Attribute Consistent Generative Adversarial Nets

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent research suggests that Generative Adversarial Nets (GANs) can produce realistic images with different attributes. To precisely control each attribute of generated images, current methods search the semantic directions in the latent space of a pretrained model. Since the generative models are trained without the supervision of attribute labels, these methods yield semantic entanglement. To tackle these concerns, we propose Attribute Consistent Generative Adversarial Nets, termed ACGAN, to train an attribute-based generative model while projecting the controllable semantic directions to an orthogonal space. Specifically, we first design an attribute quantification method to obtain the continuous labels. An attribute regressor trained by them is applied to constrain the attribute consistency between the input attribute code and the generated results. We decompose the latent code into the content code and the attribute code. The attribute code lies in an orthogonal space, so as to theoretically ensure disentanglement between attributes. Experimental results demonstrate that the proposed approach is superior in generating realistic images with multiple controllable attributes.

1 Introduction

Generative Adversarial Nets (GANs) [6] models the data distribution in a zero-sum game: training the generator and the discriminator in an adversarial manner. When the model converges, the generator maps the latent codes sampled from the latent space to the image space. The most recent deep generative models achieve great progress in producing photo-realistic images that are even indistinguishable from real-world photos. The GAN model itself, however, is unable to edit the semantic attributes of generated images, *e.g.*, to manipulate the age of the portrait or the weather of the scene pictures.

Image semantic editing can be utilized in amount of applications, ranging from image enhancement to animation design. Current methods can be grouped into two categories: *image-space editing* methods and *latent-space editing* methods. Image-space editing methods[29, 3, 4, 8] transform one image from the source domain to the target domain directly. These methods have to learn a great deal of redundant models because the transformation between every two domains requires one model. This is disastrous for multi-attribute editing tasks. Latent-space editing methods focus on searching the semantic-related directions in the latent space. Since the dimension of a latent code is much smaller than an image, these methods can simultaneously discover interpretable directions for multiple attributes. These methods can be further divided into *unsupervised methods* [26, 7], *self-supervised methods* [9, 22] and *supervised methods* [24, 30]. Among these methods, the directions discovered by unsupervised methods are unpredictable, and the generated results usually leads to identity confusion. The directions found by self-supervised methods are quite limited, *e.g.*, scale and rotation. Current supervised methods search the interpretable direction in the latent space with the binary attributes supervision. This is achieved by analysing classification hyperplanes [24] or searching directions with a pretrained attribute regressor [30]. Supervised methods cannot guarantee to completely disentangle

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

attribute directions. Changing one attribute is often accompanied by changes in other ones. These methods improves the image editing ability to a certain extent, but also suffers from entangle attributes, and it is hard to preserve the identity when editing multiple attributes.

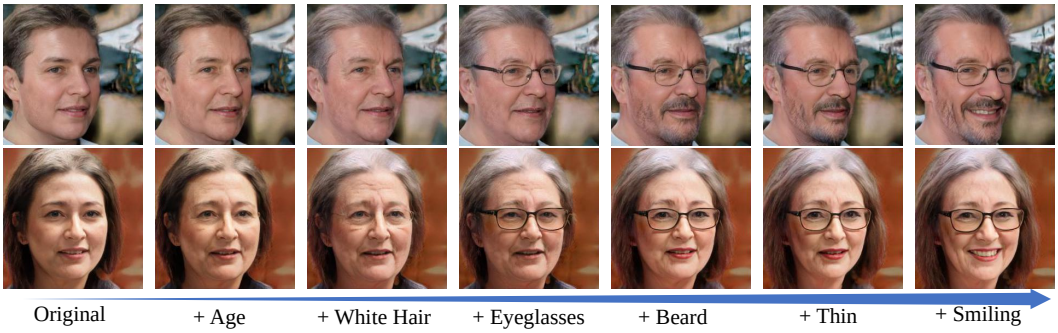

Figure 1: Accumulative attribute editing. We gradually enhance a certain attribute(age, white hair, etc.) from left to right in the portraits. Benefiting from disentanglement between attributes, the editing results are precise and smooth.

To the best of our knowledge, all the latent-space editing methods search the semantic directions on a *pretrained generative model* [23], which leads to the critical drawbacks that the generative model is not constrained by attribute labels during training phase, resulting in unavoidable semantic entanglement in the latent space. Searching semantic directions in the latent space is insufficient to address this problem.

To tackle this problem fundamentally, we propose the attribute consistent generative adversarial nets (ACGAN), which consists of a generator, a discriminator and a pretrained regressor. The latent space is decomposed into the content space and the attribute space. The input latent values consist of content values and attribute values that are sampled from the content space (subject to a Gaussian distribution) and the attribute space (subject to an uniform distribution) respectively. Since the attributes discovered by latent-space editing methods are entangled, we propose to project the attribute space to an orthogonal space, so as to theoretically ensure disentanglement between attributes.

Our methodology is expected to endow the following merits: ◇ Unified paradigm - comprehensively optimizing the attribute controllable directions in both the latent space and the generative model. Improving the attributes sensitivity of the generative model by adding the controllable parameters. ◇ Disentangle attribute space - mapping attributes into a predefined orthogonal space. The element of each basis vector indicates the attribute strength. ◇ Continuous attribute consistency - the predicted attribute values of reconstructed images is consistent with the input attribute values. In the training phase, the generative model is trained as well as mapping the semantic direction in the latent-space to an orthonormal space. The generator is supervised by the continuous attribute-consistency loss, computed by the input attribute values and the predicted attribute values of regressor. However, to the best of our knowledge, currently only one dataset [13] contains continuous attribute labels, and most image attribute datasets [18, 27, 28] only contain binary labels. To extend our approach to all the attribute datasets, we further propose an attribute quantification strategy to quantify binary attribute labels to continuous values.

Theoretically, our method can simultaneously control all the labeled attributes in the datasets. As shown in Fig. 1, we sequentially add six attributes to two face images and generate reasonable editing results with identity preservation. Extensive experimental results on face images and natural scene images demonstrate that the semantic path planned by our paradigm is smoother and more disentangle than other methods.

