EECS 545 Final Report: ICA-SeFa - A New Efficient Algorithm to do Latent Semantics Factorization in GANs

Houming Chen, Le Qin, Yutong Bi, Jiaxi Chen, Dongyang Zhao
University of Michigan
{houmingc, leqin, yutongbi, jxichen, zhaoleo}@umich.edu

Figure 1: Visualization of semantic directions found by applying ICA-SeFa on several different GAN models [20, 27]. For each group of images, the middle image is an image generated by the GAN model with a random latent code. The left and right images are images generated by the GAN model with new latent codes which are given by moving the original latent code toward and backward the directions found by ICA-SeFa. The notes below are the possible semantic directions corresponds to that movement. Models used in the moving process are in the brackets.

Abstract

Studies have shown that there are often semantically meaningful directions in the latent spaces of GANs. Looking for these semantic directions in the latent space is of great significance both in the theoretical studies and application of GANs. Some recent approaches have focused on using unsupervised algorithms to identify those directions. While most of these unsupervised algorithms are time-consuming, one very recent study proposes a state-of-the-art unsupervised algorithm called Semantic Factorization(SeFa) [36], which focuses on using only the pre-trained weights GANs to find semantics directions, making unsupervised latent semantics factorization much more efficient.

In this project, we deeply analyzed the underlying mechanism of SeFa and proposed a more complete theory to explain the excellent performance of SeFa. Motivated by the theory, we proposed a new unsupervised algorithm, ICA-SeFa, to find interpretable directions based on using ICA to decompose the pre-trained weights of GANs. Like SeFa, our algorithm only uses pre-trained weights and is computationally efficient. Experiments have shown that ICA-SeFa can find semantic directions with much better interpretabil-

ity than the directions discovered by SeFa.

1. Introduction

Generative Adversarial Networks (GANs) [3, 11, 19, 20, 21, 33], including some state-of-the-art models like BigGan [3] and StyleGan [20], have achieved great success in image synthesis. Thanks to their excellent generation qualities, GANs are widely employed in various tasks like image-to-image translation [17, 45], text-to-image translation [42], super-resolution [24], and many others. However, although GANs have reached many impressive achievements, the interpretability of GANs is still a tough question [1, 18, 20, 31, 40]. One approach to examining the interpretability of GANs is to investigate the relationship between the generated images and the latent codes. Recent studies have observed that well-trained GANs spontaneously encode many interpretable directions inside their latent space [10, 18, 31, 34, 36, 37, 40], and moving the latent codes along these directions would accordingly make human interpretable modifications to the generated image, like smiles and glasses on generated human faces [33], zooming and translation of the generated objects [18, 31], and lighting condition of the generated scenes [40]. These directions can be refered as *semantic directions* [36] (or *interpretable directions* in [37]), and identifying semantic directions in well-trained GANs has great importance in studying the interpretability of GANs [37].

Identification of semantic directions not only contributes to the interpretablity of GANs but also provides great values in practical applications. Semantic directions can control the generating process of GANs, enabling scene editing [40, 44], face editing [12, 35], and many other applications.

Nevertheless, finding semantic directions is challenging because of the latent space's high dimensionality and the immense diversity of image semantics [36]. Most of the prior works use supervised approaches [10, 18, 20, 31, 35, 40], which share some common limitations. In order to overcome these limitations, three unsupervised methods are proposed recently [13, 36, 37]. While the two early methods are computationally expensive as they require neural network training [37] or large-scale sampling [13], the latest approach, *Semantic Factorization* (*SeFa*) [36], can be computed in an efficient, closed-form way. Independent of neural network training or sampling, SeFa merely uses the pre-trained weights of the first layer of the generator, making semantic directions finding a lot more efficient [36].

In this project, we analyzed the underlying mathematical mechanism of SeFa and proposed a new theory to explain the excellent performance of SeFa in semantic direction discovery. Based on these theories, we proposed a new algorithm called ICA-SeFa. On the basis of SeFa, ICA-SeFa applies independent component analysis(ICA) [7, 16] to decompose the weights of the first layer of Gan generator. Like SeFa, ICA-SeFa only uses the pre-trained weights of the first layer of the generator and can be also computed in a very efficient way with the help of FastICA[16]. Moreover, various experiments showed that ICA-SeFa can find better semantic directions than SeFa. As directions found by SeFa might affect multiple semantic attributes, ICA-SeFa provides more disentangled and clear directions.

2. Related Works

2.1. Interpretable Disentangled Representation

Interpretable disentangled representation, including some state-of-the-art models like β-VAE [15], FactorVAE [22], β-TCVAE [5], and infoGAN [6], is another approach that can be used to study the controllability and interpretability of generative models [5, 6, 15, 22, 37]. The original goal of disentangled representation is to find a representation of a data domain, in which one latent unit represents to only one generative factors [2, 5, 6, 15, 22, 37]. Moreover, many algorithms also aim to find disentangled representations that each latent unit encodes some interpretable attribute factor of the represented data [5, 6, 15, 22].

However, finding interpretable control of the generating process is not the only goal for the interpretable disentangled representation task [5, 6, 15, 22]. First, they have to be a representation [2], which means they not only have a decoder(generator) that can generate images based on latent codes, but also have an encoder that can encode a real image to a latent code that could be used to generate a similar image [5, 6, 15, 22]. Second, the founded latent units should be disentangled, meaning that interpretable disentangled representation values not only the interpretablility of the found latent units but also the statistical independence of those latent units [15, 41].

