Unsupervised Discovery of Disentangled Interpretable Directions for Layer-Wise GAN

Haotian Hu[1], Bin Jiang[1,2(✉)], Xinjiao Zhou[1], Xiaofei Huo[1], and Bolin Zhang[1]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, Hunan, China
{huhaotian,jiangbin,zhouxinjiao,hxfhnu,onlyou}@hnu.edu.cn
[2] Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha 410082, Hunan, China

Abstract. Many studies have shown that generative adversarial networks (GANs) can discover semantics at various levels of abstraction, yet GANs do not provide an intuitive way to show how they understand and control semantics. In order to identify interpretable directions in GAN's latent space, both supervised and unsupervised approaches have been proposed. But the supervised methods can only find the directions consistent with the supervised conditions. However, many current unsupervised methods are hampered by varying degrees of semantic property disentanglement. This paper proposes an unsupervised method with a layer-wise design. The model embeds subspace in each generator layer to capture the disentangled interpretable semantics in GAN. And the research also introduces a latent mapping network to map the inputs to an intermediate latent space with rich disentangled semantics. Additionally, the paper applies an Orthogonal Jacobian regularization to the model to impose constraints on the overall input, further enhancing disentanglement. Experiments demonstrate the method's applicability in the human face, anime face, and scene datasets and its efficacy in finding interpretable directions. Compared with existing unsupervised methods in both qualitative and quantitative aspects, this study proposed method achieves excellent improvement in the disentanglement effect.

Keywords: Discovery of interpretable directions · Generative adversarial network · Unsupervised learning · Disentangled semantic

1 Introduction

Powerful image synthesis abilities and the capacity to fit domain-specific semantic information from enormous volumes of data [1–4] are two features of Generative Adversarial Networks (GANs) [5]. However, GANs do not offer a simple explanation of how it understands and utilizes the learned semantics. Until

H. Hu, X. Zhou, X. Huo and B. Zhang—Contributing authors.

T. Li et al. (Eds.): BigData 2022, CCIS 1709, pp. 21–39, 2022.
https://doi.org/10.1007/978-981-19-8331-3_2

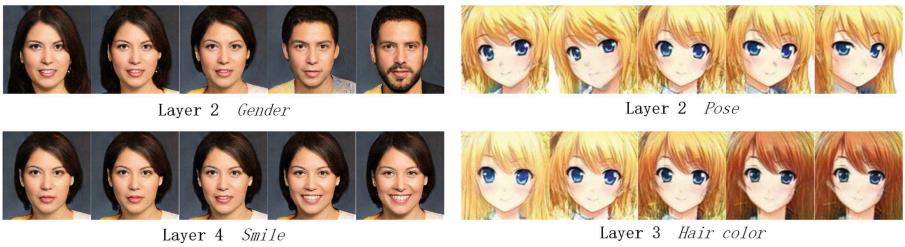

Fig. 1. Examples of interpretable directions we found on the FFHQ and Anime face datasets. For example, the "Age" attribute and "Smile" are found in layer 1 and layer 4 of the face dataset, the "Pose" is found in layer 2 of the Anime face dataset and "hairstyle" in layer 3 of the Anime face dataset.

recently, [6] analyses pre-trained convolutional neural networks and generative adversarial networks by introducing network dissection [7,8]. They find that different layers in the CNN and GAN contained units corresponding to various High-level visual concepts that are not explicitly labeled in the training data. Many findings [8–11] also show that different layers in the network can capture semantic objects with different levels of abstraction. For instance, [12] analyzes the semantic information in the scene dataset using a pre-trained GAN model. They conclude from the experimental data that the model's deep, intermediate and shallow networks correspond to color background information, entity objects, and scene structure. In order to identify semantic properties from the different layers of the generator and to be able to control them to synthesize images, many studies [3,13–24] have focused on mining the semantic information in the latent space of the GAN in recent years.

Some methods [3,13–21] add supervised conditions to the learning of GAN and discover semantic directions in latent space consistent with supervised factors. [3,13–17,19–21] are used to add supervised conditions by assigning labels, manual annotations, or pre-trained classifiers to the generated images, thus finding interpretable semantic directions. For example, the approach [15–17] draw on the contrastive pre-training language-image encoder [18] and textual information as supervised conditions to guide the generation of semantic images. Recent works, such as [15,19–21], employ pre-trained attribute classifiers as supervised conditions to steer the semantic direction in the latent space of the GAN to be consistent with the specific attribute operations. However, supervised methods are limited to finding directions that are interpretable in light of the given supervision criteria; they cannot find a wide variety of semantic directions.

Another research direction that finds interpretable semantics in latent space is to impose unsupervised constraints on the orientation of the latent space. GANSpace [22] discovers important directions in GAN latent space by applying PCA in StyleGAN latent space and BigGAN feature space. SeFa [23] discovers GAN learned latent semantic directions by decomposing model weights, and it focuses on the relationship between image changes and internal representations.

However, these methods require heavy training efforts, and they need to randomly extract a large number of random latent directions and fit them to interpretable semantic directions as much as possible. EigenGAN [24] embeds a linear subspace in each generator layer. The orthogonal basis in subspaces at various levels can capture different semantic directions during model training. However, EigenGAN only imposes a simple regularization constraint on the subspace, and its network structure is relatively simple. For these reasons, although EigenGAN can unsupervised discover many interpretable directions, these directions are poorly disentangled, i.e., there are often multiple attributes entangled together. Compared with supervised methods, unsupervised methods are able to discover more interpretable directions than expected. In fact, poor disentangled visual effects are still a challenge that many unsupervised methods face.

