Discovering Interpretable Latent Space Directions of GANs Beyond Binary Attributes

Huiting Yang[1], Liangyu Chai[1], Qiang Wen[1], Shuang Zhao[2], Zixun Sun[2], Shengfeng He[1*]
[1] School of Computer Science and Engineering, South China University of Technology
[2] Interactive Entertainment Group, Tencent Inc.

Figure 1: We propose an adversarial method, *AdvStyle*, to interpret the latent space of GANs for semantics editing. It allows manipulation of arbitrary semantics, beyond the restriction of binary attributes. Our method, to the best of our knowledge, is the first latent space exploration method enables style editing. Meanwhile, we discover disentangled semantic directions, leading to accurate multi-attribute manipulation.

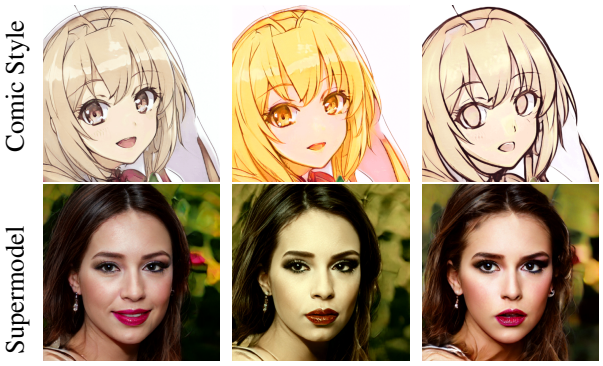
Abstract

Generative adversarial networks (GANs) learn to map noise latent vectors to high-fidelity image outputs. It is found that the input latent space shows semantic correlations with the output image space. Recent works aim to interpret the latent space and discover meaningful directions that correspond to human interpretable image transformations. However, these methods either rely on explicit scores of attributes (e.g., memorability) or are restricted to binary ones (e.g., gender), which largely limits the applicability of editing tasks, especially for free-form artistic tasks like style/anime editing. In this paper, we propose an adversarial method, AdvStyle, for discovering interpretable directions in the absence of well-labeled scores or binary attributes. In particular, the proposed adversarial method simultaneously optimizes the discovered directions and the attribute assessor using the target attribute data as positive samples, while the generated ones being negative. In this way, arbitrary attributes can be edited by collecting positive data only, and the proposed method learns a controllable representation enabling manipulation of non-binary attributes like anime styles and facial characteristics. Moreover, the proposed learning strategy attenuates the entanglement between attributes, such that multi-attribute manipulation can be easily achieved without any additional constraint. Furthermore, we reveal several interesting semantics with the involuntarily learned negative directions. Extensive experiments on 9 anime attributes and 7 human attributes demonstrate the effectiveness of our adversarial approach qualitatively and quantitatively. Code is available at https://github.com/BERYLSHEEP/AdvStyle.

1. Introduction

Generative adversarial networks (GANs) [6, 11, 19, 20] have been demonstrated power in generating high-resolution photo-realistic images by training with massive diverse data. The rationale of GANs is to learn a non-linear mapping function from the input noise latent codes to output images that conform to real data distributions. Several works reveal the vector arithmetic property of the latent space, *e.g.*, adding a learned vector to the latent code [29] or combining the latent code of two images [20], result in modifying image semantics. Although it is still uncer-

*Corresponding author (hesfe@scut.edu.cn).



(a) Origin (b) [30] (c) Ours

Figure 2: The state-of-the-art method [30] can only deal with distinct binary attributes. Non-binary attributes like comic style violate its binarily separable hyperplane assumption, leading to unsuccessful or entangled semantic manipulation. Our method does not suffer from these problems.

tain how semantics are structured in the latent space, these prior works drive researches to interpret the latent space of GANs.

A few very recent works aim to discover the meaningful directions that correspond to interpretable image transformations in the way of unsupervised or supervised learning. Despite unsupervised methods can discover directions of simple image transformations like zooming or translation [17, 28], or heuristically searching for unexpected ones like background removal [33], they cannot precisely locate user-desired target attributes.

On the other hand, supervised approaches possess better controllability. This line of study leverages a target attribute accessor to help tracing back the corresponding direction in the latent space. Specifically, Goetschalckx *et al*. [10] use the evaluator [21, 23] trained by well-scored attribute datasets to obtain the directions of memorability, aesthetic, etc. Tewari *et al*. [32] leverage annotated 3D data for mapping the control space of a 3D morphable face model to the latent space of GANs, so that they can control three semantic face parameters (pose, expressions, and scene illumination). However, well-scored attributes or 3D data (see Tab. 1 for an example) are expensive to obtain and therefore the practicability is limited. Shen *et al*. [30] extend the accessor to a pretrained binary classifier to construct a hyperplane in the latent space for discovering binary attributes like gender or eyeglasses. Notwithstanding the success of this method in editing binary attributes, there are many more attributes that are not binarily separable, and therefore their assumption on the constructed binary hyperplane is invalid for non-binary attributes. If we apply binary-based method by simply classifying the target attribute as positive (*e.g*., comic style) while leaving all the others as negative, the learned directions will correspond to incorrect and entangled semantics (see the second column of Fig. 2), as the manually classified negative samples provide ambiguous, or even misleading guidance.

