Unsupervised Discovery and Manipulation of Continuous Disentangled Factors of Variation

TOMASO FONTANINI, LUCA DONATI, MASSIMO BERTOZZI, and ANDREA PRATI,
Department of Engineering and Architecture, University of Parma, Italy

Learning a disentangled representation of a distribution in a completely unsupervised way is a challenging task that has drawn attention recently. In particular, much focus has been put in separating factors of variation (i.e., attributes) within the latent code of a Generative Adversarial Network (GAN). Achieving that permits control of the presence or absence of those factors in the generated samples by simply editing a small portion of the latent code. Nevertheless, existing methods that perform very well in a noise-to-image setting often fail when dealing with a real data distribution, i.e., when the discovered attributes need to be applied to real images. However, some methods are able to extract and apply a style to a sample but struggle to maintain its content and identity, while others are not able to locally apply attributes and end up achieving only a global manipulation of the original image.

In this article, we propose a completely (i.e., *truly*) unsupervised method that is able to extract a disentangled set of attributes from a data distribution and apply them to new samples from the same distribution by preserving their content. This is achieved by using an image-to-image GAN that maps an image and a random set of continuous attributes to a new image that includes those attributes. Indeed, these attributes are initially unknown and they are discovered during training by maximizing the mutual information between the generated samples and the attributes' vector. Finally, the obtained disentangled set of continuous attributes can be used to freely manipulate the input samples. We prove the effectiveness of our method over a series of datasets and show its application on various tasks, such as attribute editing, data augmentation, and style transfer.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**; **Computer vision problems;**

Additional Key Words and Phrases: Deep learning, unsupervised learning, mutual information maximization, image manipulation

ACM Reference format:
Tomaso Fontanini, Luca Donati, Massimo Bertozzi, and Andrea Prati. 2023. Unsupervised Discovery and Manipulation of Continuous Disentangled Factors of Variation. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 6, Article 189 (May 2023), 25 pages.
https://doi.org/10.1145/3591358

Tomaso Fontanini and Luca Donati contributed equally to this research.

This research has been financially supported by the Programme "FIL-Quota Incentivante" of University of Parma and co-sponsored by Fondazione Cariparma. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro RTX 6000 GPU used for this research.

Authors' address: T. Fontanini, L. Donati, M. Bertozzi, and A. Prati, Department of Engineering and Architecture, University of Parma, Via delle Scienze, 181/a, Parma, Italy, 43124; emails: {tomaso.fontanini, luca.donati, massimo.bertozzi, andrea.prati}@unipr.it.

1551-6857/2023/05-ART189 $15.00
https://doi.org/10.1145/3591358

1 INTRODUCTION

Generative Adversarial Networks (GANs) are capable of producing fake samples that are almost indistinguishable from real images. In the standard noise-to-image setting, a GAN is required to produce samples starting from a latent vector/code z that is fed into the generator, whose outputs are then validated by the discriminator. Therefore, given the deterministic nature of a neural network, all the information required to generate a specific feature in the output sample is contained in the latent vector. For this reason, ideally, it should be possible to change a small portion of z to influence the generation of a very specific part of the output sample. Anyway, being able to do that requires the capability of the network to identify a set of well-separated *factors of variation*, i.e., attributes. More in detail, a factor of variation corresponds to an image attribute that can be consistently discerned across a set of images [20]. Nevertheless, this task is jeopardized by the entanglement of the latent code, meaning that each of its elements has effect on multiple attributes in the generated image.

Recently, it was proved that well-trained GANs like StyleGAN [26] learn a disentangled latent space that can be interpreted and used to manipulate the outputs of the network (e.g., InterFaceGAN [43]). Previously, InfoGAN [8] proposed to maximize the mutual information between a small subset of the latent variables and the observation to achieve disentangled representations. These methods have been thoroughly explored for the noise-to-image GAN setting, but have not been extended for the discovery of factors of variations in the more challenging image-to-image scenario, where learned attributes need to be applied to real images.

There are several reasons for the increased difficulty when a disentangled representation is used to manipulate real samples instead of generated ones. First, GANs, even if very powerful, are still not able to fully match the complexity of the real distribution and, therefore, their results are merely a subset of all the possible variations found in real images. Second, when manipulating attributes of an image, its overall content needs to be maintained. This can be achieved when dealing with fake samples, since the exact latent code responsible of producing a certain image is known. Therefore, in the case of a disentangled representation, a subset of the latent space can be reasonably changed without jeopardizing the overall content of the output image. However, in the case of real samples, there is no known latent space and, therefore, a direct manipulation is not possible. To mitigate this issue, some existing methods like References [20, 53] rely on mapping (with an encoder) a real image to a compressed latent space, which is then fed into a decoder to produce a reconstructed sample. The latent space can then be edited to change the desired characteristic in the output image. However, in this case, the reconstruction is often poor and only a limited editing is feasible, i.e., only few distinguishable attributes are identified within the latent space. Finally, since the image-to-image setting is more difficult, the network could be prone to take shortcuts [45] and not learn a disentangled representation (by just ignoring some parts of the latent code or learning a set of meaningless factors of variation).

Summarizing, a good system for the identification and manipulation of factors of variation should: (i) find attributes that are discriminative and representative of the data distribution (i.e., the most common attributes among the samples, without any assumption on attribute size or intensity, unlike previous works such as References [3, 44] did); (ii) avoid entanglement between different attributes (i.e., it should be possible to change each attribute independently and separately); and (iii) have the capability to edit an arbitrary number of attributes continuously and locally, preserving the identity and/or content of the input image. An additional key feature is to be able to learn the attributes without any kind of supervision, to be also applicable to unlabelled or poorly labeled datasets.

It is also worth to highlight the difference between multi-domain image-to-image translation methods and the discovery of factors of variation in image manipulation: while in the former a set of predetermined attributes or styles needs to be applied on a source image [11, 31], in the latter (focus of this article) the goal is to find a set of automatically discovered and disentangled attributes.

In this article, an image-to-image GAN is used to achieve fully unsupervised discovery and manipulation of factors of variation. A noise vector is fed into the generator and each of its elements is used to continuously control a different factor. This noise vector is then reconstructed from the output image using a set of encoders that push the generator to learn attributes at different scales and help avoid shortcutting. Indeed, this is the basic principle of *Mutual Information Maximization*. In addition to that, we design the noise vector so that each of its elements is drawn from a continuous distribution, allowing easy control over the intensity of the attributes application.

Summarizing, the main contributions of the article are the following:

- a truly unsupervised system that can discover factors of variation (i.e., attributes) in any given dataset by maximizing the mutual information between an input vector of continuous attributes z_{attr} and the generated sample. This process also allows our model to apply the discovered attributes over the input images in a continuous way. Different from current image manipulation methods, the proposed architecture does not require any supervision both in terms of label supervision or set-level supervision;
- a way to produce meaningful changes in the generated samples and generate attributes of different sizes, starting from the values contained in the attribute vector, thanks to the introduction of a set of anti-shortcut encoders with a pyramidal pooling.

2 RELATED WORKS

Generative Adversarial Networks (GANs). Recently, GANs, first introduced in Reference [14], are dominating the field of image generation. This is due especially to their impressive syntheses capability reached after years of continuous improvement. In particular, References [2, 15] focused on training stability by proposing an alternative loss function for training GAN, called Wasserstein Loss, and introducing gradient penalty that revises the norm of gradients to respect a Lipschitz constraint over the discriminator (called "critic" in this configuration). Moreover, **conditional GANs (cGANs)** [36] were introduced to enforce a better control over the output of the generator using labels, text or images to guide the synthesis. Using class-conditioned GANs, Brock et al. [5] greatly improved the quality of the generated images. They also employed a "truncation trick" over the latent vector to achieve a good trade-off between sample fidelity and variety. Another solution to achieve high quality samples was proposed in Reference [24], where the main idea is to progressively train both the generator and the discriminator for better speed and stability. Furthermore, the concept of self-attention, first proposed in References [9, 40], was employed in GANs like Reference [51] to generate image details using clues from all feature locations and not only spatially local points.

Since its introduction, the state-of-the-art unsupervised approach in noise-to-image generation is represented by StyleGAN [26]. StyleGAN works by mapping the latent code z into a new latent space $\mathcal{W}$ using a non-linear mapping network and then using the new code w to control a series of AdaIN layers [21] in the generator. After the success of StyleGAN, an improved version was proposed [27]. The new version's main contribution is the redesign of the generator normalization, by removing the artifacts introduced by AdaIN layers. Finally, Karras et al. [25] further improved the quality of generation by assuring that unwanted information cannot leak into the hierarchical synthesis process.

Disentanglement. Using GANs to learn a disentangled representation of image attributes was deeply explored in recent years. Mathieu et al. [35] employed a class-conditioned GAN to separate the specified factors of variation linked to the labels from the unspecified one. Later, Tran et al. [46] used an encoder-decoder generator to learn a disentangled representation for pose-invariant face recognition. In addition, Reference [39] achieved face identity disentanglement with minimal supervision. Nevertheless, all these methods require supervision, therefore Hu et al. [20] proposed a completely unsupervised approach that forces the network to encode each attribute into a feature chunk through the mixing of chunks produced by two different images. Then, a decoded image with this mixed representation is generated and, to force the disentanglement, a classifier is trained to distinguish if the chunks used for this generation were extracted from the first or the second input image.

A very interesting approach for disentanglement was proposed by Chen et al. in InfoGAN [8]: their method consists in maximizing the mutual information between a small latent vector and the observation, i.e., the generated image. InfoGAN is able to learn a disentangled representation and, therefore, allows to achieve control over the generated output. For example, by changing the value of one element of the latent vector, it is possible to selectively change rotation or width of the generated **Modified National Institute of Standards and Technology (MNIST)** images. Unfortunately, InfoGAN works in the classic noise-to-image setting of GANs and, for this reason, can not be used to manipulate real images. Recently, InfoGAN-based methods like ClusterGAN [37] were able to achieve clustering in the latent space of GANs by using a mixture of one-hot and continuous variables to compose the noise used as input to the generator and, then, to employ an encoder to reconstruct this mixed noise from the generated images. Finally, recently it was also proved that the latent space of GAN architectures like StyleGAN achieves some form of disentanglement thanks to the non-linear mapping of the random noise z to the new disentangled code w. Exploiting this property, a few latent space manipulation methods like References [44, 49, 53] were introduced, achieving limited fake and real image editing by varying this latent space. In addition, Härkönen et al. [16] used Principal Components Analysis to find important latent directions and created interpretable controls for image synthesis. Finally, Alharbi and Wonka [1] achieved disentanglement through feeding multiple noise codes in separate fully connected layers.

