Interpreting the Latent Space of GANs for Semantic Face Editing

Yujun Shen\textsuperscript{1}, Jinjin Gu\textsuperscript{2}, Xiaoou Tang\textsuperscript{1}, Bolei Zhou\textsuperscript{1}
\textsuperscript{1}CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong \textsuperscript{2}CUHK (SZ)
\{sy116, xtang, bzhou\}@ie.cuhk.edu.hk, jinjingu@link.cuhk.edu.cn

Figure 1: Manipulating various facial attributes through varying the latent code of a well-trained GAN model. The first column shows the original synthesis from ProgressiveGAN \cite{17}, while each of the other columns shows the result of manipulating a specific attribute.

Abstract

Despite the recent advance of Generative Adversarial Networks (GANs) in high-fidelity image synthesis, there lacks enough understandings on how GANs are able to map the latent code sampled from a random distribution to a photo-realistic image. Previous work assumes the latent space learned by GAN follows a distributed representation but observes the vector arithmetic phenomenon of the output’s semantics in latent space. In this work, we interpret the semantics hidden in the latent space of well-trained GANs. We find that the latent code for well-trained generative models, such as ProgressiveGAN and StyleGAN, actually learns a disentangled representation after some linear transformations. We make a rigorous analysis on the encoding of various semantics in the latent space as well as their properties, and then study how these semantics are correlated to each other. Based on our analysis, we propose a simple and general technique, called InterFaceGAN, for semantic face editing in latent space. Given a synthesized face, we are able to faithfully edit its various attributes such as pose, expression, age, presence of eyeglasses, without retraining the GAN model. Furthermore, we show that even the artifacts occurred in output images are able to be fixed using same approach. Extensive results suggest that learning to synthesize faces spontaneously brings a disentangled and controllable facial attribute representation.\textsuperscript{1}

1. Introduction

Generative Adversarial Networks (GANs) \cite{12} have significantly advanced image synthesis in recent years. The rational behind is to learn the mapping from a latent space to real data distribution through adversarial training. After learning such a nonlinear mapping, GAN is capable of producing photo-realistic images by sampling latent code from a random distribution.

Existing work typically focuses on discovering a more accurate distribution from ground-truth to improve the synthesis quality \cite{35, 24, 17}, however, few efforts have been made on studying what GAN actually learns with respect to the latent space. Taking face synthesis as an example, although the latent code determines which face to produce, it remains uncertain how the latent code corresponds to

\textsuperscript{1}Code and model are available at https://shenyujun.github.io/InterFaceGAN/
various semantic attributes of the output face image, such as gender and age. Some methods are proposed to control the output image by learning a disentangled latent space [23, 9], but they require carefully designed regularizers or labeled attributes to supervise the training, while their synthesis quality is far from the appealing results achieved by unconditioned GANs [17, 18]. Radford et al. [27] first observes the vector arithmetic property in latent space, suggesting that GANs seems to be learning some semantics in the earliest hidden space. A recent work [4] further shows that some units from the intermediate layers of GAN generator are specialized to synthesize some visual concepts, such as sofa and TV in the model for living room generation. Even so, there lacks enough understanding on how GAN connects the very first latent representation to semantic contents of the synthesis, as well as how we are able to edit the output image by varying the latent code.

In this paper, we interpret the latent space of GANs trained for face synthesis, by discovering the subspaces associated with various facial attributes. We find that a well-trained GAN has already automatically disentangled various semantic concepts in the latent space. More specifically, we prove that a true-or-false binary attribute is actually encoded in a linear subspace inside the entire latent space of GAN. Upon identifying these semantic subspaces using off-the-shelf linear classifiers, we can faithfully manipulate the facial attributes (e.g., pose, expression, age) by directly varying the latent code based on the corresponding linear subspaces without retraining the model. This provides us a simple and flexible face editing approach InterFaceGAN (short for Interpret Face GANs), in which we exploit the interpreted latent semantics of any pre-trained GAN model for face editing.

In some cases, however, the above disentanglement may be concealed by some coupled attributes in training data, e.g., old people are more likely to wear eyeglasses. To better understand how GAN encodes such phenomenon in the latent space when trained to produce similar distribution as real observed data, we study the attribute correlations by proposing two metrics based on the model preference in generation and the dependence of semantic subspaces respectively. We further manage to decorrelate them by developing a conditional manipulation technique to edit some particular attribute yet preserve others. Moreover, we find that the artifacts, which sometimes occur in the output image, can also be treated as a special attribute to be manipulated and fixed using the same approach.

