Face Identity-Aware Disentanglement in StyleGAN

Adrian Suwała[1] Bartosz Wójcik[1,2,3] Magdalena Proszewska[4] Jacek Tabor[1]
Przemysław Spurek[1] Marek Śmieja[1]

[1] Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland
[2] Doctoral School of Exact and Natural Sciences, Jagiellonian University, Poland
[3] IDEAS NCBR, Warsaw, Poland
[4] University of Edinburgh
{marek.smieja}@uj.edu.pl

Abstract

Conditional GANs are frequently used for manipulating the attributes of face images, such as expression, hairstyle, pose, or age. Even though the state-of-the-art models successfully modify the requested attributes, they simultaneously modify other important characteristics of the image, such as a person's identity. In this paper, we focus on solving this problem by introducing PluGeN4Faces, a plugin to StyleGAN, which explicitly disentangles face attributes from a person's identity. Our key idea is to perform training on images retrieved from movie frames, where a given person appears in various poses and with different attributes. By applying a type of contrastive loss, we encourage the model to group images of the same person in similar regions of latent space. Our experiments demonstrate that the modifications of face attributes performed by PluGeN4Faces are significantly less invasive on the remaining characteristics of the image than in the existing state-of-the-art models.

1. Introduction

Modern generative models, such as StyleGAN [14, 15, 16], produce high-quality images, which are frequently indistinguishable from real ones. One of the current challenges is to introduce the functionality for manipulating the attributes of existing images. In the case of face images, we would like to modify the expression, the type of facial hair, or even the gender of the person in the photo.

Although the state-of-the-art conditional generative models, such as PluGeN [32] or StyleFlow [3], are capable of modifying selected face attributes, there is no guarantee that only requested attributes are changed. Experiments show that modifications of intended attributes often affect other attributes as well as the identity of a person. It means that the latent space used for modifications is so entangled that manipulating only selected attributes independently from other characteristics of the image is impossible.

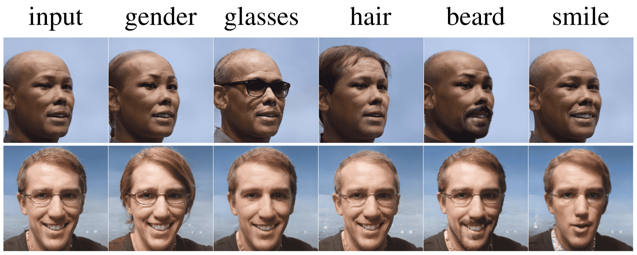

Figure 1. Sample effects of attributes manipulation performed by PluGeN4Faces.

There may be various reasons why existing models cannot create disentangled latent representation. In this paper, we argue that the conditional generative models are usually trained on generated (fake) images and they have never seen images representing the same person with different combinations of attributes. To introduce the information about the person's identity, we need to perform training on real images instead of generated ones only.

Working with real images is straightforward in autoencoder-based generative models, but there appear notable problems in the case of GANs since there is no built-in method for encoding images into the GAN latent space. The problem is especially challenging for StyleGAN architecture because of the structure of its style space. While generated images are identified by a single style code $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^{512}$, not every image can be accurately mapped into $\mathcal{W}$ [1]. To overcome this issue, most techniques (employing the encoder or gradient-based optimization) perform the search in the extended style space $\mathcal{W}_*^k$, where

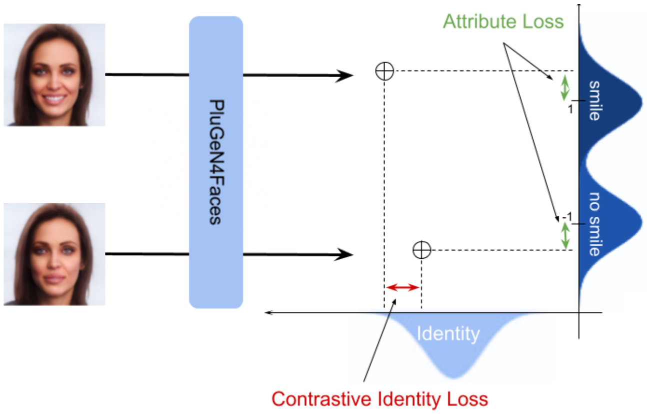

Figure 2. Explicit disentanglement of attribute and identity features performed by PluGeN4Faces. While each labeled attribute is modeled as an individual latent dimension, the contrastive loss allows us to group latent codes representing images of the same person in similar regions of the space.

a style code consists of k different 512-dimensional style vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k \in \mathbb{R}^{512}$ (typically $k = 18$) – one for each layer of the StyleGAN architecture that can receive input via AdaIn [1, 30, 35]. Operating on the whole set of style codes significantly increases the dimensionality of latent codes and theoretically makes the complexity of the problem more challenging.

In this paper, we introduce PluGeN4Faces (**Plu**gin **Ge**nerative **N**etworks for **Faces**), a plugin model for disentangling the latent space of StyleGAN in the case of face images. PluGeN4Faces provides full control on manipulating face attributes so that the modification of the requested attributes has a minimal effect on the identity of a person and the remaining face attributes (including background), see Figure 1 for sample results. PluGeN4Faces works as a plugin to pre-trained StyleGAN, which means that it does not change the weights of StyleGAN but only transforms its style space into a disentangled one. In consequence, a training process is extremely simple and absorbs limited computational resources.

