Face editing using a regression-based approach in the StyleGAN latent space

Saeid Motiian[1]
motiian@adobe.com
Siavash Khodadadeh[1]
khodadad@adobe.com
Shabnam Ghadar[1]
ghadar@adobe.com
Baldo Faieta[1]
bfaietas@adobe.com
Ladislau Bölöni[2]
Ladislau.Boloni@ucf.edu

[1] Adobe Inc
San Jose, CA 95110, USA

[2] Department of Computer Science
University of Central Florida
Orlando, FL 32816, USA

Abstract

Despite significant progress, StyleGAN-based face editing is still limited by undesirable attributes, dependencies and artifacts that decrease the quality of generated images. While more well-annotated training data would likely improve on these problems, collecting such data at scale is very expensive. We propose a face editing architecture that significantly improves the image quality, allows precise specification of individual attributes, and facilitates the introduction of new attributes. We take advantage of recent advances that couple the creation of a latent representation of an image with associated natural language as well as techniques that find linear correlations between the GAN latent space and the attributes of the image, enabling regression models. Our approach deploys carefully chosen regularization approaches that are critical to the integration of these techniques. We demonstrate the ability to edit photorealistic images of faces, originating both from GAN generation and from real images through GAN inversion.

1 Introduction

In recent years, face editing techniques based on generative models such as generative adversarial networks or variational autoencoders had witnessed an explosion in popularity, due to their ability to apply semantically meaningful local and global edits to images without the need to create an explicit representation of the modeled face. One of the most popular architectures that has been used as a basis of face edits is the StyleGAN family of models, due to their well-behaved latent space [10].

While early StyleGAN-based models that allowed the editing of attributes such as pose, age, and facial hair represented spectacular progress compared to previous approaches in the realism and quality of the edited images, significant limitations remained. For instance, for certain face/attribute combinations, the edited images exhibited significant artifacts and

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Figure 1: Examples of the image quality and range of attributes that can be edited with the proposed approach. The projected original image is in the top left followed by several edited images.

unrequested changes to other attributes. An ideal face editing system would allow the fine-grain specification of a wide range of attributes while retaining the unrelated attributes and the personal identity of the face. Furthermore, the system should be easily extendable to new attributes.

A natural way to make progress towards this goal is to use large amounts of high-quality images carefully labeled with the values of all possible attributes.

The primary insight of this paper is that the necessary amount of training data to improve the quality of the edits can be reduced by a better understanding of the latent spaces of the generator and the type of changes that lead to high-quality visual instances. For instance, recent research has shown that the StyleGAN latent space contains smooth linear directions that allow the creation of a regression model for attributes [15]. Furthermore, while there are many points in the latent space that correspond to a given attribute value, we find that a mapper that was trained with specific, well-chosen regularizations can find encodings for edited images that correspond to higher quality edits.

Our final insight is that we can improve the training process without the need of hand-annotated images by taking advantage of recent advances in models that contain joint encodings of images and natural language. For instance, the CLIP [17] architecture can be used to generate annotations for the training data for a wide range of attributes, by providing descriptive labels.

The contributions of this paper can be summarized as follows:

1. We created a face editing architecture that uses regression in the StyleGAN latent spaces with appropriate regularizations to find disentangled attribute directions.

2. We designed a CLIP-based approach to extend the training data for the face editor.

Figure 1 illustrates the image quality and broad range of attributes that can be edited using the proposed approach. Figure 2 shows the high-level concepts of face editing.

2 Related Work

Using GANs to edit images has recently attracted a lot of attention due to the outstanding ability of current generative models to generate high-resolution real-looking photos.