Our contributions are summarized as follows:

- We propose a GAN-based face editing approach, InterFaceGAN, and evaluate it on the state-of-the-art face synthesis models, including ProgressiveGAN [17] and StyleGAN [18]. We achieve high-quality semantic image manipulation by controlling the facial attributes without retraining the model, as shown in Fig.1. We also present an efficient method for conditional editing, i.e., manipulating specific attribute while other attributes are preserved.
- Our approach is able to fix the artifacts produced by GANs. It is surprising that GAN has also encoded “quality” as a semantic subspace in the latent space. Our work provides insights on further understanding and improving GANs.

1.1. Related Work

Generative Adversarial Networks. GANs [12] have brought wide attention in recent years. The efforts made to improve GANs lie in various aspects, including designing better objective functions [27, 36], improving synthesis diversity [35, 24, 7], image resolution [17, 18], as well as training stability [1, 13, 5]. Despite this tremendous success, little work has been done on understanding what GANs have learned in the process of synthesizing the real visual world. Prior work [27, 32] observed the vector arithmetic property in the latent space. Bau et al. [4] analyzed GANs by visualizing the spatial feature map and understanding the behavior of different units in intermediate layers. However, detailed study on the fine-grained relationship between input latent space and semantic attributes of output images is still missing.

Disentangled Representation Learning with GAN. Besides improving GANs to synthesize images in an unconditional way, plenty of work has been done to control the contents and attributes of the outputs. CGAN [23] was firstly proposed to add constraints into the training procedure. Specifically, additional label together with the random latent code is fed into the generator, and then used as supervision to ensure that GAN outputs image with desired category. In this way, latent code and the auxiliary label are considered as decomposed such that changing one item will not affect the other. This idea is further extended with more carefully designed loss functions [25, 31], introduction of semantic attribute features [34, 3, 33, 30], as well as novel architectures [10, 29] to improve the disentanglement and synthesis quality. However, all these approaches require additional information involved in GAN learning. InfoGAN [9] learned disentangled latent space unsupervisedly by adding regularizers to the generator to maximize the mutual information. Different from previous learning-based methods, this work explores the disentanglement of semantics in the latent space of unconstrained GANs without any retraining or redesigning the models themselves.
Study on Latent Space of GAN. Latent space is treated as Riemannian manifolds by recent work [8, 2, 19]. They focus on exploring how to make the output image vary more smoothly through interpolation in latent space. This idea is improved in [20] by employing feature-based metrics as the path length in image space. Some work [28] observed that the linear paths in latent space can closely approximate geodesics on generated manifold. There are also some methods targeting at the inversion from image space back to latent space [26, 37, 22] for better image manipulation. GLO [6] optimized the generator and latent code simultaneously to learn a better latent space. Unlike them, this paper studies the latent space by probing the hidden semantic subspaces using linear attribute classifiers. Some concurrent work also explore the semantics in latent space of GANs for image manipulation: [16] studied the steerability of GAN model by shifting the latent distribution and achieved the control of camera motion and image color tone, while [11] improved the memorability of the output image via varying the latent code.

2. Latent Space Interpretation

2.1. Problem Definition

Given a well-trained GAN model, the generator can be formulated as a deterministic function \( g : Z \rightarrow \mathcal{X} \). Here, \( Z \subseteq \mathbb{R}^d \) denotes the \( d \)-dimensional latent space, where the random sample \( z \) is drawn from a specific distribution. \( \mathcal{N}(0, I_d) \) is commonly used [24, 17, 18, 7]. \( \mathcal{X} \) stands for the image space, where each sample \( x \) possesses certain semantic attributes, making it distinguishable from others. For example, gender (male vs. female) and age (old vs. young) are both discriminative attributes of face images.

Suppose we have an attribute scoring function \( f_A : \mathcal{X} \rightarrow \mathcal{A} \), where \( \mathcal{A} \subseteq \mathbb{R}^m \) represents the attribute space consisting of \( m \) attributes. We can use it to label any sample in \( \mathcal{X} \) with \( a = f_A(x) \), no matter \( x \) comes from real data or synthesized data. This work aims at exploring how GAN builds the relationship between \( z \) and \( f_A(g(z)) \) when it learns to map the latent space to real data observation.

2.2. Semantics in Latent Space

Prior work [27, 6] has observed the vector arithmetic phenomenon in GANs. Let \( z_1, z_2, z_3 \) stand for three latent codes which lead to syntheses “man w/ glasses”, “man w/o glasses”, and “woman w/o glasses” respectively, then \( g(z_1 - z_2 + z_3) \) will output a “woman w/ glasses”. This observation raises many questions to be answered. For instance, is adding up \( z_1 - z_2 \) able to add glasses to any person? Whether it is the vector \( z_1 - z_2 \) that helps the man wear glasses, or \( z_3 - z_2 \) that feminizes the man? Does these two attributes disentangled with each other under the representation of GAN model? These questions motivate us to explore the hidden semantics inside the latent space.