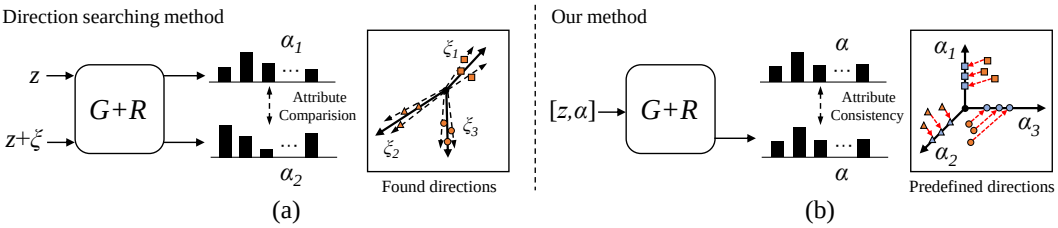

Figure 2: The framework of directions searching methods [26, 30] (left) and our method (right). In (a), the directions found by [26, 30] is accomplished through comparing the attribute difference between the original image and the edited image. The found directions are entangled. Instead, as shown in (b), our method trains the generator with attribute-consistency loss between the input attribute values and the regressed attribute of the generated image. Our predefined directions are orthonormal while directions of directions searching methods are not.

2 Related Work

Generative adversarial nets [6] based generative models [10, 11, 17] have greatly improved the photo-realistic of generated images. GANs are constructed by a generator and a discriminator, which are trained alternately in an adversarial manner. The generator maps the latent code to the generate images. The discriminator is a binary-classifier used to distinguish the generated images from the real ones. A more complete understanding is that the latent code and the generative model jointly determine the appearance of generated images. Recently, a lightweight GAN[17] model has been released, benefiting from its efficient network structure and data augmentation strategy, it only takes a few hours to complete training on a single GPU.

Image space editing aims to directly transfer input images to expected style (style transfer) or image domain (image to image translation). With the development of deep learning, neural style transfer [5, 16, 19] has become the mainstream method for style transfer. While style transfer focuses on different art styles, image-to-image translation [3, 4, 8, 29] solves a more general image domain translation problem. The drawback of these methods is that editing must be a translation from one style/domain to another style/domain and the degree of translation is uncontrollable at all.

Latent space editing in GANs. Current latent space editing methods focus on searching semantic directions in the latent space of a pretrained GAN model[26, 7, 30, 24, 9, 22] in a *self-supervised*, *unsupervised* or *supervised* manner. *Self-supervised methods* [9, 22] searched interpretable directions by augmenting data with simple geometric transformations, e.g., rotation and scaling, which limits the capacity of these methods to searching complex attributes. *Unsupervised methods* searched the semantic directions by learning a set of distinguished directions [26] or identifying the important latent directions based on principal components analysis (PCA) [7]. These methods might find meaningful directions, yet they are unpredictable and require human interpretation, resulting in hard specifying the desired attribute direction. *Supervised methods* aimed at capturing the transformation of each attribute by seeking the semantic directions in the latent space under the supervision of attribute labels.

3 Method

3.1 Motivation

Current latent space editing methods suffer from attribute entanglement because direction searching is conducted on a pretrained generative model that is not constrained by attribute labels during the training phase. Since these methods cannot guarantee the discovered directions independent, which inevitably leads to semantic entanglement in the latent space.

This problem provides the main intuition behind our method. Namely, we aim to learn the orthogonal attribute directions that are easy to distinguish from each other. We achieve this via projecting the attribute directions into an orthogonal space, and decomposing latent code into content code and orthonormal attribute code. Specifically, we propose an attribute consistent GAN (ACGAN) with

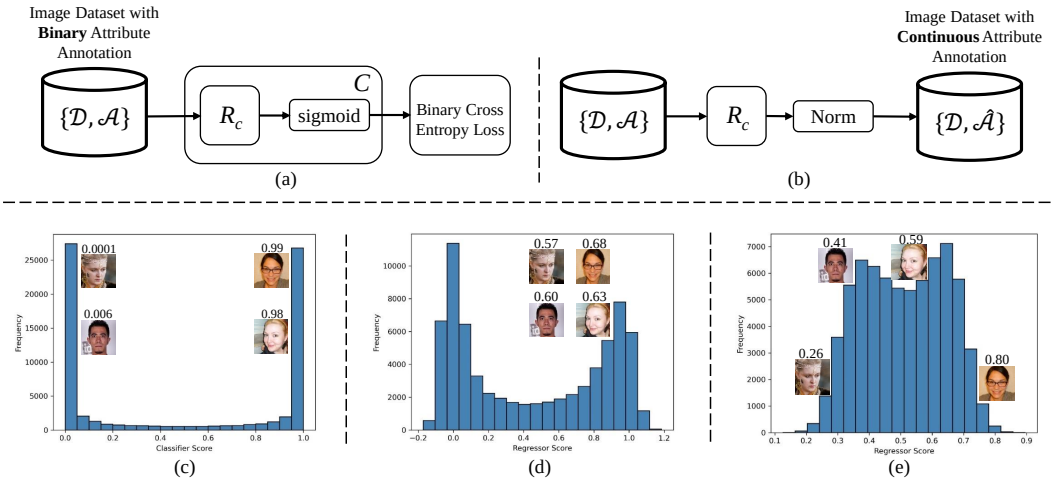

Figure 3: Continuous attributes generation. We first train an attribute classifier C that consists of a continuous attribute regressor R_c and a sigmoid layer as shown in (a) then we normalize outputs of R_c as continuous labels as shown in (b). We collect attribute prediction values of classifier C trained by $\mathcal{A}$, regressor R trained by $\mathcal{A}$ and regressor C trained by $\hat{\mathcal{A}}$ in (c), (d) and (e) respectively.

the attribute-consistency supervision. Different from [30], our attribute regressor R is trained by continuous label. Apart from the implementation details of R, the key difference between existing direction searching methods and our ACGAN is that the decomposition of latent space and training ACGAN from scratch. This solves the attribute entanglement problem fundamentally in latent space editing for the first time and makes editing multiple attributes simultaneously possible.