Image translation and manipulation. Being able to apply a certain style or attribute to an image is a core task for GANs. For this purpose, image-to-image models are usually employed, conditioned with an input image (from a domain) and targeting another output domain. There exist many different variants of these models; in earliest works, a paired dataset was required to force the mapping between source and target domains [22, 47]. Being of limited applicability, new approaches [28, 50, 54] that could employ unpaired datasets were rapidly introduced. In these models, no ground truth could be used to validate the translated image, therefore techniques imposing a cycle consistency loss over the model were proposed to maintain the original content of the input images during the translation. Usually, all these models are able to translate only one or two domains at a time and require to train multiple models to achieve the translation to multiple domains. To overcome this limitation, StarGAN [10] proposed an architecture capable to achieve multi-domain image-to-image translation using a single model. This architecture uses a one-hot label as target domain and its discriminator is trained both to distinguish between real and fake samples and to classify domains. Notably, only discrete, binary attributes are allowed (0 or 1). Next, AttGAN [18] improved the StarGAN results by modelling the relation between the latent representation and the attributes. Other approaches to multi-domain image-to-image translation include the use of relative attributes [48] or meta-learning [12]. Recently, StarGANv2 [11] proposed to substitute domain labels with style codes, producing different images across multiple domains.

An alternative approach to translate a target style to an input image is proposed in FUNIT [31] that, along with its successor COCO-FUNIT [42], employs a generator network composed by a content encoder, a class encoder and a decoder with AdaIN layers. The class encoder produces a class code used to guide the AdaIN layers in the decoder and to apply the class style over the content.

Differences with literature. The method proposed here borrows some ideas from the previous models, but differs from them in several aspects. In particular, regarding the InfoGAN-based methods, our system does not fall in the noise-to-image category, but it is able to learn a disentangled attribute vector starting from real images. This increases the complexity of the learning problem by several orders of magnitude, since it implies dealing with a much more complex data distribution.

In addition to that, compared to more traditional image-to-image frameworks like StarGAN, our algorithm is completely unsupervised and is able to learn and apply a much higher number of attributes at the same time. Also, the attributes that are applied are continuous in a range of values, allowing a control over the intensity of their application. However, StarGAN v2 is able to produce multi-modal results, meaning that, given a target domain, it is possible to obtain multiple different results for a single input image. Nevertheless, this multi-modality is indeed random and not controllable and is produced by mixing a latent code z and a domain $\hat{x}$, while in our case we achieved a *continuous multi-modality*, since we can change the intensity of each of the learned attribute (independently) to produce different and various results.

Moreover, methods like FUNIT, StarGANv2 and, in general, AdaIN-based methods, have the problem of struggling to apply local changes to the images and tend to perform more a generic whole-image texture transfer (e.g., using a cat as content and a tiger as style will result in a tiger with the exact same pose of the cat) or an artistic-transfer (e.g., the Tour Eiffel painted in van Gogh's style). This "prior" is caused by the fact that AdaIN is a global image normalization technique (naturally attuned to big attributes), in which we can at best influence only in which layers of the network the style will be injected. The proposed system, instead, is tailored to freely learn significant attributes (therefore also small, local attributes) and to independently apply them, meaning that our method is able to change, e.g., a subject's lips without touching its eyebrows or the beard.

Finally, latent space manipulation methods work very well when editing generated images, but have difficulties when the input is a real image drawn from a dataset or a never-seen-before image. To address this problem, GAN inversion [53] was employed by adding an encoder to extract the latent space from an input image and using it to reconstruct the same image (the so-called inversion). Nevertheless, often this reconstruction is poor even with the most recent methods. Indeed, GAN inversion methods require to use a pre-trained model, therefore limiting their application to the domains where that model performs very well, while the proposed system is conversely trained end-to-end and can work with a large variety of different datasets.

In addition to that, manipulating the latent space usually requires supervision to find the *directions* corresponding to different domains, while the proposed system is fully unsupervised. A strong limitation of unsupervised latent manipulation methods (such as Reference [44]) is that they perform unsupervised attributes discovery from the pre-trained latent space by clustering it or discovering attributes directions in that space. Such approaches strongly depend on the specific discovering strategy. In the case of Reference [44], the "best" attribute directions are selected by employing a factorization algorithm for latent semantic discovery that aims at finding the directions that can cause large variations. While the qualitative results of such a method are good, we argue that this optimization strategy misses small, valuable attributes (such as eyebrows position, glasses, etc.) and is also dataset-specific. Moreover, the approach in Reference [44] discovers attribute directions, but knows nothing of the attributes magnitudes, while our method maps each attribute in a range of values.

2.1 Lack of Supervision in the Proposed Method

A matter of discussion in the field of image-to-image translation is the amount of supervision required for a system to produce good results and also when a certain method can be defined as "unsupervised." More in detail, commonly, unsupervised learning should imply the lack of prior knowledge about the true labels of the classes (or attributes in our case) that the system aims at identifying or generating. Nevertheless, this is not always the case in the image-to-image translation research area.

As an example, to better frame this problem, FUNIT [31] claims to be unsupervised, but it requires having set-level knowledge about the different styles during training, i.e., it selects images belonging to a certain style at each step and organize them in different sets. Indeed, during inference it is able to produce style that has never been seen before, but it is clear that a certain amount of supervision is needed to reach the final result.

Recently, TUNIT [4] was the first proposing a "truly unsupervised" method to solve the need of set-level annotations that FUNIT had. Indeed, in this case the domains were automatically separated using an unsupervised differentiable clustering algorithm [23]. Nevertheless, different than our method, TUNIT makes use of AdaIN layers and for this reason performs more a style/texture transfer than an attribute manipulation. Also, the clustering method is able to find domain-level separation in the dataset (e.g., animal species) and struggle when searching for fine-grained attributes.

We argue that our method is "truly unsupervised," and in addition to that, it is trained end-to-end, is able to transfer fine-grained attributes, and does not rely on other external methods to separate the different attributes.

3 SELF ATTRIBUTE-MAPPING GAN

The goal of our proposal is to design a GAN able to map an image x and a set of attributes z_{attr} into another valid image $\hat{x}$, preserving the content of image x, while only altering those specific attributes. We also aim to train such a GAN in a truly unsupervised fashion.

These attributes are a set of common and distinguishable image features, which usually are dataset-specific. For a dataset of handwritten digits (like MNIST), attributes could be digit width, slant and thickness; for a dataset of people faces (like CelebA), these could be hair color, eyebrows thickness and facial expression, just to name a few. More formally, these attributes are a set of factors of variation shared among several images (so they are not instance-specific, such as people identities for CelebA). In this work, we will focus on continuous attributes, but the approach can be applied also to discrete or categorical attributes (as proposed in Reference [8]). Therefore, in our work, z_{attr} is a vector of c elements $\{z^0_{attr}, \dots, z^{c-1}_{attr}\}$, that are drawn from a continuous distribution and are each defined in the range $[-1,1]$ (but, as it will be clearer later in the article, this range can be expanded at will).

Also, the GAN generator will be an image-to-image generator, instead of a vanilla GAN noise-to-image generator:

$$\hat{x} = G(x, z_{attr}). \tag{1}$$

Another key feature we aim to achieve, is to have disentangled attributes, so that z^0_{attr} should be a specific attribute (e.g., hair color) and z^1_{attr} must be an unrelated attribute (e.g., smile opening), as shown in Figure 1.

3.1 Training and Losses

To train such a GAN in a truly unsupervised fashion, we developed a custom technique: x is drawn from the distribution $\mathcal{X}$ and each element of z_{attr} is randomly picked. The mapping will produce

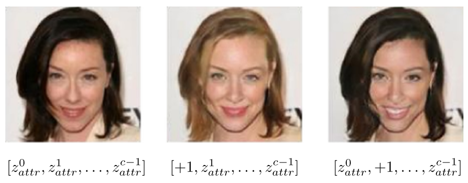
$[z^0_{attr}, z^1_{attr}, \dots, z^{c-1}_{attr}]$
$[+1, z^1_{attr}, \dots, z^{c-1}_{attr}]$
$[z^0_{attr}, +1, \dots, z^{c-1}_{attr}]$

Fig. 1. Examples of disentangled and continuously controllable attributes. Left picture is input image, z^0_{attr} changes hair color, while z^1_{attr} controls smile intensity. The remaining attributes are unchanged.

an image $\hat{x}$ that must belong to the same distribution $\mathcal{X}$. Indeed, the underlying idea of using the same distribution for both the input and the target is one important piece in avoiding the need of any supervision. Finally, $\hat{x}$ must contain the desired set of attributes z_{attr}. Other regularization constraints will be described in Section 3.4.

The first constraint can be easily imposed by feeding $\hat{x}$ into a discriminator. In this way, the network will learn to generate realistic images that belong to the distribution $\mathcal{X}$. Indeed, this is the standard GAN *adversarial loss* $\mathcal{L}_{adv}$, which is used to obtain sharp results from the generator.

Imposing the second constraint is more complex. Under a supervised setting, labels or ground truths can be used to force the application of one or more attributes. Nevertheless, aiming at a truly unsupervised learning, these cannot be exploited. Therefore, at every training step, we pick a set of random attributes z_{attr} and an input image x, and let the network apply freely these random attributes into the input image x obtaining the output $\hat{x}$. The network will be free to choose its own representation of z_{attr}: in particular, which attributes to encode, which ones to ignore, and their order.