Method	Annotation type	Annotation example (attribute: value)
[10]	Scored attribute	Memorability: ★★★☆☆
[32]	3D annotation	3D vertices
[30]	Binary attribute	Young and Old: 0/1
Ours	Positive attribute only	Supermodel style : 1

Table 1: Different types of annotations required by supervised latent space exploration methods.

In this paper, we aim at interpreting the latent space of GANs beyond binary attributes. To this end, we propose an adversarial method, AdvStyle, that takes only target positive samples for training, and the attribute assessor is trained to distinguish the target positive samples and the generated negative samples. In this way, our approach focuses only on finding the positive direction that the generated images are indistinguishable from the target data, producing disentangled directions of various semantics. As a result, AdvStyle can perform multi-attribute editing without any orthogonal constraint. Moreover, we dynamically update the attribute assessor, rather than using the pre-trained ones. This helps to bridge the domain gap between the generated images and the training data of the attribute assessor. The involuntarily learned negative directions, on the other hand, reveal some interesting and unexpected semantics. Some results of our method are shown in Fig. 1.

Our contributions are summarized as follows:

- We propose an adversarial method to discover the directions of arbitrary attributes in the latent space of a pre-trained GAN, leading to intuitive editing on non-binary attributes beyond the limitation of manual annotations.
- We show that the adversarially learned directions are well-disentangled, therefore multi-attribute manipulation can be done without additional constraint.
- We further study the interesting and unexpected semantics which are involuntarily captured by our negative directions, it helps revealing how the latent space is organized.
- We extend our method to real image editing with GAN inversions methods, it can serve as a flexible and practical editing tool for users.

2. Related work

As latent space exploration methods are discussed above, we focus on unconditional and conditional GANs in here.

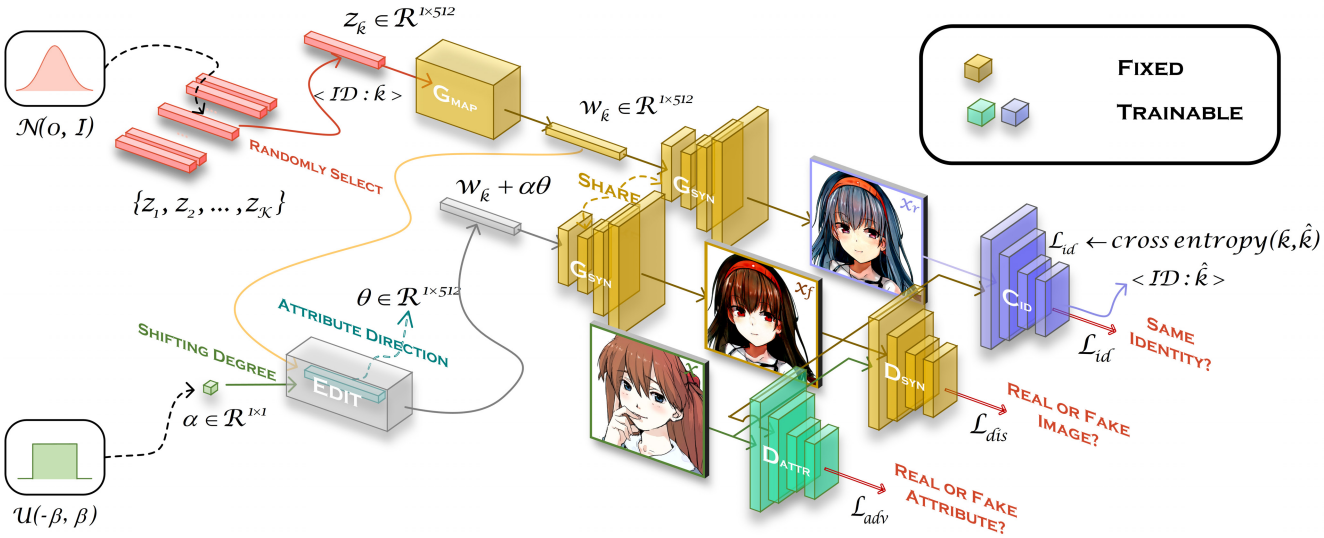


Figure 3: Overview of the proposed adversarial method AdvStyle. Given a pre-trained GAN model, our goal is to discover the shifting direction θ in the latent space, so that it can transform the original generated image x_r to x_f to contain user-desired attributes. Adversarial loss, distribution loss, and identity loss are proposed to preserve the features of the attribute, image quality, and identity, respectively.

