DeepFaceEditing: Deep Face Generation and Editing with Disentangled Geometry and Appearance Control

SHU-YU CHEN*, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences, China
FENG-LIN LIU*, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences, China
YU-KUN LAI, School of Computer Science and Informatics, Cardiff University, UK
PAUL L. ROSIN, School of Computer Science and Informatics, Cardiff University, UK
CHUNPENG LI, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences , China
HONGBO FU, School of Creative Media, City University of Hong Kong, HK
LIN GAO†, Institute of Computing Technology, CAS and University of Chinese Academy of Sciences , China
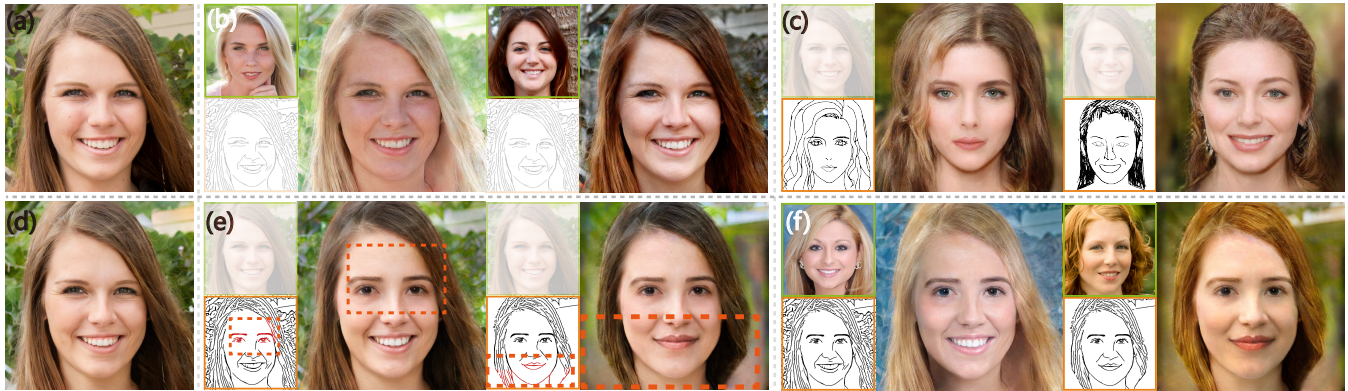
Fig. 1. Our DeepFaceEditing method allows users to intuitively edit a face image to manipulate its geometry and appearance with detailed control. Given a portrait image (a), our method disentangles its geometry and appearance, and the resulting representations can faithfully reconstruct the input image (d). We show a range of flexible face editing tasks that can be achieved with our unified framework: (b) changing the appearance according to the given reference images while retaining the geometry, (c) replacing the geometry of the face with a sketch while keeping the appearance, (e) editing the geometry using sketches, and (f) editing both the geometry and appearance. The inputs used to control the appearance and geometry are shown as small images with green and orange borders, respectively. Original images courtesy of mcsfphotos, Wonder Woman, UGA CAES/Extension, tvnewsbadge, and KcMacPhoto.

Recent facial image synthesis methods have been mainly based on conditional generative models. Sketch-based conditions can effectively describe the geometry of faces, including the contours of facial components, hair structures, as well as salient edges (e.g., wrinkles) on face surfaces but lack

effective control of appearance, which is influenced by color, material, lighting condition, etc. To have more control of generated results, one possible approach is to apply existing disentangling works to disentangle face images into geometry and appearance representations. However, existing disentangling methods are not optimized for human face editing, and cannot achieve fine control of facial details such as wrinkles. To address this issue, we propose DeepFaceEditing, a structured disentanglement framework specifically designed for face images to support face generation and editing with disentangled control of geometry and appearance. We adopt a local-to-global approach to incorporate the face domain knowledge: local component images are decomposed into geometry and appearance representations, which are fused consistently using a global fusion module to improve generation quality. We exploit sketches to assist in extracting a better geometry representation, which also supports intuitive geometry editing via sketching. The resulting method can either extract the geometry and appearance representations from face images, or directly extract the geometry representation from face sketches. Such representations allow users to easily edit and synthesize face images, with decoupled control of their geometry and appearance. Both qualitative and quantitative evaluations show the superior detail and appearance control abilities of our method compared to state-of-the-art methods.

*Authors contributed equally.
†Corresponding author.

Authors' addresses: Shu-Yu Chen, Fenglin Liu, Chunpeng Li, and Lin Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences. Yu-Kun Lai and Paul L. Rosin are with the School of Computer Science and Informatics, Cardiff University. Hongbo Fu is with the School of Creative Media, City University of Hong Kong. Authors' e-mails: chenshuyu@ict.ac.cn, fenglinliu@std.uestc.edu.cn, LaiY4@cardiff.ac.uk, RosinPL@cardiff.ac.uk, cpli@ict.ac.cn, hongbofu@cityu.edu.hk, gaolin@ict.ac.cn.

1 INTRODUCTION

Generating realistic face images is an active research topic in image synthesis. Recent techniques such as StyleGAN [Karras et al. 2019, 2020] allow random generation of realistic face images. However, many of them lack direct control of facial geometry or appearance. Adding conditions to the generation process is necessary to generate specific face images of interest, since face images are results of multiple factors, including geometry, appearance, head pose, viewpoint, etc.

Sketch-based conditions have been commonly used in existing image-to-image translation techniques [Isola et al. 2017; Wang et al. 2018], since sketching provides an easy and expressive means to depict desired geometry, including shape contours and salient surface edges. Multiple sketch-to-image techniques [Chen et al. 2020; Chen and Hays 2018; Li et al. 2020; Sangkloy et al. 2017] have been proposed to generate realistic face images from edge maps or even rough sketches. However, since it is difficult to infer the appearance information from sketches alone, the above techniques lack enough control of appearance during the generation process. On the other hand, there exist various techniques [Kim et al. 2020; Kolkin et al. 2019; Yoo et al. 2019] which allow the generation of realistic faces with their appearance as similar to given reference images as possible. However, such techniques focus on global appearance transfer and often do not well preserve geometric details of source images. Recent image disentangling works [Kim et al. 2020; Park et al. 2020] offer promising frameworks to have decoupled control of multiple attributes such as geometry and appearance. However, disentangling techniques for processing general image types do not provide optimal results for face generation. These methods treat images holistically and do not provide detailed, localized control, which is essential for face image editing.

In this work, we present DeepFaceEditing, a novel face generation and editing framework that allows disentangled control of geometry and appearance (Fig. 1). The key enabler is a structured disentanglement framework specifically designed for face images. Observing that faces have a fixed structure of facial components, we adopt a local-to-global framework, consisting of two modules: *Local Disentanglement* (*LD*) and *Global Fusion* (*GF*). With paired images and sketches, the *LD* module disentangles facial components into two independent representations: geometry and appearance. The former encodes the geometry information including shapes of face and facial components as well as salient geometric details such as wrinkles. This can be specified explicitly as a sketch or implicitly as a geometry reference image. In contrast, the appearance representation encodes the information related to color, material, lighting

conditions, etc. The *GF* module is trained to fuse local representations to obtain globally consistent face images. Unlike many other forms of disentanglement, our disentanglement makes the resulting representations easy to manipulate, and in particular, the geometry is represented using sketches, which can be edited intuitively for both overall shapes and details.

To ensure reliable disentanglement, our geometry space is designed to be a shared space of both sketches and images. In this way, we can not only make use of the geometry information in a geometry reference image but also develop a sketch-based interface for intuitive geometry manipulation. Figure 1 shows representative editing examples achieved with our interface, which supports decoupled control of appearance and geometry. We will also show the applications of our technique to face generation and face image style transfer.

We perform extensive qualitative and quantitative experiments, which show that disentangled control of geometry and appearance allows flexible face image generation and editing, and our approach outperforms state-of-the-art methods for various applications.

The main contributions of our work are summarized as follows:

- We present a structured disentanglement framework for face image representation and synthesis, which ensures geometry and appearance are well decoupled and can be manipulated separately.
- To achieve this, our network architecture involves local disentanglement (LD) modules for individual facial components to make learning more efficient, and a global fusion (GF) module for effective fusion of feature maps generated by LD modules. The LD modules are designed to embed both images and sketches of local regions to a shared space to ensure appropriate disentanglement, and allow sketches to be used as an explicit representation for the geometry, which is essential for detailed editing.
- We present a novel interactive system that supports real-time editing of portrait images, which allows detailed editing of the geometry through an intuitive sketch-based interface, as well as modifying the appearance by providing an appearance reference image.

2 RELATED WORK

Our work is closely related to several topics, including neural face image synthesis, neural face image editing, neural image disentanglement, and style transfer.

2.1 Neural Face Image Synthesis

In recent years, conditional generative models like Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] have been widely used in face generation. They are capable of generating realistic face images from a random sampling of the Gaussian distribution [Karras et al. 2019]. These methods are not specifically designed to control the detail or appearance of faces explicitly. Several solutions [Li et al. 2016; Richardson et al. 2020] have been proposed to control the image synthesis with specific conditions by using conditional GANs [Mirza and Osindero 2014]. Also inspired by conditional GANs [Mirza and Osindero 2014], Lee et al. [2020b]

and Gu et al. [2019] take a semantic mask as input and synthesize a new face based on a reference portrait. Instead of directly feeding a semantic layout as input, Park et al. [2019] propose spatially-adaptive normalization to progressively inject semantic information and achieve better visual quality for scene images.

Such methods support controllable image editing by editing semantic masks, but lack precise control of details within label maps. Another user-friendly interactive mode of geometry control is sketching. Compared with label maps, sketches can be easily drawn by users and artists, and thus have often been adopted for depicting the shape of desired content [Chen et al. 2009]. Based on Pix2Pix [2017] for general image-to-image translation, SketchyGAN [Chen and Hays 2018] and LinesToFacePhoto [Li et al. 2019] synthesize realistic images by converting sketches to distance maps. On the basis of pix2pixHD [2018], an extension of Pix2Pix for generating high-resolution images, Li et al. [2020] further introduce DeepFacePencil, a novel sketch-based face image synthesis framework that is robust to hand-drawn sketches. To produce high-quality faces from rough or incomplete sketches, Chen et al. [2020] present DeepFaceDrawing, which takes a local-to-global approach and leverages manifold projection to enhance the generation quality and robustness from freehand sketches. Besides 2D image generation, Han et al. [2017] develop a sketching system for 3D face and caricature modeling. The above works use sketches to depict target geometry, and have little or no control of appearance during editing. However, the appearance plays a central role for realistic image generation. Sangkloy et al. [2017] propose to use additional color strokes to indicate preferred colors for objects. While they achieve impressive results, the synthesized images still exhibit various artifacts like color leaking. For generating high-quality face images, it is essential to represent the complex appearance, which is difficult to achieve by only using color strokes. It also needs extra effort to paint the color strokes in addition to the geometry sketches, and the consistency between different color strokes should also be maintained for visual realism. It is thus not straightforward to extend the method by Sangkloy et al. [2017] to generate realistic face images from freehand drawings and through interactive editing.

