InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs

Yujun Shen, Ceyuan Yang, Xiaoou Tang, *Fellow, IEEE,* and Bolei Zhou, *Member, IEEE*

Abstract—Although Generative Adversarial Networks (GANs) have made significant progress in face synthesis, there lacks enough understanding of what GANs have learned in the latent representation to map a random code to a photo-realistic image. In this work, we propose a framework called InterFaceGAN to interpret the disentangled face representation learned by the state-of-the-art GAN models and study the properties of the facial semantics encoded in the latent space. We first find that GANs learn various semantics in some linear subspaces of the latent space. After identifying these subspaces, we can realistically manipulate the corresponding facial attributes without retraining the model. We then conduct a detailed study on the correlation between different semantics and manage to better disentangle them via subspace projection, resulting in more precise control of the attribute manipulation. Besides manipulating the gender, age, expression, and presence of eyeglasses, we can even alter the face pose and fix the artifacts accidentally made by GANs. Furthermore, we perform an in-depth face identity analysis and a layer-wise analysis to evaluate the editing results quantitatively. Finally, we apply our approach to real face editing by employing GAN inversion approaches and explicitly training feed-forward models based on the synthetic data established by InterFaceGAN. Extensive experimental results suggest that learning to synthesize faces spontaneously brings a disentangled and controllable face representation.

Index Terms—Generative adversarial network, face editing, interpretability, explainable artificial intelligence, disentanglement.

✦

1 INTRODUCTION

RECENT years have witnessed the great success of Generative Adversarial Networks (GANs) [2] in high-fidelity face synthesis [1], [3], [4]. Based on adversarial training, GANs learn to map a random distribution to the real data observation and then produce photo-realistic images from randomly sampled latent codes.

Despite the appealing synthesis quality, it remains much less explored about what knowledge GANs learn in the latent representation and how we can reuse such knowledge to control the generation process. For example, given a latent code, how does GAN determine the attributes of the output face, *e.g.*, an elder man or a young woman? How are different attributes organized in the latent space? Can we manipulate the attributes of the synthesized face as we want? How does the attribute manipulation affect the face identity? Can we apply a well-trained GAN model for real image editing?

To answer these questions, we propose a novel framework, termed as *InterFaceGAN*, to *Inter*pret the *Face* representation learned by *GAN*s. We employ some off-the-shelf classifiers to predict semantic scores for synthesized images. In this way, we can bridge the latent space and the semantic space and further utilize such a connection for representation analysis. In particular, we analyze how an individual semantic is encoded in the latent space both theoretically and empirically. It turns out that a true-or-false attribute aligns with a linear subspace of the latent space. Based on this discovery, we study the entanglement between

different semantics emerging in the latent representation and manage to disentangle them via subspace projection.

After identifying the latent semantics, InterFaceGAN proposes a simple yet effective pipeline for face editing. By linearly varying the latent code, we can manipulate the gender, age, expression, presence of eyeglasses, and even face pose of the synthesized image, as shown in Figure 1 (a). Moreover, thanks to our disentanglement analysis, we propose conditional manipulation to alter one attribute without affecting others, as shown in Figure 1 (b). More importantly, InterFaceGAN controls the face semantics by reusing the GAN knowledge *without* retraining the model.

To better interpret the representation learned by GANs, we conduct a thorough analysis of the editing results made by InterFaceGAN. First, we compare the semantic scores before and after the manipulation to quantitatively evaluate the identified semantics. Then, we make a layer-wise analysis on StyleGAN, whose generator is with per-layer stochasticity [3], to explore how semantics originate from the latent representation layer by layer. Finally, considering the importance of the identity information for faces, we make in-depth identity analysis to see how identity is preserved in the manipulation process as well as how identity is sensitive to different facial attributes.

Applying a GAN model to real image editing is another important issue since GANs commonly lack the inference ability. In this work, we apply two approaches to extend InterFaceGAN for real face manipulation. One works with GAN inversion, which can invert a target image back to a latent code. The other uses InterFaceGAN to build a dataset that contains the pairs of synthetic images before and after manipulation and then train a pixel-to-pixel model [5] on this synthetic dataset. We compare these approaches and evaluate their strengths and weaknesses.

• *Y. Shen, C. Yang, X. Tang, and B. Zhou are with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong SAR, China.*
E-mail: {sy116, yc019, xtang, bzhou}@ie.cuhk.edu.hk

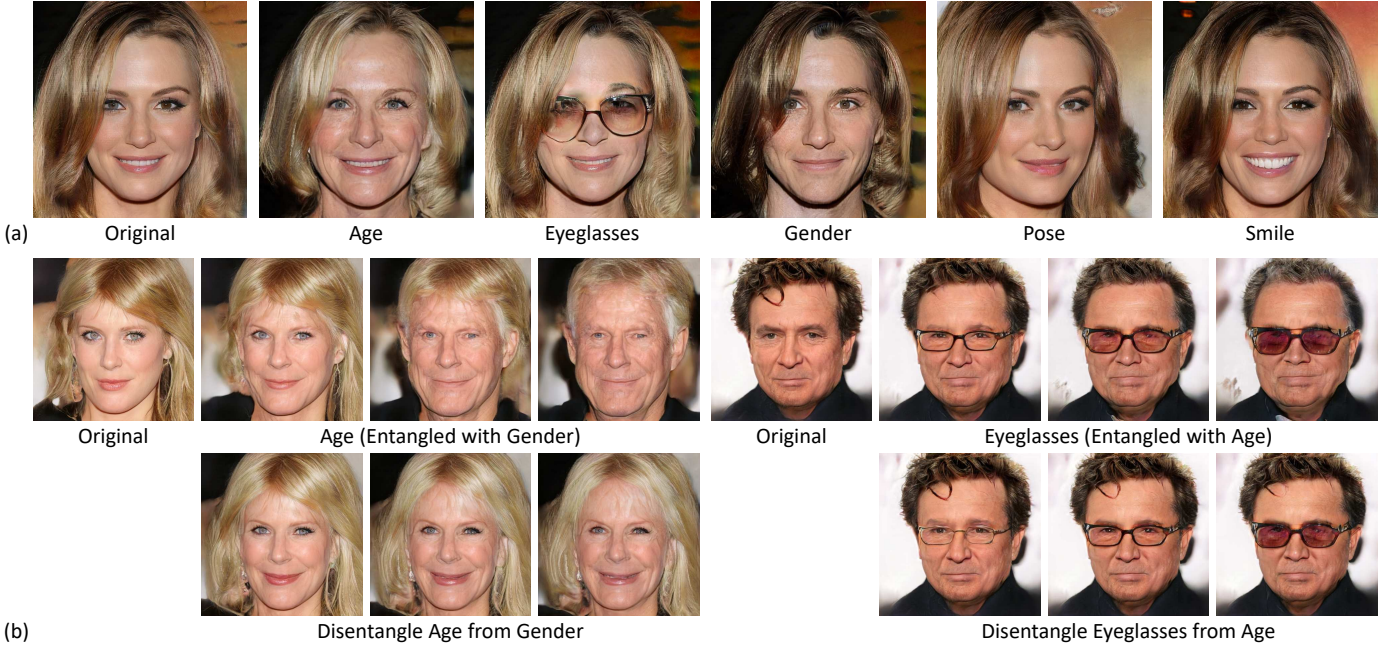*Code and models are available at https://genforce.github.io/interfacegan/.*

Fig. 1. (a) **Manipulating various facial attributes** through varying the latent codes of a well-trained GAN model. (b) **Conditional manipulation** results using InterFaceGAN, where we can better disentangle the correlated attributes (top row) and achieve more precise control of the facial attributes (bottom row). All results are synthesized by PGGAN [1].

The preliminary result of this work is published at [6]. Compared to the conference paper, we include the following new contents: (i) a detailed analysis of the face representation learned by StyleGAN [3] and its comparison to PGGAN [1]; (ii) a comparison between the entanglement of identified latent semantics and the attribute distribution of training data, which sheds light on how GANs learn to encode various semantics in the training process; (iii) quantitative evaluation of the editing results achieved by InterFaceGAN; (iv) layer-wise analysis of the per-layer representation learned by StyleGAN [3]; (v) identity analysis of the manipulated images; (vi) a new method for real face editing, which is to train feed-forward models on the synthetic data collected by InterFaceGAN.

2 RELATED WORK

Generative Adversarial Networks. Due to the great potential of GAN [2] in producing photo-realistic images, it has been widely applied to image editing [7], [8], super-resolution [9], [10], image inpainting [11], [12], video synthesis [13], [14], *etc*. Many attempts have been made to improve the synthesis quality and training stability of GANs [1], [3], [4], [15], [16], [17], [18], [19], [20]. Despite this tremendous success, little work has been done on understanding how GANs learn to connect the latent representation with the semantics in the real visual world.

Study on Latent Space of GANs. Latent space of GANs is generally treated as Riemannian manifold [21], [22], [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 [24], [25]. Bojanowski *et al.* [26] optimized the generator and latent code simultaneously to learn a better representation. However, the studies on how a well-trained GAN can encode different semantics

in the latent space and reuse this semantic knowledge to control the generation process are still missing. Bau *et al.* [27] found 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 synthesis. Some work [28], [29] observed the vector arithmetic property in the latent space. This work provides a detailed analysis of how semantics are encoded in the face representation learned by GANs from the perspective of a single semantic and the disentanglement of multiple semantics. Some concurrent work also explores the latent semantics in GANs: Goetschalckx *et al.* [30] improves the memorability of the output image. Jahanian *et al.* [31] studies the steerability of GANs concerning camera motion and image color tone. Yang *et al.* [32] observes the semantic hierarchy emerging in the scene synthesis models. Differently, we focus on interpreting the face representation by theoretically and empirically studying how various semantics originate from and are organized in the latent space. We further extend our method to real image editing.

