MichiGAN:
Multi-Input-Conditioned Hair Image Generation for Portrait Editing

ZHENTAO TAN, University of Science and Technology of China
MENGLEI CHAI, Snap Inc.
DONGDONG CHEN, Microsoft Cloud AI
JING LIAO, City University of Hong Kong
QI CHU*, University of Science and Technology of China
LU YUAN, Microsoft Cloud AI
SERGEY TULYAKOV, Snap Inc.
NENGHAI YU, University of Science and Technology of China

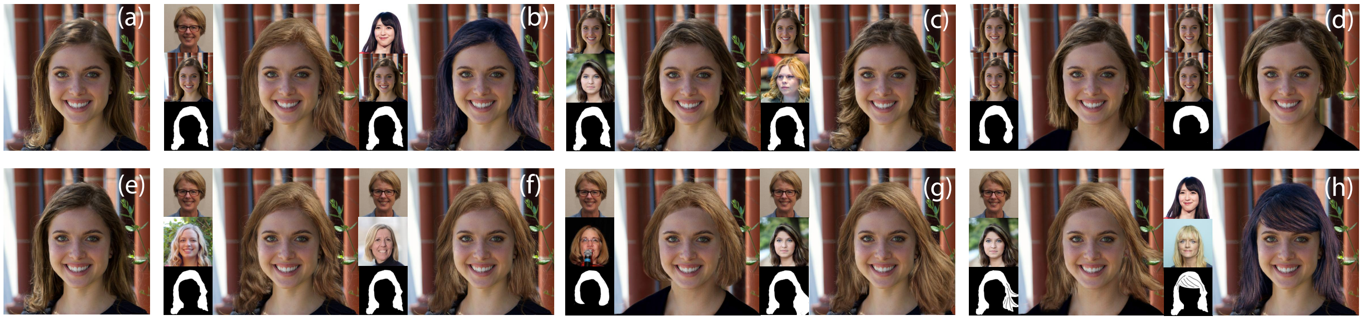

Fig. 1. Given an input portrait image (a), MichiGAN is capable of enabling multiple input conditions for disentangled hair manipulation. It not only supports editing one attribute individually including appearance (b), structure (c), and shape (d) while keeping the background unchanged, but also allows manipulating multiple attributes (f&g) jointly. The users can also leverage our interactive system to create more customized hair manipulation results (h). (e) is the reconstruction result when all the condition inputs are from (a) itself.

Despite the recent success of face image generation with GANs, conditional hair editing remains challenging due to the under-explored complexity of its geometry and appearance. In this paper, we present MichiGAN (Multi-Input-Conditioned Hair Image GAN), a novel conditional image generation method for interactive portrait hair manipulation. To provide user control over every major hair visual factor, we explicitly disentangle hair into four orthogonal attributes, including shape, structure, appearance, and background. For each of them, we design a corresponding condition module to represent, process, and convert user inputs, and modulate the image generation pipeline in ways that respect the natures of different visual attributes. All these condition modules are integrated with the backbone generator to form the final end-to-end network, which allows fully-conditioned hair generation from multiple user inputs. Upon it, we also build an interactive portrait hair editing system that enables straightforward manipulation of hair by projecting intuitive and high-level user inputs such as painted masks, guiding strokes, or reference photos to well-defined condition representations. Through extensive experiments and evaluations, we demonstrate the superiority of our method regarding both result quality and user controllability.

CCS Concepts: • **Computing methodologies → Image manipulation**; **Neural networks**; *Computer vision representations*; *Graphics systems and interfaces.*

Additional Key Words and Phrases: interactive portrait editing; conditional hair image generation; generative adversarial networks

ACM Reference Format:
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, Article 1 (July 2020), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Human hair is so delicate, variable, and expressive that it constantly plays a unique role in depicting the subject in a face image. Given its diversity and flexibility, the urge to manipulate hair in a portrait photo, with a full spectrum of user control, has been long coveted

*Corresponding author.