Also, the generation qualities of the generators in these disentangled representation methods are usually not comparable to the qualities of current state-of-the-arts GANs [37]. Therefore, it is still important to discover semantic directions in pre-trained models of current state-of-the-art GANs [37], like BigGan [3] and StyleGan [20].

2.2. Supervised methods in Semantic Direction Discovery

Supervised semantic discovery algorithms [10, 18, 20, 31, 35, 40] often sample a large number of latent codes and generate images with these latent codes. Then, pseudo-labels will be given to the produced images and their corresponding latent codes. Finally, a supervised models will be trained on the labeled latent codes. Since sampling and model training are usually computationally expensive, these supervised approaches all share the common weakness that they are time-consuming.

In addition, these supervised algorithms can be further divided into two categories, and each category has its own defects. The first type of supervised algorithms produces pseudo-labels by pre-trained models. For example, some studies [20, 35] use predictors pre-trained on the CelebA dataset [25] to predict certain semantic attributes of the

faces generated by GANs, and another research [40] uses predictors pre-trained on multiple databases [23, 29, 39, 43] to predict semantic attributes of synthesized scenes. After labeling the attributes to latent codes, a supervised model could be trained to discover the relationship between the latent codes and the generated images' semantic attributes. The major limitation of this type of algorithm is that they are highly dependent on the pre-trained networks [36, 37]. The second type of supervised algorithms relies on some statistical or self-supervised approaches to produce pseudo-labels [18, 31]. However, this type of algorithms could only handle some simple semantics that can be obtained automatically, like basic zooming and translation [37].

2.3. Unsupervised Methods in Semantic Direction Discovery

As supervised algorithms have those limitations, some recent studies focus on finding semantic directions with unsupervised algorithms [13, 36, 37]. These unsupervised algorithms do not require pre-trained predictors and could still find detailed semantics like glasses [37] and facial expressions [13, 36]. One approach [37] trained a reconstructor network could recognize semantic directions from a set of candidate directions [37]. Another study proposed an algorithm called GanSpace [13] which conducts principal component analysis(PCA) [30] on the latent space of StyleGAN [20] and the feature space of BigGAN [3] to get semantic directions. However, these two methods are still time-consuming because of the high computational complexity of networks training and large-scale sampling [36].

In order to do discover semantic directions efficiently, a recent study has proposed SeFa [36], an unsupervised closed-form latent semantics factorization algorithm that does not require any kind of training or sampling. Being a closed-form algorithm, SeFa [36] can do semantics factorization in a highly efficient manner. It has state-of-the-art performances on a variety of GANs [36], including PGGAN [19], StyleGAN [20], and BigGAN [3].

3. Motivations of ICA-SeFa

3.1. Preliminaries

In GANs, the generator $G(\cdot)$ is a neural network that maps from the latent space $\mathcal{Z} \subset \mathbb{R}^d$ to the image space $\mathcal{I} \subset \mathbb{R}^{H\times W\times C}$, where H, W, C respectively denotes the height, width, and number of channels of the generated images.

3.2. SeFa [36]

As a neural network, $G(\cdot) : \mathcal{Z} \to \mathcal{I}$ transforms an input latent vector $\boldsymbol{z} \in \mathcal{Z}$ to an image step by step[36]. The first step of the generator is to multiply the input latent vector $\boldsymbol{z} \in \mathcal{Z}$ by a weight matrix A. Then, the second step of the generator is to add a bias b on the result vector $A\boldsymbol{z}$. The original motivation of SeFa focused on these two steps of the network, as they directly acts on the latent space [36]. [36] defined these two steps as a function G_1, which is

$$G_1(\boldsymbol{z}) \triangleq A\boldsymbol{z} + b \tag{1}$$

where $z \in \mathcal{Z}$ and A and b are the weight and bias used in the first layer of generator [36].

If a direction $\boldsymbol{n} \in \mathbb{R}^d$ is a semantic direction, one can manipulate the target semantic by adding a multiple of an identified direction n on the latent code z [36]. Then, consider the first step of the generator (1), one will have

$$G_1(\boldsymbol{z} + \alpha\boldsymbol{n}) = A(\boldsymbol{z} + \alpha\boldsymbol{n}) + b \tag{2}$$
$$= G_1(\boldsymbol{z}) + \alpha A\boldsymbol{n} \text{ [36]} \tag{3}$$

The original motivation of SeFa is to find the directions that can cause largest variations to $G_1(z + \alpha n)$. Therefore, when it comes to finding k most important directions $\{n_1, n_2, ..., n_k\}$, according to (3), SeFa only need to solve

$$\boldsymbol{N}^* = \underset{\{\boldsymbol{N}\in\mathbb{R}^{d\times k}, \boldsymbol{n}_i^T\boldsymbol{n}_i=1\forall i=1,...,k\}}{\operatorname{argmax}} \sum_{i=1}^{k} \|A\boldsymbol{n}_i\|_2^2 \tag{4}$$

Where $\boldsymbol{N}^*$ is the matrix of $[\boldsymbol{n}_1\ \boldsymbol{n}_2\ ...\ \boldsymbol{n}_k]$ [36]

By solving (3), one can know that the k directions $\{\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_k\}$ found by SeFa is exactly the eigenvectors of A^TA associated with the k largest eigenvalues. [36].

3.3. A New Theory about SeFa

SeFa shows excellent performance in finding versatile semantic directions in the latent space of GANs [36]. However, as the original motivation of SeFa is only to maximize the changes after appyling G_1 on the latent code [36], there is still a logical gap about why directions found by SeFa would lead to interpretable changes to the generated image. Therefore, here we proposed a more complete explanation for the excellent performance of SeFa.