Semantic Face Editing with GANs. Semantic face editing aims at manipulating the facial attributes of a given image. Compared to unconditional GANs, which can generate images arbitrarily, semantic editing expects the model to change the target attribute yet maintain other information of the input face. To achieve this goal, existing methods require carefully designed loss functions [33], [34], [35], introduction of additional attribute labels [7], [36], [37], [38], [39], or special architectures [40], [41] to train new models. Unlike previous learning-based methods, this work explores the interpretable semantics inside the latent space of *fixed* GAN models. By reusing the semantic knowledge spontaneously learned by GANs, we can unleash its manipulation capability and *turn unconstrained GANs to controllable GANs* through varying the latent code.

GAN Inversion. The generator in GANs typically takes a latent code as the input and outputs a synthesized image. Hence, it leaves no space for the inference on real images. To solve this problem, a common practice is to get the reverse mapping from the image space to the latent space, which is also known as GAN Inversion [8], [42], [43], [44], [45]. Prior work either performed instance-level optimization [46], [47], [48] or explicitly learned an encoder to reverse the generator [49], [50], [51]. Some methods combined these two ideas by using the encoder to produce a good starting point for optimization [52], [53], [54]. There are also some works changing the optimization objects by using multiple codes to reconstruct a single image [55] or optimizing the model weight together with the latent code [56]. Being orthogonal to those approaches, our work interprets the latent representation and utilizes GAN inversion as a tool to reuse GAN knowledge for real image editing.

Image-to-Image Translation. Image-to-Image translation learns a deterministic model to transfer images from one domain to another. It is widely used for real image editing considering its fast inference speed [5], [57], [58], [59], [60], [61]. Some attempts increased the diversity of the translated images by introducing stochasticity [62], [63] or translating images among multiple domains [64], [65]. However, all these models rely on paired data or domain labels, which are not easy to obtain. In this work, we manage to leverage the semantics learned by GANs to create unlimited synthetic data pairs. We can then apply the knowledge encoded in the latent representation to feed-forward real image editing by training image-to-image translation networks with such synthetic data.

3 FRAMEWORK OF INTERFACEGAN

In this section, we first provide a rigorous analysis of the properties of the semantics emerging in the latent representation learned by GANs and then construct a pipeline of utilizing the identified semantics for face editing.

3.1 Semantics in Latent Space

Given a well-trained GAN model, the generator can be viewed as a deterministic function $g : \mathcal{Z} \rightarrow \mathcal{X}$. 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 [1], [3], [18], [20]. $\mathcal{X}$ 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 $f_S : \mathcal{X} \rightarrow \mathcal{S}$, 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 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 synthesis changes continuously [3], [20], [28]. It implicitly means that the semantics contained in the image also change gradually. According to **Property 1**, the interpolation from $\mathbf{z}_1$ to $\mathbf{z}_2$ defines a direction in $\mathcal{Z}$, which further defines a hyperplane. We therefore make an assumption[1] that for any binary semantic (*e.g.*, male

---

1. This assumption is empirically demonstrated in Section 4.

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 one side of the hyperplane yet turns into the opposite when across the boundary.

Given a hyperplane with 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. Moreover, it is just when the "distance" changes its numerical sign that the semantic attribute reverses. We therefore expect these two items 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.*

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.*[2]

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 compute the mean and covariance matrix of the semantic scores $\mathbf{s}$ as

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

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

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 entangle with each other. $\mathbf{n}_i^T \mathbf{n}_j$ can be used to measure the entanglement between the i-th and j-th semantics to some extent.

---

2. 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**.

3.2 Manipulation in Latent Space

In this part, we introduce how to use the semantics found in the 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.

Conditional Manipulation. When there is more than one attribute, editing one may affect another since some semantics can be entangled. 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 make different vectors orthogonal. As shown in Figure 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". If there are multiple attributes to be conditioned on, we subtract the projection from the primal direction onto the plane constructed by all conditioned directions. Note that the proposed conditional manipulation requires each semantic subspace to be independent of others. In other words, if two semantics share the same subspace, it is hard to isolate one from the other via subspace projection, but it rarely happens in a high-dimensional space.

Real Image Manipulation. InterFaceGAN enables semantic editing from the latent space of a *fixed* GAN model. To manipulate real images, we should infer the best latent code that can reconstruct the target image, *i.e.*, GAN inversion. For this purpose, both optimization-based [54] and learning-based [51] approaches can be used. We thoroughly evaluate their strengths and weaknesses in Section 6. We also use InterFaceGAN to prepare synthetic data pairs and then train image-to-image translation models [5] to achieve real face editing, with the results shown in Section 6.

3.3 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 easily plug in more attributes as long as the attribute classifier is available.

To better predict these attributes from synthesized images, we train an auxiliary attribute prediction model using the annotations from the CelebA dataset [66] with ResNet-50 network [67]. This model is trained with multi-task losses to simultaneously predict smile, age, gender, eyeglasses, as well as the 5-point facial landmarks (*i.e.*, left eye, right eye, nose, left corner of mouth, right corner of mouth). Here, the facial landmarks are used to compute yaw pose, which is also treated as a binary attribute (*i.e.*, left *v.s.* right). Besides the landmarks, all other attributes are learned as a 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
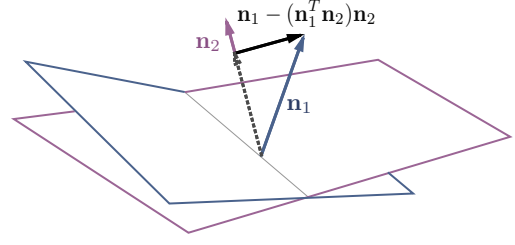

Fig. 2. Illustration of the **conditional manipulation via subspace projection**. The projection of $\mathbf{n}_1$ onto $\mathbf{n}_2$ is subtracted from $\mathbf{n}_1$, resulting in a new direction $\mathbf{n}_1 - (\mathbf{n}_1^T\mathbf{n}_2)\mathbf{n}_2$.

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 from the latent space. There are two main reasons for preparing such a large-scale dataset: (i) to eliminate the randomness caused by sampling and make sure the distribution of the sampled code is as expected, and (ii) to get enough wearing-glasses samples, which are rare in CelebA-HQ dataset [1].

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. We sort the corresponding scores for each attribute and choose $10K$ samples with the highest scores and $10K$ with the lowest ones as candidates. The reason in doing so is that the prediction model is not perfect and may produce wrong predictions 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. The remaining 30% samples are used for verifying how the linear classifier behaves. Here, for SVM training, the inputs are the $512d$ latent codes and the predicted attribute scores are treated as binary labels. Here, note that learning a "bias" term in the SVM classifier or not barely affects the learned direction. That is because we only choose the samples with the highest confidence level for training and they can be easily separated. We have tried to add the "bias" term and the learned bias turns out to be small.

4 INTERPRETING FACE REPRESENTATION

In this section, we apply InterFaceGAN to interpret the face representation learned by state-of-the-art GAN models, *i.e.*, PGGAN [1] and StyleGAN [3], both of which can produce high-quality faces with 1024×1024 resolution. PGGAN is a representative of the traditional generator where the latent code is only fed into the very first convolutional layer. By contrast, StyleGAN proposed a style-based generator, which first maps the latent code from latent space $\mathcal{Z}$ to a disentangled latent space $\mathcal{W}$ before applying it for the generation. In addition, the disentangled latent code is fed to all convolutional layers.[3]

3. When feeding $\mathbf{w}$ to each convolutional layer, StyleGAN supports truncation to ensure the synthesis quality. We use truncation 0.7 (1 means no truncation) for the first eight layers in all experiments.
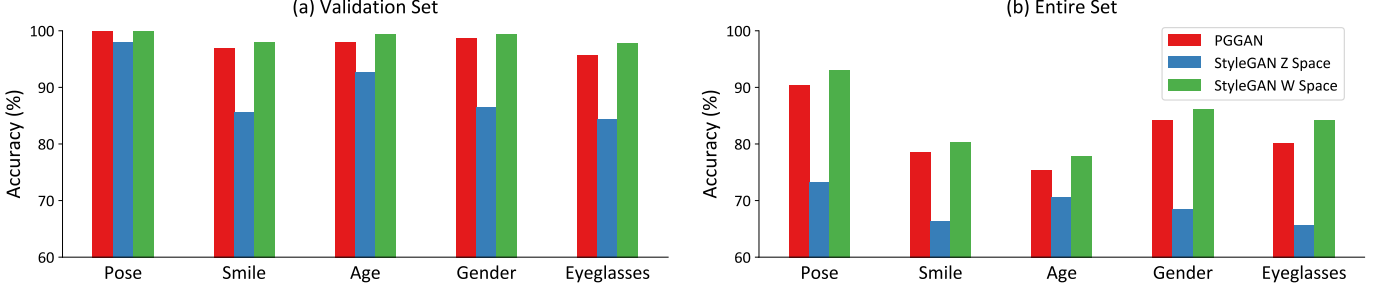
Fig. 3. **Classification accuracy** (%) on the latent separation boundaries of PGGAN [1] and StyleGAN [3] with respect to different attributes.

4.1 Separability of Latent Space

Section 3.1 introduces our assumption that for any binary attribute, there exists a hyperplane in the latent space such that all samples from the same side are with the same attribute. In this part, we first evaluate this assumption.

4.1.1 PGGAN

For each attribute, we will get a latent boundary after the training of the linear SVM classifier. We evaluate the classification performance on the validation set ($3K$ positive testing samples and $3K$ negative testing samples) and the entire set (remaining $480K$ samples besides the $20K$ candidates with high confidence level). Figure 3 shows the results. We find that all linear boundaries of PGGAN achieve over 95% accuracy on the validation set and over 75% on the entire set, suggesting that for a binary attribute, there indeed exists a linear hyperplane in the latent space that can well separate the data into two groups.