In contrast to competitive models, PluGeN4Faces is trained on face images retrieved from movie frames, which can present a given person in various poses and with different attributes. The information about a person's identity is used in PluGeN4Faces by employing a contrastive loss. Namely, we encourage the model to group images of the same person in similar regions of latent space, see Figure 2. To use real images in training, we implement PluGeN4Faces as a conditional invertible normalizing flow, where the condition represents the identifier of the style code. In other words, PluGeN4Faces transforms every style code $\mathbf{w}_i$, for $i = 1, \ldots, k$, by the flow conditioned on the index i. In this way, we are able to implement a compact disentanglement module operating on real images.

We evaluate PluGeN4Faces on face images retrieved from the FFHQ database as well as movie frames. We show that PluGeN4Faces allows for effective manipulation of face attributes. Moreover, the applied modifications preserve the person's identity to a significantly greater extent than in competitive models. The presented sample results are supported by the quantitative analysis, which confirms the advantage of PluGeN4Faces over related models.

The contribution of the paper is summarized as follows:

- We introduce a plugin to StyleGAN for manipulating the attributes of real images. In contrast to existing models, it is trained on real images encoded into StyleGAN style space using the encoder network.
- We improve the representation disentanglement in conditional generative models by applying a type of contrastive loss, which explicitly encodes the person's identity. In consequence, the manipulation of the requested attributes is less invasive on the remaining image characteristics (including person's identity).
- The proposed solution is evaluated in a strict quantitative way, which allows for a fair comparison with related models. The proposed metrics together with our sample results clearly demonstrate the advantage of PluGeN4Faces over competitive methods.

Our code is available at: `https://github.com/gmum/plugen4faces`. Demo app is available at: `https://gmum.ii.uj.edu.pl/plugen/`.

2. Related work

Conditional VAE (cVAE) is one of the first methods of including additional label information in a generative model [17], which has been successfully applied in a variety of disciplines including image generation [18, 28, 33]. However, the independence of latent codes and labels is not assured, which has a negative impact on the generation quality. Conditional GAN (cGAN) is an alternative that is able to produce examples of significantly better quality [4, 12, 22, 24, 25] , but the training of the model is more difficult [19]. Fader Networks [20] overcome this limitation by combining components of cVAE and cGAN, as they use both encoder-decoder architecture and the discriminator, which predicts the image attributes from a corresponding latent vector obtained from the encoder. As with previous methods, Fader Networks does not preserve the disentanglement of attributes, moreover, the training is even more difficult than that of standard GANs.

While the described approaches focus on creating conditional generative models from scratch, recent work frequently focuses on manipulating the latent codes of pretrained networks. In this scenario, data complexity is not that big of a limitation, hence flow models can be easily applied. StyleFlow [3] and PluGeN [32] operate on the latent space of GAN using a normalizing flow module: conditional CNF [9] and NICE [7], respectively. While StyleFlow is adapted to work only on StyleGAN [16], PluGeN demonstrates great results also with other models and in different domains. For StyleGAN, they are both trained using latent codes sampled from latent space W and attributes of images that correspond to them. Competitive approaches include [8, 10, 23, 29]. InterFaceGAN [26] aims to manipulate various properties of the facial semantics via linear models applied to the latent space of GANs. HijackGAN [31] goes beyond linear models and designs a proxy model to traverse the latent space of GANs.

Along with the latent codes manipulation techniques, methods of embedding examples into the GAN latent space can be used to allow manipulation of existing examples. There are two main embedding approaches: (i) an encoder network that maps an image into the latent space [30], (ii) an optimization algorithm that iteratively improves a latent code so that it produces a desired image [1, 2, 36]. Moreover, combinations of these two approaches exist, in which the encoder outputs an approximate embedding that is then improved by the optimization algorithm [34]. These methods allow us to train our model using real images, which are encoded into the extended StyleGAN latent space $\mathcal{W}_*^k$ that enables manipulation of existing images. As shown in [1], the use of $\mathcal{W}_*^k$ latent space instead of $\mathcal{W}$ reduces the alteration of the original image.

3. Identity-aware disentanglement

Overview PluGeN4Faces is a conditional invertible normalizing flow module (cINF), which is attached to the style space of StyleGAN. It transforms the style codes of pretrained StyleGAN into a disentangled space so that:

- the labeled attributes are modeled by the individual latent coordinates,
- images of the same person are grouped in similar regions of the latent space.

While realizing the first of the above conditions allows us to edit the values of requested attributes, the second one prevents severe changes in the image during attribute manipulation.

In this section, we first review the StyleGAN architecture and recall the way of encoding real images into its style space. Next, we present a probabilistic structure of PluGeN4Faces, and cINF mapping function. We discuss the training procedure and the inference phase.

StyleGAN architecture StyleGAN architecture [15] consists of two main parts: (a) a mapping network that transforms latent codes $\mathbf{z} \in \mathcal{Z}$ sampled from Gaussian noise $\mathcal{N}(\mu, I)$ to the style vectors $\mathbf{w} \in \mathcal{W}$, (b) a synthesis network that creates an image from the style code replicated several times. The replicated style codes represent the inputs to subsequent layers of the synthesis network.

Instead of manipulating latent codes $\mathbf{z} \in \mathcal{Z}$, we usually operate on the style space $\mathcal{W}$ to perform attribute modification, which was shown to be significantly more disentangled [3]. However, it is well-known that not all real images can be encoded into the StyleGAN's style space $\mathcal{W}$ [35]. A typical approach for coping with this issue is to extend the search space and look for k different style codes $(\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathcal{W}_*^k$, which together could synthesize the original input [1, 30]. Each $\mathbf{w}_i$ represents the input to the i-th layer of the synthesis network. Even though a sequence of style codes from the extended style space does not reflect any latent code $\mathbf{z}$, it allows for the convenient reconstruction and manipulation of real images. One can design an encoder [30] or implement a gradient-based procedure for embedding real images into the extended style space. In this paper, we employ an encoder network.