According to Property 1 below, both \( z_1 - z_2 \) and \( (z_3 - z_2) \) define a hyperplane in \( Z \). Thus, we make an assumption\(^2\) that for any attribute that can be treated as a bi-classification problem, there exists a hyperplane in the latent space serving as the separation boundary. Samples from the same side of the boundary will have same attribute.

Under such hypothesis, when a sample lies near the boundary and is moved across the hyperplane, the corresponding attribute will turn into the opposite. According to Property 2 below, random samples drawn from \( \mathcal{N}(0, I_d) \) are very likely to locate close enough to a given boundary. We therefore should be able to manipulate almost all syntheses with the above operation.

\(^2\)This assumption is empirically demonstrated in Sec.3.2.

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

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

\[^2\]This assumption is empirically demonstrated in Sec.3.2.

\[^3\]When \( d = 512 \), we have \( P(|n^T z| > 5.0) < 1e^{-6} \). It suggests that almost all sampled latent codes are expected to locate within 5 unit-length to the boundary. Proof of this property can be found in Appendix.
f(g(z_{edit})) = f(g(z)) + \lambda \alpha \text{ after editing. Similarly, } \alpha < 0 \text{ will make the synthesis look more negative.}

**Conditional Manipulation.** When the case comes to \( m \) different attributes, we have

\[
a \equiv f_A(g(z)) = \Lambda N^T z,
\]

where \( a = [a_1, \ldots, a_m]^T \) denotes the attribute scores, \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \) is a diagonal matrix containing the linear coefficients, while \( N = [n_1, \ldots, n_m] \) indicates the separation boundaries. Aware of the distribution of random sample \( z \), which is \( \mathcal{N}(0, I_d) \), we can easily compute the mean and covariance matrix of the attribute scores \( a \) with

\[
\mu_a = \mathbb{E}(\Lambda N^T z) = \Lambda N^T \mathbb{E}(z) = 0, \quad (4)
\]

\[
\Sigma_a = \mathbb{E}(\Lambda N^T zz^T N \Lambda^T) = \Lambda N^T \mathbb{E}(zz^T) N \Lambda^T = \Lambda N^T N \Lambda. \quad (5)
\]

We therefore have \( a \sim \mathcal{N}(0, \Sigma_a) \), which is a multivariate normal distribution. Different entries of \( a \) are independent if and only if \( \Sigma_a \) is a diagonal matrix, which requires \( \{n_1, \ldots, n_m\} \) to be orthogonal with each other. This is the most ideal case where the attributes are mutually independent in the training set. For most cases, however, the training data is biased to some particular attribute combinations, e.g., male celebrities are more inclined to wear glasses compared to females. Under such situation, it is hard to find the accurate boundary for each attribute.

Nevertheless, such imbalance can also be captured by GAN, since it is trained with the objective to produce indistinguishable distribution from real data. We can manually find some transformed subspace by forcing \( N^T N \) in Eq.\((5)\) to be diagonal. Projection is one of the most efficient methods to orthogonalize different vectors. As shown in Fig.2, given two hyperplanes with normal vectors \( n_1 \) and \( n_2 \) respectively, we can easily find a projected direction \( n_1 - (n_1^T n_2) n_2 \), 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 are more than one attribute to be conditioned on, just subtract the projection from the primal direction onto the plane that is constructed by all conditioned directions.

![Figure 2: Illustration of the conditional manipulation in subspace. The projection of \( n_1 \) onto \( n_2 \) is subtracted from \( n_1 \), resulting in a new direction \( n_1 - (n_1^T n_2) n_2 \).](image)

### 2.4. InterFaceGAN Procedure

In this part, we introduce the procedure of the GAN-based semantic editing approach, InterFaceGAN:

**Step-1:** Semantic Boundary Searching. We first use data-driven method to find the decision boundary for each attribute. For example, we can manually label a collection of synthesized data, and then train off-the-shelf linear classifier by solving a bi-classification problem. Then the classifier will provide a normal direction of the boundary.

**Step-2:** Semantic Manipulation. For a particular boundary, we randomly sample some latent codes and push them towards the positive direction of the hyperplane. There are two possible outcomes. For one case, if the original point lies in the negative side, the corresponding attribute should inverse after the shift in latent space. For the other case, if the original point lies in the positive side already, the attribute score should get increased, e.g., from smile to laugh. Similarly, the above operation is applied to the negative direction.