Sketches can be used to depict not only shape contours (similar to label maps) but also fine-level details like wrinkles on faces. However, as discussed previously, sketches themselves cannot support appearance control. To address this issue, Zhang et al. [2020] and Lee et al. [2020a] propose exemplar-based image translation methods to generate photo-realistic images by learning dense cross-domain correspondence. These methods could be applied to different datasets including face edge maps but the face edge maps used in their work are not freehand sketches. Liu et al. [2020] further present an exemplar-based image synthesis method with freehand sketches. The line-sketch generator is adopted to produce multiple sketches to improve the model's robustness and thus handle freehand sketches. However, these methods are not specifically designed for photo-realistic face image synthesis considering the face structures. Robust high-quality face synthesis methods with detailed control using freehand sketches and appearance references are still missing.

2.2 Neural Face Editing

Face image editing has been a long-standing topic with the ability of providing flexible human-computer interaction, whose results have been significantly improved since the emergence of deep neural networks. Here, we mainly review the neural-network-based approaches. These approaches can be further grouped into three categories according to the intermediary they provide to the user for editing, namely direct editing, semantic-mask-guided editing and sketch-based editing. For direct editing, Brock et al. [2016] propose an introspective adversarial network (IAN), which turns rough painting strokes into user-desired photo-realistic images by directly manipulating images. For semantic-mask-guided editing, pioneering works such as [Gu et al. 2019; Lee et al. 2020b; Zhu et al. 2020] incorporate semantic masks with conditional GANs [Mirza and Osindero 2014] for interactive editing of face images, which have been discussed in previous sub-sections. For the sketch-based methods, FaceShop [Portenier et al. 2018] introduces a face image editing system to synthesize realistic images according to user inputs: masks, simple sketches, and colored strokes. DeepFillv2 [Yu et al. 2019b] introduces gated convolution to learn dynamic features and presents a patch-based GAN discriminator to generate high quality inpaintings in a free-form manner. It is also applied to face editing under the guidance of input sketches. Based on U-Net [Ronneberger et al. 2017], Jo et al. [2019] present an extra style loss to generate more robust and higher quality results, requiring minimal effort from users. While these methods can generate high quality edited faces with guiding colored strokes and simple sketches within local areas, our method further supports editing of faces in both the local and global manners. In addition, since these methods treat face editing as an image completion problem with sketch guidance, they can only generate images based on the appearance of input images (i.e. not support using appearance from other images). Meanwhile, as demonstrated in [Jo and Park 2019], non-photorealistic artifacts could be generated with purely sketch and color strokes. Recently, Yang et al. [2020] present another face editing method based only on sketches. With a sketch refinement strategy, their method is robust for human-drawn sketches but lacks the control of face appearance.

2.3 Neural Image Disentanglement

Learning a disentangled representation aims at modeling the factorization of data variation. Previous works have introduced the disentanglement into image-to-image translation to facilitate multi-domain translation or allow the manipulation of certain image attributes while retaining others. For example, Liu et al.[2018] disentangle images into domain-invariant representations to generate realistic results across multiple domains by changing the domain labels. The work of [Gonzalez-Garcia et al. 2018] decouples images into a shared part and an exclusive part to achieve multi-modal image translation. Furthermore, several works (e.g., [Lee et al. 2020c; Yu et al. 2019a]) have extended the disentangled representation to provide domain-invariant and domain-specific representations to perform multi-domain and multi-modal translations simultaneously. However, these methods have focused on multi-domain translation and their disentanglement mainly aims at holistic attributes whereas

our method focuses on disentangled representations that respect facial component structure and support detailed editing.

Another group of methods apply disentangled representations of latent code to control predefined attributes of faces. For instance, the method by Zhang et al. [2019] is trained on labeled input data pairs by swapping designated parts of embeddings to control specific attributes. Deng et al. [2020] imitate the 3D rendering process and introduce contrastive learning to learn a disentangled latent space. Many other works (e.g., [Abdal et al. 2020; Härkönen et al. 2020; Shen et al. 2020; Tewari et al. 2020]) have tried to analyze and disentangle the latent code of some pretrained GAN space [Karras et al. 2019] also with labeled data of specific attributes. Although these works successfully disentangle the latent space, they could only control a limited number of predefined attributes such as gender, expression, and age, due to the use of labeled data in the training stage. In contrast, our method provides users with more freedom to edit faces by changing sketches and to swap the face appearance by changing their appearance code.

Several research works have also proposed to disentangle specific elements and manipulate images. Nguyen-Phuoc et al. [2019] and Schwarz et al. [2020] utilize 3D representations and disentangle images into pose, shape, and appearance. However, these methods can only generate images randomly and do not support editing with detailed control, except for changing image viewpoints. Wang et al. [2019] explicitly disentangle face deformations and appearance with two parallel networks to achieve expression editing. While their work is limited to only changing facial expressions, the method in [Park et al. 2020] also encodes images into geometry and appearance components with the idea of swapping and co-occurrent patch discriminating statistics. Compared with this work, our method leverages a sketch as a constraint instead of the patch discriminator for precise facial detail editing. MichiGAN [Tan et al. 2020] is specifically designed for photo-realistic hair image generation of portraits conditioned on decoupled attributes. In our work, we aim to disentangle geometry and appearance of face images for intuitive and detailed control of face generation. This benefits further applications such as freehand face editing and sketching.

2.4 Style Transfer

Style transfer focuses on changing the style of an image while preserving its key content. One category of style transfer methods (e.g., [Huang et al. 2018; Liu et al. 2019; Zhu et al. 2017b,a]) requires a collection of target images and treats this problem as image-to-image translation. Another category of methods achieves style transfer with an arbitrary style image. Starting from the pioneering work [Gatys et al. 2016], the follow-up works (e.g., [Kolkin et al. 2019; Li and Wand 2016; Mechrez et al. 2018]) take various content and style representations to generate attractive stylized images by iterative optimization. DST [Kim et al. 2020] further develops a novel geometry-aware style transfer method by deforming a content image to match the geometry of a style image. Another path of improvement is to eliminate the time-consuming optimization process. These works (e.g., [Huang and Belongie 2017; Li et al. 2017; Liao et al. 2017]) find dense correspondence or manipulate features in pre-trained networks to synthesize high-quality stylized images

in real time. Although the above works generate interesting artistic results, these are not particularly suitable for photo-realistic image style transfer. To handle this problem, WCT^2 [Yoo et al. 2019] employs wavelet operations and progressive stylization to synthesize high resolution stylized photo-realistic images within a few seconds. In our work, we disentangle the geometry and appearance of face images to represent content and style information, respectively. Our method not only supports photo-realistic synthesis of stylized faces but also allows detailed editing with sketches.

3 METHODOLOGY

In this section, we formalize the structure of our proposed face image generation architecture in detail. Inspired by Huang et al. [2017] and DeepFaceDrawing [Chen et al. 2020], we leverage a local-to-global framework to generate high-quality face images. Specifically, we decompose a face image into five components ("left-eye", "right-eye", "nose", "mouth", and "background") and process them using individual network modules. After component-level generation, we fuse the image patches into globally consistent results. Therefore, our architecture, called DeepFaceEditing, comprises 5 *Local Disentanglement* (Sec. 3.1) modules responsible for disentangling the geometry and appearance for each component, and a *Global Fusion* (Sec. 3.2) module responsible for fusing component features and generating high-quality results with global consistency. During training, we adopt a swapping scheme [Park et al. 2020] with a cycle-consistency constraint to enhance the robustness and generalization ability of our framework.

3.1 Local Disentanglement

By incorporating this module, we aim to extract both the geometry and appearance features for each face component and generate local image patches from these features. To achieve this, we design a Geometry Encoder and an Appearance Encoder for obtaining geometry features and appearance features, respectively. Furthermore, we design an Image Synthesis Generator to combine the geometry features and appearance features from a pair of image and sketch or two images (one providing the geometry features while the other providing appearance features) to obtain the translated component-level image patches.

Geometry Encoder. Sketches depict the contours of real images, and are inherently suitable for geometry information extraction. Thus, for sketch inputs, we can directly extract pure geometry information using an auto-encoder network. The main challenge here is to extract geometry features from real images. Given a real facial component image as input, an intuitive approach extracting its geometry features is to first use a pre-trained image-to-sketch translation network to translate the real image into the sketch domain and then send the generated sketch into the geometry encoder for sketches. While this approach is promising, we show that there exists a unified way to extract the geometry information from sketches and real images.

Specifically, we achieve this by training two auto-encoders, one for sketches and the other for images, and aligning the latent distribution of the image space to that of the sketch space, ensuring that only the geometry information is encoded. We first train a network
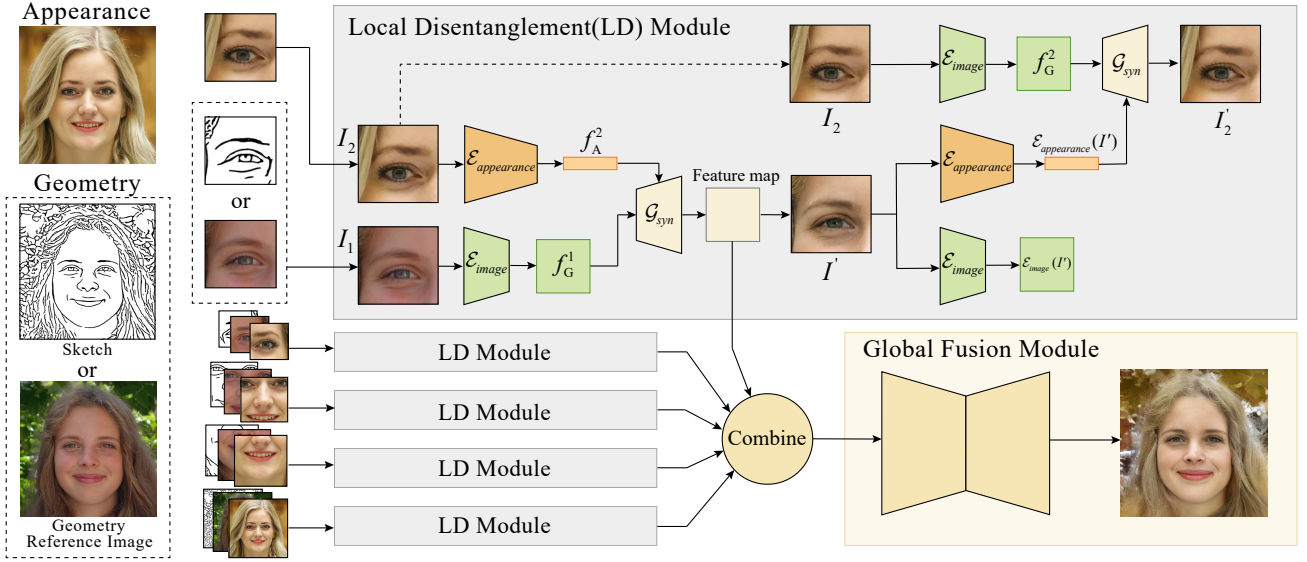
Fig. 2. An overview of our framework. Our pipeline leverages a local-to-global strategy. The upper-right shows an illustration of a Local Disentanglement (LD) module (for "right eye"), which uses $\mathcal{E}_{image}$ (when using a geometry reference image) or $\mathcal{E}_{sketch}$ (when using a sketch for controlling the geometry) to extract the geometry features and $\mathcal{E}_{appearance}$ to extract the appearance features. We use a swapping strategy to generate a new image I' with the geometry of I_1 and appearance of I_2. Taking the generated image I' as the appearance reference, along with the geometry of I_2, should lead to the reconstructed image I_2' that ideally matches I_2, allowing us to utilize a cycle loss to regularize training. Note that in our LD modules, such swapping and cycle constraints are applied to local regions corresponding to "left eye", "right eye", "nose", "mouth", and "background". The required geometry information can be explicitly given by a sketch or implicitly given by a reference image. Original images courtesy of Mats Adamczak and Senterpartiet (Sp).