We also visualize some samples in Figure 4 by ranking them by the "distance" to the decision boundary. Those extreme cases (top and bottom rows) are very unlikely to be directly sampled, instead constructed by moving a latent code towards the normal direction "infinitely". Here, to achieve "zero" manipulation step, we randomly sample latent codes within the separation hyperplane, while to achieve "infinite" manipulation step, we directly use the normal vector **n** (or $-$**n**) as the sampled code. From Figure 4, we can tell that the positive samples and negative samples are distinguishable to each other with respect to the corresponding attribute. This further demonstrates that the latent space is linearly separable and InterFaceGAN can find the separation hyperplane successfully.

4.1.2 StyleGAN

Compared to PGGAN, StyleGAN employs two latent spaces, which are the native latent space $\mathcal{Z}$ and the mapped latent space $\mathcal{W}$. We analyze the separability of both spaces, and the results are shown in Figure 3. Note that $\mathcal{W}$ space is not subject to normal distribution like $\mathcal{Z}$ space, but we can conduct a similar analysis on $\mathcal{W}$ space, demonstrating the generalization ability of InterFaceGAN.

Figure 3 shows three observations: (i) Boundaries from both $\mathcal{Z}$ space and $\mathcal{W}$ space separate the validation set well. Even though the performances of $\mathcal{Z}$ boundaries on the entire set are not satisfying, it may be caused by the inaccurate attribute prediction on the ambiguous samples. (ii) $\mathcal{W}$ space shows much stronger separability than $\mathcal{Z}$ space. That is because $\mathbf{w} \in \mathcal{W}$, instead of $\mathbf{z} \in \mathcal{Z}$, is the code
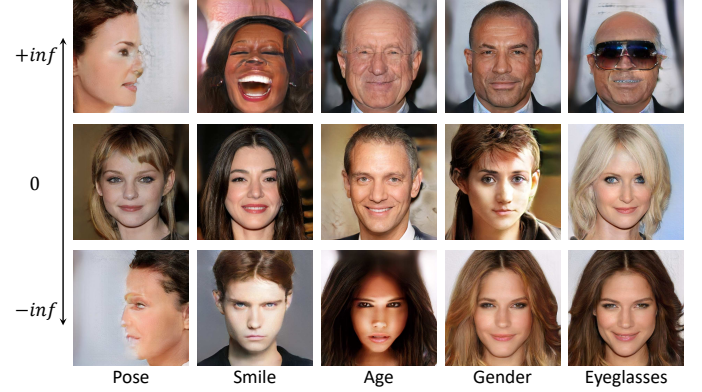

Fig. 4. Synthesized samples by PGGAN [1] with the distance near to (middle row) and extremely far away from (top and bottom rows) the separation boundary. Each column corresponds to a particular attribute.

finally fed into the generator. Accordingly, the generator tends to learn various semantics based on $\mathcal{W}$ space. (iii) The accuracy of StyleGAN $\mathcal{W}$ space is higher than the PGGAN $\mathcal{Z}$ space. The reason is that the semantic attributes may not be normally distributed in real data. Compared to $\mathcal{Z}$ space, which is subject to the normal distribution, $\mathcal{W}$ space has no constraints and hence is able to better fit the underlying real distribution. This is consistent with the design of $\mathcal{W}$ space in StyleGAN [3].

4.2 Semantics in Latent Space for Face Manipulation

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

4.2.1 PGGAN

Manipulating Single Attribute. Figure 5 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 change. Furthermore, although there lacks enough data with extreme poses in the training set, GAN can hallucinate how profile faces should look. The same situation also appears on the eyeglasses attribute. We can manually create many 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 in the latent space.

Fig. 5. **Single attribute manipulation** results with PGGAN [1]. The top left shows the same person under gradually changed poses. The remaining samples correspond to the results of manipulating four different attributes. The central one is the original synthesis for each triplet, while the left and right stand for the results by moving the latent code towards the negative and positive directions respectively.
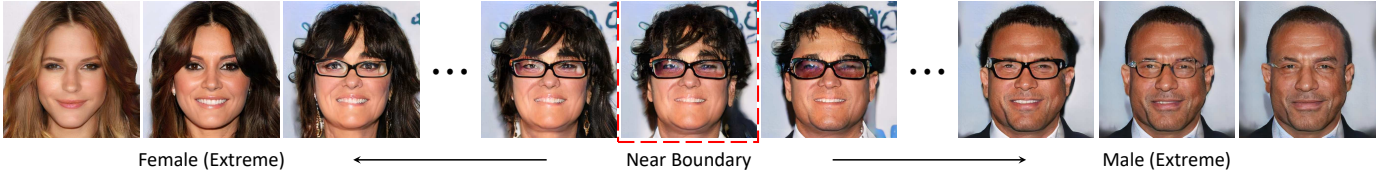

Fig. 6. Illustration of the **distance effect** by taking gender manipulation with PGGAN [1] as an example. The image in the red dashed box stands for the original synthesis. Our approach performs well when the latent code locates close to the boundary. However, when the distance keeps increasing, the synthesized images are no longer like the same person.

Distance Effect of Semantic Subspace. When manipulating the latent code, we observe a noticeable distance effect: the samples will suffer from severe changes in appearance if being moved too far from the boundary, and finally tend to become the extreme cases shown in Figure 4. Figure 6 illustrates this phenomenon by taking gender editing as an instance. Near-boundary manipulation works well. When samples go beyond a certain region, however, the editing results are no longer like the original face anymore. However, this effect does not affect our understanding of the disentangled semantics in the latent space. That is because such extreme samples are unlikely to be directly drawn from a standard normal distribution, which is pointed out in *Property 2* in Section 3.1. Instead, they are constructed manually by keeping moving a normally sampled latent code along a certain direction.

Fixing Artifacts. We further apply our approach to fix the artifacts that sometimes occur in the synthesis. We manually labelled $4K$ deficient synthesis and then trained a linear SVM to find the separation hyperplane. We surprisingly find that GAN also encodes such quality information in the latent space. Based on this discovery, we manage to correct some mistakes GAN has made in the generation process by moving the latent code towards the positive "quality" direction, as shown in Figure 7.

4.2.2 StyleGAN

We further apply InterFaceGAN to StyleGAN by manipulating the latent codes in both $\mathcal{Z}$ space and $\mathcal{W}$ space. Figure 8 shows the following observations: (i) InterFaceGAN works well on the style-based generator. We manage to edit the attributes by altering the latent code in either $\mathcal{Z}$ space or $\mathcal{W}$ space. (ii) By learning from a more diverse dataset, FF-HQ [3], StyleGAN learns various semantics more thoroughly. For example, StyleGAN can produce high-quality profile faces (first example) and generate children

Fig. 7. Examples on **fixing the artifacts** that PGGAN [1] made. The first row shows several bad synthesis results, while the following two rows present the gradually corrected synthesis by moving the latent codes towards the positive "quality" direction.

when making people younger (second example). This is beyond the ability of PGGAN, which is trained on CelebA-HQ dataset [1]. (iii) Some attributes correlate with each other. For example, people are wearing eyeglasses when turning old (second example). A more detailed analysis of this phenomenon will be discussed in Section 4.3. (iv) $\mathcal{W}$ space shows better performance than $\mathcal{Z}$ space, especially for long-distance manipulation. In other words, when the latent code locates near the separation boundary, manipulations in $\mathcal{Z}$ space and $\mathcal{W}$ space have a similar effect. However, when the latent code goes further from the boundary, manipulating one attribute in $\mathcal{Z}$ space might affect another. Taking pose editing (first example) as an instance, the skin color of the target person varies when moving the latent code along the pose direction. By contrast, $\mathcal{W}$ space shows stronger robustness.
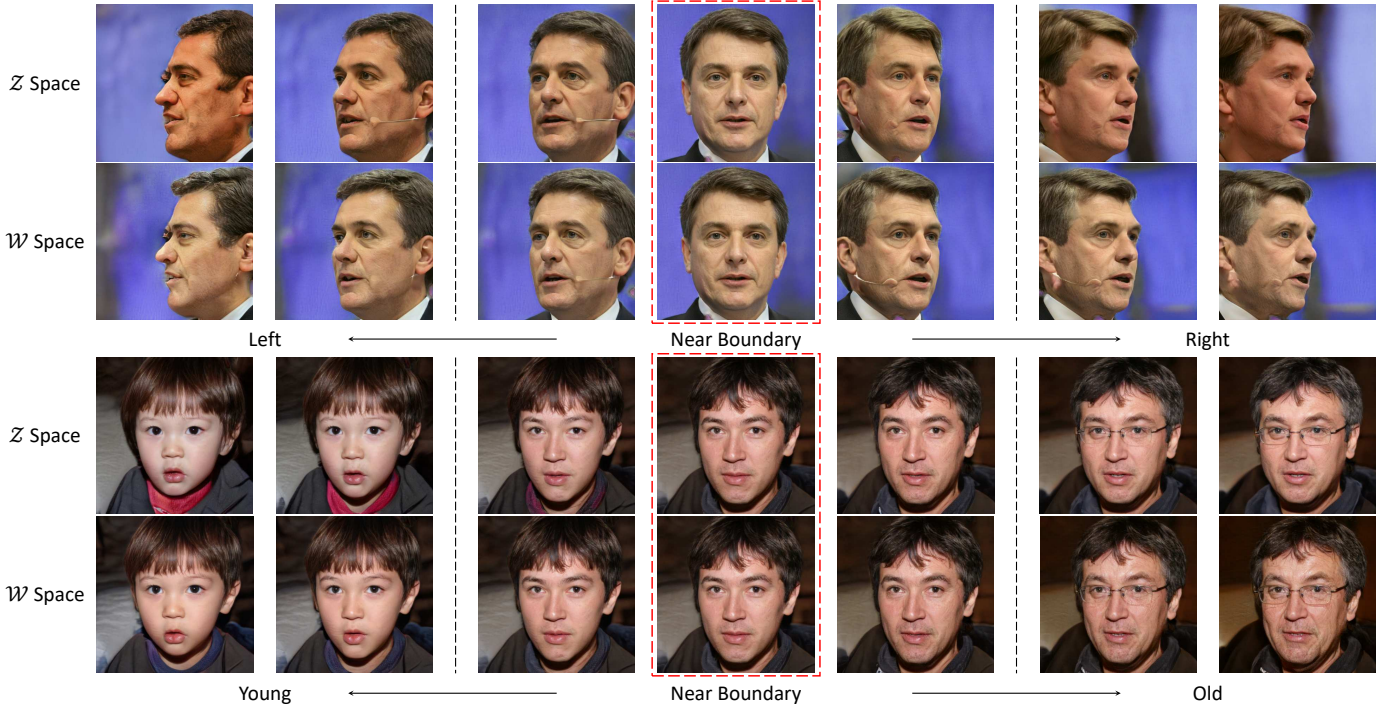
Fig. 8. **Attribute editing** results on StyleGAN [3]. For two attributes pose and age, the top row shows the manipulation results with respect to $\mathcal{Z}$ space, whilst the bottom row corresponds to $\mathcal{W}$ space. Images in red dashed boxes represent the original synthesis. Images between two black dashed lines stand for near-boundary manipulation, and the other images stand for long-distance manipulation.