3.2 Continuous Label Generation

Suppose we have an image set $\mathcal{D} = \{I_k\}_{k=1}^n \subset \mathbb{R}^{h \times w \times 3}$ and the attribute set $\mathcal{A} = \{y_k\}_{k=1}^n \subset \mathbb{R}^d$, where each pair $\{I_k, y_k\}$ is the k-th image and the corresponding attribute label in dataset $\{\mathcal{D}, \mathcal{A}\}$ respectively, each $y_k = [y_{k,1}, y_{k,2}, \cdots, y_{k,N}]$, $y_{k,i}$ is 0 for negative attribute or 1 for positive attribute, N is the number of attribute categories. Our target is to generate continuous pseudo-labels $\{\hat{y}_k\}_{k=1}^n$ where $\{\hat{y}_k\}$ contains continuous scalar between 0 and 1.

Our basic idea is to train a classifier C that consist of a continuous attribute regressor R_c and a sigmoid layer on $\mathcal{D}$ supervised by $\mathcal{A}$ as shown in Fig.3(a) with binary cross entropy loss:

$$\mathcal{L}_{BCE} = \sum_{k=1}^{n} \sum_{i=1}^{d} [y_{k,i} \cdot \log x_{k,i} + (1 - y_{k,i}) \cdot \log (1 - x_{k,i})], \tag{1}$$

As illustrated in Fig. 3(b), we generate continuous attribute labels $\hat{y}_k = [\hat{y}_{k,1}, y_{k,2}, \cdots, \hat{y}_{k,N}]$ by normalizing $\{R_c(I_K)\}_{k=1}^n$:

$$\hat{y}_{k,i} = \frac{R_c(I_k)_i - \min\limits_{1 \le t \le n} R_c(I_t)_i}{\max\limits_{1 \le t \le n} R_c(I_t)_i - \min\limits_{1 \le t \le n} R_c(I_t)_i}. \tag{2}$$

Note that $R_c(\cdot)$ does not contain $sigmoid$ function. The reason we use the value, i.e., $R_c(I_K)$, before $sigmoid$ to compute continuous pseudo-labels by Eq. 2 is that $sigmoid$ is a soft version of step function:

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0, \end{cases} \tag{3}$$

where χ_A is the indicator function of A. The $sigmoid$ will map $R_c(I_K)$ to $x_{k,i}$ that is close to 0 or 1. Now we have continuous pseudo-label set $\hat{\mathcal{A}} = \{\hat{y}_k\}_{k=1}^n$. Our attribute regressor R is trained by image dataset with continuous attribute labels $\{\mathcal{D}, \hat{\mathcal{A}}\}$.

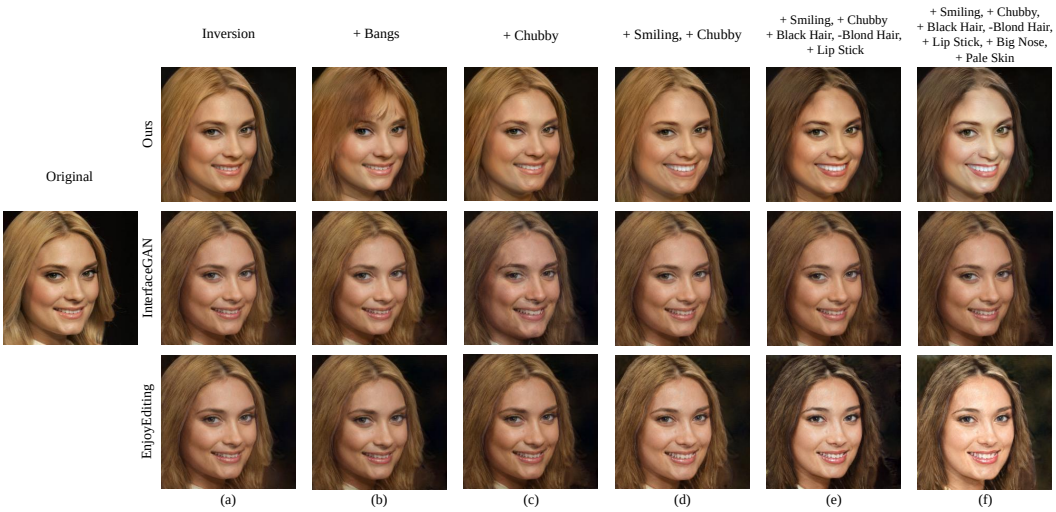

Figure 4: Single attribute editing (b, c) and multi attributes editing (d, e, f) evaluation. On the first row, results generated by our ACGAN are tremendously precise even for enhancing or reducing from 1 to 7 attributes including *smiling, chubby, black Hair, blond Hair, lip stick, big Nose* and *pale skin*.

3.3 Attribute Consistent GAN

Overview Our ACGAN can be implemented based on any unconditional GAN and is composed of 3 parts: a generator G, a discriminator D, and a pretrained attribute regressor R. Leaving the discriminator unchanged, the attribute code α and content code z are fed into the generator as the inputs. R is utilized to supervise the attributes of generated samples via attribute regression loss: the objective of the generator is no longer just to "deceive" the discriminator, but also to satisfy the attribute consistency, i.e., the attribute of the generated samples equal to the input attribute code. In short, in order to transform any unconditional GAN into ACGAN, only two changes need to be implemented: (1) decomposing the input values into content values and attribute values; (2) adding attribute regression loss.

Attribute-consistency loss The key point to control the attributes of the generated samples is that the attributes of the generated samples are constrained by the attribute regressor in the training phase. This is achieved through calculating l_2-norm between input attribute code α and predicted attribute $R(G(z|\alpha))$, i.e., $\|R(G(z|\alpha)) - \alpha\|_2$.

How to sample α in the training phase? We investigate two sampling strategies: 1) sampling from the pseudo-label set $\hat{\mathcal{A}}$, 2) sampling from a uniform distribution $\mathcal{U}(0, 1)$.

