173] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *European Conference on Computer Vision (ECCV)*, pages 218–234, 2018.

[174] Yaxing Wang, Lichao Zhang, and Joost van de Weijer. Ensembles of generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training*, 2016.

[175] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: learning to generate images from new categories without forgetting. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2018.

[176] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[177] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *European Conference on Computer Vision (ECCV)*, 2018.

[178] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 2018.

[179] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5542–5551, 2018.

[180] Dan Xu, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5363–5371, 2017.

[181] Zili Yi, Hao Zhang, Ping Tan Gong, et al. Dualgan: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[182] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning representations (ICLR)*, 2016.

[183] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[184] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[185] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[186] Lu Yu, Yongmei Cheng, and Joost van de Weijer. Weakly supervised domain-specific color naming based on attention. In *International Conference on Pattern Recognition (ICPR)*, pages 3019–3024. IEEE, 2018.

[187] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[188] Lu Yu, Lichao Zhang, Joost van de Weijer, Fahad Shahbaz Khan, Yongmei Cheng, and C Alejandro Parraga. Beyond eleven color names for image understanding. *Machine Vision and Applications*, 29(2):361–373, 2018.

[189] Xiaoming Yu, Xing Cai, Zhenqiang Ying, Thomas Li, and Ge Li. Singlegan: Image-to-image translation by a single-generator network using multiple generative adversarial learning. In *Asian Conference on Computer Vision (ACCV)*, 2018.

[190] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[191] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xialei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2017.

[192] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 2017.

[193] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[194] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Synthetic data generation for end-to-end thermal infrared tracking. *Transactions on Image Processing*, 28(4):1837–1850, 2019.

[195] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[196] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, pages 649–666. Springer, 2016.

[197] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep networks as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[198] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[199] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[200] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 10815–10824, 2018.

[201] Hao Zheng, Yong Cheng, and Yang Liu. Maximum expected likelihood estimation for zero-resource neural machine translation. In *AAAI International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[202] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *International Conference on Learning representations (ICLR)*, 2014.

[203] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016.

[204] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 487–495, 2014.

[205] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Yong Yu, and Jun Wang. Activation maximization generative adversarial nets. In *International Conference on Learning representations (ICLR)*, 2018.

[206] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[207] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 465–476, 2017.

[208] James Zou and Londa Schiebinger. Ai can be sexist and racist—it's time to make it fair, 2018.

[209] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *European Conference on Computer Vision (ECCV)*, September 2018.