Authors' addresses: Zhentao Tan, University of Science and Technology of China, tzt@mail.ustc.edu.cn; Menglei Chai, Snap Inc., mchai@snap.com; Dongdong Chen, Microsoft Cloud AI, cddlyf@gmail.com; Jing Liao, City University of Hong Kong, jingliao@cityu.edu.hk; Qi Chu, University of Science and Technology of China, qchu@ustc.edu.cn; Lu Yuan, Microsoft Cloud AI, luyuan@microsoft.com; Sergey Tulyakov, Snap Inc., stulyakov@snap.com; Nenghai Yu, University of Science and Technology of China, ynh@ustc.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2020 Association for Computing Machinery.
0730-0301/2020/7-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to facilitate truly personalized portrait editing – from extending a particular hair wisp, altering the structure of a local region, to entirely transferring the hair color and appearance.

However, unlike most other parts of the human face, its peculiar geometry and material make hair extremely difficult to analyze, represent, and generate. The complexity of visual perception comes from multiple factors that the user may intend to edit or preserve. If we take a closer look, by roughly decomposing the hair region into its corresponding mask shape and color appearance, a few key aspects can be identified that are crucial to the visual quality. Shape-wise, the intricate mask boundaries, and varying opacity can lead to challenges of precise hair shape control and seamless background blending. Appearance-wise, the anisotropic nature of hair fibers tends to entangle both the strand flows and the material properties, in which the former renders the major local structures while the latter determines the color palette and inter-strand variations in a globally consistent manner. Therefore, to achieve controllable hair manipulation, we need to have the system not only capable of generating photo-realistic hair images, but also able to bridge various user controls and use them to condition the hair generation in ways that respect the natures of different visual factors.

Fortunately, the latest advances in deep neural networks, or more specifically, generative adversarial networks (GAN), have driven significant progress on human face synthesis. Recent successes, such as ProgressiveGAN [Karras et al. 2018] and StyleGAN [Karras et al. 2019], already make it possible to produce highly realistic human face, including hair, from a random latent code. Yet conditioned generation, the heart of interactive face editing, is thought to be more challenging and less explored. Despite the recent efforts on conditioned face generation with semantic maps [Lee et al. 2019] or sketches [Jo and Park 2019], we are still far from reaching high-quality hair editing that is both fully-controllable and user-friendly.

In this work, we present *MichiGAN* (Multi-Input-Conditioned Hair Image GAN), a novel conditional hair image generation method for interactive portrait manipulation. To achieve so, we explicitly disentangle the information of hair into a quartet of attributes – *shape*, *structure*, *appearance*, and *background*, and design deliberate representations and conditioning mechanisms to enable orthogonal control within the image generation pipeline.

To this end, we propose three distinct condition modules for these attributes according to their particular perceptual characteristics and scales. Specifically, spatially-variant attributes of *shape* and *structure* are represented as fuzzy semantic masks and weighted orientation maps, respectively, which modulate the network with the conditional normalization structures [Huang and Belongie 2017; Park et al. 2019]. More globally, *appearance* is encoded through our mask-transformed feature extracting network, which acts as the latent code input to the very beginning of the network and bootstraps the generator with appearance styles guided by arbitrary references. In addition, a *background* encoder is placed parallel to the generation branch, which keeps background intact by progressively injecting features into the generator in a mask-aware way without loss of generation capability.

By integrating all these condition modules, we have an end-to-end image generation network that can provide complete control over every major hair attribute. Tailored for such a complicated network, we further develop a data preparation and augmentation pipeline for disentangling hair into compatible training data pairs, together with a combination of novel loss functions to achieve smooth and robust training. By extensively evaluating MichiGAN on a wide range of in-the-wild portrait images and comparing it with state-of-the-art conditioned image generation methods and alternative network designs, we demonstrate the superiority of the proposed method on both result quality and controllability.

Moreover, we build an interactive portrait hair editing system based on MichiGAN. It enables straightforward manipulation of hair by projecting intuitive and high-level user inputs such as painted masks, guiding strokes, or reference photos to well-defined condition representations. By using this hair editing interface, users are allowed to modify the hair shape through mask painter, alter or inpaint local hair structure with sparse stroke annotations, replace hair appearance using another hair reference, or even edit multiple attributes all at once concurrently.

