Interpreting the Latent Space of GANs for Semantic Face Editing

Yujun Shen[1], Jinjin Gu[2], Xiaoou Tang[1], Bolei Zhou[1]
[1]The Chinese University of Hong Kong [2]The Chinese University of Hong Kong, Shenzhen
{sy116, xtang, bzhou}@ie.cuhk.edu.hk, jinjingu@link.cuhk.edu.cn

Figure 1: Manipulating various facial attributes through varying the latent codes of a well-trained GAN model. The first column shows the original synthesis from PGGAN [21], while each of the other columns shows the results of manipulating a specific attribute.

Abstract

Despite the recent advance of Generative Adversarial Networks (GANs) in high-fidelity image synthesis, there lacks enough understanding of how GANs are able to map a latent code sampled from a random distribution to a photo-realistic image. Previous work assumes the latent space learned by GANs follows a distributed representation but observes the vector arithmetic phenomenon. In this work, we propose a novel framework, called InterFaceGAN, for semantic face editing by interpreting the latent semantics learned by GANs. In this framework, we conduct a detailed study on how different semantics are encoded in the latent space of GANs for face synthesis. We find that the latent code of well-trained generative models actually learns a disentangled representation after linear transformations. We explore the disentanglement between various semantics and manage to decouple some entangled semantics with subspace projection, leading to more precise control of facial attributes. Besides manipulating gender, age, expression, and the presence of eyeglasses, we can even vary the face pose as well as fix the artifacts accidentally generated by GAN models. The proposed method is further applied to achieve real image manipulation when combined with GAN inversion methods or some encoder-involved models. Extensive results suggest that learning to synthesize faces spontaneously brings a disentangled and controllable facial attribute representation.[1]

1. Introduction

Generative Adversarial Networks (GANs) [15] have significantly advanced image synthesis in recent years. The rationale behind GANs is to learn the mapping from a latent distribution to the real data through adversarial training. After learning such a non-linear mapping, GAN is capable of producing photo-realistic images from randomly sampled latent codes. However, it is uncertain how semantics originate and are organized in the latent space. Taking face synthesis as an example, when sampling a latent code to produce an image, how the code is able to determine various semantic attributes (*e.g.*, gender and age) of the output face, and how these attributes are entangled with each other?

[1]Code and models are available at this link.

Existing work typically focuses on improving the synthesis quality of GANs [40, 28, 21, 8, 22], however, few efforts have been made on studying what a GAN actually learns with respect to the latent space. Radford *et al.* [31] first observes the vector arithmetic property in the latent space. A recent work [4] further shows that some units from intermediate layers of the GAN generator are specialized to synthesize certain visual concepts, such as sofa and TV for living room generation. Even so, there lacks enough understanding of how GAN connects the latent space and the image semantic space, as well as how the latent code can be used for image editing.

In this paper, we propose a framework *InterFaceGAN*, short for *Inter*preting *Face GAN*s, to identify the semantics encoded in the latent space of well-trained face synthesis models and then utilize them for semantic face editing. Beyond the vector arithmetic property, this framework provides both theoretical analysis and experimental results to verify that *linear* subspaces align with different *true-or-false* semantics emerging in the latent space. We further study the disentanglement between different semantics and show that we can decouple some entangled attributes (*e.g.*, old people are more likely to wear eyeglasses then young people) through the linear subspace projection. These disentangled semantics enable precise control of facial attributes with any given GAN model *without retraining*.

Our contributions are summarized as follows:

- We propose InterFaceGAN to explore how a single or multiple semantics are encoded in the latent space of GANs, such as PGGAN [21] and StyleGAN [22], and observe that GANs spontaneously learn various latent subspaces corresponding to specific attributes. These attribute representations become disentangled after some linear transformations.
- We show that InterFaceGAN enables semantic face editing with any *fixed* pre-trained GAN model. Some results are shown in Fig.1. Besides gender, age, expression, and the presence of eyeglasses, we can noticeably also vary the face pose or correct some artifacts produced by GANs.
- We extend InterFaceGAN to real image editing with GAN inversion methods and encoder-involved models. We successfully manipulate the attributes of real faces by simply varying the latent code, even with GANs that are not specifically designed for the editing task.

1.1. Related Work

Generative Adversarial Networks. GAN [15] has brought wide attention in recent years due to its great potential in producing photo-realistic images [1, 17, 6, 40, 28, 21, 8, 22]. It typically takes a sampled latent code as the input and outputs an image synthesis. To make GANs applicable for real image processing, existing methods proposed to reverse the mapping from the latent space to the image space [30, 42, 27, 5, 16] or learn an additional encoder associated with the GAN training [13, 12, 41]. Despite this tremendous success, little work has been done on understanding how GANs learn to connect the input latent space with the semantics in the real visual world.

Study on Latent Space of GANs. Latent space of GANs is generally treated as Riemannian manifold [9, 2, 23]. Prior work focused on exploring how to make the output image vary smoothly from one synthesis to another through interpolation in the latent space, regardless of whether the image is semantically controllable [24, 32]. GLO [7] optimized the generator and latent code simultaneously to learn a better latent space. However, the study on how a well-trained GAN is able to encode different semantics inside the latent space is still missing. Some work has observed the vector arithmetic property [31, 36]. Beyond that, this work provides a detailed analysis of the semantics encoded in the latent space from both the property of a single semantic and the disentanglement of multiple semantics. Some concurrent work also explores the latent semantics learned by GANs. Jahanian *et al.* [20] studies the steerability of GANs concerning camera motion and image color tone. Goetschalckx *et al.* [14] improves the memorability of the output image. Yang *et al.* [38] explores the hierarchical semantics in the deep generative representations for scene synthesis. Unlike them, we focus on facial attributes emerging in GANs for face synthesis and extend our method to real image manipulation.

Semantic Face Editing with GANs. Semantic face editing aims at manipulating facial attributes of a given image. Compared to unconditional GANs which can generate image arbitrarily, semantic editing expects the model to only change the target attribute but maintain other information of the input face. To achieve this goal, current methods required carefully designed loss functions [29, 10, 35], introduction of additional attribute labels or features [25, 39, 3, 37, 34], or special architectures [11, 33] to train new models. However, the synthesis resolution and quality of these models are far behind those of native GANs, like PGGAN [21] and StyleGAN [22]. Different from previous learning-based methods, this work explores the interpretable semantics inside the latent space of *fixed* GAN models, and *turns unconstrained GANs to controllable GANs* by varying the latent code.

2. Framework of InterFaceGAN

In this section, we introduce the framework of InterFaceGAN, which first provides a rigorous analysis of the semantic attributes emerging in the latent space of well-trained GAN models, and then constructs a manipulation pipeline of leveraging the semantics in the latent code for facial attribute editing.

