Coarse-to-Fine: Facial Structure Editing of Portrait Images via Latent Space Classifications

YIQIAN WU, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab
YONG-LIANG YANG, University of Bath
QINJIE XIAO, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab
XIAOGANG JIN*, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab

Fig. 1. Given a portrait image (top) with a double chin as input, our method can automatically generate a new portrait without a double chin (bottom). The proposed approach is able to preserve facial identity thanks to the seamless integration of the new chin with the original face through editing the semantic latent code in the StyleGAN latent space.

Facial structure editing of portrait images is challenging given the facial variety, the lack of ground-truth, the necessity of jointly adjusting color and shape, and the requirement of no visual artifacts. In this paper, we investigate how to perform chin editing as a case study of editing facial structures. We present a novel method that can automatically remove the double chin effect in portrait images. Our core idea is to train a fine classification boundary in the latent space of the portrait images. This can be used to edit the chin appearance by manipulating the latent code of the input portrait image while preserving the original portrait features. To achieve such a fine separation boundary, we employ a carefully designed training stage based on latent codes of paired synthetic images with and without a double chin. In the testing stage, our method can automatically handle portrait images with only a refinement to subtle misalignment before and after double chin editing. Our model enables alteration to the neck region of the input portrait image while keeping other regions unchanged, and guarantees the rationality of neck structure and the consistency of facial characteristics. To the best of our knowledge, this presents the first effort towards an effective application for editing double chins. We validate the efficacy and efficiency of our approach through extensive experiments and user studies.

CCS Concepts: • **Computing methodologies → Image manipulation**.

Additional Key Words and Phrases: Face editing, double chin, latent code manipulation, StyleGAN

ACM Reference Format:
Yiqian Wu, Yong-Liang Yang, Qinjie Xiao, and Xiaogang Jin. 2021. Coarse-to-Fine: Facial Structure Editing of Portrait Images via Latent Space Classifications. *ACM Trans. Graph.* 40, 4, Article 46 (August 2021), 13 pages. https://doi.org/10.1145/3450626.3459814

*Corresponding author.

Authors' addresses: Yiqian Wu, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab; Yong-Liang Yang, University of Bath; Qinjie Xiao, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab; Xiaogang Jin, State Key Lab of CAD&CG, Zhejiang University; ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab.

1 INTRODUCTION

Portrait images, which mainly represent human faces, are ubiquitous due to the rapid advances of mobile computing and photographic techniques. They are highly important for social media applications (e.g., Facebook, Twitter) that play an increasingly significant role in human social life. This motivates researchers to investigate computational methods for portrait image editing so that imperfections such as shadows, noises, and distortions can be easily corrected [Arakawa and Nomoto 2005; Scherbaum et al. 2011; Shih et al. 2019; Xiao et al. 2020].

Albeit successful, previous portrait image editing works mainly concern adjusting image colors [Li et al. 2015; Liang et al. 2014; Scherbaum et al. 2011] or globally warping the face region [Shih et al. 2019; Xiao et al. 2020; Zhao et al. 2018]. How to effectively edit facial structure in part or even remove a structural feature is largely unexplored. This is a more challenging problem as both color and shape of the facial structure need to be partially updated, and the edits need to be compatible with the surrounding region to avoid visual artifacts.

As an early exploration of facial structure editing of portrait images, we investigate how to effectively and efficiently edit the chin structure as an important case study. The main reason is that the shape and color of the chin region can easily vary due to head pose and/or lighting condition when taking portrait images, which easily affects the facial structure with a common double chin effect (see the top row in Fig. 1) or not. In this work, we perform chin editing by removing the double chin effect, but our general approach can also be exploited for the reverse process, i.e., adding double chin effect (more discussion in Section 6).

In terms of double chin removal, although conventional image editing tools such as PhotoShop can be used, the manual process requires specific skills and is usually time-consuming [Kelby 2011]. This motivates us to investigate a computational approach that is inherently challenging. First, portrait faces usually have big variations on gender, pose, color, shape, etc. These factors can easily affect the problem setting and result in case-by-case parameter tuning. Also, simply adjusting colors around the double chin to blur, lighten or darken the "pseudo" chin is not satisfactory since the neck preserves its shape and affects the quality of the overall portrait.

Inspired by the recent success of generative adversarial networks (GANs) for high-quality portrait image synthesis [Brock et al. 2019; Karras et al. 2018], especially controlled generation with disentangled facial features (e.g., identity, pose, hairstyle, skin color, etc.) by StyleGAN [Karras et al. 2019], we formulate chin editing as a portrait image synthesis problem in the latent space of portrait images. Our goal is to automatically synthesize a new portrait image without a double chin while keeping the features of the original portrait image beyond the double chin region unchanged. The main idea is to train a fine separation boundary and edit the input portrait image's latent code towards a domain that contains faces without a double chin while preserving original portrait features. This makes sense, as moving latent code along the normal vector of the proposed fine separation boundary can preserve features other than the double chin during the editing process.

In this paper, we present a novel method that achieves the above goal. As the latent codes in StyleGAN do not disentangle the double chin feature, we first train a classifier to score latent codes according to double chin prevalence. To avoid tedious data collection, we directly use randomly sampled latent codes and their corresponding portrait images for training. This yields a coarse separation boundary that can be used to synthesize an intermediate portrait without a double chin. However, other facial features, such as shape and pose, cannot be well preserved. To resolve this, we introduce a diffusion process to blend the original portrait and the intermediate portrait with the help of a neck mask that can separate the double chin feature from other features, resulting in paired portrait images with and without a double chin and their corresponding latent codes. Finally, we utilize the paired latent codes to train a fine double chin separation boundary, allowing for chin editing while keeping other facial features in the testing stage. The subtle misalignment of input and output faces is further refined by image warping. We evaluate our work by demonstrating high-quality chin editing results on portrait images with varying gender, ages, skin color, pose, etc. We also conduct a user study to show that our results accord with human preferences.

In summary, our work makes the following major contributions:

- We present the first automatic chin editing method for portrait images. It can generate a new facial structure without double chin while consistently leaving other regions unchanged.
- We introduce a novel facial editing approach at the structural level based on coarse-to-fine separation boundary training, which allows direct editing in the latent space of the portrait image with plausible semantic manipulation and facial identity preservation.
- We create the first large-scale chin editing dataset to facilitate future research. The dataset contains 13,990 pairs of realistic portrait images with and without a double chin. [1]

2 RELATED WORK

2.1 Face Manipulation for Portrait Images

There is an increasingly wide range of face manipulation applications, since people pay ever greater attention to their portrait images on social networks. Editing the facial texture can cultivate a more captivating portrait appearance. Style transfer methods [Chang et al. 2018; Shih et al. 2014] transfer the style of a given portrait photo onto a new one. In addition to skin texture, lighting [Sun et al. 2019; Zhou et al. 2019] and shadow [Zhang et al. 2020] can also be controlled. While some works are adopted for manipulating a single facial attribute [Li et al. 2016; Shen and Liu 2017], another group of deep-learning-based methods exploit a wide variety of intermediate representations. For example, MaskGAN [Lee et al. 2020] learns to traverse on the mask manifold to manipulate the face, and provides the freedom to edit images interactively. SEAN [Zhu et al. 2020a] achieves localized facial editing with segmentation masks and style reference image as input. SC-FEGAN [Jo and Park 2019] leverages more flexible freeform user input to generate high-quality synthetic images, such as face mask, sketch, and color. However, all of the