This work aims to develop a network to explore more interpretable semantics in GAN's latent space. So we design a network structure with a linear subspace [24] in each generator layer to discover highly disentangled interpretable directions through an unsupervised approach during GAN learning semantic knowledge (Fig. 1 shows the examples). However, unsupervised approaches are usually worse than supervised ones regarding the degree of disentanglement of the found semantic directions. Inspired by [3], we introduce the mapping network of styleGAN to improve the disentanglement of interpretable directions discovered unsupervised. Many studies [3,25,26] also show that the latent space W of styleGAN is rich in disentangled properties and that the space W can learn the more disentangled semantic information better than the original space Z. Inspired by this, we introduce the latent mapping network in the network structure in order to improve the disentanglement of attributes of interpretable directions discovered in an unsupervised way. In addition, to further improve the disentanglement between the subspace feature dimensions, we introduce Orthogonal Jacobian regularization [27] to disentangle the model by constraining the orthogonal properties between changes caused by the output of each feature dimension. Compared with the method of EigenGAN, we directly impose constraints on the model's inputs to disentangle the learned directions of feature dimensions in each layer. The trial outcomes demonstrate a significant improvement in disentanglement for our strategy.

Overall, we would like to emphasize the following as our primary contributions:

- We suggest an unsupervised method for discovering disentangled interpretable directions in a layer-wise GAN's latent space.
- To overcome the attribute entanglement problem of unsupervised methods, we add a latent mapping network and Orthogonal Jacobian regularization to the model. The latent mapping network transforms the generator's input to an intermediate latent space with rich disentangled semantics. Meanwhile, Orthogonal Jacobian regularization imposes constraints on the overall w-vector of the input generator to improve its orthogonality.
- The experiment results demonstrate our approach's ability to identify distinct and disentangled semantics on various datasets (e.g., human face, anime face,

scene). Compared with existing unsupervised methods in both qualitative and quantitative aspects, ours achieves excellent improvement in the disentanglement effect.

2 Related Works

2.1 Generative Adversarial Networks

GAN's fundamental structural components are a generative network and a discriminative network. The goal of the generative network is to map the noise obtained from random sampling to a high-fidelity image while fooling the discriminative network as much as possible. On the other hand, the discriminant network must decide if the image sent by the generative network is genuine or fake. Therefore, the training is completed by gradually making the generative network capable of generating realistic images as the generative network and the discriminative network confront each other.

Because the training process of GAN is usually complex and unstable, plenty of researches carry out on regularization [1,28,29] and loss function [30,31] in order to improve the ability of GAN to learn semantic knowledge. Our proposed method requires designing subspace structures in each layer of the generative network such that it can sense the changes in sample distribution from random noise fitting to generate realistic images and capture interpretable changes as semantic directions.

2.2 Semantic Discovery for GANs

After GAN models were developed, it was discovered that the latent space of GAN typically contains semantically significant vector operations. Therefore, many studies [8,12–14,22,23,32–37] have been devoted to mining these vectors and using them for image editing.

Supervised Methods. For some methods [8,12–14,32,33] to extract interpretable directions from latent space, manual annotations or outright labels must be added as supervision conditions. InterfaceGAN [13] is a classical supervised approach to semantic face editing by interpreting the latent semantics discovered by GANs. InterfaceGAN allows for exact control of facial characteristics (e.g., gender, age, expression, glasses). However, it also necessitates sampling a sizable amount of labeled data with the aid of an attribute predictor that has already been trained. In StyleFlow [32], labels are used in conjunction with a continuous normalized flow technique to localize the semantic directions in GANs. Additionally, methods [8,12,14,33] use pre-trained semantic predictors to identify interpretable semantics in the latent space. For instance, [12] finds semantic directions in latent space containing scene information by using target detectors to locate entity classes, attributes, and structural information. Even though supervised algorithms can discover interpretable directions of higher quality from

latent space, they frequently necessitate the insertion of pricey external supervision conditions. Additionally, it can only find expected semantic features; it cannot find additional, unanticipated directions.

Unsupervised Methods. GANspace [22] performs Principal Component Analysis (PCA) on the latent space of pre-trained StyleGAN and BigGAN models without any constraint to find major interpretable directions. SeFa [23] discovers the latent semantic directions that GAN has learned by decomposing the model weights and examining the connection between image changes and internal representations. It does not depend on any training or labeling. In recent years, three approaches [34–36] capture key interpretable directions from a pre-trained GAN model. They all involved training a reconstructor and a direction matrix. Specifically, the reconstructor anticipates the changes received from the direction matrix to predict interpretable directions and displacements, whereas the direction matrix is utilized to detect semantic changes in latent space. They all use an unsupervised method to extract interpretable directions from the latent space of the GAN, but this also depends on how well the pre-trained GAN performed. Similarly, latentCLR [37] also uses training a direction matrix to discover interpretable directions from the pre-trained GAN model. However, the difference is that it employs a self-supervised contrast learning loss function to optimize training. These methods are post-processing techniques, meaning that they must get a pre-trained generative network. Only that can they find semantics, even though they do not necessitate the additional insertion of supervision conditions. As a result, their effectiveness heavily depends on how well the pre-trained GAN performs, and they also need to be operated in two steps. Instead of depending on a pre-trained GAN model, our approach only needs one operation step-capturing interpretable directions from changes in the samples during the GAN training.