Probabilistic structure of PluGeN4Faces We assume that every image $\mathbf{x}$ is described by the composition of the attribute and non-attribute vectors $(\mathbf{c}, \mathbf{s})$, where $\mathbf{c} \in \mathbf{C} = (C_1, \dots, C_M)$ and $\mathbf{s} \in \mathbf{S} = (S_1, \dots, S_{N-M})$. While each attribute variable $c_i \in C_i$ contains information about the selected attribute, the non-attribute vector $\mathbf{s}$ is used to describe the remaining characteristic of data including background and personal identity in the case of face images. To control the value of every attribute independently of each other, a factorized form of the probability distribution of the random vector $(\mathbf{C}, \mathbf{S})$ is assumed. Given a vector of true labels $\mathbf{y} = (y_1, \dots, y_M)$, the conditional distribution of $(\mathbf{c}, \mathbf{s})$ is defined by

$$p_{\mathbf{C},\mathbf{S}|\mathbf{Y}=\mathbf{y}}(\mathbf{c}, \mathbf{s}) = \prod_{i=1}^{M} p_{C_i|Y_i=y_i}(c_i) \cdot p_{\mathbf{S}}(\mathbf{s}) \text{ , for } (\mathbf{c}, \mathbf{s}) \in \mathbb{R}^N.$$

In the above formula, the i-th label y_i affects only the i-th attribute variable C_i. As a parametric form of $p_{C_i|Y_i=y_i}$, we use a 1-dimensional Gaussian density $\mathcal{N}(y_i, \sigma)$. By changing the condition $Y_i = y_i$, we modify the mean of the Gaussian. The distribution of the non-attribute vector is modeled as a multivariate standard Gaussian density $\mathcal{N}(\mathbf{0}, \mathbf{I}_{N-M})$. The non-attribute vector $\mathbf{s}$ is responsible for covering information about a person's identity, image background, etc., so images presenting the same person should have similar values of $\mathbf{s}$.

Invertible mapping To realize the above parameterization, a two-way mapping between the style space of the pretrained StyleGAN and the disentangled space $(\mathbf{C}, \mathbf{S})$ has to be established. Since we work with real images (not only generated ones), we employ the StyleGAN encoder [30], which produces a sequence of style codes $\{\mathbf{w}_1, \dots, \mathbf{w}_k\} \in$

$\mathcal{W}_*^k$ representing a given image $\mathbf{x}$ in the subsequent layers of the StyleGAN synthesis network. Thus, we need to map a sequence of style codes $(\mathbf{w}_i)_{i=1}^k$ (representing a single image $\mathbf{x}$) into a sequence of the attribute and non-attribute vectors $(\mathbf{c}_i, \mathbf{s}_i)_{i=1}^k$. To find such an invertible transformation, we use a conditional INF (cINF), which is parametrized by the identifier of the style code. More precisely, the cINF, $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^N$, takes the style code $\mathbf{w}_i$ and the index of i-th layer as a condition and returns a disentangled representation of $\mathbf{w}_i$ as $(\mathbf{c}_i, \mathbf{s}_i) = \mathcal{F}(\mathbf{w}_i|\text{layer} = i)$. Here, both $\mathbf{c}_i = (c_1^i, \dots, c_M^i)$ and $\mathbf{s}_i = (s_1^i, \dots, s_{N-M}^i)$ are vectors corresponding to a given $\mathbf{w}_i$.

Training The conditional INF is trained by minimizing the negative log-likelihood taken over all style codes. Given a sequence of style codes $(\mathbf{w}_i)_{i=1}^k$ representing an image $\mathbf{x}$ with labels $\mathbf{y}$, we aim at minimizing:

$$-\sum_{i=1}^{k} \log p_{\mathbf{W}_i|\mathbf{Y}=\mathbf{y}}(\mathbf{w}_i) =$$
$$-\sum_{i=1}^{k} \left(\sum_{j=1}^{M} \log p_{C_i^j|Y_i=y_i}(c_i^j) + \log p_{\mathbf{S}_i}(\mathbf{s}_i) + \right.$$
$$\left. \log \left| \det \frac{\partial \mathcal{F}^{-1}(\mathbf{w}_i|\text{layer} = i)}{\partial \mathbf{w}_i} \right| \right), \quad (1)$$

where $(\mathbf{c}_i, \mathbf{s}_i) = \mathcal{F}^{-1}(\mathbf{w}_i|\text{layer} = i)$ are the attribute and non-attribute vectors describing the i-th style code $\mathbf{w}_i$ (in the i-th StyleGAN layer).

In addition to the negative log-likelihood minimization, which focuses on modeling labeled attributes, we introduce a contrastive loss responsible for the explicit encoding of the face identity. Thanks to the contrastive loss, manipulating the labeled attributes will have a minimal effect on changing other attributes (including identity) of the face image.