2.1. Semantics in the Latent Space

Given a well-trained GAN model, the generator can be formulated as a deterministic function $g : \mathcal{Z} \to \mathcal{X}$. Here, $\mathcal{Z} \subseteq \mathbb{R}^d$ denotes the d-dimensional latent space, for which Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is commonly used [28, 21, 8, 22]. $\mathcal{X}$ stands for the image space, where each sample $\mathbf{x}$ possesses certain semantic information, like gender and age for face model. Suppose we have a semantic scoring function $f_S : \mathcal{X} \to \mathcal{S}$, where $\mathcal{S} \subseteq \mathbb{R}^m$ represents the semantic space with m semantics. We can bridge the latent space $\mathcal{Z}$ and the semantic space $\mathcal{S}$ with $\mathbf{s} = f_S(g(\mathbf{z}))$, where $\mathbf{s}$ and $\mathbf{z}$ denote the semantic scores and the sampled latent code respectively.

Single Semantic. It has been widely observed that when linearly interpolating two latent codes $\mathbf{z}_1$ and $\mathbf{z}_2$, the appearance of the corresponding synthesis changes continuously [31, 8, 22]. It implicitly means that the semantics contained in the image also change gradually. According to *Property 1*, the linear interpolation between $\mathbf{z}_1$ and $\mathbf{z}_2$ forms a direction in $\mathcal{Z}$, which further defines a hyperplane. We therefore make an assumption[2] that for any binary semantic (*e.g.*, male *v.s.* female), there exists a hyperplane in the latent space serving as the separation boundary. Semantic remains the same when the latent code walks within the same side of the hyperplane yet turns into the opposite when across the boundary.

Given a hyperplane with a unit normal vector $\mathbf{n} \in \mathbb{R}^d$, we define the "distance" from a sample $\mathbf{z}$ to this hyperplane as

$$\mathrm{d}(\mathbf{n}, \mathbf{z}) = \mathbf{n}^T \mathbf{z}. \tag{1}$$

Here, $\mathrm{d}(\cdot, \cdot)$ is not a strictly defined distance since it can be negative. When $\mathbf{z}$ lies near the boundary and is moved toward and across the hyperplane, both the "distance" and the semantic score vary accordingly. And it is just at the time when the "distance" changes its numerical sign that the semantic attribute reverses. We therefore expect these two to be linearly dependent with

$$f(g(\mathbf{z})) = \lambda \mathrm{d}(\mathbf{n}, \mathbf{z}), \tag{2}$$

where $f(\cdot)$ is the scoring function for a particular semantic, and $\lambda > 0$ is a scalar to measure how fast the semantic varies along with the change of distance. According to *Property 2*, random samples drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ are very likely to locate close enough to a given hyperplane. Therefore, the corresponding semantic can be modeled by the linear subspace that is defined by $\mathbf{n}$.

Property 1 *Given $\mathbf{n} \in \mathbb{R}^d$ with $\mathbf{n} \neq \mathbf{0}$, the set $\{\mathbf{z} \in \mathbb{R}^d : \mathbf{n}^T\mathbf{z} = 0\}$ defines a hyperplane in $\mathbb{R}^d$, and $\mathbf{n}$ is called the normal vector. All vectors $\mathbf{z} \in \mathbb{R}^d$ satisfying $\mathbf{n}^T\mathbf{z} > 0$ locate from the same side of the hyperplane.*

[2]This assumption is empirically verified in Sec.3.1.

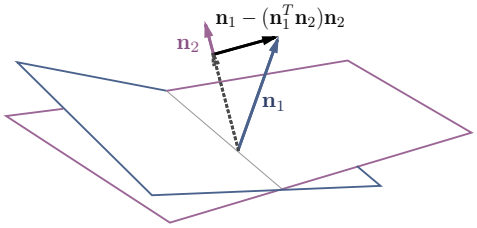

Figure 2: Illustration of the conditional manipulation in subspace. The projection of $\mathbf{n}_1$ onto $\mathbf{n}_2$ is subtracted from $\mathbf{n}_1$, resulting in a new direction $\mathbf{n}_1 - (\mathbf{n}_1^T\mathbf{n}_2)\mathbf{n}_2$.

Property 2 *Given $\mathbf{n} \in \mathbb{R}^d$ with $\mathbf{n}^T\mathbf{n} = 1$, which defines a hyperplane, and a multivariate random variable $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we have $\mathrm{P}(|\mathbf{n}^T\mathbf{z}| \leq 2\alpha\sqrt{\frac{d}{d-2}}) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$ for any $\alpha \geq 1$ and $d \geq 4$. Here, $\mathrm{P}(\cdot)$ stands for probability and c is a fixed positive constant.*[3]

Multiple Semantics. When the case comes to m different semantics, we have

$$\mathbf{s} \equiv f_S(g(\mathbf{z})) = \Lambda \mathbf{N}^T \mathbf{z}, \tag{3}$$

where $\mathbf{s} = [s_1, \ldots, s_m]^T$ denotes the semantic scores, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ is a diagonal matrix containing the linear coefficients, and $\mathbf{N} = [\mathbf{n}_1, \ldots, \mathbf{n}_m]$ indicates the separation boundaries. Aware of the distribution of random sample $\mathbf{z}$, which is $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we can easily compute the mean and covariance matrix of the semantic scores $\mathbf{s}$ as

$$\boldsymbol{\mu}_\mathbf{s} = \mathbb{E}(\Lambda \mathbf{N}^T \mathbf{z}) = \Lambda \mathbf{N}^T \mathbb{E}(\mathbf{z}) = \mathbf{0}, \tag{4}$$

$$\begin{aligned} \boldsymbol{\Sigma}_\mathbf{s} &= \mathbb{E}(\Lambda \mathbf{N}^T \mathbf{z}\mathbf{z}^T \mathbf{N}\Lambda^T) = \Lambda \mathbf{N}^T \mathbb{E}(\mathbf{z}\mathbf{z}^T)\mathbf{N}\Lambda^T \\ &= \Lambda \mathbf{N}^T \mathbf{N}\Lambda. \end{aligned} \tag{5}$$

We therefore have $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{s})$, which is a multivariate normal distribution. Different entries of $\mathbf{s}$ are disentangled if and only if $\boldsymbol{\Sigma}_\mathbf{s}$ is a diagonal matrix, which requires $\{\mathbf{n}_1, \ldots, \mathbf{n}_m\}$ to be orthogonal with each other. If this condition does not hold, some semantics will correlate with each other and $\mathbf{n}_i^T\mathbf{n}_j$ can be used to measure the entanglement between the i-th and j-th semantics.

2.2. Manipulation in the Latent Space

In this part, we introduce how to use the semantics found in latent space for image editing.