4.3 Disentanglement Analysis and Conditional Manipulation

This section discusses the disentanglement between different semantics encoded in the latent representation and evaluates the conditional manipulation approach.

4.3.1 Disentanglement Measurement

Besides how one semantic can be well encoded in the latent space, we also study the correlation between multiple semantics and examine the way to decouple correlated attributes. In particular, we use the following metrics for analysis.

1) *Attribute correlation of real data*. We use the prepared predictors to predict the attribute scores of real data on which the GAN model is trained. Then we compute the correlation coefficient ρ between any two attributes with $\rho_{A_1,A_2} = \frac{Cov(A_1,A_2)}{\sigma_{A_1}\sigma_{A_2}}$. Here, A_1 and A_2 represent two random variables with respect to these two attributes. $Cov(\cdot,\cdot)$ and σ denote covariance and standard deviation respectively.

2) *Attribute correlation of synthesized data*. We also compute the attribute correlation score on the $500K$ synthesized data. By comparing this score to that of the real data, we can get clues on how GANs learn to encode such semantic knowledge in the latent representation.

3) *Semantic boundary correlation*. Given any two semantics, with the latent boundaries $\mathbf{n}_1$ and $\mathbf{n}_2$, we compute the cosine similarity between these two directions with $\cos(\mathbf{n}_1, \mathbf{n}_2) = \mathbf{n}_1^T \mathbf{n}_2$. Here, $\mathbf{n}_1$ and $\mathbf{n}_2$ are both unit vectors. This metric is used to evaluate how the above attribute correlation is reflected in our InterFaceGAN framework.

4.3.2 PGGAN

Disentanglement Analysis. Table 1 reports the correlation metrics of PGGAN model trained on CelebA-HQ dataset [1]. By comparing the attribution correlation of real data (Table 1 (a)) and that of synthesized data (Table 1 (b)), we can tell that they are very close to each other. For example, "pose" and "smile" are almost independent to other attributes, while "gender", "age", and "eyeglasses" are highly correlated with each other from both real data and synthesized data. For example, elder men are more likely to wear eyeglasses. This implies that GANs learn the underlying semantic distribution from real observation when trained to synthesize images. Then, by comparing such attribution correlation with the boundary correlation (Table 1 (c)), we also find that they behave similarly. This suggests that InterFaceGAN is able to not only accurately identify the semantics encoded in that latent representation but also capture the entanglement among them.

Conditional Manipulation. As shown in Table 1 (c), if two boundaries are not orthogonal to each other, modulating the latent code along one direction will affect the other. Hence, we propose conditional manipulation via subspace projection to reduce this entanglement as much as possible. Details are described in Section 3.2. Figure 9 shows the discrepancies between unconditional manipulation and conditional manipulation. Taking the top-left example in Figure 9 as an instance, the results tend to become male when being edited to get old (top row). We address this issue by subtracting its projection onto the gender direction from the age direction, resulting in a new direction. By moving latent codes along this projected direction, we can make sure the gender component is barely affected in the editing process (bottom row). Figure 10 shows a more

TABLE 1
Disentanglement analysis on PGGAN [1].

(a) Training data.

	Pose	Smile	Age	Gender	Glasses
Pose	1.00	-0.01	0.00	0.00	0.01
Smile		1.00	0.03	-0.09	-0.03
Age			1.00	0.20	0.15
Gender				1.00	0.19
Glasses					1.00

(b) Synthesized data.

	Pose	Smile	Age	Gender	Glasses
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
Glasses					1.00

(c) Semantic boundaries.

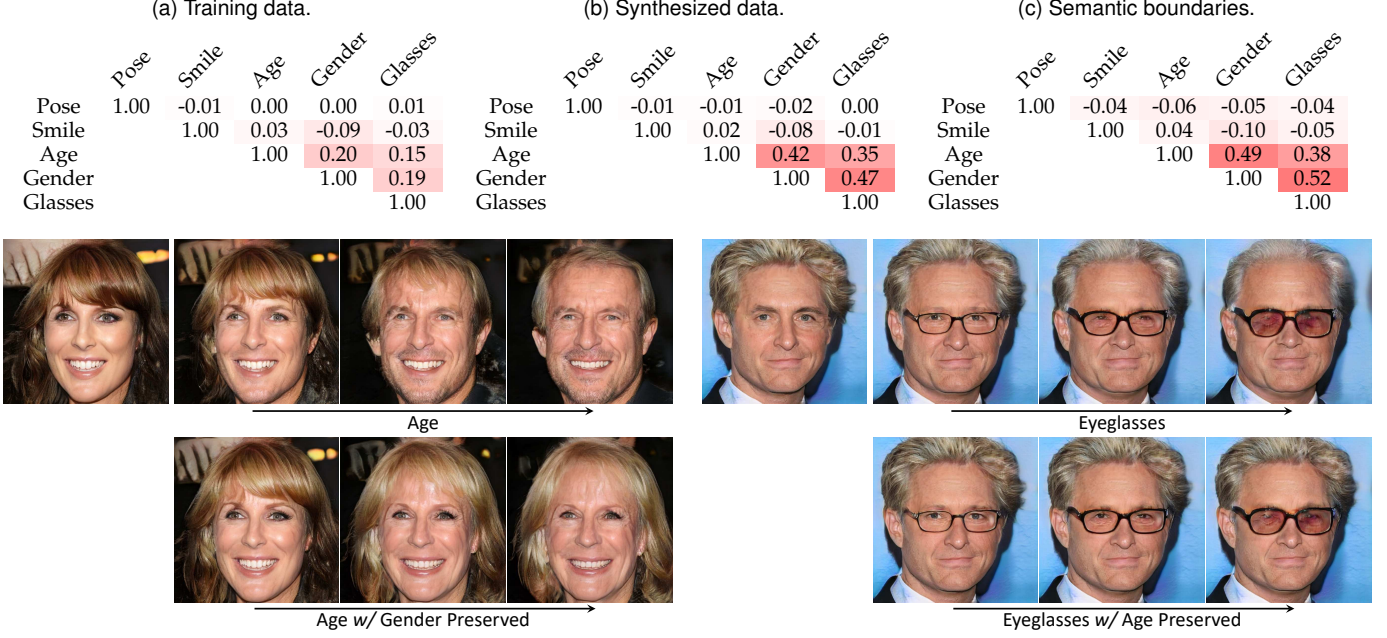|	Pose	Smile	Age	Gender	Glasses
Pose	1.00	-0.04	-0.06	-0.05	-0.04
Smile		1.00	0.04	-0.10	-0.05
Age			1.00	0.49	0.38
Gender				1.00	0.52
Glasses					1.00

Fig. 9. **Conditional manipulation** results using PGGAN [1]. Left: Manipulating age attribute by preserving gender. Right: Manipulating eyeglasses attribute by preserving age. For each example, the top row shows the unconditional editing results while the bottom row shows the conditional manipulation.

complicated case where we perform manipulation with multiple conditions. Taking "eyeglasses" attribute as an example, in the beginning, adding eyeglasses is entangled with changing both age and gender. But we manage to disentangle eyeglasses from age and gender by forcing the eyeglasses direction to be orthogonal to the other two. Note that the hair region of the bottom right result in Figure 10 is mixed with the background. That is because hair affects "gender", but only using "eyeglasses" and "age" as conditions are hard to demonstrate the demarcation between hair and background. This may be improved if we can identify another subspace using a background predictor and use it as an additional condition. Even so, the results in Figure 9 and Figure 10 demonstrate that our proposed conditional approach helps to achieve precise attribute control.

4.3.3 StyleGAN

Disentanglement Analysis. We conduct a similar analysis on the StyleGAN model trained on FF-HQ dataset [3]. As mentioned above, StyleGAN introduces a disentangled latent space $\mathcal{W}$ beyond the original latent space $\mathcal{Z}$. Hence, we analyze the boundary correlation from both of these two spaces. Results are shown in Table 2. Besides the conclusions from PGGAN, we have three more observations. (i) "Smile" and "gender" are not correlated in CelebA-HQ dataset (Table 1 (a)), but entangled in FF-HQ dataset (Table 2 (a)). Such "data bias" may lead to a different latent representation. (ii) $\mathcal{W}$ space (Table 2 (c)) is indeed more disentangled than $\mathcal{Z}$ space (Table 2 (b)), which is consistent with the conclusion from Karras *et al.* [3]. In $\mathcal{W}$ space, almost all boundaries are orthogonal to each other. (iii) The boundary correlation from $\mathcal{W}$ space no longer aligns with the semantic distribution
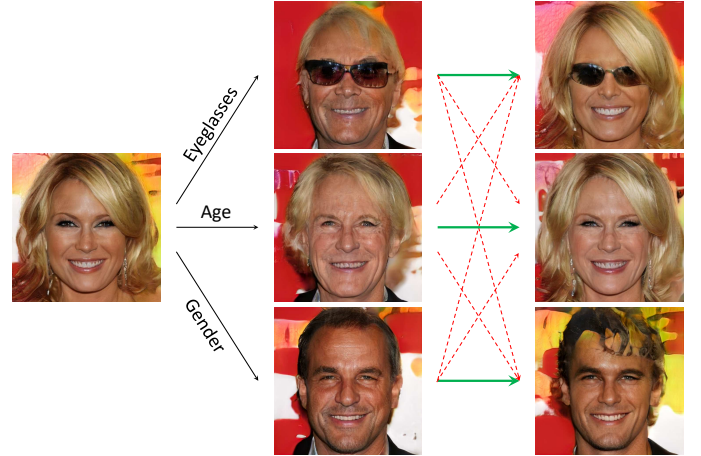

Fig. 10. **Conditional manipulation with more than one conditions** using PGGAN [1]. Left: Original synthesis. Middle: Manipulations along a single boundary. Right: Conditional manipulations. Green arrows indicate the primal direction and red arrows indicate the directions to be conditioned on.

from real data (Table 2 (a)). That may be because $\mathcal{W}$ space is subject to a more complicated distribution than Gaussian distribution and the boundaries found by linear classifiers may not be very accurate. Using a non-linear classifier may solve this problem.