When the generator is used to generate images, the latent code $\boldsymbol{z} \in \mathcal{Z}$ will be randomly chosen from the latent space $\mathcal{Z}$. Therefore, for a randomly generated images, the input to the generator is actually a random vector $\boldsymbol{Z}$, whose sample space is the latent space $\mathcal{Z}$.

Then, the first step of the generator is to transform the random vector $\boldsymbol{Z}$ to another random vector $\boldsymbol{Y}$ by the linear transformation given by matrix the A, that is,

$$\boldsymbol{Y} \triangleq A\boldsymbol{Z} \tag{5}$$

After the first step, the random vector $\boldsymbol{Y}$ will be transformed into a generated image in the later stages of generator. Therefore, every entry of $\boldsymbol{Y}$ will contribute to some attributes of the generated image (although these attributes might not be interpretable).

Let the dimension of $\boldsymbol{Y}$ be m. Since $\boldsymbol{Y}$ is a random vector, then the entries of $\boldsymbol{Y}$ will be random variables $\{Y_1, Y_2, ..., Y_m\}$. That is, $\boldsymbol{Y} = [Y_1\ Y_2\ ...\ Y_m]$. Therefore, each entry Y_i will contribute to some attributes of the final generated image. However, these random variables are not

necessarily independent, so there might be some relations between these random variables. Then, the key idea of our theory is: *we assume that a reasonable, meaningful edition to the image generation process should be consistent with the relations among* $\{Y_1, Y_2, ..., Y_m\}$.

For example, there might be some equal relations between those random variables, like $Y_1 = 2Y_2$. Then, if we want to edit the image generation process, we should make sure that this relation still holds after our edition. However, in general cases, there can hardly be any exactly equal relations like $Y_1 = 2Y_2$. Even if there are some equal relations, we do not need to worry about them since simply editing the latent code won't break those equal relation. However, there might be some *approximate relations* between $\{Y_1, Y_2, ..., Y_m\}$. For example, we might found that the value of y_1 might be always closed to $2Y_2$. Here we denote such relations as $Y_1 \approx 2Y_2$. If we want to make a reasonable, meaningful edition to the image generation process, we should also be consistent with these approximate relations.

In fact, *there can be multiple ways to define the approximate relations between* $\{Y_1, Y_2, ..., Y_m\}$, and different definitions would lead to different methods to find semantic directions. In our theory, the reason that SeFa could produce interpretable directions is because SeFa is good at finding *linear approximate relations*.

Here we define the **linear approximate relations**. The goal is to find a random variable Y_0 and a unit vector $\hat{t}$ such that $\boldsymbol{Y} \approx \hat{\boldsymbol{t}} Y_0$. Then, we referred $\boldsymbol{Y} \approx \hat{\boldsymbol{t}} Y_0$ as the *linear approximate relations* of $\{Y_1, Y_2, ..., Y_m\}$. Here, $\hat{\boldsymbol{t}}$ is defined as

$$\hat{\boldsymbol{t}} = \underset{\hat{\boldsymbol{t}} \in \mathbb{R}^m, |\hat{\boldsymbol{t}}|=1}{\operatorname{argmax}} \ Var(\hat{\boldsymbol{t}}^T \boldsymbol{Y}) \tag{6}$$

This approximate relation suggests that a reasonable edition to the generating process should move Y along the direction of $\hat{t}$.

The intuition is very simple. We know that

$$cov(Y_i, Y_j) = cor(Y_i, Y_j)\sqrt{var(Y_i)var(Y_j)} \tag{7}$$

for every $i, j \in \{1, .., m\}, i \neq j$. Therefore, when random variables in $\{Y_1, Y_2, ..., Y_m\}$ have strong linear correlations, $cov(Y_i, Y_j)$ will be closed to $\sqrt{var(Y_i)var(Y_j)}$. For the extreme case, if there is really a random variable y_0 and a unit vector t such that $\boldsymbol{Y} = \boldsymbol{t} Y_0$, we will have

$$Var(\hat{\boldsymbol{t}}^T \boldsymbol{Y}) = \sum_{i=1}^{m} Var(\hat{\boldsymbol{t}}_i Y_i) \tag{8}$$

$$= \sum_{i=1}^{m} \hat{\boldsymbol{t}}_i^2 Var(Y_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} \hat{\boldsymbol{t}}_i \hat{\boldsymbol{t}}_j cov(Y_i, Y_j) \tag{9}$$

$$= (\sum_{i=1}^{m} \hat{\boldsymbol{t}}_i \sqrt{Var(Y_i)})^2 \tag{10}$$

$$= (\sqrt{Var(Y_0)} \sum_{i=1}^{m} \hat{\boldsymbol{t}}_i^T \boldsymbol{t}_i)^2 \tag{11}$$

Then, by restricting $|\hat{\boldsymbol{t}}| = 1$, $Var(\hat{\boldsymbol{t}}^T \boldsymbol{Y})$ reaches maximum when $\hat{\boldsymbol{t}} = \boldsymbol{t}$.

If one can calculate (6) and find $\hat{\boldsymbol{t}}$, then the according semantic direction $\boldsymbol{n}$ induced by *linear approximate relation* will be a vector satisfying $A(\boldsymbol{Z} + \boldsymbol{n}) = \boldsymbol{Y} + c\hat{\boldsymbol{t}}$ where $c \in \mathbb{R}$. Then since $\hat{\boldsymbol{t}}$ is a unit vector, by (5), we have

$$\hat{\boldsymbol{t}} = \frac{A\boldsymbol{n}}{|A\boldsymbol{n}|} \tag{12}$$

In order to calculate (6), one need to know the distribution of $\boldsymbol{Y}$. However, it is very difficult to know the real distribution of $\boldsymbol{Y}$ without large-scale sampling, but large-scale sampling will make the algorithm computationally inefficient [36]. Therefore, one can find some alternative method to estimate the distribution of $\boldsymbol{Y}$.