Single Attribute Manipulation. According to Eq.(2), to manipulate the attribute of a synthesized image, we can easily edit the original latent code $\mathbf{z}$ with $\mathbf{z}_{edit} = \mathbf{z} + \alpha\mathbf{n}$. It will make the synthesis look more positive on such semantic with $\alpha > 0$, since the score becomes $f(g(\mathbf{z}_{edit})) = f(g(\mathbf{z})) + \lambda\alpha$ after editing. Similarly, $\alpha < 0$ will make the synthesis look more negative.

[3]When $d = 512$, we have $P(|\mathbf{n}^T\mathbf{z}| > 5.0) < 1e^{-6}$. It suggests that almost all sampled latent codes are expected to locate within 5 unit-length to the boundary. Proof can be found in **Appendix**.

Conditional Manipulation. When there is more than one attribute, editing one may affect another since some semantics can be coupled with each other. To achieve more precise control, we propose *conditional manipulation* by manually forcing $\mathbf{N}^T\mathbf{N}$ in Eq.(5) to be diagonal. In particular, we use projection to orthogonalize different vectors. As shown in Fig.2, given two hyperplanes with normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$, we find a projected direction $\mathbf{n}_1 - (\mathbf{n}_1^T\mathbf{n}_2)\mathbf{n}_2$, such that moving samples along this new direction can change "attribute 1" without affecting "attribute 2". We call this operation as conditional manipulation. If there is more than one attribute to be conditioned on, we just subtract the projection from the primal direction onto the plane that is constructed by all conditioned directions.

Real Image Manipulation. Since our approach enables semantic editing from the latent space of a *fixed* GAN model, we need to first map a real image to a latent code before performing manipulation. For this purpose, existing methods have proposed to directly optimize the latent code to minimize the reconstruction loss [27], or to learn an extra encoder to invert the target image back to latent space [42, 5]. There are also some models that have already involved an encoder along with the training process of GANs [13, 12, 41], which we can directly use for inference.

3. Experiments

In this section, we evaluate InterFaceGAN with state-of-the-art GAN models, PGGAN [21] and StyleGAN [22]. Specifically, the experiments in Sec.3.1, Sec.3.2, and Sec.3.3 are conducted on PGGAN to interpret the latent space of the traditional generator. Experiments in Sec.3.4 are carried out on StyleGAN to investigate the style-based generator and also compare the differences between the two sets of latent representations in StyleGAN. We also apply our approach to real images in Sec.3.5 to see how the semantics implicitly learned by GANs can be applied to real face editing. Implementation details can be found in **Appendix**.

3.1. Latent Space Separation

As mentioned in Sec.2.1, our framework is based on an assumption that for any binary attribute, there exists a hyperplane in latent space such that all samples from the same side are with the same attribute. Accordingly, we would like to first evaluate the correctness of this assumption to make the remaining analysis considerable.

We train five independent linear SVMs on pose, smile, age, gender, eyeglasses, and then evaluate them on the validation set ($6K$ samples with high confidence level on attribute scores) as well as the entire set ($480K$ random samples). Tab.1 shows the results. We find that all linear boundaries achieve over 95% accuracy on the validation set and over 75% on the entire set, suggesting that for a binary attribute, there exists a linear hyperplane in the latent space that can well separate the data into two groups.

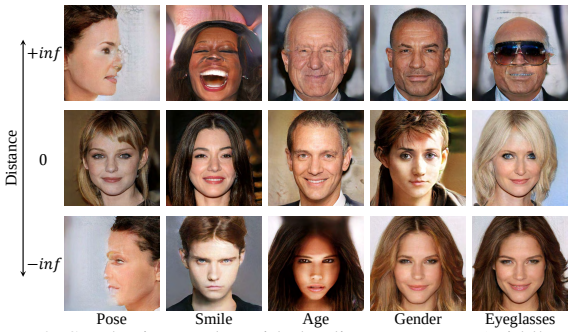

Figure 3: Synthesis samples with the distance near to (middle row) and extremely far away from (top and bottom rows) the separation boundary. Each column corresponds to a particular attribute.

Table 1: Classification accuracy (%) on separation boundaries in latent space with respect to different attributes.

Dataset	Pose	Smile	Age	Gender	Eyeglasses
Validation	100.0	96.9	97.9	98.7	95.6
All	90.3	78.5	75.3	84.2	80.1

We also visualize some samples in Fig.3 by ranking them with the distance to the decision boundary. Note that those extreme cases (first and last row in Fig.3) are very unlikely to be directly sampled, instead constructed by moving a latent code towards the normal direction "infinitely". From Fig.3, we can tell that the positive samples and negative samples are distinguishable to each other with respect to the corresponding attribute.

3.2. Latent Space Manipulation

In this part, we verify whether the semantics found by InterFaceGAN are manipulable.

Manipulating Single Attribute. Fig.4 plots the manipulation results on five different attributes. It suggests that our manipulation approach performs well on all attributes in both positive and negative directions. Particularly on *pose* attribute, we observe that even the boundary is searched by solving a bi-classification problem, moving the latent code can produce continuous changing. Furthermore, although there lacks enough data with extreme poses in the training set, GAN is capable of imagining how profile faces should look like. The same situation also happens on eyeglasses attribute. We can manually create a lot of faces wearing eyeglasses despite the inadequate data in the training set. These two observations provide strong evidence that GAN does not produce images randomly, but learns some interpretable semantics from the latent space.

Distance Effect of Semantic Subspace. When manipulating the latent code, we observe an interesting distance effect that the samples will suffer from severe changes in appearance if being moved too far from the boundary, and