One of the early approaches to control the image generation of GANs was conditional GAN [13] that passes the labels to both generator and discriminator during training to

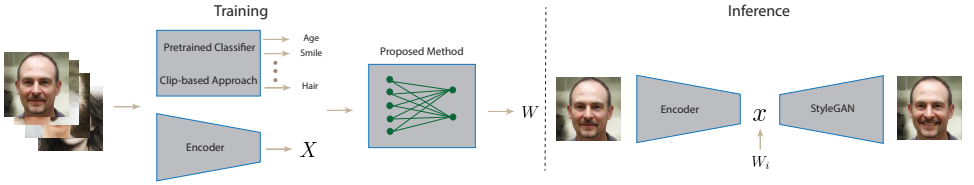


Figure 2: **Training**: After the data preparation step, we train a regression network for available attributes using appropriate regularizations. The attribute directions $W = [W_1, W_2, \ldots, W_{N_a}]$ are the weights of the regression network. **Inference**: For a given face, we first encode it to the StyleGAN latent space and add the direction W_i to edit the attribute i.

condition image generation to different classes. Different techniques were proposed for conditional GANs training using paired [7] or unpaired data [2, 26]

More recent approaches for controlling image generation focus on editing the latent vector corresponding to an initial image such that the newly edited image has the same identity but different attributes like "Smile" or "Facial hair". These approaches significantly reduce the computational resources required to train high-quality models such as StyleGAN [3]. [5] use PCA on sampled latent codes to find directions in the latent space such that moving in those directions could result in change in the target attribute. [25] use low-rank subspace to control the generation of GANs. StyleFlow [1] uses continuous normalizing flows and a neural differential equations solver to train a non-linear function for editing vectors in latent space. [24] train a transformer with losses in latent space to edit attributes.

[18] propose InterfaceGAN based on the assumption that a hyperplane or separation boundary exists for each attribute in the latent space. Thus, by moving in the orthogonal direction to that hyperplane, we can change the target attribute. [11] train a lat-2-lat model to map a latent vector to a new latent such that the image generated from the new one has the target attribute. [21] discover style channels based on gradient maps from the generated image with respect to each channel in S space and identify the channels that control a target attribute.

Recently, [6] leveraged attention to find the layers that have a greater impact on target attributes. This automatic way of finding these layers removes the need for some manual adjustments. [4] use disentangled transforms and instance-aware search to edit a latent vector and corresponding generated image. [8] proposes a neural network that finds directions in the latent space based on target parameters without conditioning the network on the latent vector. StyleRig [19] uses losses defined between the generated image and original image based on the pose, illumination and expression to edit the latent vector. [14] propose a method that combines attributes and face identity features from two different images to generate a new image with losses on identity and attributes. Another method for editing directions is StyleCLIP [16] that leverages CLIP [17] text and image encoder to find S channels in latent space that have the highest correlation with the target text.

3 Face editing

Let us consider an image of a face I, represented in a StyleGAN latent space by a vector x. We will call an attribute of the image *quantifiable* if we can measure the degree at which it is present in the face with a real number in a given interval. Most attributes commonly considered in face editing such as age, degree of smile or the angle of the head position are quantifiable with all possible values in an interval corresponding to realistic images. For other

attributes, such as "wears eyeglasses", intermediate values might not correspond to real-world pictures.

We define the goal of face editing as the modification of a quantifiable attribute a of the image with a positive or negative degree α. To achieve this, we are searching for a direction in the latent space W_a associated with a such that $I_e = StyleGAN(x + \alpha W_a)$ is the image with the desired edit applied.

When we refer to the latent space in which the vector x is encoded, we are considering several possibilities corresponding to different locations in the flow of activations in a StyleGAN architecture: the $\mathcal{W}$ space of the original implementation, the extended $\mathcal{W}^+$ space that allows different styles to be applied at different levels of the hierarchy and the so-called stylespace S which includes the additional affine transformation at each level. Previous research [23] has shown that all the latent spaces of StyleGAN are highly semantic with respect to the attributes. However, there are important differences between $\mathcal{W}$, $\mathcal{W}^+$ and S in the way they map related images, making them suitable for different tasks. For instance, we found that the best results are obtained when using the S space for editing real faces. In the following discussion we will assume that all the images had been encoded in the appropriate latent space using a suitable encoder.