[1] https://github.com/oneThousand1000/coarse-to-fine-chin-editing

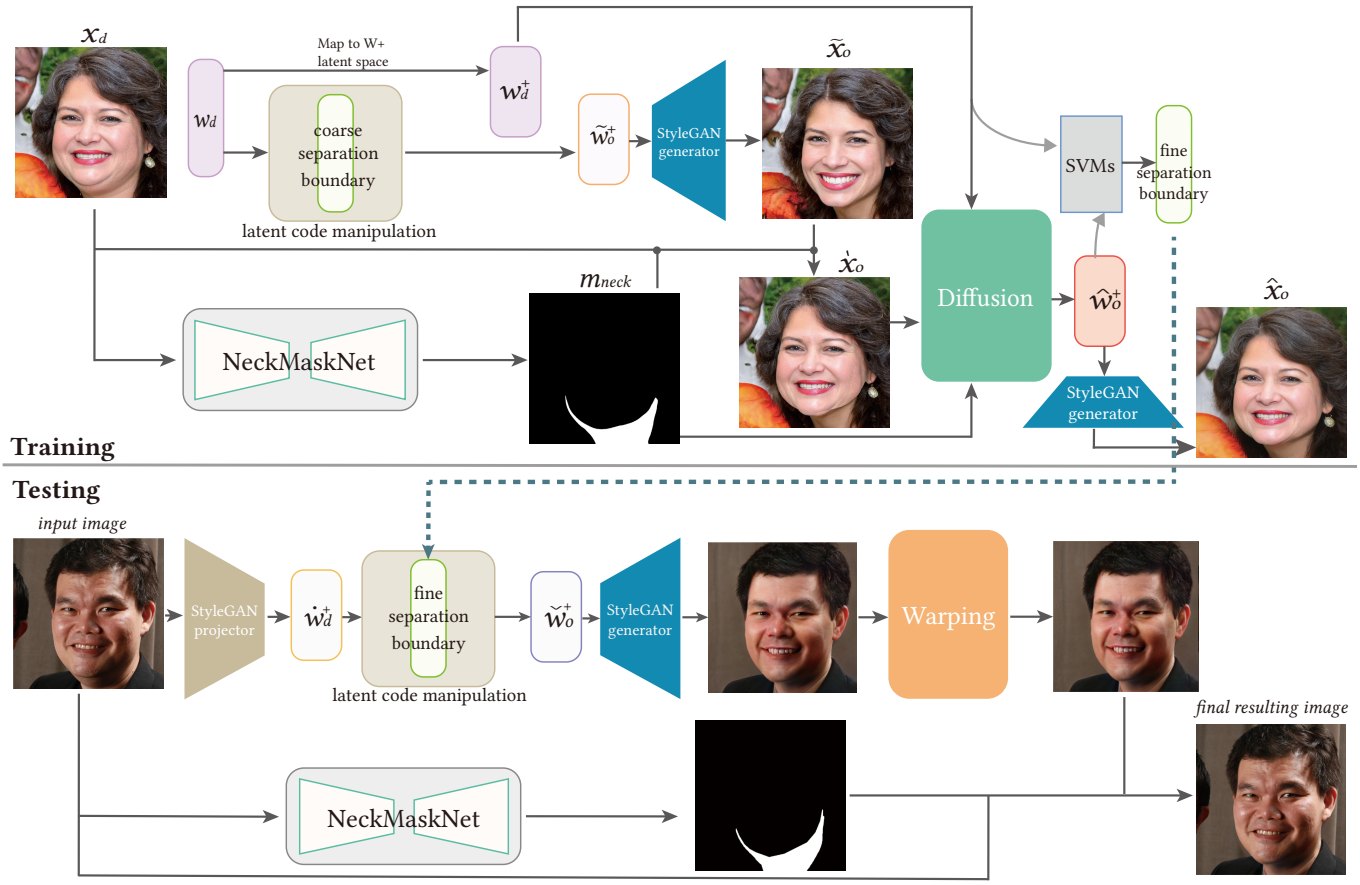

Fig. 2. The pipeline of our model. In training, our goal is to train a fine separation boundary. Given a latent code w_d and its corresponding portrait image x_d with visible double chin, we first use a coarse separation boundary (Subsection 3.2) to edit w_d and get $\tilde{w}_o^+$, which is fed into the StyleGAN generator to get a new portrait image $\tilde{x}_o$. w_d is mapped to the $W+$ latent space and get w_d^+. Meanwhile, our $NeckMaskNet$ (Subsection 3.3) automatically extracts a neck mask m_{neck} from x_d. Then, we get a synthetic image $\dot{x}_o$ by using m_{neck} to paste the neck region of $\tilde{x}_o$ onto x_d. After that, $\dot{x}_o$, m_{neck} and w_d^+ guide our $Diffusion$ approach (Subsection 3.4) to generate an optimized latent code $\hat{w}_o^+$ and a portrait image $\hat{x}_o$. We finally leverage $\hat{w}_o^+$ and w_d^+ to train a fine separation boundary (Subsection 3.5). Note that the coarse separation boundary has worse performance in facial preservation than the fine separation boundary. In testing, we first use a projector to project the input real image to the latent code $\dot{w}_d^+$. Then we use the fine separation boundary to edit $\dot{w}_d^+$ and generate a new latent code $\check{w}_o^+$ along with a new portrait image. Finally, we input the new portrait image to our $Warping$ approach (Subsection 3.6) to correct the misalignment and generate the final resulting image by pasting the new portrait's neck region onto the original image. Input image is provided by FFHQ and Flickr user Kevin Lee (CC BY-NC 2.0).

above methods do not work well in eliminating double chin since generating a new chin requires further geometric considerations than simply editing facial color and facial shape. Variational Autoencoders can also be used for high-level manipulation of facial expressions [Yeh et al. 2016]. For 3D-based face manipulation, the 3D morphable model (3DMM) [Blanz and Vetter 1999] allows geometrical face editing. Abundant 3D face information and constraints lead to more effective results of portrait reshaping [Xiao et al. 2020].

2.2 Portrait Synthesis based on GANs

Since first introduced in [Goodfellow et al. 2014], generative adversarial networks (GANs) have been widely used to synthesize realistic images. Recently, researchers have explored a series of approaches to increase the model capability and result quality of GANs [Brock et al. 2019; Gulrajani et al. 2017; Karras et al. 2018; Mao et al. 2017; Radford et al. 2016]. For the specific portrait synthesis problem, a face completion algorithm based on a deep generative model [Li et al. 2017] can directly generate contents from random noise for the missing regions. Chen et al. [2020] generate high-quality face images from freehand sketches(even rough and/or incomplete sketches), which serve as soft constraints. StarGAN [Choi et al. 2018] achieves image-to-image translations for multiple domains using a single model. However, it is still challenging how semantic attributes are determined by latent code in the latent space.