**Step-3:** Conditional Manipulation. For a collection of boundaries, we compute the cosine distance between the corresponding normal vectors to see how these attributes are entangled. Based on this, we can figure out the preference of GAN in combining these attributes when producing a fake image. Moreover, we derive conditional directions by preforming linear transformations onto current boundaries, as described in Sec.2.3. Then we repeat Step-2 on the new directions to achieve conditional manipulation.

### 3. Experiments

In this section, we conduct experiments on two state-of-the-art face synthesis models\(^4\), i.e., ProgressiveGAN [17] and StyleGAN [18], to interpret the semantics hidden in the latent space of the pre-trained GAN models as well as edit the semantic attributes of the output images. We directly employ the pre-trained models released by the original papers without tuning. Since all these two models are trained without any constraints, they are appropriate for understanding how native GANs learn the mapping from latent space to image space. Specifically, experiments in Sec.3.2, Sec.3.3, and Sec.3.4 are conducted on ProgressiveGAN to interpret the latent space of traditional generator. Experiments in Sec.3.5 are carried out on StyleGAN model to investigate the novel style-based generator and also compare the difference between the two sets of latent representations in StyleGAN. We also apply our approach to real image in Sec.3.6, to see how the disentangled semantics implicitly learned by GANs can be applied to real face editing.

\(^4\)Official ProgressiveGAN models can be found at [https://github.com/tkarras/progressive_growing_of_gans](https://github.com/tkarras/progressive_growing_of_gans), and StyleGAN at [https://github.com/NVlabs/stylegan](https://github.com/NVlabs/stylegan).
3.1. 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 [21] with ResNet-50 network [14]. 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 ProgressiveGAN and StyleGAN are with $1024 \times 1024$ resolution, we resize them to $224 \times 224$ 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 latent code is as expected, and (ii) to get enough wearing-glasses samples, which are really rare in ProgressiveGAN model.

To find the semantic boundaries in latent space, which is the first step of InterFaceGAN described in Sec.2.4, 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. 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.

3.2. Latent Space Separation

As mentioned in Sec.2.2, 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 same attribute. Accordingly, we would like to first evaluate the correctness of this assumption to make the remaining analyses considerable.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{Dataset} & \textbf{Pose} & \textbf{Smile} & \textbf{Age} & \textbf{Gender} & \textbf{Eyeglasses} \\
\hline
Validation & 100.0 & 96.9 & 97.9 & 98.7 & 95.6 \\
All & 90.3 & 78.5 & 75.3 & 84.2 & 80.1 \\
\hline
\end{tabular}
\caption{Classification accuracy (%) on separation boundaries in latent space with respect to different attributes.}
\end{table}

Following Sec.3.1, we train five independent linear SVMs on pose, smile, age, gender, and eyeglasses. The trained SVMs are then evaluated on the validation set as well as the entire set. Samples whose prediction scores lie in range $(0.4, 0.6)$ are filtered out, since samples with such scores are unreliable. Tab.1 shows the results. We find that, all linear boundaries achieve over $95\%$ accuracy for samples with high confidence level on attribute score. Even tested on all synthesized samples, these SVMs still have more than $75\%$ accuracy, experimentally demonstrating that for a binary attribute, there exists a linear hyperplane in the latent space that can well separate the data into two groups.

To further verify that these SVMs indeed capture the semantic information, we 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 obviously distinguishable to each other with respect to the corresponding attribute.

3.3. Latent Space Manipulation

As mentioned in Sec.2.3, once we have the separation boundary, we should be able to semantically edit synthesized images by manipulating the latent code along the corresponding normal direction.

\textbf{Single Attribute Editing.} Fig.4 plots the editing 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 pose is training with a linear SVM as a bi-classification problem, moving towards the
normal direction of the decision hyperplane can actually produce continuous changing. Furthermore, although there lacks enough data with extreme poses in the training set, GAN is capable of imagining how such faces should look like by moving the latent code along the direction corresponding to pose subspace. Same situation happens on eyeglasses attribute. Despite the inadequate number of wearing-eyeglasses samples in training data, we can manually create a lot by manipulating the latent code. These two observations provide strong evidence that GAN does not produce images randomly, but actually 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 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\(^5\), however, the editing results are no longer like the original face any more. But this effect does not affect our understanding about the disentangled semantics in latent space. That is because such extreme samples are very unlikely to be directly drawn from standard normal distribution, which is pointed out in Property 2 in Sec.2.2. 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.

\(^5\)We choose 5.0 as the threshold.