3.1 Model

Let us consider a collection of N images $I_1 \dots I_N$ and a set of N_A attributes $a_1 \dots a_{N_A}$. The training data we are considering will have the form $D = (x_i, y_i^1, y_i^2, \dots, y_i^{N_A})_{i=1}^N$, where x_i is the latent encoding of image I_i and y_i^k is the value of the quantifiable attribute a_k in that image. Determining the attribute values y_i^k is one of the major challenges of this work. As we discuss in Section 4, the values can be either collected by human annotation, pretrained regressors, and/or in an unsupervised manner. Furthermore, our architecture allows the training even if some of the attribute values are missing.

[15] has shown that it is possible to train regression models such that given a StyleGAN latent code x can accurately predict the magnitude of an attribute y^j (age in years or head pose in degrees) in the corresponding image. The regressor achieves this by measuring the distance between a latent code from a separating hyperplane induced by a matching attribute linear latent direction:

$$y^j = W_j^T \cdot x \tag{1}$$

The direction for attribute j can be found by:

$$W_j^* = \underset{W_j}{\operatorname{argmin}} L(S) = \|W_j^T X - Y\|_2 \ , \tag{2}$$

where $Y = [y_1^j, y_2^j, \dots, y_N^j]$ and $X = [x_1, x_2, \dots, x_N]$. x_i is the latent vector of the i-th image in the dataset, y_i^j is the value for the target attribute j, and N is the number of images in training.

This suggests that if we move in the attribute's linear latent direction, the amount of the corresponding attribute of a given image would change. Let's assume $y_1 = W^T \cdot x$ if we move in the attribute direction W we have:

$$y_2 = W^T \cdot (x + \alpha W) = W^T \cdot x + \alpha W^T \cdot W = y_1 + \alpha, \tag{3}$$

assuming that W has a unit norm.

3.2 Regularization techniques

As any solutions satisfying Eq. 1 also change other, unrelated attributes and/or do not keep the identity of the person, we cannot naively use W for face editing. We can restrict the system to solutions that retain the identity of the person and minimize the changes to other attributes by adding regularizers to the optimization criteria. We deployed two types of regularizers based on weight magnitude and weight orthogonality respectively. We found that the appropriate choice of regularizations depends on the latent space used for the input.

Weight magnitude regularizers based on the $L1$ and $L2$ metrics are commonly used when performing regression in a high-dimensional feature space. When used in the context of the $\mathcal{W}^+$ latent space, their primary effect is to reduce the impact of insignificant independent variables and prevent overfitting. In practice, we found that both highly improve the quality of edited images and show similar performance.

The situation is different in the case of the $\mathcal{S}$-space. Recent research had shown that vectors in the $\mathcal{S}$ space are sparse [22]. This makes us strongly prefer the $L1$ regularizer which specifically encourages sparsity, compared to the $L2$ regularizer which does not have such an effect. This is especially important when our starting point is a real-world photograph encoded into the StyleGAN latent space. The complexity of the encoding process usually creates encodings that are only approximately sparse, with small values rather than zeros. Having a sparse direction helps ameliorate this artifact.

Orthogonality regularization: Let us consider two orthogonal attribute directions W_1 and W_2 (meaning $W_1^T \cdot W_2 = 0$) and $y_1 = W_1^T \cdot x$ and $y_2 = W_2^T \cdot x$. We note that y_2 does not change by moving in W_1 direction:

$$y_2^e = W_2^T \cdot (x + \alpha W_1) = W_2^T \cdot x + \alpha W_2^T \cdot W_1 = W_2^T \cdot x = y_2 \tag{4}$$

Similarly, y_1 does not change by moving in the W_2 direction. The same technique can be applied to all the N_A attributes.

In most cases we prefer disentangled attributes: we aim to be able to edit an attribute without changing any of the other attributes. For instance, we want to be able to change the age without changing the gender. Imposing an orthogonality regularization on these attributes can achieve this goal.

It is important to note, however, that not every named attribute is logically independent. For instance, the *Smile* and *Smirk*, while not identical, would likely need to have correlated latent directions. Another example is attribute pairs such as *Beard* and *Facial Hair*, where it is impossible to increase the former without increasing the latter as well. We need to be careful not to apply orthogonality regularizations for such pairs.