Conditional Manipulation. We also evaluate the proposed conditional manipulation on the style-based generator to verify its generalization ability. In particular, given a sample, we manipulate its attribute from both $\mathcal{Z}$ space and $\mathcal{W}$ space, and then perform conditional manipulation in $\mathcal{Z}$ space. Note that such conditional operation is *not* applicable to $\mathcal{W}$ space. As shown in Table 2 (c), all boundaries are almost orthogonal to each other. As a result, projection barely

TABLE 2
Disentanglement analysis on StyleGAN [3].

(a) Training data.	Pose	Smile	Age	Gender	Glasses
Pose	1.00	-0.02	0.00	-0.02	-0.01
Smile		1.00	0.01	-0.17	-0.03
Age			1.00	0.14	0.21
Gender				1.00	0.20
Glasses					1.00

(b) Semantic boundaries from $\mathcal{Z}$ space.	Pose	Smile	Age	Gender	Glasses
Pose	1.00	-0.03	0.03	-0.01	-0.08
Smile		1.00	-0.28	-0.42	-0.20
Age			1.00	0.33	0.72
Gender				1.00	0.44
Glasses					1.00

(c) Semantic boundaries from $\mathcal{W}$ space.	Pose	Smile	Age	Gender	Glasses
Pose	1.00	0.00	0.02	0.03	-0.03
Smile		1.00	0.03	-0.06	0.02
Age			1.00	0.07	0.05
Gender				1.00	0.00
Glasses					1.00

Fig. 11. **Conditional manipulation** analysis on StyleGAN [3] by taking age manipulation as an example. Images in red dashed boxes represent the original synthesis, while the others show the manipulation process. $\mathcal{W}$ space is more disentangled than $\mathcal{Z}$ space, especially for long-distance manipulation, but the entanglement in $\mathcal{Z}$ space can be diminished by the proposed conditional manipulation.

changes the primal direction. Figure 11 gives an example about the entanglement between "age" and "eyeglasses". In Figure 11, manipulating from $\mathcal{Z}$ space and $\mathcal{W}$ space produces similar results when the latent code still locates near the boundary. For long-distance manipulation, $\mathcal{W}$ space (first row) shows superiority over $\mathcal{Z}$ space (second row), *e.g.*, hair length and face shape do not change in the first row. Even so, "age" and "eyeglasses" are still entangled in both spaces. However, we can use subspace projection to decorrelate "eyeglasses" from "age" in $\mathcal{Z}$ space (third row), resulting in more precise control. This demonstrates the effectiveness of the proposed conditional manipulation approach.

5 QUANTITATIVE ANALYSIS ON MANIPULATION

We have shown the qualitative results in Section 4.2 and Section 4.3 to exhibit the controllable disentangled semantics identified by InterFaceGAN. In this part, we quantitatively analyze the properties of these disentangled semantics as well as the face manipulation approach. We conduct the following analysis: (i) whether the manipulation can indeed increase or decrease the attribute score, and how manipulating one attribute affects the scores of other attributes (Section 5.1); (ii) how GANs learn the face representation layer by layer (Section 5.2); and (iii) how the attribute manipulation affects the face identity (Section 5.3).

5.1 Evaluating Editing Performance with Re-scoring

After manipulation, we further predict the attribute scores of the resulted face. In this way, we can compute the score change compared to the input face. We use the re-scoring to verify whether the manipulation has been successfully performed. For example, when we move the latent code towards "male" direction (*i.e.*, the positive direction of "gender" boundary), we would expect the "gender" score to increase. This metric can also be used to evaluate the disentanglement between different semantics. For example, if we want to see how "gender" and "age" correlate with each other, we can move the latent code along the "gender" boundary and see how the "age" score varies.

We use $2K$ synthesis for re-scoring analysis on PG-GAN [1], StyleGAN [3] $\mathcal{Z}$ space, and StyleGAN $\mathcal{W}$ space. The results in Table 3 indicate that: (i) InterFaceGAN can convincingly increase the target semantic scores by manipulating the appropriate attributes (see diagonal entries). (ii) Manipulating one attribute may affect the scores of other attributes. Taking PGGAN (Table 3 (a)) as an example, when manipulating "age", "gender" score also increases. This is consistent with the observation from Section 4.3. To some extent, we can treat this as an another disentanglement measurement. Under this metric, we also see that $\mathcal{W}$ space (Table 3 (c)) is more disentangled than $\mathcal{Z}$ space (Table 3 (b)) in StyleGAN. (iii) This new metric is asymmetric. Taking PGGAN (Table 3 (a)) as an example, when we manipulate "age", "eyeglasses" is barely affected.

TABLE 3
Re-scoring analysis on the semantic manipulation achieved by InterFaceGAN. Each row shows the results by manipulating a particular attribute.
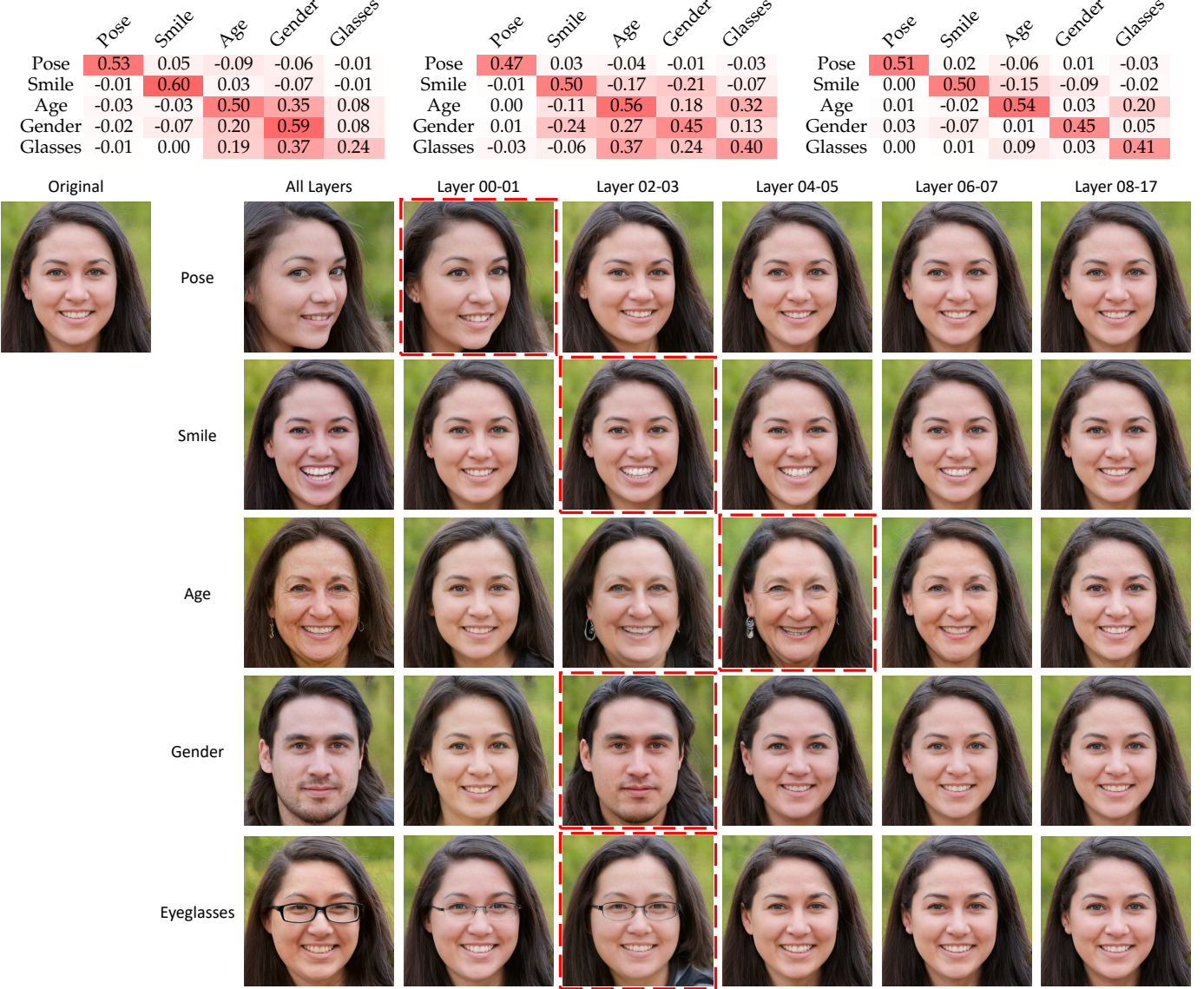
(a) PGGAN [1].