According to (5), (6) can be changed to

$$\hat{\boldsymbol{t}} = \underset{\hat{\boldsymbol{t}} \in \mathbb{R}^n, |\hat{\boldsymbol{t}}|=1}{\operatorname{argmax}} \ Var(\hat{\boldsymbol{t}}^T A\boldsymbol{Z}) \tag{13}$$

Then, by (12)

$$\boldsymbol{n} = \underset{\boldsymbol{n} \in \mathbb{R}^d, |\boldsymbol{n}|=1}{\operatorname{argmax}} \ Var((\frac{A\boldsymbol{n}}{|A\boldsymbol{n}|})^T A\boldsymbol{Z}) \tag{14}$$

Then we do an approximation, we can assume entries of $\boldsymbol{Z}$ are independent random variables, and each of them has the same variance σ^2. This might be always true for some simple GANs, but it is not generally true for some advanced GAN like StyleGan[20] which would embed the initial random input into a latent space. However, since the embedding process are usually done by some very simple networks which won't increase the dimension of the initial input, we can still approximately regard the entries of $\boldsymbol{Z}$ being independent random variables with the same variances.

Then, (14) will become

$$\boldsymbol{n} = \underset{n\in\mathbb{R}^d,|\boldsymbol{n}|=1}{\operatorname{argmax}} |(\frac{A\boldsymbol{n}}{|A\boldsymbol{n}|})^T A|\sigma^2 \tag{15}$$

$$= \underset{\boldsymbol{n}\in\mathbb{R}^d,|\boldsymbol{n}|=1}{\operatorname{argmax}} \frac{|A^T A\boldsymbol{n}|}{|A\boldsymbol{n}|} \tag{16}$$

$$= \underset{\boldsymbol{n}\in\mathbb{R}^d,|\boldsymbol{n}|=1}{\operatorname{argmax}} \sqrt{\frac{(\boldsymbol{n}^T A^T A\boldsymbol{n})(\boldsymbol{n}^T A^T A\boldsymbol{n})}{\boldsymbol{n}^T A^T A\boldsymbol{n}}} \tag{17}$$

$$= \underset{\boldsymbol{n}\in\mathbb{R}^d,|\boldsymbol{n}|=1}{\operatorname{argmax}} |A\boldsymbol{n}| \tag{18}$$

Then, according to our arguments, this $\boldsymbol{n}$ will be the direction that is most consistent with the *approximate linear relations* of $\{Y_1, ..., Y_m\}$. Therefore, it is likely to be a semantic direction.

After we found one direction $\boldsymbol{n}_1$ with this method, we can actually continue to proceed the similar process to find more directions $\boldsymbol{n}_2, \boldsymbol{n}_3, ..., \boldsymbol{n}_k$ that are consistent with the linear relations of $\{Y_1, ..., Y_m\}$. However, we hope that the founded semantic directions are disentangled, which means changing one semantic attribute should not affect other semantic attributes. Therefore, we need to make sure that all the founded directions should be orthogonal to each other, so that if moving the latent code along one direction won't cause changes in other directions. Therefore, after we have found i directions, the $(i+1)$th direction should be orthogonal to the previous directions, that is,

$$\boldsymbol{n}_i = \underset{\boldsymbol{n}_i\in\mathbb{R}^d,|\boldsymbol{n}_i|=1,\boldsymbol{n}_i\perp\boldsymbol{n}_j\forall j\in\{1,...,i-1\}}{\operatorname{argmax}} |A\boldsymbol{n}| \tag{19}$$

Then, (19) is actually one version of principle component decomposition [30, 38], which can be solved by singular value decomposition. By solving (19), we can simply find that $\boldsymbol{n}_1, ..., \boldsymbol{n}_k$ are just the eigenvectors of $A^T A$ associated with the k largest eigenvalues [38]. This result is exactly same as the algorithm proposed in SeFa [36].

3.4. Apply ICA to Explore More Complicated Relations

The good performance of SeFa showed that fitting the linear relationship among $\{Y_1, Y_2, ..., Y_m\}$ could discover interpretable directions. However, there might be more complicated relations among $\{Y_1, Y_2, ..., Y_m\}$. Therefore, Therefore, we hope that $\boldsymbol{Y}$ is actually a linear combination of some independent random variables. That is,

$$W\boldsymbol{Y} = \boldsymbol{S} \tag{20}$$

where entries of $\boldsymbol{S}$ are independent. Then, a reasonable edition to the generating process should move $\boldsymbol{Y}$ along columns of W^{-1}. In this case, no relations (no matter linear or none-linear) among $\{Y_1, Y_2, ..., Y_m\}$ would be broken because of the independence of entries of $\boldsymbol{S}$. Moreover, when components of $\boldsymbol{S}$ are independent, rows of W will be orthogonal with each other, so directions along the columns of W^{-1} will also be directions along the rows of W. However, it might be impossible to find such W. Nevertheless, we can approximate this W by maximizing the non-Gaussianity of the components of $\boldsymbol{S}$ [16].