Putting all the regularization techniques together, the total loss used to train our network will be

$$W_j^* = \underset{W_j}{\operatorname{argmin}} L(X) = \|W_j^T X - Y\| + \quad \beta_1 L_1(W_j)/L_2(W_j) + \beta_2 \Sigma_{k=1,k\neq j}^{N_A} W_j^T \cdot W_k \,, \tag{5}$$

where X is the latent code (S or $\mathcal{W}^+$) and N_A is the number of attributes available in training.

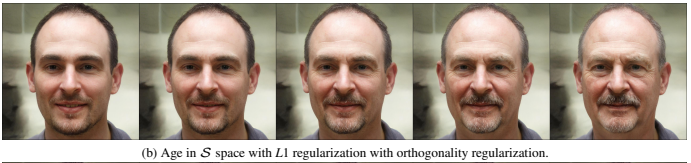
3.3 Model architecture

Our face editing architecture solves Eq. 5 using a multilayer perceptron (MLP) with linear activation. The MLP takes a vector in the $\mathcal{S}$-space or $\mathcal{W}^+$ space as input and outputs N_A values, the number of attributes in training. In this implementation, the attribute directions are the weights of the MLP after training. We can use the mean square loss for the first term in Eq. 5 and adding $L1$ ($L2$) and orthogonality regularizations on the weights are trivial.

Figure 3: Examples for face editing. The unmodified images are in the center.

Figure 4: The effects of $L2$ regularization in $\mathcal{W}^+$ space.

(a) Age in $\mathcal{S}$ space with $L1$ regularization without orthogonality regularization.



(b) Age in $\mathcal{S}$ space with $L1$ regularization with orthogonality regularization.

Figure 5: The effects of orthogonality regularization.

4 Enhancing data collection using joint natural language and image models

Using the approach described in the previous section to find the directions for specific attributes relies on the existence of high-quality training data. While it is comparatively easy to collect a variety of face images, providing quantitative annotations for a range of attributes is a difficult and expensive labeling task. These labels can be acquired through a variety of techniques. For some attributes, manual annotation might be available from existing datasets. For other attributes, such as *Age*, *Smile* or *Hair*, there might exist pre-trained, off-the-shelf or proprietary classifiers or regressors that allow the generation of attributes automatically from the image of a face. In this work, we took advantage of all these techniques.

However, for most attributes, especially newly proposed ones, neither existing annotations, nor pretrained classifiers exist. Furthermore, creating new classifiers would require, on their turn, the existence of labeled data. Another difficulty is that certain attributes, such as *Curly Hair* are subjective, and difficult to consistently label with a numerical value. A possible solution to this dilemma is to take advantage of recent research in joint natural language / image models, such as the CLIP [17] model.

Our main idea is to replace y^j in Eq. 1 with unsupervised scores generated by CLIP. CLIP

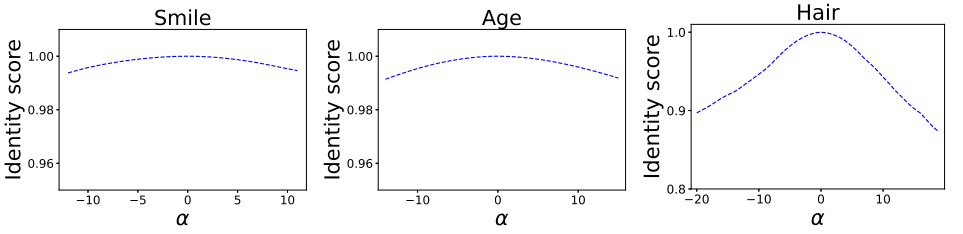


Figure 6: Identity change for face editing. For each value for α and a target attribute, we edit the original image and compute the cosine similarity between Arcface embeddings of the two images: $score = E_o \cdot E_e$ where E_o and E_i are the normalized Arcface embeddings of the original image and the edited image. The numbers are the average scores of 15 faces.

has two encoder networks: the text encoder $\mathcal{E}_T$ and the image encoder $\mathcal{E}_I$. Let us assume we have a set of M phrases for target attributes, $\mathcal{P} = \{p_1, \ldots, p_M\}$. Having an unlabeled dataset $\mathcal{U} = \{I_i\}_{i=1}^N$, we find the CLIP image feature vectors of every image by preprocessing and passing them into $\mathcal{E}_I$. As a result, we have CLIP image features for every image in our dataset.