2.3 Disentanglement Learning with Orthogonal Regularization

Many studies [38–40] have started with regularization to achieve disentanglement, aiming to enhance disentanglement, including regularization in the network training procedure. InfoGAN [38] enables variables to have interpretable information by constraining the relationship between latent code and generated results. Peebles et al. [39] proposes to include the regularization term Hessian Penalty in the generative model, encouraging a generative model's Hessian with regard to its input to be diagonal. So it can be used to find interpretable directions in BigGAN's latent space in an unsupervised fashion. In order to encourage the learnt representations to be orthogonal, D Wang et al. [41] implement orthogonal regularization on the weighting matrices of the model in the manner of $\left\|W^T W - I\right\|_2$, where W is the weight matrix and I is an identity matrix. Furthermore, Bansal et al. [42] introduced another form of regularization by considering both $\left\|W^T W - I\right\|_2$ and $\left\|W W^T - I\right\|_2$. However, the authors also note that this format does not always perform better than $\left\|W^T W - I\right\|_2$ and even performs worse on specific tasks, as evidenced by experimental findings.

EigenGAN [24] also uses regularization in the form of $\left\|W^T W - I\right\|_2$ in the subspace. According to the experimental findings, improving the disentanglement in the discovered interpretability direction is limited by adding the regularization of [41] alone. Thus, we introduce Orthogonal Jacobian regularization (OroJaR) [27], which limits the orthogonal characteristics of each input dimension in the model to disentangle the model by constraining the orthogonal properties of each dimension of the input between the changes induced by the output. In contrast to earlier approaches, the OroJaR can enable the generative model to learn disentangled variants more effectively.

3 Methods

3.1 Overview

Moving in latent space along a specific interpretable direction can get the visual effect after changing the corresponding semantic properties. Our goal is to discover some interpretable directions in GAN learning specific domain knowledge. Figure 2 depicts the overall framework of our model. Firstly, the latent code $z \in Z = \mathcal{N}\left(\mu, \sigma^2\right)$ is obtained by random sampling, where $z = [z_1, z_2, \ldots, z_l]$ and l is the number of generator layers. Following that, mapping network transformation is used to obtain the latent vector $\mathrm{w_i} = \mathrm{f(z_i)}$, where $\mathrm{w} = [w_1, w_2, \ldots, w_l], w_i \in \mathcal{W} \subseteq \mathbb{R}^l$. Here, $\mathrm{f}(\cdot)$ represents the mapping network

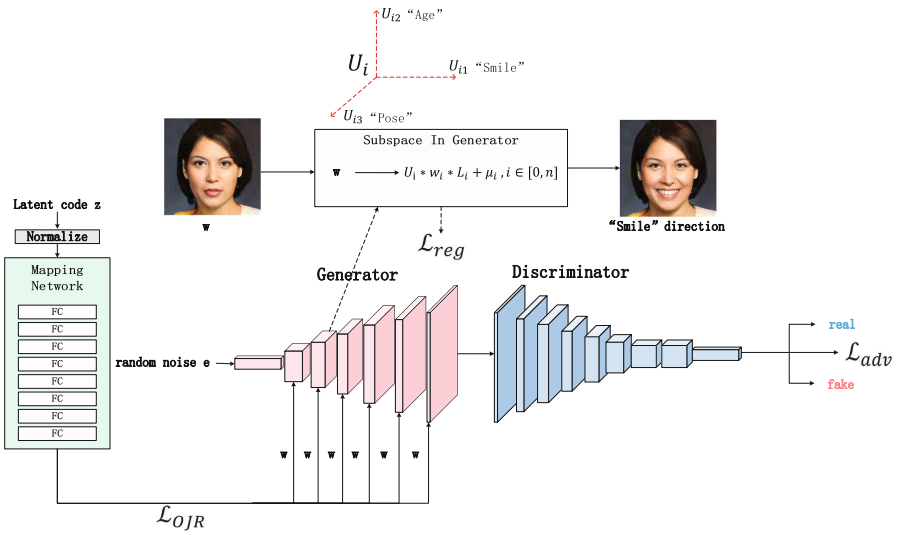

Fig. 2. The architecture of proposed method, which composed of a latent mapping network, one generator and discriminator. The randomly sampled latent code z is converted into the vector w by the latent mapping network and fed into the generator along with the randomly sampled noise. The subspace model of the generator learns interpretable directions from the sample variations during training.

implemented using the multilayer perceptron, and each z_i shares the same f(·). As a result, we train just one latent mapping network. The generator G(·) receives the w vector as input to produce the fake image x_{fake}, which is then supplied to the discriminator D(·) with the genuine image x_{real} for judgment. Eventually, the generator subspace learns interpretable directions from the changes in the sample distribution during training.

3.2 Layer-Wise Semantic Discovering Model

Numerous research [25,26] have shown that the space W of styleGAN is rich in disentangled semantics. Moreover, the intermediate latent space can learn more semantically disentangled information than the original latent space. Therefore, inspired by [3], we convert the model input to the intermediate latent space. First, the latent code $z = [z_1, z_2, \ldots, z_l]$ is randomly sampled for each layer of the generator's subspace where the subscript ℓ denotes the number of generator layers. Furthermore, the z is normalized and input to the latent mapping network. Additionally, the latent mapping network outputs a vector $\mathrm{w} = [w_1, w_2, \ldots, w_l]$ of the space W without altering the latent code's size. The w-vector is then individually input to each stage of the generator. This design choice is beneficial for the final disentangled interpretable directions. Moreover, our generative adversarial model, which draws inspiration from [3,12,24,43], adopts the layer-wise design concept. And the StyleGAN [3] and BigGAN [43]) also introduce it to improve the training stability and synthesis quality. The layer-wise GANs input constants at the first layer and latent code at each subsequent layer, in contrast to traditional GANs that only input latent code at the first layer. Like [24], we feed w into each generator layer and random noise into the first layer. Using the FFHQ dataset as an example, the generator takes the random noise $\epsilon \subseteq \mathbb{R}^{512}$ in the first layer and w_i in each layer as input and the synthetic image x_{fake} as output. The discriminator facilitates the adversarial training by judging $x_{fake} = G([w_1, w_2, \ldots, w_l])$ and x_{real}, which eventually enables the generator to synthesize high-fidelity face images. The generator's subspace model learns the interpretable direction in the face images unsupervised from changes in the sample distribution during adversarial training. Notably, in contrast to the original latent code z, which obeys a Gaussian distribution, [25] states that the distribution of the intermediate latent code w cannot be explicitly modeled. Therefore, since the latent mapping network converts z into w, the changes learned by the subspace model will be more disentangled thanks to the distribution of w.