Unfortunately, the network could decide to ignore the input attributes, instead of encoding them in the output image. To prevent this, a **Mutual Information Maximization (MIM)** technique [8, 37] was introduced. In practice, an external network called MIM Encoder is added to the architecture. This encoder will be tasked to reconstruct z_{attr} using only the generated image $\hat{x}$. This forces both the generator to properly apply the attributes on the generated image, and the encoder to be able to reconstruct them only from that image (i.e., the attributes must be well recognizable). The network will learn to properly code these factors of variation in the generated image and will also do that in a disentangled way, as also demonstrated in Reference [8] in a much simpler setting. We call this *attribute reconstruction loss* $\mathcal{L}_{attr}$:

$$\mathcal{L}_{attr} = \|\hat{z}_{attr} - z_{attr}\|_2, \tag{2}$$

where $\hat{z}_{attr} = E(\hat{x})$ and E is the MIM Encoder network.

Figure 2 shows the overall topology of the network, including the generator, the MIM encoder, and the discriminator. As the figure suggests, the final version of the architecture contains several MIM encoders described in Section 3.3.

3.2 Shortcuts

While, in theory, imposing the above constraints should be sufficient for achieving attribute changes, in practice this is not the case. While training an encoder E to reconstruct z_{attr} works well for a generic noise-to-image GAN (like References [8, 37]), the same does not apply well in the image-to-image GAN context. As shown in Figure 3(d), the encoder is indeed able to reconstruct the attributes z_{attr}, but no changes are visible in the generated image $\hat{x}$. This is due to the fact that both the generator and the encoder are taking shortcuts for adding the attributes to the images, which is a known problem in image-to-image translation [45]. In other words, the network learns the simplest possible transformation that can minimize the given loss, which can consist in

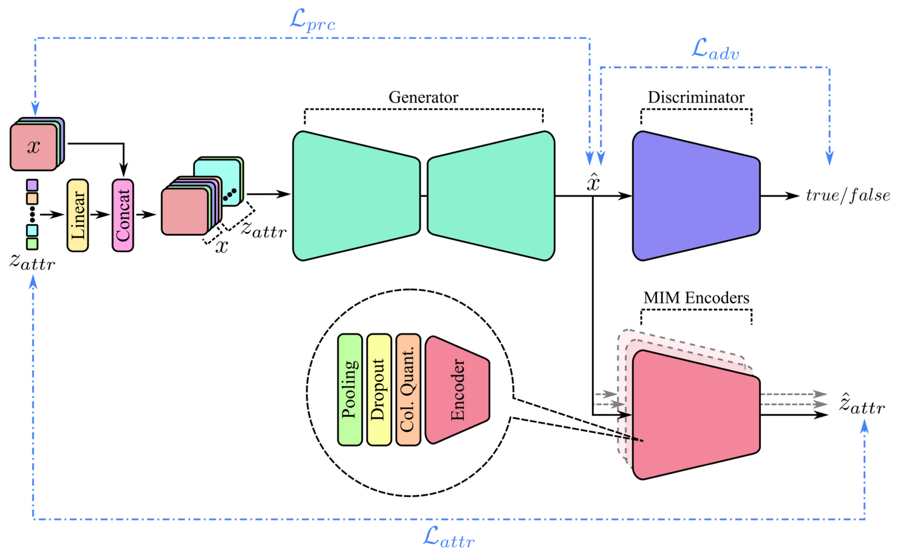

Fig. 2. Topology of the network. First, the attributes vector z_{attr} with c elements is reshaped using a linear layer into an image with the same height and width of the input image x and c channels. Then, the reshaped vector is concatenated with x and used as the input for the generator. Subsequently, the generator outputs the translated image $\hat{x}$ and the perceptual loss $\mathcal{L}_{prc}$ is computed between x and $\hat{x}$. $\hat{x}$ is then fed into the discriminator trained with the adversarial loss $\mathcal{L}_{adv}$. Finally, the set of MIM Encoders $\mathbf{E}$ (each preceded by a different-sized pooling followed by an anti-shortcut module composed by dropout and color quantization) is tasked to reconstruct z_{attr} from $\hat{x}$ using the attribute reconstruction loss $\mathcal{L}_{attr}$.

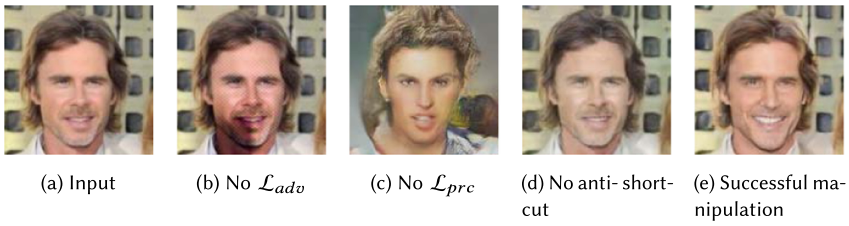
(a) Input
(b) No $\mathcal{L}_{adv}$
(c) No $\mathcal{L}_{prc}$
(d) No anti- shortcut
(e) Successful manipulation

Fig. 3. Qualitative ablation studies for the proposed system. Subfigures (a) and (b) show an input image and a successful attribute manipulation (with added smile) using the complete proposed system. Instead, subfigures (b), (c), and (d) show problematic transfers due to the suppression of losses or anti-shortcut module.

encoding the z_{attr} values in single pixels in the output image, without changing the whole image. In some way, the network is performing an adversarial attack to itself.

The main causes for shortcuts are:

- Large generated space dimension: RGB images have a resolution of 128×128 and three channels for a total of $256 \times 256 \times 256$ color values for each pixel. Hiding information in such large space is easy for the network.
- Locality: It is easy for a network to hide an attribute in a single location in the form of a pixel (e.g., it simply encodes z^0_{attr} as the top left pixel of the image, as a grayscale value).

To prevent the network to take such shortcuts, we have implemented three appropriate solutions applied to the images before the MIM encoder step (see Figure 2, bottom center):

- Pooling: It reduces the space dimensionality, addressing the first cause of shortcutting; while the full resolution is retained for the discriminator to create good quality images, pooling is applied to the input of the encoders;
- Dropout: By randomly shutting down some pixels, we prevent the network from encoding information in few locality-stable pixels (solving the locality problem);
- Color quantization: We reduce the color depth of the images to 2 or 3 bits per pixel per color channel (instead of 8). This strongly limits the networks in exploiting this space for shortcuts.

3.3 Multiple Attribute Sizes

The MIM encoder devoted to reconstruct attributes can struggle when a high number of attributes are used. Moreover, experiments demonstrated that a single encoder tends to focus more on large-scale attributes, discarding the small-scale ones. For example, when working with human faces, the network would learn attributes like hair color or length, while not learning small-scale attributes like smile opening or nose size. This behaviour is due mainly to the pooling layer, which prevents shortcuts, but reduces the resolution of the image fed into the encoder.

To solve this problem, the attribute reconstruction task is divided into a number of smaller tasks executed by a series of encoders $\mathbf{E} = \{E_1, \ldots, E_m\}$, each of which is preceded by a different pooling function. Pooling's role is twofold: reducing shortcuts (as described in Section 3.2) and guiding each encoder in learning attributes at different scale. More specifically, a large pooling size would erase fine details, therefore pushing the encoder to focus more on large-scale attributes, while a small pooling size would allow the encoder to focus on small-scale attributes.

As a consequence, the *attribute reconstruction loss* $\mathcal{L}_{attr}$ is reformulated as follows:

$$\mathcal{L}_{attr} = \left\| E_1(\hat{x}) - z_{attr}^{[0:\alpha)} \right\|_2 + \left\| E_2(\hat{x}) - z_{attr}^{[\alpha:2\alpha)} \right\|_2 + \cdots + \left\| E_m(\hat{x}) - z_{attr}^{[(m-1)\alpha:m\alpha)} \right\|_2, \tag{3}$$

where m is the number of MIM encoders and $\alpha = |z_{attr}|/m$. In this notation, $z_{attr}^{[0:\alpha)}$ represents the first subset of z_{attr} including elements from z_{attr}^0 to z_{attr}^α (excluded).

Alternatively, we can consider the concatenation of MIM encoders outputs $\mathbf{E}(x) = E_1(x) \oplus E_2(x) \oplus \cdots \oplus E_m(x)$ and write the attribute reconstruction loss as

$$\mathcal{L}_{attr} = \|\mathbf{E}(\hat{x}) - z_{attr}\|_2. \tag{4}$$

3.4 Preserving Identities

A major requirement of our system that is common to most of image manipulation frameworks is maintaining the class/identity of the input image during the translation to the output domain. In supervised methods like Reference [10] this is solved by a reconstruction loss that forces the network to reverse the output back to its original source domain.

Unfortunately, given the truly unsupervised approach of our method, the network is unaware of the attributes contained in the images. One possible solution would be to extract the attributes of the input image using the MIM encoder and then apply them to the generated sample to obtain back the original source image (a cyclic loss). However, the factors of variation discovered by the network and applied to the input sample could be not invertible (e.g., for *bangs*, once erased, it would be impossible to reconstruct them identical to the original). Therefore, a loss designed to maintain the identity should be independent from z_{attr} to be applicable also to the not-invertible manipulations.

For this reason, a perceptual loss between the input and the output of the generator capable of maintaining the overall content during the translation is preferable. Using as perceptual metric the LPIPS [52] the loss is defined as

$$\mathcal{L}_{prc} = \phi\left(x, G\left(x, z_{attr}\right)\right), \tag{5}$$

where ϕ is the LPIPS perceptual distance.

Summarizing, the global training loss becomes

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_1 \mathcal{L}_{attr} + \lambda_2 \mathcal{L}_{prc}, \tag{6}$$

where λ_1 and λ_2 are the attribute reconstruction loss and the perceptual loss weights, and need to be carefully balanced. After some training epochs, the network converges in producing realistic images with manipulated attributes.