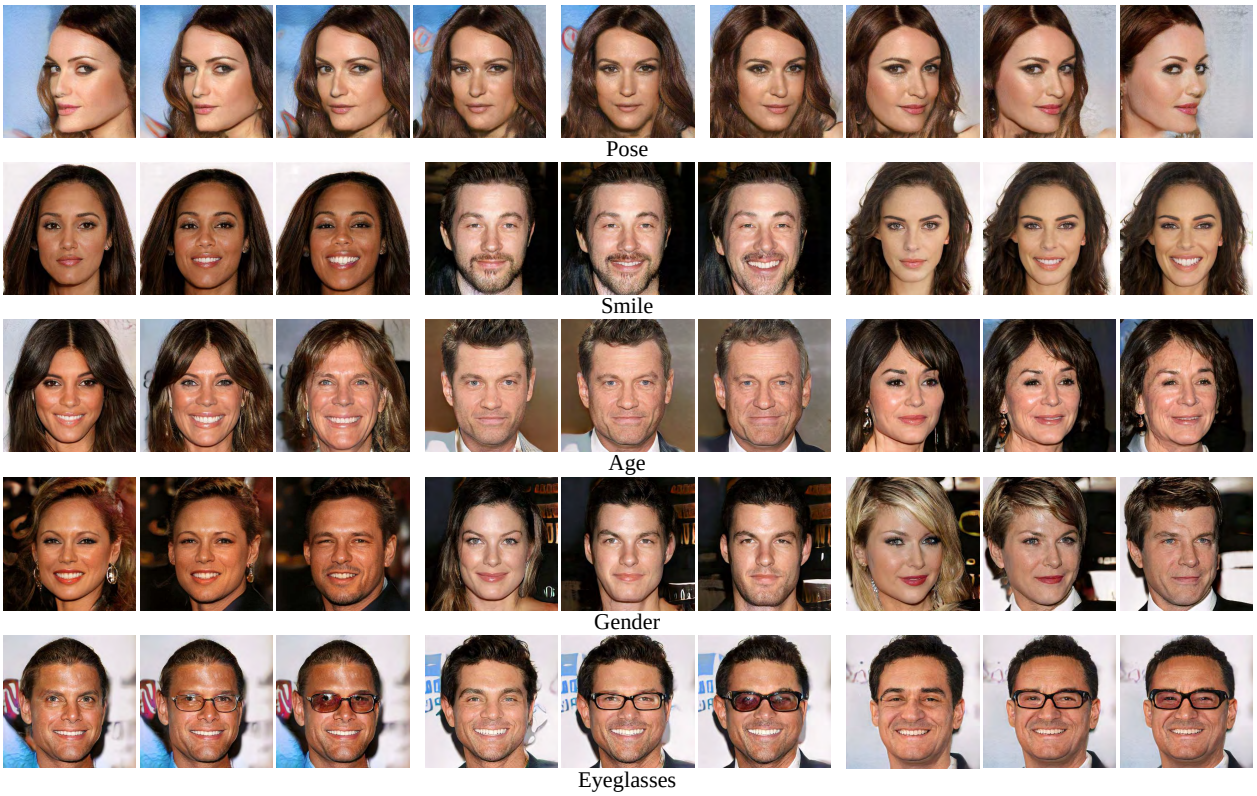

Figure 4: Single attribute manipulation results. The first row shows the same person under gradually changed poses. The following rows correspond to the results of manipulating four different attributes. For each set of three samples in a row, the central one is the original synthesis, while the left and right stand for the results by moving the latent code along negative and positive direction respectively.

Figure 5: Illustration of the distance effect by taking gender manipulation as an example. The image in the red dashed box stands for the original synthesis. Our approach performs well when the latent code locates close to the boundary. However, when the distance keeps increasing, the synthesized images are no longer like the same person.

finally tend to become the extreme cases shown in Fig.3. Fig.5 illustrates this phenomenon by taking gender editing as an instance. Near-boundary manipulation works well. When samples go beyond a certain region[4], however, the editing results are no longer like the original face anymore. But this effect does not affect our understanding of the disentangled semantics in latent space. That is because such extreme samples are very unlikely to be directly drawn from a standard normal distribution, which is pointed out in **Property 2** in Sec.2.1. Instead, they are constructed manually by keeping moving a normally sampled latent code along a certain direction. In this way, we can get a better interpretation on the latent semantics of GANs.

Artifacts Correction. We further apply our approach to fix the artifacts that sometimes occurred in the synthesized

Figure 6: Examples on fixing the artifacts that GAN has generated. First row shows some bad generation results, while the following two rows present the gradually corrected synthesis by moving the latent codes along the positive "quality" direction.

[4]We choose 5.0 as the threshold.

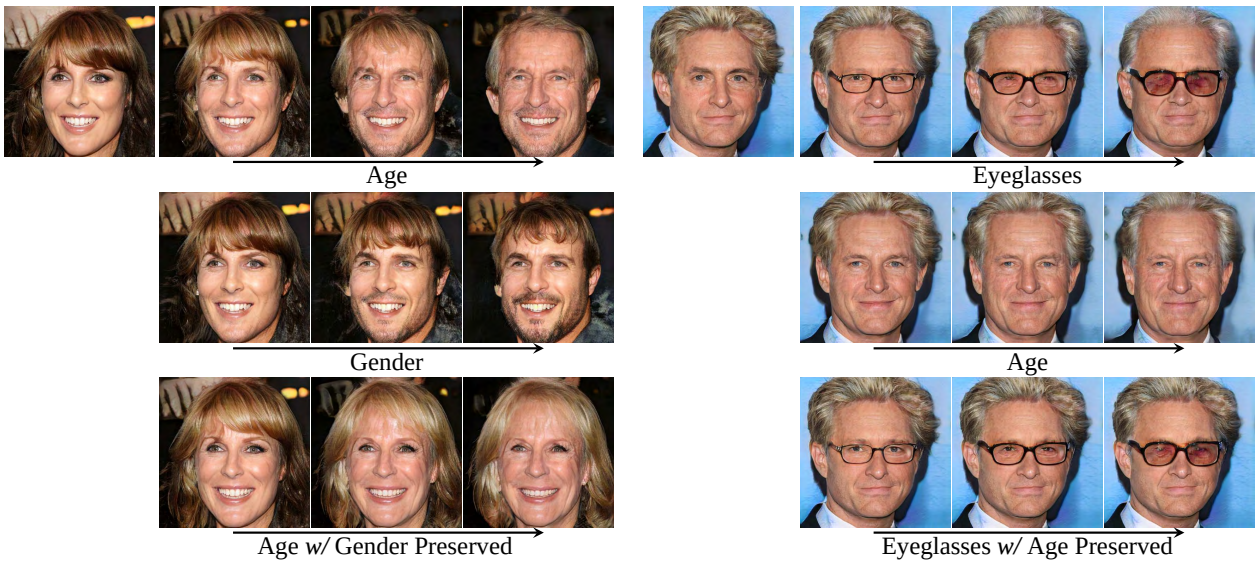

Figure 7: Examples for conditional manipulation. The first two rows show the manipulation results along with the original directions learned by SVMs for two attributes independently. The last row edits the faces by varying one attribute with the other one unchanged.

outputs. We manually labeled $4K$ bad synthesis and then trained a linear SVM to find the separation hyperplane, same as other attributes. We surprisingly find that GAN also encodes such information in latent space. Based on this discovery, we are capable of correcting some mistakes GAN has made in the generation process, as shown in Fig.6.

3.3. Conditional Manipulation

In this section, we study the disentanglement between different attributes and evaluate the conditional manipulation approach.

Correlation between Attributes. Different from [22] which introduced perceptual path length and linear separability to measure the disentanglement property of latent space, we focus more on the relationships between different hidden semantics and study how they are coupled with each other. Here, two different metrics are used to measure the correlation between two attributes. (i) We compute the cosine similarity between two directions as $\cos(\mathbf{n}_1, \mathbf{n}_2) = \mathbf{n}_1^T \mathbf{n}_2$, where $\mathbf{n}_1$ and $\mathbf{n}_2$ stand for unit vectors. (ii) We treat each attribute score as a random variable, and use the attribute distribution observed from all $500K$ synthesized data to compute the correlation coefficient ρ. Here, we have $\rho_{A_1 A_2} = \frac{Cov(A_1, A_2)}{\sigma_{A_1} \sigma_{A_2}}$, where A_1 and A_2 represent two random variables with respect to two attributes. $Cov(\cdot, \cdot)$ stands for covariance, and σ denotes standard deviation.