Therefore, our goal is to find k multiple directions, we need to find $\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, ...\hat{\boldsymbol{w}}_k$ by maximizing the non-Gaussianity of $\hat{\boldsymbol{w}}_1\boldsymbol{Y}, \hat{\boldsymbol{w}}_2\boldsymbol{Y}, ...\hat{\boldsymbol{w}}_k\boldsymbol{Y}$ [16]. Then, meaningful editions of the generating process should be moving $\boldsymbol{Y}$ along the directions of $\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, ...\hat{\boldsymbol{w}}_k$. However, this is still a hard question as we don't know the distribution of $\boldsymbol{Y}$. However, by (5), for a vector $\hat{\boldsymbol{w}}$, we have

$$\hat{\boldsymbol{w}}Y = \hat{\boldsymbol{w}}A\boldsymbol{Z} \tag{21}$$

Like in 3.3, we found that we can actually give a reasonable estimation of the non-Gaussianity of $\hat{\boldsymbol{w}}A\boldsymbol{Z}$ by using only using $\hat{\boldsymbol{w}}A$.

For example, kurtosis can be a good measure for non-Gaussianity [16], and if we assume that the entries of $\boldsymbol{Z} =$

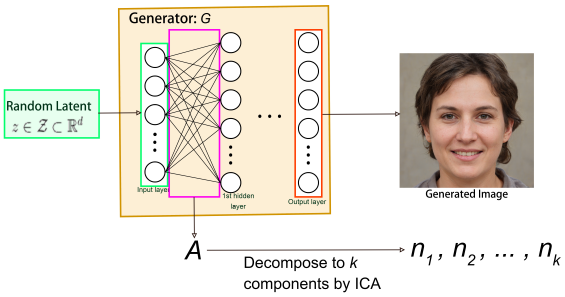

Figure 2: Visualization of procedures of ICA-SeFa. We used FastICA to decompose the first layer weight A to independent components $\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_k$, which are just the found semantic directions.

$[Z_1, ..., Z_d]$ are independent, then

$$Kurt(\hat{\boldsymbol{w}}A\boldsymbol{Z}) - 3 \tag{22}$$

$$= \frac{\sum_{i=1}^{d}(\sum_{j=1}^{m}\hat{\boldsymbol{w}}_i A_{i,j})^4 Var(Z_i)^2 (Kurt(Z_i) - 3)}{\left[\sum_{i=1}^{d}(\sum_{j=1}^{m}\hat{\boldsymbol{w}}_i A_{i,j})^2 Var(Z_i)\right]^2} \tag{23}$$

If entries of $\boldsymbol{Z}$ have the same variance σ^2 and kurtosis κ, and assume that the $\hat{\boldsymbol{w}}A$ is unbiased, i.e., $E(\hat{\boldsymbol{w}}A) = 0$, then

$$\underset{\hat{\boldsymbol{w}}\in\mathbb{R}^n, |\hat{\boldsymbol{w}}|=1}{\operatorname{argmax}} \; Kurt(\hat{\boldsymbol{w}}AZ) \tag{24}$$

$$= \underset{\hat{\boldsymbol{w}}\in\mathbb{R}^n, |\hat{\boldsymbol{w}}|=1}{\operatorname{argmax}} \frac{\sum_{i=1}^{m}(\sum_{j=1}^{d}\hat{\boldsymbol{w}}_i A_{i,j})^4 \sigma^4 (\kappa - 3)}{\left[\sum_{i=1}^{m}(\sum_{j=1}^{d}\hat{\boldsymbol{w}}_i A_{i,j})^2 \sigma^2\right]^2} \tag{25}$$

$$= \underset{\hat{\boldsymbol{w}}\in\mathbb{R}^n, |\hat{\boldsymbol{w}}|=1}{\operatorname{argmax}} \frac{\sum_{i=1}^{m}(\sum_{j=1}^{d}\hat{\boldsymbol{w}}_i A_{i,j})^4}{\left[\sum_{i=1}^{m}(\sum_{j=1}^{d}\hat{\boldsymbol{w}}_i A_{i,j})^2\right]^2} \tag{26}$$

$$= \underset{\hat{\boldsymbol{w}}\in\mathbb{R}^n, |\hat{\boldsymbol{w}}|=1}{\operatorname{argmax}} \; Kurt(\hat{\boldsymbol{w}}A) \tag{27}$$

Therefore, approximately, the non-Gaussianity of $\hat{\boldsymbol{w}}Y$ will be maximized when we maximize the non-Gaussianity of $\hat{\boldsymbol{w}}A$.

Then, if we want to find k semantic directions, we need to find $\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, ...\hat{\boldsymbol{w}}_k$ by maximizing the non-Gaussianity of $\hat{\boldsymbol{w}}_1 A, \hat{\boldsymbol{w}}_2 A, ...\hat{\boldsymbol{w}}_k A$.

4. Method

We use the notation defined in the previous section 3. Then, according to the theory we proposed in 3.4, we only need to find a $k \times m$ matrix W such that columns of WA have large non-Gaussianity. Then, the semantic directions will just be rows of W.

Indeed, this is exactly the same objective as performing FastICA on A [16], which aimed to decompose the matrix A to independent components $\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_k$ [16].

Therefore, we can just perform FastICA on the matrix A, and the found semantic directions will just be the result independent components $\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_k$. A visualization of the algorithm is shown in Figure 2. Since FastICA can be computed very efficiently [16], ICA-SeFa can also be computed in a very efficient manner.

5. Evaluation Methods

5.1. Evaluation Methods for Related Tasks

Interpretability of semantic directions is difficult to measure [4, 37]. Compared to similar tasks, quantitative evaluation for the interpretability of the semantic directions found by unsupervised algorithms is still a hard problem [13, 36, 37]. To illustrate the great difficulties of evaluating unsupervised semantic direction discovery compared to related tasks, we first give a brief review on the evaluation methods for related tasks and describe the difficulties of using similar techniques to evaluate unsupervised semantic direction discovery.