We also generate a new set of antonyms for every phrase in $\mathcal{P}$. In other words, we have $\mathcal{A} = \{antonym(p_i)\}$ for every phrase $p_i \in \mathcal{P}$. The *antonym* function is based on NLTK library [12]. Next, we calculate the text features using the CLIP text encoder by passing every phrase and their antonyms into $\mathcal{E}_T$. As a result, we get two matrices of size $M \times 512$. We refer to these two matrices by M_P and M_A. For every image in our batch during training, we compute the two following similarity scores: $S_{I_\mathcal{P}} = \mathcal{E}_I(I_i) * M_P$ and $S_{I_\mathcal{A}} = \mathcal{E}_I(I_i) * M_A$. We compute the element-wise softmax between $S_{I_\mathcal{P}}$ and $S_{I_\mathcal{A}}$. This gives us a vector of length M in which every element shows the score for similarity for this phrase compared to its antonym. We use this vector as values for y^j in Eq. 1.

5 Experiments

In this section we evaluate the ability of the proposed technique to find the latent direction and edit attributes of photorealistic images such as *Age*, *Smile*, *Gender*, *Eyeglasses*, *Head Pose*, *Hair*, *Chubbiness*, *Beaming*, and *Curly Hair*.

For training the attribute directions we used 60k images from the FFHQ [9] dataset. We use a pretrained attribute regressor to collect labels for the *Age*, *Smile*, *Gender*, *Eyeglasses*, and *Head Pose* attributes. We used the CLIP-based approach discussed in Section 4 for the *Chubbiness*, *Beaming*, and *Curly Hair* attributes.

To find a latent encoding for an image we use the pretrained e4e encoder [20] to inverse an image to the latent space W^+ followed by the mapping network in StyleGAN to get the $\mathcal{S}$ vectors.

We have investigated the approach using both $\mathcal{S}$ and $\mathcal{W}^+$ as the latent space. Using $\mathcal{S}$-space has the advantage of being more disentangled than the other intermediate latent spaces where, each style channel is shown to control a distinct visual attribute in a highly localized and disentangled manner. We found that the $\mathcal{W}^+$ space has an advantage when an edit requires a significant geometric change, such as for a *Head Pose* edit with a significant angle change.

In the following we discuss the results with respect to several considerations.

A. Attribute Manipulation We use FFHQ images and their collected labels to solve Eq. 5 using the network described in 3.3. For *Chubbiness*, *Beaming*, and *Curly Hair* attributes, we did not use any orthogonality regularization in order to investigate the effectiveness of the

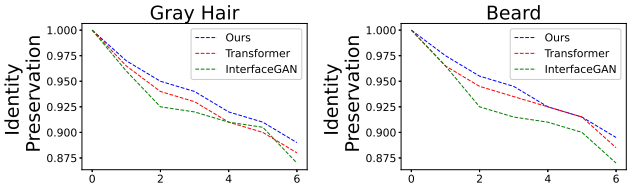


Figure 7: A quantitative comparison of identity preservation based on Figure 4 of [24]. For each method, we edit each target attribute with several scaling factors and generate the modified images.

CLIP-based approach. The output of the network is a set of attribute directions (W_i for i-th attribute). Image editing can be done by adding attribute directions to the S (or W^+) vector. For single attribute editing we have $I_e = StyleGAN(S + \alpha_i W_i)$, where α_i controls the amount of the edit for i-th attribute.