In summary, the main contributions of this work are:

- An *explicit disentanglement of hair visual attributes*, and a set of *condition modules* that implement the effective condition mechanism for each attribute with respect to its particular visual characteristics;
- An *end-to-end conditional hair generation network* that provides complete and orthogonal control over all attributes individually or jointly;
- An *interactive hair editing system* that enables straightforward and flexible hair manipulation through intuitive user inputs.

The training code, pre-trained models, and the interactive system are made publicly available to facilitate future research.

2 RELATED WORK

Portrait Manipulation. Portrait image manipulation is a long-standing while extensively studied area in both vision and graphics communities. Early approaches tackle specific application-driven problems with conventional image editing techniques, such as face swapping [Bitouk et al. 2008], face shape beautification [Leyvand et al. 2008], and portrait enhancement [Joshi et al. 2010]. More recently, the popularization of image-based 3D face reconstruction [Blanz and Vetter 1999] enables lots of 3D-aware portrait manipulation methods, on expression editing [Yang et al. 2011], relighting [Wang et al. 2009], aging [Kemelmacher-Shlizerman et al. 2014], camera manipulation [Fried et al. 2016], geometry reshaping [Cao et al. 2014], identity replacement [Dale et al. 2011; Kemelmacher-Shlizerman 2016], and performance reenactment [Thies et al. 2016]. However, the limited fitting accuracy and model quality of face proxy tends to affect the result realism. Thanks to the recent progress on deep neural networks, high-fidelity portrait manipulation has been made feasible in multiple scenarios, including sketch-based generation [Jo and Park 2019; Lee et al. 2019], attribute editing [Geng et al. 2019; Shu et al. 2017], view synthesis [Nagano et al. 2019], and relighting [Sun et al. 2019].

Hair Modeling and Editing. As a critical yet challenging component of the human face, hair is also of great interest to researchers. Most hair acquisition pipelines focus on capturing hair geometry from various image inputs, including multi-view [Hu et al. 2014;

Luo et al. 2013; Paris et al. 2008; Zhang et al. 2017a] and single-view [Chai et al. 2016, 2013; Hu et al. 2015; Liang et al. 2018; Saito et al. 2018; Zhou et al. 2018]. Utilizing the coarse hair geometry approximated from single-view images, a few hair editing methods are proposed on hair transferring [Chai et al. 2012], animation [Chai et al. 2013], relighting [Chai et al. 2015], and morphing [Weng et al. 2013]. But the lack of in-depth understanding and manipulation approaches to many of key visual factors greatly compromise the result quality and editing flexibility. The highly realistic portrait synthesis results from recent GAN-based image generated methods [Karras et al. 2019] have shown great potentials in bridging this quality gap. However, conditioned image generation is generally thought to be much more difficult than generation from random codes. Although some recent work [Jo and Park 2019; Lee et al. 2019; Qiu et al. 2019; Wei et al. 2018] have achieved progress on hair generation conditioned by limited types of inputs, these methods are unfortunately not intuitively controllable and universally applicable. In this paper, we propose a set of disentangled hair attributes to cover the full spectrum of visual hair factors, and for each attribute, we design a conditional module that could effectively control the generator with user-friendly inputs. This completely-conditioned and controllable hair generation has not been achieved before.

GAN-based Image Generation. Thanks to the powerful modeling capacity of deep neural networks, varieties of generative models have been proposed to learn to synthesize images in an unconditional setting. Typical methods include generative adversarial networks (GANs) [Goodfellow et al. 2014; Mao et al. 2017; Metz et al. 2017] and variation auto-encoders (VAE) [Doersch 2016; Sønderby et al. 2016]. Compared to VAE, GANs are demonstrated to be more popular and capable of modeling fine-grained details. Starting from a random latent code, GANs can generate fake images with the same distribution as natural images in a target domain. The famous work ProgressiveGAN [Karras et al. 2018] first leveraged a progressive generative network structure that can generate very highly realistic facial images, including hair. By further incorporating the latent and noise information into both the shallow and deep layers, StyleGAN [Karras et al. 2019] further improved the generation quality significantly. Our work is built on GANs but aims for the conditional image generation task, which is much more challenging due to multiple elaborate conditions for controlling desired hair synthesis. Utilizing these pre-trained GAN networks, many recent works [Abdal et al. 2019; Bau et al. 2019] achieve convincing image editing results (including portrait) by manipulating the embedded features in the latent spaces of these networks.