Similar to [24], we set up a subspace structure $\mathrm{M}_i = [U_i, L_i, \mu_i], i \in [0, l]$ in each generative network layer to capture interpretable directions, where

- U is the orthogonal basis of the subspace, which aims to discover interpretable directions in latent space. $U_i = [U_{i1}, U_{i2}, \ldots, U_{id}]$, where d denotes the dimension of the subspace. Besides, it also represents the number of semantic directions discovered. Each basis vector $U_{ij} \in \mathbb{R}^{H_i \times W_i \times C_i}$ is used to discover one interpretable direction.

- L is the importance matrix $L_i = \text{diag}\,(l_{i1}, l_{i2}, \ldots, l_{id})$, where the absolute value of l_{ij} indicates the importance of U_{ij} for the semantic change at level i^{th}.
- μ denotes the starting point of subspace operations.

The above three parameter values change with the training of the model, and when the training of the model is completed, we need to edit the interpretability direction found by U_{ij}. Firstly, the latent code $\mathrm{z} = [z_1, z_2, \ldots, z_l]$ is randomly sampled and converted to vector $w = f(z)$. Then it is inputted into the subspace of each layer of the generator, activating U_i in the particular i^{th} layer and calculating to get the subspace coordinate points, as shown in Eq. (1).

$$\omega_i = U_i * w_i * L_i + \mu_i = \sum_{j}^{d} U_{ij} * w_{ij} * l_{ij} + \mu_i \tag{1}$$

Then ω_i is added to the network features in the i^{th} layer for calculation, which determines the semantic changes in the i^{th} layer of the generator.

3.3 Orthogonal Jacobian Regularization

In terms of disentanglement, supervised methods are typically more effective than unsupervised methods for finding interpretable directions in GAN. It is so that methods with additional supervised conditions can find directions that are more different from other directions to achieve disentanglement since they are better at identifying the target interpretable directions. Some methods increase the regularization of the training in order to achieve disentanglement [38,39]. Additionally, EigenGAN [24] incorporates the Hessian penalty [39] in training. However, this is insufficient for discovering interpretable semantics in GAN since many semantics-induced changes are usually spatially dependent (e.g., Pose, Hairstyle, etc.). It is not sufficient to constrain each element alone. Instead, [27] proposed that constraining the changes induced by each latent factor in a holistic manner can achieve better disentanglement.

Inspired by this, we also introduce Orthogonal Jacobian regularization [27] in the model to achieve disentanglement. Let $h_i = G_i\,(w_i)$ in the i^{th} layer of the generator, where h_i is the network feature and G_i denotes the output of the i^{th} layer. For the $w_i = [w_{i1}, w_{i2}, \ldots, w_{id}]$ at layer i^{th}, to make w_{ij} and w_{iq}, $j \neq \mathrm{q}, \mathrm{j}, \mathrm{q} \in [0,\ \mathrm{d}]$ the changes induced in the output of the i^{th} layer independent, it is necessary to make the Jacobian vectors in each dimension of the input orthogonal to each other, as shown in Eq. (2).

$$\left[\frac{\partial G_i}{\partial w_{ij}}\right]^T \frac{\partial G_i}{\partial w_{iq}} = 0 \tag{2}$$

The orthogonality of the Jacobian vectors of w_{ij} and w_{iq} implies that they are also uncorrelated. Similar to [27], we consider using Orthogonal Jacobian regularized loss functions for all input dimensions to help the model learn to disentangle interpretable directions. As shown in Eq. (3).

$$L_{OJR} = \sum_{i=1}^{l} \sum_{j=1}^{d} \sum_{j \neq q} \left| \left[\frac{\partial G_i}{\partial w_{ij}} \right]^T \frac{\partial G_i}{\partial w_{iq}} \right|^2 \tag{3}$$

where l represents the generator's overall number of layers.

Optimization Objective. In addition, to further improve the model's effectiveness in finding the disentangled interpretable directions, we add the Hessian penalty [39] to the model for constraining the vectors w of the input generators. Also, inspired by [41], an orthogonal loss function for U is introduced in order to achieve the disentanglement between the interpretable directions found by U, as shown in Eq. (4),

$$\mathcal{L}_{re.g.} = \left\| U_i^T U_i - I \right\|^2 \tag{4}$$

Therefore, the loss functions of the generator and discriminator are shown in Eq. (5) and Eq. (6).

$$\mathcal{L}_G = \mathcal{L}_{adv\_G} + \mathcal{L}_{\text{hes}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{OJR} \tag{5}$$

$$\mathcal{L}_D = \mathcal{L}_{adv\_D} \tag{6}$$

where $\mathcal{L}_{adv\_G}$ and $\mathcal{L}_{adv\_D}$ are consistent with the adversarial objective function of GAN [5], and $\mathcal{L}_{hes}$ denotes the Hessian penalty [39].