To construct our contrastive loss, we take n images $\mathbf{x}_1, \dots, \mathbf{x}_n$ of a given person and encode them into the style space of StyleGAN using the encoder network. Such images can be retrieved from subsequent frames of movies. For each image, the encoder produces a sequence of style codes, which represent the input to subsequent layers of StyleGAN generator. For transparency, we restrict our attention to the l-th layer in the following description. For n images, we have n style codes $\mathbf{w}_1, \dots, \mathbf{w}_n$, in which $\mathbf{w}_i$ is the representation of $\mathbf{x}_i$ in the l-th layer (we drop the index of the layer for simplicity). Making use of conditional INF, we find a disentangled representation of $\mathbf{w}_i$ as

$$(\mathbf{c}_i, \mathbf{s}_i) = \mathcal{F}(\mathbf{w}_i|\text{layer} = l).$$

To force the structure on non-attribute variables, where images of the same person are represented by similar non-attribute vectors, we apply the following contrastive loss:

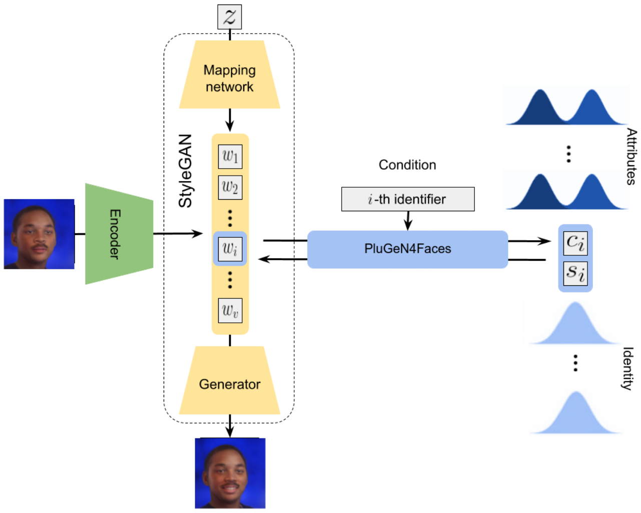

Figure 3. Architecture of PluGeN4Faces. Given the representation of the input image using a sequence of style codes, PluGeN4Faces uses INF to model labeled attributes as individual latent dimensions. The remaining characteristic of the image (including the person's identity) are modeled in separate dimensions using contrastive loss.

$$\sum_{i \neq j} \|\mathbf{s}_i - \mathbf{s}_j\|^2 = 2n \sum_{i=1}^{n} \|\mathbf{s}_i - \mathbf{m}\|^2, \quad (2)$$

where the mean $\mathbf{m} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{s}_i$ is used to reduce the number of comparisons [27]. Minimization of (2) leads to mapping the set of n input images to similar values of non-attributes vectors. We apply this loss to images representing the same person.

To sum up, the complete loss of PluGeN4Faces is given by taking together the introduced contrastive loss (2) and negative log-likelihood (1). For the first loss component, we need a set of images representing the same person, while for the second one, we use images with labeled attributes.

4. Experiments

Experimental setting We consider Flickr-Faces-HQ dataset (FFHQ) containing 70 000 high-quality images of resolution 1024 × 1024. The Microsoft Face API was used to label 8 attributes in each image (gender, glasses, hair/bald, facial hair/beard, expression/smile, age, pitch, and yaw).

Additionally, to explicitly control the person's identity we use images retrieved from video clips. More precisely, we use images from videos and celebrity interviews scraped from YouTube with 573 videos, an average of 19.33 images per video and 12 194 images in total. As in the case of FFHQ dataset, attributes of every image are also labeled using Microsoft Face API.

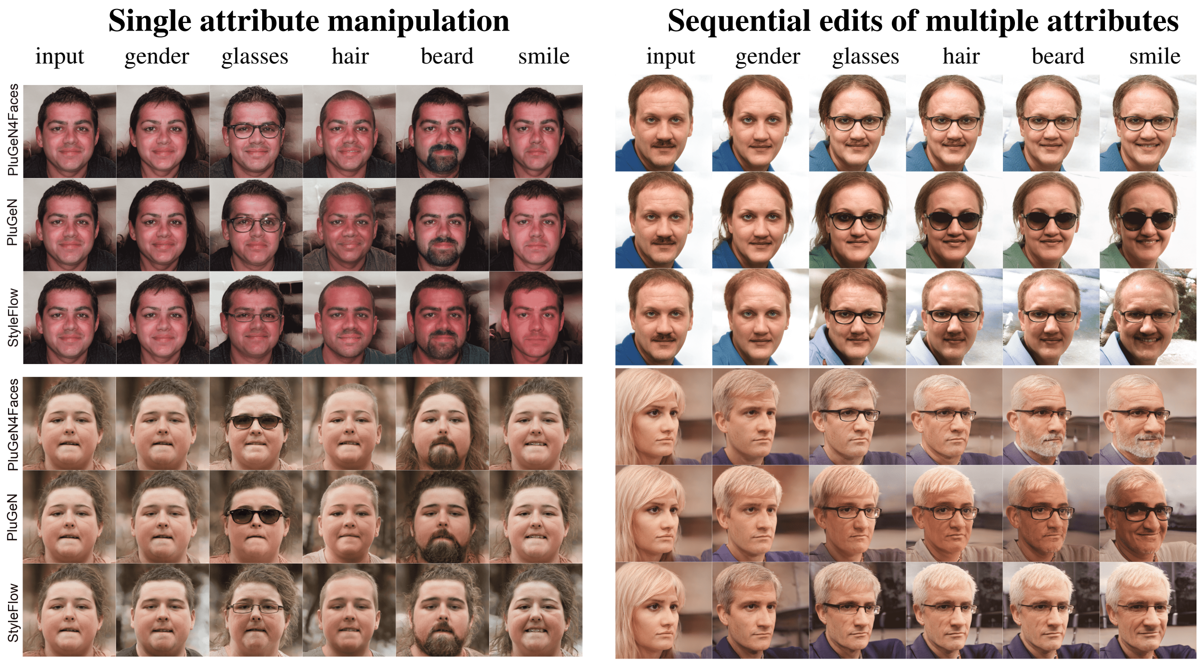

Figure 4. Single attribute manipulations (left) and sequential edits of multiple attributes (right).