When attributes in $\mathcal{A}$ are semantically tangled, the combination of certain attribute values may be unreasonable. Taking CelebA dataset as an example, the two attributes *goatee* and *male* are highly tangled. This combination is unreasonable where *goatee* is 1 and *male* is 0. Considering that each pseudo label in $\hat{\mathcal{A}}$ corresponds to a real image in $\mathcal{D}$, the first strategy can guarantee α is reasonable. Attribute-consistency loss can be formulated as:

$$\mathcal{L}_{Reg}(G, R) = \mathbb{E}_{z\sim\mathcal{N}(0,1),\alpha\sim\hat{\mathcal{A}}}\|R(G(z|\alpha)) - \alpha\|_2 \tag{4}$$

As for the second strategy, we may sample unreasonable α from $\mathcal{U}(0, 1)$, but this sampling strategy has two indispensable advantages compared to the first one:

1) Combination of attribute values in $\hat{\mathcal{A}}$ is quite biased. Let us assume that each variable in α obeys normal distribution $\mathcal{N}(0, 1)$. If we directly sample α from $\hat{\mathcal{A}}$, the medium attribute value (close to 0.5) has the highest sampling probability, while the probability of sampling extreme attribute value (close to 0 or 1) will be very small. This actually contradicts our goal: when the input attribute changes linearly from 0 to 1, we wish ACGAN can linearly generate samples of the corresponding attribute value. In this process, we treat neutral attribute values and extreme attribute values equally, so sampling from $\mathcal{U}(0, 1)$ is a better strategy.

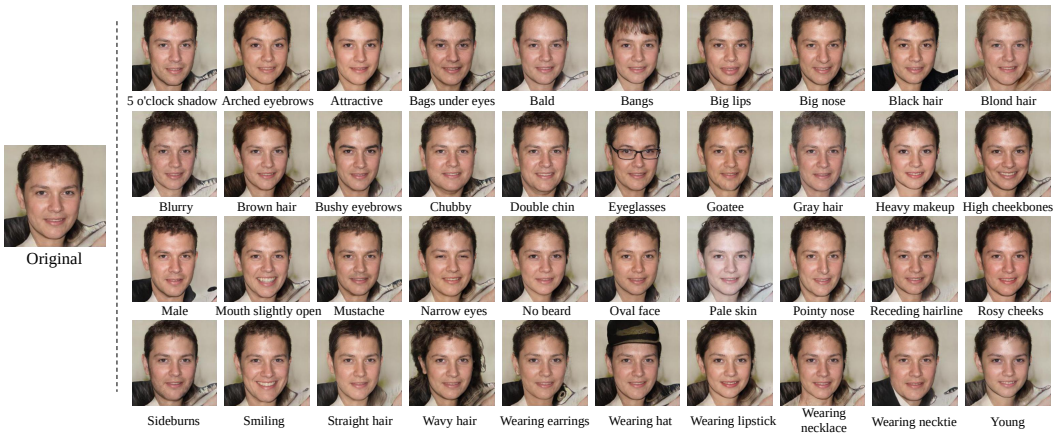

Figure 5: Edited samples of all 40 attribute in CelebA dataset. The original image is generated under all attribute values equal to 0.5. For each attribute editing, we simply increase that attribute value to 1 and keep content values and other attribute values unchanged.

2) The samples in $\hat{\mathcal{A}}$ may be limited. Note that α lies in a high-dimensional space $\mathbb{R}_N$. It is insufficient to sample n points in $\mathbb{R}_N$, but this is exactly what we do when sampling from $\hat{\mathcal{A}}$. If we adopt the second strategy, the number of sampled points depends on the number of iterations we train ACGAN, which is much larger than n.

Generally, sampling α from $\mathcal{U}(0,1)$ is a more universal strategy. Attribute regression loss can be formulated as:

$$\mathcal{L}_{Reg}(G,R) = \mathbb{E}_{z\sim\mathcal{N}(0,1),\alpha\sim\mathcal{U}(0,1)}\|R(G(z|\alpha)) - \alpha\|_2 \tag{5}$$

Overall network loss After introducing the attribute regression loss implemented by R, all we need to do is adding attribute regression loss to original GAN loss:

$$\min_G \max_D V(D,G,R) = \mathcal{L}_{GAN}(G,D) + \lambda\mathcal{L}_{Reg}(G,R), \tag{6}$$

where $\mathcal{L}_{GAN}$ is the original adversarial loss such as Wasserstein loss in WGAN[2] or least squares loss in LSGAN[20], λ is a hyper parameter balancing L_{Reg}.

4 Experiments

4.1 Implementation Details

We conduct experiments on face synthesis and natural scene synthesis. In this section, we will introduce base model and architecture of C and R, then describe implementation details in these two scenarios separately.

Base model. We implement the proposed ACGAN base on light-weight GAN [17] and StyleGAN2 [11]. The attribute code α is concatenated with z for light-weight GAN and w for StyleGAN2.

Architecture of C and R. We adopt ResNet-50 [25] as the backbone architecture of classifier C and regressor R. The last fully connected layer is replaced with a new one with the dimension of N corresponding to attribute categories. In Sec.3.2 we isolate $sigmoid$ from attribute classifier C, so C and R can share the same network architecture.

Face synthesis. The facial attribute knowledge (the parameters of C and R) is learned from CelebA dataset [18], which is a large-scale face attributes dataset consists of more than 200K celebrity images, each with 40 attribute annotations at 178×218 resolution. Both C and R are trained for 5 epochs with an Adam optimizer and a learning rate of 0.0001. Our ACGAN model is optimized on Flickr-Faces-HQ (FFHQ) [10] dataset, which consists of $70,000$ high-quality images at 1024×1024 resolution and contains considerable variation in terms of age, ethnicity and image background. FFHQ enables the model to generate high-fidelity images.

Figure 6: Comparing our ACGAN with image-to-image translation methods CycleGAN [29] RelGAN [21] and DRIT++ [15] for 5 attributes editing.

Natural scene synthesis. The natural attribute knowledge is learned from Transient Attribute Database [14] for regressor R. Since this dataset has continuous attribute values between 0 and 1, it is unnecessary to train an attribute classifier C. R follows the same training set as in face synthesis task for 300 epochs. Motivated by [30], we selected 40,726 pictures related to natural scenes from Place dataset [27] and SUN dataset [28] to expand the Transient Attribute dataset, and utilized this augmented natural scene dataset to optimize our ACGAN model.

