Computer Vision and Image Understanding
journal homepage: www.elsevier.com

Controlling biases and diversity in diverse image-to-image translation

Yaxing Wang[a,**], Abel Gonzalez-Garcia[a], Luis Herranz[a], Joost van de Weijer[a]

[a]*Computer Vision Center, Edifici O, Universidad Autónoma de Barcelona, 08193, Bellaterra, Spain.*

ABSTRACT

The task of unpaired image-to-image translation is highly challenging due to the lack of explicit cross-domain pairs of instances. We consider here diverse image translation (DIT), an even more challenging setting in which an image can have multiple plausible translations. This is normally achieved by explicitly disentangling content and style in the latent representation and sampling different styles codes while maintaining the image content. Despite the success of current DIT models, they are prone to suffer from bias. In this paper, we study the problem of bias in image-to-image translation. Biased datasets may add undesired changes (e.g. change gender or race in face images) to the output translations as a consequence of the particular underlying visual distribution in the target domain. In order to alleviate the effects of this problem we propose the use of semantic constraints that enforce the preservation of desired image properties. Our proposed model is a step towards unbiased diverse image-to-image translation (UDIT), and results in less unwanted changes in the translated images while still performing the wanted transformation. Experiments on several heavily biased datasets show the effectiveness of the proposed techniques in different domains such as faces, objects, and scenes.

© 2024 Elsevier Ltd. All rights reserved.

1. Introduction

Image-to-image translation (simply image translation hereinafter) is a powerful framework to apply complex data-driven transformations to images [16, 22, 26, 27, 48, 46]. The transformation is determined by the data collected from the input and output domains, which can be arranged as explicit input-output instance pairs [22] or just the looser pairing at set level [26, 31, 48, 56], 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[1] in the output [21, 27, 57] by formulating image translation as a mapping from an input image to a (conditional) output distribution (see Fig. 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 [27, 57]. 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 [8]. Similar gender bias is observed in image captioning [18]. Here we focus on the kind of biases that may affect image translation systems. Although bias is inherent to data collection, it is certainly

**Corresponding author: Tel.: +34-64444248;
e-mail: yaxing@cvc.uab.es (Yaxing Wang)

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

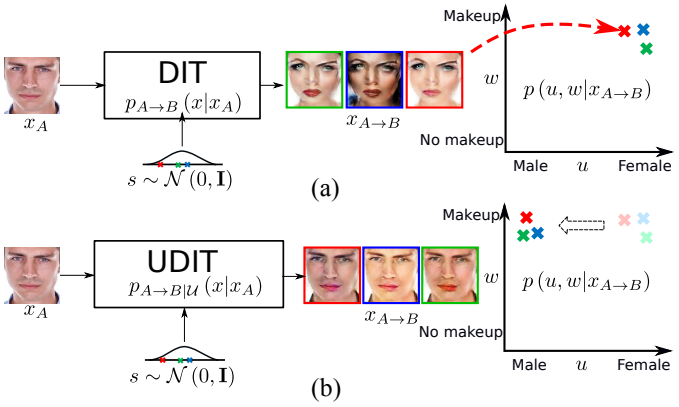

Fig. 1: Diverse image-to-image translation in a very biased setting (domain A: mostly white males without makeup, domain B: white females with makeup): (a) biased translations, (b) with semantic constraint to alleviate bias while keeping relevant diversity.

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 [20, 23, 54, 58].

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[2]. With such biases, the translator learns to generate female faces with makeup even when the input is a male face (see Fig. 1a). While the change in the makeup attribute is desired, the change in identity and gender are not.

In this paper we propose to make the image translator counter undesired biases, by incorporating *semantic constraints* that enforce minimizing the undesired changes (e.g. see Fig. 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.

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

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 [7, 21, 22, 28, 32, 52, 56, 57]. Most image translation approaches employ conditional Generative Adversarial Networks (GANs) [17], 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 Isola et al. [22], pix2pix, 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 [26, 31, 43, 48, 56] 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.

Some papers focus on makeup for the human face. UGAN [55] aims to erase source-specific characteristics and boost the characteristics specific to the target. Zhang et al. [51] focuses on generating diverse makeup faces given single input. In a similar spirit, BeautyGAN [30] proposed both global and local losses to improve the translation between women without makeup and women with makeup, but the loss computaion relies on the guidance of a face mask output by a face parsing model [47]. Finally, Paired-CycleGAN [9] only adds makeup to women at training time.