Figure 3(a-d) shows face editing results for four attributes (*Hair*, *Smile*, *Age*, and *Head Pose*). Each row contains five generated images corresponding to five α values, from a negative value to a positive value. The image in the center corresponds to $\alpha = 0$ which is the projected original image. Figure 3 shows that only the target attribute changes when editing an image. All other attributes and the identity stay the same. For *Head Pose*, we solve Eq. 5 using the network described in 3.3 with $L2$ regularization in $\mathcal{W}^+$ space. For the multi-attribute editing, we have $I_e = StyleGAN(S + \sum_{i=1}^{n} \alpha_i W_i)$, where α_i controls the amount of the edit for attribute i. In Figure 3(h), we do several edits given the original projected image in left: making the person younger, adding more hair, smile and eyeglasses. Although this all can be done in one forward pass, we break it down into several parts in order to show the intermediate images. Figure 3(e-g) shows face editing results for the *Curly Hair*, *Chubbiness*, and *Beaming* attributes. Refer to supplementary material for examples of face editing for the *Eyeglasses* and *Gender* attributes.

B. Identity Preservation. For most face editing tasks, the implicit goal is to change attributes without changing the identity of the person represented in the picture. We measure the amount of identity change by comparing the outputs of a pretrained face network (Arcface [3]) given the original image and edited images. The editing results are highly dependent on the parameter α. Since the StyleGAN would generate non-realistic faces for very high/low values of α (because very high/low values push vectors in S or W out of the StyleGAN latent distributions), finding a lower bound and an upper bound for α is crucial. Given that different methods would produce different attribute directions (and different lower/upper bounds) for a target attribute, comparing the identity change amounts is not straightforward. We $L2$-normalize our attribute directions and find the appropriate lower bound and upper bound for α using *manual inspection*. Figure 6 shows the amount of identity change for the applicable range of α. Even for values near the upper bound and lower bound, the amount of the identity drops is small. The cosine distance between the original image and the edited image is less than 0.01. Figure 7 comparing the level of identity preservation for the *Latent Transformer* model [24], *InterfaceGAN* and our approach. We find that our approach has a slight advantage compared to the other approaches.

C. Editing Disentanglement. How much editing a face using a target attribute direction would affect the rest of the attributes? The optimum direction for a target attribute is the one that does not affect the rest of the attributes. We compare the outputs of the pretrained classifier given the original image and the edited image. We use a pretrained classifier that estimates *Smile*, *Age*, *Hair*, and *Gender* attributes given an image. Figure 8 shows the performance for

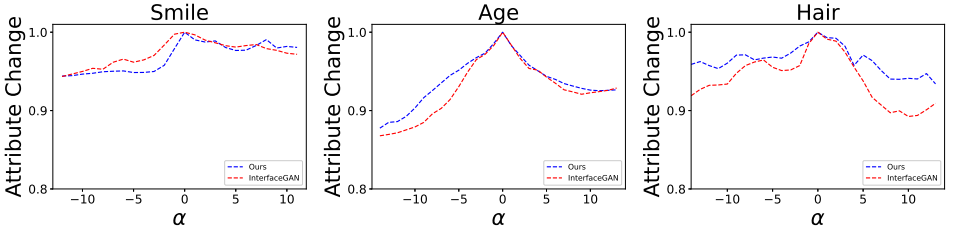


Figure 8: Attribute change for face editing. For the given attribute and α, we edit the original image and compute the $L2$ distance between the outputs of a pretrained classifier given the two images. The numbers are the average scores of 15 faces, and we remove the target attribute when computing $L2$ distance. We report $1 - L2$ instead of $L2$ to be consistent with Figure 6.

the appropriate range of α for *Smile*, *Age*, and *Hair* attributes. Even with the values close to the lower bounds and upper bounds, the attributes change is small. We also normalize directions of InterfaceGAN method with **manual inspection** so that both approaches (ours and InterfaceGAN) generate visually similar faces for a given α. In this way we can compare two methods. For most values of α our method provides less attribute change.

D. Importance of $L2$/$L1$ Regularization. We solve Eq. 5 for the *Head Pose* attribute in $\mathcal{W}^+$ space using the $L2$ regularization and $\beta_1 = 0, 100, 10000$ in order to investigate the benefit of using the regularization. Figure 4 shows that the direction obtained by $\beta_1 = 0$ (not using $L2$ regularization) does not change the *Head Pose* attribute and ruins the quality of the images. The direction obtained by using $\beta_1 = 100$ changes the *Head Pose* attribute but changes the identity of the test image. When β_1 is large enough, the regression model provides the desired attribute direction. We observed similar behavior when training directions in $\mathcal{S}$ space.