4 Experiments

4.1 Experiment Settings

Datasets. To assess the efficacy of our approach, we used the FFHQ [3], Danbooru2019 Portraits [44], and LSUN-Church datasets [45]. They include the Danbooru2019 Portraits dataset, which has 30,2652 anime face photos, and the FFHQ dataset, which has 70,000 HD resolution face photographs. We aim to demonstrate the interpretable directions discovered from the dataset and evaluate their disentanglement. We also apply the method to LSUN-Church and present the interpretable directions we find in scene photographs and animal face images further to illustrate the resilience and efficacy of our proposed method.

Implementation Details. We perform all experiments using the Pytorch toolbox on a single NVIDIA GeForce RTX 1080Ti 11 GB. We reduce the image's resolution to 256 * 256 and increase the batch size to 8 due to the limitation of video memory size. And we select the Adam algorithm as the optimizer and set the initial learning rate to 1e-4.

Fig. 3. Interpretable directions found in different layers of the FFHQ dataset [3]. The intensity of the attribute editing locates at $\in [-4\sigma, 4\sigma]$. And each dimension of the orthogonal basis corresponds to a specific semantic direction. We only show a few of the most meaningful attributes in the figure.

4.2 Non-trivial Visual Effects

First, we show the interpretable directions learned by our method for each layer of the subspace model during the GAN training. Figure 3 displays some illustrations of interpretable directions that the model discovered while learning about faces in the FFHQ dataset, where "Li Dj" denotes the j^{th} dimension of the i^{th} layer of the generator network. In addition, a larger value of i indicates a shallower network layer. By setting $x_{\text{shift}} \in [-4\sigma, 4\sigma]$ and replacing the position of the dimension corresponding to that layer in the latent code z with the value of x_{shift}, we can initially activate a particular dimension of the subspace in a specific layer. Then the semantic editing result image is obtained by traversing the coordinate value in $[-4\sigma, 4\sigma]$ in a specific interpretable direction. As can be observed, progressing in the interpretable direction causes the image's general semantics to progressively shift in that way in order to provide an editing effect that is aesthetically acceptable to humans.

As shown in Fig. 3, the shallow subspaces of the model (layer5, layer6) tend to learn lower-level semantic attributes, such as L5D5 learning to "sunlight", the "skin tone" in L5D6, and the "hue" in L6D6. It is obvious that the shallow subspaces are more eager to discover color-related interpretable directions. The intermediate layer subspaces of the model are skewed to discover regional

Fig. 4. Examples of interpretable directions found on the Anime face dataset [44] (left) and the LSUN-Church dataset [45] (right)

structural changes as the number of layers deepens. For instance, L3 learns the "hair color" attribute while L4 finds changes in position. We discover the deep subspace's propensity to discover high-level interpretable directions related to the abstract features of the deep network layer of the generator learning face knowledge. As seen in the figure, the deep subspace discovers the face's high-level semantic features, e.g., L0 and L1 found attributes such as "glasses", "gender", "hairstyle", and "age". However, the deeper subspaces do not have the same degree of disentanglement as the shallow ones. For instance, the attribute "beard" also appears in the figure when the attribute "gender" advances in the direction of "male". It is probable that the semantic qualities of "male" that the generator learned always include the attribute "beard". As a result, both attributes frequently show up in specific interpretable directions.

To sum up, shallow subspaces in generators often learn low-level features. In contrast, deeper subspaces can find more intricate and high-level interpretable directions, which is in line with the conclusion of Yang Ceyuan [12]. In addition, the conclusion reached by Bau David [6] in GAN models exploring hierarchical semantics, that various layers in a GAN model can find different levels of semantics, is validated by our method since we also use layer-wise ideas in the building of GAN models. However, there is still some entanglement in the interpretability directions of the deeper subspaces because the deeper layers of generators typically learn more abstract features.

On the other hand, we also apply our method to the anime face dataset and scene dataset LSUN-Church in order to confirm the method's applicability. As seen in Fig. 4, we can also find FFHQ-like interpretable directions in the anime face dataset, e.g., "Gender", "Hairstyle", "Bangs", etc. Similar to the findings of FFHQ, anime discovers high-level attributes like "Hair color" in the deep subspace and low-level interpretable directions like "Gender" in the shallow subspace. In addition, the Church dataset contains interpretable directions, e.g., "Vegetation", "Clouds", "Night", "Sky", and "Building color".