Other implementation details. We set hyperparameter λ to denote the number of selected attributes for balancing attribute regression loss, i.e., $\lambda = 40$ for face synthesis and $\lambda = 5$ for natural scene synthesis. We use a single Titan RTX for all experiments.

4.2 Face Synthesis

Score comparison by classifier and regressor. We divide $[0, 1]$ into 20 equal intervals, and plot frequency histogram of attribute *smiling* score on FFHQ dataset. As shown in Fig.3(c), sigmoid will push output of attribute classifier C close to 0 or 1. Attribute regressor directly trained by $\mathcal{A}$ also generates most prediction values close to 0 or 1 as demonstrated in Fig. 3(d). On the contrary, our attribute regressor R trained by continuous labels $\hat{\mathcal{A}}$ produces continuous values from 0 to 1 as in Fig.3(e).

Qualitative comparison. We evaluated the attribute controllability on single and multiple attributes, and compare our method with two advanced supervised methods: InterFaceGAN [24] and EnjoyEdit-

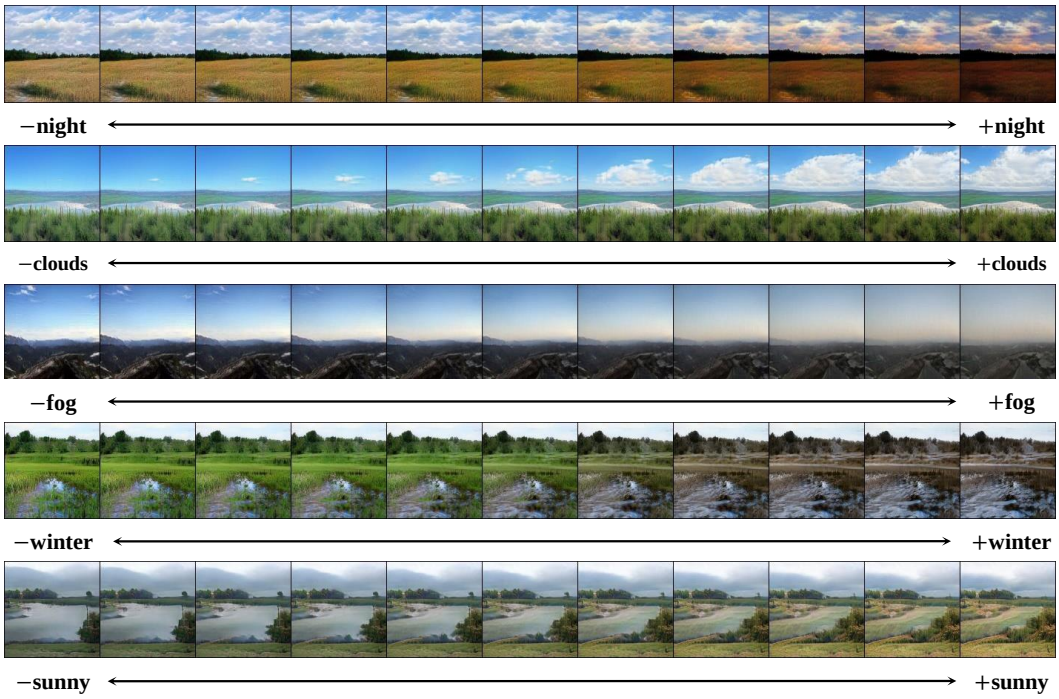

Figure 7: Edited samples of 5 attributes in Transient Attributes dataset [13]. We demonstrate continuous changes in samples with each attribute increasing from 0 to 1 in the above picture.

ing [30]. Since the official code of EnjoyEditing is not released yet, we re-implement it by ourselves thanks to their concise framework. To fairly compare the performance of the two methods, we adopt [1] to encode one real image into the latent space of StyleGAN2, and compare the attribute editing performance on the same image.

Fig. 4 shows results of single attribute editing (b, c) and multi attributes editing (d, e, f). For adding bangs, our ACGAN is the only method that adds bangs successfully, while InterFaceGAN [24] and EnjoyEditing [30] just move Hairline a little forward. For adding chubby, our ACGAN achieves most precised and distangled editing result. The result of InterFaceGAN is older than the original image, and the result of EnjoyEditing is not obvious at adding chubby.

For multiple attributes editing evaluation, our ACGAN can modify multiple attributes simultaneously without changing the image identity, as shown in Fig.4. In contrast, the baselines fails to add few attributes or change irrelevant attributes. InterfaceGAN [24] does not make much change to the inverted image when it comes to multiple attribute editing in Fig 4(c) and (d). EnjoyEditing [30] obviously changes hair color to be blonder when we decrease the *blond hair* attribute. The texture details also became more blurred, leads to a worse qualitative appearance as shown in Fig. 4(d).

Quantitative comparison. We evaluate composite attributes editing score δ and identity preserving score ϵ by user study. Following [30, 12], we accumulatively add three sets of composite attributes on the initial image I_0 to generate the corresponding images I_1, I_2, I_3. For each pair in $\{(I_0, I_1), (I_1, I_2), (I_2, I_3)\}$, we invited 30 users to answer: (1) do the pair of images satisfy corresponding composite attributes editing for composite attributes editing score and (2) are the pair of images identical except the previous attribute editing for identity score. For identity preserving, this score is meaningful only when attributes editing is satisfied. Based on this, we propose weighted identity preserving score $\epsilon_w = \delta * \epsilon$. The evaluation results are reported in Table.1. The two numbers in each cell are δ and weighted ϵ_w respectively. Our method shows superiority on composite attributes editing in both editing precision and weighted identity preserving evaluations.

Edited results of all 40 attributes. Previous supervised direction searching methods are only compatible with few (< 4) attributes of the CelebA dataset. Our method has made great progress in this regard, being able to edit all the 40 attributes of the generated samples. Specifically, all the

Table 1: Quantitative evaluation of composite attributes editing. The two numbers in each cell are attribute editing score and weighted identity preserving score respectively. Our method shows superiority on composite attributes editing in both editing precision and weighted identity preserving.