Conditional Image Generation. Many recent works leverage conditional GAN (cGAN) for image generation, conditioned on different types of inputs, such as category labels [Brock et al. 2019; Mirza and Osindero 2014; Odena et al. 2017], text [Reed et al. 2016; Yin et al. 2019; Zhang et al. 2017b], or images [Isola et al. 2017; Jo and Park 2019; Lee et al. 2019; Park et al. 2019; Wang et al. 2018]. Among these various forms, image-conditioned generation, i.e., image-to-image translation, lies in the heart of interactive image editing. Pix2Pix [Isola et al. 2017] proposes a unified framework for different applications, such as transforming a semantic segmentation mask or sketch to a photo-realistic image. It is further improved by Pix2PixHD [Wang et al. 2018] with multi-scale generators for high-resolution image generation. SPADE [Park et al. 2019] proposes spatially-adaptive normalization for better converting segmentation mask into photo-realistic image. A few works [Hong et al. 2018; Singh et al. 2019; Yao et al. 2018] propose disentangled image generation, which share similar motivations as ours. Recently, some works have made efforts on interactive facial image editing using conditional GAN. For example, MaskGAN [Lee et al. 2019] focuses on facial image generation based on fine-grained semantic masks. SC-FEGAN [Jo and Park 2019] extends the conditional inputs to sketches and color, which allows users to control the shape and color of the face. Despite the fact that great progress has been made in conditional image generation, we are still far from reaching high-quality hair editing that is both fully-controllable due to its complexity. In this paper, we focus on hair image generation conditioned on multiple attributes that can enable user-friendly interactive hair editing.

3 OVERVIEW

Given an input portrait image I, our MichiGAN aims at performing conditional hair editing to the hair region according to various user control inputs, while keeping the background unchanged. The output is denoted as I_{out}.

As illustrated in Fig. 2, the proposed pipeline consists of a backbone generation network and three condition modules, which modulate the image generation network in various ways that will be introduced later. Thanks to these three condition modules, the network provides orthogonal controls over four different hair attributes: shape, structure, appearance, and background; represented as semantic hair mask M, dense orientation map O, hair appearance reference I_{ref}, and the background region of the input image I, respectively. By combining all condition inputs, we can formulate the generation process as a deterministic function $\mathcal{G}$ as:

$$I_{out} = \mathcal{G}(M, O, I_{ref}, I). \tag{1}$$

These conditions, except the background, can either come from new sources in the interactive system or simply be extracted from the target/reference images. Therefore, users are not just allowed only to edit one attribute but fix all others, but can change multiple of them altogether at the same time.

We will elaborate on the technical method and the entire system in the next three sections, which are organized as follows. We first introduce the representations of our conditions, structures, and mechanisms of their corresponding condition modules (Sec. 4). Then, we present the backbone generation network of MichiGAN that integrates all condition modules, and discuss its training strategies (Sec. 5). Finally, our interactive hair editing system, built upon MichiGAN, is introduced together with various interaction methods that enable user-friendly control over all attributes (Sec. 6).

4 CONDITION MODULES

In this section, we introduce our condition modules for controlling specific semantic attributes of hair during editing. In light of different natures of target attributes, their corresponding condition modules should vary as well. Here we propose three distinct