E. Importance of Orthogonality Regularization. We train a latent direction for the *Age* attribute without orthogonality regularization in $\mathcal{S}$ space and report the results in Figure 5. The figure shows that there is an entanglement between *Smile* and *Age* for negative values of α, when the person becomes younger. But the latent direction obtained by using orthogonality regularization (Figure 3 (a)) does not show that entanglement.

F. Qualitative Comparison with InterfaceGAN and Latent Transformer [24]. Figures *(1) and (2) of the supplementary material* show the comparison between our method and InterfaceGAN. We trained InterfaceGAN on the same training data, labels, and CLIP labels we used for our model. In general, both methods work well but we noticed some small artifacts (entanglement between attributes) by using InterfaceGAN. For *Hair* and *Age* attributes, there are hair color change for InterfaceGAN-based edits. Also, it looks like the *Age* attribute change (the person looks younger) when increasing *Hair* attribute for InterfaceGAN-based edits. For *Smile* attribute, it looks like eyes become narrower when increasing *Hair* attribute for InterfaceGAN-based edits. Furthermore, we observe better disentanglement when we use CLIP scores with our method. We observe more changes in skin tone, eyes and background for *Curly Hair* and *Beaming* attributes when we train them by InterfaceGAN. We include more examples in the *supplementary material* (Figures (3 - 6)). We saw similar pattern when comparing our approach with the Latent Transformer approach. Please refer to Figure *(7) of the supplementary material for visual comparison.*

G. More Qualitative results. We added visual results for more face attributes and comparisons in the *supplementary material*.

H. Conclusion. We showed that linear regression can be used in the StyleGAN latent spaces for face editing using appropriate regularizers.

References

[1] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *arXiv preprint arXiv:2008.02401*, 2020.

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition (CVPR)*, pages 8188–8197, 2020.

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

[4] Yuxuan Han, Jiaolong Yang, and Ying Fu. Disentangled face attribute editing via instance-aware latent space search. *arXiv preprint arXiv:2105.12660*, 2021.

[5] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *Advances in neural information processing systems (NeurIPS)*, pages 9841–9850, 2020.

[6] Xianxu Hou, Xiaokang Zhang, Linlin Shen, Zhihui Lai, and Jun Wan. GuidedStyle: Attribute knowledge guided style manipulation for semantic face editing. *arXiv preprint arXiv:2012.11856*, 2020.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of the IEEE Conf. on computer vision and pattern recognition (CVPR)*, pages 1125–1134, 2017.

[8] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition (CVPR)*, pages 4401–4410, 2019.

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition*, pages 8110–8119, 2020.

[11] Siavash Khodadadeh, Shabnam Ghadar, Saeid Motiian, Wei-An Lin, Ladislau Bölöni, and Ratheesh Kalarot. Latent to latent: A learned mapper for identity preserving editing of multiple face attributes in stylegan-generated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3184–3192, 2022.

[12] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

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

[14] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *ACM transactions on graphics (TOG)*, pages 1–14, 2020.

[15] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. LARGE: Latent-based regression through GAN semantics. *arXiv preprint arXiv:2107.11186*, 2021.

[16] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proc. of the IEEE/CVF Int'l Conf. on computer vision (ICCV)*, pages 2085–2094, 2021.

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[18] 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 (TPAMI)*, 2020.

[19] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition (CVPR)*, pages 6142–6151, 2020.

[20] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM transactions on graphics (TOG)*, pages 1–14, 2021.

[21] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. *arXiv preprint arXiv:2011.12799*, 2020.

[22] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. In *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition (CVPR)*, pages 12863–12872, 2021.

[23] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *arXiv preprint arXiv:2101.05278*, 2021.

[24] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *Proc. of the IEEE/CVF Int'l Conf. on computer vision (ICCV)*, pages 13789–13798, 2021.

[25] Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zhengjun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in GANs. *arXiv preprint arXiv:2106.04488*, 2021.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the IEEE Int'l Conf. on computer vision (ICCV)*, pages 2223–2232, 2017.