To evaluate the proposed disentanglement model access to an independent face attribute classifier is needed. For this purpose, we train the ResNet-18 model [11] on the FFHQ and 10 000 randomly generated StyleGAN face images. The model is trained with 8 outputs in a multi-label manner, treating the Microsoft Face API labels as targets. We standardize the labels as well as apply the shrinkage loss [21] as we find that it helps with dataset imbalance. We use the same loss for binary and continuous labels as this works equally well for classification [13]. Although not all of the face attribute labels are binary, we call this model *classifier* in the remainder of this paper to avoid any confusion with the other models used in the experiments.

We use StyleGAN (version 2) as a backbone model, which was trained on FFQH dataset. Real images are encoded to the extended latent space $\mathcal{W}_*^k$ of StyleGAN, where $k = 18$, using the encoder network [30]. In consequence, every image is represented using a sequence of style codes $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \in \mathcal{W}_*^k$, where $\mathbf{w}_i \in \mathbb{R}^{512}$. The encoder is trained on the combination of images from FFHQ and movie datasets.

PluGeN4Faces is instantiated using conditional RealNVP flow model [6] that operates on the individual latent codes $\mathbf{w}_i \in \mathbb{R}^{512}$ of StyleGAN. The condition is an identifier i of the style code (being the input to the i-th StyleGAN layer) represented as a one-hot vector.

As a baseline, we choose two state-of-the-art conditional models, PluGeN and StyleFlow, which can be used with a pre-trained StyleGAN. PluGeN uses NICE flow model to transform individual style codes $\mathbf{w}_i$ to disentangled space. In other words, PluGeN uses a single shared NICE model (with the same parameters) as a mapping between each style code and the target disentangled space. StyleFlow is parameterized by the conditional continuous flow, where the conditioning factor corresponds to the labeled attributes. Similarly to PluGeN, StyleFlow uses a single flow, which is applied to various style codes.

Qualitative results In this section, we illustrate the sample results produced by the proposed model. First, we perform a single edit of binary attributes. Next, we consider sequential edits, where subsequent modifications on binary attributes are added one by one. In both cases, we perform a minimal modification needed to change a decision of the attribute classifier. More precisely, we perform a gradual change of the attribute and inspect the reaction of the attribute classifier on the modified attribute of the generated image. If the classifier recognizes the attribute of the generated image with sufficient confidence, then we stop modification and return the generated image. By making use of an independent classifier, we are guaranteed to obtain a fair comparison regardless of the scale used by the models.

Figure 4 presents the results of single (left) and sequential edits (right). At first glance, all considered models give visually appealing effects and perform successfully the requested modifications. Observe however that PluGeN and StyleFlow changed the ethnicity and age of the person in the top left example when modifying the attribute ”hair”.