5.1.1 Evaluation Methods for Interpretable Disentangled Representation

There are some good evaluation metrics for interpretable disentangled representation, like Z-diff [15] and MIG [5]. However, one cannot easily apply similar methods to measure the interpretability of semantic directions. First, as stated in section 2.1, interpretability is not the only goal of interpretable disentangled representation, many metrics for disentangled representation measures not only interpretability but also some other aspects like independence of the latent units [15]. Second, those metrics highly relies on the encoder of the representation algorithms [5, 6, 15, 22, 41]. As semantic direction discovery algorithms can provide interpretable modifications for GANs, they do not aim to provide interpretable representation. Unlike from interpretable disentangled representation, semantic direction discovery algorithms does not provide an encoder that can encode an image to an representation using the semantic attributes it finds [13, 36, 37].

5.1.2 Evaluation Methods for Supervised Semantic Direction Discovery

As stated in 2.2, supervised algorithms can be divided into two categories. The first category uses statistical or self-supervised approaches [18, 31]. However, those approaches could only handle simple semantics like basic zooming and translation [18, 31, 37]. Therefore, these algorithms can be evaluated by some simple object detection [18] or evaluated on some labeled dataset that only has some simple variations [31] like dSprite [26]. However, similar methods cannot be used if the algorithm is aimed to find directions that control more complicated semantic attributes.

Another category of algorithms applies pre-trained attribute predictors to generate psudo-labels [20, 35, 40]. These algorithms can be evaluated with the attribute predictors it used. For example, InterFaceGAN [35] was trained under the supervision of a multi-label attribute predictor based on ResNet50 [14] pre-trained on the CelebA dataset [25]. In order to evaluate InterFaceGAN, [35] proposed a method called re-scoring analysis. They sampled 2K latent code samples and then measured the averaged change of each predicted score after moving the latent code along every founded direction for a given distance [35]. Finally, a re-scoring matrix was produced such that the value on ith row and jth column reflects the average change of the jth predicted scores after moving the latent along the ith direction for a given distance [35]. Ideally, the values on the diagonal line of the matrix should have greater absolute values, and other values should be close to zero [35]. Such re-scoring analysis could be modified to evaluate unsupervised model [36] (see 5.2), but it also has a major limitation which will be described in 5.2 in details.

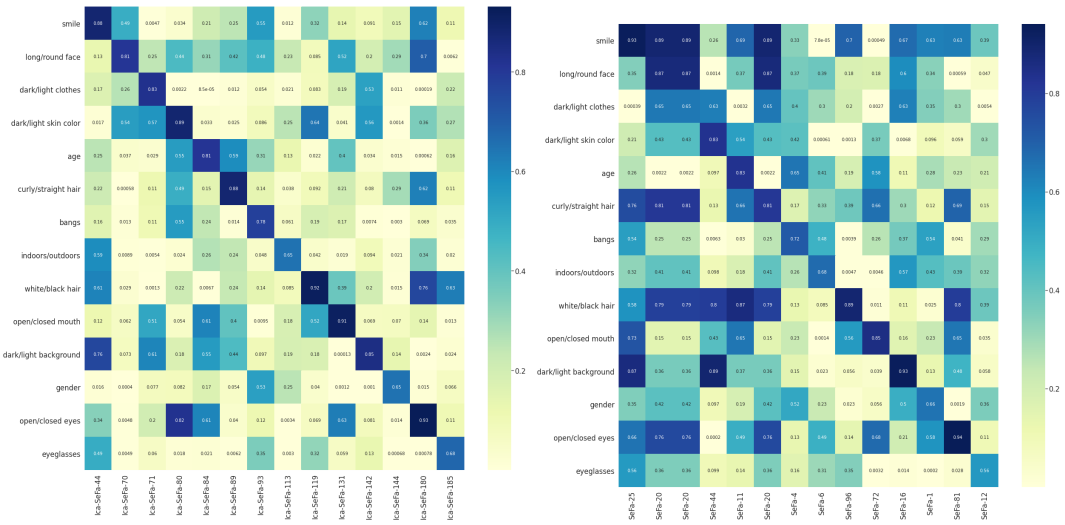

Figure 3: Results of performing **re-scoring analysis based on CLIP [32]** on some selected common directions discovered by ICA-SeFa (left), and SeFa (right). Each row represents a semantic direction, and each column represents a specific direction corresponds to that semantic direction. The value in each cell indicates the average correlation described in 5.3

5.2. Evaluation Methods for Unsupervised Semantic Direction Discovery

Although there are many quantitative evaluation metrics for related areas, the main evaluation method for unsupervised semantic direction discovery is still through visual inspection and qualitative results because of the difficulties to measure interpretability and the unpredictability of unsupervised algorithm. However, some of the previous studies stilled tried to propose numerical evaluation metrics for unsupervised semantic direction discovery [36, 37]. One approach to numerically evaluate the model is through experiments based on human annotation. For example, Voynov and Babenko's study [37] developed an annotation-based evaluation metric called ***mean-opinion-score* (MOS)**. For each direction $\boldsymbol{n}$, MOS sampled 10 random latent codes $\{\boldsymbol{z}_1, ..., \boldsymbol{z}_{10}\}$, and for each latent code $\boldsymbol{z}_i$, images generated by $G(\boldsymbol{z}_i + s\boldsymbol{n})$ were plotted where s are integers from -8 to 8 [37]. Then, some human annotators will be asked to evaluate the direction by looking at the 170 generated images. Each human annotator will be asked to answer a list of questions about the 170 images; then, an opinion score will be calculated based on the answer provided by the annotator[37]. After all of the annotators finish their evaluation, the MOS value is calculated by taking the average from all the opinion scores [37].