	Pose	Smile	Age	Gender	Glasses
Pose	0.53	0.05	-0.09	-0.06	-0.01
Smile	-0.01	0.60	0.03	-0.07	-0.01
Age	-0.03	-0.03	0.50	0.35	0.08
Gender	-0.02	-0.07	0.20	0.59	0.08
Glasses	-0.01	0.00	0.19	0.37	0.24

(b) StyleGAN $\mathcal{Z}$ space.

	Pose	Smile	Age	Gender	Glasses
Pose	0.47	0.03	-0.04	-0.01	-0.03
Smile	-0.01	0.50	-0.17	-0.21	-0.07
Age	0.00	-0.11	0.56	0.18	0.32
Gender	0.01	-0.24	0.27	0.45	0.13
Glasses	-0.03	-0.06	0.37	0.24	0.40

(c) StyleGAN $\mathcal{W}$ space.

	Pose	Smile	Age	Gender	Glasses
Pose	0.51	0.02	-0.06	0.01	-0.03
Smile	0.00	0.50	-0.15	-0.09	-0.02
Age	0.01	-0.02	0.54	0.03	0.20
Gender	0.03	-0.07	0.01	0.45	0.05
Glasses	0.00	0.01	0.09	0.03	0.41

Fig. 12. **Layer-wise manipulation** results with StyleGAN [3]. On the top-left corner is the raw synthesis. The images on the first column are the results of varying the codes at all the layers. Images in red dashed boxes highlight the best layer-wise manipulation results.

But when we manipulate "eyeglasses", "age" score increase a lot. The same phenomenon also happens to "gender" and "eyeglasses". This provides us with adequate information about the entanglement between the semantics learned in the latent representation.

5.2 Per-Layer Representation Learned by GANs

Unlike the traditional generator, StyleGAN [3] feeds the latent code to all convolutional layers. This enables us to study the per-layer representation. Given a boundary, we can use it to only vary the latent codes that are fed into some particular layers. In practice, we divide the 18 layers into 5 groups, *i.e.*, 00-01, 02-03, 04-05, 06-07, and 08-17. Then we conduct the same experiment as in re-scoring analysis to examine the importance of each group for each attribute.

Table 4 and Figure 12 show the quantitative and qualitative results respectively. From Table 4, we can see

TABLE 4
Layer-wise analysis on the semantic manipulation achieved by InterFaceGAN using StyleGAN [3]. Each row shows the results by manipulating a particular attribute and how the scores of itself and others vary.

Layer	All	00-01	02-03	04-05	06-07	08-17
Pose	0.51	0.42	0.20	0.03	0.01	0.00
Smile	0.50	0.02	0.32	0.24	0.08	0.01
Age	0.54	0.09	0.20	0.23	0.19	0.04
Gender	0.45	0.05	0.44	0.10	0.02	0.00
Glasses	0.41	0.23	0.28	0.01	0.00	0.00

that "pose" is mostly controlled at layer 00-01, "smile" is controlled at layer 02-05, "age" is controlled at layer 02-07, "gender" is controlled at layer 02-03, and "eyeglasses" is controlled at layer 00-03. All attributes are barely affected by editing layer 08-17. That is because layer 08-17 mainly correspond to texture, such as skin color and

TABLE 5
Identity discrepancy after the face manipulation using InterFaceGAN. Larger number (within range [0, 1]) means lower similarity.

Layer	PGGAN $\mathcal{Z}$ Space	StyleGAN $\mathcal{Z}$ Space	StyleGAN $\mathcal{W}$ Space					
			All	00-01	02-03	04-05	06-07	08-17
Pose	0.48	0.41	0.46	0.39	0.28	0.07	0.03	0.01
Smile	0.24	0.31	0.21	0.04	0.20	0.10	0.04	0.01
Age	0.53	0.47	0.28	0.12	0.18	0.09	0.06	0.01
Gender	0.61	0.51	0.40	0.11	0.37	0.13	0.03	0.01
Glasses	0.55	0.49	0.37	0.21	0.29	0.09	0.06	0.01

background [3]. We may identify their roles if we have texture-related attribute classifiers. Visualization results in Figure 12 also gives the same conclusion. It implies that GANs learn different representations at different layers. This provides us with some insights into a better understanding of the learning mechanism of GANs.

5.3 Effect of Learned Semantics on Face Identity

Identity is essential for face analysis, thus we perform the identity analysis to see how identity information varies in the manipulation process using InterFaceGAN. We employ a face recognition model to extract the identity features from the faces before and after semantic editing. The face recognition model is trained for the face verification task. It extracts a $256d$ feature vector from a given face image and uses cosine distance as the metric to evaluate the similarity between two subjects.

Table 5 shows the results corresponding to different latent spaces from different models, from which we have the following observations: (i) "Gender" affects the identity most and "smile" affects the identity least. To some extent, this can be used to verify how sensitive the face identity is to a particular attribute. For example, "pose" and "eyeglasses" seem to also affect the identity a lot. This makes sense since large pose is always the obstacle in face recognition task and eyeglasses are commonly used to disguise identity in the real world. We may use InterFaceGAN to synthesize more hard samples to in turn improve the face recognition model. (ii) StyleGAN $\mathcal{W}$ space best preserves the identity information due to its disentanglement property. That is because identity is much more complex than other semantics. A more disentangled representation is helpful in identity control. (iii) As for the layer-wise results, we can get a similar conclusion to the layer-wise analysis in Section 5.2.

6 REAL IMAGE MANIPULATION

In this section, we apply the semantics implicitly learned by GANs to real face editing. Since most of the GAN models lack the inference function to encode real images, we try two approaches. One is based on GAN inversion, which inverts the target image to the latent code for further editing. The other uses InterFaceGAN to generate synthetic image pairs and trains additional feed-forward pixel-to-pixel models.

6.1 Combining GAN Inversion with InterFaceGAN

Recall that InterFaceGAN achieves semantic face editing by moving the latent code along a particular direction in the latent space. Accordingly, for real image editing, one straightforward way is to invert the target face back to a latent code. It turns out to be a non-trivial task because GANs do not fully capture all the modes and the diversity of the true distribution, which means it is difficult to perfectly recover any real image with a finitely dimensional latent code. 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 [47], [54]. The other is encoder-based, where an extra encoder network is trained to learn the inverse mapping [43], [51]. We integrate these two baseline approaches into InterFaceGAN to see how our proposed manipulation pipeline performs on real image editing.

Figure 13 and Figure 14 show the manipulation results by using optimized-based inversion approach and encoder-based inversion approach respectively. In Figure 13, we use StyleGAN [3] as the generative model. Following prior work [47], [48], [54], we treat the layer-wise styles (*i.e.*, **w** for all layers) as the optimization target such that the input image can be better reconstructed. In Figure 14, we use LIA [51] as the generative model, which learns an additional encoder beyond the original two-player game in GANs. We observe that both approaches work well together with InterFaceGAN and we are able to realistically edit the input faces with various attributes, like age and face pose. It demonstrates the generalization ability of InterFaceGAN. We also notice that the optimization-based method better preserves the identity information and the face details. But the encoder-based approach is much faster. In particular, to invert a single image, the optimization-based method needs around 3 minutes while the encoder-based method needs less than 0.1 seconds. More results can be found in **Appendix**.

6.2 Training with Paired Synthetic Data Collected from InterFaceGAN

Another way to apply InterFaceGAN to real image editing is to train additional models. Unlike existing face manipulation models [7], [41], [64] that are trained on a real dataset, we use InterFaceGAN to build a synthetic dataset for training. There are two advantages: (i) With recent advancement, GANs can produce high-quality image [3], [4], significantly narrowing the domain gap. (ii) With the strong manipulation capability of InterFaceGAN, we can easily create unlimited paired data, which is difficult to obtain in the real world. Taking eyeglasses editing as an example, we can sample numerous latent codes, move them along the "eyeglasses" direction, and re-score them to select the ones with highest score change. With paired data as input and supervision, we train pix2pixHD [5] to achieve face editing. However, pix2pixHD has its shortcoming
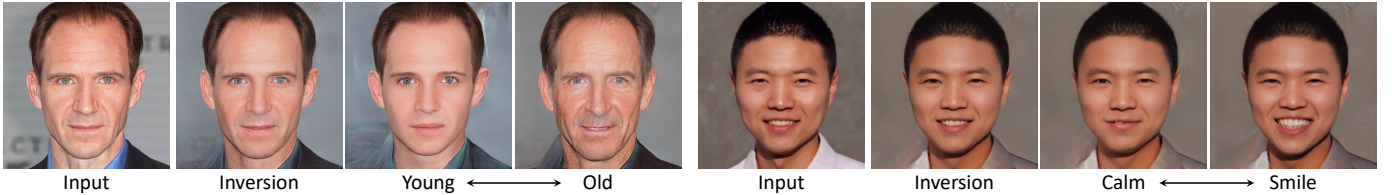
Fig. 13. **Manipulating real faces with GAN inversion**. Given a target image, we first invert it by optimizing the latent code and then manipulate the inverted code with InterFaceGAN. StyleGAN [3] is used as the generative model.
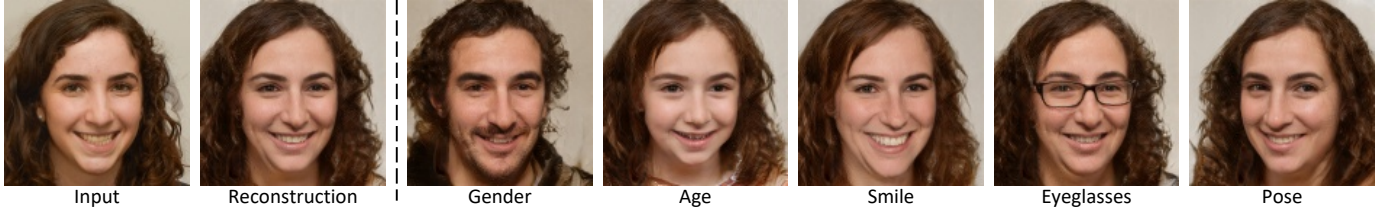