consisting of an encoder $\mathcal{E}_{sketch}$ and a decoder $\mathcal{D}_{sketch}$ to generate an intermediate feature of sketches, as shown Fig. 3 (top). To retain essential spatial information, the latent space in the bottleneck layer is not in the form of a vector but instead a low-resolution feature map of dimension $h_G \times w_G \times c_G$, where h_G, w_G and c_G are the height, width and channel number for the geometry latent feature map. Both the input and output of this network are sketches. Note that the sketches here can be either edge maps extracted from images or hand-drawn sketches. For hand-drawn sketches, especially incomplete ones during the sketching process, we apply sketch manifold projection [Chen et al. 2020] during preprocessing to improve robustness.

Let I_{in} denote a real image and S_{in} denote its corresponding sketch. We extract the geometry feature of S_{in} through the pretrained $\mathcal{E}_{sketch}$, formulated as $f_G = \mathcal{E}_{sketch}(S_{in})$. Then we train an encoder $\mathcal{E}_{image}$ to map the corresponding image I_{in} into the latent geometry space of sketches, denoted as $f_G' = \mathcal{E}_{image}(I_{in})$. To encourage f_G and f_G' to follow the same distribution, we impose constraints on each layer of $\mathcal{D}_{sketch}$ when we feed f_G and f_G' into the pre-trained decoder $\mathcal{D}_{sketch}$. We also add an L1 loss between the outputs $\mathcal{D}_{sketch}(f_G)$ and $\mathcal{D}_{sketch}(f_G')$.

Appearance Encoder. Appearance is another important aspect of a facial image. The mapping between facial geometry and real face images is clearly one-to-many, but by specifying the appearance, such ambiguities can be resolved. We employ another encoder $\mathcal{E}_{appearance}$ to extract the appearance feature. The appearance encoder leverages global average pooling (i.e., for each feature channel,
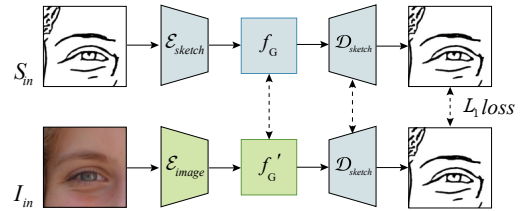

Fig. 3. Illustration of Geometry Encoder. With pre-trained $\mathcal{E}_{sketch}$ and $\mathcal{D}_{sketch}$, which take a sketch S_{in} as input, we train Geometry Encoder $\mathcal{E}_{image}$, which takes a real image I_{in} as input and generates a geometry feature map f_G' that matches f_G. During training, we fix the parameters of $\mathcal{E}_{sketch}$ and $\mathcal{D}_{sketch}$, and add L1 losses to enforce consistency between the geometry latent feature maps f_G and f_G', outputs of layers of $\mathcal{D}_{sketch}$ and output sketches of $\mathcal{D}_{sketch}$, as indicated by the dashed arrows. Original images courtesy of Mats Adamczak.

taking the average over all the spatial locations in the feature map), to eliminate spatial information and extract an appearance feature which is independent from its geometry feature. As appearance features are extracted for individual local regions, dropping spatial information does not cause significant loss of useful information. Results of the experiments on disentangled interpolation on face appearance and geometry (see Fig. 16) demonstrate that $\mathcal{E}_{appearance}$ can learn a continuous and smooth facial appearance space efficiently.
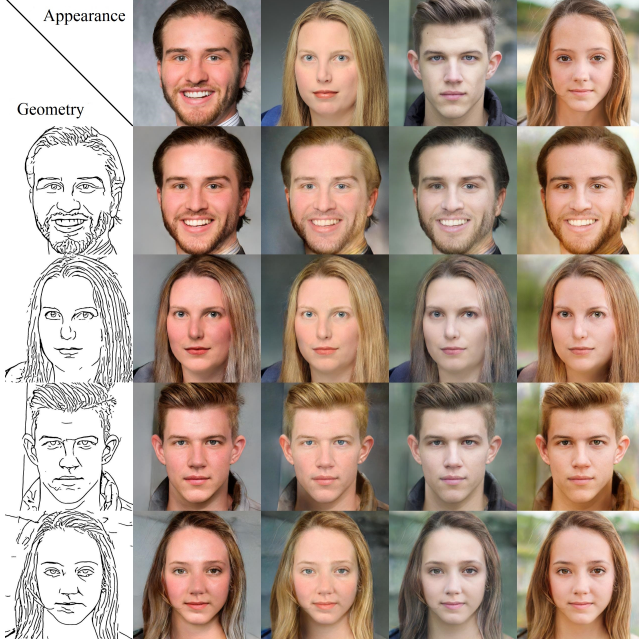
Fig. 4. Face generation with different geometry and appearance inputs. The geometry inputs (first column) use edge maps from the same set of images as those used for appearance (first row). Each face image is generated with the sketch in the same row for geometry and appearance reference image in the same column for appearance. Therefore, the images on the main diagonal look the same as the corresponding appearance images. This indicates the accurate reconstruction ability of our method. Original images courtesy of Dan Milner, Greg Mooney, BLSZ_BD and christopher simon.

Image Synthesis Generator. Given independent geometry and appearance representations, an image synthesis generator takes them as inputs and generates reconstruction or swapping results. To control the appearance of generated face images, we adopt Adaptive Instance Normalization (AdaIN) [Huang and Belongie 2017] in our facial image synthesis generator. Specifically, this generator comprises 4 residual blocks and 4 up-sampling layers, and the appearance features are injected into them. Then, we obtain an embedding feature which has the same resolution as the input image but with 64 channels. Finally, an image consistent with the input geometry and appearance features is predicted via a single convolution layer.

Unlike those sketch-to-image tasks, our framework takes real facial images as inputs and disentangles appearance features with geometry features under the assistance of sketches. Furthermore, a network that could provide controllable sketch-based face editing would be more appealing to users. Therefore, to achieve this, we randomly feed both the features generated by $\mathcal{E}_{sketch}$ and $\mathcal{E}_{image}$ into the image synthesis generator during the training stage. During inference, either geometry reference images or sketches can provide the geometry information and generate photo-realistic facial images with specific appearance. With appearance and geometry features disentangled, we can generate face images possessing the geometry and appearance from different sources, as shown in Fig. 4.

3.2 Global Fusion

This module generates the final face images from local image features encoded by the *Local Disentanglement* modules. To translate local image features into a complete and natural-looking face image, one possible way is to directly combine the local image patches generated by applying one or more convolution layers to the component-level feature maps (like the process of generating I' in Fig. 2). However, this straightforward approach is prone to exhibiting artifacts on the boundaries of different components. So instead of directly combining component-level image patches, our method combines the generated intermediate feature maps of the *Local Disentanglement* modules before sending them into an image generation network. In this way, our network aggregates more information flow and is capable of generating high-quality images.

More concretely, our *Global Fusion* module comprises three units: an encoder, residual blocks and a decoder, like DeepFaceDrawing [2020] and pix2pixHD [Wang et al. 2018]. Given the feature map of the background component, we replace certain patches of it with the corresponding generated features of other components, in the order of "mouth", "nose", "left-eye", and "right-eye" to reduce the impact of overlapping between components. Then we feed the combined feature map into the *Global Fusion* module to generate a new face with desired appearance and geometry.

3.3 Training Process

In this part, we introduce the training process of the aforementioned modules at length. We train the entire framework in a step-by-step manner. Specifically, we first train the *Local Disentanglement* modules, and then train the *Global Fusion* module with parameters of the *Local Disentanglement* modules fixed.

Training Dataset. To train our network, we need a large-scale dataset of sketch-image pairs. At the same time, the sketches in the sketch-image pairs are required to be salient, natural and similar to hand-drawn sketches. Traditional edge extraction methods, such as HED [Xie and Tu 2015] and Canny [Canny 1986], often fail to produce ideal edge maps. Therefore, we follow DeepFaceDrawing [Chen et al. 2020] and use the Photocopy filter in Photoshop followed by sketch simplification to build this dataset. We use FFHQ [Karras et al. 2019] as our training data. In this way, we generate 32.2k sketch-image pairs as our dataset and randomly select 29.9k pairs for training and 2300 pairs for testing. The resolution of both images and sketches is set to 512×512.

Disentanglement Training. The training process of each *Local Disentanglement* module consists of three steps. First, as described in Sec. 3.1, $\mathcal{E}_{sketch}$ and $\mathcal{D}_{sketch}$ are trained to learn a geometry latent space for sketches using L1 reconstruction loss. Once $\mathcal{E}_{sketch}$ is trained, we can represent the geometry feature as $f_G = \mathcal{E}_{sketch}(S_{in})$. Then, we train the network $\mathcal{E}_{image}$, which takes a real image I_{in} as input and predicts a geometry feature $f'_G = \mathcal{E}_{image}(I_{in})$ following the same distribution as the learned geometry space. The loss function L_{image} is defined as follows:

$$L_{image}(\mathcal{E}_{image}) = \sum_{i=0}^{N} ||\mathcal{D}_{sketch}^{(i)}(f'_G) - \mathcal{D}_{sketch}^{(i)}(f_G)||_1, \quad (1)$$

where $N = 7$ is the number of layers of decoder $\mathcal{D}_{sketch}$. The index 0 corresponds to the input feature map, the index N corresponds to the output image, and other indices are intermediate feature maps. Note that when optimizing the parameters of $\mathcal{E}_{image}$, we fix the weights of $\mathcal{E}_{sketch}$ and $\mathcal{D}_{sketch}$. Finally, we train the appearance encoder $\mathcal{E}_{appearance}$ and the image synthesis generator $\mathcal{G}_{syn}$ with weights of $\mathcal{E}_{sketch}$ and $\mathcal{E}_{image}$ fixed. During the last step of training, we randomly feed the geometry feature f_G of sketches or f_G' of real images into $\mathcal{G}_{syn}$. In the following parts, we denote both f_G and f_G' as G, without distinguishing their sources. We further introduce a swapping strategy and adopt cycle-consistency loss to disentangle a real facial image into appearance and geometry. To ensure photorealism of generated images, we also adopt the muti-scale discriminator [Wang et al. 2018] and adversarial loss.