Evaluation methods based on human assessment are usually not as preferable as quantitative evaluation methods that could be completely done by programs [4]. Therefore, [36] proposed a quantitative evaluation metric for unsupervised semantic direction discovery, which is called **re-scoring analysis** [36]. Similar to the re-scoring analysis InterfaceGan [35] used, which is mentioned in 5.1.2, this evaluation method applied CelebA dataset[25] to train a multilabel attribute predictor based on Resnet50 [14], and then sampled 2K latent code samples and measures the averaged change of each predicted scores after moving the latent code along every founded direction for a given distance [36]. However, unlike supervised method, unsupervised method discovers semantic directions completely by itself, so the discovered semantic directions do not automatically correspond to the predefined attributes. Therefore, for each labeled attribute, [36] chose the semantic direction that can cause the highest averaged change to the predicted score as the semantic direction corresponds to that labeled attribute. Then, they draw the re-scoring matrix as described in 5.1.2 and evaluate the performance of SeFa in the same way [35].

However, the re-scoring analysis proposed in [35] has a great limitation. Since the attribute predictor was trained on the CelebA dataset[25], it can only identify semantic attributes that are labeled in the CelebA dataset [25]. However, as unsupervised methods discover semantic directions in a completely unsupervised manner, it is highly possible that unsupervised algorithms could find directions that is not labeled in CelebA [25]. Therefore, such re-scoring analysis based on attribute predictors trained on CelebA [25] might not fully evaluate the performance of unsupervised algorithms for semantic direction discovery.

5.3. Re-scoring Analysis Based on CLIP[32]

In order to overcome the limitation of the re-scoring analysis proposed by Shen [36], which is described in 5.2, we proposed a new evaluation method, which we called **re-scoring analysis based on CLIP**.

As joint representations being a popular topic in current deep learning studies, one recent research proposed a model based on Contrastive Language-Image Pre-training (CLIP) [32] which learns a multi-modal embedding space.

The CLIP model can be used to estimate the semantic similarity between a given image and a text. Based on this characteristic, CLIP can do object classification tasks in a zero-shot manner [32], i.e., if the text descriptions of the objects are provided, then the pre-trained CLIP model can classify the images of those objects without extra training [32]. Algorithms based on CLIP had reached state-of-the-art performance in various deep learning tasks [9, 28, 32].

Therefore, in order to measure more attributes found by ICA-SeFa, we built multiple binary attribute predictors based on CLIP. For each attribute that we want to measure, we came up with a pair of binary text expressions to describe that direction. For example, if we want to measure the attribute that decides whether a person wears a pair of glasses, then we come up with two text expressions: "a person wearing glasses" and "a person not wearing glasses". Then, using these two text expressions as labels, we can build a zero-shot binary classifier with the help of CLIP [32]. Then, we can use this binary classifier as the attribute predictor for this specific semantic attribute.

Therefore, unlike the re-scoring analysis based on CelebA dataset [25] which can only evaluate a limited number of semantic attributes, our method can evaluate any attributes as long as human languages can describe them.

Besides building attribute predictors with CLIP instead of training attribute predictors with CelebA dataset [25], we proposed another modification to the original re-scoring analysis [36]. The original re-scoring analysis measures the averaged change of each predicted score after moving the latent code along every founded direction for a given distance [36]. However, we suggest that this averaged change does not have a clear statistical meaning. In fact, the correlation should be a better measure to reflect the relation between the edited distance along the semantic direction and the corresponding predicted attribute score for the generated image. If the edited distance of the latent code along one direction has a high correlation with the score predicted by one attribute predictor, then it suggests that moving the latent along this direction would likely change the semantic attribute predicted by that predictor.

Therefore, in our re-scoring analysis based on CLIP, we sampled 500 random latent codes for each founded direction and manipulated them along the direction with the degree from -7 to 7. Then, we calculated the correlation between the edited distance and the predicted attribute scores. Then, we applied Fisher's z transformations to estimate the average correlation [8] among these 500 groups of data.

6. Experiment Results

As our algorithm is inspired by SeFa [36], which is also an efficient unsupervised semantic discovery algorithm, we performed qualitative comparison, experiments measuring MOS score, and re-scoring analysis based on CLIP to compare our algorithm with SeFa. We experimented the two algorithms on the official StyleGan2 model pre-trained on ffhq dataset [20], and each algorithm was used to find 200 directions.

6.1. Qualitative Results

We applied the semantic directions found by ICA-SeFa and SeFa on the mean latent code of the generator and visualized the output. Then, we selected some common semantic directions found by ICA-SeFa and SeFa. The result images are shown in Figure. 4 in Appendix B.

From the comparison, we can see that although SeFa can find some semantic directions, it cannot separate those semantic attributes well. While moving along one semantic direction, some other attributes are also changing. On the contrary, for ICA-SeFa, one direction only decides one semantic attribute. When moving along the direction, other attributes of the image were preserved well.

6.2. MOS

We recruited 8 annotators to evaluate the performance of ICA-SeFa and SeFa with MOS scores [37]. Since we have a huge number of directions(400) to evaluate, we didn't measure the MOS for all directions and calculate the mean MOS score as described in [37], Instead, we selected 35 directions found by ICA-SeFa and 35 directions found by SeFa that we and the annotators believe shows best interpretability. Then, we measured the MOS scores for the selected direction and ranked them based on their MOS scores.