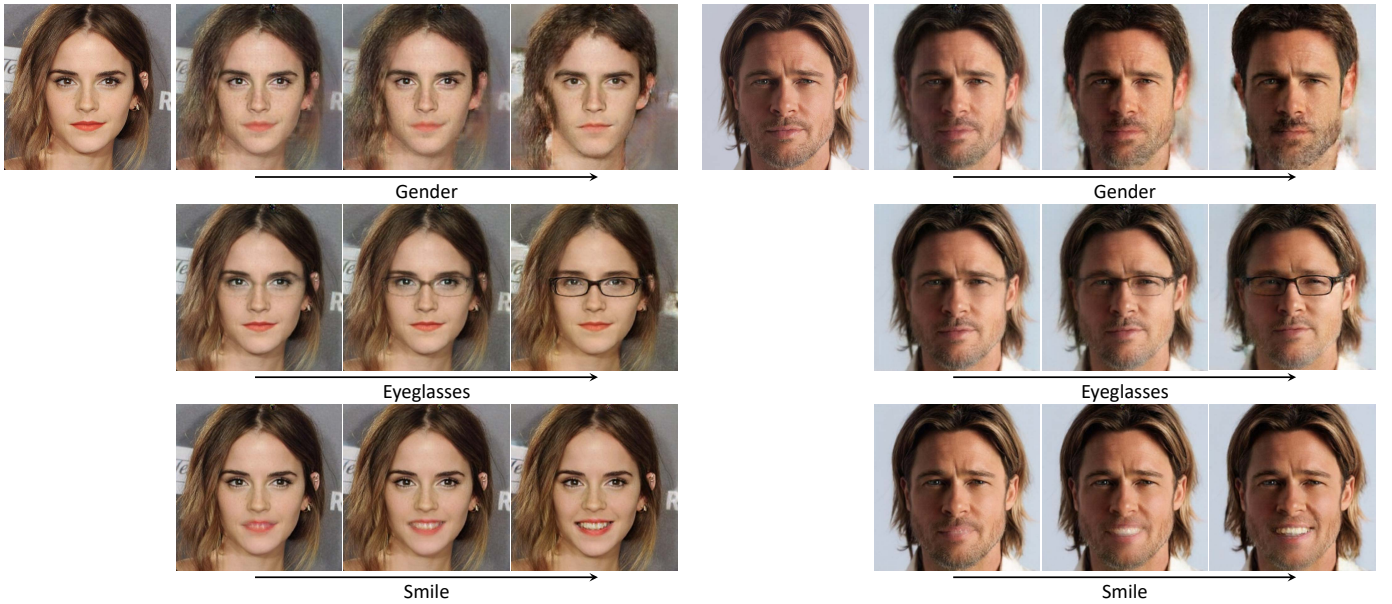Fig. 14. **Manipulating real faces with encoder-decoder generative model**, LIA [51].

Fig. 15. **Real image manipulation by learning additional pix2pixHD models** [5] on synthetic dataset collected from InterFaceGAN. DNI [68] is used to get continuously varying results.

that we cannot manipulate the attribute in a continuous manner [69]. To solve this problem, we introduce DNI [68] by first training an identical mapping network and then fine-tuning it for a particular attribute. We summarize the training pipeline as:

1) Prepare $10K$ synthetic pairs for each attribute.
2) Learn an identical pix2pixHD model, which is to transfer the input domain to itself.
3) Fine-tune the model to transfer a certain attribute by using the original synthesis as the input and the manipulation results with InterFaceGAN as the supervision.
4) Interpolate the model weights from the identical model to the fine-tuned model for gradual editing.

We choose "gender", "eyeglasses", and "smile" as the target attributes. Figure 15 visualizes some results. We can conclude that: (i) The pix2pixHD models trained on the synthetic dataset can successfully manipulate the input face with respect to the target attribute. It suggests that the data generated by InterFaceGAN can well support model

training, which may lead to more applications. (ii) For "gender" attribute, we only use female as the input and use male as the supervision. However, after the model training, we can even use this model to add moustache onto male faces as shown in Figure 15 (right example). (iii) Following DNI [68], we interpolate the weights between the identical model and the fine-tuned model. By doing so, we can gradually manipulate the attributes of the input face. The main advantage of learning an additional feed-forward model is its fast inference speed. Also, compared to the encoder-based inversion approach in Section 6.1, pix2pixHD model better preserves the identity information. (iv) During the weight interpolation process, we find that "smile" attribute does not perform as well as "gender" and "eyeglasses". That is because smiling is not a simple pixel-to-pixel translation task but requires a reasonable movement of lips. It is also why pix2pixHD can not be applicable to learning pose rotation, which requires larger movement. Accordingly, the primary limitation of this kind of approach is that it can only transfer some easy-to-map semantics, such as wearing "eyeglasses".

7 DISCUSSION AND CONCLUSION

Interpreting the representation learned by GANs and the disentanglement of attributes is vital for understanding the internal mechanism of deep generative models. In this work, we provide some pilot studies in this direction by taking face synthesis models as an example. There are many future works to be done. As our visual world is far more complicated than faces, it will be interesting to look into the generative models trained to synthesize generic objects and scenes. For example, for scene generation, besides learning semantics for the entire image, the model should create a spatial layout and learn to synthesize any individual objects inside. From this perspective, we need a more general method to interpret the GAN models beyond faces. For face models, there are also many directions worth further studying. As we have already discussed in Section 4.2, our method may fail for long-distance manipulation due to the linear assumption. More adaptive and expressive manipulation models, such as non-linear models, would solve this problem On the other hand, we use off-the-shelf classifiers as auxiliary predictors. This limits the semantics we can find in the latent space since we may not have the proper classifiers or the attribute may not be well defined or annotated. Hence, identifying the semantics emerging from synthesizing images in an unsupervised learning manner is worth further exploring.

To conclude, we interpret the disentangled face representation learned by GANs and conduct a thorough study on the emerging facial semantics. By leveraging the semantic knowledge encoded in the latent space, we are able to edit the attributes in face images realistically. The conditional manipulation technique is further introduced to disentangle different attributes for more precise control of face editing. Further experiments suggest that InterFaceGAN is applicable to real image manipulation.

ACKNOWLEDGMENTS

REFERENCES

[1] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *ICLR*, 2018.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019.

[4] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *arXiv preprint arXiv:1912.04958*, 2019.

[5] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018.

[6] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *CVPR*, 2020.

[7] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *NeurIPS*, 2017.

[8] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba, "Semantic photo manipulation with a generative image prior," *SIGGRAPH*, 2019.

[9] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[10] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *ECCV Workshop*, 2018.

[11] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *CVPR*, 2017.

[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *ICCV*, 2019.

[13] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *NeurIPS*, 2018.

[14] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," *arXiv preprint arXiv:1910.12713*, 2019.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *NeurIPS*, 2017.

[17] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *ICLR*, 2018.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *ICML*, 2019.

[20] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *ICLR*, 2019.

[21] N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer, and P. van der Smagt, "Metrics for deep generative models," in *AISTAT*, 2018.

[22] G. Arvanitidis, L. K. Hansen, and S. Hauberg, "Latent space oddity: on the curvature of deep generative models," in *ICLR*, 2018.

[23] L. Kuhnel, T. Fletcher, S. Joshi, and S. Sommer, "Latent space non-linear statistics," *arXiv preprint arXiv:1805.07632*, 2018.

[24] S. Laine, "Feature-based metrics for exploring the latent space of generative models," in *ICLR Workshop*, 2018.

[25] H. Shao, A. Kumar, and P. Thomas Fletcher, "The riemannian geometry of deep generative models," in *CVPR Workshop*, 2018.

[26] P. Bojanowski, A. Joulin, D. Lopez-Pas, and A. Szlam, "Optimizing the latent space of generative networks," in *ICML*, 2018.

[27] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," in *ICLR*, 2019.

[28] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[29] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *CVPR*, 2017.

[30] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "Ganalyze: Toward visual definitions of cognitive image properties," in *ICCV*, 2019.

[31] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," in *ICLR*, 2020.

[32] C. Yang, Y. Shen, and B. Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *arXiv preprint arXiv:1911.09267*, 2019.

[33] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017.

[34] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016.

[35] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, 2017.

[36] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Towards large-pose face frontalization in the wild," in *ICCV*, 2017.

[37] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Towards open-set identity preserving face synthesis," in *CVPR*, 2018.

[38] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with gan for transferring multiple face attributes," in *ECCV*, 2018.

[39] Y. Shen, B. Zhou, P. Luo, and X. Tang, "Facefeat-gan: a two-stage approach for identity-preserving face synthesis," *arXiv preprint arXiv:1812.01288*, 2018.

[40] C. Donahue, A. Balsubramani, J. McAuley, and Z. C. Lipton, "Semantically decomposing the latent spaces of generative adversarial networks," in *ICLR*, 2018.

[41] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, "Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis," in *CVPR*, 2018.

[42] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," in *NeurIPS Workshop*, 2016.

[43] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *ECCV*, 2016.

[44] Z. C. Lipton and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," in *ICLR Workshop*, 2017.

[45] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *TNNLS*, 2018.

[46] F. Ma, U. Ayaz, and S. Karaman, "Invertibility of convolutional generative networks from partial measurements," in *NeurIPS*, 2018.

[47] A. Rameen, Q. Yipeng, and W. Peter, "Image2stylegan: How to embed images into the stylegan latent space?" in *ICCV*, 2019.

[48] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan++: How to edit the embedded images?" *arXiv preprint arXiv:1911.11544*, 2019.

[49] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *ICLR*, 2017.

[50] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *ICLR*, 2017.

[51] J. Zhu, D. Zhao, and B. Zhang, "Lia: Latently invertible autoencoder with adversarial learning," *arXiv preprint arXiv:1906.08090*, 2019.

[52] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba, "Seeing what a gan cannot generate," in *ICCV*, 2019.

[53] ——, "Inverting layers of a large generator," in *ICLR Workshop*, 2019.