Tab.2 and Tab.3 report the results. We can tell that attributes behave similarly under these two metrics, showing that our InterFaceGAN is able to accurately identify the semantics hidden in latent space. We also find that pose and smile are almost orthogonal to other attributes. Nevertheless, gender, age, and eyeglasses are highly correlated with each other. This observation reflects the attribute correlation in the training dataset (*i.e.*, CelebA-HQ [21]) to some extent, where male old people are more likely to wear eyeglasses. This characteristic is also captured by GAN when learning to produce real observation.

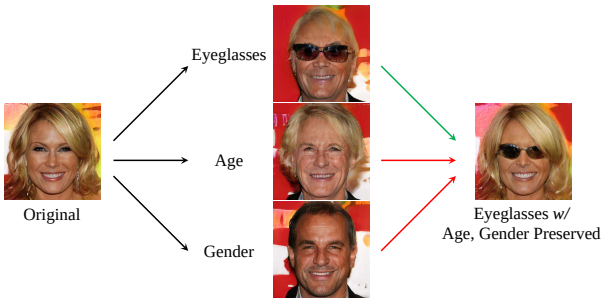

Figure 8: Examples for conditional manipulation with more than one conditions. Left: Original synthesis. Middle: Manipulations along single boundary. Right: Conditional manipulation. **Green** arrow: Primal direction. **Red** arrows: Projection subtraction.

Table 2: Correlation matrix of attribute boundaries.

	Pose	Smile	Age	Gender	Eyeglasses
Pose	1.00	-0.04	-0.06	-0.05	-0.04
Smile	-	1.00	0.04	-0.10	-0.05
Age	-	-	1.00	0.49	0.38
Gender	-	-	-	1.00	0.52
Eyeglasses	-	-	-	-	1.00

Table 3: Correlation matrix of synthesized attribute distributions.

	Pose	Smile	Age	Gender	Eyeglasses
Pose	1.00	-0.01	-0.01	-0.02	0.00
Smile	-	1.00	0.02	-0.08	-0.01
Age	-	-	1.00	0.42	0.35
Gender	-	-	-	1.00	0.47
Eyeglasses	-	-	-	-	1.00

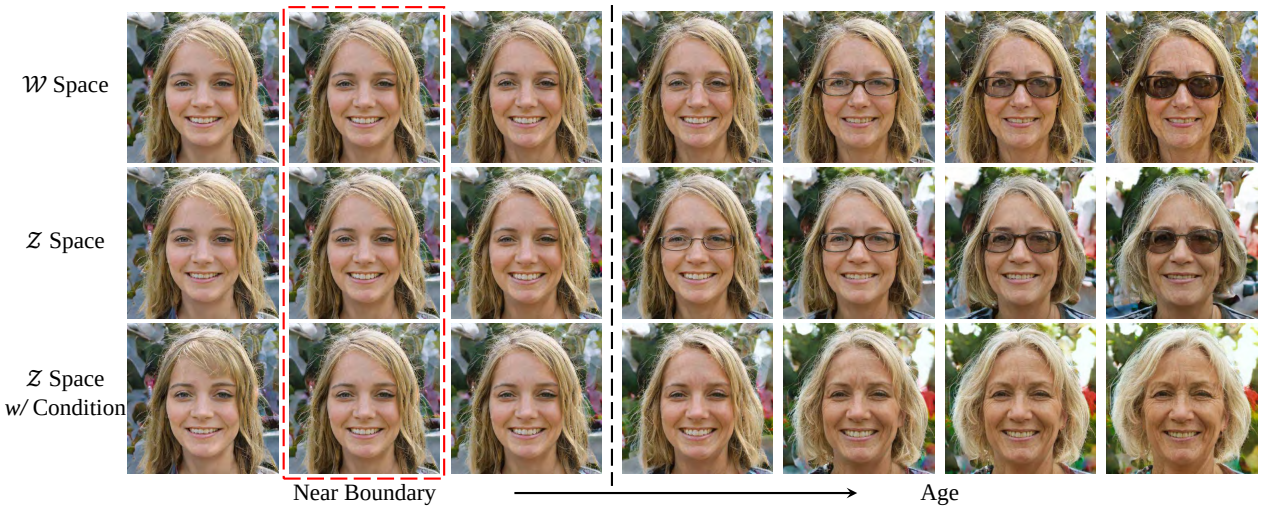

Figure 9: Analysis on the latent space $\mathcal{Z}$ and disentangled latent space $\mathcal{W}$ of StyleGAN [22] by taking age manipulation as an example. $\mathcal{W}$ space behaves better for long term manipulation, but the flaw in $\mathcal{Z}$ space can be fixed by projection (*i.e.*, conditional manipulation) to achieve better performance.

Conditional Manipulation. To decorrelate different semantics for independent facial attribute editing, we propose conditional manipulation in Sec.2.2. Fig.7 shows some results by manipulating one attribute with another one as a condition. Taking the left sample in Fig.7 as an example, the results tend to become male when being edited to get old (first row). We fix this problem by subtracting its projection onto the gender direction (second row) from age direction, resulting in a new direction. In this way, we can make sure the gender component is barely affected when the sample is moved along the projected direction (third row). Fig.8 shows conditional manipulation with more than one constraint, where we add glasses by conditionally preserving age and gender. In the beginning, adding eyeglasses is entangled with changing both age and gender. But we manage to add glasses without affecting age and gender with projection operation. These two experiments show that our proposed conditional approach helps to achieve independent and precise attribute control.

3.4. Results on StyleGAN

Different from conventional GANs, StyleGAN [22] proposed style-based generator. Basically, StyleGAN learns to map the latent code from space $\mathcal{Z}$ to another high dimensional space $\mathcal{W}$ before feeding it into the generator. As pointed out in [22], $\mathcal{W}$ shows much stronger disentanglement property than $\mathcal{Z}$, since $\mathcal{W}$ is not restricted to any certain distribution and can better model the underlying character of real data.