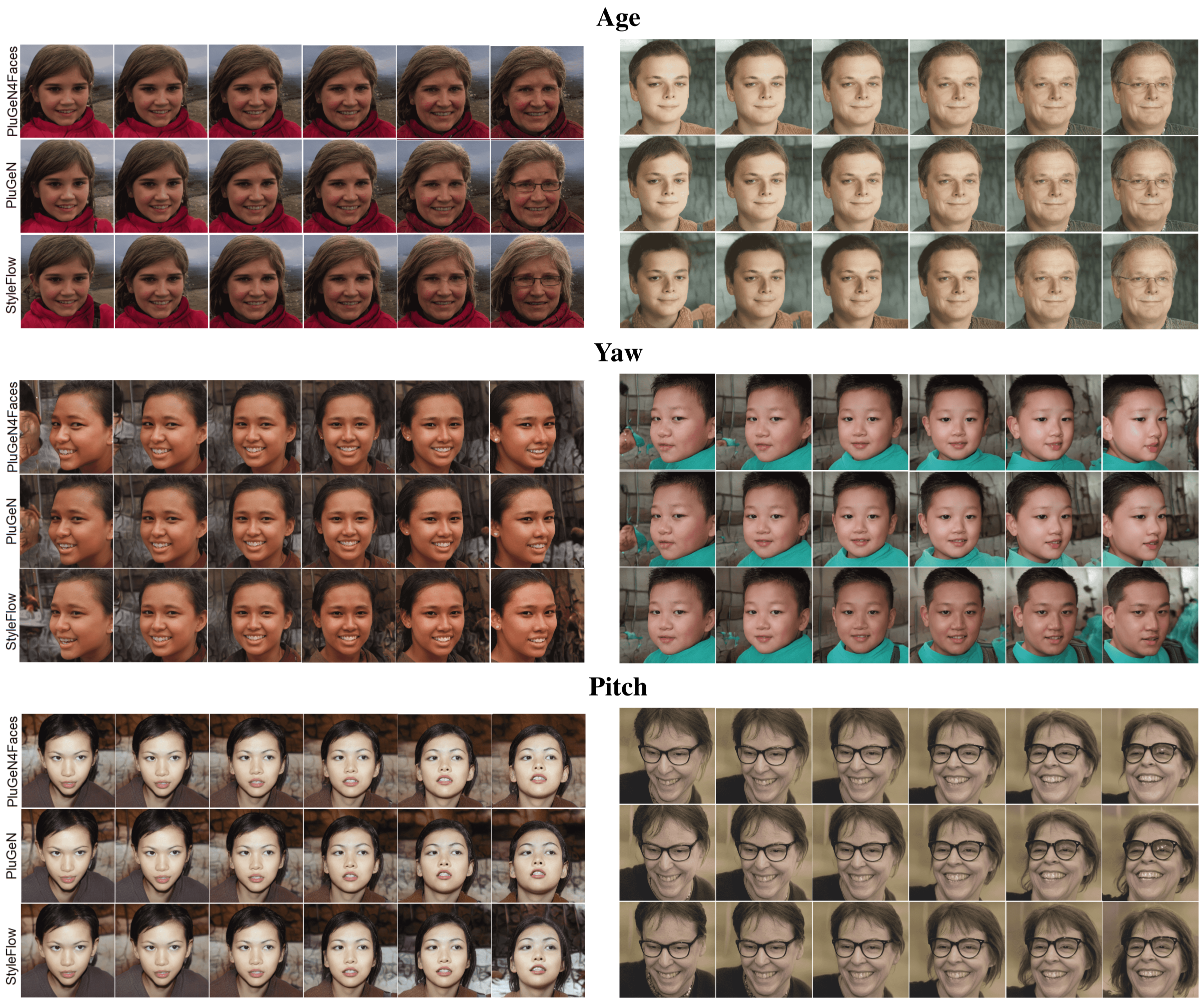

Figure 5. Interpolation on the extreme values of continuous attributes.

Such a behavior is not accepted and does not hold in the case of PluGeN4Faces (see the 1st row of Figure 1, where PluGeN4Faces added hair without changing the ethnicity). It is impressive that all models were able to combine the attribute ”beard” with a woman’s face in the bottom left example. Nevertheless, the face produced by PluGeN4Faces has more female features than the ones generated by PluGeN and StyleFlow. Looking at sequential edits (right), it is evident that PluGeN4Faces kept the color of clothes and background unchanged, which is not the case of PluGeN and StyleFlow. Moreover, the type of glasses is also unaffected by attribute manipulations performed by PluGeN4Faces. On the downside, it should be noted that all models make the face slightly older when the attributes ”bald” or ”beard” are used.

We also illustrate the manipulations of continuous attributes by showing the path between two extreme values of a given attribute, see Figure 5. Although the requested modifications have been successfully realized by the models, PluGeN4Faces was less invasive to the images. PluGeN could not avoid adding glasses when changing the age (left); it modified the gender of the child’s face when turning the head left (middle right); it changed the color of clothes in the bottom left example. StyleFlow modified the age of a child when turning his head right (middle right) as well as added male features to the face presented in the bottom right example when the head was turned down. PluGeN4Faces was free of the aforementioned drawbacks, which demonstrates that it better disentangles the image space and is able to preserve more of the original features during edits.

Identity preservation In this part, we support our sample results with quantitative evaluation, which aims at verifying how well PluGeN4Faces disentangles the image representation. To this end, we change a single attribute of a given image and compare the resulting picture with the original

Table 1. Identity disentanglement. For each image, we change of the values of attributes listed in rows and compare the relation between original input image and the modified one in terms of 4 measures: (i–ii) MSE between image embedings taken from Face Recogiontion and ArcFace models, (iii) PSNR and (iv) SSIM applied on raw images.

	PluGeN4Faces (ours)				**PluGeN**				**StyleFlow**			
	FR MSE ↓	ArcFace MSE ↓	Raw PSNR ↑	Raw SSIM ↑	FR MSE ↓	ArcFace MSE ↓	Raw PSNR ↑	Raw SSIM ↑	FR MSE ↓	ArcFace MSE ↓	Raw PSNR ↑	Raw SSIM ↑
male	0.20	**0.25**	**30.34**	**0.84**	**0.20**	0.26	29.96	0.83	0.25	0.35	26.97	0.75
female	**0.22**	**0.28**	**29.58**	**0.84**	0.24	0.31	26.49	0.80	0.27	0.38	26.91	0.74
glasses	0.40	0.65	20.97	0.65	0.42	0.64	19.62	0.64	**0.36**	**0.53**	**22.20**	**0.66**
no glasses	**0.12**	**0.11**	**39.05**	**0.95**	0.12	0.11	37.37	0.93	0.20	0.21	27.93	0.77
bald	**0.14**	**0.18**	**29.50**	**0.82**	0.22	0.27	24.32	0.74	0.21	0.28	27.45	0.72
hair	**0.07**	**0.04**	38.67	**0.95**	0.10	0.07	33.19	0.90	0.10	0.09	**38.77**	0.88
old	0.45	**0.67**	**22.75**	**0.66**	0.45	0.72	20.65	0.62	**0.45**	0.70	20.63	0.57
young	**0.43**	**0.63**	**22.71**	**0.69**	0.46	0.75	20.61	0.63	0.43	0.73	21.40	0.60
beard	0.29	0.35	23.54	0.75	0.33	0.47	21.25	0.67	**0.21**	**0.23**	**31.18**	**0.80**
no beard	**0.10**	**0.07**	**39.58**	**0.94**	0.11	0.09	35.11	0.91	0.15	0.15	32.10	0.83
smile	**0.11**	**0.07**	**35.75**	**0.93**	0.14	0.10	29.87	0.86	0.17	0.16	29.83	0.79
no smile	0.19	0.16	29.63	**0.86**	0.22	0.21	24.31	0.74	**0.17**	**0.15**	**30.96**	0.81
up	**0.22**	**0.23**	24.58	**0.76**	0.24	0.27	22.30	0.71	0.26	0.35	**25.32**	0.67
down	**0.16**	**0.14**	28.63	**0.84**	0.18	0.17	26.13	0.80	0.18	0.22	**33.44**	0.78
right	**0.25**	**0.32**	19.88	**0.60**	0.25	0.32	19.30	0.59	0.29	0.41	**23.72**	0.55
left	**0.22**	**0.27**	21.58	**0.65**	0.22	0.27	20.88	0.64	0.26	0.36	**26.64**	0.60
avg	**0.22**	**0.28**	**28.54**	**0.79**	0.24	0.31	25.71	0.75	0.25	0.33	27.84	0.72