More concretely, as shown in Fig. 2, given two component images I_1 (I_1 could be either a real image or sketch) and I_2 (I_2 should be a real image) in the training set, we can extract geometry features f_G^1 and f_G^2 from I_1 and I_2 by passing them through the pre-trained $\mathcal{E}_{sketch}$ or $\mathcal{E}_{image}$. The appearance feature f_A^2 from I_2 is extracted by using $\mathcal{E}_{appearance}$. By swapping the geometry feature of I_2 with that of I_1, we generate image I' using the geometry feature $f_G^1 = \mathcal{E}_{image}(I_1)$ of image I_1 and the appearance feature $f_A^2 = \mathcal{E}_{appearance}(I_2)$ of image I_2, as $I' = \mathcal{G}_{syn}(f_G^1, f_A^2)$. With the geometry of I_1 and the appearance of I', we can establish a cyclic reconstruction of image I_2, as $I_2' = \mathcal{G}_{syn}(f_G^2, \mathcal{E}_{appearance}(I'))$. We also introduce a self-reconstruction loss: when both $\mathcal{E}_{image}$ and $\mathcal{E}_{appearance}$ take the same image (such as I_1) as input, we can reconstruct it using its geometry and appearance features. The self-reconstruction of I_1 can be formulated as $I_1' = \mathcal{G}_{syn}(f_G^1, \mathcal{E}_{appearance}(I_1))$. Overall, we adopt the following loss terms to train our *Local Disentanglement* modules:

- Self-reconstruction loss: When the geometry and appearance come from the same image, i.e. $I_1 = I_2 = I$, the self-consistency of our framework requires that we can reconstruct I after passing it through our framework. The self-reconstruction loss contains three terms: 1) Perceptual loss [Johnson et al. 2016] L_{VGG}, which measures the visual similarity between the generated images and input images by a pre-trained VGG-19 model; 2) Feature matching loss [Wang et al. 2018] L_{FM} of discriminators, which aims to stabilize the training process; 3) Lab color loss L_{Lab} [Tan et al. 2020], which calculates the chromatic distance in the a and b channels for controlling the color tone by converting images to the CIE-LAB color space. The self-reconstruction loss can be formulated as follows:

$$L_{Recon} = \alpha_1 L_{Lab}(I_1, I_1') + \alpha_2 L_{FM}(I_1, I_1') + \alpha_3 L_{VGG}(I_1, I_1'), \quad (2)$$

In our experiments, we empirically set $\alpha_1, \alpha_2, \alpha_3 = 1, 10, 10$.

- Cyclic swapping loss: To disentangle the geometry and appearance features more thoroughly, we adopt swapping to generate face images from geometry and appearance features that are from different sources, i.e. $I_1 \neq I_2$. The cyclic swapping loss L_{Swap} contains two terms: L_{Geo} and L_{Cycle}. Next we introduce the L_{Geo} and L_{Cycle} in detail.

Our network aims at a complete disentanglement of geometry and appearance feature for a facial image. When generating I' after swapping the appearance of I_1 to that of I_2, the structure of I'

should be maintained. So we introduce geometry loss L_{Geo} to keep the geometry unchanged of the generated image by comparing it to input image:

$$L_{Geo} = ||\mathcal{E}_{image}(I_1) - \mathcal{E}_{image}(I')||_1, \quad (3)$$

To ensure the appearance of the swapped image I' is the same as that of I_2, we introduce a cycle consistency loss. With the geometry of I_2 and the appearance of the swapped image I', the generated image $I_2' = \mathcal{G}_{syn}(\mathcal{E}_{image}(I_2), \mathcal{E}_{appearance}(I'))$ should cyclically reconstruct image I_2. We use the above formulation of reconstruction loss L_{Recon} to achieve the cycle consistency constraint:

$$L_{Cycle} = \alpha_1 L_{Lab}(I_2, I_2') + \alpha_2 L_{FM}(I_2, I_2') + \alpha_3 L_{VGG}(I_2, I_2'), \quad (4)$$

where the hyper-parameters $\alpha_1, \alpha_2, \alpha_3$ are same as the above setting. Finally, L_{Swap} can be formulated as follows:

$$L_{Swap} = \tau_1 L_{Geo} + \tau_2 L_{Cycle}, \quad (5)$$

where we empirically set $\tau_1, \tau_2 = 1, 1$ in our experiments.

- Adversarial loss: Also, we adopt the muti-scale discriminator $\mathcal{D}$ to encourage the distribution of generated images to match the distribution of real images:

$$\begin{aligned} L_{GAN} = {}& \gamma_1 \mathbb{E}[\log \mathcal{D}(I_1)] + \gamma_2 \mathbb{E}[\log \mathcal{D}(I_2)] \\ & + \gamma_3 \mathbb{E}[1 - \log(\mathcal{D}(I_1'))] + \gamma_4 \mathbb{E}[1 - \log(\mathcal{D}(I_2'))] \\ & + \gamma_5 \mathbb{E}[1 - \log \mathcal{D}(I')], \end{aligned} \quad (6)$$

where we balance the weights of generator $\mathcal{G}_{syn}$ and discriminator $\mathcal{D}$ with $\gamma_1 = \gamma_2 = 0.5$, $\gamma_3 = \gamma_4 = \gamma_5 = 0.33$ in all our experiments.

Our final objective L_{LD} is simply the sum of the above three losses (as we have already weighted each term in these losses) and minimizing L_{LD} will lead to the optimization of three networks: $\mathcal{G}_{syn}$, $\mathcal{E}_{appearance}$, and $\mathcal{D}$. L_{LD} is formulated as follows:

$$L_{LD}(\mathcal{G}_{syn}, \mathcal{E}_{appearance}, \mathcal{D}) = L_{Recon} + L_{Swap} + L_{GAN}. \quad (7)$$

Global Fusion Training. With pre-trained *Local Disentanglement* modules, we are able to train the *Global Fusion* module which fuses the features encoded by *Local Disentanglement* modules together and generates the final results. Similar to the previous stage, we use the adversarial loss, feature matching loss, and perceptual loss for the *Global Fusion* module. Note that we do not need the swapping strategy in this stage since it does not involve any disentanglement.

4 EXPERIMENTS

In this section, we show our experimental setup as well as discuss the results of our experiments. We have done extensive experiments from three aspects, namely comparison with state-of-the-art methods, ablation study and user study. The results show the effectiveness of our proposed method and its superiority to the existing and alternative approaches.

First, we make extensive comparisons with the state-of-the-art methods. Since the source of geometry feature for image generation can be either real facial images or sketches, we compare our proposed method with two main branches of image-to-image translation methods: 1) style-transfer-based methods like WCT[2] [Yoo et al. 2019], STROTSS [Kolkin et al. 2019], DST [Kim et al. 2020]

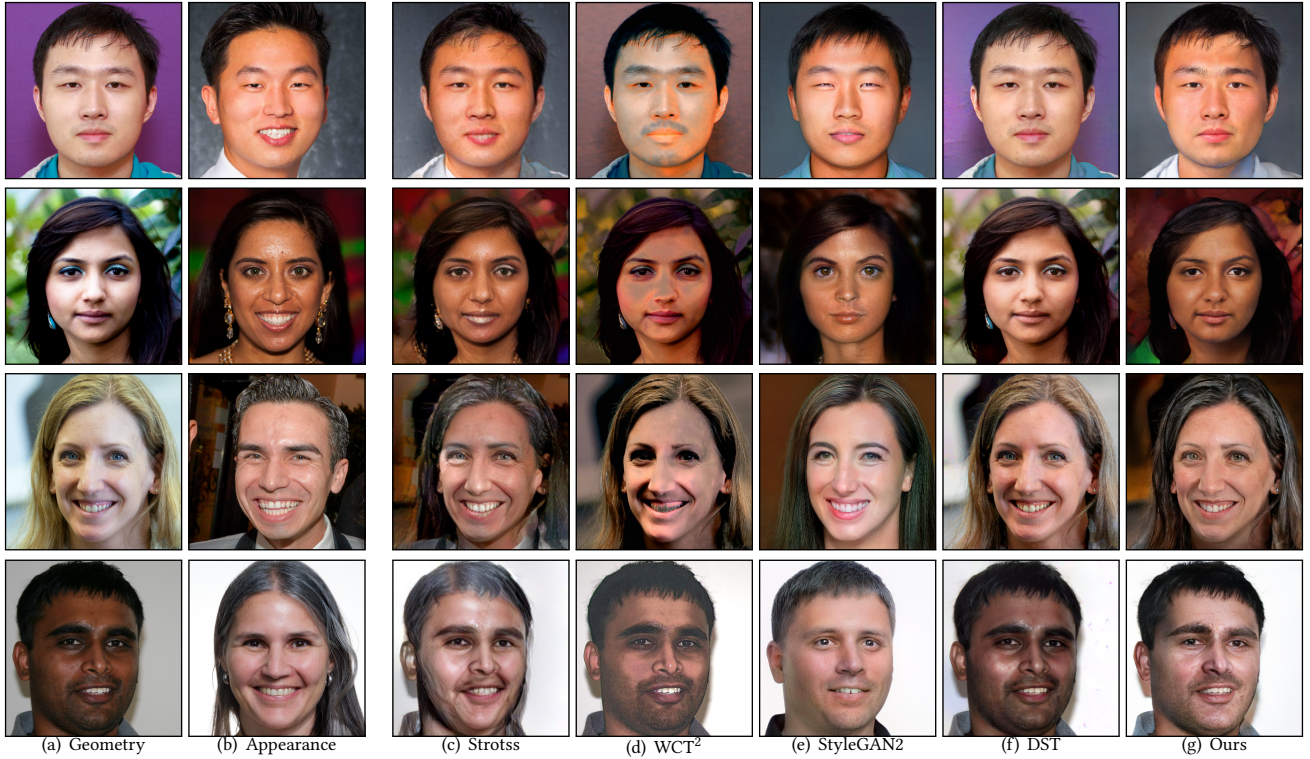| (a) Geometry | (b) Appearance | (c) Strotss | (d) WCT^2 | (e) StyleGAN2 | (f) DST | (g) Ours |