We did a similar analysis on both $\mathcal{Z}$ and $\mathcal{W}$ spaces of StyleGAN as did to PGGAN and found that $\mathcal{W}$ space indeed learns a more disentangled representation, as pointed out by [22]. Such disentanglement helps $\mathcal{W}$ space achieve strong superiority over $\mathcal{Z}$ space for attribute editing. As shown in Fig.9, age and eyeglasses are also entangled in StyleGAN model. Compared to $\mathcal{Z}$ space (second row), $\mathcal{W}$ space (first row) performs better, especially in long-distance manipulation. Nevertheless, we can use the conditional manipulation trick described in Sec.2.2 to decorrelate these two attributes in $\mathcal{Z}$ space (third row), resulting in more appealing results. This trick, however, cannot be applied to $\mathcal{W}$ space. We found that $\mathcal{W}$ space sometimes captures the attributes correlation that happens in training data and encodes them together as a coupled "style". Taking Fig.9 as an example, "age" and "eyeglasses" are supported to be two independent semantics, but StyleGAN actually learns an eyeglasses-included age direction such that this new direction is somehow orthogonal to the eyeglasses direction itself. In this way, subtracting the projection, which is almost zero, will hardly affect the final results.

3.5. Real Image Manipulation

In this part, we manipulate real faces with the proposed InterFaceGAN to verify whether the semantic attributes learned by GAN can be applied to real faces. Recall that InterFaceGAN achieves semantic face editing by moving the latent code along a certain direction. Accordingly, we need to first invert the given real image back to the latent code. It turns out to be a non-trivial task because GANs do not fully capture all the modes as well as the diversity of the true distribution. To invert a pre-trained GAN model, there are two typical approaches. One is the optimization-based approach, which directly optimizes the latent code with the fixed generator to minimize the pixel-wise reconstruction error [27]. The other is the encoder-based, where an extra encoder network is trained to learn the inverse mapping [42]. We tested the two baseline approaches on PGGAN and StyleGAN.

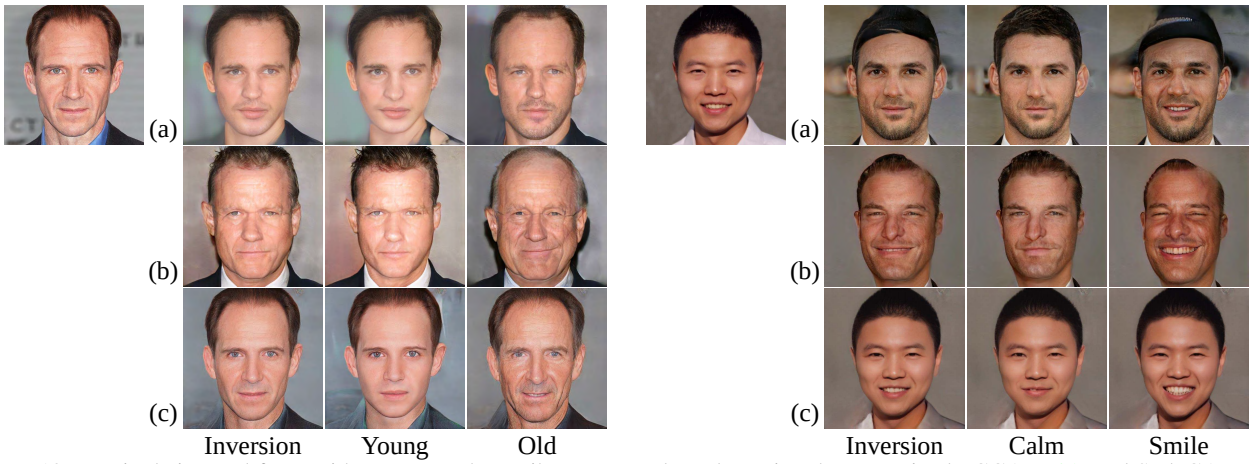

Figure 10: Manipulating real faces with respect to the attributes age and gender, using the pre-trained PGGAN [21] and StyleGAN [22]. Given an image to edit, we first invert it back to the latent code and then manipulate the latent code with InterFaceGAN. On the top left corner is the input real face. From top to bottom: (a) PGGAN with optimization-based inversion method, (b) PGGAN with encoder-based inversion method, (c) StyleGAN with optimization-based inversion method.

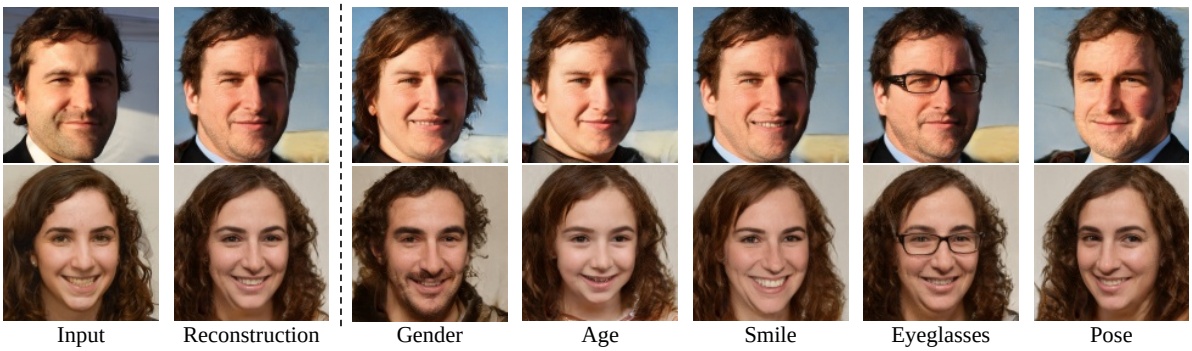

Figure 11: Manipulating real faces with LIA [41], which is a encoder-decoder generative model for high-resolution face synthesis.

Results are shown in Fig.10. We can tell that both optimization-based (first row) and encoder-based (second row) methods show poor performance when inverting PGGAN. This can be imputed to the strong discrepancy between training and testing data distributions. For example, the model tends to generate Western people even the input is an Easterner (see the right example in Fig.10). Even unlike the inputs, however, the inverted images can still be semantically edited with InterFaceGAN. Compared to PGGAN, the results on StyleGAN (third row) are much better. Here, we treat the layer-wise styles (*i.e.*, $\mathbf{w}$ for all layers) as the optimization target. When editing an instance, we push all style codes towards the same direction. As shown in Fig.10, we successfully change the attributes of real face images *without* retraining StyleGAN but leveraging the interpreted semantics from latent space.

We also test InterFaceGAN on encoder-decoder generative models, which train an encoder together with the generator and discriminator. After the model converges, the encoder can be directly used for inference to map a given image to latent space. We apply our method to interpret the latent space of the recent encoder-decoder model LIA [41]. The manipulation result is shown in Fig.11 where we successfully edit the input faces with various attributes, like age and face pose. It suggests that the latent code in the encoder-decoder based generative models also supports semantic manipulation. In addition, compared to Fig.10 (b) where the encoder is separately learned after the GAN model is well-prepared, the encoder trained together with the generator gives better reconstruction as well as manipulation results.

4. Conclusion

We propose InterFaceGAN to interpret the semantics encoded in the latent space of GANs. By leveraging the interpreted semantics as well as the proposed conditional manipulation technique, we are able to precisely control the facial attributes with any fixed GAN model, even turning unconditional GANs to controllable GANs. Extensive experiments suggest that InterFaceGAN can also be applied to real image editing.