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 centered 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. 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.
Artifacts Correction. We further apply our approach to fix the artifacts that GAN has generated. First row shows some bad generation results, while the following two rows present the gradually corrected syntheses by moving the latent codes along the positive “quality” direction.

3.4. Conditional Manipulation

Besides identifying the latent semantics as well as editing face synthesis along single attribute direction, we further explore the relations between different semantic subspaces that ProgressiveGAN has learned.

Correlation between Attributes. Different from [18] 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 $\cos(n_1, n_2) = n_1^T n_2$, where $n_1$ and $n_2$ 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 $\rho$. Here, we have $\rho_{A_1,A_2} = \frac{\text{Cov}(A_1,A_2)}{\sigma_{A_1}\sigma_{A_2}}$, where $A_1$ and $A_2$ represent two random variables with respect to two attributes. $\text{Cov}(\cdot, \cdot)$ stands for covariance, and $\sigma$ 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 to each other. This observation reflects the attribute correlation in the training dataset (i.e., CelebA-HQ [17]) to some extent, where male old people are more likely to wear eyeglasses. This characteristic is also captured by GAN when learning to produce the real observation.

Conditional Manipulation. To decorrelate different semantics for independent facial attribute editing, we propose conditional manipulation in Sec.2.3. Fig.7 shows some results by manipulating one attribute with another one as 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 constraints, where we add glasses by conditionally preserving age and gender. At 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 attribute control.

3.5. Results on StyleGAN

The very recent model, StyleGAN [18], differs from traditional GANs with the novel design of style-based generator. Basically, StyleGAN learns to map the latent code from space $Z$ to another high dimensional space $W$ before feeding it into the generator. As pointed out in [18], $W$ shows much stronger disentanglement property than $Z$, since $W$ is not restricted to any certain distribution and can better model the underlying character of real data. This design makes StyleGAN more flexible to training set with arbitrary attribute distribution.

Table 2: Correlation matrix computed from the normal vectors of different 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 computed from the attribute distribution of synthesized data.

|        | 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 |
We did a similar analysis on both $\mathcal{Z}$ and $\mathcal{W}$ spaces of StyleGAN as did to ProgressiveGAN and found that $\mathcal{W}$ space indeed learns a more disentangled representation, as pointed out by [18]. 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.2.3 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\(^6\).

3.6. Real Image Manipulation

In this part, we try to manipulate real faces with the proposed InterFaceGAN to verify whether the semantic attributes learned by GAN can be applied to data from different domain (i.e., all test images are not seen by the model in training stage). 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 due to the fact that 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 fixed generator to minimize the pixel-wise reconstruction error [22]. The other is the encoder-based, where an independent encoder network is trained to learn the inverse mapping [37]. We tested the two baseline approaches and the results on ProgressiveGAN and StyleGAN 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 ProgressiveGAN. This can be imputed to the strong discrepancy between training and testing data distributions. For example, the model tends to

\(^{6}\)More details can be found in Appendix.
Figure 9: Analysis on the latent space $Z$ and disentangled latent space $W$ of StyleGAN [18] by taking age manipulation as an example. $W$ space behaves better for long term manipulation, but the flaw in $Z$ space can be fixed by projection (i.e., conditional manipulation) to achieve better performance.

![Figure 9](image)

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

![Figure 10](image)

generate Western people even the input is a Easterner (see the right sample in Fig.10). Even so, the manipulations made by InterFaceGAN based on the inverted images are still satisfying, i.e., the desired attributes are indeed modified.

Compared to ProgressiveGAN, the results on StyleGAN (third row) are much better. Here, we treat the layer-wise styles (i.e., the disentangled latent codes $w$ for all 18 convolutional layers) as the optimization target. When editing an instance, we move all latent codes toward the same semantic direction. As shown in Fig.10, we successfully change the attributes of real face images without retraining StyleGAN. This benefits from the disentangled semantics GAN has learned in the latent space.

4. Conclusion

We interpret the semantics hidden in the latent space of well-trained GANs. By leveraging the interpreted attributes spontaneously learned by GAN, we propose the InterFaceGAN approach to faithfully edit the synthesized images. Conditional manipulation technique is further introduced to decorrelate different semantics thus results in more independent attribute editing.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.