2.1. Generative Adversarial Networks

Generative Adversarial Network [11] is composed of two parts. One is the generator which maps the simple latent space distribution to arbitrarily complex data distribution of a dataset, the other is the discriminator that is to distinguish the real distribution from the generated fake data. Various works have been made to improve the performance of GANs from different aspects, *e.g.*, by improving the discriminator [9], or by carefully designing the generator network structure [14, 35] or the loss function [5, 18]. The state-of-the-art models like StyleGAN [20] and BigGAN [6] can produce high-fidelity and high-resolution images. Among them, one important feature of StyleGAN is that it maps the input vector $z \in \mathcal{Z}$ to the intermediate latent vector $w \in \mathcal{W}$, which can "unwarp" $\mathcal{W}$ and the factors of variation become much more linear in the intermediate latent space. This mapping not only helps the generator to produce realistic images but also for better analyzing the property of the linear subspace. More importantly, it provides the possibility to further conduct semantic editing in the latent space. In this paper, we focus on the latent space $\mathcal{W}$ but we also show the comparison in the latent space $\mathcal{Z}$.

2.2. Semantic Editing with Conditional GANs

Unconditional GANs can only generate images randomly. To control the generation of GANs, it is required to carefully design the loss function, network structure, or introduced additional prior knowledge [7, 8, 13, 34]. For example, Lu *et al*. [25] output the high-resolution image for the low-resolution input that satisfies the given semantic attributes. Most models can only manipulate several specific attributes once the models are trained. Beyond that, the generated image quality is still far behind the unconditional GANs like StyleGAN [20] and BigGAN [6]. Different from the above conditional methods, exploring the attributes directions in the latent space is more straightforward for semantic editing, and different attributes can be manipulated without re-training the network. Besides, the learned directions can be applied to real images using GAN inversion method [2, 12, 31].

3. Approach

3.1. Problem Formulation

Fig. 3 illustrates the overall framework of the proposed AdvStyle. We use the pre-trained model of StyleGAN [20], which is composed of the mapping network G_{map} and the synthesis network G_{syn}. Given a random generated latent code $z \in \mathcal{Z}$ from the standard normal distribution, the mapping network G_{map} maps the latent code z to the intermediate latent code $w \in \mathcal{W}$. The synthesis network G_{syn} employs the latent code w and output a high-resolution image x_r. Our goal is to find the direction θ that corresponds to the target attribute in the latent space of a pre-trained GAN, such that a newly edited image x_f can be obtained by

$$\begin{aligned} w &= G_{map}(z), \\ x_f &= G_{syn}(w + \alpha\theta), \end{aligned} \tag{1}$$

where $\alpha \in \mathbb{U}[-\beta, \beta]$ is a metaphorical knob to control the changing degree of the target attribute.

The direction is learned using the attribute data $S_{data}(x) = \{x_i | i = 1, ..., N\} \in R$, where N is the number

of target attribute images in the training set. Note that no negative sample is needed in the training set. To train the target attribute direction, three components play an important role. First, an attribute assessor D_{attr} which is trained to distinguish the generated images from the target attribute dataset. An identity classifier C_{id} is used to match identity information of the generated images using the original latent code and the newly edited one. Lastly, the discriminator D_{syn} of the pre-trained model is involved to determine whether the input image conforms to the training dataset distribution.

In our method only the attribute assessor D_{attr} and identity classifier C_{id} are trainable , all the other components remain fixed (yellow blocks in Fig. 3). The most critical component of our method is the attribute assessor D_{attr} which is trained for driving the direction θ to reach its attribute manipulation purpose. It is done by adversarially enhancing the quality of transformed images concerning the target attribute distribution and optimizing the attribute direction θ. Our objective function is to solve a min-max problem:

$$\mathcal{L}_\theta = \arg\min_{\theta, C_{id}} \max_{D_{attr}} \mathcal{L}(\theta, C_{id}, D_{attr}). \tag{2}$$

3.2. Architecture

3.2.1 Attribute Assessor D_{attr}

The attribute assessor D_{attr} is designed to discriminate the target attribute from the transformed images. The attribute assessor D_{attr} contains three inputs: real target attribute image x, transformed image x_f, and shifting degree α. x_f is transformed from the original generated image x_r, by following the attribute direction $\alpha\theta$.