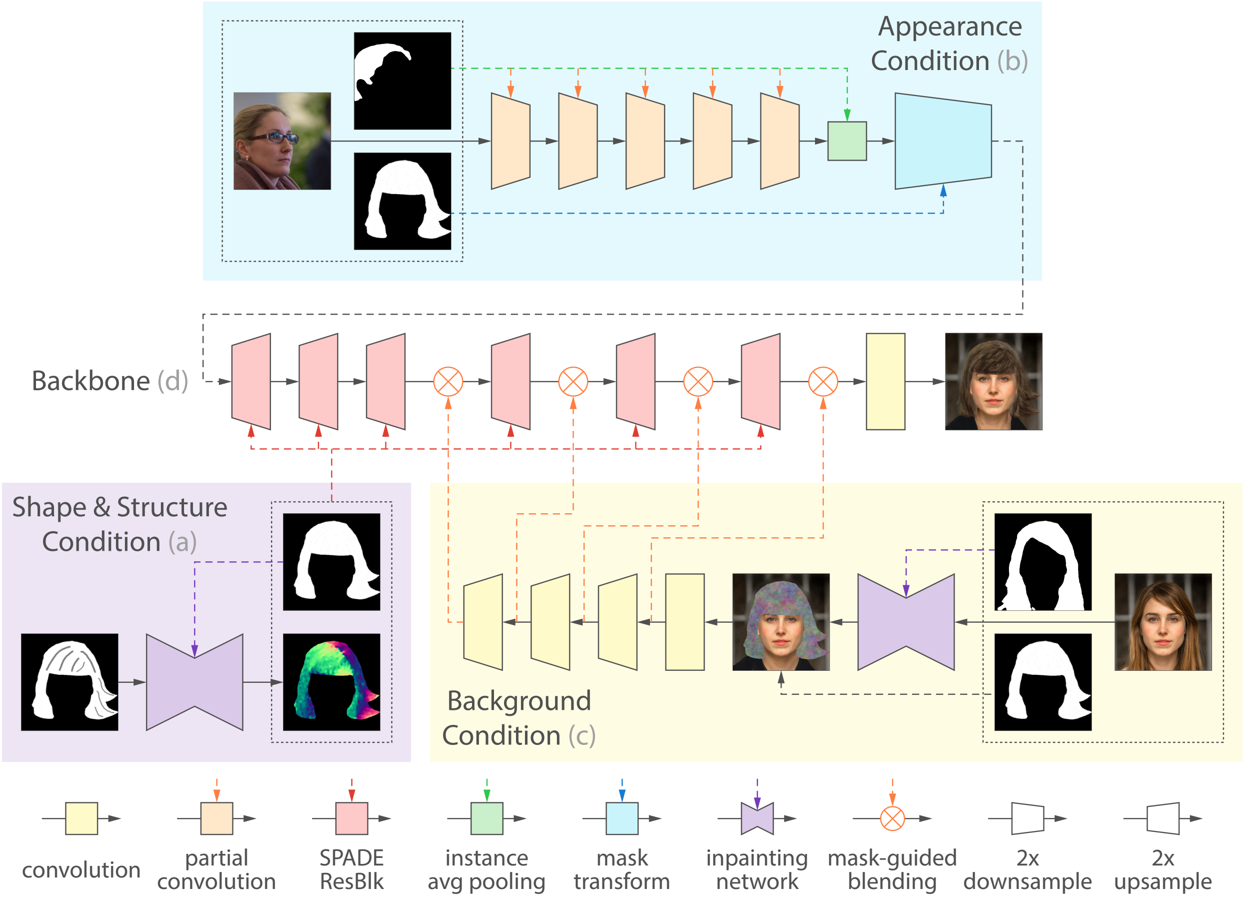

Fig. 2. **The overall pipeline of MichiGAN.** It consists of a backbone generation network (d) and three condition modules for shape & structure (a), appearance (b), and background (c).

modules to handle four types of main attributes, and also introduce the condition representation, component network design, and integration to the backbone network for each of them.

4.1 Shape & Structure

Hair shape. We represent the hair shape as the 2D binary mask of its occupied image region, similar to those image generation methods conditioned by semantic label maps [Park et al. 2019].

However, unlike other types of objects, the visual translucency and complexity of hair boundaries make binary mask inevitably a coarse approximation of the shape. To robustly generate realistic shape details and natural blending effects from rough masks either estimated with semantic segmentation networks or interactively painted by users, we relax the strictness of the condition by adopting fuzzy mask conditioning, to allow a certain level of flexibility within the boundary area. To achieve so, we dilate or erode the mask by a random width for each data pair during training, to decouple the boundary details with the accurate shape contours to some extent.

Hair structure. Regarding the structure of the target object, despite its generality, a mask shape is usually too sparse and ambiguous to condition the generation, which is particularly true for hair. But luckily, hair is such a special object with strong anisotropy and homogeneity that you can always depict the internal structure of hair with the strand flows uniformly distributed within the region. Therefore, we use a dense orientation map to represent the hair structure, which defines the 2D orientation angle for each pixel.