Table 2. Attributes disentanglement measured by the accuracy (higher is better). For each image, we change of the values of attributes listed in rows and verify whether the remaining attributes (listed in columns) stay unchanged. We report the percentage of successes (accuracy). In the last column, we also report the accuracy of modifying the requested attribute (listed in rows).

	gender	glasses	bald	beard	smile	avg.	acc. of modif.
	PluGeN4Faces (ours)						
gender	-	96.99	**90.90**	85.75	89.27	90.72	**91.94**
glasses	**95.25**	-	92.01	86.69	89.48	90.86	99.10
bald	**94.79**	97.17	-	**86.98**	**90.23**	**92.29**	**96.19**
beard	**94.92**	96.46	**93.41**	-	**90.75**	**93.88**	66.91
smile	**95.84**	**96.13**	93.41	86.86	-	**93.06**	**98.14**
avg.						**92.16**	**90.46**
	PluGeN						
gender	-	**97.70**	90.69	**85.81**	89.87	**91.02**	84.28
glasses	93.28	-	92.57	86.77	89.68	90.58	**99.41**
bald	93.74	**97.20**	-	86.48	89.87	91.82	72.37
beard	86.82	**97.14**	93.03	-	90.34	91.83	75.93
smile	92.17	96.05	**93.45**	86.75	-	92.10	97.28
avg.						91.47	85.86
	StyleFlow						
gender	-	95.38	90.46	85.65	**90.23**	90.43	90.52
glasses	94.48	-	**92.82**	**87.09**	**90.42**	**91.20**	98.70
bald	91.86	95.46	-	86.77	87.32	90.35	73.80
beard	83.47	95.80	92.59	-	89.70	90.39	**77.65**
smile	94.92	96.11	93.39	**87.34**	-	92.94	76.04
avg.						91.06	83.34

image (before modification). Again, for a fair comparison, we employ a classifier and apply a minimal modification which is accepted by the attribute classifier.

To compare the difference between images, we apply two approaches. In the first one, we calculate the mean square error (MSE) between embeddings of the original and modified images taken from a pre-trained network. To this end, we employ two networks applicable to processing face images: ArcFace[1] [5] and FR[2]. A model with a lower MSE preserves more features (including identity) from the original image. Second, to explicitly compare the difference between images we also use the PSNR and SSIM measures applied to raw images. Such measures suit perfectly to compare the modification of low-level features such as the background.

Table 1 shows how the proposed measures react to changing subsequent face attributes. Each row corresponds to the requested value of the modified attribute. The results consistently confirm that PluGeN4Faces obtains significantly better scores than PluGeN and StyleFlow in most cases. One can observe that modifying the ”age” attribute has a significant effect on the disentanglement measures, which suggests that changing the age leads to changes in a person’s identity. It is interesting that modifying gender in face images has a moderate influence on face identification. This could mean that both models successfully disentangled this attribute from the remaining image information. The smallest changes are observed for manipulating ”smile” and

[1] https://github.com/deepinsight/insightface

[2] https://github.com/ageitgey/face_recognition

”hair” attributes.

Attributes disentanglement We also verify the disentanglement between labeled attributes in a strict quantitative way. Namely, we force the change of a single attribute and verify whether the values of other labeled attributes changed as well. Ideally, the values of the remaining attributes should stay intact.

For binary attributes (smile gender, glass, hair, and beard), we apply a standard accuracy measure, which shows whether the classifier keeps its original prediction on non-modified attributes. Additionally, we employ a ranking measure, which can be used for discrete as well as continuous attributes because classifier scores do not have to be discretized in this case. In this approach, we rank input (non-modified) images based on the scores returned by the classifier on the attribute A_i. Next, we change the value of the attribute B and again calculate the ranking using the classifier scores based on the attribute A_i. We compare the rankings before and after the change using the Rank Correlation Coefficient (Spearman's ρ), which gives a maximal value of 1, for two identical rankings. Higher values indicate better disentanglement. We repeat this experiment for all attributes $A_1, \ldots, A_k$.

Table 2 shows that all models obtain the average accuracy on non-target attributes above 90% and around 80% on the attributes being modified, which means that it is still more difficult to perform the modification than to keep the values of other features. Taking the average of accuracy scores reveals that PluGeN4Faces outperforms PluGeN and StyleFlow in both metrics. Looking at the ranking correlation presented in Table 3, we observe that the advantage of PluGeN4Faces over PluGeN and StyleFlow is even higher. It gives higher scores in 41 out of 56 cases.

The lowest correlation scores were obtained when we modified the age attribute (which aligns with the conclusion of the previous experiment). It was almost impossible to keep the ranking on the glasses attribute, which might be explained by the fact that the training does not contain young people wearing glasses. Previous sample results presented in Figure 5 also showed that increasing the age attribute accidentally leads to adding glasses. Analogical negative behavior occurs in the case of beard and hair attributes, which are highly correlated with age. This analysis shows that it is very difficult to overcome the bias introduced in a training set and provide high-quality disentanglement between some face attributes.