3.2.2 Identity Classifier C_{id}

Users typically desire to edit images on specific semantics without changing the original identity. To retain identity of the transformed image, we design the identity classifier C_{id} to learn the identity features of the original generated image x_r and the transformed image x_f. We consider the original generated image x_r as source identity image. For each training step, we first initialize K latent vectors, each latent vector $z \in \mathbb{N}(0, I)$. For each iteration, the k-th latent vector z is randomly selected to generate the images x_{r_k} and x_{f_k}, where $x_{r_k} = G_{syn}(G_{map}(z_k)), x_{f_k} = G_{syn}(G_{map}(z_k) + \alpha\theta))$. We formulate C_{id} as a 1-of-K classification problem, with the purpose of classifying both x_{r_k} and x_{f_k} to the k-th class, by using the cross-entropy loss:

$$\begin{aligned} \phi(x, k) &= \sum_j -\{y_x\}_j log(\{C_{id}(x)\}_j), \\ \arg\min_{\Theta_{C_{id}}} \mathcal{L}_{C_{id}} &= \phi(x_{r_k}, k) + \lambda\phi(x_{f_k}, k), \end{aligned} \tag{3}$$

where λ is a loss weight coefficient to balance the contribution of different generated images, $\{y_x\}_j$ is 1 if the predicted category of sample x is k and 0 otherwise, $\{C_{id}(x)\}_j$ is the predicted probability of C_{id} that for observation sample x belongs to class j. We train the identity classifier C_{id} together with the training of interpretable direction θ, and it serves as a feature generator for measuring the identity similarity between two images.

3.3. Training Objectives for Direction Discovering

The loss function $\mathcal{L}(\theta, C_{id}, D_{attr})$ in Eq. (2) consists of three parts: (1) the adversarial loss $\mathcal{L}_{adv}$, which pushes the direction to achieve the target manifold transformation, (2) the identity loss $\mathcal{L}_{id}$ that preserves the face identity, and (3) the distribution loss $\mathcal{L}_{dis}$, which maintains the generated image quality. We use a weighted additive form for the loss function:

$$\mathcal{L}(\theta, C_{id}, D_{attr}) = \mathcal{L}_{adv} + \lambda_1\mathcal{L}_{id} + \lambda_2\mathcal{L}_{dis}, \tag{4}$$

where λ_1, λ_2 are different weights to balance three losses. In all our experiments, we empirically set $\lambda_1 = 100$, $\lambda_2 = 1$, and found this setting balances the target attribute characteristic, identity information, and image quality well.

3.3.1 Adversarial Loss

The adversarial loss is applied to both the target direction θ and attribute assessor D_{attr}, as it is a min-max game that drives the direction to target attribution, which value indicates the degree of the given image belonging to the target attribute. Specifically, we use the Relativistic Average HingeGAN (RaHingeGAN) loss function [18] to calculate the adversarial loss. Different from the standard discriminator which simply judges the possibility of whether the given image is a real image, the Relativistic average Discriminator (RaD) estimates the probability that the given data is more realistic than opposite type samples on average. We can formulate RaD as $D(x) = f(D_{attr}(x) - \mathbb{E}(D_{attr}(x_{opp})))$, where $\mathbb{E}$ is the average for all opposite type images x_{opp} in the mini-batch. We found that the RaHingeGAN loss helps to learn better attribute representations. The concrete adversarial loss is defined as follow:

$$\begin{aligned} &\mathcal{L}_{adv}(\theta, D_{attr}) = \\ &\quad \mathbb{E}_{x\sim R}[f_1((D_{attr}(x) + \alpha) - \mathbb{E}_{x_f\sim Q}(D_{attr}(x_f)))] + \\ &\quad \mathbb{E}_{x_f\sim Q}[f_2(D_{attr}(x_f) - \mathbb{E}_{x\sim R}(D_{attr}(x) + \alpha))], \end{aligned} \tag{5}$$

where f_1, f_2 are scalar-to-scalar functions, and x, x_f belongs to the target attribute real image distribution R and target attribute generated image distribution Q, respectively.

3.3.2 Identity Loss

Editing on the original faces is an essential requirement for editing tools, either for real faces or anime ones. Our goal

is that when users manipulate the attributes, the identity information of faces can be preserved to the greatest extent. We adopt high-level features of the identity classifier C_{id} to calculate the identity loss

$$\mathcal{L}_{id} = d^{cos}(f_{id}^{r}, f_{id}^{f}), \tag{6}$$

where $d^{cos}(.,.)$ is the cosine distance function, and f_{id}^{r},f_{id}^{f} are the features of the original generated images x_r and the transformed images x_f, produced by the identity classifier C_{id}.

3.3.3 Distribution Loss

To prevent the learned directions go too far from the original image distributions, we also introduce the discriminator D_{syn} to retain the quality of the transformed image. The discriminator D_{syn} is pre-trained and part of the standard GAN. Similar to the adversarial loss, we also choose the RaHingeGAN to calculate the distribution loss. By using RaHingeGAN, the discriminator D_{syn} is to estimate the possibility that the given image is more realistic than the opposite type image. The distribution loss function is defined as follow:

$$\begin{aligned}\mathcal{L}_{dis} =& \mathbb{E}_{x\sim R}[g_1(D_{syn}(x) - \mathbb{E}_{x_f\sim Q}(D_{syn}(x_f)))]+ \\ & \mathbb{E}_{x_f\sim Q}[g_2(D_{syn}(x_f) - \mathbb{E}_{x\sim R}(D_{syn}(x)))],\end{aligned} \tag{7}$$

where $g_1(x) = ReLU(1 + x)$, $g_2 = ReLU(1 - x)$.