2.3 Latent Code Manipulation

Our work is related to latent code manipulation. Adapting the image prior learned by GANs to image statistics [Bau et al. 2019] allows for semantic image editing while reconstructing the input image. mGANprior [Gu et al. 2020] leverages multiple latent codes to generate and compose multiple feature maps at intermediate layers. InterFaceGAN [Shen et al. 2020] explores how semantics are encoded in the latent space of GANs. GANSpace [Härkönen et al. 2020] explores the linearity of the GAN space and its latent directions based on Principal Component Analysis. Fader Networks [Lample et al. 2017] rely on an encoder-decoder architecture to learn a disentangled latent space. StyleGAN [Karras et al. 2019] has state-of-the-art performance in generating high-quality portrait images with disentangled styles. StyleGAN2 [Karras et al. 2020] redesigns the model architecture and training methods of StyleGAN and improves distribution quality metrics and image quality. StyleGAN has an intermediate latent space W, in which facial features can be disentangled. The capability to embed the input latent code into the latent space W makes StyleGAN applicable to face latent code manipulation tasks. In-domain GAN inversion [Zhu et al. 2020b] proposes an encoder that not only projects a given image to the latent space of GANs, but also serves as a regularizer to fine-tune the latent code. Image2StyleGAN [Abdal et al. 2019] embeds a given image into the latent space of StyleGAN and applies attribute level feature transfer to images. Although in-domain GAN and Image2StyleGAN can transfer new semantic information to the target image, they require a separate source image with feasible semantics while our method can automatically edit the target image. GIF [Ghosh et al. 2020] applies StyleGAN2 to FLAME (a generated 3D face model) to improve disentanglement and produces photo-realistic images with explicit control. StyleRig [Tewari et al. 2020] achieves a face rig-like control based on StyleGAN and a 3DMM, and it provides explicit control over a set of semantic control parameters. However, the above methods cannot be used in our scenario due to mainly focusing on the face region.

3 METHODOLOGY

In this section, in order to discuss our chin editing method specifically, we first give an overview of the proposed method, then present its major components, finally elaborate on the implementation details.

3.1 Overview

Fig. 2 shows an overview of our novel GANs-based method that can automatically edit the double chin from an input portrait image while preserving facial identity. Given a portrait image with a visible double chin and its corresponding latent code (encoded by the StyleGAN projector), our goal is to edit the latent code using a fine separation boundary of chin structure (with or without double chin), such that a new portrait image without double chin can be generated while keeping the rest region of the portrait unchanged. Our pipeline consists of two stages. In the training stage, we aim to achieve coarse-to-fine separation boundary training. We first use randomly sampled latent codes and their corresponding portrait images to train a coarse separation boundary (Subsection 3.2), such that the input latent code can be edited by moving along the normal vector of the coarse separation boundary to generate a portrait image with a new chin. We map the latent codes to the $W+$ latent space and apply a $StyleMixing$ approach to preserve skin characteristics. To better preserve facial identity, the original portrait image is fed into the $NeckMaskNet$ (Subsection 3.3) to extract a neck mask, which is then employed to get an intermediate synthetic image from the two portrait images, serving as a better facial prior. Since the intermediate portrait image's posture and shape of the face are different from the original one, there exists a visible misalignment in the intermediate image. We therefore apply a diffusion method to eliminate the problem by putting the intermediate image, the neck mask, and the latent code to the $Diffusion$ process (Subsection 3.4). $Diffusion$ outputs an optimized latent code along with the resulting portrait image. Finally, we leverage the optimized latent codes to train a fine separation boundary (Subsection 3.5). In the testing stage, the input portrait image is fed to a StyleGAN projector to get its latent code. The fine separation boundary directly edits the latent code and generates a new portrait image. Although the fine separation boundary has better performance in preserving facial identity, subtle misalignment still exists along the face's edge in the new portrait image. In $Warping$ process (Subsection 3.6), we perform a warping operation to eliminate such misalignment in order to get the resulting image.

3.2 Coarse Double Chin Separation Boundary

This subsection proposes a coarse double chin separation boundary to edit the input latent code. We aim to edit the original portrait image x_d's latent code w_d and generate a new portrait image with the double chin eliminated. We employ the separation hyperplane training approach by InterFaceGAN [Shen et al. 2020] to explore the separation boundary in the latent space. Facial semantics can be edited by moving along the normal vector of the separation boundary. The latent space W of StyleGAN is "unwrapped" from the latent space Z and is less entangled with more linearity, preventing StyleGAN from generating mixed face styles. We therefore search the double chin's separation boundary on disentangled latent space W and use this separation boundary to eliminate the double chin.

To do so, we need to bridge the space of the latent code and the space of double chin. As the synthesis network g in StyleGAN can map the latent code to the image space X, this can be achieved at the higher dimensional image level. A classifier named $DoubleChinClassifier$ is developed to score a portrait image that is generated from latent code as:

$$s = C(g(T(w, \psi))), \tag{1}$$

where s denotes the score, C denotes the $DoubleChinClassifier$, $g(T(w, \psi))$ denotes the synthesized portrait image, T denotes the truncation trick and stack operation of StyleGAN (as shown in Fig. 3), and ψ denotes the hyperparameter in truncation trick that scales the deviation of the given w from the center $\bar{w}$. We suppose that an image with double chin will be scored as $s = 1$, otherwise $s = 0$.

DoubleChinClassifier Training Data. We first collect dataset $X_o = \{x_o^i \mid_{i=0}^{n_o}\}$ (w/o double chin) and $X_d = \{x_d^i \mid_{i=0}^{n_d}\}$ (w/ double chin). We select 1,100 portrait images from CelebAMask-HQ [Lee et al.

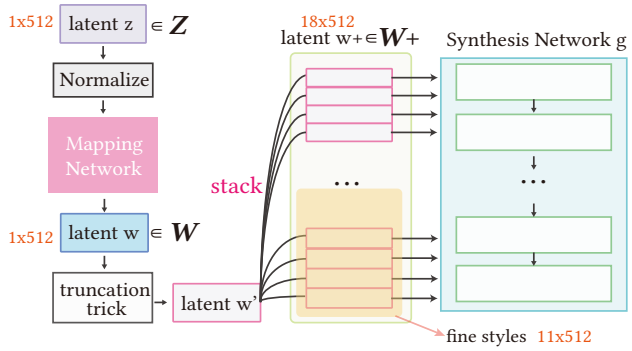

Fig. 3. The pipeline of StyleGAN. The randomly sampled latent code z with the shape of $[1, 512]$ is normalized and fed into Mapping Network. The Mapping Network maps z to the latent space W and outputs w, then the truncation trick T is applied to w before it is fed into each layer in Synthesis Network g. A new stacked latent code w^+ with the shape of $[18, 512]$ can be obtained by stacking each layer's input of Synthesis Network g, the space of w^+ is noted as $W+$. We define the last 11 layers of w^+ as w^+'s fine styles.

2020] and 1,814 portrait images from FFHQ [Karras et al. 2019] that have visible double chins. We also select 3,001 portrait images without double chins from CelebAMask-HQ. The whole dataset can be denoted as:

$$(X, S) = \{(x_d, 1) \mid x_d \in X_d\} \cup \{(x_o, 0) \mid x_o \in X_o\}. \quad (2)$$

The portrait images with double chin scores are randomly split into a training set of size 5,598 and a testing set of size 317.

We use *ResNeXt*-50($32\times4d$) architecture [Xie et al. 2017] to train *DoubleChinClassifier*. Specifically, we generate a rectangle mask for each image and only feed the neck region into *DoubleChinClassifier* to avoid the interference of other redundant information (see Fig. 4).