[54] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," *arXiv preprint arXiv:2004.00049*, 2020.

[55] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code gan prior," in *CVPR*, 2020.

[56] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting deep generative prior for versatile image restoration and manipulation," *arXiv preprint arXiv:2003.13659*, 2020.

[57] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *arXiv preprint arXiv:1612.00215*, 2016.

[58] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019.

[59] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[60] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NeurIPS*, 2017.

[61] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[62] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[63] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *NeurIPS*, 2017.

[64] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018.

[65] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *CVPR*, 2020.

[66] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[68] X. Wang, K. Yu, C. Dong, X. Tang, and C. C. Loy, "Deep network interpolation for continuous imagery effect transition," in *CVPR*, 2019.

[69] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," *arXiv preprint arXiv:2003.03581*, 2020.

[70] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of data science*. Cambridge University Press, 2020.

APPENDIX A
PROOF

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

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 $P(|\mathbf{n}^T\mathbf{z}| \leq 2\alpha\sqrt{\frac{d}{d-2}}) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$ for any $\alpha \geq 1$ and $d \geq 4$. Here $P(\cdot)$ 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 $P(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$, where z_1 denotes the first entry of $\mathbf{z}$.

As shown in Figure 16, 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 $P(H) \leq P(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}})$. Now, we will show $P(H) \geq (1 - 3e^{-cd})(1 - \frac{2}{\alpha}e^{-\alpha^2/2})$

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

$$P(H) = P(|z_1| \leq 2\alpha\sqrt{\frac{d}{d-2}}|R \leq 2\sqrt{d})P(R \leq 2\sqrt{d})$$

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

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

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

$$= \int_0^{2\sqrt{d}} P(|z_1| \leq \frac{2\sqrt{d}}{r}\frac{\alpha}{\sqrt{d-2}}|R = 1)f(r)dr$$

$$\geq \int_0^{2\sqrt{d}} P(|z_1| \leq \frac{\alpha}{\sqrt{d-2}}|R = 1)f(r)dr$$

$$\geq \int_0^{2\sqrt{d}} (1 - \frac{2}{\alpha}e^{-\alpha^2/2})f(r)dr$$

$$= (1 - \frac{2}{\alpha}e^{-\alpha^2/2})\int_0^{2\sqrt{d}} f(r)dr$$

$$= (1 - \frac{2}{\alpha}e^{-\alpha^2/2})P(0 \leq R \leq 2\sqrt{d}).$$

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

$$P(H) = (1 - \frac{2}{\alpha}e^{-\alpha^2/2})P(0 \leq R \leq 2\sqrt{d})$$

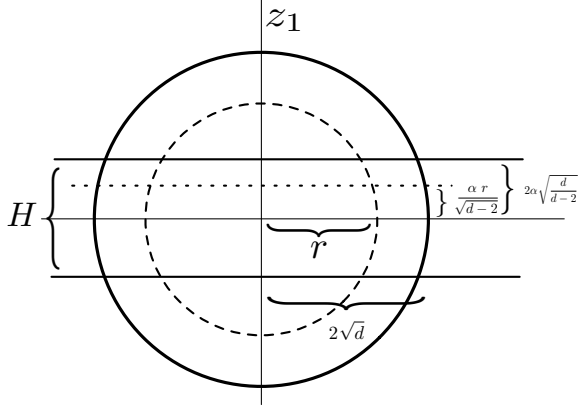$$\geq (1 - \frac{2}{\alpha}e^{-\alpha^2/2})(1 - 3e^{-cd}).$$

Fig. 16. 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".
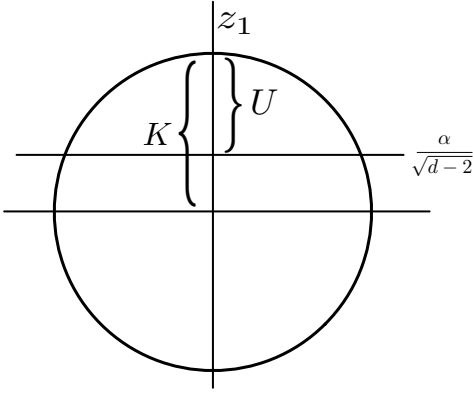

Fig. 17. Diagram for **Theorem 1**.

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 Figure 17 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

$$surf(U) = \int_{\frac{\alpha}{\sqrt{d-2}}}^{1} (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1$$

$$\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^{1} e^{-\frac{d-2}{2}z_1^2} A(d-1) dz_1$$

$$\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^{1} \frac{z_1\sqrt{d-2}}{\alpha} e^{-\frac{d-2}{2}z_1^2} A(d-1) dz_1$$

$$\leq \int_{\frac{\alpha}{\sqrt{d-2}}}^{\infty} \frac{z_1\sqrt{d-2}}{\alpha} e^{-\frac{d-2}{2}z_1^2} A(d-1) dz_1$$

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

Similarly, we have

$$surf(K) = \int_0^1 (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1$$

$$\geq \int_0^{\frac{1}{\sqrt{d-2}}} (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1$$

$$\geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2})^{\frac{d-2}{2}} A(d-1).$$

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

$$surf(K) \geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2})^{\frac{d-2}{2}} A(d-1)$$

$$\geq \frac{1}{\sqrt{d-2}} (1 - \frac{1}{d-2}\frac{d-2}{2}) A(d-1)$$

$$= \frac{A(d-1)}{2\sqrt{d-2}}.$$

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 [70]) *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

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

APPENDIX B
REAL IMAGE MANIPULATION

Besides the examples shown in the main paper, we present more results on real face editing by integrating Inter-FaceGAN with GAN inversion methods, including both optimization-based [54] and encoder-based [51]. Figure 18 compares these two approaches. We can tell that the optimization-based method better recovers the input images and hence better preserves the identity information. But for both methods, the interpretable semantics inside the latent representation are capable of faithfully editing the corresponding facial attributes of the reconstructed face.

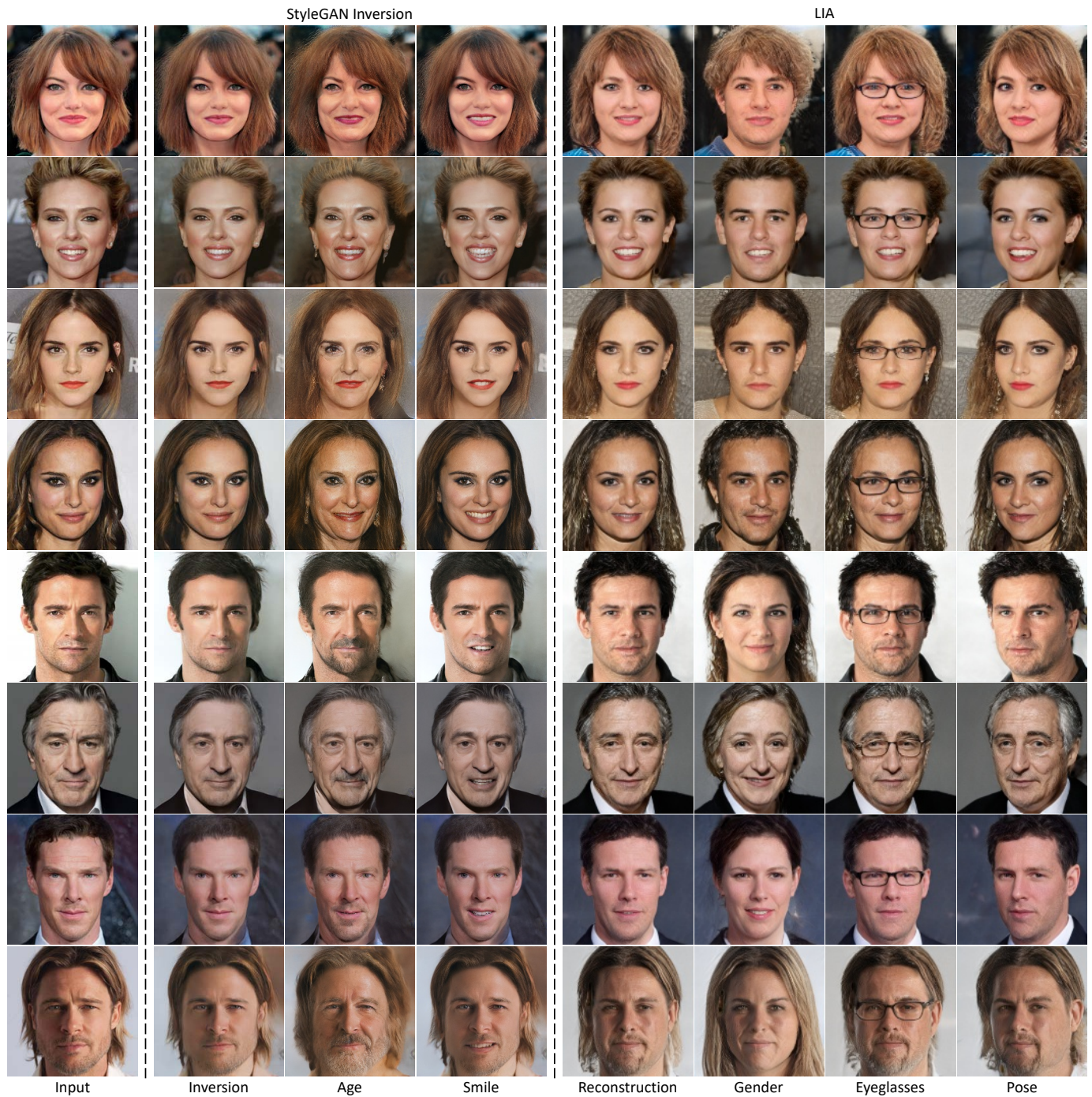|	StyleGAN Inversion				LIA		
Input	Inversion	Age	Smile	Reconstruction	Gender	Eyeglasses	Pose

Fig. 18. **Real image manipulation** with optimization-based GAN inversion approach [54], and encoder-based GAN inversion approach [51].