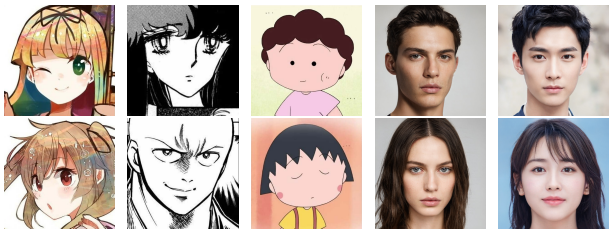
3.4. Implementation Details

3.4.1 Training Details

We use the Adam solver [22] to jointly optimize the direction θ, identity classifier C_{id}, and attribute assessor D_{attr} with batch size of 1. The learning rate of the direction θ is set to 0.0005, and the identity classifier C_{id} and the attribute assessor D_{attr} are trained with a learning rate of 0.0001. We perform $3 \cdot 10^4$ steps to obtain the attribute direction θ. The attribute assessor architecture is the same as the discriminator D_{syn}, and we use the Resnet-18 [16] model for the identity classifier C_{id}.

3.4.2 Layer-wise Editing

StyleGAN [20] naturally provides coarse control via the intermediate latent space. For example, the low-resolution(4^2-8^2), middle-resolution(16^2-32^2), and high-resolution($64^2 - 1024^2$) features bring controllability on high-level structure, facial features, and fine styles, respectively. For better disentanglement, we apply layer-wise editing with our trained direction: $w_i = w + m_i \cdot \alpha\theta$, where $m_i = 1$ when i layer is editable and $m_i = 0$ otherwise.



Itomugi-Kun [4] | Comic [3, 26] | Maruko | Super Model [1] | Chinese Celebrity [1]

Figure 4: Examples of our collected 3 anime styles and 2 human styles.

4. Experiments

We implement the proposed AdvStyle in Pytorch [27] on a PC with an Nvidia GeForce RTX 2070 GPU. AdvStyle is able to edit generated images in real-time, which takes 22ms for producing a 1024×1024 image. Finding the direction is also fast, it takes about 4 hours to locate the desired attribute. Next, we introduce how we collect the training data with a single label, and thoroughly evaluate our approach on the collected datasets in terms of quantitative and qualitative results, by mainly comparing with the state-of-the-art supervised method InterFaceGAN [30].

4.1. Attribute Datasets

We evaluate the proposed method on 9 anime attributes and 7 human attributes. For animate attributes, we collect 6 character properties (*open mouth, blunt bangs, hair length, black hair, blonde hair, pink hair*) and 1 style (*Itomugi-Kun*) from the Danbooru2018 dataset [4]. The other two styles are, *comic style* from Manga109 [3, 26], *Maruko style* from a manually collected dataset of Japanese anime Chibi Maruko-chan. All the images are resized to 512×512 for training.

For human face editing, five binary attributes (*pose, old, female, smile, eyeglasses*) are trained using the CelebA datasets [24] and two style attributes (*supermodel style* and *Chinese celebrity style*) are trained on datasets collected from [1] with 1024×1024.

We show five style attributes in Fig 4 for a better understanding of the target style. Editing results for all attributes can be found in supplementary materials.

4.2. Binary Attribute Manipulation

We first evaluate the proposed method on traditional binary attributes. Fig. 5 shows the manipulation results. For anime attribute editing in Fig. 5 (a) to (b), we can see that InterFaceGAN is severely entangled with *hair color*, for both two attributes. This is because the anime datasets like Danbooru [4] do not provide a "*closed mouth*" label, and thus InterFaceGAN can be only trained on non-open mouth data (*i.e.*, all images without the open mouth label) which results

(a) - ⟵ Open Mouth ⟶ + (b) - ⟵ Short Hair ⟶ + (c) - ⟵ Old ⟶ + (d) - ⟵ Female ⟶ +

Figure 5: Manipulation on binary attributes. Images are generated by moving in positive or negative directions.