	Young-, Black Hair-, Blond Hair-, Gray Hair+		Smile+, Bushy Eyebrows+, Big Nose+		Chubby-, Double Chin-, Oval Face-	
	δ	ϵ	δ	ϵ	δ	ϵ
InterFaceGAN[24]	45.4	34.0	52.8	35.5	60.3	40.6
EnjoyEditing[30]	65.9	49.1	76.0	59.1	58.9	46.6
Ours	87.7	70.6	84.3	68.5	85.9	65.0

attribute values are initialized to 0.5. For each attribute editing, we simply modify the editing attribute value to 1 while keep other attribute values and original content values unchanged. Fig. 5 demonstrates the effectiveness of the proposed attribute orthogonal space, which empowers our model to well disentangle most attributes except for four "*wearing*" attribute: wearing necklace, wearing necktie, wearing earrings and wearing hat. While analyzing the proportion of these wearable attributes, we found that the failure editing on these attributes is not caused by quantity. For instance, as a wearable attribute, eyeglasses are comparable to the other four wearable attributes in the quantity proportion but present reasonable results. This might because the size and the position of the eyeglasses in the face are relatively fixed, while the other wearable items are not, which makes the eyeglasses attribute editing is easier to learn than other wearable attributes.

4.3 Natural Scene Synthesis

Comparison to image-to-image translation methods. Following [30], we compare our ACGAN with image-to-image methods CycleGAN [29] RelGAN [21] and DRIT++ [15] in Fig.6. We split Transient Attributes dataset into two parts based on atrribute value: **A** part (attribute value > 0.5) and **B** part (attribute value ≤ 0.5). These methods are trained from scratch by translation image from **A** part to **B** part and vice versa. As shown in Fig. 6, compared with other methods, we notice that our ACGAN accurately change the attributes of the original images, which suggests that our model performs well in natural attributes editing.

Continuous attributes editing. Since the attributes of natural images are easy to observe, we conduct continuous attributes edits on Transient Attributes dataset. Similar to face synthesis, ACGAN also shows superior performance in editing natural scene attributes, i.e., "night", "cloud", "fog", "winter" and "sunny". We present our natural scene attribute continuously editing results in Fig.7. Benefiting from disentangled attribute from content, ACGAN achieves disentangled and accurate attribute editing in natural scenes with regards to image identity preservation.

5 Conclusion

We propose an attribute continuous quantification method based on probability model to solve the scarcity of quantified attributes. Based on this, we propose attribute consistent generative adversarial nets constrained by attribute-consistency loss. By decomposing latent code into content values and orthogonal attribute values, we realize the disentanglement between attributes theoretically for the first time. Extensive experiments prove that our method produces attribute editing results with high fidelity and attribute disentanglement. Our method can be used to control the generated results of GAN or edit real pictures. But there are still areas for improvements. Attribute consistency still require a pretrained attribute regressor. If attribute learning is combined with GAN training, there is still room for further improvement in the performance of attribute editing. The second area is that the editing of real images depends on the result of GAN inversion, which is determined by the structure of GAN itself and the GAN inversion method. The development of these two fields plays a key role in improving the performance of our ACGAN and existing direction searching methods.

References

[1] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4431–4440. IEEE, 2019.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[3] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8789–8797. IEEE Computer Society, 2018.

[4] Y. Choi, Y. Uh, J. Yoo, and J. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8185–8194. IEEE, 2020.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33, 2020.

[8] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5967–5976. IEEE Computer Society, 2017.

[9] A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[10] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019.

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. IEEE, 2020.

[12] M. Kowalski, S. J. Garbin, V. Estellers, T. Baltrusaitis, M. Johnson, and J. Shotton. CONFIG: controllable neural face image generation. In *ECCV (11)*, volume 12356 of *Lecture Notes in Computer Science*, pages 299–315. Springer, 2020.

[13] P. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4):149:1–149:11, 2014.

[14] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014.

[15] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128(10):2402–2417, 2020.

[16] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. *arXiv preprint arXiv:1705.08086*, 2017.

[17] B. Liu, Y. Zhu, K. Song, and A. Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. *CoRR*, abs/2101.04775, 2021.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[19] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6997–7005. IEEE Computer Society, 2017.

[20] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2813–2821. IEEE Computer Society, 2017.

[21] W. Nie, N. Narodytska, and A. Patel. Relgan: Relational generative adversarial networks for text generation. In *International conference on learning representations*, 2018.

[22] A. Plumerault, H. L. Borgne, and C. Hudelot. Controlling generative models with continuous factors of variations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[24] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9240–9249. IEEE, 2020.

[25] A. Verma, H. Qassim, and D. Feinzimer. Residual squeeze CNDS deep learning CNN model for very large scale places image recognition. In *8th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2017, New York City, NY, USA, October 19-21, 2017*, pages 463–469. IEEE, 2017.

[26] A. Voynov and A. Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9786–9796. PMLR, 2020.

[27] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.*, 127(3):302–321, 2019.

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[30] P. Zhuang, O. Koyejo, and A. G. Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. *CoRR*, abs/2102.01187, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Implementation details

We describe ACGAN training process in Algorithm.1 and attribute regressor training procedure in Algorithm.2.