Acknowledgement: This work is supported in part by the Early Career Scheme (ECS) through the Research Grants Council of Hong Kong under Grant No.24206219 and in part by SenseTime Collaborative Grant.

Appendix

A. Overview

This appendix contains the following information:

- We introduce the implementation details of the proposed InterFaceGAN in Sec.B.
- We provide the detailed proof of *Property 2* in the main paper in Sec.C.
- Please also refer to this video to see continuous attribute editing results.

B. Implementation Details

We choose five key facial attributes for analysis, including pose, smile (expression), age, gender, and eyeglasses. The corresponding positive directions are defined as turning right, laughing, getting old, changing to male, and wearing eyeglasses. Note that we can always plug in more attributes easily as long as the attribute detector is available.

To better predict these attributes from synthesized images, we train an auxiliary attribute prediction model using the annotations from the CelebA dataset [26] with ResNet-50 network [18]. This model is trained with multi-task losses to simultaneously predict smile, age, gender, eyeglasses, as well as the 5-point facial landmarks. Here, the facial landmarks will be used to compute yaw pose, which is also treated as a binary attribute (left or right) in further analysis. Besides the landmarks, all other attributes are learned as bi-classification problem with softmax cross-entropy loss, while landmarks are optimized with l_2 regression loss. As images produced by PGGAN and StyleGAN are with 1024×1024 resolution, we resize them to 224×224 before feeding them to the attribute model.

Given the pre-trained GAN model, we synthesize $500K$ images by randomly sampling the latent space. There are mainly two reasons in preparing such large-scale data: (i) to eliminate the randomness caused by sampling and make sure the distribution of the latent codes is as expected, and (ii) to get enough wearing-glasses samples, which are really rare in PGGAN model.

To find the semantic boundaries in the latent space, we use the pre-trained attribute prediction model to assign attribute scores for all $500K$ synthesized images. For each attribute, we sort the corresponding scores, and choose $10K$ samples with highest scores and $10K$ with lowest ones as candidates. The reason in doing so is that the prediction model is not absolutely accurate and may produce wrong prediction for ambiguous samples, *e.g.*, middle-aged person for age attribute. We then randomly choose 70% samples from the candidates as the training set to learn a linear SVM, resulting in a decision boundary. Recall that, normal directions of all boundaries are normalized to unit vectors.

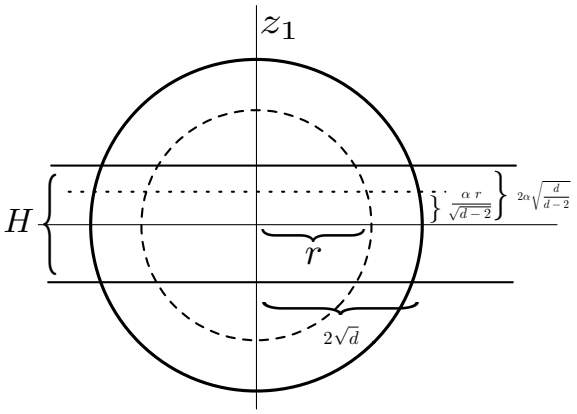

Figure 12: Illustration of *Property 2*, which shows that most of the probability mass of high-dimensional Gaussian distribution lies in the thin slab near the "equator".

Remaining 30% are used for verifying how the linear classifier behaves. Here, for SVM training, the inputs are the $512d$ latent codes, while the binary labels are assigned by the auxiliary attribute prediction model.

C. Proof

In this part, we provide detailed proof of *Property 2* in the main paper. Recall this property as follow.

Property 2 *Given $\mathbf{n} \in \mathbb{R}^d$ with $\mathbf{n}^T\mathbf{n} = 1$, which defines a hyperplane, and a multivariate random variable $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we have $\mathrm{P}(|\mathbf{n}^T\mathbf{z}| \leq 2\alpha\sqrt{\frac{d}{d-2}}) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$ for any $\alpha \geq 1$ and $d \geq 4$. Here $\mathrm{P}(\cdot)$ stands for probability and c is a fixed positive constant.*

Proof.

Without loss of generality, we fix $\mathbf{n}$ to be the first coordinate vector. Accordingly, it suffices to prove that $\mathrm{P}(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$, where z_1 denotes the first entry of $\mathbf{z}$.

As shown in Fig.12, let H denote the set

$$\{\mathbf{z} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}_d) : ||\mathbf{z}||_2 \leq 2\sqrt{d}, |z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}\},$$

where $||\cdot||_2$ stands for the l_2 norm. Obviously, we have $\mathrm{P}(H) \leq \mathrm{P}(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}})$. Now, we will show $\mathrm{P}(H) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$

Considering the random variable $R = ||\mathbf{z}||_2$, with cumulative distribution function $F(R \leq r)$ and density function $f(r)$, we have

$$\begin{aligned}\mathrm{P}(H) &= \mathrm{P}(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}|R \leq 2\sqrt{d})\mathrm{P}(R \leq 2\sqrt{d}) \\ &= \int_0^{2\sqrt{d}} \mathrm{P}(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}|R = r)f(r)dr.\end{aligned}$$

According to *Theorem 1* below, when $r \leq 2\sqrt{d}$, we have

$$\begin{aligned}
\mathrm{P}(H) &= \int_0^{2\sqrt{d}} \mathrm{P}(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}} | R = r) f(r) dr \\
&= \int_0^{2\sqrt{d}} \mathrm{P}(|z_1| \leq \frac{2\sqrt{d}}{r} \frac{\alpha}{\sqrt{d-2}} | R = 1) f(r) dr \\
&\geq \int_0^{2\sqrt{d}} \mathrm{P}(|z_1| \leq \frac{\alpha}{\sqrt{d-2}} | R = 1) f(r) dr \\
&\geq \int_0^{2\sqrt{d}} (1 - \frac{2}{\alpha} e^{-\alpha^2/2}) f(r) dr \\
&= (1 - \frac{2}{\alpha} e^{-\alpha^2/2}) \int_0^{2\sqrt{d}} f(r) dr \\
&= (1 - \frac{2}{\alpha} e^{-\alpha^2/2}) \mathrm{P}(0 \leq R \leq 2\sqrt{d}).
\end{aligned}$$

Then, according to *Theorem 2* below, by setting $\beta = \sqrt{d}$, we have

$$\begin{aligned}
\mathrm{P}(H) &= (1 - \frac{2}{\alpha} e^{-\alpha^2/2}) \mathrm{P}(0 \leq R \leq 2\sqrt{d}) \\
&\geq (1 - \frac{2}{\alpha} e^{-\alpha^2/2})(1 - 3e^{-cd}).
\end{aligned}$$

Q.E.D.