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 [22] 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 [57] 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 [1, 21, 27] 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 [21], which divides the 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 pre-

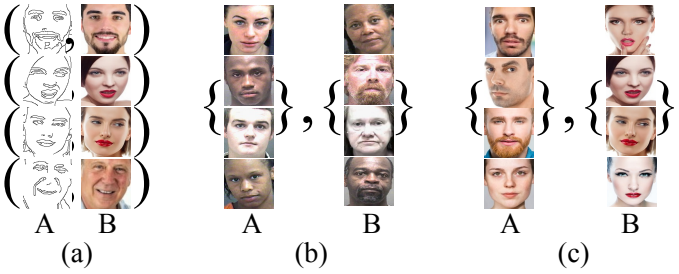

Fig. 2: Examples of training sets for image translation: (a) paired edge-photo, (b) unpaired young-old (well-aligned biases), and (c) unpaired without-with makeup (misaligned in gender).

served 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 [3], which is beneficial for tasks such as cross-domain classification [5, 6, 15, 33] or retrieval [16]. In the context of generative models, Mathieu et al. [34] 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 [11] 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 [37] or by solving analogical relationships through representation arithmetic [38]. The work of [16] 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 [45]. Bias is multifaceted, and datasets can be biased in many ways (e.g. illumination conditions, capture devices, class imbalance, scale [19]). Dataset bias can be addressed and improve cross-dataset generalization [14, 24]. A related problem is domain adaptation [13, 36] 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 [8, 18]. Therefore, tackling bias is an increasingly important topic in machine learning [20, 23, 54, 58]. Here we focus on the specific problem of understanding bias in image translation.

3. Diverse image translation

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 [21, 27] 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. 4a).

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

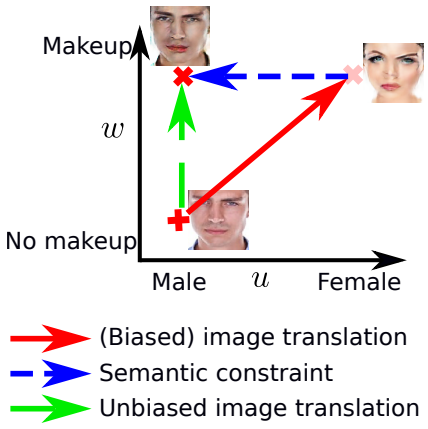

Fig. 3: Geometric interpretation of the semantic constraint unbiasing the translation.

Fig. 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 $p_j(w, u|x_j)$, 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 $u_i = h(x_i)$, and the corresponding translation $x_{i\to j}$ will have $w_{i\to j} = g(x_{i\to j})$ and $u_{i\to j} = h(x_{i\to j})$. The image translation is *successful* if $w_{i\to j} \neq w_i$ 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. 3), which stems from the original bias in the training dataset.

4. Unbiased diverse image translation

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 $(w_j, u_j) = g(x_j)$. 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 have

$$\begin{gathered} u_{i\to j} = h(x_{i\to j}) \\ u_i = h(x_i) \\ u_{i\to j} = u_i \\ \forall x_{i\to j} \sim p_{i\to j}(x|x_i), \forall x_i \sim p_i(x) \end{gathered} \tag{1}$$

where $u_{i\to j} = u_i$ indicates that the unwanted properties remain unchanged throughout the translation. Similarly

$$\begin{gathered} w_{i\to j} = g(x_{i\to j}) \\ w_j = g(x_j) \\ w_{i\to j} = w_j \\ \forall x_{i\to j} \sim p_{i\to j}(x|x_i), \forall x_j \sim p_j(x|x_i), \forall x_i \sim p_i(x) \end{gathered} \tag{2}$$

where $w_{i\to j} = w_j$ indicates 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. 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. 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. 2b). However, Fig. 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 the constraints to preserve of unwanted properties and ensuring the desired change in the wanted properties 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 (4.1) via the addition of a semantic regularization constraint that enforces the preservation of u properties during translation, while constraint (4.1) 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.

4.2. Semantic regularization constraint

Here we propose an Unbiased DIT model (UDIT) that enforces constraint (4.1) 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 (4.1) on the wanted changes is implicitly enforced by the DIT model, including the unpaired setting. Fig. 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*

$$\mathcal{L}_{\mathcal{U}}^{u_i} = \mathbb{E}_{x_i \sim p_i(x)}\left[||u_{i\to j} - u_i||\right], \tag{3}$$

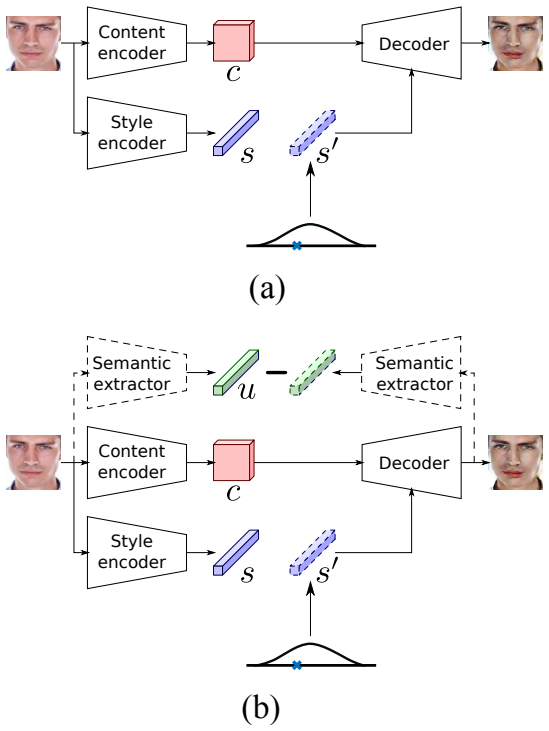

Fig. 4: Diverse image-to-image translation (DIT): (a) biased, (b) unbiased (i.e. UDIT) via a semantic constraint implemented with a semantic extractor.