(a) Origin (b) A vs. non-A in $\mathcal{W}$ (c) A vs. B in $\mathcal{W}$ (d) A vs. B in $\mathcal{Z}$ (e) Ours

Figure 6: Exploration of non-binary settings of InterFaceGAN [30] on *blonde hair* and *supermodel* attributes. A vs. non-A means InterFaceGAN is trained with A attribute samples as positive while all the non-A samples as negative. In another setting A vs. B, in which B represents a manually selected attribute, *i.e.*, *blonde hair* and *Chinese celebrity* in this example.

in entangled directions. Although the *hair length* attribute of InterfaceGAN is trained on *long hair* vs. *short hair*, this attribute is indeed continuous and the boundary is not distinct. As a consequence, the resulted direction is entangled with *hair color* and *open mouth* attributes. On the contrary, our editing results are highly disentangled from all the other attributes and maintain the original identity well.

A similar problem of InterFaceGAN can be found in human face editing. As shown in Fig. 5(c) and (d), the *old* direction of InterfaceGAN is entangled with *eyeglasses*, and the *female* direction is entangled with beard, smile, and hairstyle. Our proposed AdvStyle not only focuses on the target attribute but also the identity information, resulting in changes in local target attribute while preserving identity.

4.3. Non-binary Attribute Manipulation

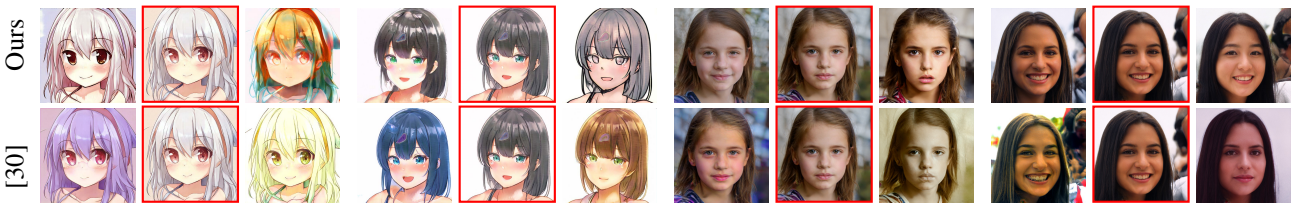
Binary-based method like InterFaceGAN [30] relies on highly distinctive positive/negative attribute pair, such as the binary attributes of *male* vs. *female*, without ambiguous association with other properties. However, many attributes cannot find clear opposites, like *supermodel* style attribute. There are two ways to collect pseudo-binary attribute pairs for InterFaceGAN: 1) we can consider attribute A as positive while leaving all *non-A* images as negative (Fig. 6(b)); 2) given an attribute A, we select another related attribute B as negative (Fig. 6(c) and (d)). We examine these solutions in Fig. 6.

When training InterFaceGAN with A vs. *Non-A*, we can see in Fig. 6(b) that the network is confused by the ambiguous negative samples, *e.g.*, mixes blonde hair with blonde face color or wrongly extracted supermodel features. On the other hand, InterFaceGAN performs relatively better (Fig. 6(c)) if we manually select a more distinct attribute pair, in this case, *blonde hair* vs. *black hair* and *supermodel* vs. *Chinese celebrity*. However, it still cannot disentangle the *blonde hair* attribute from the *open mouth* and cannot capture distinctive *supermodel* attribute. In contrast, the proposed method in Fig. 6(e) shows the best disentanglement. In addition, we examine the performance of InterFaceGAN in the latent space $\mathcal{Z}$ (Fig. 6(d)). Unsurprisingly, the latent space $\mathcal{Z}$ is more entangled as stated in StyleGAN [20], resulting in large variations of the identity. As the performance of InterFaceGAN trained with A vs. B varies largely depending on B attribute, we use A vs. *Non-A* to train InterFaceGAN in all the comparisons.

We further compare our AdvStyle with InterFaceGAN on more different style attributes. As shown in Fig. 7, InterFaceGAN fails to capture the unique style features (see Fig. 4 for comparison). Particularly, for anime style editing (a) and (b), InterFaceGAN emphasizes on global color distributions only. While in (c) and (d), enforcing style attributes to become binary ones making the training distracts from unique facial characteristics, *e.g.*, *supermodel* attribute direction of InterFaceGAN can only learn cool expressions but yellowish colors. On the contrary, our method captures representative features for both anime and human face styles. For example, the edited result of *Itomugi-Kun* style shows vivid color shading, and the results of both *supermodel* and *Chinese celebrity* directions exhibit typical facial characteristics of two different types of faces, while they are disentangled from other attributes like hairstyle or smile.

4.4. Involuntarily Learned Negative Directions

An interesting feature of our AdvStyle is that we do not manually assign negative labels for training. This is important for discovering the correct positive direction. For non-binary attributes, the opposites of them are usually ambiguous, and thus a wrongly assigned negative label may



(a) - ⟵ Itomugi-Kun ⟶ + (b) - ⟵ Comic ⟶ + (c) - ⟵ Supermodel ⟶ + (d) - ⟵ Chinese Celebrity ⟶ +

Figure 7: Manipulation on non-binary attributes. Images are generated by moving in positive or negative directions.

prevent the network from finding the correct direction. In here, we study the latent space by exploring the involuntarily learned negative directions, and reveal how the latent space is organized.