3.5 Overall Network Architecture

By putting all these pieces together, we designed the overall network architecture, which is shown in Figure 2 and is composed by a generator G, a discriminator D and a set of encoders $\mathbf{E}$. All the sub-networks (G, D, and $\mathbf{E}$) use Switchable Normalization [34]. First, the attribute vector z_{attr} is fed into a linear layer and reshaped to the dimension $h \times w \times c$, where h and w are the height and width of the source image x and $c = |z_{attr}|$ is the number of attributes. The reshaped vector is concatenated to x and used as input for the generator which performs 4 downsampling until a bottleneck composed by three residual blocks and then four upsampling until the desired resolution.

The discriminator D follows the DCGAN architecture [41] and is used to distinguish between real and fake samples.

Finally, the MIM encoders $\mathbf{E}$ are very similar in structure to D, but they are all preceded by a pooling layer and an anti-shortcut module composed by dropout and color quantization (as described in Section 3.2). Each MIM encoder takes as input the generator output $\hat{x}$ and reconstructs a subset of z_{attr}, as described in Section 3.3. In our case, with three encoders, we choose to set the pooling size to 16, 8, and 2, respectively.

This multi-scale approach leads, with reference to the case of facial attributes, the first encoder to force the learning of large-scale features like *hair color* or *skin tone*, the second one to focus more on medium-scale features like *hair style* or *lightning*, and finally, the last one on small-scale features like *smile strength* or *chin length*.

4 EXPERIMENTS

To prove its effectiveness, this section will report qualitative and quantitative experiments of our system on popular datasets and different applications.

All experiments were executed on a NVIDIA GTX 1080 TI. The network was trained for 100k iterations using Adam optimizer [29] with a learning rate of 0.0001. Finally, the loss weights λ_1 and λ_2 were set to 20 and 10, respectively.

4.1 Datasets

We used these popular datasets for our experiments:

- CelebA [33]. CelebA is a large-scale (200k images) dataset of celebrity faces. Each picture is 178×218 pixels in resolution, but it is a common practice to use the center 128×128 crop of the face. It is a de-facto standard benchmark on faces generation with GANs, and has been used in many recent works, such as References [8, 10].
- MNIST [30]. MNIST is the old but gold standard dataset for digit recognition. It contains samples of the ten arabic digits, handwritten by different users. It is composed by 60,000 samples of grayscale, 28×28 images.

- AFHQ [11]. Animal Faces HQ is a relatively recent dataset, proposed alongside the work of StarGANv2. It contains animal faces grouped in three categories: dogs, cats and wildlife. Each class has about 5,000 examples, and the resolution is 512×512 pixels.
- Anime Faces Dataset [7]. This dataset contains 63,632 cropped anime face images. Images sizes vary from 90×90 to 120×120, but they were resized to 128×128.
- Flowers [38]. A dataset consisting of 8,189 images belonging to 102 flowers categories. This particular dataset was chosen to reinforce the effectiveness claim of the proposed method on images that do not contains faces.

4.2 Evaluation Metrics

Evaluating generative models is always tricky due to the lack of strong unambiguous metrics, especially in the case of unsupervised methods like ours. In this article, a threefold approach was chosen based on three different metrics that validate different aspects of the generated samples. In particular, the three metrics are:

- **Fréchet Inception Distance (FID).** FID score [19] uses the Fréchet distance [13] to measure the distance between two image distributions. This gives both information of how much the generated distribution is similar to the original one, and how much it is a good, valid distribution in general. However, FID lacks providing information about content preservation and attribute generation in an image-to-image task (two aspects that are central to our proposal).
- **Face Recognition Distance (FRD).** To express how well the content is preserved during the image-to-image translation, we looked into an identity preservation metric. In the context of CelebA, on which most of our quantitative experiments will be run, FRD [32] emerged as the ideal metric. In detail, FRD is computed as the visual distance between the input and the output, and is calculated on their embeddings extracted from a network pretrained with VGGFace2 dataset [6].
- **Attribute Generation Sensitivity (AGS).** Finally, a metric is also needed to evaluate how much and how well the attributes are generated in the output image. For this purpose, we introduce the AGS metric to measure the quality of the generated attributes. In particular, for the CelebA dataset, to produce a consistent evaluation also with other methods, we trained a Resnet50 [17] classifier on the original training set (reaching 94% accuracy on the test set and 74% average per-attribute sensitivity). The trained classifier is used as an arbiter in judging the generated images. AGS is defined as the percentage of images generated with a specific attribute that effectively contain that same attribute.

To compute FID, we run the computation over the whole test set of images, and, for each image, we choose a random z_{attr} attribute to be set to +1. This way, we simulate the addition of any single attribute at standard intensity. FID score is computed between the original distribution and the one generated in this way.

FRD score is computed by first generating images as described above for FID. Then, FRD score is obtained by averaging distances between every original image and the corresponding generated images obtained from it.

On the contrary, AGS is a *per attribute* metric, i.e., it needs to be computed separately for each attribute. In particular, we pick from the dataset an attribute a to test and compare it against one of our z_{attr} attributes v. In this context, $AGS(a, v)$ attests how well the attribute v (from our system) generates images that contain label a (from the original dataset). To compute $AGS(a, v)$, we run over the whole CelebA test set while forcing our system to generate image alterations with $v = +1$. The generated images are classified by the aforementioned Resnet50 arbiter and the

Table 1. Evolution of FID and FRD when Changing the Number of Attributes

z_{attr} dim	FID ↓	FRD ↓
9	11.9	0.645
18	9.25	0.750
27	14.74	0.840

obtained classification sensitivity (i.e., the percentage of images containing the desired attribute) represents the value of $AGS(a, v)$. Unfortunately, being our system truly unsupervised, we do not know beforehand which element v of z_{attr} mimics at best the attribute a of CelebA that we want to test. However, we can easily find the z_{attr} element $\bar{v}$ that best corresponds to a label, by computing

$$\bar{v} = \arg\max_{v} AGS(a, v) \quad \forall v \in z_{attr}. \tag{7}$$

4.3 Ablation Study

To demonstrate the effects on performance of each part of the system, an extensive ablation study was performed.

First, in Table 1, we show the evolution of the results when changing the dimension of z_{attr}. For this test, we experimented with a z_{attr} length of 9, 18, and 27 meaning that our model will try to discover that number of attributes in the dataset. Indeed, it can be seen how decreasing the number of attributes will result in a lower FRD but on the expense of the number of attributes discovered. However, increasing the number of attributes (e.g., to 27) will results in a drawback in the quality of the generated images and, even if more fine-grained attributes (such as "direction of gaze") will be found, most of the additional discovered attributes will have little meaning and produce only a slight change in the output images. For these reasons, all the following tests will be performed with a z_{attr} length of 18.

Regarding the network architecture, the proposed system is composed by three main components and each of them is fundamental to obtain the desired result of attribute discovery and manipulation: the generator network takes as input an image x and an attribute vector z_{attr} and produces a translated image $\hat{x}$ which contains the attributes defined in z_{attr}, the discriminator forces the generator to produce realistic samples and the set of MIM encoders reconstructs the original attributed vector z_{attr} from $\hat{x}$ forcing the generator to properly apply the attributes on the generated image. To prove this, Table 2 shows how the losses are used to enforce the expected behaviour from the different parts on the system. First, the adversarial loss was detached. Looking at the AGS and FID scores, without the help of the discriminator, the network failed completely in producing meaningful attributes and just applied random dark and white squares artifacts over the images. This can be seen also in Figure 3(b) reported in Section 3.2. Next, the perceptual loss $\mathcal{L}_{prc}$ was disabled and the results proved it to be fundamental to maintain the content during the translation, as clearly expressed by the FRD score. Examples of the generated samples without $\mathcal{L}_{prc}$ are presented in Figure 3(c).

Finally, the usefulness of the anti-shortcut modules is evaluated. More in detail, each of them (multiple pooling, dropout and color quantization) was toggled or detached in different combinations to see their individual effect on the results. These experiments are shown in Table 3. Overall, the best results were obtained enabling all the anti-shortcut modules. However, it can sometimes happen that, for a single attribute, a different combination of anti-shortcut modules produces almost the same result as the fully enabled configuration (like in the case of bangs, where the dropout

Table 2. Ablation Studies for the Different Losses

				Attribute Generation Sensitivity (%) ↑				
$\mathcal{L}_{prc}$	$\mathcal{L}_{adv}$	FID ↓	FRD ↓	Bushy Eyebrows	Pale Skin	Smiling	Bangs	Black Hair
✓		37.18	0.480	23.1	4.9	50.7	19.3	24.2
	✓	31.65	1.198	16.6	38.8	82.9	50.7	35.9
✓	✓	9.25	0.750	41.7	27.9	72.5	32.6	35.3

The full loss proposed in the article has the best FID by far, meaning that the other distributions are quite lacking. Also, by the removal of the perceptual loss, the Face Recognition Distance rockets close to 1.2, meaning that the identity is lost. Still, without the perceptual loss, the network is sometimes able to produce more convincing attributes (such as smiling, bangs).

Table 3. Ablation Studies for the Different Anti-shortcut Methods

					Attribute Generation Sensitivity (%) ↑				
Dropout	Color Quantization	Pooling	FID ↓	FRD ↓	Bushy Eyebrows	Pale Skin	Smiling	Bangs	Black Hair
			6.96	0.419	6.1	13.6	44.8	14.4	25.4
		✓	7.62	0.490	14.1	5.7	49.1	15.6	12.7
	✓		6.08	0.235	9.6	5.5	45.0	15.1	19.2
✓			5.53	0.302	8.9	4.0	44.8	15.7	17.1
	✓	✓	16.48	0.825	20.5	20.4	60.3	35	15.5
✓	✓		8.57	0.519	17.5	6.8	47.5	16.2	23.1
✓		✓	7.54	0.576	23.9	21.9	52.3	17.2	21.2
✓	✓	✓	9.25	0.750	41.7	27.9	72.5	32.6	35.3

The table shows that the full anti-shortcut module (Dropout + Color Quantization + Pooling) outperforms the other combination of methods in Attribute Generation Sensitivity, while showing only modest degradation of FID and FRD.

has minimal impact on the AGS). These cases are very rare and they simply reflect the fact that some anti-shortcut modules are more effective on some attributes than others. However, the average attribute generation sensitivity strongly advocates for the use of all the three modules.