Fig. 5. Comparisons with state-of-the-art methods for face style transfer. In each row, (a) is a geometry reference image and (b) is an appearance reference image. (c)~(f) are the results by the existing methods and our results are shown in (g). WCT^2 could better preserve the geometry of (a), but cannot well reconstruct the appearance. As shown in the third example, DST cannot capture the appearance for the hair region. Our method achieves visually the best results. Original images courtesy of ImagineCup, Dan Milner, Lori Thantos, John Benson, Charles Mims, ShashiBellamkonda and Sebastiaan ter Burg.

and StyleGAN2 [Karras et al. 2020], where we take real images as inputs and evaluate the ability of face style transfer; 2) sketch-to-image translation methods like pSp [Richardson et al. 2020], Liu et al. [2020] and Zhang et al. [2020], where we take sketches as inputs and evaluate the ability of sketch-based face image editing. We also compare with some image editing methods based on sketch inputs. Then, we conduct ablation studies to justify the necessity of each component in our framework. Finally, a perception study is conducted to test whether the images generated by our method look more appealing to users compared with alternative methods, indicating the visual quality of our controllable face generation results. As shown in our demo video, our interactive editing interface achieves real-time performance. All the above experiments are carried out on a PC with an Intel i7-7700CPU, 32GB RAM, and two Nvidia RTX 2080Ti GPUs. And DeepFaceEditing is implemented in Pytorch [Paszke et al. 2019] and Jittor [Hu et al. 2020]. We use ADAM [Kingma and Ba 2015] with 0.0002 learning rate, $\beta_1 = 0.5$ and $\beta_2 = 0.99$. We use the maximum batch size 2 that fits in memory on our GPUs for 512×512 resolution. We will release the code and data for facilitating future research. Please find the details of the network architectures in the supplemental material.

4.1 Results and Evaluations

Face Style Transfer. In our framework, we can extract geometry features from either images or sketches. Hence, we can disentangle the geometry and appearance of an image, and generate new images by swapping geometry and/or appearance. This allows us to compare our method with state-of-the-art style transfer methods, including WCT^2 [Yoo et al. 2019], STROTSS [Kolkin et al. 2019], DST [Kim et al. 2020], and StyleGAN2 [Karras et al. 2020]. We use the official implementations for all the above methods. By default, 512×512 resolution is adopted, except that StyleGAN2 uses input and output images of 1024×1024. The comparisons between all the methods are shown in Fig. 5. Our method is capable of generating high-quality photo-realistic results and combining the geometry and appearance of different images without obvious artifacts, and outperforms state-of-the-art style-transfer methods. STROTSS [Kolkin et al. 2019] overall can capture the style of appearance images, but their method does not produce spatially consistent transfer results, leading to salient artifacts. DST [Kim et al. 2020] generates high-quality results, but their results fail to capture the appearance of the reference images, possibly due to their strong geometry constraint. For StyleGAN2 [Karras et al. 2020], the identities of the generated faces are changed, since their projected latent code could not reconstruct the images accurately and the mixing operation can

(a) Geometry

(b) Appearance

(c) pSp.
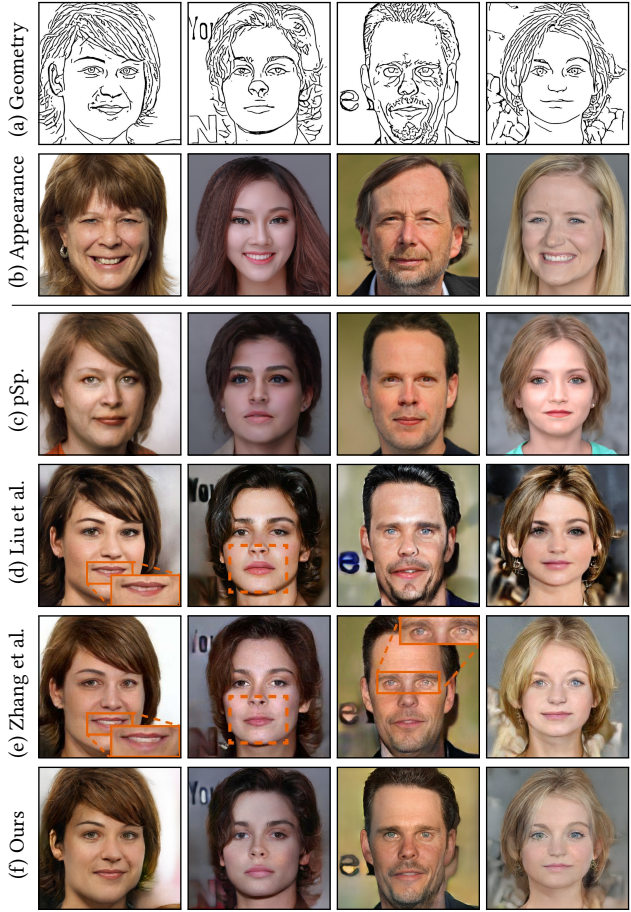
(d) Liu et al.

(e) Zhang et al.

(f) Ours

Fig. 6. Comparisons with sketch-to-image methods with the same input sketch (a) and appearance reference (b). The results of pSp (c) often fail to respect the input sketches, in particular the synthesized hair deviates significantly from what the sketches depict. The results of Liu et al. [2020] (d) and Zhang et al. [2020] (e) could well preserve the geometry indicated in the sketches, but the former suffers from the changes of appearance, while the latter exhibits some blurry artifacts in local regions. Our method is able to generate results with detailed local regions and respect the geometry and appearance references well. Original images courtesy of Aquatic Sciences Center, Việt Anh Trương, Peter Komposch and WVU LAW.

Sketch-based Face Image Synthesis. As our method can directly extract the geometry feature from a sketch and combine it with the appearance feature extracted from a reference image, it can be used for sketch-to-image translation. As shown in Fig. 6, we
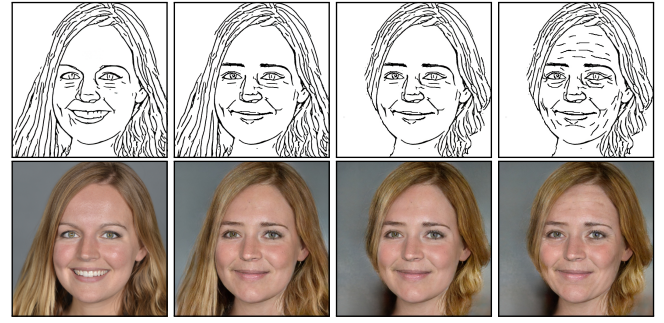

Fig. 7. Our model supports flexible face editing based on sketches, without requiring users to pre-define a region of interest. Thus our method provides users with more space of freedom and creativity. This figure shows a sequence of edited sketches and their corresponding generated face images. The first column shows the original image and its corresponding geometry. From left to right, we gradually add wrinkles on the sketch (through changing the geometry) while maintaining the appearance feature, achieving the effect of gradual aging. Original images courtesy of WVU LAW.

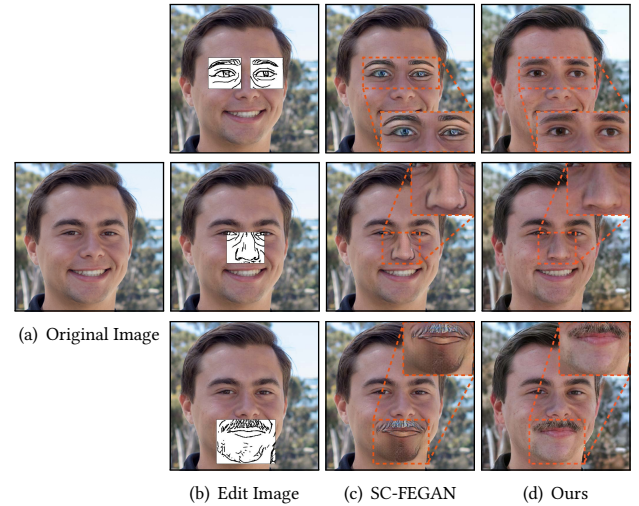(a) Original Image (b) Edit Image (c) SC-FEGAN (d) Ours

Fig. 8. Sketch-based face editing results. Our method leads to more realistic results than SC-FEGAN, a state-of-the-art approach based on image completion. Original images courtesy of SEDS UCSD.

affect the geometry of the generated images. STROTSS [Kolkin et al. 2019], DST [Kim et al. 2020] and StyleGAN2 [Karras et al. 2020] all require an optimization process to generate style transfer images or projected latent code, thus cannot be real-time. Different from the above three algorithms, WCT^2 [Yoo et al. 2019] is a fast feed-forward method without any optimization for style transfer. Despite the realistic face images generated by WCT^2 [Yoo et al. 2019], the style transfer results of their method show a mixed appearance of two reference images without disentangled geometry and appearance.

thus compare our method with the existing solutions for this task, including pSp [Richardson et al. 2020], [Liu et al. 2020], and [Zhang et al. 2020]. For a fair comparison, we use their official source code but re-train their networks on our training dataset. pSp combines an encoder with the StyleGAN decoder and can be applied to sketch-to-image translation. However, since the sketch geometry is encoded into the latent code, generated faces by pSp often do not faithfully respect the input sketches. The style mixing operation adopted in pSp also affects the geometry of synthesized faces undesirably. Liu et al. [2020] design a general network to translate a sketch to an image given a reference image. The geometry of their synthesized faces is maintained well while the appearance shows variations from the reference images (see the color of mouth). Zhang et al. [2020]
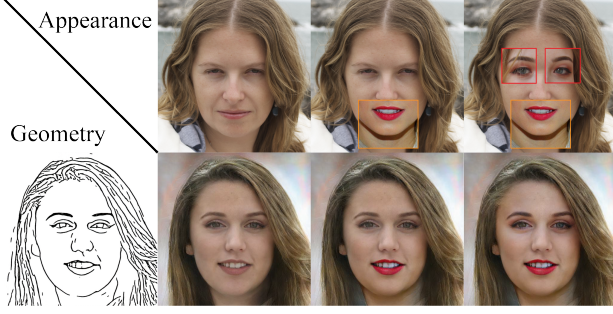
Fig. 9. Our model enables face appearance editing, where we fix the sketch (therefore fixing the geometry feature) and replace target components of the face image. The model is able to robustly fuse the newly added component-level image patches with the original face component images and generate photo-realistic edited face images. Original images courtesy of IoanaP.
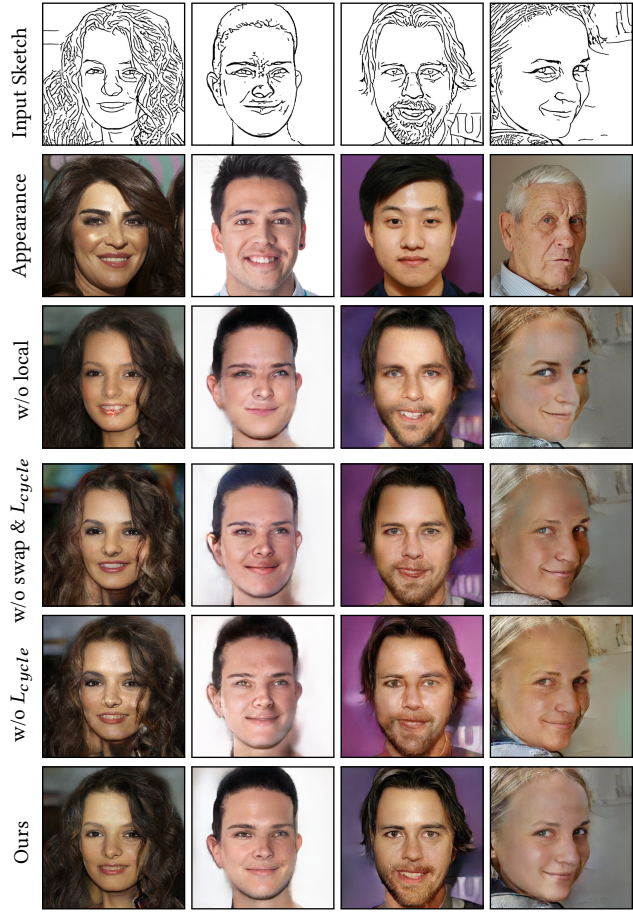