Negative Directions of Binary Attributes. As shown in Fig. 5, the involuntarily learned negative binary attributes are consistent with our common sense. Specifically, the learned negative directions of *open mouth*, *short hair*, *old*, *female*, correctly correspond to the reverse attributes *closed mouth*, *long hair*, *young*, *male*, without any entanglement.

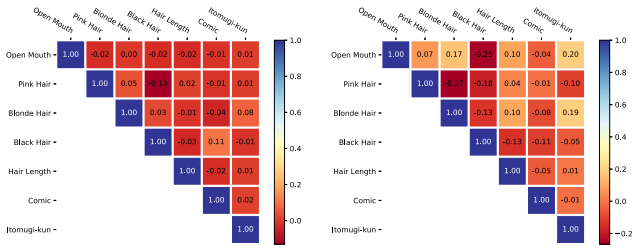
Negative Directions of Non-binary Attributes. For non-binary attributes, it would be interesting to figure out the semantics correspond to their reverse directions in the latent space. As shown in Fig. 7, the reverse directions of styles are surprisingly interpretable. The positive direction of the *Itomugi-Kun* style is to strengthen the color shading of the image, while its negative direction doing the opposite to produce plain shading anime. For human style, the negative direction of *Chinese celebrity* yields the opposite characteristic of Chinese faces (*e.g.*, deep eye socket). All these directions are unexpected but meaningful, demonstrating the effectiveness of the proposed learning strategy.

4.5. Multi-attribute Manipulation

4.5.1 Attributes Correlation

It is of great importance to allow users to control multiple attributes at the same time. However, this is impossible if different attributes are entangled with each other. To evaluate whether the learned directions are disentangled, we first compute the correlation matrix between different attributes. We use cosine similarity to measure the correlation: $\cos(\theta_1, \theta_2) = \frac{\theta_1 \cdot \theta_2}{|\theta_1||\theta_2|}$, where θ_1 and θ_2 are two different attribute direction vectors. We also compute the correlation matrix of the binary attribute-based method InterFaceGAN [30]. For a fair comparison, we mainly evaluate identity attributes as they are easier to perform binary classification (we add two styles for reference). We construct the binary datasets of InterFaceGAN by dividing the target attribute samples as positive and all the others as negative.

Fig. 8 shows the correlation matrices of the two methods. As can be seen, most directions of our AdvStyle are highly disentangled, *i.e.*, they are orthogonal to each other.



(a) Our Correlation Matrix (b) InterFaceGAN Correlation Matrix

Figure 8: Correlation matrices between different attributes. Unlike the binary attribute-based method InterFaceGAN [30] produces entangled directions, the directions we found are more uncorrelated.

Some interesting facts can be found in our correlation matrix. For example, *black hair* correlates to *comic* style, as both of them tend to generate black edges. On the contrary, the learned directions of InterFaceGAN are highly entangled. More importantly, their correlations are not interpretable, indicating that their learning strategy cannot deal with mixed and ambiguous attribute data.

4.5.2 Manipulating Multi-attribute

Because of the highly disentangled attributes, we can simultaneously manipulate multiple attributes while still retaining the identity features. We modify multiple attributes by a simple arithmetic operation: $\theta_{mul} = \sum_{i=1}^{N} \alpha_i \theta_i, x = G_{syn}(G_{map}(z) + \theta_{mul}), \theta_{mul} \in \mathbb{R}^{1\times 512}$, where N is the number of attribute directions applies to a vector $w = G_{map}(z)$ at the same time and α_i is the shifting degree for direction θ_i. As shown in Fig. 9(a), the proposed method can manipulate up to 4 attributes at the same time, while still preserving identity information.

We further compare with an unsupervised method [15] in Fig. 9(b) for human face editing. Their method can unsupervisedly find out 3 human attributes, therefore we only compare to these three attributes in Fig. 9(b). While the two competitors can produce well-disentangled results in single attribute editing, the combined results contain unexpected facial variations. Our results, on the contrary, are disentangled from all the other factors. Note that multi-attribute manipulation is achieved using separately learned direc-

(a) Multi-attribute anime editing

(b) Multi-attribute face editing

Figure 9: Results of multi-attribute manipulation.

tions, without additional constraint like conditional manipulation [30].

4.6. Ablation Study

In our method, only the adversarial loss takes the responsibility for finding the correct direction. Fig. 10(b) shows that in the model with only the adversarial loss, we can still locate the corresponding attribute. However, without the constraints of other losses, the network only focuses on fooling the attribute discriminator, driving the manipulation on the target attribute too far from the original. The identity loss, as shown in Fig. 10(c), preserves the character identity well after editing. Without the distribution loss, we observe that the generated characters throw away some details like hair shading or face effects compared with the original synthesis. Fig. 10(d) adds these details back and we believe they are important to conform to the original data distributions. Finally, our all losses model produces images that faithful to target attributes while maintaining the identity and image quality of the original synthesis. Our layer-wise editing strategy also contributes to the final performance, as it can produce semantic-specific latent code in each layer. We can see that with layer-wise editing (left part of Fig. 10(f)) maintains the original skin color better than without it (right part of Fig. 10(f)).