Separation Boundary Training Data. We randomly sample 50,000 latent codes in the latent space W, and generate 50,000 portrait images. Then we leverage *DoubleChinClassifier* to label all these portrait images and build the separation boundary training dataset as:

$$(W, S) = \{(w_i, s_i) \mid_{i=0}^{50,000}\}, \quad (3)$$

where w_i denotes the latent code, and s_i denotes the double chin score. In (W, S), there are 7,766 w_i's whose scores are set as $s = 1$, otherwise $s = 0$.

For the separation boundary training, we randomly choose 7,523 latent codes with $s = 1$ and 7,523 latent codes with $s = 0$ from (W, S), split them into a training set of size 13,540 and a validation set of size

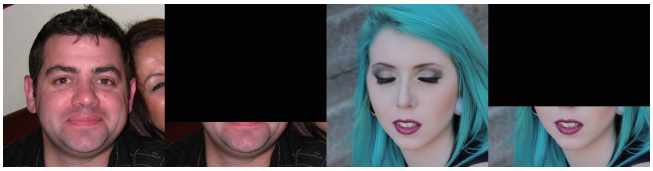

Fig. 4. From left to right: x_d, masked x_d, x_o, masked x_o. The masked images are the inputs of *DoubleChinClassifier*. Images are provided by FFHQ and Flickr user lu_lu, Will Mendonça (CC BY-NC 2.0).

Fig. 5. The results of latent code editing in latent space W and in the latent space Z. From left to right: the original image x_d, the image $\bar{x}_o$ edited by the separation boundary in the latent space W, the image edited by the separation boundary in the latent space Z, and the *StyleMixing* result $\tilde{x}_o$.

1,506, and then use SVMs to train the coarse separation boundary b_c. We describe the boundary as "coarse" since the separation boundary has worse performance in facial identity preservation than a fine separation boundary which will be addressed later. The output of latent code manipulation is illustrated in Fig. 5. To show the latent space W is less entangled, we also train the separation boundary in the latent space Z for comparison. The separation boundary edits latent code as:

$$\bar{w}_o = w_d + \alpha \times n_c, \quad (4)$$

where $\bar{w}_o$ denotes the edited latent code, n_c is the normal vector of separation boundary b_c, and α is a hyperparameter controlling the weight of n_c.

As shown in Fig. 5, after editing latent code w_d, we get the latent code $\bar{w}_o$ and a portrait image $\bar{x}_o$ without double chin. However, the skin characteristics of $\bar{x}_o$ is different from that of x_d. Therefore, we use the *StyleMixing* approach to preserve the skin characteristics as follows. First, we apply the truncation trick and stack operation to $\bar{w}_o$ and w_d (as shown in Fig. 3), then we get $\bar{w}_o^+ = T(\bar{w}_o, \psi) \in W+$ and $w_d^+ = T(w_d, \psi) \in W+$. Second, we replace the fine styles of $\bar{w}_o^+$ with those of w_d^+ (see the definition of fine styles in Fig. 3), and get the mixed latent code $\tilde{w}_o^+$ as:

$$\tilde{w}_o^+ = \sigma(\bar{w}_o^+, w_d^+), \quad (5)$$

where σ denotes the *StyleMixing* operation. Finally, we generate a new portrait image $\tilde{x}_o$ which preserves the geometric features of $\bar{x}_o$ and the skin characteristics of x_d (see Fig. 5).

3.3 Mask Generation

After portrait editing by coarse separation boundary b_c, we get a new portrait image $\tilde{x}_o$. Although $\tilde{x}_o$ contains the geometric features we need, facial identity is not well preserved, especially in the region above the chin. To better preserve facial identity, we further propose *NeckMaskNet* to generate a neck mask in order to allow other facial regions to be directly copied from the original image x_d.

Lee et al. [2020] contribute the CelebAMask-HQ dataset that provides a hand-annotated mask dataset, and such a dataset is leveraged to train a *Faceparsing* model that generates masks of all facial components and accessories. We try to exploit this approach, but it turns out that *FaceParsing* cannot accurately predict a neck mask that suits our needs, and some of the hand-annotated neck masks in CelebAMask-HQ are not precise. One example is shown in Fig. 6, where the neck mask generated by *FaceParsing* cannot fully cover the chin region, causing problematic chin editing results. We thus

Fig. 6. The comparison between the results of two neck mask generating methods. From left to right: the original image provided by FFHQ and Flickr user USAID Asia (CC BY-NC 2.0), the mask generated by *FaceParsing*, and the mask generated by our method *NeckMaskNet*.

propose using the landmarks of the chin to refine the *FaceParsing* prediction, and attempt several face landmark detection methods, such as 3DDFA [Guo et al. 2020; Zhu et al. 2019], face alignment [Bulat and Tzimiropoulos 2017], and PRNet [Feng et al. 2018], to extract chin landmarks. Our experiments (Section 4) show that the face alignment method provides the most stable results.

The process of *NeckMaskNet* is illustrated in Fig. 7. The neck mask m_{neck} is obtained from the *FaceParsing* outputs while being constrained using the chin polyline l_{chin} as follows:

$$m_{neck} = l_{chin} \ominus (m_{face} \cup \dot{m}_{neck}), \tag{6}$$

where l_{chin} denotes the polyline composed by chin landmarks, m_{face} and $\dot{m}_{neck}$ denote the face mask and neck mask predicted by *FaceParsing*, $\cup$ is the union operation, and $\ominus$ is the cut operation that only retains part of the mask below l_{chin}. We also perform anti-aliasing at the mask boundary to achieve smooth editing results.

3.4 Diffusion

After extracting the mask m_{neck} from x_d using *NeckMaskNet* (see subsection 3.3), we copy and paste the neck region of $\tilde{x}_o$ onto x_d as:

$$\dot{x}_o = m_{neck} \odot \tilde{x}_o + (1 - m_{neck}) \odot x_d, \tag{7}$$

where $\odot$ denotes the element-wise multiplication, and $\dot{x}_o$ denotes the synthetic image. Note that the neck region of x_d is always larger than $\tilde{x}_o$, thus the neck of $\tilde{x}_o$ can be cut completely. Fig. 8 shows an example of x_d and $\dot{x}_o$.

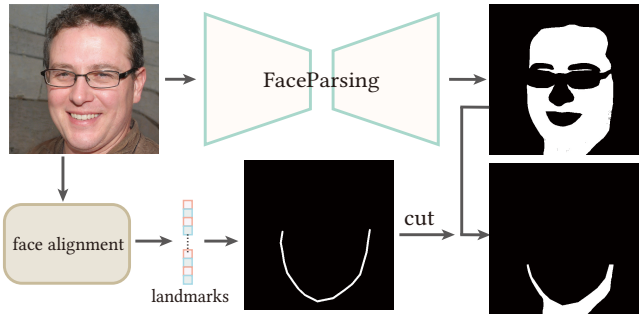

Fig. 7. Neck mask generation pipeline and the construction of *NeckMaskNet*.

Fig. 8. Neck region handling. From left to right: x_d, $\dot{x}_o$, $\hat{x}_o$.

It can be seen that there is non-negligible inconsistency in $\dot{x}_o$ (as highlighted by the green box in Fig. 8). Our goal in this section is to optimize $\dot{x}_o$ and get an image $\hat{x}_o$ without such artifacts. Given that $m_{neck} \odot \tilde{x}_o$ provides geometric features and skin characteristics of the new chin, and $(1 - m_{neck}) \odot x_d$ provides the face features of x_d, $\dot{x}_o$ can be utilized as priors to guide the optimization.