[2] G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent space oddity: on the curvature of deep generative models. In ICLR,
2018. 3
[3] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Towards open-set identity preserving face synthesis. In CVPR, 2018. 2, 11
[4] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Visualizing and understanding generative adversarial networks. In ICLR, 2019. 2
[5] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017. 2
[6] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776, 2017. 3
[7] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In ICRL, 2019. 2, 3
[8] N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer, and P. van der Smagt. Metrics for deep generative models. In AISTAT, 2018. 3
[9] 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. 2
[10] C. Donahue, A. Balsubramani, J. McAuley, and Z. C. Lipton. Semantically decomposing the latent spaces of generative adversarial networks. In ICLR, 2018. 2
[11] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola. Ganalyze: Toward visual definitions of cognitive image properties. arXiv preprint arXiv:1906.10112, 2019. 3
[12] 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. 1, 2
[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In NeurIPS, 2017. 2
[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 5
[15] J. Hopcroft and R. Kannan. Foundations of Data Science. 2014. 19
[16] A. Jahanian, L. Chai, and P. Isola. On the “steerability” of generative adversarial networks. arXiv preprint arXiv:1907.07171, 2019. 3
[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In ICLR, 2018. 1, 2, 3, 4, 7, 16, 18
[18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948, 2018. 2, 3, 4, 7, 8, 9, 14, 16, 18
[19] L. Kuhnle, T. Fletcher, S. Joshi, and S. Sommer. Latent space non-linear statistics. arXiv preprint arXiv:1805.07632, 2018. 3
[20] S. Laine. Feature-based metrics for exploring the latent space of generative models. In ICLR Workshop, 2018. 3
[21] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In ICCV, 2015. 5
[22] F. Ma, U. Ayaz, and S. Karaman. Invertibility of convolutional generative networks from partial measurements. In NeurIPS, 2018. 3, 8
[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014. 2
[24] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In ICLR, 2018. 1, 2, 3
[25] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In ICLR, 2017. 2
[26] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Alvarez. Invertible conditional gans for image editing. In NeurIPS Workshop, 2016. 3
[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016. 2, 3
[28] H. Shao, A. Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. In CVPRW, 2018. 3
[29] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang. Faceidgan: Learning a symmetry three-player gan for identity-preserving face synthesis. In CVPR, 2018. 2, 11
[30] 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. 2
[31] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In CVPR, 2017. 2, 11
[32] 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. 2
[33] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In ECCV, 2018. 2
[34] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In ICCV, 2017. 2, 11
[35] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318, 2018. 1, 2
[36] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In ICCV, 2017. 2
[37] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In ECCV, 2016. 3, 8
Appendix

We show more manipulation and conditional manipulation results on ProgressiveGAN in Sec.A and Sec.B. We then show analysis on StyleGAN in Sec.C. Finally, we provide a detailed proof of Property 2 in Sec.D.

A. Manipulation Results

In this section, we show manipulation results on ProgressiveGAN with respect to different facial attributes, include pose (Fig.11), expression (Fig.12), age (Fig.13), gender (Fig.14), and eyeglasses (Fig.15). We even found a boundary for synthesis quality such that we can fix some artifacts generated by GAN, as shown in Fig.16. Please refer to the video YouTube to check continuous attributes editing.

To achieve facial attribute manipulation, prior work proposed to train conditional GANs [31, 34, 3, 29]. However, training GANs with constraints by involving additional attribute labels as supervision is challenging, and the synthesis quality is far below the unconstrained GANs, such as ProgressiveGAN and StyleGAN. Instead, our InterFaceGAN is able to turn unconditional high-fidelity GANs into controllable GANs by identifying latent semantic subspaces using widely available off-the-shelf facial attribute predictors.
Figure 12: Expression manipulation from calm (top) to smile (bottom) on ProgressiveGAN.

Figure 13: Age manipulation from young (top) to old (bottom) on ProgressiveGAN.
Figure 14: Gender manipulation from female (top) to male (bottom) on ProgressiveGAN.

Figure 15: Eyeglasses manipulation from no eyeglasses (top) to wearing eyeglasses (bottom) on ProgressiveGAN.
B. Conditional Manipulation Results

In this section, we evaluate the proposed conditional manipulation approach. Previously, we train a linear SVM classifier, to find the separation boundary with respect to a particular attribute. However, these boundaries might not be accurate enough for two reasons. First, the pre-trained auxiliary attribute model, which is used to assign attribute score for each synthesis, might not be very accurate on synthesized images since they are previously trained on real images. This effect can be eliminated, to some extent, by only using extremely-high-score samples (positive) and extremely-low-score samples (negative) for boundary searching. Second, there are some correlations of two or more attributes in training data. For example, old people are more likely to wear eyeglasses than young people, and there are more males than females in elderly people from training data. In this case, boundaries will overlap with each other. For example, there may exist a boundary targeting at separating old male from young female.