4.7. Real Image Manipulation

To allow editing on the real image, we adopt a GAN inversion method [2] to obtain the latent code of a real image. Although the GAN inversion method cannot perfectly recover the original input, the obtained latent code can be used by the proposed method for editing. Similar observations with synthesis image editing can be found that InterfaceGAN [30] is entangled with the *eyeglasses* attribute. The first column of Fig. 1 shows two additional results on real image style editing and multi-attribute manipulation.

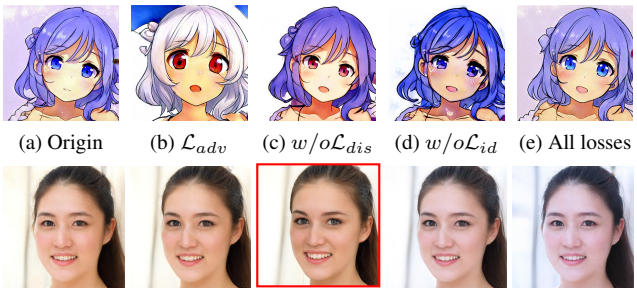


Figure 10: (a) to (e) are the ablation study of our loss functions with the *open mouth* attribute. (f) evaluates our layer-wise editing strategy on *Chinese celebrity* style.

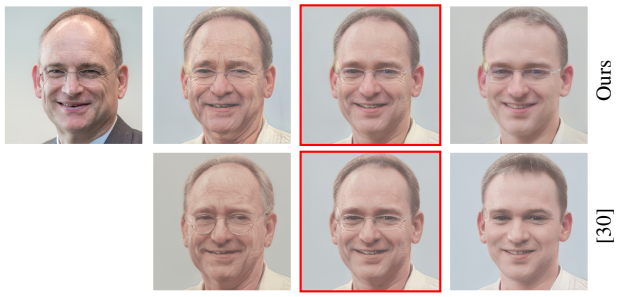


Figure 11: Real image manipulation of *age* attribute. Note that our direction is only trained with *old* samples.

5. Conclusion

We propose an adversarial method, AdvStyle, to interpret the latent space of GANs for image attribute manipulation. Our method can get rid of the dependence on binary data, to explore directions of non-binary attributes. We show extensive results on AdvStyle, demonstrating its effectiveness on both single and multi-attribute manipulation. The proposed method can also discover unexpected negative directions involuntarily.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61972162), and CCF-Tencent Open Research fund.

References

[1] http://www.seeprettyface.com/.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, pages 4432–4441, 2019.

[3] Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, and Hikaru Ikuta. Building a manga dataset "manga109" with annotations for multimedia applications. *IEEE MultiMedia*, 27(2):8–18, 2020.

[4] Anonymous, Danbooru community, and Gwern Branwen. Danbooru2019: A large-scale crowdsourced and tagged anime illustration dataset. https://www.gwern.net/Danbooru2019, January 2020.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, volume abs/1809.11096, 2019.

[7] Yubei Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I-Ming Pao, and Jiaya Jia. Semantic component decomposition for face attribute manipulation. In *CVPR*, pages 9851–9859, 2019.

[8] Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Sscgan: Facial attribute editing via style skip connections. In *ECCV*, 2020.

[9] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. In *ICLR*, 2017.

[10] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, pages 5743–5752, 2019.

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

[12] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020.

[13] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *CVPR*, pages 3431–3440, 2019.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5767–5777, 2017.

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

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

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

[18] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. In *ICLR*, 2019.

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

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4396–4405, 2018.

[21] Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *ICCV*, pages 2390–2398, 2015.

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

[23] Shu Kong, Xiaohui Shen, Zhe L. Lin, Radomír Mech, and Charless C. Fowlkes. Photo aesthetics ranking network with attributes and content adaptation. In *ECCV*, 2016.

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

[25] Yongyi Lu, Yu-Wing Tai, and Chi-Keung Tang. Attribute-guided face generation using conditional cyclegan. In *ECCV*, 2018.

[26] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[28] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2016.

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

[31] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, Christian Theobalt, et al. Pie: Portrait image embedding for semantic control. In *SIGGRAPH Asia*, 2020.

[32] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020.

[33] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, 2020.

[34] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *ECCV*, 2018.

[35] Yang Yu, Zhiqiang Gong, Ping Zhong, and Jiaxin Shan. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In *International Conference on Image and Graphics*, pages 97–108, 2017.