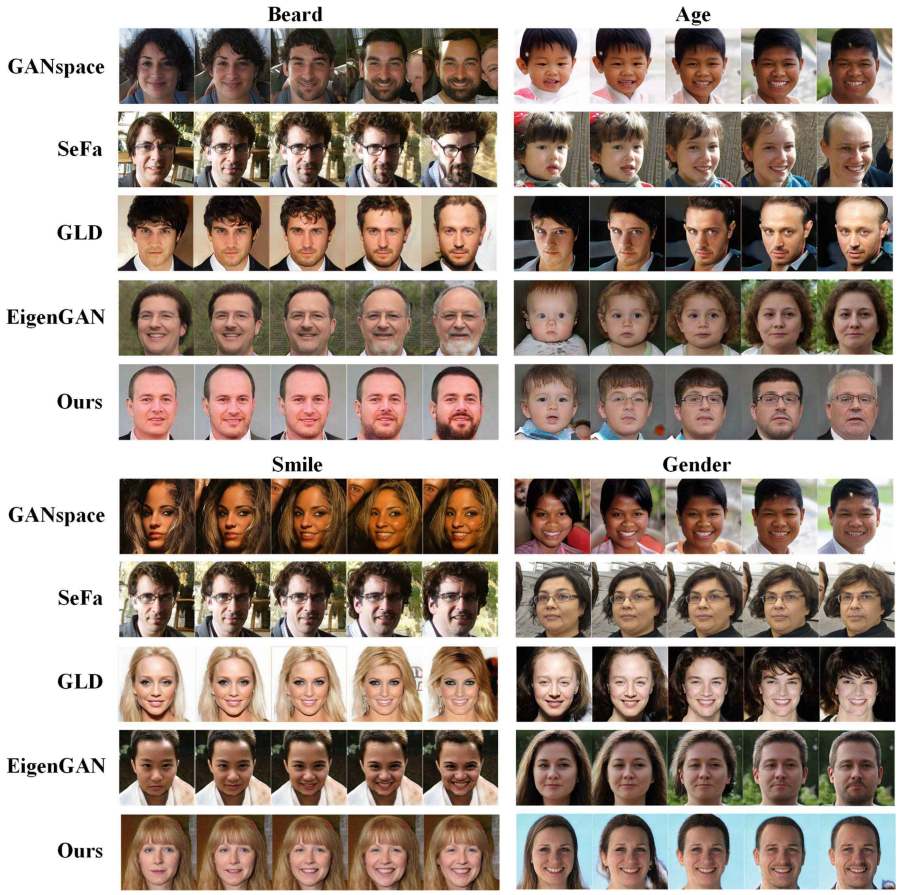

Fig. 5. Qualitative comparison among GANspace [22], SeFa [23], GLD [35], EigenGAN [24] and ours in four common interpretable directions

4.3 Comparison

In this section, to further demonstrate the effectiveness of our approach, we have chosen several classical unsupervised methods for qualitative and quantitative comparisons. Regarding the comparison method, we choose a few traditional unsupervised approaches, including GANspace [22], SeFa [23], GLD [35], and EigenGAN [24] for comparison.

4.3.1 Qualitative Analysis

We compare the interpretable directions found by different methods on the FFHQ face dataset. We download and immediately call the official models offered by GANspace, SeFa, GLD, and EigenGAN to prevent the effects of different machine performances on the model. Then, we call several methods to move

the same step along that direction for the found interpretable direction to get the semantically edited results. Figure 5 lists a number of common semantic attributes on faces found in common by several methods. All methods achieve visual effects that are consistent with the target attributes. Compared with the other methods, EigenGAN and ours obtain smoother changes in visual effects in interpretable directions of movement. Since all methods use an unsupervised approach, the attribute disentanglement is not good enough. In terms of attribute disentanglement, however, our method is still superior to the others, e.g., the change of "Beard" is not entangled with "Gender" or "hair".

4.3.2 Quantitative Analysis

FID Analysis. For a quantitative comparison, we perform the following experiment. To evaluate the effect of the edited visuals obtained by moving along a specific interpretable direction on the image quality, we estimate the Fr'echet Inception Distance (FID) [46] for the images before and after editing. We chose an interpretable direction corresponding to the visual effect of the "smile" attribute for the calculation since all five methods can find it on FFHQ. Figure 6 illustrates the computation of the FID values between the original and edited images following several methods to change the semantics of the images along the semantic direction of the "smile" in different steps. The folded data in the figure demonstrates that our method achieves lower FID values at several edit intensities. Therefore, the edited image obtained by changing along the specific semantics direction found by our method is convincing in terms of visual effect.

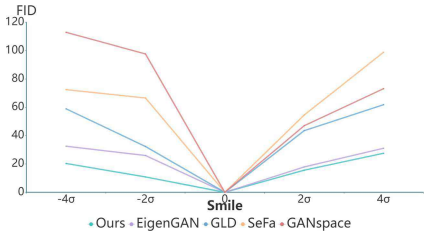

Fig. 6. FID plots for the semantic direction of "smile" produced by different methods. And a smaller FID value indicates that the editing has less impact on the image quality.

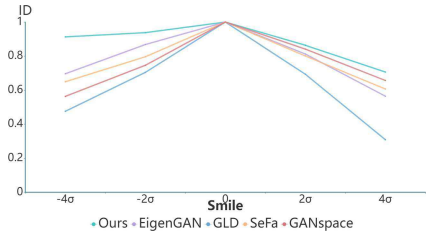

Fig. 7. ID plots for the semantic direction of "smile" produced by different methods. The closer the ID value is to 1, the better the identity information is preserved.

ID Preservation Analysis. On the other hand, we use ArcFace [47] to extract embedding vectors from edited images and compare identity preservation to the other four methods to assess our method's effect on the identity information of the original face after changing the semantics along the interpretable direction. This is a rather fair assessment of how well each method preserves the original facial identity information features while changing the semantics of the image.

Similar to the evaluation scheme of FID, we still chose the semantic direction of"smile". We calculate the identity similarity before and after changing the semantics of the face images in different steps along the "smile" direction (the value closer to 1 means the identity information is better preserved). Figure 7 illustrates that the ID value calculated by our method is closer to 1 than the other methods, thus proving the validity of preserving identity information.