To further prove these findings, Figure 4 depicts how the metric of AGS responds to a variation in attribute generation intensity. In other words, instead of imposing a +1 in the desired z_{attr} element, we tested a wider range [−2, +2] for that generation attribute. Indeed, the full proposed module allowed to obtain almost linear correlation between the attribute intensity in z_{attr} and the AGS, proving that the network was able to effectively grasp the attribute and to produce continuous interpolations of it.

On the opposite, as it is especially visible in the *Black_Hair* and *Smiling* plots, without dropout, color quantization or pooling the network failed to describe the attribute (flat line dynamics).

4.4 Experiments on Set of Attributes Change

Once fully trained, our architecture makes available at inference time two useful networks:

- the image-to-image generator G, which can be used to change a set of attributes in an input image by manipulating z_{attr};
- the MIM encoder(s) **E**, which are used to discover any factor of variation (z_{attr} array).

Given these two trained networks, there are several potential applications for which our network can be used.

The first natural application for the proposed system is altering one or more attributes in an input image. The goal is to select a new, unseen, input image x, and change some of its z_{attr} attributes. It is worth remembering that, given its innovative design, the system enables the independent

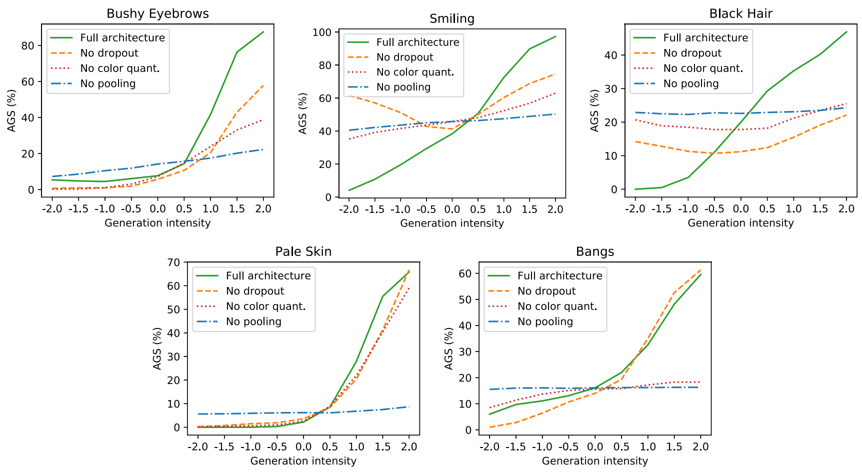

Fig. 4. Ablation studies for the different anti-shortcut methods. The $[-2, +2]$ generation range has been explored. As can be seen, the full architecture has the best AGS values at +1 and +2 intensity. Moreover, the full architecture plot exhibits a strong correlation between generation intensity and AGS, whereas the other combinations often depict a flat line (denoting absence of correlation).

manipulation of each attribute (thanks to the disentanglement) in a controllable and continuous way.

However, to change only one relevant attribute, we first need to know the original set of attributes that refers to the input image. More in detail, to extract the initial set of attributes from an image, we can simply use the MIM encoders on it:

$$z_{attr-initial} = \mathbf{E}(x). \tag{8}$$

Retrieving the starting set of attributes $z_{attr-initial}$ for the input image x allows to alter only some attributes, by leaving the others untouched from the input image (as opposite of what has been done during the training phase, where all of the attributes were modified at random).

Once we have the initial set of attributes, one or more of its elements can be changed to obtain the modified set $z_{attr-modified}$, and use it within the generator G to create the final, manipulated image $\bar{x} = G(x, z_{attr-modified})$.

In this way, as shown in Figure 1 at the beginning of the article, the network can generate an output image with the same content of x, but with the selected attributes changed. Indeed, being the system truly unsupervised, the discovered attributes are not labelled. Therefore, to change a specific attribute, this needs to be identified within z_{attr} by trial.

4.4.1 Qualitative Analysis. The initial experiments on single attributes change were conducted on MNIST dataset. Given the simple nature of the dataset, only a single encoder (with pooling 4) was employed and, by doing so, the system managed to successfully discover (in a truly unsupervised fashion) factors of variation of MNIST digits such as *rotation*, *digit height*, *digit width*, and *thickness*, as shown in Figure 5.

Then, extensive experimentation was carried out on the CelebA dataset. For this setting, a set of three MIM encoders was employed and the dimension of z_{attr} was set to 18 (6 attributes for each encoder). Some examples of continuous discovered attributes are shown in Figure 6. It is immediately clear how different pooling sizes in the MIM encoders $\mathbf{E}$ forced the generator network to semantically produce different attributes. In particular, a pooling size of 16 generates high-level

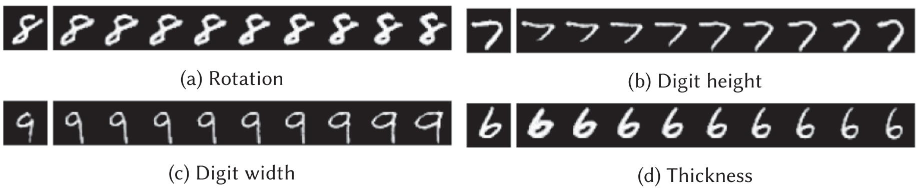

Fig. 5. Examples of continuous attributes learned on MNIST. On the left there are the input images, while each row is a different attribute variation, interpolated from −1.25 to +1.25 (i.e., the first and the last images are actually out of training limits [−1, +1]).

features, such as hair color or skin tone, while a pooling size of 8 results in mid-level features like small hair changes but also shadows and lighting. Finally, a small pooling size of 2 allows the generator to sample low-level features, such as smile size, chin and eyebrows shape.

Figure 6, besides showing a sample selection of attributes unsupervisedly learned by the system, also highlights a very useful feature of it, as compared to other methods such as StarGAN or AttGAN: complete uniform interpolation of learned attributes. The generated attributes intensities can be freely varied from −1 to +1 leading to continuous (if not linear) changes in the output images. These intensities can also be pushed out of the training range (e.g., intensity +2) obtaining more "aggressive" attributes (see Figure 7); this remarks a good attribute stability of the proposed system.

As a final experiment, we aim at demonstrating how well these automatically learned attributes generalize across different identities. The proof of this can be seen in Figure 17 of the appendix that shows a selection of identities, randomly chosen in the test set of CelebA, where it can be appreciated that the learned attributes are consistent across identities.

4.4.2 Quantitative Analysis. To more objectively score and evaluate our system, we computed the quantitative metrics listed in Section 4.2 and compared them with other systems. Unfortunately, in addition to the lack of acknowledged evaluation metrics for image-to-image translation methods (as discussed in Section 4.2), we also faced the issue of choosing the method(s) to compare with.

In fact, to the best of our knowledge, our proposal is the first method to perform end-to-end truly unsupervised real-image fine-grained attribute manipulation. Therefore, our choice for quantitative comparison fell on supervised image manipulation methods that provide end-to-end image generation. Methods like AttGAN [18], StarGAN [10], and RelGAN [48] are very similar to our proposal in terms of task, but they are fully supervised. In this context, *full* supervision means that they need ground truth labels during both the training and the inference. As a further difference, while AttGAN and StarGAN provide n binary attributes ([0, 1], e.g., *Smiling/Not_Smiling*, *Bangs/Not_Bangs*), our method and RelGAN accept n real numbers and are trained in generating continuous attribute variations (as shown in Figure 6).

The comparison with these methods is shown in Table 4. Table 4 shows that our method beats every other method in FID, particularly StarGAN and RelGAN. The FRD score of our proposal is similar to other methods (slightly better than StarGAN), while RelGAN is outperforming the others on this metric. Clearly, these results are already positive for our method: it generates good quality distributions on par or even better than *supervised* methods.

Regarding the metric of AGS, since AttGAN, StarGAN, and RelGAN are supervised, we simply set the tested attribute label (e.g., *Smiling*) to +1 and add that attribute to all the images in the CelebA test set.

As already commented and foreseeable, FID/FRD and AGS scores measure auspicable, yet conflicting, features of the generation networks: The former scores correspond to good quality images (but potentially with no attribute changed), while the latter score evaluates how good is the