We use the oriented filter to estimate confident pixel-wise orientation maps, which has been widely used in hair reconstruction [Paris et al. 2008] and rendering [Wei et al. 2018]. Given a bank of oriented filter kernels $\{K_\theta\}$ (we use 32 Gabor filters in our implementation), with $\theta \in [0, \pi)$, we estimate the orientation label map o' of image I, and its associated confidence w, at pixel position p as:

$$o'_p = \arg\max_\theta \left|(K_\theta \otimes I)_p\right|, \tag{2}$$

$$w_p = \max_\theta \left|(K_\theta \otimes I)_p\right|, \tag{3}$$

where $\otimes$ represents the convolution operator.

Continuous rotation representation is non-trivial for neural networks [Zhou et al. 2019]. The orientation label map o' contains discrete angle values in $[0, \pi)$, which are actually non-continuous since orientation is cyclic, i.e., orientation 0 equals to π. Thus, direct use of the orientation label map will introduce ambiguity issues.

We convert the label map o' to two-channel continuous orientation map O' for each pixel p as follows:

$$O'_p = [\cos(2 \cdot o'_p), \sin(2 \cdot o'_p)], \tag{4}$$

and do the backward conversion as:

$$o'_p = \arctan(O'_p)/2. \tag{5}$$

We use the orientation map in two ways during both network training and inference to achieve structure control:

- **Condition input.** We use a dense orientation map as the input to the shape & structure condition module. However, the raw output O' from orientated filters may suffer from inaccurate results due to the existence of noise and dark regions. Therefore, we calculate the final dense orientation map O by applying one pass of Gaussian smoothing to O', weighted locally by confidence w.
- **Training supervision.** To explicitly enforce supervision on the structure of the result, we also propose a novel structural loss during training. Basically, we formulate the orientation estimation steps as one differentiable layer in the network following Eq. (2,3) after output I_{edit}, and measure per-pixel L1 orientation distance in the hair region weighted by confidence w. The detailed formulation is discussed in Sec. 5.2.

We generate condition input of both shape and structure by stacking M and O together. Regarding the spatial-variant natures of them, we adopt spatially-adaptive normalization [Park et al. 2019] to achieve conditioning, as shown at Fig. 2 (a).

4.2 Appearance

Different from shape or structure, hair appearance describes the globally-consistent color style that is invariant to specific shape or structure. It includes multiple factors such as the intrinsic albedo color, environment-related shading variations, and even the granularity of wisp styles. Due to its global consistency, appearance can be compactly represented. Given the complexity of natural hair styles, instead of generating the appearance features from scratch, we address hair appearance as a style transfer problem, which extracts style from the reference image and transfers it to the target.

To force disentanglement of hair appearance, we design the appearance condition module such that it operates globally and is not susceptible to local shape and structure. Furthermore, it should be mask-invariant to be able to generalize to any hair mask at test time.

The detailed architecture of our appearance condition network is shown in Fig. 2 (b). To this end, we have the following key ideas that are critical to the result quality:

- To ensure that the encoded appearance features focus only on the hair region and are not influenced by background information, we adopt partial convolution [Liu et al. 2018] with the hair mask M_{ref} of the reference image;
- In order to capture the global hair appearance and discard spatial-variant information as much as possible, we add an instance-wise average pooling layer to the output of the last partial convolution, within the reference hair region M_{ref};
- To align the appearance code with the target hair shape, and uniformly distribute the extracted appearance feature, we spatially duplicate the reference hair appearance feature to every pixel within the target hair shape M to produce the appearance map A.

Therefore, the final appearance map A is calculated as:

$$A = \frac{\sum (A'_{ref} * M_{ref})}{\sum M_{ref}} \odot M, \tag{6}$$

where A'_{ref} is the output of the last partial convolution, $*$ denotes element-wise multiplication, and $\odot$ denotes the duplication operator.

The output of our mask-transformed feature extracting network, the appearance map A, is used as the latent code input to the very beginning of the backbone generation network to bootstrap the generator with specific appearance styles instead of random noises.