Inspired by the in-domain GAN inversion method [Zhu et al. 2020b], we employ a *Diffusion* process to perform optimization on the latent code w_d^+ under the guidance of $\dot{x}_o$. This achieves a smooth transition between the original face and the new chin.

Let $\hat{w}_o^+$ denote the optimized latent code. To preserve facial identity, we use a reconstruction loss to penalize the pixel difference between $g(\hat{w}_o^+)$ and $\dot{x}_o$ except the neck region:

$$L_r = \|(1 - m_{neck}) \odot g(\hat{w}_o^+) - (1 - m_{neck}) \odot \dot{x}_o\|_2. \tag{8}$$

We also adopt a pre-trained VGG16 model for the perceptual loss to penalize the high-level (geometric) feature difference between $g(\hat{w}_o^+)$ and $\dot{x}_o$:

$$L_p = \|\phi(g(\hat{w}_o^+)) - \phi(\dot{x}_o)\|_2, \tag{9}$$

where ϕ denotes the pre-trained VGG16 model.

Our overall diffusion objective function is as follows:

$$L = \lambda_r L_r + \lambda_p L_p, \tag{10}$$

where λ_r and λ_p are hyperparameters to weigh different losses.

We use w_d^+ as the initial value of $\hat{w}_o^+$, and optimize the latent code by minimizing the full loss L, which forces $\hat{x}_o = g(\hat{w}_o^+)$ to maintain geometric features of $\dot{x}_o$ and preserve the facial identity of x_d.

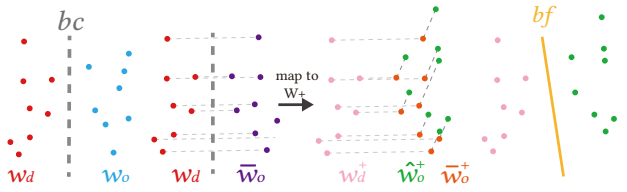

Fig. 9. An illustrative example of how to obtain a fine separation boundary. Each image from left to right: Training the coarse separation boundary (the gray dotted line) from randomly sampled w_d and w_o (red points and blue points). Using a coarse separation boundary to edit w_d (red points) and get $\bar{w}_o$ (purple points). Mapping w_d and $\bar{w}_o$ to the $W+$ latent space, getting w_d^+ (pink points) and $\bar{w}_o^+$ (orange points). Using images generated from w_d^+ and $\bar{w}_o^+$ as prior information to seek $\hat{w}_o^+$ (green points). Training the fine separation boundary in the $W+$ latent space (the yellow solid line) from w_d^+ (pink points) and $\hat{w}_o^+$ (green points).

Fig. 10. The comparison between coarse separation boundary and fine separation boundary. From left to right: the original image generated from w_d^+, image generated from $\sigma(w_d^+ + \alpha \times n_c, w_d^+)$, and image generated from $\sigma(w_d^+ + \alpha \times n_f, w_d^+)$. Notice that we set the same hyper-parameter α.

3.5 Fine Double Chin Separation Boundary

In Section 3.2, we described how to use SVMs to train a separation boundary b_c that coarsely separates latent codes causing double chin or not. The training data presents various synthesized portraits that are scored by classifier C. The disadvantages of using b_c are two-fold. First, it is obvious that an individual's double chin is difficult to be detected when looking down. As such, C is more likely to label portraits looking down as $s = 0$. Therefore, the portrait image $\tilde{x}_o$ always tends to have a downward posture, which is different from the original image x_d that comparatively appears looking ahead (see Fig.10). We also use 3DDFA [Guo et al. 2020; Zhu et al. 2019] to evaluate pose score and build a new separation boundary training dataset to train a pose separation boundary as we did in Subsection 3.2, then use the Conditional Manipulation approach proposed by InterFaceGAN [Shen et al. 2020] to get a new normal vector. We hope the new vector can eliminate double chin without affecting posture, but the attribute of double chin and the attribute of posture are difficult to disentangle. Second, b_c has poor performance in preserving facial identity. This is because we use randomly sampled latent code w_d and w_o (corresponding to synthesized images with and without double chin) to train coarse separation boundary b_c, and this cannot preserve facial identity as w_d and w_o have no paired information with each other.

On the other hand, after diffusion, we get latent code pairs $\hat{w}_o^+$ and w_d^+. Also, $\hat{w}_o^+$ is close to the latent code edited by an "ideal" separation boundary that can remove double chin while preserving facial identity. Considering the linearity of latent space, we assume that w_d^+ can be converted to $\hat{w}_o^+$ by linear transformation instead of using diffusion. Therefore, we leverage w_d^+ and $\hat{w}_o^+$ to train a fine separation boundary b_f in the latent space $W+$ (see Fig. 9).

To achieve coarse-to-fine separation boundary training, we first build a new separation boundary training dataset:

$$(W+, S)' = \{(w_d^+, 1)\} \cup \{(\hat{w}_o^+, 0)\}. \tag{11}$$

We then train the fine separation boundary on the new dataset and apply it to real image processing. Given a real image as input, we first use the projector provided by StyleGAN to get the corresponding latent code $\dot{w}_d^+$, then we use the normal vector of b_f to edit $\dot{w}_d^+$ similar to what we did with b_c as:

$$\check{w}_o^+ = \sigma(\dot{w}_d^+ + \alpha \times n_f, \dot{w}_d^+), \tag{12}$$

Fig. 11. The warping strategy on a real image provided by FFHQ and Flickr user Moodycamera Photography (CC BY-NC 2.0). Images in the first row (from left to right): The synthetic image without warping, the synthetic image with warping (the green box highlights misalignment). Images in the second row (from left to right): the original image with selected points, the edited image with selected points, the original image with Delaunay triangulation, and the edited image with Delaunay triangulation.

where α denotes a hyperparameter controlling the weight of b_f's normal vector n_f, and σ denotes the *StyleMixing* operation mentioned in Subsection 3.2. As shown in Fig. 10, b_f can not only eliminate double chin but also preserve facial posture and identity. More results can be found in Section 4.

3.6 Warping

As shown in Fig. 11, although b_f can help to generate $x_{b_f} = g(\check{w}_o^+)$ with facial identity preservation, there still exists subtle misalignment along the face edge (as highlighted by the green box in Fig. 11) in the image $x = m_{neck} \odot x_{b_f} + (1 - m_{neck}) \odot x_d$. Thus we introduce $Warping$ to diminish the misalignment in this process.

$Warping$ operates as follows. First, we use $FaceParsing$ to extract the face skin mask and the neck skin mask from x_{b_f} and x_d, and select points as:

$$P^{neck} = \{(px_i^{neck}, py_i^{neck}) \mid_{i=0}^{10}\}, \tag{13}$$

$$P^{face_1} = \{(px_i^{face_1}, py_i) \mid_{i=0}^{10}\} \cup P^0 \cup P^{neck}, \tag{14}$$

$$P^{face_2} = \{(px_i^{face_2}, py_i) \mid_{i=0}^{10}\} \cup P^0 \cup P^{neck}, \tag{15}$$

where $(px_i^{face_1}, py_i)$ denotes the points on the face's edge from x_d, $(px_i^{face_2}, py_i)$ denotes the points on the face's edge from x_{b_f}, and P^0 denotes a fixed points set. Note that $(px_i^{face_1}, py_i)$ and $(px_i^{face_2}, py_i)$ have the same y coordinate. $(px_i^{neck}, py_i^{neck})$ denotes the points on the neck's edge from x_{b_f}. In order to keep the shape of neck unchanged, we use the same neck points set P^{neck} to build P^{face_1} and P^{face_2}.