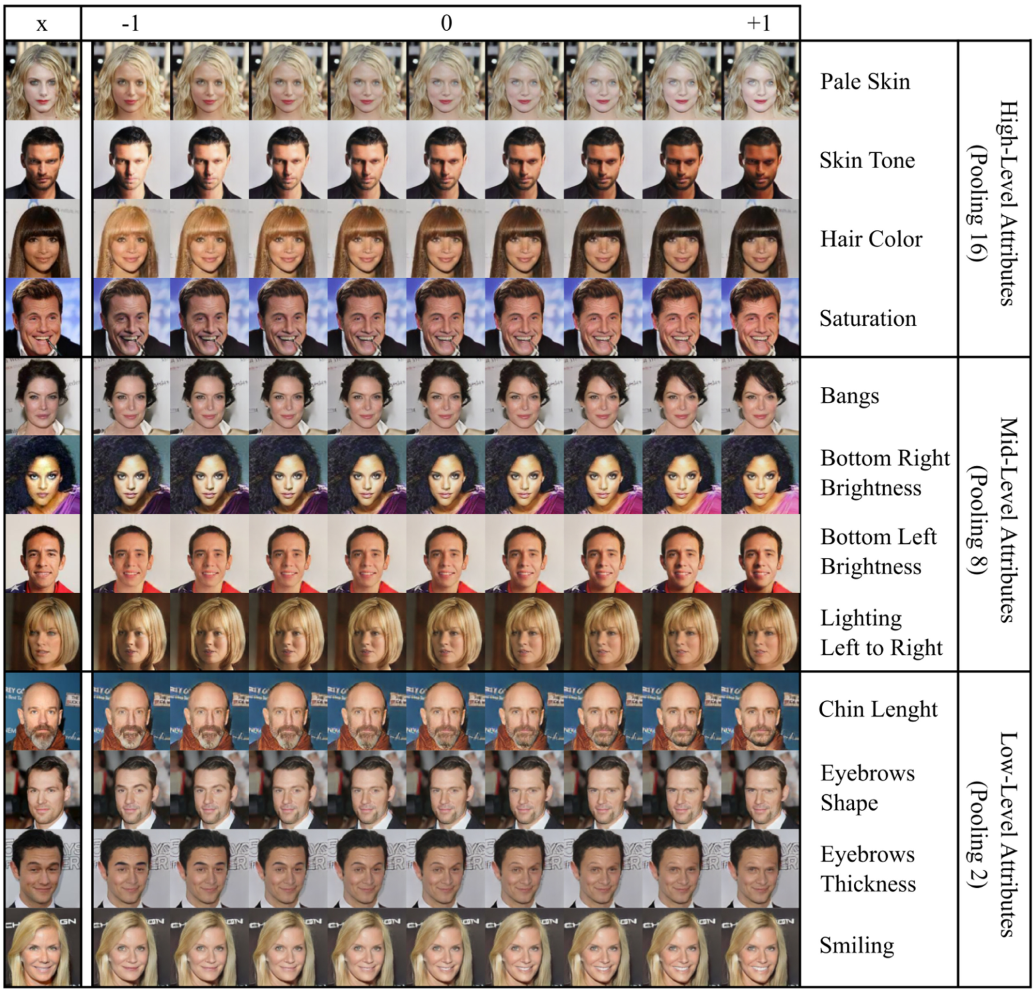

Fig. 6. Attributes learned on the CelebA dataset. First column represents the input image x and each subsequent column contains an output sample produced by altering one element of the attribute vector $z_{attr-initial}$ (extracted from $\mathbf{E}(x)$) from -1 to 1. Finally, each row contains a different attribute obtained altering a different element of $z_{attr-initial}$ each time. It is important to note that the value 0 does not represent a neutral state, but rather apply an attributes with half the intensity.

Table 4. Comparison with other Image-to-image Attribute Alteration Methods

				Attribute Generation Sensitivity (%) ↑				
Method		FID ↓	FRD ↓	Bushy Eyebrows	Pale Skin	Smiling	Bangs	Black Hair
StarGAN [10]	Superv.	19.74	0.812	88.2	91.5	97.9	95.8	90.6
AttGAN [18]	Superv.	9.57	0.713	83.9	68.0	96.8	83.8	55.3
RelGAN [48]	Superv.	12.04	0.415	36.7	56.2	94.4	81.1	55.8
Ours	Truly unsup.	9.25	0.750	41.7	27.9	72.5	32.6	35.3
Ours (+2 intensity)	Truly unsup.	11.55	0.834	87.5	65.8	97.3	59.5	46.9

This is a very unfair scenario for our method, since all of the compared systems are “fully supervised” (labels are required during both training and inference).

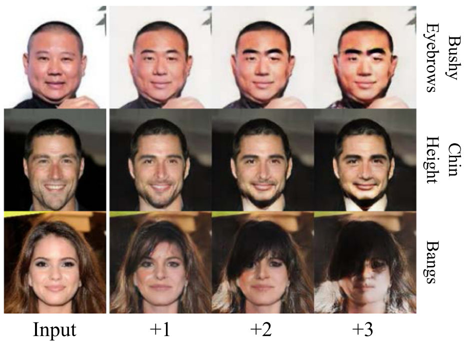

Fig. 7. Images created by the network by pushing the generation intensity out of the training range (which was [−1, +1]). Some attributes exhibit better out-of-domain stability than others; nevertheless, the generative stability of the proposal is confirmed.

network in applying factors of variation. In fact, while StarGAN achieves the best AGS in all the attributes and w.r.t. all the other methods, it also presents the worst scores of FID and FRD, meaning that the attributes are successfully changed, but the resulting images are changed in identity and distribution.

Likewise, our method underperforms the others in attribute generation sensitivity, which is however reasonable and explainable by the fact that is truly unsupervised. Even more, it should be noted that, to fairly compare with other methods on CelebA labels, we select the automatically discovered attributes in our method, which are the most similar to those in CelebA. However, these attributes are not exactly the same, and this tends to lower the AGS for our method, compared to others. For instance, the attribute *Pale_Skin* is not exactly learned by our network, which instead learns an attribute that corresponds more to "face brightness." In other words, our network learns well human-understandable concepts like face brightness, but is not tailored (due to its unsupervised training) to learn unrepresentative, rare attributes such as *Pale_Skin* (which represents 4% of the whole CelebA dataset).

Table 4 also reports the last row with +2 intensity of attribute change for our method, instead of +1. This test was performed to demonstrate that, given our complete control over the attribute changes and their intensities, we can overcome to the limitation of AGS by applying stronger variation to the attributes. In this way, we can achieve AGS values close to (or even better than) AttGAN and RelGAN, at the cost of worse FID/FRD scores (but still in a truly unnsupervised fashion). In other words, while the +1 intensity setting reproduces better the input distribution, the +2 intensity setting focuses more on creating convincing attributes.

The plots reported in Figure 8 support this claim. They have been generated by varying the generation intensity in the range [−2, +2]. It is evident that increasing the strength of the attribute is feasible and somehow linear, and allows to reach an AGS value comparable with AttGAN and RelGAN for most of the attributes. Moreover, the plots show a strong correlation between the generation intensity and the actual generated attributes. This correlation is the most evident proof that our method is able to fully grasp a human interpretable attribute in a continuous, interpolable and often linear way.

4.5 Qualitative Analysis on other Datasets

To further prove the effectiveness of the proposed system, experiments were carried out also on additional datasets (AFHQ, Anime faces and Flowers), proving that our architecture is not tailored to work only on a single category of images like human faces. Qualitative results of these tests are

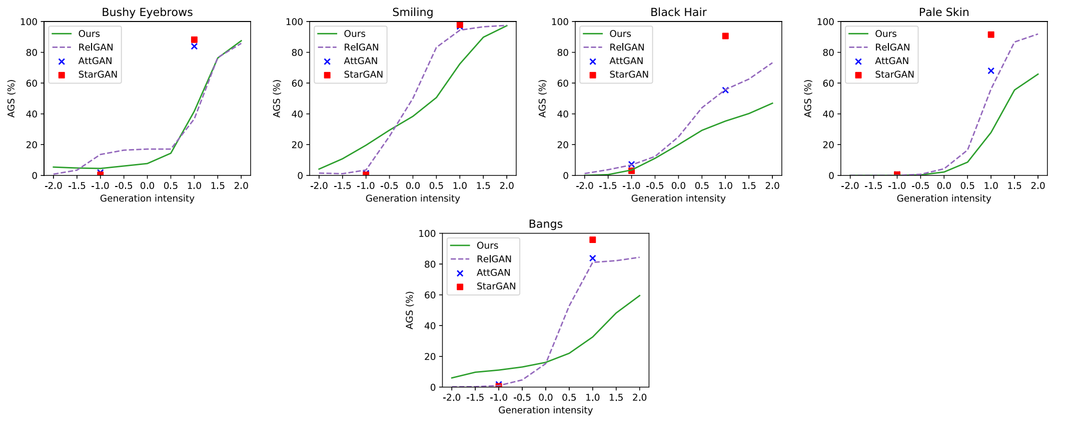

Fig. 8. Comparison with other image-to-image attribute alteration methods in terms of AGS/Generation intensity dynamics. Despite our method being the only truly unsupervised one, it exhibits performance comparable to supervised alternatives on most attributes. The linear generation dynamic further proves the claims of continuous attribute discovery.

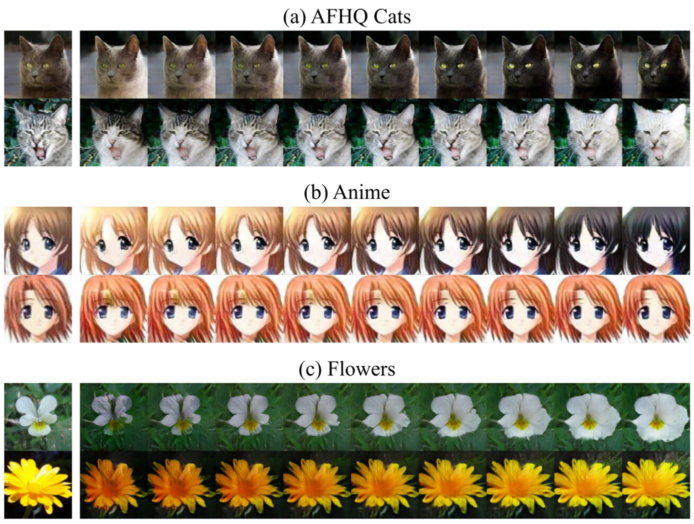

Fig. 9. Some attributes learned on additional datasets.

presented in Figure 9. Indeed, the network was able to discover factors of variations even in these alternative settings.

Given the different nature of these new datasets, some small finetuning in the training parameters was necessary. In particular, they are composed by a much smaller number of images than CelebA. Therefore, classic data augmentation (horizontal flip and affine transforms) was performed during training to artificially augment the dataset size. Finally, the network was trained for 50k iterations instead of 100k, since that proved to be enough to obtain good results.

4.6 Comparison with TUNIT and StarGANv2

Recently, both TUNIT [4] and StarGANv2 [11] proposed two competitive methods for multi-domain image synthesis. We argue that they both perform a very different task than our method and, in particular, they are not able to discover and apply fine grained attributes continuously.

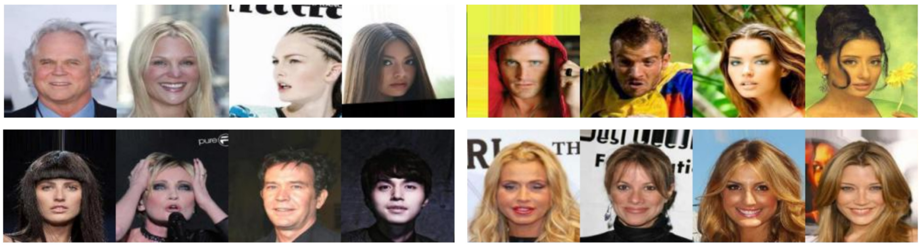
Fig. 10. Some clusters found by the TUNIT unsupervised algorithm.