where $\mathcal{U}$ represents the semantic properties we want to keep unchanged throughout the translation. By including $\mathcal{L}_{\mathcal{U}}^{u_i}$ in our training objective (sec. 5.2), we are effectively conditioning the output conditional distribution to $\mathcal{U}$, i.e. $p_{i\to j|\mathcal{U}}(x|x_i)$, and hence alleviating the unwanted bias in the output samples $x_{i\to j} \sim p_{i\to j|\mathcal{U}}(x|x_i)$, when $\mathcal{U}$ is properly designed. Fig. 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. Implementing UDIT

Our UDIT framework consists of two parts: the semantic extractor and the image-to-image translator.

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), which we train on a suitable external dataset. The CNN may also be initialized with models pretrained in large datasets (e.g. ImageNet [41], DeepFace [44]). 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.

5.2. Image-to-image translator

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 [21] 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 [2].

In the following, we detail the remaining losses and present our full model. The final loss consists of several losses. The *adversarial loss* classifies the real data of target domain from the synthesized data. The *image reconstruction loss* guarantees that the translated image keeps the structure of the input image. Finally, the *latent code reconstruction loss* regularizes the latent code to preserve both content and style information.

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

$$\begin{aligned}\mathcal{L}_{GAN}^{x_j} = \frac{1}{2}\mathbb{E}_{x_i\sim p_i(x),s'\sim\mathcal{N}(0,\mathbf{I})}\left[(D_j(G_j(c_i,s'))^2\right] \\ + \mathbb{E}_{x_j\sim p_j(x)}\left[(D_j(x_j)-1)^2\right].\end{aligned} \tag{4}$$

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

$$\mathcal{L}_{recon}^{x_i} = \mathbb{E}_{x_i \sim p_i(x)} \big[||G_i(c_i, s_i) - x_i||_1 \big]. \quad (5)$$

Latent code reconstruction loss. The translated image is further encoded in both content and style, and the following feature reconstruction losses are applied

$$\mathcal{L}_{recon}^{c_i} = \mathbb{E}_{x_i \sim p_i(c), s' \sim \mathcal{N}(0, \mathbf{I})} \big[||E_j^c(G_j(c_i, s')) - c_i|| \big], \quad (6)$$

$$\mathcal{L}_{recon}^{s_i} = \mathbb{E}_{x_i \sim p_i(c), s' \sim \mathcal{N}(0, \mathbf{I})} \big[||E_j^s(G_j(c_i, s')) - s'|| \big]. \quad (7)$$

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.

Full Objective. The loss used to trained UDIT follows MUNIT's loss combined with the semantic constraint loss (3) as follows

$$\begin{aligned} \mathcal{L} =& \mathcal{L}_{GAN}^{x_A} + \mathcal{L}_{GAN}^{x_B} + \lambda_x (\mathcal{L}_{recon}^{x_A} + \mathcal{L}_{recon}^{x_B}) \\ & \lambda_c (\mathcal{L}_{recon}^{c_A} + \mathcal{L}_{recon}^{c_B}) + \lambda_s (\mathcal{L}_{recon}^{s_A} + \mathcal{L}_{recon}^{s_B}) \\ & \lambda_{\mathcal{U}} (\mathcal{L}_{\mathcal{U}}^{u_A} + \mathcal{L}_{\mathcal{U}}^{u_B}), \end{aligned} \quad (8)$$

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

6. Experimental results

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 [39] 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. 6.5) with the purpose of preserving both gender and ethnicity.

Cityscapes [12]→Synthia [40] contains real and synthesized urban scenes that are biased towards a particular time of the day (day/night). Cityscapes [12] contains real street scenes captured from a moving vehicle during daytime (3000 images). Synthia [40], 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 [56] 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 [22, 21]. 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 1 and 2 specify the exact number of images used in our biased datasets for training and testing, respectively. Table 3 reports the setting to train the metric network. Note for the biased makeup dataset, the used gender classifier is externally trained on Adience dataset [29].

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 1: Details of datasets used for *training* the image translation models. Abbreviations used: 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 2: Details of datasets used for *testing* the image translation models. Abbreviations used: f=female, m=male, y=young, o=old, citys=cityscapes, syn=synthia, txt=textured.

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 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.