We use point set P^{face_1} to construct a Delaunay triangulation D, then we leverage D to warp x_{b_f}. Namely, we pull P^{face_2} to P^{face_1} so that the face edge of x_{b_f} is forced to align with the face edge of x_d. Then we get the resulting portrait image as:

$$x = m_{neck} \odot warp(x_{b_f}) + (1 - m_{neck}) \odot x_d, \tag{16}$$

where $warp$ denotes the warp operation in $Warping$.

Based on the above analysis, we obtain a fine separation boundary that can directly manipulate semantics in the latent space while preserving facial identity. The misalignment along the face edge in our results can be efficiently addressed by a simple warping operation. In Eqn.16 we use the original image to perfectly preserve the input image contexts, even the difference between the non-neck region of the original image x_d and the synthesized images x_{b_f} is subtle. By now, we achieve the double chin removal result.

3.7 Implementation Details

Our implementation is based on StyleGAN2 as it further improves the distribution quality metrics and image quality. Our system requires the latent code of the input portrait image. Therefore when we apply our method to a real image, we use the projector provided by StyleGAN2 to encode the real image and get its corresponding latent code.

For the neck mask generated from *NeckMaskNet*, we perform anti-aliasing to make the transition more smooth at the edge of the mask. More specifically, we first dilate the neck mask using a 15×15 kernel, then blur the mask using a 25×25 kernel, finally adding the blurred mask to the original mask.

For *Diffusion*, we use w_d^+ as the initial value of $\hat{w}_o^+$. We use the Adam solver as the optimizer by setting $\lambda_r = 1.0$ and $\lambda_p = 5e^{-5}$. The latent code is optimized with a learning rate of 0.01, and we train each image for 100 iterations.

For *Separation Boundary Training*, we use SVMs to train separation boundaries. For the training of b_f, we define each latent code w^+ as:

$$w^+ = [w_0, w_1, \cdots, w_{16}, w_{17}], \tag{17}$$

where each $w_i (0 \leq i \leq 17)$ is a 1×512 vector. We define the dataset $(W+, S)'$ as the combination of 18 datasets:

$$(W+, S)'_i = \{(w_d^i, 1)\} \cup \{(\hat{w}_o^i, 0)\} (0 \leq i \leq 17), \tag{18}$$

where w_d^i and $\hat{w}_o^i$ denote the vector in w_d^+ and $\hat{w}_o^+$. Then we train 18 normal vectors separately using those 18 datasets with the same strategy similar to the one we use in the latent space W. The normal vector n_f of size 18×512 is the stack of the 18 normal vectors:

$$n_f = [n_0, n_1, \cdots, n_{16}, n_{17}]. \tag{19}$$

4 EVALUATION

In this section, we show the results of the extensive evaluation of our method. We first present some of the achieved results and their computational performance, then show ablation studies to verify our methodology. Finally, our method is compared with prior works that have to be adapted to remove the double chin, in order to demonstrate its advantages.

4.1 Results

We test our method on a vast number of portrait images with variations on gender, pose, face size, skin color, etc. Fig. 17 shows a gallery of results automatically generated from our method. It can be seen that the double chin is successfully removed and other features of the original portrait image are well preserved (We empirically set the default value of hyper-parameter α in Eqn. 12 as -6 and fine tune the value of α for some extreme cases). As our method is based on latent code manipulation, for images in the wild, we first encode the image into the $W+$ latent space using the projector provided by StyleGAN2. Note that by changing hyper-parameter α in Eqn. 12, we can further quantify the degree of double chin removal. This provides extra freedom for the user to control the result (see supplemental video for results with continuously adjusted α).

4.1.1 Computational Performance. The running time statistics of individual steps in our method are described as follows. On average, projecting a real image to latent code takes 232.15 seconds per image. Using the separation boundary to edit the input latent code and generating the output image takes 0.42 seconds. Generating the neck mask takes 0.75 seconds. For the $Diffusion$ process, it takes 23.55 seconds per image for 100 iterations. The image warping takes 1.45 seconds. In the testing stage, regardless of the projection operation, the core steps of our method take less than 3 seconds. All the other steps take negligible time. Our experiments are based on a desktop PC with i7-9700 3.0GHz CPU, 16 GB memory, and GeForce RTX 2060 GPU of 13.9 GB memory.

4.1.2 Chin Editing Dataset. As it is not feasible to collect real-world paired images with and without double chin, to facilitate future research on this topic, we also build up a dataset with paired images, where the double chin removal is based on our method. We first generate 13,990 portrait images, which have visible double chins. Then we use our proposed method to remove the double chin in those images and get a double chin dataset *DoubleChin*:

$$DoubleChin = \{[(x_d^i, w_d^{+i}), (x_o^i, w_o^{+i})] \mid_{i=0}^{n=13,990}\}, \tag{20}$$

where x_d^i and w_d^{+i} denote the original portrait image and its corresponding latent code, and x_o^i and w_o^{+i} denote the corresponding outputs.

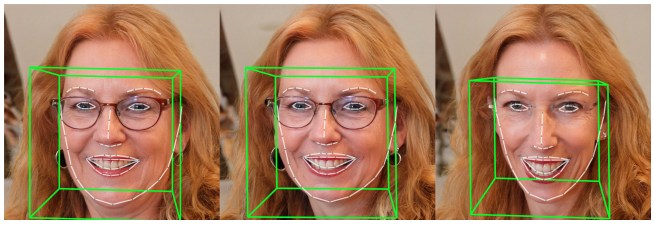

Fig. 12. Comparison between the landmarks and postures of the results edited by coarse separation boundary and fine separation boundary. The first image is the original image, the second image is the one edited by the fine separation boundary, and the third image is the one edited by the coarse separation boundary. We set the same hyper-parameter α in Eqn. 12 for the two separation boundaries.

4.2 Ablation Studies

4.2.1 Coarse Separation Boundary vs. Fine Separation Boundary. As described in Section 3.5, the fine separation boundary b_f does not only eliminate the double chin but also preserves facial identity, as shown in Fig. 12. We directly edit the latent codes of 50 randomly selected real images and 50 randomly sampled synthetic images using b_c and b_f respectively and set the hyper-parameter α as the same value, then generate images from the edited latent codes. Then we use 3DDFA [Guo et al. 2020; Zhu et al. 2019] to detect the landmarks and posture of those images generated from the edited latent codes (see Fig.12). Since no ground-truth portrait without a double chin is available, we conduct quantitative comparisons on the average Euclidean distance of landmarks and postures between the original portrait images and edited portrait images, as shown in Tab. 1. This shows that the results generated based on the fine separation boundary can well preserve facial pose and identity, allowing a simple warping approach to resolve the misalignment rather than the time-consuming diffusion method.