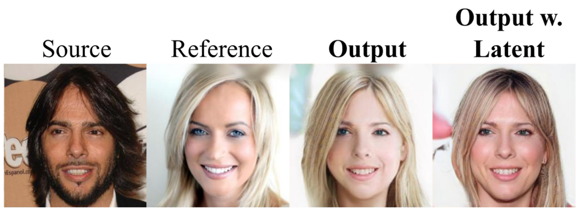

Fig. 11. Result applying the "blond hair" attributes with StarGANv2 using reference and latent code style.

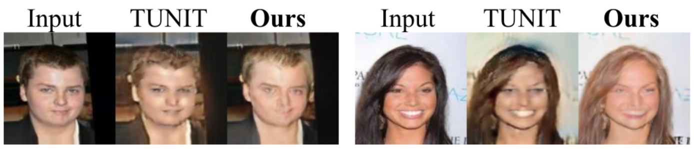

Fig. 12. Comparison with TUNIT on "blond hair."

To prove it, we first trained TUNIT on the CelebA dataset with the number of domains to be found by the unsupervised clustering algorithm set to 18. The objective was to see if TUNIT would have been able to find several meaningful fine-grained attributes as our proposed method. On the contrary, given that its architecture was designed to perform a global domain transfer (which is implicit in the use of AdaIN layers) TUNIT failed this task as it can be seen in Figure 10. Indeed, it was only able to separate images based on global attributes such as background color or hair color (bottom right cluster can be identified as "blond hair," see Figure 11 for a comparison with our method).

Finally, we wanted to prove if StarGANv2 was able to apply a target domain locally on an input image. To do so, we trained it over a simple domain such as "blond hair" and generated some samples using both a reference image and a latent style code. An example can be seen in Figure 12. Indeed, it is clear that StarGANv2, even if the generated result is visually pleasing, cannot apply the target attribute locally but, on the contrary, completely change the look and gender of the generated sample. For this reason, other than being fully supervised, it is not comparable with our method.

4.7 Qualitative Results on other Applications

Most of our experiments were related to the application of set of attribute change. However, to prove the effectiveness of the proposed network also in other scenarios, the following subsections will show qualitative results also on style transfer and data augmentation.

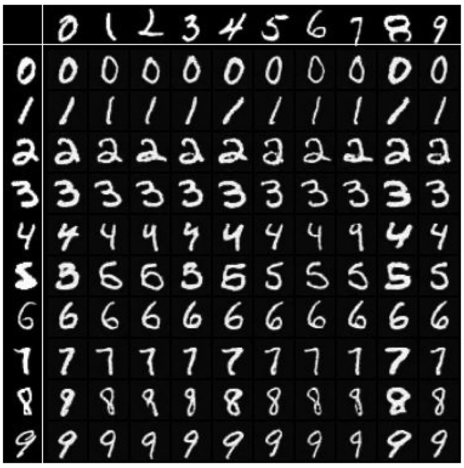
Fig. 13. Style transfer matrix on MNIST. Left column shows the content images, while top row the style images. Attributes like digit rotation, thickness, width, and height are well transferred with the style, while preserving the original content.

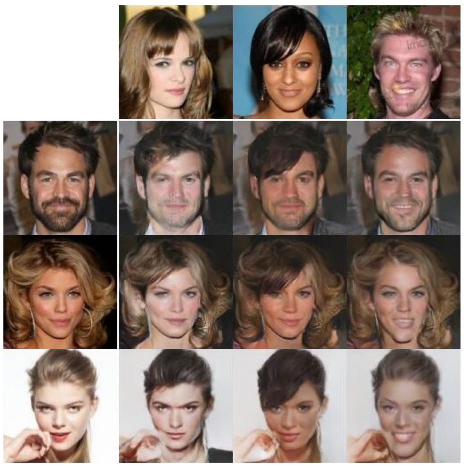
Fig. 14. Style transfer matrix on CelebA. Left column shows the content images, while top row the style images. Attributes like bangs, face tone, smile, sad expression, chin height, and hair color are transferred with the style, while preserving the original content.

4.7.1 Style Transfer. In recent years, image-to-image GANs have been extended to the application of style transfer (e.g., in References [11, 20, 31]), where the generator receives a content image c and a style image s and tries to generate an output image o with the content of c and the style of s. Actually, most of the papers addressing style transfer, only apply the pose of the content image c with the identity of the style image s, as also mentioned in Section 2, while our system transfers the whole set of discovered attributes from the style image to the content image.

More in detail, style transfer can be easily obtained by using the generator G on the content image c with the attributes extracted by the MIM encoders $\mathbf{E}$ from the style image s: $o = G(c, \mathbf{E}(s))$.

Obviously, the style that can be transferred is limited to the set of attributes discovered by the network while training.

We run style transfer qualitative tests both on the MNIST dataset and the CelebA dataset. As can be seen in Figures 13 and 14, the attributes of the first row pictures (styles) have been transferred to the contents of the first column pictures.

4.7.2 Data Augmentation. Another possible application of our system is data augmentation. In fact, the trained generator can generate new variations of a given image (of a certain class) by randomly changing the learned attributes. The augmented data can be then used to train, for instance, a classifier (such as a face recognition system) with more samples. What is crucial for success of data augmentation is to generate images that are as much realistic as possible, and this is granted in our system by the fact that the network has learned the original distribution $\mathcal{X}$, and the uniform mapping of some of its attributes.

Data augmentation can be easily obtained by using the generator G with the real image x and a randomly uniform vector $\hat{z}_{attr}$: $\hat{x} = G(x, \hat{z}_{attr})$.

Qualitative results for the data augmentation task are shown in Figures 15 and 16, for MNIST and CelebA, respectively. Pictures show that generated images are diverse, the identity is preserved, and fake images are indistinguishable from the original ones.

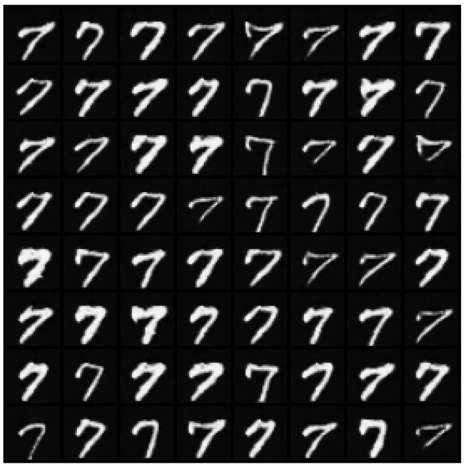

Fig. 15. Visual example of data augmentation over MNIST dataset with our method. All of these digits were data augmented from a single sample.

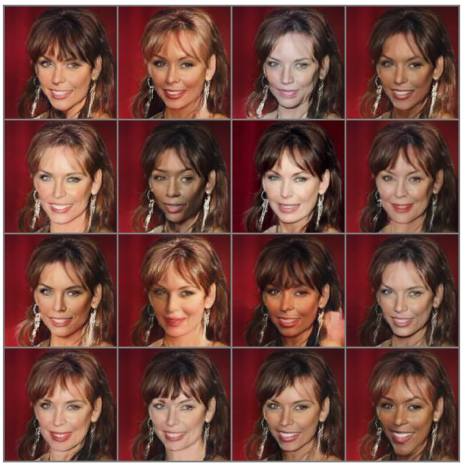

Fig. 16. Visual example of data augmentation over CelebA dataset with our method. All of these faces were data augmented from a single sample, except one, the original, that is mixed amongst them. A human observer can hardly distinguish the original sample in such a picture, giving credit to our system generation quality. Our guess for the original picture location is for the picture in second row, third column.

Table 5. Classification Accuracy over MNIST Validation After Our Version of Data Augmentation, Compared with Traditional Computer Vision Data Augmentations

Method	Classification accuracy on MNIST validation (%)	Method	Classification accuracy on MNIST validation (%)
1-shot tradit. augm.	25.82	5-shot tradit. augm.	35.05
1-shot our method	90.50	5-shot our method	93.44

1-shot means one sample per class data augmentation, while 5-shot means five samples per class.

Moreover, we obtained some quantitative results by the creating a benchmark task for data augmented single-shot classification. This task is formulated as follows: A human expert picks one image sample per class over a dataset. The proposed network will generate data augmentation using that sole image (per class), i.e., under a scenario of few-shot classification. Afterwards, the generated data-augmented distribution will be used to train a classifier able to work on the original dataset and new samples alike. This is both a valid application and a suitable task to evaluate how well the network can generate variations (diversity) without compromising the identity (class).

We trained an off-the-shelf classifier (e.g., ResNet18 [17]) using only the set of augmented images. Table 5 shows the quantitative results in terms of classification accuracy for original MNIST dataset. It is clear that the augmented data greatly helped the classifier, which was able to reach 90.5% classification accuracy over the original MNIST validation set. To perform a comparison with more traditional data augmentation techniques and to find a baseline for our method, we used the same ten samples (one for each digit/class in MNIST) and generated augmented samples using rotations, translations, shears, zooms. In this case (first row of Table 5) the classifier only reaches an accuracy of 25.82%, confirming that the data generated with our network are much more plausible and diverse and helped the classifier to perform better. Table 5 also shows another setting where five samples are selected for each digit/class. The results also in this setting show that our data augmentation is better than classical ones.

5 CONCLUSIONS

The proposed network has proven that is possible to create an image-to-image GAN capable of learning and modifying continuous image attributes in a truly unsupervised manner. We have proven this claim through complete ablation studies and comparisons with state-of-the-art *supervised* networks on the same task. This comparison proved that we are on par and sometimes better than supervised methods in image generation quality and identity preservation, and comparable, even if moderately lower, in attribute generation quality. Moreover, we have tested this system in three different applications (set of attribute changes, style transfer and data augmentation) and across diverse datasets (CelebA, MNIST, AFHQ, Anime, Flowers). The results highlight the wide applicability of the proposed network.