6.2. Semantic extractor and Classifier

In this paper, we introduce the semantic extractor and the classifier. The former prevents changing the unwanted properties, such as gender when our task is makeup. The latter is to evaluate the performance of our generated model to keep the unwanted properties. There are two differences between both classifiers. First, the dataset used to train the model is different. Taking the biased makeup as an example: the semantic extractor is trained on VGG-Face [35] (section. 6.4), while the classifier to evaluate our model is trained on the face dataset which contains two classes: (1) male (makeup, no makeup) and (b) female (makeup, no makeup), which guarantees that the trained classifier is robust to makeup. Second, we leverage different architectures. Although both classifiers are based on VGG backbone, the semantic extractor has an additional convolutional layer with the purpose of extracting effectively the semantic information. The classifier to evaluate our model has the same architecture as VGG.

6.3. Baselines and variants

We compare our method with the following approaches:

MUNIT [21] 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. In order to do so, MUNIT introduces domain-specific encoders, generators, and discriminators.The encoders output both the content code and the style code. The content code contains pose information, while the style code aims to represent the stylistic appearance information. The learned content and style are input into the generator to synthesize the output sample. At test time, MUNIT takes as an input the source image and different style codes to achieve diverse outputs.

DRIT [27] 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.

NICE-GAN [10] investigates sharing weights between encoder and discriminator for compactness and training effectiveness. More specifically, their encoder is part of the discriminator and it is trained as such via a decoupling mechanism.

U-GAT-IT [25] combines an attention module with a new learnable normalization function, which enables the handling of geometric changes.

We compare the previous baselines with different configurations of the proposed UNIT approach. In particular we study variants with and without Pooling Index(PI). The code is available [3].

[3] https://github.com/yaxingwang/UDIT

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 4: LPIPS distance on Biased makeup.

6.4. Robustness to specific biases.

Evaluating the generated images is challenge [4], 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. $f(x_{i\to j})$. 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 $\delta(x_i) = f(x_i) - f(x_{i\to j})$. 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 network [35]. We call this measure *ID distance* and only compute it for the face datasets.

Diversity. Several image translation approaches [57, 21, 27] measure the diversity of the outputs by using the perceptual similarity metric LPIPS [53], which is based on differences between deep features We follow the protocol introduced in [57] and average the LPIPS distance between 19 random pairs of outputs for 100 different input images.

6.5. 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 network for face recognition [35] and select its highest level convolutional features as semantic feature. The model has been trained with VGG-Face [35], which contains over 2000 different identities. VGG-Face is based on VGG-Very-Deep-16 architecture [42]. In this paper, we employ the convolutional layers of VGG-Face to perform semantic constraint.

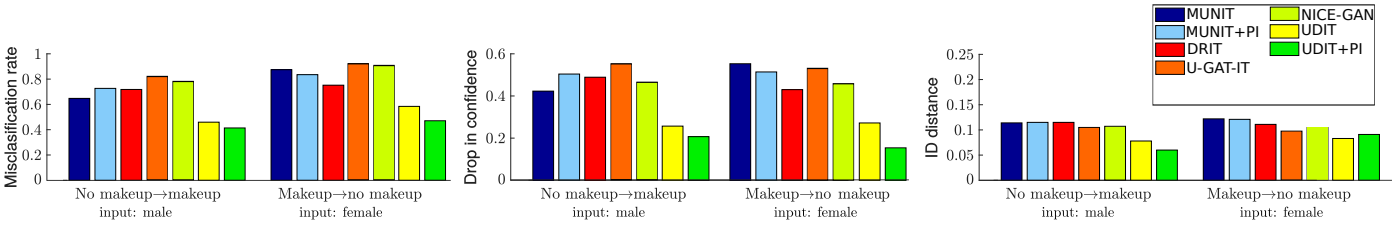

Fig. 5: Robustness to bias on Biased makeup: (left) misclassification rate, (middle) drop in confidence, (right) ID distance.

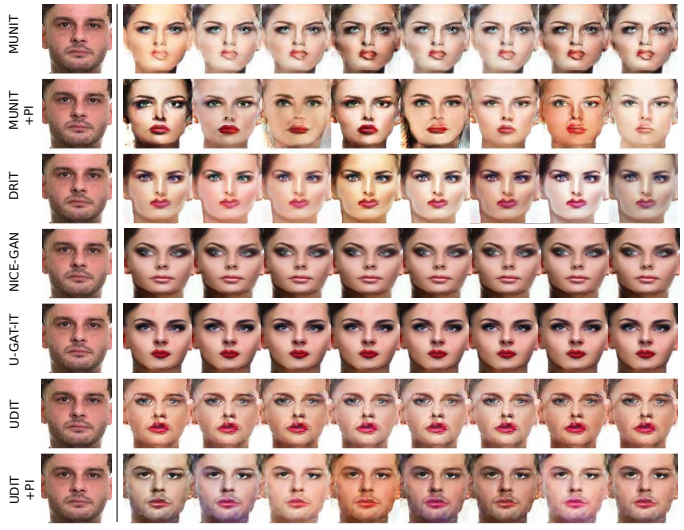

Fig. 6: Example translations for Biased makeup when applying makeup to a male. UDIT uses identity as semantic constraint. The methods NICE-GAN and U-GAT-IT do not have diversity and generate the exact same image several times.