Table 1. Quantitative comparisons on the average Euclidean distance of landmarks and posture.

	real image		synthetic image	
	b_f	b_c	b_f	b_c
posture	0.0546	0.3153	0.0945	0.5005
landmarks	183.1285	680.8092	279.6109	978.7780

4.2.2 Face Landmarks Detection Method. In Section 3.3, we propose to use the landmarks of the chin to refine the prediction of *FaceParsing*. Preserving facial identity requires that landmarks of the chin align with the face edge. Face alignment predicts landmarks that roughly align with the face edge, while 3DDFA and PRNet often predict some erroneous points. As shown in Fig. 13, there is non-negligible inconsistency in the results of 3DDFA and PRNet (as highlighted by the green box), which deviates from the face shape, while face alignment provides the most stable results in our case.

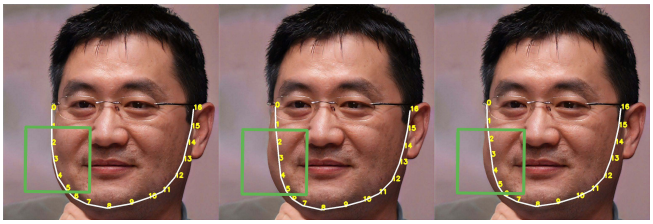

Fig. 13. The comparison between several face landmark detection methods. From left to right: the landmarks of the chin detected by face alignment, 3DDFA, and PRNet, respectively.

4.3 Comparisons

Although we make the first effort towards automatic double chin removal, we still compare our method with existing face editing methods that require adaptation, and even manual efforts to achieve similar effects, including MaskGAN [Lee et al. 2020], SC-FEGAN [Jo and Park 2019], and inpainting [Yu et al. 2018, 2019] (see Fig. 14). For MaskGAN, we use *FaceParsing* to extract the neck mask and face mask, then manually edit the output mask to reduce the redundant chin region. For SC-FEGAN, we use the official SC-FEGAN demo to add a mask on the double chin then manually draw chin sketches to generate a new chin. For inpainting, we leverage our neck mask generation method and the *ChinEditing* dataset to train an inpainting model by setting the images with double chin and the corresponding neck masks as the training dataset. We input the test image and its neck mask to the inpainting model and get the result.

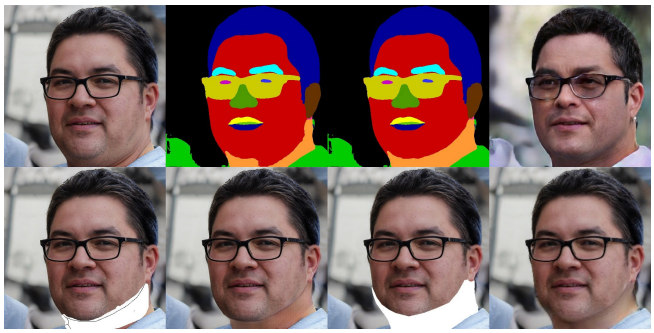

Fig. 14. Comparisons with different methods. Images in the first row (from left to right): the original image, the masks predicted by *FaceParsing*, the masks edited manually (face region is reduced), and the output of MaskGAN. Images in the second row (from left to right): the input of SC-FEGAN (masks and sketches are added manually), the output of SC-FEGAN, the input of inpainting (neck mask is predicted by our $NeckMaskNet$), and the output of inpainting.

Fig. 15 shows some typical results generated from different methods, including three synthetic images and three real images that have different postures. It can be seen that MaskGAN significantly changes facial identity during editing, SC-FEGAN generates non-smooth chin edges, and inpainting generates fuzzy results with obvious artifacts on the chin and neck. In contrast, our model generates more stable and plausible results.

5 USER STUDY

We further evaluate whether our results meet human expectations by conducting a user study. This consists of two participant-based experiments: (1) Given mixed untouched images and edited (double chin removed) images, we measure whether participants can differentiate these images in-between. (2) Verify user preferences on our results over results generated by other methods in Section 4.3.

For the first study, we used 10 resultant images with double chin removed and another 10 untouched images without a double chin, all the images are randomly sampled. We recruited 27 participants for this task. All 20 images were presented to each participant in a random order, and each participant was asked to score each image (0 for edited, and 1 for untouched). Results show that 50.4% of the untouched images were labeled as untouched, while 55.9% of the edited images were also labeled as untouched. The difference in the

Fig. 15. Comparisons to MaskGAN, SC-FEGAN, and generative-inpainting. In the first row (input images), the three portraits on the left are real images provided by FFHQ, Flickr user Ville Pohjanheimo, USAID Asia, and Jamwhy (CC BY-NC 2.0), and the three portraits on the right are synthetic images. From top to bottom, original portraits, MaskGAN, SC-FEGAN, generative-inpainting results, and ours, respectively.

scores between edited images and untouched images is negligible, demonstrating the quality of our result.

For the second study, we compared our method to MaskGAN [Lee et al. 2020], SC-FEGAN [Jo and Park 2019], and inpainting [Yu et al. 2018, 2019]. We presented 10 groups of images in a random order to 22 participants. Each image group consisted of 5 images, including the randomly sampled input image with double chin followed by the results generated by different methods in a random order. The participants were asked to select the best result in their opinion for eliminating the double chin. Statistics show that 66.36% of subjects chose our result, 7.27% of subjects chose MaskGAN, 12.27% of subjects chose SC-FEGAN, and 14.10% of subjects chose inpainting, indicating that our method outperformed others.

6 DISCUSSION

While we have demonstrated highly realistic chin editing results, our method has a few limitations (see Fig. 16). First, removing a double chin usually causes the reduction of the neck region along with vacancies in the background, which will be filled by the edited latent code. The more accurate the input latent code is, the more effectively the missing region is reconstructed. Note that portrait images in the wild need to be encoded first to get the input latent code, but the precision depends on the generality of the StyleGAN encoder, which might affect the resulting quality. Using inpainting to complete the vacancies of background could be a solution. The fine separation boundary also changes the shape of clothes near the neck, leading to misalignment along the edge of clothes (see the collar highlighted by the green box in Fig. 16). Second, as StyleGAN cannot generalize well to objects attached to the face, such as earrings and microphones, the

Fig. 16. Limitations. In the first row (from left to right): the first image displays inconsistency in the background, the second image has misalignment along the collar's edge, and part of the microphone disappears, the third image's chin is shortened. In the second row, our method cannot process the extreme case. The three images in the first row and the first image in the second row are provided by FFHQ and Flickr user Philip McMaster, Fotografia UFC, Rutgers Nursing, Ian (CC BY-NC 2.0).

neck mask sometimes covers those objects, causing visible artifacts in the results. In some extreme cases, the neck mask covers parts of the face region, leading to chin shortening. Using a more precise neck mask could eliminating such a limitation. Finally, we notice that our approach cannot remove the double chin of some faces with extreme postures. We attribute this to the training data. In our randomly sampled training data, most of the faces have postures close to neutral. Therefore our double chin separation boundary may not perform well for extreme cases. This limitation can be overcome by adding more faces with different poses in our dataset.

In the future, we would like to exploit our approach for the reverse chin editing process, i.e., adding the chin structure. In this case, the latent code of the input portrait can be updated along the opposite direction of n_f w.r.t. the separation boundary by setting the hyperparameter α as a positive value (in contrast to the negative value as in Fig. 17 for chin structure removal). Other image editing possibilities at the structural level are also worth exploration based on our approach, including but not limited to hair manipulation, face makeup design, object style editing, etc.