Fig. 10. The results of the ablation study. "w/o local", "w/o swap & L_{Cycle}", and "w/o L_{Cycle}" are our method without the local-to-global strategy, without swapping and cyclic loss, and without the cyclic loss, respectively. Original images courtesy of RAFFI YOUREDJIAN, Matthew Roth, ImagineCup and srgpicker.

propose a cross-domain correspondence network by finding a dense mapping between an input image and a reference image, and then use a warped exemplar to generate results, which, however, may lack the control of details. There are obvious artifacts in the results of Zhang et al. [2020], e.g., the fuzzy nose and the blurred eyes shown in the second and third columns of Fig. 6 (e). It is easy to see that our method combines the geometry and appearance features consistently and generates realistic and detailed results (e.g., the resulting eyebrows, mouth and gradually changed hair). More results generated by our method are shown in the supplementary material with diverse geometry and appearance.

Face Editing. As for face image editing, we compare our method with the state-of-the-art technique SC-FEGAN [2019]. Taking sketches as inputs, our method is able to edit the face images based on modifications to sketches, both globally and locally. In contrast, SC-FEGAN [2019] is not able to synthesize a face image entirely from an input sketch since their method is based on image completion. For a fair comparison, we thus only edit partial sketches and set paired rectangle masks for SC-FEGAN. Without released training code from the authors, we use their official editing GUI with pre-trained model for comparison. As shown in Fig. 8, our method produces more realistic and visually consistent results than SC-FEGAN (see the second row for an edited nose). In addition, their method requires a pre-defined mask for users to specify an area of interest, thus constraining the user experience. Pre-defining such areas sometimes is not easy, especially for detailed lines. For example, in Fig. 7, we draw lines around the chin to generate wrinkles, resulting in lighting and shadow variations. In such cases it is difficult for users to pre-define areas of variations.

With disentangled geometry and appearance, we can also edit the global and/or local appearance without changing the geometry of the face. As shown in Fig. 9, given a sketch and appearance image as inputs, our method can generate a face retaining both global sketch wrinkles and appearance colors. With the fixed sketch, i.e. fixed geometry feature, we give two new reference images replacing the mouth and/or two eyes cropped from other images respectively. Our model is able to robustly fuse the newly added component-level image patches with the original face component images and generate photo-realistic edited face images.

4.2 Ablation Study

We conduct an ablation study on the test set to show the impact of individual key components of our system. Since our method is based on a local-to-global framework, we first show the results without the fusion of local parts. As our swapping strategy contains appearance swapping and cyclic reconstruction, we perform ablation study in two ways, one is to remove swapping and L_{Cycle}, the other is to remove L_{Cycle} but keep L_{Geo}. From Fig. 10, it is obvious that without the local-to-global strategy, the results exhibit artifacts in local details. As shown in the last column, even for non-frontal faces, the local-to-global strategy improves the quality of local details. Results in the fourth row to the last row illustrate that when swapping and cycle consistency loss are utilized, the quality of the synthesized results improves. On the other hand, our baseline model without swapping and L_{Cycle} only successfully performs reconstruction

during training, where the geometry reference and appearance reference are matched. When the appearance or geometry features are changed during testing, the color translation seems to behave like a pixel-by-pixel copy, resulting in obvious inappropriate colorization of pixels. These results confirm that this network does poorly in separating the appearance and geometry. By adding the swapping strategy, the baseline without L_{Cycle} avoids such wrong colors, but does not properly combine those two features. To further enhance the information flow, we add the cycle consistency loss L_{Cycle} to decouple the features from the generated results. This constraint effectively improves the quality of image synthesis.

4.3 Perception Study

As for the tasks of style transfer and sketch-to-image translation, the face images generated by alternative methods tend to generate reasonably realistic face images (see Figs. 5, 6), and the visual quality differences are subtle. It is therefore difficult for existing general image quality measures such as FID [Heusel et al. 2017] and LPIPS [Zhang et al. 2018] to distinguish between their qualities. To evaluate the visual quality and the faithfulness (i.e., the similarity to the geometry and appearance images) of synthesized faces, we conducted a perception study.

The evaluation was done via two online questionnaires. The first perception study aimed to evaluate the effect of style transfer. We showed two input images, the geometry reference image and the appearance reference image, and five synthesized images (including WCT[2] [Yoo et al. 2019], STROTSS [Kolkin et al. 2019], DST [Kim et al. 2020], StyleGAN2 [Karras et al. 2020] and ours) for each example, placed side by side in a random order to avoid bias. Each participant was asked to evaluate 20 examples according to three criteria: the maintenance of the geometry reference, the similarity to the appearance reference and the realism of generated images, each in a five-point Likert scale (1 = strongly negative to 5 = strongly positive). In total, 40 participants participated in this study and we got 40 (participants) × 20 (questions) = 800 subjective evaluations for each method. We performed one-way ANOVA tests on five methods in aspects of 'Geometry', 'Appearance' and 'Realism' corresponding to the three criteria respectively. As shown in Fig. 11(a), the statistics of the evaluation results were plotted. We found significant effects of our method for all three criteria: geometry ($F_{(4,95)} = 3.4164, p < 0.05$), appearance ($F_{(4,95)} = 2.6196, p < 0.05$) and realism ($F_{(4,95)} = 2.6012, p < 0.05$).

The second perception study was conducted to evaluate the quality of sketch-to-image translation. Similar to the former one, for each example we showed the user two input images, the sketch image for geometry and the appearance image, and three synthesized images by Liu et al. [2020], Zhang et al. [2020] and ours, placed side by side in a random order. Each participant was asked to evaluate 20 examples according to three criteria: the visual quality of synthesized images, the similarity to the input sketch and the faithfulness in appearance, each in a five-point Likert scale (1 = strongly negative to 5 = strongly positive). In total, we got 40 (participants) × 20 (questions) = 800 subjective evaluations for each method. The second column of Fig. 11(b) shows the statistics of these three methods. We also performed the ANOVA tests on the three aspects, and
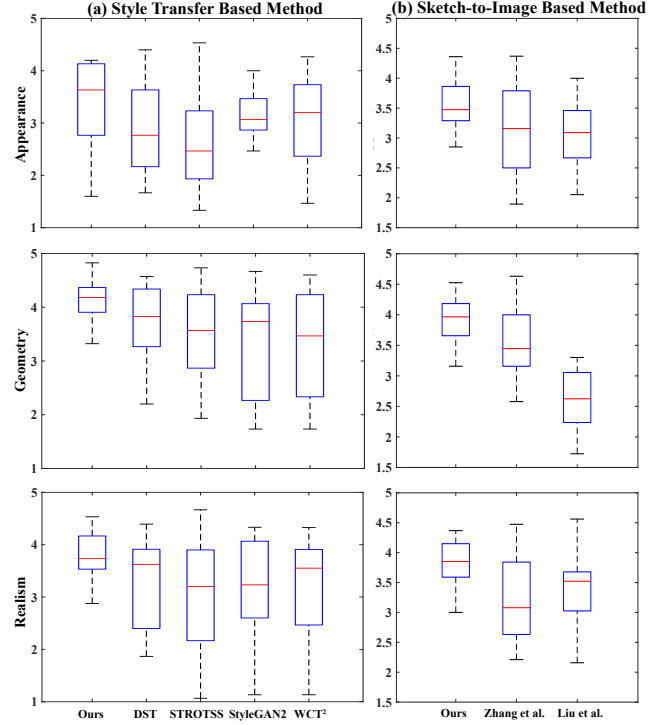

Fig. 11. Box plots of the average quality and faithfulness perception scores in terms of similarity between generated faces and the reference faces in appearance and geometry respectively, as well as visual realism of the generated faces, based on the participants in the two user studies. (a) The comparison of style transfer with five methods: WCT[2] [Yoo et al. 2019], STROTSS [Kolkin et al. 2019], DST [Kim et al. 2020], StyleGAN2 [Karras et al. 2020], and ours. (b) the evaluation of sketch-to-image synthesis with Liu et al. [2020], Zhang et al. [2020], and our method.
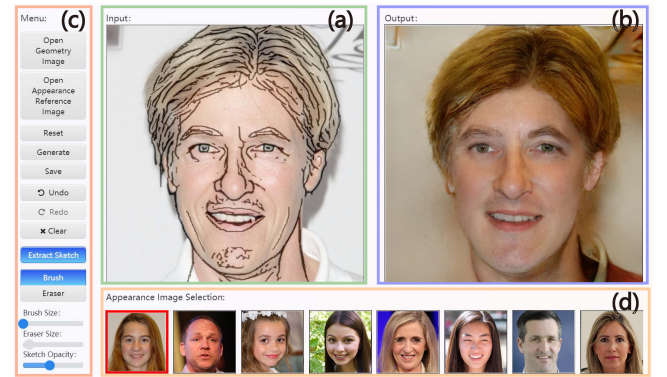