Recall the boundary correlation matrix in Tab.2. It shows that gender, age, and eyeglasses are correlated to each other to some degree. Due to this reason, we found that for some syntheses, if we move the corresponding latent code towards the age boundary, the gender also changes, as shown in Fig.17. Such situation sometimes happens on eyeglasses manipulation, as shown in Fig.18, where the people incline to getting old when wearing eyeglasses.

To decorrelate the attributes for better disentangled manipulation, we propose a conditional manipulation method to edit one attribute yet preserve another through projection. For example, we want to make the person turn older while keep gender the same. After getting the age and gender boundaries, we project the normal direction with respect to age onto that of gender, and find a new direction by subtracting the projection from age direction. In this way, when a latent code moves along the new direction, the gender component is barely affected. Extensive results of conditional manipulation are shown in Fig.17 and Fig.18. After introducing such projected linear transformation, the manipulation results are more satisfying.

C. StyleGAN Discussion

In this section, we analyze the latent spaces of a very recent model, StyleGAN [18]. Compared to traditional GANs, StyleGAN proposed a style-based generator, which first maps the latent code (random noise) from latent space $Z$ to a disentangled latent space $W$ before applying it for generation. We perform InterFaceGAN on both of these two spaces and then analyze their differences.
Figure 17: Conditional manipulation results with ProgressiveGAN. Top row shows age changing with the original boundary. Bottom row shows age changing with the projected boundary, where gender is preserved as the same.

Figure 18: Conditional manipulation results with ProgressiveGAN. Top row shows eyeglasses changing with the original boundary. Bottom row shows eyeglasses changing with the projected boundary, where age is preserved as the same.
C.1. Latent Space Separation

First, we trained several SVM classifiers to find the separation boundaries in latent space. Recall that we assume the latent space to be normally distributed in the main paper. However, even though $W$ space does not fulfill this condition, we can also do similar analysis on $W$ space, demonstrating the generalization ability of InterFaceGAN.

Table 4 shows the comparison results between different latent spaces, including $Z$ space of ProgressiveGAN [17], $Z$ space of StyleGAN [18], and $W$ space of StyleGAN. We can see that $W$ space indeed achieves stronger disentanglement compared to $Z$ space in StyleGAN. According to [18], $W$ space is not subject to some fixed distribution, making it more flexible to biased data distribution. Our experiments in Tab.4 also affirm this statement.

However, it turns out that $Z$ space of StyleGAN exhibits much weaker disentanglement than $Z$ space of ProgressiveGAN. There are probably two reasons resulting in this phenomenon. First, ProgressiveGAN model we used is trained on CelebA-HQ [17] dataset, while StyleGAN model is trained on FF-HQ [18] dataset. These two datasets may have different distributions from attribute aspect, leading to different learning difficulties. Second, the style-based generator may focus on learning the disentangled $W$ space, but put less effort on mapping $Z$ space to $W$ space. Even so, $Z$ space also encodes some semantics according to Tab.4.

C.2. Latent Space Correlation

We then analyze the correlation between different attributes. Here, same as in the main paper, we compute the correlation matrix from two different aspects: (i) attribute distribution observed from the synthesized data, and (ii) normal vector similarity between two attribute boundaries. Tab.5 shows the computation results by directly using the attribute scores of synthesized images, while Tab.6 and Tab.7 present the results corresponding to attribute boundaries in $Z$ space and $W$ space respectively.

As mentioned in the main paper, these two metrics are consistent with each other on ProgressiveGAN model. However, such consistency is not applicable for StyleGAN. For $Z$ space, boundaries are not well disentangled. For instance, eyeglasses and smile barely correlate with each other in Tab.3, but are highly coupled in Tab.6. This is also illustrated in Sec.C.1. For $W$ space, almost all attributes are orthogonal to each other in Tab.7. In other words, $W$ space captures the attribute correlation in real data for training, and then encodes such relationship as a new “style”. For example, “man with eyeglasses” may be considered as a coupled style in $W$ space instead of two independent attributes, i.e., “male” and “eyeglasses”. In this case, it is hard to apply the conditional manipulation trick, described in Sec.B, to reveal the hidden disentanglement between two attributes.

C.3. Comparison Results

We further visualize some editing results by manipulating the latent codes in both $Z$ space and $W$ space. We have the following observations from Fig.19.