Further, although not being a technical limitation, the high-quality facial adjustment results provided by the present work may raise ethical concerns. Special considerations such as the watermarking technique may be taken into account to verify the authenticity of the image content. In our work, the real test images for facial structure editing are all from the FFHQ dataset with Creative Commons license - CC BY-NC 2.0., and with the owner of the image source being credited for each paper figure (except for synthetic portraits). The real images from CelebAMask-HQ are only used for training the chin separation boundary but not manipulated. Moreover, we would like to call for extra attention to the possible ethics concerns of facial structure editing. The usage of our method (so as many other methods) for manipulating a portrait image should have consent from the corresponding owner.

7 CONCLUSION

In this paper, we propose a novel face structure editing method and introduce the first automatic chin editing method for portrait images as a case study. The key idea is to train a fine separation boundary and generate an image containing a new chin that best matches the original portrait by editing the latent code of the original portrait. In the training stage, we propose coarse-to-fine separation boundary training. We first leverage a coarse double chin separation boundary in the latent space and edit the original latent code to eliminate the double chin. To only locally alter the neck region while keeping other regions unchanged, we employ a neck mask and use a diffusion approach to integrate the new chin into the original image, generating paired images with and without a double chin. The latent codes of the paired images are used to refine the double chin separation boundary, leading to plausible results on portrait images in the testing stage. We consider our work as an interesting step towards facial structure editing in the latent space and hope it can inspire more works in the future.

ACKNOWLEDGMENTS

Xiaogang Jin was supported by the National Key R&D Program of China (Grant No. 2017YFB1002600), the National Natural Science Foundation of China (Grant Nos. 61972344, 61732015). Yong-Liang Yang was supported by RCUK grant CAMERA (EP/M023281/1, EP/T014865/1), and a gift from Adobe. We thank the anonymous reviewers for their detailed comments, NVIDIA Research for making the Flickr-Faces-HQ (FFHQ) dataset, and all Flickr users for sharing their portrait photos under the Creative Commons License.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*. IEEE, 4431–4440.

Kaoru Arakawa and Kohei Nomoto. 2005. A system for beautifying face images using interactive evolutionary computing. In *2005 International Symposium on Intelligent Signal Processing and Communication Systems*. 9–12.

David Bau, Hendrik Strobelt, William S. Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. 2019. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.* 38, 4 (2019), 59:1–59:11.

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*. ACM, 187–194.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Adrian Bulat and Georgios Tzimiropoulos. 2017. How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230, 000 3D Facial Landmarks). In *IEEE International Conference on Computer Vision, ICCV 2017*. IEEE Computer Society, 1021–1030.

Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. 2018. PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. IEEE Computer Society, 40–48.

Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. 2020. Deep Generation of Face Images from Sketches. *CoRR* abs/2006.01047 (2020).

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. IEEE Computer Society, 8789–8797.

Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. 2018. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV (Lecture Notes in Computer Science, Vol. 11218)*. Springer, 557–574.

Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J. Black, and Timo Bolkart. 2020. GIF: Generative Interpretable Faces. *CoRR* abs/2009.00149 (2020).

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*. 2672–2680.

Jinjin Gu, Yujun Shen, and Bolei Zhou. 2020. Image Processing Using Multi-Code GAN Prior. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 3009–3018.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 5767–5777.

Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z. Li. 2020. Towards Fast, Accurate and Stable 3D Dense Face Alignment. In *Computer Vision - ECCV 2020 - 16th European Conference*. 152–168.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Youngjoo Jo and Jongyoul Park. 2019. SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*. IEEE, 1745–1753.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 8107–8116.

Scott Kelby. 2011. *Professional portrait retouching techniques for photographers using photoshop*. Pearson Education.

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Fader Networks: Manipulating Images by Sliding Attributes. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 5967–5976.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 5548–5557.

Chen Li, Kun Zhou, and Stephen Lin. 2015. Simulating makeup through physics-based manipulation of intrinsic image layers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*. IEEE Computer Society, 4621–4629.

Mu Li, Wangmeng Zuo, and David Zhang. 2016. Deep Identity-aware Transfer of Facial Attributes. *CoRR* abs/1610.05586 (2016).

Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. 2017. Generative Face Completion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. IEEE Computer Society, 5892–5900.

Lingyu Liang, Lianwen Jin, and Xuelong Li. 2014. Facial Skin Beautification Using Adaptive Region-Aware Masks. *IEEE Trans. Cybern.* 44, 12 (2014), 2600–2612.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least Squares Generative Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017*. IEEE Computer Society, 2813–2821.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Kristina Scherbaum, Tobias Ritschel, Matthias B. Hullin, Thorsten Thormählen, Volker Blanz, and Hans-Peter Seidel. 2011. Computer-Suggested Facial Makeup. *Comput. Graph. Forum* 30, 2 (2011), 485–492.

Wei Shen and Rujie Liu. 2017. Learning Residual Images for Face Attribute Manipulation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. IEEE Computer Society, 1225–1233.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 9240–9249.

Yi-Chang Shih, Wei-Sheng Lai, and Chia-Kai Liang. 2019. Distortion-free wide-angle portraits on camera phones. *ACM Trans. Graph.* 38, 4 (2019), 61:1–61:12.

Yi-Chang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. 2014. Style transfer for headshot portraits. *ACM Trans. Graph.* 33, 4 (2014), 148:1–148:14.

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi. 2019. Single image portrait relighting. *ACM Trans. Graph.* 38, 4 (2019), 79:1–79:12.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020. StyleRig: Rigging StyleGAN for 3D Control Over Portrait Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 6141–6150.

Qinjie Xiao, Xiangjun Tang, You Wu, Leyang Jin, Yong-Liang Yang, and Xiaogang Jin. 2020. Deep Shapely Portraits. In *MM '20: The 28th ACM International Conference on Multimedia*. ACM, 1800–1808.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. IEEE Computer Society, 5987–5995.

Raymond A. Yeh, Ziwei Liu, Dan B. Goldman, and Aseem Agarwala. 2016. Semantic Facial Expression Editing using Autoencoded Flow. *CoRR* abs/1611.09961 (2016).

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. 2018. Generative Image Inpainting With Contextual Attention. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*. IEEE Computer Society, 5505–5514.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. 2019. Free-Form Image Inpainting With Gated Convolution. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*. 4470–4479.

Xuaner Cecilia Zhang, Jonathan T. Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E. Jacobs. 2020. Portrait shadow manipulation. *ACM Trans. Graph.* 39, 4 (2020), 78.

Haiming Zhao, Xiaogang Jin, Xiaojian Huang, Menglei Chai, and Kun Zhou. 2018. Parametric Reshaping of Portrait Images for Weight-change. *IEEE Computer Graphics and Applications* 38, 1 (2018), 77–90.

Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David Jacobs. 2019. Deep Single-Image Portrait Relighting. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*. IEEE, 7193–7201.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020b. In-Domain GAN Inversion for Real Image Editing. In *Computer Vision - ECCV 2020 - 16th European Conference (Lecture Notes in Computer Science, Vol. 12362)*. Springer, 592–608.

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020a. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 5103–5112.

Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. 2019. Face Alignment in Full Pose Range: A 3D Total Solution. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 1 (2019), 78–92.

Fig. 17. Diverse results generated by our method. The first, the second, and the fifth rows are real images provided by FFHQ and Flickr user Alvin Smith, minjungkim, Lori Thantos, Marcel B. and Jane Cantral (CC BY-NC 2.0). Other rows are synthetic images. For the first four rows, images are edited by b_f with decreasing hyperparameter α (overlayed in the bottom right corner).