Fig. 12. A screenshot of our sketch-based interface for face editing. (a) and (b) are an input sketch overlaid on top of a geometry image and the corresponding generated face image, respectively. (c) and (d) are the control function panel and the appearance image list, respectively. Original images courtesy of NothingToDeclare, WRI Ross Center for Sustainable Cities, Pawel Loj, seanccochran, Marcus Harzem, Malcolm Slaney, peiflickr and TRE - RJ.
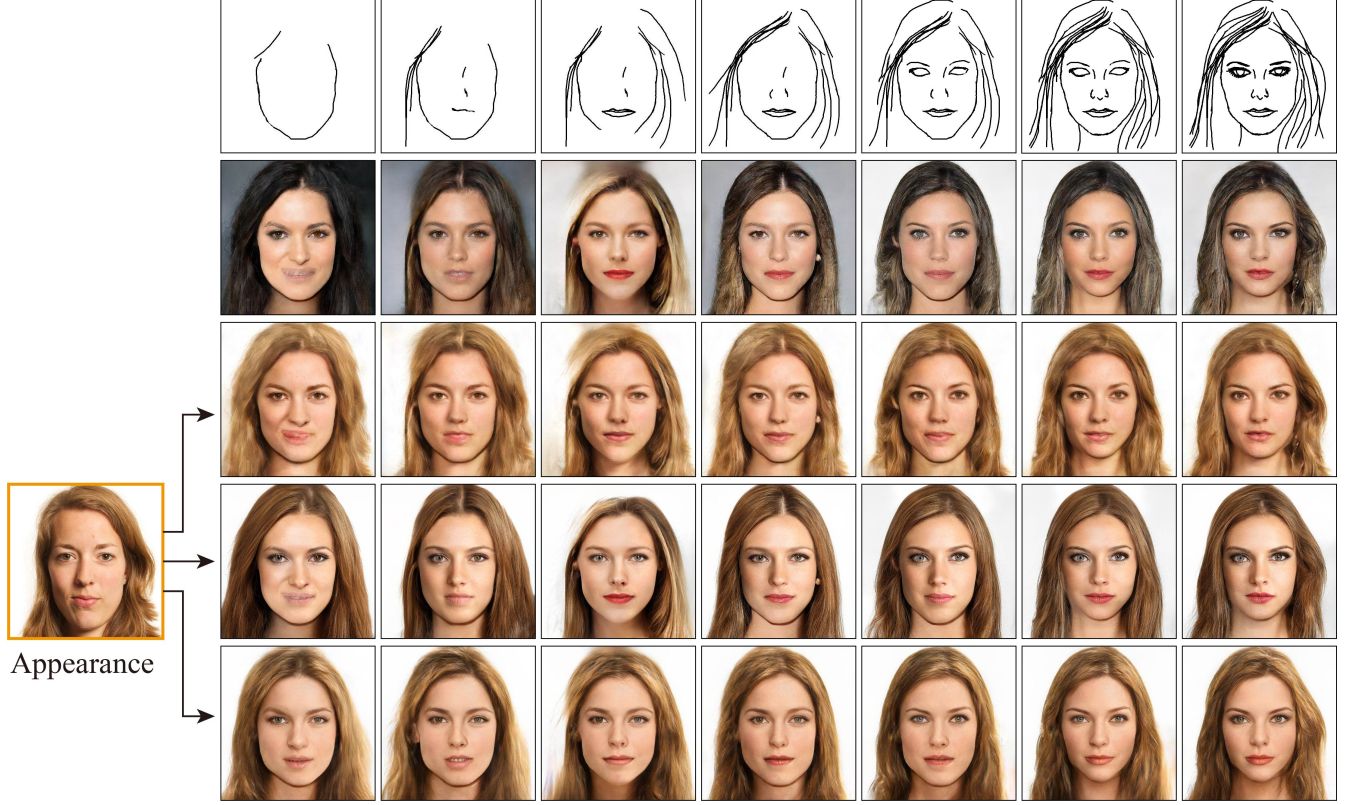
Fig. 13. Comparisons of image generation given a sequence of hand-drawn sketches (1st row). The results by DeepFaceDrawing [Chen et al. 2020] (2nd row) exhibit clear appearance changes during editing. The 3rd row shows the results of using a style transfer technique [Kolkin et al. 2019] to transfer the style of a reference image (Left) to the images generated by DeepFaceDrawing. The 4th row shows the results of using swapping autoencoder [Park et al. 2020] with the same input as the 3rd row. Taking the sketches (1st row) and the appearance reference (Left) as input, our method (bottom row) generates more stable and realistic results. Original images courtesy of Sebastiaan ter Burg.

get the values for geometry ($F_{(2,57)} = 8.9514, p < 0.05$), appearance ($F_{(2,57)} = 3.8033, p < 0.05$) and realism ($F_{(2,57)} = 6.0887, p < 0.05$). It is clear that our method achieved a significant improvement over the other two methods.

5 APPLICATIONS

Thanks to the disentanglement of geometry and appearance, our method can be adapted to many applications. In this section, we describe three applications: 1) Sketch editing interface; 2) Hand-drawn sketch to face image generation; 3) Disentangled morphing.

5.1 Sketch Editing Interface

First, we design a real-time sketch-based face editing user interface (Fig. 12), enabling detailed face editing via sketching. The control panel (Fig. 12 (c)) consists of some necessary tools such as image opening tools, editing tools, etc. We also offer a sketch extraction function, facilitating users to translate a photo to a sketch for further editing. Our system generates synthesized results in real-time, based on modifications of sketches. A list of reference appearance images are placed at the bottom of interface (Fig. 12 (d)), and users can

upload their own face images to the list. By selecting an appearance image, users can control the appearance of the generated results.

5.2 Hand-drawn Sketch to Image Generation

The editing module in our system enables users to draw faces via coarse or fine sketches. For users with little experience in drawing, they can produce realistic faces even from simple sketches (see the first sketch image only drawing an outline of a face in Fig. 13). If the user is expert at drawing, he/she can edit the sketches with a brush or an eraser in our system to generate a more specific face closer to his/her conception. As shown in Fig. 13, a progressive sketch sequence is drawn by the user, and we compare the synthesized face sequence with DeepFaceDrawing [Chen et al. 2020], DeepFaceDrawing with style transfer and swapping autoencoder from a reference image, and our results based on the geometry of sketches and appearance of the same reference. It is evident that the results of DeepFaceDrawing (the 2nd row) show faces with diverse appearances, e.g. varied hair and skin colors. Such sudden and uncontrollable changes are not desired by the user who is making incremental changes to the sketch. Even combined with a style transfer method [Kolkin et al. 2019] using the left image
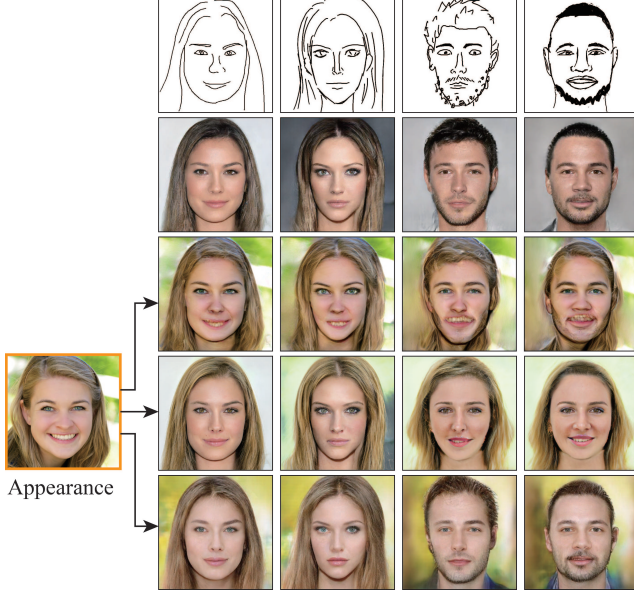
Fig. 14. Comparisons of face generation given hand-drawn sketches (1st row). DeepFaceDrawing [Chen et al. 2020] (2nd row) has no control of appearance. The results of using a style transfer technique [Kolkin et al. 2019] (3rd row) have some artifacts on the mouth and hair. When using the swapping autoencoder [Park et al. 2020], the results (the last two columns in the 4th row) do not retain the face geometry specified by the sketches well. Taking the sketches (1st row) and the appearance reference (Left) as input, our method (bottom row) generates stable results. Original images courtesy of Tom Kunz.
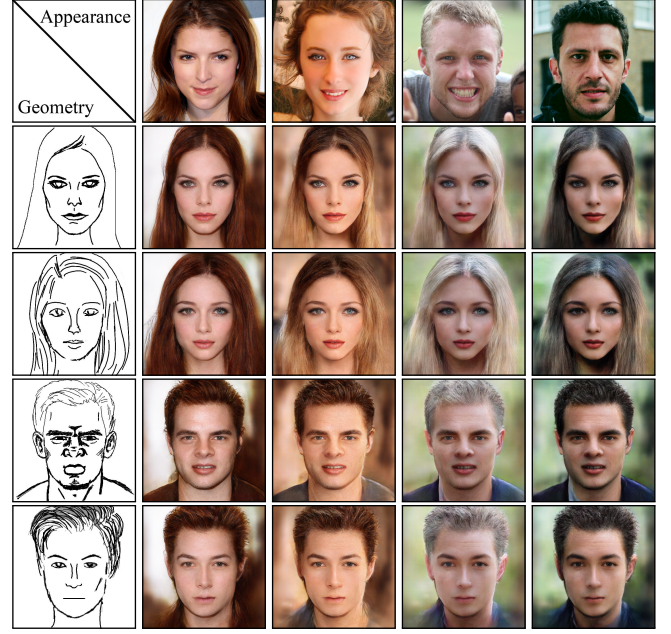

Fig. 15. Face synthesis results with hand-drawn sketches. With the control of appearance, our method can generate diverse photo-realistic faces from hand-drawn sketches. Original images courtesy of David Shankbone, Mark Peers, Kayla Hayes and uncoolbob.

as a reference, the results of DeepFaceDrawing show some artifacts in local areas like the mouth or eyes. Besides, our method is more efficient than [Kolkin et al. 2019], which takes around 1 minute to generate a result. Combining DeepFaceDrawing with the swapping autoencoder [Park et al. 2020], the results show slight color difference with the reference image and are also affected by the generation of DeepFaceDrawing (4th column). Even with the appearance swapping, differences between editing frames also exist (especially on hair), while our results are more robust and faithful to incremental changes. A similar comparison is shown in Fig. 14 where we use full hand-drawn sketches instead of an editing sequence. DeepFaceDrawing [Chen et al. 2020] generates plausible face images (2nd row), but without appearance control. Applying a style transfer technique [Kolkin et al. 2019] and the swapping autoencoder [Park et al. 2020] to the results of DeepFaceDrawing can partially address this problem. However, applying these two steps in succession may accumulate errors. The results of [Kolkin et al. 2019] (3rd row) have some artifacts on the mouth and hair. The swapping autoencoder [Park et al. 2020] (4th row) can generate attractive results, but the resulting faces may not retain the geometry of the sketches (third and last columns). As shown in Fig. 14 (bottom row) and Fig. 15, in our system, users can control both the geometry with hand-drawn sketches and the appearance with diverse references.