Those convolutional layers contains 5 blocks. The first block contains two convolutional layers. Each of the remained blocks consists of one max pooling and three convolutional layers, except for the second block which employ one max pooling and two convolutional layers.All convolutional layer has same structure which is composed of 3×3 filters with stride 1. Note we use VGG-Face for both *biased makeup* and *MORPH* datasets.

Qualitative evaluation. Fig. 6 compares image translations obtained with MUNIT [21], MUNIT with pooling indices (PI), DRIT [27], 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.

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: Classifier accuracy for different D values. Boldface indicates the selected value for the semantic constraint.

Robustness to unwanted changes. Fig. 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. 6. DIT baselines perform poorly at maintaining gender and identity, including MUNIT with PI. Both NICE-GAN and U-GAT-IT fails to obtain diversity except for generating the target images (e.g., male without makeup). 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 paper we only employ the UDIT+PI variant and refer to it simply as UDIT, unless stated otherwise.

Diversity. Table 4 shows the LPIPS distance of the different evaluated methods. UDIT models seem to be notably decreasing the LPIPS distance in comparison 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. 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).

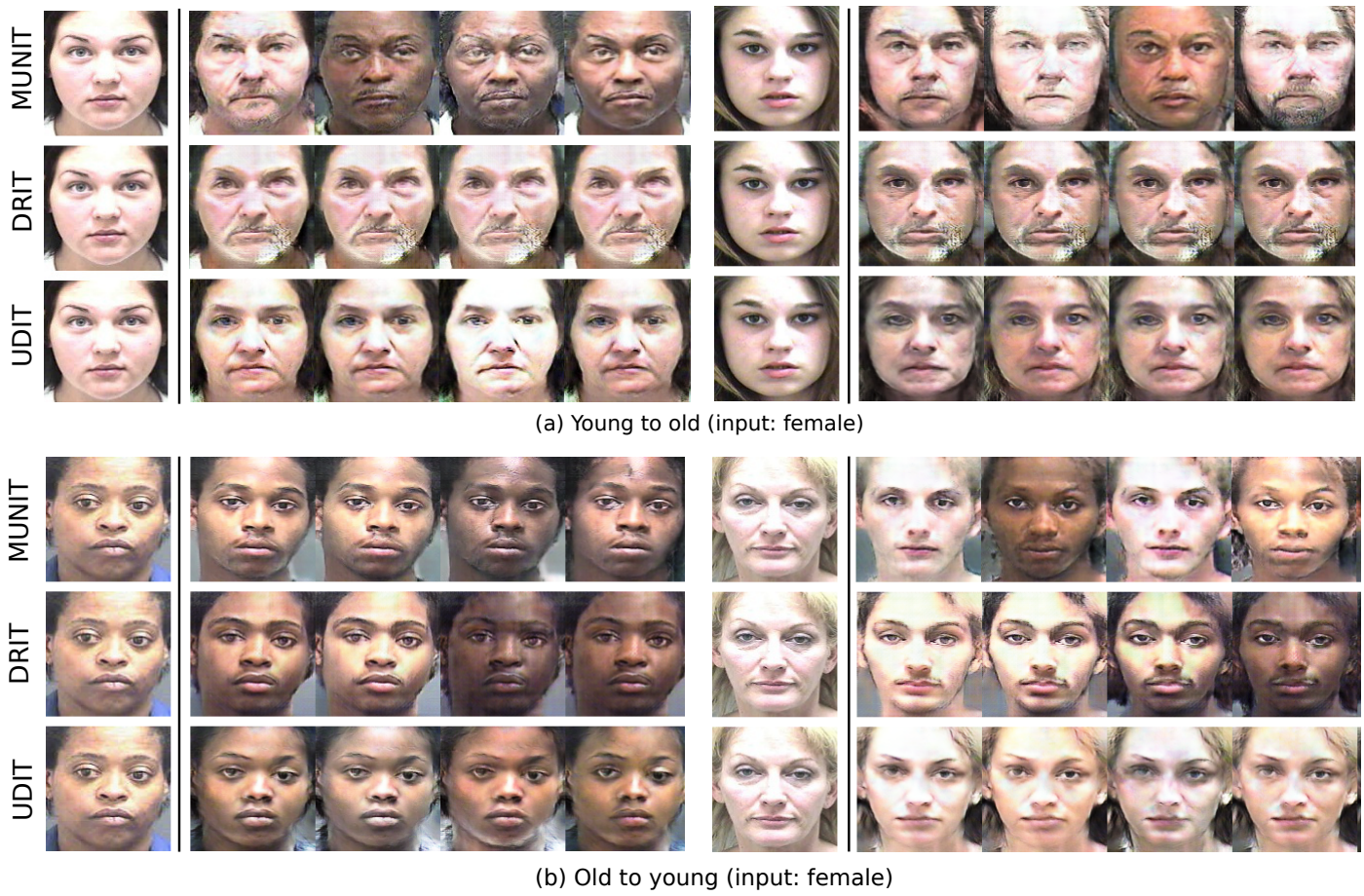

Fig. 7: Example translations on MORPH by biased DIT methods (MUNIT/DRIT) and our UDIT with semantic constraint on identity.

6.6. MORPH