Algorithm 1: Training ACGAN Algorithm

Input: A pretrained attribute regressor R, an initialized GAN consisting of G with parameters θ_G and D with parameters θ_D, batch size K; max interation M, an image dataset Q, learning rate μ;

while $interation \leq M$ **do**

  Sample $z \sim \mathcal{N}(0,1), \alpha \sim \mathcal{U}(0,1)$; Compute $I = G([z, \alpha])$;

  Compute $\hat{\alpha} = R(I)$;

  Compute $L = \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{Reg}(\alpha, \hat{\alpha})$;

  Update $\theta_G = \mu * \partial L / \partial \theta_G$;

end

Result: G and D as ACGAN

Algorithm 2: Training Regressor Algorithm

Input: An image set $\mathcal{D}$ and the corresponding attribute set $\mathcal{A}$, an initialized classifier C where $C(\cdot) = sigmoid(R_c(\cdot))$, R_c is a part of C, an initialized regressor R;

1. Train classifier C with dataset $\{\mathcal{D}, \mathcal{A}\}$;
2. Inference R_c on $\mathcal{D}$ to generate unnormalized attribute labels $\hat{\mathcal{A}}_u$;
3. Normalize $\hat{\mathcal{A}}_u$ to $[0, 1]$ by Eq. 2 to generate $\hat{\mathcal{A}}$;
4. Train regressor R with dataset $\{\mathcal{D}, \hat{\mathcal{A}}\}$;

Result: Continuous attribute label $\hat{\mathcal{A}}$ and the Regressor R

A.1 Inversion details

Considering that the optimization-based inversion methods are more accurate than the model-based methods, we adopt Image2StyleGAN [1] as the inversion method for our ACGAN and baselines. For baselines, we invert a real image into *W+* space of StyleGAN2. For our ACGAN, since the latent code is decomposed into content code and attribute code, we predict attribute code using the pretrained attribute regressor R and optimize content code towards the input real image.

A.2 Questionnaire details

As discussed in quantitative comparison of face synthesis, we generate images $\{I_0, I_1, I_2, I_3\}$ with accumulative composite attributes. For our ACGAN, we first generate I_0 with all attributes set to 0.5, then we reduce *young*, *black hair*, *blond hair* and increase *gray hair* (by 0.3 mostly) on attribute code of I_0 to generate I_1. Note that the attribute values are added in an accumulative manner, so we add *smile bushy* and *big nose* on attribute code of I_1 to generate I_2. The way we adopt to generate I_3 is similar to I_2. For baselines [24] [30], we generate I_0 by randomly sampling latent code, and generate $\{I_1, I_2, I_3\}$ through latent space arithmetic but also in an accumulative manner.

We present some generated face image sequences $\{I_0, I_1, I_2, I_3\}$ in Fig. 8. For each pair in $\{(I_0, I_1), (I_1, I_2), (I_2, I_3)\}$ of all three different methods, we have the following questions:

1. Does the image pair $\{I_0, I_1\}$ satisfy the property changes of "older", "less black hair", "less blond hair" and "more gray hair"?
2. Ignoring the attribute changes described in the previous question, does the image pair $\{I_0, I_1\}$ have the same identity?
3. Does the image pair $\{I_1, I_2\}$ satisfy the property changes of "more smiling", "more bushy eyebrows" and "bigger nose"? attribute
4. Ignoring the property changes described in the previous question, does the image pair $\{I_1, I_2\}$ have the same identity?

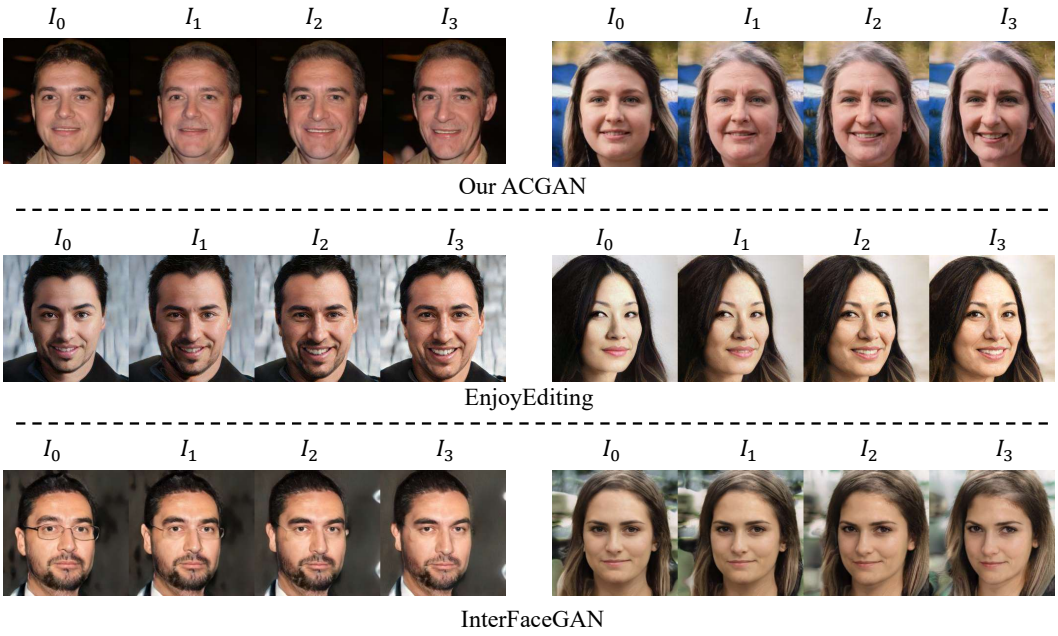

Figure 8: Questionnaire samples. The names of our method and the compared approaches are hidden when we conduct the questionnaire survey in the user study.

Table 2: The weighted identity preserving score is acquired by ArcFace. This score is computed in the same manner as Table 1.

	Young-, Black Hair-, Blond Hair-, Gray Hair+	Smile+, Bushy Eyebrows+, Big Nose+	Chubby-, Double Chin-, Oval Face-
InterFaceGAN [24]	39.0	42.3	50.4
EnjoyEditing [30]	51.2	62.9	53.8
Ours	**61.5**	**64.9**	**67.5**

5. Does the image pair $\{I_2, I_3\}$ satisfy the property changes of "less chubby", "less double chin" and "less oval face"?
6. Ignoring the attribute changes described in the previous question, does the image pair $\{I_2, I_3\}$ have the same identity?

B Results and Comparisons

B.1 Additional comparison to existing methods

Face recognition. Considering the great progress achieved in face recognition, we introduce the advanced ArcFace model to evaluate the identity preserving score of each image pair described in A.2. Specifically, we calculate the Cosine similarity of each image pair by ArcFace as the identity preserving score, then we obtain the weighted identity preserving score in the same manner as Tab. 1. As shown in Table 2, our method achieves superior performance compared with InterFaceGAN [24] and EnjoyEditing [30]. The experimental results further validate the strong identity preserving ability of our ACGAN while editing multiple attributes.