Theorem 1 *Given a unit spherical* $\{\mathbf{z} \in \mathbb{R}^d : ||\mathbf{z}||_2 = 1\}$, *we have* $\mathrm{P}(|z_1| \leq \frac{\alpha}{\sqrt{d-2}}) \geq 1 - \frac{2}{\alpha} e^{-\alpha^2/2}$ *for any* $\alpha \geq 1$ *and* $d \geq 4$.

Proof.

By symmetry, we just prove the case where $z_1 \geq 0$. Also, we only consider about the case where $\frac{\alpha}{\sqrt{d-2}} \leq 1$.

Let U denote the set $\{\mathbf{z} \in \mathbb{R}^d : ||\mathbf{z}||_2 = 1, z_1 \geq \frac{\alpha}{\sqrt{d-2}}\}$, and K denote the set $\{\mathbf{z} \in \mathbb{R}^d : ||\mathbf{z}||_2 = 1, z_1 \geq 0\}$. It suffices to prove that the surface of U area and the surface of K area in Fig.13 satisfy

$$\frac{surf(U)}{surf(K)} \leq \frac{2}{\alpha} e^{-\alpha^2/2},$$

where $surf(\cdot)$ stands for the surface area of a high dimensional geometry. Let $A(d)$ denote the surface area of a d-dimensional unit-radius ball. Then, we have

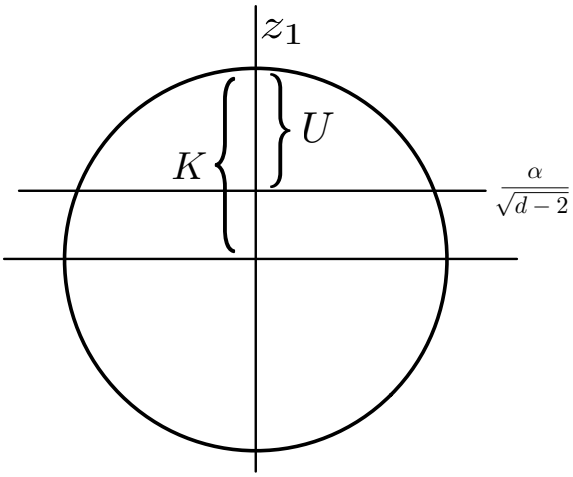

Figure 13: Diagram for *Theorem 1*.

$$\begin{aligned}
surf(U) &= \int_{\frac{\alpha}{\sqrt{d-2}}}^1 (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1 \\
&\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^1 e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1 \\
&\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^1 \frac{z_1\sqrt{d-2}}{\alpha} e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1 \\
&\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^\infty \frac{z_1\sqrt{d-2}}{\alpha} e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1 \\
&= \frac{A(d-1)}{\alpha\sqrt{d-2}} e^{-\alpha^2/2}.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
surf(K) &= \int_0^1 (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1 \\
&\geq \int_0^{\frac{1}{\sqrt{d-2}}} (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1 \\
&\geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2})^{\frac{d-2}{2}} A(d-1).
\end{aligned}$$

Considering the fact that $(1-x)^a \geq 1 - ax$ for any $a \geq 1$ and $0 \leq x \leq 1$, we have

$$\begin{aligned}
surf(K) &\geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2})^{\frac{d-2}{2}} A(d-1) \\
&\geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2} \frac{d-2}{2}) A(d-1) \\
&= \frac{A(d-1)}{2\sqrt{d-2}}.
\end{aligned}$$

Accordingly,

$$\frac{surf(U)}{surf(K)} \leq \frac{\frac{A(d-1)}{\alpha\sqrt{d-2}}e^{-\alpha^2/2}}{\frac{A(d-1)}{2\sqrt{d-2}}} = \frac{2}{\alpha}e^{-\alpha^2/2}.$$

Q.E.D.

Theorem 2 (Gaussian Annulus Theorem [19]) *For a d-dimensional spherical Gaussian with unit variance in each direction, for any $\beta \leq \sqrt{d}$, all but at most $3e^{-c\beta^2}$ of the probability mass lies within the annulus $\sqrt{d} - \beta \leq ||\mathbf{z}||_2 \leq \sqrt{d} + \beta$, where c is a fixed positive constant.*

That is to say, given $\mathbf{z} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}_d)$, $\beta \leq \sqrt{d}$, and a constant $c > 0$, we have

$$\mathrm{P}(\sqrt{d} - \beta \leq ||\mathbf{z}||_2 \leq \sqrt{d} + \beta) \geq (1 - 3e^{-c\beta^2}).$$

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 2

[2] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *ICLR*, 2018. 2

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *CVPR*, 2018. 2

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Visualizing and understanding generative adversarial networks. In *ICLR*, 2019. 2

[5] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, 2019. 2, 4

[6] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 2

[7] Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018. 2

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 2, 3

[9] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick van der Smagt. Metrics for deep generative models. In *AISTAT*, 2018. 2

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016. 2

[11] Chris Donahue, Akshay Balsubramani, Julian McAuley, and Zachary C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. In *ICLR*, 2018. 2

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017. 2, 4

[13] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017. 2, 4

[14] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019. 2

[15] 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. 1, 2

[16] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020. 2

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 9

[19] John Hopcroft and Ravi Kannan. *Foundations of Data Science*. 2014. 11

[20] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2020. 2

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 1, 2, 3, 4, 6, 8

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 3, 4, 6, 7, 8

[23] Line Kuhnel, Tom Fletcher, Sarang Joshi, and Stefan Sommer. Latent space non-linear statistics. *arXiv preprint arXiv:1805.07632*, 2018. 2

[24] Samuli Laine. Feature-based metrics for exploring the latent space of generative models. In *ICLR Workshop*, 2018. 2

[25] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *NeurIPS*, 2017. 2

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 9

[27] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *NeurIPS*, 2018. 2, 4, 7

[28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 2, 3

[29] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 2

[30] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NeurIPS Workshop*, 2016. 2

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 3

[32] Hang Shao, Abhishek Kumar, and P Thomas Fletcher. The riemannian geometry of deep generative models. In *CVPR Workshop*, 2018. 2

[33] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *CVPR*, 2018. 2

[34] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. Facefeat-gan: a two-stage approach for identity-preserving face synthesis. *arXiv preprint arXiv:1812.01288*, 2018. 2

[35] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017. 2

[36] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *CVPR*, 2017. 2

[37] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *ECCV*, 2018. 2

[38] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *arXiv preprint arXiv:1911.09267*, 2019. 2

[39] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017. 2

[40] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 2

[41] Jiapeng Zhu, Deli Zhao, and Bo Zhang. Lia: Latently invertible autoencoder with adversarial learning. *arXiv preprint arXiv:1906.08090*, 2019. 2, 4, 8

[42] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 2, 4, 7