Qualitative evaluation. Fig. 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. 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 better preserved. These results are in line with the observations in Fig. 7.

6.7. Cityscapes → Synthia-night

Semantic constraint. We train a binary classifier for daytime classification based on VGG16 [42] using both real and synthetic images. We use 6000 realistic images from BDD-100K [49] with a 50/50 daytime distribution. As synthetic images we use 6000 images from a disjoint subset of Synthia [40], 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 variant presented in sec. 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). We select $D = 16$ as it offers a good trade-off between small size and accuracy.

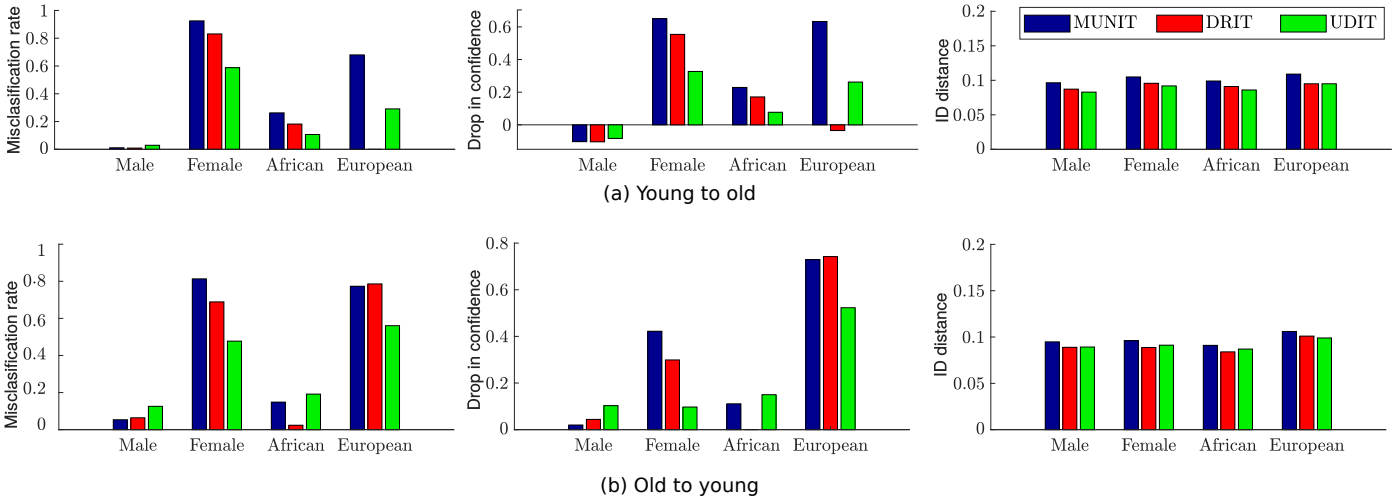

Fig. 8: Robustness to bias on MORPH: (a)*young* to *old* and (b)*old* to *young*: (left) misclassification rate, (middle) drop in confidence, and (right) ID distance.

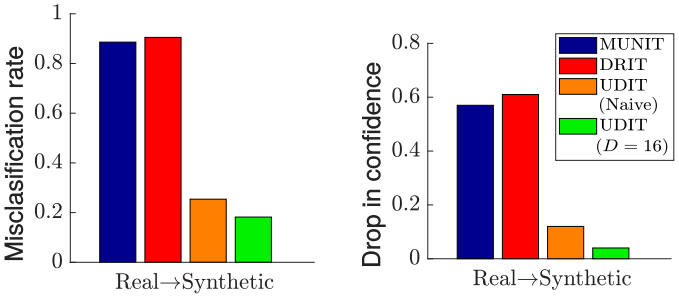

Fig. 9: Robustness to bias in terms of misclassification rate and drop in confidence .

Results. Figs. 9 and 10 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.

6.8. Biased handbags

Semantic constraint. In this section, we still construct the classifier based on VGG16 [42]. We consider two different semantic constraints depending on the experiment. For Handbags-texture we train a color classifier selecting 500 images per color from [50]. 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 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. 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. 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.

7. Conclusion

In this paper 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.