(i) InterFaceGAN works well on StyleGAN, which employs a style-based generator. We can edit particular attribute by moving the latent code along certain direction in either $Z$ space or $W$ space.
Figure 19: Attribute editing results on StyleGAN model without retraining. For each attribute, top row shows the manipulation results with respect to $\mathcal{Z}$ space, whilst bottom row corresponds to $\mathcal{W}$ space. Images in red dashed box represent original syntheses. Images between two black dashed lines stand for near-boundary manipulation, and other images stand for long-distance manipulation.
(ii) By learning from a more diverse dataset, FF-HQ [18], GAN model learns the semantics more thoroughly. For example, StyleGAN can even generate children when making people younger. This is beyond the ability of ProgressiveGAN, which is trained on CelebA-HQ [17]. Also, StyleGAN is capable of producing faces with extreme poses.

(iii) \( \mathcal{W} \) space learns better disentanglement than \( \mathcal{Z} \) space, especially for long-distance manipulation. In other words, when the latent code locates near the separation boundary (inside two dashed lines), manipulations in \( \mathcal{Z} \) and \( \mathcal{W} \) space have similar effect. However, when the latent code goes further from the boundary, manipulation in \( \mathcal{Z} \) space will affect other attributes. Taking gender editing as an example, the person in red box takes off his eyeglasses when moving along the gender direction. Compared to \( \mathcal{Z} \) space, \( \mathcal{W} \) space shows stronger robustness.

(iv) Some attributes are correlated to each other due to the inaccurate boundary. For example, people are tending to become happier when being feminized (third sample), and people are wearing eyeglasses when turning old (second sample). These observations are consistent with the results in Tab.5. As discussed in Sec.C.2, such correlations may be considered as new styles in \( \mathcal{W} \) space.

D. Proof

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

**Property 2** Given \( n \in \mathbb{R}^d \) with \( n^T n = 1 \), which defines a hyperplane, and a multivariate random variable \( z \sim \mathcal{N}(0, I_d) \), we have \( P(|n^T z| \leq 2\alpha \sqrt{\frac{d}{d-2}} \geq (1 - 3e^{-c\alpha d})/(1 - 2\alpha e^{-\alpha^2/2}) \) 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 \( n \) to be the first coordinate vector. Accordingly, it suffices to prove that \( P(|z_1| \leq 2\alpha \sqrt{\frac{d}{d-2}} \geq (1 - 3e^{-c\alpha d})/(1 - 2\alpha e^{-\alpha^2/2}) \), where \( z_1 \) denotes the first entry of \( z \).

As shown in Fig.20, let \( H \) denote the set

\[
\{ z \sim \mathcal{N}(0, I_d) : ||z||_2 \leq 2\sqrt{d}, |z_1| \leq 2\alpha \sqrt{\frac{d}{d-2}} \},
\]

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 \( P(H) \geq (1 - 3e^{-c\alpha d})/(1 - 2\alpha e^{-\alpha^2/2}) \).

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

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

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

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

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

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

\[
P(H) = (1 - 2\alpha e^{-\alpha^2/2}) \int_0^{2\sqrt{d}} f(r) dr
\]

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

Q.E.D.

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

Let \( U \) denote the set \( \{ z \in \mathbb{R}^d : \|z\|_2 = 1, z_1 \geq \frac{\alpha}{\sqrt{d-2}} \} \), and \( K \) denote the set \( \{ z \in \mathbb{R}^d : \|z\|_2 = 1, z_1 \geq 0 \} \). It suffices to prove that the surface of \( U \) area and the surface of \( K \) in Fig. 21 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

\[
surf(U) = \int_{\sqrt{d-2}}^1 (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1
\]
\[
\leq \int_{\sqrt{d-2}}^1 e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1
\]
\[
\leq \frac{1}{\alpha} \int_{\sqrt{d-2}}^1 z_1 \sqrt{d-2} e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1
\]
\[
\leq \frac{1}{\alpha} \int_{\sqrt{d-2}}^\infty z_1 \sqrt{d-2} e^{-\frac{d-2}{2} z_1^2} A(d-1) dz_1
\]
\[
= \frac{A(d-1)}{\alpha \sqrt{d-2}} e^{-\alpha^2/2}.
\]

Similarly, we have

\[
surf(K) = \int_0^1 (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1
\]
\[
\geq \int_0^{\sqrt{d-2}} (1 - z_1^2)^{\frac{d-2}{2}} A(d-1) dz_1
\]
\[
\geq \frac{1}{\sqrt{d-2}} \left( 1 - \frac{1}{d-2} \right) A(d-1).
\]

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

\[
surf(K) \geq \frac{1}{\sqrt{d-2}} \left( 1 - \frac{1}{d-2} \right) A(d-1)
\]
\[
\geq \frac{1}{\sqrt{d-2}} \left( 1 - \frac{d-2}{d-2} \right) A(d-1)
\]
\[
= \frac{A(d-1)}{2\sqrt{d-2}}.
\]

Accordingly,

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

Q.E.D.