5.3 Disentangled Morphing

Our method learns two disentangled latent spaces of the geometry and appearance respectively and enables a novel application of generating face images with reference of given geometry or appearance patterns. We demonstrate via disentangled face image morphing that a clear disentanglement between geometry and appearance has been achieved by our method. As shown in Fig. 16, we achieve controllable interpolation along two dimensions (Appearance and Geometry). Given two input images (I_1, I_2), our method can extract the geometry code f_G^1, f_G^2 and the appearance code f_A^1, f_A^2 from both images. Then, we conduct linear interpolation in the geometry and/or appearance latent spaces, using the corresponding latent codes from image I_1 and I_2. The results demonstrate that our interpolation faces are controllable and every interpolated face enjoys high fidelity.

6 CONCLUSION, LIMITATIONS AND FUTURE WORK

This work presented a structured disentanglement framework for face generation and editing, carried out in a local-to-global manner. Our key observation is that geometry and appearance features of face images can be effectively disentangled, and sketches serve as an ideal intermediate representation for geometry features. Therefore, sketches can impose a strong constraint during disentanglement. The component-level geometry and appearance disentanglement is achieved by *Local Disentanglement* modules which are trained using a swapping strategy, while the *Global Fusion* module performs coherent local-to-global image generation from feature maps of facial image patches. Through extensive experiments, we prove that
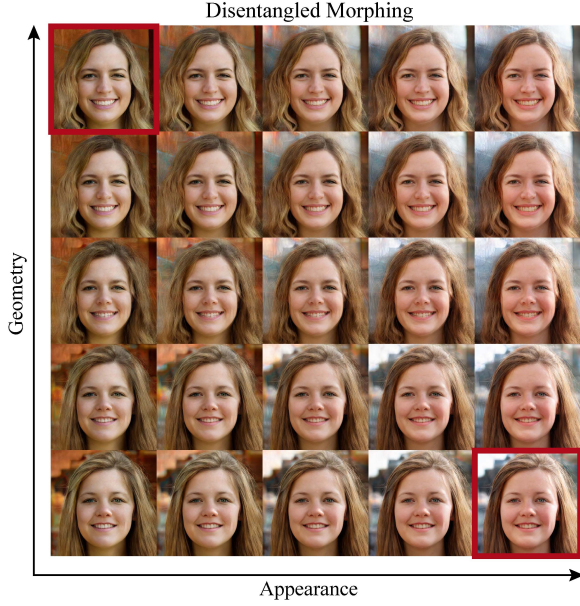
Disentangled Morphing

Fig. 16. The results of disentangled morphing by interpolating the top-left and bottom-right faces (in red boxes). The remaining face images are generated automatically with our method. In each row, the appearance of the images is interpolated while keeping the geometry unchanged, while in each column, the geometry is interpolated while retaining the appearance. Original images courtesy of Lydia Liu and SISTERKLAAS Lore.

(a) Geometry   (b) Appearance   (c) result

Fig. 17. A less successful example. When an appearance image contains a large occlusion, there may be some color bias in the generated result. Original images courtesy of UC Davis College of Engineering and taymtaym.

our approach can generate much more realistic results than existing methods. We also adapted our system for novel applications such as sketch editing, hand-drawn sketch based face image generation and disentangled face morphing.

One limitation of this work is that we only disentangle the geometry and appearance while other attributes such as head pose have not been considered in our current implementation. As shown in Figs. 17 and 18, when there is a large rotation of the subject's head or there is substantial occlusion, there may be some color bias in the generated results. Lighting is another challenging problem for most existing face synthesis methods and is not explicitly disentangled in our framework, so it is difficult to finely control complicated lighting conditions by sketches or reference images. As future work, it would be useful to explore disentanglement of other attributes such as head pose and lighting to make the method more

Fig. 18. The results for non-frontal faces. Images in the first column provide appearance and the rest images are generated by our method. The inputs of sketches can be seen in the supplementary material. As the rotation angle increases, the quality of the results decreases. Original images courtesy of Senterpartiet (Sp) and Phillip Nguyen.

general. Besides, sketches have semantic ambiguity and in some extreme cases, it is hard even for humans to distinguish the accurate boundary between neck, hair and background. This ambiguity may sometimes cause some artifacts on the outer boundary of the face foreground, leading to blurred hair. As future work, semantic masks may be combined with sketches to generate more attractive results. Furthermore, while we allow detailed editing of geometry through sketching, our current approach uses an appearance reference image to control the appearance of the generated face image. This could be improved using other forms of input, such as strokes with color, which would be more flexible, such as [Sangkloy et al. 2017]. This is a promising research direction in the future but it remains challenging, because the face appearance not only contains the color but also the material attributes, and it is very difficult to stroke the complex materials of the face. The consistency between the color strokes should also be maintained. These will be explored in the future research work.

ACKNOWLEDGMENTS

REFERENCES

Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. 2020. StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. *arXiv preprint* (2020). arXiv:2008.02401

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2016. Neural Photo Editing with Introspective Adversarial Networks. In *ICIR*.

John Canny. 1986. A computational approach to edge detection. *PAMI* (1986), 679–698.

Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. 2020. DeepFaceDrawing: Deep Generation of Face Images from Sketches. *ACM Trans. Graph.* 39, 4, 72:1–72:16.

Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. 2009. Sketch2Photo: Internet Image Montage. *ACM Trans. Graph.* 28, 5 (2009), 1–10.

Wengling Chen and James Hays. 2018. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In *CVPR*. 9416–9425.

Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning. In *CVPR*. 5153–5162.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *CVPR*. 2414–2423.

Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. 2018. Image-to-image translation for cross-domain disentanglement. In *NIPS*. 1294–1305.

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

Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. 2019. Mask-Guided Portrait Editing With Conditional GANs. In *CVPR*. 3436–3445.

Xiaoguang Han, Chang Gao, and Yizhou Yu. 2017. DeepSketch2Face: A Deep Learning Based Sketching System for 3D Face and Caricature Modeling. *ACM Trans. Graph.* 36, 4 (2017), 126:1–126:12.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *NIPS*. 9841–9850.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*. 6626–6637.

Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. 2020. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Information Sciences*.

Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 2017. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. In *ICCV*. 2458–2467.

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization. In *ICCV*. 1510–1519.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-image Translation. In *ECCV*. 179–196.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *CVPR*. 5967–5976.

Youngjoo Jo and Jongyoul Park. 2019. SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color. In *ICCV*. 1745–1753.

Justin Johnson, Alexandre Alahi, and Fei-Fei Li. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*. 694–711.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*. 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 *CVPR*. 8107–8116.

Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2020. Deformable Style Transfer. In *ECCV*. 246–261.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICIR*.

Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2019. Style Transfer by Relaxed Optimal Transport and Self-Similarity. In *CVPR*. 10051–10060.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020b. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *CVPR*. 5548–5557.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. 2020c. DRIT++: Diverse image-to-image translation via disentangled representations. *IJCV*, 2402–2417.

Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. 2020a. Reference-Based Sketch Image Colorization Using Augmented-Self Reference and Dense Semantic Correspondence. In *CVPR*. 5800–5809.

Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *CVPR*. 2479–2486.

Mu Li, Wangmeng Zuo, and David Zhang. 2016. Convolutional Network for Attribute-driven and Identity-preserving Human Face Generation. *arXiv preprint* (2016).

Yuhang Li, Xuejin Chen, Feng Wu, and Zheng-Jun Zha. 2019. LinesToFacePhoto: Face Photo Generation From Lines With Conditional Self-Attention Generative Adversarial Networks. In *ACM Multimedia*. 2323–2331.

Yuhang Li, Xuejin Chen, Binxin Yang, Zihan Chen, Zhihua Cheng, and Zheng-Jun Zha. 2020. DeepFacePencil: Creating Face Images from Freehand Sketches. In *ACM Multimedia*. 991–999.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal Style Transfer via Feature Transforms. In *NIPS*. 386–396.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer through Deep Image Analogy. *ACM Trans. Graph.* 36, 4 (2017), 120:1–120:15.

Alexander H. Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. 2018. A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation. In *NIPS*. 2595–2604.

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. 2020. Self-Supervised Sketch-to-Image Synthesis. *arXiv preprint* (2020). arXiv:2012.09290

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-Shot Unsupervised Image-to-Image Translation. In *ICCV*. 10550–10559.

Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. 2018. The Contextual Loss for Image Transformation with Non-Aligned Data. In *ECCV*. 800–815.

Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784* (2014).

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In *ICCV*. 7587–7596.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *CVPR*. 2337–2346.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. 2020. Swapping Autoencoder for Deep Image Manipulation. In *NIPS*. 7198–7211.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NIPS*. 8024–8035.

Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018. Faceshop: Deep sketch-based face image editing. *ACM Trans. Graph.* 37, 4 (2018), 99:1–99:13.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2020. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. *arXiv preprint* (2020). arXiv:2008.00951

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2017. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. 234–241.

Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling Deep Image Synthesis With Sketch and Color. In *CVPR*. 6836–6845.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *NIPS*. 20154–20166.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *CVPR*. 9240–9249.

Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. 2020. MichiGAN: Multi-Input-Conditioned Hair Image Generation for Portrait Editing. *ACM Trans. Graph.* 39, 4 (2020), 95:1–95:13.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Perez, Michael Zollhofer, and Christian Theobalt. 2020. StyleRig: Rigging StyleGAN for 3D Control Over Portrait Images. In *CVPR*. 6141–6150.

J. Wang, J. Zhang, Z. Lu, and S. Shan. 2019. DFT-Net: Disentanglement of Face Deformation and Texture Synthesis for Expression Editing. In *International Conference on Image Processing (ICIP)*. 3881–3885.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *CVPR*. 8798–8807.

Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. In *ICCV*. 1395–1403.

Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. 2020. Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches. In *ECCV*. 601–617.

Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. 2019. Photorealistic Style Transfer via Wavelet Transforms. In *ICCV*. 9035–9044.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. 2019b. Free-Form Image Inpainting With Gated Convolution. In *ICCV*. 4470–4479.

Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. 2019a. Multi-mapping Image-to-Image Translation via Learning Disentanglement. In *NIPS*. 2990–2999.

Jianfu Zhang, Yuanyuan Huang, Yaoyi Li, Weijie Zhao, and Liqing Zhang. 2019. Multi-Attribute Transfer via Disentangled Representation. In *AAAI*. 9195–9202.

Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. 2020. Cross-domain Correspondence Learning for Exemplar-based Image Translation. In *CVPR*. 5142–5152.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. 586–595.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward Multimodal Image-to-Image Translation. In *NIPS*. 465–476.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017a. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *ICCV*. 2242–2251.

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. In *CVPR*. 5103–5112.