Re-scoring Analysis. The disentanglement of the semantic attribute is also essential in the task of interpretable direction discovery. Therefore, the evaluation of several methods for semantic attribute disentanglement is also an indispensable part. We perform rescoring analysis with the help of a well-trained attribute predictor [23], which can recognize 40 facial attributes on the celebA [8] dataset. Specifically, an interpretable direction is first selected for shifting to change the image semantics. The edited image is then scored using the attribute predictor. Attribute disentanglement can be assessed numerically based on the changes in the scores of other semantic directions throughout the shift to a particular direction, in addition to qualitatively analyzing whether the identified direction accurately represents the relevant semantic attribute. Figure 8 shows the test results of EigenGAN (Fig. 8a) and our method (Fig. 8b). From the results in the figure, it is not difficult to conclude the observations: (i) Like EigenGAN, the interpretable directions discovered by our method do control the change of specific semantics. (ii) Some attributes are more easily entangled, such as "gender" and "beard", which are associated with the performance of both methods, probably since "beard" is often the same as "male" when GAN learns facial features. (iii) Our method performs better than EigenGAN in terms of attribute disentanglement, e.g., EigenGAN entangles "gender" and "age" together.

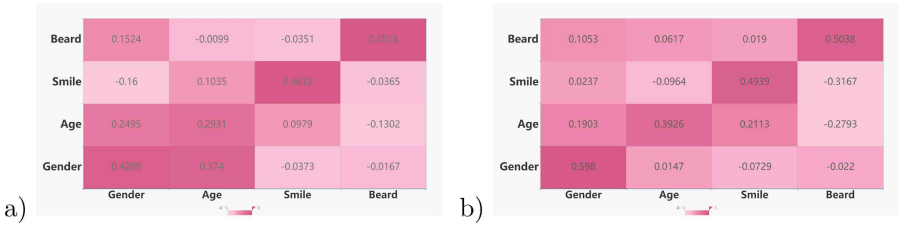

Fig. 8. The results of re-scoring analysis after training EigenGAN [24] (a) and our method(b) on the FFHQ dataset [3]. The data in the table shows how the scores of other semantic features change after moving in a specific semantic direction.

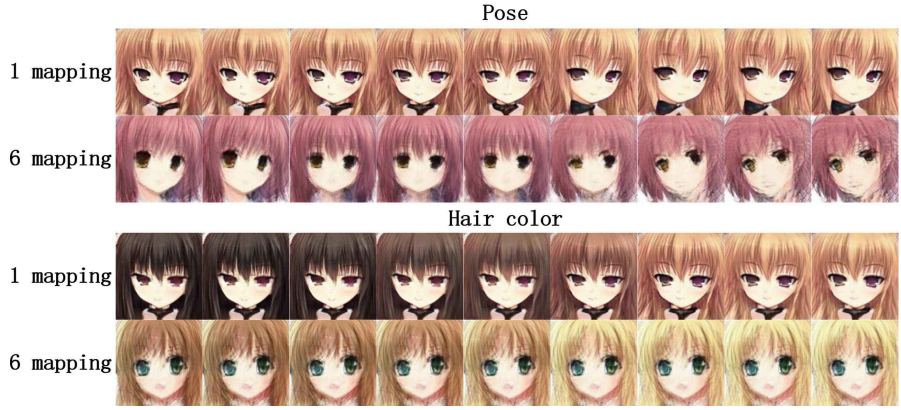

Fig. 9. The visual effects of two strategies for editing the "Pose" and "Hair color" attributes with training one mapping network and six mapping networks.

4.4 Ablation Studies

In our method design, the latent code z of each layer is obtained by random sampling and then input to the mapping network to transform to the intermediate latent space to get the w vector. In our architecture, only one mapping network is trained, and the latent code z of each layer in GAN is transformed through this mapping network. However, since each subspace in the GAN learns different levels of feature for each layer, we propose the following conjecture. Suppose a separate mapping network is trained for the subspace of each layer. Is there an improvement in the editing effect for the interpretable directions learned in the subspace of each layer? In this section, we design an ablation experiment to verify whether the above conjecture is feasible.

First, we compare two strategies: training six mapping networks (our method is designed with six layers of subspaces) versus training just one mapping network. Finally, we compare the editing effects in the interpretability direction for both strategies discovered in the same layer and dimension.

Figure 9 shows the visual effect of editing the attribute "pose" found in L4D1 and the attribute "hair color" found in L5D4 for the above two strategies. Although both methods identify the same interpretable direction in the same dimension at the same layer and the disentanglement and semantic effects are likewise positive, it is clear that the visual effect produced by the training procedures for six mapping networks contains glaring artifacts.

It is well known that the adversarial training process of generative adversarial networks is prone to instability for various reasons compared to other models. Inspired by [48,49], the following two conjectures were obtained regarding the reasons for the causes of artifacts when training six mapping networks. (i) Networks with more layers are typically stronger than networks with fewer tiers in the neural network. In contrast, our model has a simpler structure and fewer network layers (the generator has only six layers). Suppose the additional mapping

network is trained for the subspace model of each layer. In that case, the difference in the distribution of w-vectors obtained between each mapping network will become increasingly large. Moreover, J Wulff et al. [50] also conclude from their analysis that the original noise space obeys a Gaussian distribution, but the intermediate latent space W cannot be clearly modeled. The above strategy may bring complex negative information into the model and destabilize the model's adversarial training, leading to artifacts in the output images when combined with the layer-wise design idea of our method. (ii) We add Jacobian regularization to the model to impose constraints on the input of the entire generator. When combined with the analysis in the first point, the introduction of data into the model with large distributional differences may result in competition between the Jacobian regularization loss and the adversarial training loss, which in turn undermines the stability of the model training.