The top 10 directions found by SeFa and ICA-SeFa are shown in Table 1 and Table 2 in Appendix A. We can see that the top ICA-SeFa directions have a higher MOS score than the top SeFa direction from the table.

6.3. Re-scoring analysis based on CLIP

We performed the re-scoring analysis based on CLIP as we described in 5.3. Since not all human interpretable attributes can be well described in human language in the form we described in 5.3 (for example, the pose attributes of the person cannot be described since words "left" and "right" are not well-defined for generated picture), we selected some attributes that both found by SeFa and ICA-SeFa and performed the re-scoring analysis based on CLIP.

The results are shown in Figure 3. We can see that in most cases, one direction found by ICA-SeFa is strongly correlated to only one semantic attribute. Nonetheless, the directions found SeFa are likely correlated to multiple semantic attributes. This shows the semantic directions found by ICA-SeFa have more precise and disentangled meanings.

7. Conclusion

In this work, we examined the underlying mechanism of SeFa [36] and explained the great power of SeFa by proposing a more comprehensive theory. Based on this theory, we then proposed a new algorithm to do latent semantics factorization on pre-trained with GANs called ICA-SeFa. Experiments have shown that our algorithm can find better interpretable semantic directions than the directions discovered by SeFa [36].

Author Contributions

Houming Chen proposed the idea, developed the theory, invented the algorithm, and implemented the algorithm. Le Qin recruited anotators, designed and conducted experiments measuring MOS, analyzed the experiment results, and helped Houming Chen with the implementation of the algorithm. Yutong Bi and Jiaxi Chen did reviews on related researches and helped Houming Chen with the development of the theory. Dongyang Zhao did reviews on related researches and wrote code to analyzed the data produced in the rescoring analysis based on CLIP. All co-authors were involved in writing this report. All co-authors equally contributed to this project.

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2, 3

[4] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019. 6, 7

[5] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018. 2, 6

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016. 2, 6

[7] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. 2

[8] David M Corey, William P Dunlap, and Michael J Burke. Averaging correlations: Expected values and bias in combined pearson rs and fisher's z transformations. *The Journal of general psychology*, 125(3):245–261, 1998. 8

[9] Federico A Galatolo, Mario GCA Cimino, and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. *arXiv preprint arXiv:2102.01645*, 2021. 8

[10] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5744–5753, 2019. 2

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 2

[12] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2020. 2

[13] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2, 3, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[15] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 2, 6

[16] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000. 2, 5, 6

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[18] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2019. 2, 3, 6

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2, 3

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 3, 4, 6, 8

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2

[22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2, 6

[23] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on graphics (TOG)*, 33(4):1–11, 2014. 3

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 2, 6, 7, 8

[26] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 6

[27] Nearcyan. "this anime does not exist," jan. 21, 2021. 1

[28] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 8

[29] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014. 3

[30] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 3, 5

[31] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *arXiv preprint arXiv:2001.10238*, 2020. 2, 3, 6

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 7, 8

[33] Alec Radford, Luke Metz, and Soumith Chintala. Un-

supervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[34] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2

[35] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, 2020. 2, 6, 7

[36] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[37] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 2, 3, 6, 7, 8

[38] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003. 5

[39] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 3

[40] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision*, pages 1–16, 2021. 2, 3, 6

[41] Julian Zaidi, Jonathan Boilard, Ghyslain Gagnon, and Marc-André Carbonneau. Measuring disentanglement: A review of metrics. *arXiv preprint arXiv:2012.09276*, 2020. 2, 6

[42] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 2

[43] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 3

[44] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020. 2

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

Appendix A : MOS results

Directions	MOS	Possible Semantics
92th direction of ICA-SeFa	1.00	Beards
109th direction of ICA-SeFa	1.00	Length of face
114th direction of ICA-SeFa	1.00	Head direction
116th direction of ICA-SeFa	1.00	Shirt color
131th direction of ICA-SeFa	1.00	Openness of mouth
144th direction of ICA-SeFa	1.00	Gender
149th direction of ICA-SeFa	1.00	Hat
167th direction of ICA-SeFa	1.00	Color of lips
180th direction of ICA-SeFa	1.00	Size of eye
185th direction of ICA-SeFa	1.00	Glasses

Table 1: Top 10 directions with highest MOS(Mean Opinion Score) of ICA-SeFa. 200 directions are decomposed and examined in total. For each row, the possible semantic is verbalized by us and annotators.

Directions	MOS	Possible Semantics
44th direction of SeFa	0.88	Light
14th direction of SeFa	0.75	Head direction
19th direction of SeFa	0.75	Bangs direction
25th direction of SeFa	0.75	Glasses
27th direction of SeFa	0.75	Light
36th direction of SeFa	0.75	Size of eye
42th direction of SeFa	0.75	Wrinkle
65th direction of SeFa	0.75	Color of hair
20th direction of SeFa	0.75	Width of face
24th direction of SeFa	0.75	Hairline

Table 2: Top 10 directions with highest MOS(Mean Opinion Score) of SeFa. 200 directions are examined in total. For each row, the possible semantic is verbalized by an annotator.

Appendix B : Comparison of qualitative results of similar semantic directions

Figure 4: Qualitative comparison of the latent semantics discovered by ICA-SeFa (left), and SeFa (right). ICA-SeFa can find better directions that only change one attributes, while multiple attributes are linked in directions decomposed by SeFa.