Table 3. Attributes disentanglement measured by the ranking correlation (higher is better). For each image, we change the values of attributes listed in rows and verify whether the ranking of the remaining attributes (listed in columns) given by the classifier outputs stay unchanged. We report the correlation between rankings before and after the change.

	gender	glasses	bald	beard	smile	age	pitch	yaw
				PluGeN4Faces (ours)				
gender	-	**87.01**	**90.91**	78.83	95.17	96.53	**98.92**	**99.79**
glasses	**93.65**	-	**91.51**	**96.15**	**95.20**	95.83	**98.31**	**99.79**
bald	**93.98**	**89.18**	-	**96.17**	**96.87**	**98.68**	**99.05**	**99.75**
beard	**90.81**	**86.97**	**91.49**	-	**94.48**	96.58	98.54	**99.71**
smile	**95.50**	**88.61**	**95.53**	**96.86**	-	**98.38**	**98.96**	**99.74**
age	**86.02**	**39.34**	**83.36**	**87.94**	**89.28**	-	95.50	**99.58**
pitch	**95.57**	**89.25**	**94.45**	**96.94**	**96.25**	**98.93**	-	**99.82**
yaw	91.41	83.49	90.59	93.66	93.35	96.96	97.60	-
avg	**92.42**	**80.55**	**91.12**	**92.36**	**94.37**	97.41	**98.12**	**99.74**
				PluGeN				
gender	-	86.86	89.73	**79.93**	**95.56**	**96.84**	98.53	99.65
glasses	92.52	-	91.14	95.44	94.09	**96.00**	97.74	99.64
bald	92.95	87.07	-	95.14	95.33	98.11	98.73	99.60
beard	85.43	85.65	88.66	-	93.41	**97.21**	**98.57**	99.47
smile	90.66	85.87	94.02	93.73	-	98.19	98.51	99.59
age	80.06	38.11	76.66	79.38	89.00	-	**96.13**	99.44
pitch	94.47	85.30	94.02	96.41	95.84	98.62	-	99.74
yaw	**92.42**	**84.31**	**92.48**	**95.18**	**94.62**	**98.32**	**98.01**	-
avg	89.78	79.02	89.53	90.74	93.97	**97.61**	98.03	99.59
				StyleFlow				
gender	-	80.42	87.13	65.90	94.37	95.64	97.76	99.42
glasses	91.11	-	90.69	93.99	93.48	95.03	97.65	99.46
bald	89.72	83.20	-	93.86	92.88	97.28	97.97	99.03
beard	80.51	84.55	89.33	-	93.80	95.89	97.84	99.08
smile	92.84	86.65	92.73	95.57	-	97.81	98.30	99.62
age	82.13	34.16	75.60	80.92	88.34	-	93.24	98.74
pitch	90.44	82.07	91.69	94.46	94.15	97.76	-	99.51
yaw	87.72	78.93	86.40	92.49	91.91	95.45	95.53	-
avg	87.78	75.71	87.65	88.17	92.70	96.41	96.90	99.26

5. Conclusion

We introduced PluGeN4Faces for disentangling face attributes from the person's identity. The proposed model works as a plugin to the pre-trained StyleGAN model, which makes it extremely easy to use in practice. Our key idea relies on By applying contrastive learning on images retrieved from movie frames that contain information about a person's identity. Our experiments demonstrate that PluGeN4Faces is focused on manipulating the requested attributes and is less invasive to the remaining image attributes than the existing methods.

Acknowledgement

This research has been supported by the flagship project entitled “Artificial Intelligence Computing Center Core Facility” from the Priority Research Area DigiWorld under the Strategic Programme Excellence Initiative at Jagiellonian University. This research was supported by the National Science Centre (Poland), grant no. 2022/45/B/ST6/01117 (A. Suwała and M. Śmieja) and grant no. 2021/43/B/ST6/01456 (P. Spurek).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images?, 2020.

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021.

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[6] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[7] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

[8] Yue Gao, Fangyun Wei, Jianmin Bao, Shuyang Gu, Dong Chen, Fang Wen, and Zhouhui Lian. High-fidelity and arbitrary face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16115–16124, 2021.

[9] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models, 2018.

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

[11] 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.

[12] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019.

[13] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25:49–59, 2016.

[14] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[15] 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.

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[17] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*, 2014.

[18] Jack Klys, Jake Snell, and Richard Zemel. Learning latent subspaces in variational autoencoders. *arXiv preprint arXiv:1812.06190*, 2018.

[19] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[20] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *arXiv preprint arXiv:1706.00409*, 2017.

[21] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *Proceedings of the European conference on computer vision (ECCV)*, pages 353–369, 2018.

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[23] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Disentangling in latent space by harnessing a pretrained generator. *arXiv preprint arXiv:2005.07728*, 2(3), 2020.

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[25] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[26] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[27] Marek Śmieja and Jacek Tabor. Spherical wards clustering and generalized voronoi diagrams. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2015.

[28] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[29] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[30] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

[31] Hui-Po Wang, Ning Yu, and Mario Fritz. Hijack-gan: Unintended-use of pretrained, black-box gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7872–7881, 2021.

[32] Maciej Wołczyk, Magdalena Proszewska, Łukasz Maziarka, Maciej Zieba, Patryk Wielopolski, Rafał Kurczab, and

Marek Smieja. Plugen: Multi-label conditional generation from pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, number 8, pages 8647–8656, 2022.

[33] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.

[34] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. Indomain gan inversion for real image editing, 2020.

[35] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*, 2020.

[36] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2021.