5 Conclusion

This paper suggests a method for finding disentangled interpretable directions in GAN's latent space in an unsupervised form. We adopt the layer-wise idea to construct the GAN and add the subspace model to the generator to capture the interpretable directions. Since space W is rich in disentangled semantics, we also introduce a latent mapping network to convert the model input to the intermediate latent vector w. In addition, to further achieve well disentanglement, we add the Orthogonal Jacobian regularization to the model to impose constraints on the overall model input. According to the experimental results, compared with existing methods, ours achieves excellent improvement in the disentanglement effect, both in terms of qualitative analysis of the editing effect in the interpretable direction and quantitative analysis of the degree of disentanglement.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under grant 62072169, and Natural Science Foundation of Hunan Province under grant 2021JJ30138.

References

1. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
2. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
3. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
4. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of styleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
5. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)

6. Bau, D., Zhu, J.-Y., Strobelt, H., Lapedriza, A., Zhou, B., Torralba, A.: Understanding the role of individual units in a deep neural network. Proc. Natl. Acad. Sci. **117**(48), 30071–30078 (2020)
7. Zhou, B., Bau, D., Oliva, A., Torralba, A.: Interpreting deep visual representations via network dissection. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2131–2145 (2018)
8. Bau, D., et al.: GAN dissection: visualizing and understanding generative adversarial networks. arXiv preprint arXiv:1811.10597 (2018)
9. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
10. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene CNNs. arXiv preprint arXiv:1412.6856 (2014)
11. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6541–6549 (2017)
12. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. Int. J. Comput. Vision **129**(5), 1451–1466 (2021). https://doi.org/10.1007/s11263-020-01429-5
13. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of GANs for semantic face editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9243–9252 (2020)
14. Goetschalckx, L., Andonian, A., Oliva, A., Isola, P.: GANalyze: toward visual definitions of cognitive image properties. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5744–5753 (2019)
15. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-edit: fine-grained facial editing via dialog. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13799–13808 (2021)
16. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: text-driven manipulation of styleGAN imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2085–2094 (2021)
17. Couairon, G., Grechka, A., Verbeek, J., Schwenk, H., Cord, M.: FlexIT: towards flexible semantic image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18270–18279 (2022)
18. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
19. Yao, X., Newson, A., Gousseau, Y., Hellier, P.: A latent transformer for disentangled face editing in images and videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13789–13798 (2021)
20. Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G.: GAN-control: explicitly controllable GANs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14083–14093 (2021)
21. Lang, O., et al.: Explaining in style: training a GAN to explain a classifier in StyleSpace. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 693–702 (2021)
22. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: GANspace: discovering interpretable GAN controls. Adv. Neural. Inf. Process. Syst. **33**, 9841–9850 (2020)

23. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1532–1540 (2021)
24. He, Z., Kan, M., Shan, S.: EigenGAN: layer-wise eigen-learning for GANs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14408–14417 (2021)
25. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for styleGAN image manipulation. ACM Trans. Graph. (TOG) **40**(4), 1–14 (2021)
26. Wu, Z., Lischinski, D., Shechtman, E.: StyleSpace analysis: disentangled controls for styleGAN image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12863–12872 (2021)
27. Wei, Y., et al: Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6721–6730 (2021)
28. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
29. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for GANs do actually converge? In: International Conference on Machine Learning, pp. 3481–3490. PMLR (2018)
30. Nowozin, S., Cseke, B., Tomioka, R.: f-GAN: training generative neural samplers using variational divergence minimization. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
31. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
32. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: StyleFlow: attribute-conditioned exploration of styleGAN-generated images using conditional continuous normalizing flows. ACM Trans. Graph. (TOG) **40**(3), 1–21 (2021)
33. Shen, Y., Yang, C., Tang, X., Zhou, B.: InterFaceGAN: interpreting the disentangled face representation learned by GANs. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 2004–2018 (2020)
34. Cherepkov, A., Voynov, A., Babenko, A.: Navigating the GAN parameter space for semantic image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3671–3680 (2021)
35. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the GAN latent space. In: International Conference on Machine Learning, pp. 9786–9796. PMLR (2020)
36. Tzelepis, C., Tzimiropoulos, G., Patras, I.: WarpedGANSpace: finding nonlinear RBF paths in GAN latent space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6393–6402 (2021)
37. Yüksel, O.K., Simsar, E., Er, E.G., Yanardag, P.: LatentCLR: a contrastive learning approach for unsupervised discovery of interpretable directions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14263–14272 (2021)
38. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 29 (2016)

39. Peebles, W., Peebles, J., Zhu, J.-Y., Efros, A., Torralba, A.: The hessian penalty: a weak prior for unsupervised disentanglement. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 581–597. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_35
40. Zhu, X., Xu, C., Tao, D.: Learning disentangled representations with latent variation predictability. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 684–700. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_40
41. Wang, D., Cui, P., Ou, M., Zhu, W.: Deep multimodal hashing with orthogonal regularization. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
42. Bansal, N., Chen, X., Wang, Z.: Can we gain more from orthogonality regularizations in training deep networks? In: Advances in Neural Information Processing Systems, vol. 31 (2018)
43. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
44. Branwen, G., Gokaslan, A.: Danbooru 2019: a large-scale crowdsourced and tagged anime illustration dataset (2019)
45. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
46. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
47. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
48. Liang, J., Zeng, H., Zhang, L.: Details or artifacts: a locally discriminative learning approach to realistic image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5657–5666 (2022)
49. Ledig, C., 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, pp. 4681–4690 (2017)
50. Wulff, J., Torralba, A.: Improving inversion and generation diversity in StyleGAN using a gaussianized latent space. arXiv preprint arXiv:2009.06529 (2020)