B.2 Additional visual results

In Fig. 1, 4, 5, 6 and 7 of the main paper, we have shown our results of editing attributes on facial images and natural scene images. Here we show additional attributes editing results in Fig. 9, Fig. 10, Fig. 11 and Fig. 12.

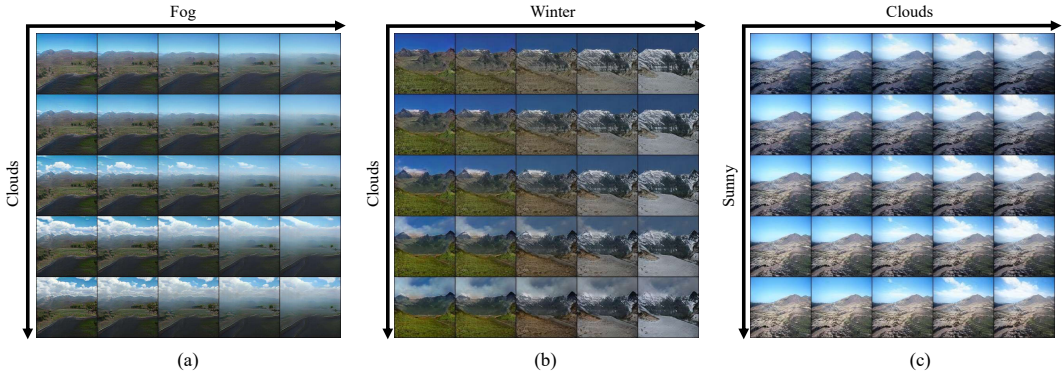

Figure 9: Multiple attributes editing results. We manipulate attribute of images towards *clouds* and **fog** in (a), *clouds* and *winter* in (b), *sunny* and *clouds* in (c). Zoom in for better view.

Natural scene synthesis. We present additional results of multiple attributes editing on natural scene images in Fig. 9. The generated images generated by our method are manipulated on two attributes simultaneously. Fig. 9(a) shows the results of editing *Clouds* and *Fog*, the top left picture shows neither fog nor clouds, the downwards pictures tend to generate more clouds and less fog, the rightward pictures tend to generate more fog and less clouds; the bottom right picture contains both clouds and fog. Fig. 9(b) and Fig. 9(c) show similar performance in disentangling *Clouds* with *Winter* and *Sunny*. This experiment demonstrates the advanced performance of our method in disentangling natural scene attributes.

Figure 10: (**Video figure**, best viewed in Adobe Acrobat) Face image generation by our ACGAN towards multiple attributes editing. More examples can be generated by the our codes in the supplementary material.

Face synthesis. Here we show additional results of accumulatively manipulating multiple attributes on face images. The attributes *Age*, *Eyeglasses*, *Beard*, *Chubby* and *Smiling* are added and removed in turn to various images with *Male* attribute. For the image with *Female* attribute, we replace the *Beard* attribute with the *Makeup* attribute. As shown in Fig. 10, Fig. 11 and Fig. 12, our ACGAN is qualified to complex manipulation on multiple facial attributes.

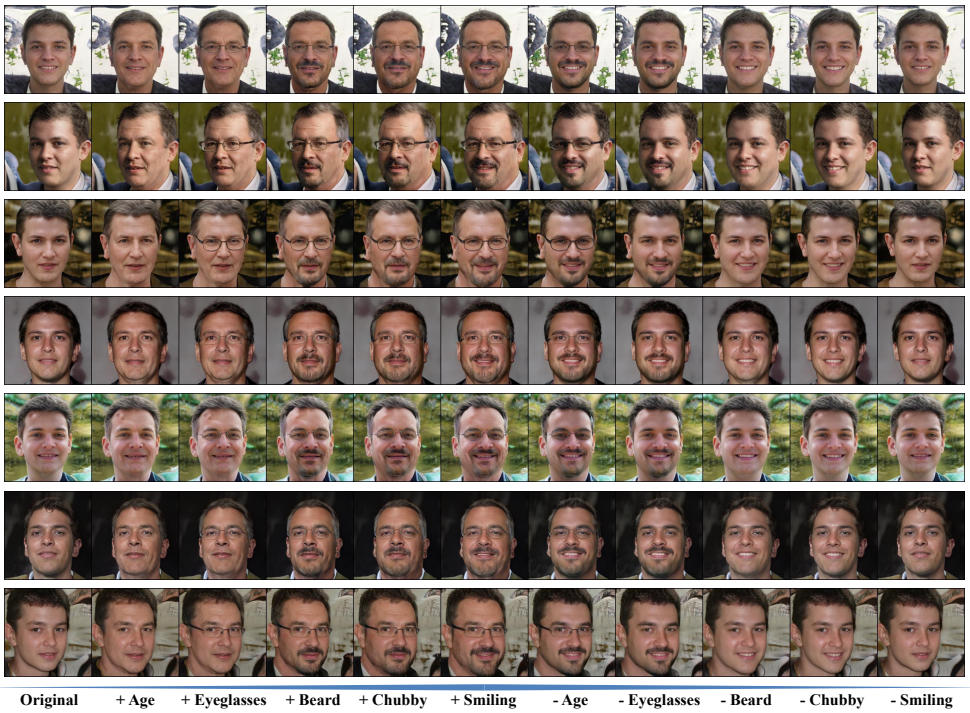

Figure 11: Additional accumulative attributes editing results. We manipulate attributes of images with *Male* attributes towards *+Age*, *+Eyeglasses*, *+Beard*, *+Chubby*, *+Smiling*, *-Age*, *-Eyeglasses*, *-Beard*, *-Chubby* and *-Smiling*. Zoom in for better view.

Figure 12: Additional accumulative attributes editing results. We manipulate attributes of images with *Female* attributes towards *+Age*, *+Eyeglasses*, *+Makeup*, *+Chubby*, *+Smiling*, *-Age*, *-Eyeglasses*, *-Makeup*, *-Chubby* and *-Smiling*. Zoom in for better view.