8. Acknowledgements

We acknowledge the support from Huawei Kirin Solution, the Spanish projects TIN2016-79717-R and RTI2018-102285-A-I00, the CERCA Program of the Generalitat de Catalunya, and the EU Marie Sklodowska-Curie grant agreement No.6655919.

References

[1] Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A., 2018. Augmented cyclegan: Learning many-to-

Fig. 10: Results on Cityscapes → Synthia-night. Example translations by MUNIT and UDIT with two variants of the semantic constraint.

many mappings from unpaired data, in: International Conference on Machine Learning.

[2] Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[3] Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[4] Borji, A., 2019. Pros and cons of gan evaluation measures. Computer Vision and Image Understanding 179, 41–65.

[5] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D., 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[6] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D., 2016. Domain separation networks, in: Advances in Neural Information Processing Systems.

[7] Bozorgtabar, B., Rad, M.S., Ekenel, H.K., Thiran, J.P., 2019. Learn to synthesize and synthesize to learn. Computer Vision and Image Understanding .

[8] Buolamwini, J., Gebru, T., 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, pp. 77–91.

[9] Chang, H., Lu, J., Yu, F., Finkelstein, A., 2018. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 40–48.

[10] Chen, R., Huang, W., Huang, B., Sun, F., Fang, B., 2020. Reusing discriminators for encoding towards unsupervised image-to-image translation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[11] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P., 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Information Processing Systems.

[12] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[13] Daumé III, H., 2007. Frustratingly easy domain adaptation. Proceedings of the Annual Meeting of the Association of Computational Linguistics .

[14] Fang, C., Xu, Y., Rockmore, D.N., 2013. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias, in: Proceedings of the International Conference on Computer Vision, pp. 1657–1664.

[15] Ganin, Y., Lempitsky, V., 2015. Unsupervised domain adapta-

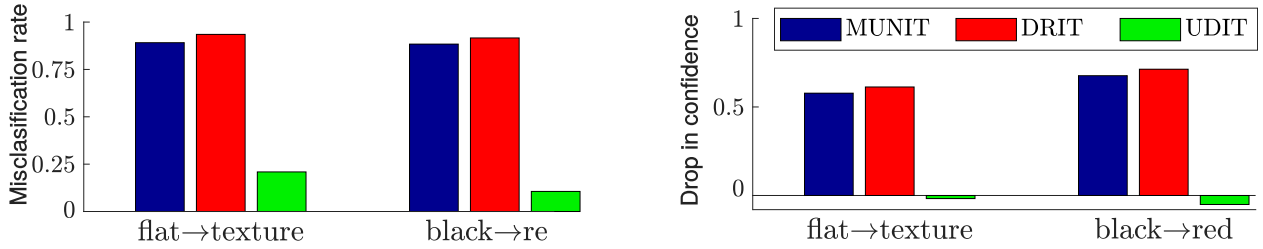

Fig. 11: Robustness to bias on Biased handbags.

Fig. 12: Example translations for Handbags-texture (left) and Handbags-color (right). Better viewed electronically, zoom might be necessary to appreciate the changes in texture.

tion by backpropagation, in: International Conference on Machine Learning.

[16] Gonzalez-Garcia, A., van de Weijer, J., Bengio, Y., 2018. Image-to-image translation for cross-domain disentanglement, in: Advances in Neural Information Processing Systems.

[17] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in Neural Information Processing Systems.

[18] Hendricks, L.A., Burns, K., Saenko, K., Darrell, T., Rohrbach, A., 2018. Women also snowboard: Overcoming bias in captioning models, in: Proceedings of the European Conference on Computer Vision, Springer. pp. 793–811.

[19] Herranz, L., Jiang, S., Li, X., 2016. Scene recognition with cnns: objects, scales and dataset bias, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 571–579.

[20] Howard, A., Zhang, C., Horvitz, E., 2017. 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 (ARSO), IEEE. pp. 1–7.

[21] Huang, X., Liu, M.Y., Belongie, S., Kautz, J., 2018. Multimodal unsupervised image-to-image translation. Proceedings of the European Conference on Computer Vision .

[22] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[23] Jiang, H., Nachum, O., 2019. Identifying and correcting label bias in machine learning. arXiv preprint arXiv:1901.04966 .

[24] Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A., 2012. Undoing the damage of dataset bias, in: Proceedings of the European Conference on Computer Vision, Springer. pp. 158–171.

[25] Kim, J., Kim, M., Kang, H., Lee, K., 2019. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. arXiv preprint arXiv:1907.10830 .

[26] Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J., 2017. Learning to discover cross-domain relations with generative adversarial networks. International Conference on Machine Learning .

[27] Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H., 2018. Diverse image-to-image translation via disentangled representations. Proceedings of the European Conference on Computer Vision .

[28] Lekic, V., Babic, Z., 2019. Automotive radar and camera fusion

using generative adversarial networks. Computer Vision and Image Understanding doi:10.1016/j.cviu.2019.04.002.