This article also proves that the MIM framework is not limited to the noise-to-image GAN scenario (such as in References [8, 37]), but can also be applied to end-to-end training of image-to-image GANs. However, our work also made evident that MIM alone is not suitable to work in high dimensional spaces like image-to-image GANs, since it is prone to the creation of shortcuts. Opportunely, our work provides valid solutions to this critic problem, in the form of anti-shortcut modules placed before each MIM encoder. A multi-scale set of MIM encoders is also introduced to allow the network to learn features of different sizes.

Finally, the proposed network can provide a continuous (sometimes linear) and controllable variation of the generated attributes, where many state-of-the-art supervised methods only permit binary variations of the attributes.

A ONLINE APPENDIX

A.1 Additional Results About the Consistency Across Multiple Source Identities

Figure 17 shows the consistency of the attributes learned with the proposed method. The first row depicts a set of input images (identities). Every other couple of rows represents the variation of a single Z_{attr} attribute across these identities. Each row represents the two extremes of a learned attribute interval: +1 and −1.

A.2 Additional Quantitative Evaluation

We performed an additional quantitative comparison between our method and the one of Hu et al. [20]. More in detail, to evaluate the quality of the disentanglement, they trained a linear binary classifier over several attributes by using the latent space of their autoencoder as input. However, we trained the same linear classifier over the reconstructed $\hat{z}_{attr}$ produced by the MIM encoders. Results can be seen in Table 6. Our model is able to reach results that are comparable with the ones of Hu et al., even if it uses a much smaller latent dimension. Indeed, results of Reference [20] are obtained with a latent dimension of 512, while we managed to reach very good results with a latent dimension of only 18 (i.e., the number of attributes). This proves the optimal disentangling capability of our system. At the same time, we did also an experiment using a latent dimension of 27, and we reached even higher results, demonstrating also the flexibility of our system w.r.t. the number of discovered attributes.

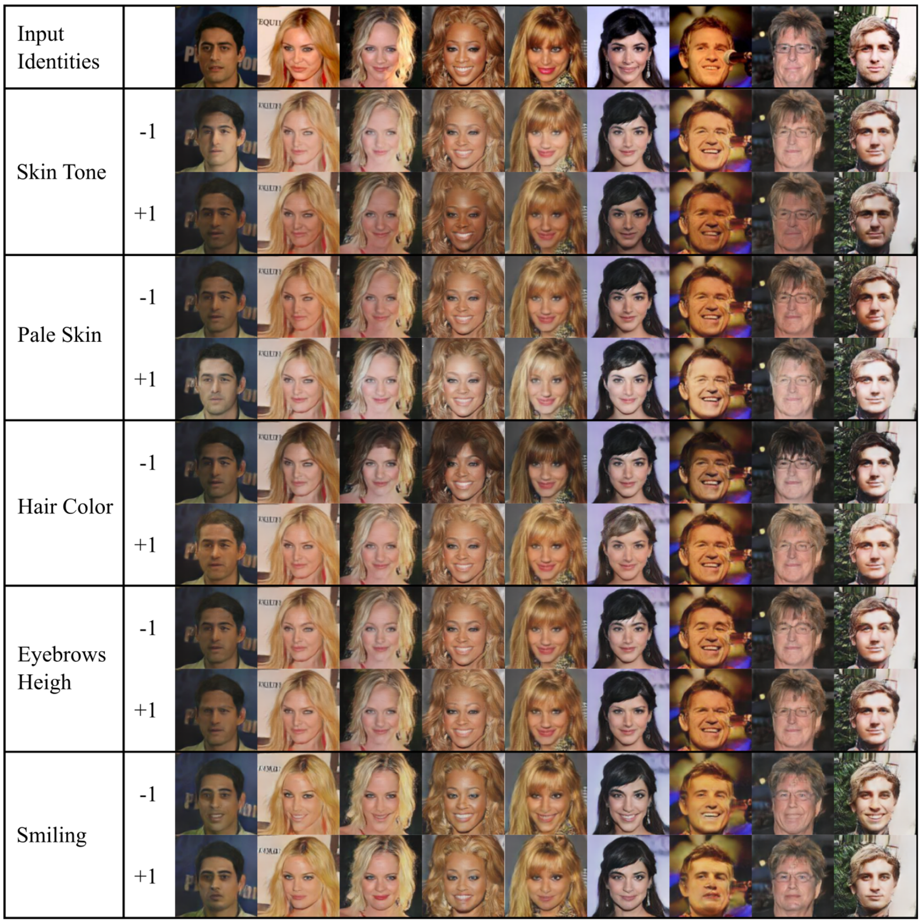

Fig. 17. Consistency of the attributes learned over CelebA dataset across multiple source identities.

Table 6. Comparison between Our Method and Reference [20]

Method	Lat. Dim.	Eyebr.	Attr.	Bangs	Black	Blond	Makeup	Male	Mouth	Beard	Wavy	Hat	Lips	Avg.
[20] (DCGAN)	512	72.2%	68.5%	88.8%	75.7%	89.9%	76.9%	80.1%	73.6%	83.8%	70.5%	95.8%	78.6%	79.5%
[20] (BEGAN)	512	73.0%	69.7%	90.2%	79.6%	89.3%	78.9%	85.4%	77.1%	88.1%	70.8%	96.4%	81.7%	81.7%
Ours	18	73.2%	60.9%	86.6%	74.6%	85.9%	62.0%	60.4%	75.9%	84.1%	69.1%	95.5%	58.1%	73.9%
Ours	27	73.4%	61.3%	87.8%	77.8%	86.2%	62.6%	61.7%	79.1%	84.0%	70.0%	96.4%	59.2%	74.9%

We reached comparable results with a much smaller latent space.

REFERENCES

[1] Yazeed Alharbi and Peter Wonka. 2020. Disentangled image generation through structured noise injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5134–5142.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning.* PMLR, 214–223.

[3] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. 2020. Rethinking the truly unsupervised image-to-image translation. Retrieved from https://arXiv:2006.06500.

[4] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. 2021. Rethinking the truly unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 14154–14163.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large-scale GAN training for high fidelity natural image synthesis. Retrieved from https://arXiv:1809.11096.

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. 2018. Vggface2: A dataset for recognising faces across pose and age. In *Proceedings of the 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG'18)*. IEEE, 67–74.

[7] Brian Chao. 2019. Anime Face Dataset. Retrieved from https://github.com/bchao1/Anime-Face-Dataset.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. Retrieved from https://arXiv:1606.03657.

[9] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. Retrieved from https://arXiv:1601.06733.

[10] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8188–8197.

[12] Tomaso Fontanini, Eleonora Iotti, Luca Donati, and Andrea Prati. 2020. MetalGAN: Multi-domain label-less image synthesis using cGANs and meta-learning. *Neural Netw.* 131 (2020), 185–200.

[13] Maurice Fréchet. 1957. Sur la distance de deux lois de probabilité. *Comptes Rendus Hebdomadaires Des Seances de L Academie Des Sciences* 244, 6 (1957), 689–692.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. Retrieved from https://arXiv:1406.2661.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein GANs. Retrieved from https://arXiv:1704.00028.

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable GAN controls. *Adv. Neural Info. Process. Syst.* 33 (2020), 9841–9850.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[18] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. AttGAN: Facial attribute editing by only changing what you want. *IEEE Trans. Image Process.* 28, 11 (2019), 5464–5478.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Info. Process. Syst.* 30 (2017).

[20] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. 2018. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3399–3407.

[21] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*. 1501–1510.

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

[23] Xu Ji, João F. Henriques, and Andrea Vedaldi. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9865–9874.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of GANs for improved quality, stability, and variation. Retrieved from https://arXiv:1710.10196.

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Adv. Neural Info. Process. Syst.* 34 (2021), 852–863.

[26] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of styleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.

[28] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1857–1865.

[29] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Retrieved from https://arXiv:1412.6980.

[30] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. Retrieved from http://yann.lecun.com/exdb/mnist/.

[31] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 10551–10560.

[32] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. 2021. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 10785–10794.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV'15).*

[34] Ping Luo, Jiamin Ren, Zhanglin Peng, Ruimao Zhang, and Jingyu Li. 2018. Differentiable learning-to-normalize via switchable normalization. Retrieved from https://arXiv:1806.10779.

[35] Michael Mathieu, Junbo Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. 2016. Disentangling factors of variation in deep representations using adversarial training. Retrieved from https://arXiv:1611.03383.

[36] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. Retrieved from https://arXiv:1411.1784.

[37] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. 2019. ClusterGAN: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4610–4617.

[38] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing.*

[39] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. 2020. Face identity disentanglement via latent space mapping. *ACM Trans. Graph.* 39, 6, Article 225 (Nov. 2020), 14 pages. https://doi.org/10.1145/3414685.3417826

[40] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. Retrieved from https://arXiv:1606.01933.

[41] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. Retrieved from https://arXiv:1511.06434.

[42] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. 2020. Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. Retrieved from https://arXiv:2007.07431.

[43] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of GANs for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 9243–9252.

[44] Yujun Shen and Bolei Zhou. 2021. Closed-form factorization of latent semantics in GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 1532–1540.

[45] Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. 2017. Challenges in disentangling independent factors of variation. Retrieved from https://arXiv:1711.02245.

[46] Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning GAN for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1415–1424.

[47] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 8798–8807.

[48] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. 2019. RelGAN: Multi-domain image-to-image translation via relative attributes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 5914–5922.

[49] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. 2020. Generative hierarchical features from synthesizing images. Retrieved from https://arxiv.org/abs/2007.10379.

[50] Masoumeh Zareapoor and Jie Yang. 2021. Equivariant adversarial network for image-to-image translation. *ACM Trans. Multimedia Comput., Commun. Appl.* 17, 2s (2021), 1–14.

[51] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning.* PMLR, 7354–7363.

[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 586–595.

[53] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain GAN inversion for real image editing. In *Proceedings of the European Conference on Computer Vision.* Springer, 592–608.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision.* 2223–2232.

Received 2 August 2022; revised 9 February 2023; accepted 2 April 2023