[29] Levi, G., Hassner, T., 2015. Age and gender classification using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 34–42.

[30] Li, T., Qian, R., Dong, C., Liu, S., Yan, Q., Zhu, W., Lin, L., 2018. Beautygan: Instance-level facial makeup transfer with deep generative adversarial network, in: Proceedings of the 26th ACM international conference on Multimedia, pp. 645–653.

[31] Liu, M.Y., Breuel, T., Kautz, J., 2017. Unsupervised image-to-image translation networks, in: Advances in Neural Information Processing Systems, pp. 700–708.

[32] Liu, X., Van De Weijer, J., Bagdanov, A.D., 2019. Exploiting unlabeled data in cnns by self-supervised learning to rank. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 1862–1878.

[33] Liu, Y.C., Yeh, Y.Y., Fu, T.C., Wang, S.D., Chiu, W.C., Wang, Y.C.F., 2018. Detach and adapt: Learning cross-domain disentangled deep representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[34] Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y., 2016. Disentangling factors of variation in deep representation using adversarial training, in: Advances in Neural Information Processing Systems.

[35] Parkhi, O.M., Vedaldi, A., Zisserman, A., 2015. Deep face recognition, in: Proceedings of the British Machine Vision Conference.

[36] Patel, V.M., Gopalan, R., Li, R., Chellappa, R., 2015. Visual domain adaptation: A survey of recent advances. IEEE signal processing magazine 32, 53–69.

[37] Reed, S., Sohn, K., Zhang, Y., Lee, H., 2014. Learning to disentangle factors of variation with manifold interaction, in: International Conference on Machine Learning.

[38] Reed, S.E., Zhang, Y., Zhang, Y., Lee, H., 2015. Deep visual analogy-making, in: Advances in Neural Information Processing Systems.

[39] Ricanek, K., Tesafaye, T., 2006. Morph: A longitudinal image database of normal adult age-progression, in: Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, IEEE. pp. 341–345.

[40] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M., 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243.

[41] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, 211–252.

[42] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

[43] Taigman, Y., Polyak, A., Wolf, L., 2017. Unsupervised cross-domain image generation, in: International Conference on Learning Representations.

[44] Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708.

[45] Torralba, A., Efros, A.A., 2011. Unbiased look at dataset bias, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1521–1528.

[46] Wang, Y., van de Weijer, J., Herranz, L., 2018. Mix and match networks: encoder-decoder alignment for zero-pair image translation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[47] Wei, Z., Sun, Y., Wang, J., Lai, H., Liu, S., 2017. Learning adaptive receptive fields for deep image parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2434–2442.

[48] Yi, Z., Zhang, H.R., Tan, P., Gong, M., 2017. Dualgan: Unsupervised dual learning for image-to-image translation., in: Proceedings of the International Conference on Computer Vision, pp. 2868–2876.

[49] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T., 2018a. Bdd100k: A diverse driving video database with scalable annotation tooling. Proceedings of the European Conference on Computer Vision .

[50] Yu, L., Cheng, Y., van de Weijer, J., 2018b. Weakly supervised domain-specific color naming based on attention, in: Proceedings of the International Conference on Pattern Recognition, IEEE. pp. 3019–3024.

[51] Zhang, H., Chen, W., He, H., Jin, Y., 2019. Disentangled makeup transfer with generative adversarial network. arXiv preprint arXiv:1907.01144 .

[52] Zhang, L., Gonzalez-Garcia, A., van de Weijer, J., Danelljan, M., Khan, F.S., 2018a. Synthetic data generation for end-to-end thermal infrared tracking. IEEE Transactions on Image Processing 28, 1837–1850.

[53] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018b. The unreasonable effectiveness of deep networks as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[54] Zhao, S., Ren, H., Yuan, A., Song, J., Goodman, N., Ermon, S., 2018. Bias and generalization in deep generative models: An empirical study, in: Advances in Neural Information Processing Systems, pp. 10815–10824.

[55] Zhu, D., Liu, S., Jiang, W., Gao, C., Wu, T., Guo, G., 2019. Ugan: Untraceable gan for multi-domain face translation. arXiv preprint arXiv:1907.11418 .

[56] Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[57] Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E., 2017b. Toward multimodal image-to-image translation, in: Advances in Neural Information Processing Systems, pp. 465–476.

[58] Zou, J., Schiebinger, L., 2018. Ai can be sexist and racistits time to make it fair.

Appendix

Tables 6-10 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 11.

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 6: 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 7: 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]

(a) affine parameter μ

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]

(b) affine parameter σ

Table 8: Networks for the estimation of the affine parameters that are used in the AdaIN layer. The parameters (a) μ and (b) σ scale and shift the normalized content, respectively. Note that (a) and (b) 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 9: 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 10: 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 11: Abbreviations used in other tables.