4.3 Background

Finally, we need to keep all non-hair pixels (namely the background) untouched, so that we can restrict the modification to the target hair region M only. This is not as trivial as it appears [Bau et al. 2019], since a) it wastes lots of network capability and greatly limits the foreground generation flexibility, if we force the GAN to reconstruct the same background content; and b) it introduces severe blending artifacts, if we only generate the foreground region and then blend it with the original background as a post-processing step.

To pass necessary background information to the generation network in a gentle way that it can easily reconstruct the background pixels and won't be distracted by the additional constraints, we design our background condition module to be an encoder network placed parallel to the last few layers in the backbone network, as shown at Fig. 2 (c). The encoder tries to produce multi-level feature maps that could be easily reused by the generator to reconstruct the background. The encoded feature after each convolution layer is designed to have the same dimension as the input to the corresponding layer in the backbone, so that it could be merged with the input feature to feed the generator progressively.

We perform feature merging in a mask-guided way that it keeps original generator features in the foreground hair region, but replace the background with the output of the encoder. The input feature map to the ith last layer in the backbone F_i is blended between the output of the $i+1$th last layer in the backbone F_i^g, and the output of the ith layer in the condition module F_i^b as:

$$F_i = F_i^g * M + F_i^b * (1 - M). \tag{7}$$

The input to the background encoder I_{back} is the original background region. We mask out the foreground region by blending the original input I_{tag} with random noise pattern N:

$$I_{back} = N * M' + \mathcal{I}(I * M_{in}, M_{in} - M_{in} \cap M) * (1 - M'). \tag{8}$$

Here M_{in} represents the hair mask of the input image I. M' indicates the dilated version of the target mask M, in order to get rid of remaining hair information in the background due to inaccurate segmentation or stray hair strands. This also implicitly asks the network to refine the thin area $M' - M$ to achieve better boundary blending quality. $\mathcal{I}(I, M)$ is the background image inpainter [Liu et al. 2018] that fills the hole mask M with image content from I, in case the edit mask M is smaller than the original hair mask M_{in}.

5 BACKBONE GENERATOR

After introducing all our condition modules, we now present the backbone generation network that integrates all condition inputs to produce the final results, and its training strategy.

5.1 Architecture

Given conditional information obtained through the aforementioned condition modules, the generation network, serves as the backbone data flow of the system, aims at producing desired portrait image output that meets all the conditions. According to their particular perceptual characteristics and scales, the outputs of three condition modules are treated in various ways.

As shown in Fig. 2 (d), our generator is a sequential concatenation of six up-sampling SPADE residual blocks (ResBlk) [Park et al. 2019] and a convolutional layer that output the final generation result. As a brief recap of three types of condition inputs:

- **The appearance module** inputs a feature map A to the very first SPADE ResBlk, replacing randomly sampled latent code in traditional GANs;
- **The shape & structure module** feeds every SPADE ResBlk with stacked hair mask M and orientation map O through the conditional input to modulate the layer activations;
- **The background module** progressively blends background features into the outputs of the last SPADE ResBlks.

5.2 Training Strategy

Considering that we do not have the ground truth images when the conditional inputs come from different sources, training the proposed MichiGAN is essentially an unsupervised learning problem. However, most existing unsupervised learning algorithms [Huang et al. 2018; Liu et al. 2019; Zhu et al. 2017] are usually performing worse than supervised learning regarding the visual quality of the generated results. And more importantly, these unsupervised learning methods are targeted at image-wise translation between different domains, making them less feasible for our specific editing tasks with pixel-wise constraints. Motivated by [He et al. 2018] which faces a similar problem on image colorization, we propose to train our MichiGAN in a pseudo-supervised way. Specifically, we feed the four conditional inputs extracted from the same source image into MichiGAN and enforce it to recover the original image with explicit supervisions during training, and generalize it to arbitrary combinations of conditions from various sources in inference. Thanks to our hair attribute disentanglement and the powerful learning capacity, MichiGAN gains the generalization ability by leveraging information from all attributes during the supervised training, since ignoring any of them will incur larger losses.

To preprocess the training data, for each training portrait image I, we generate its hair mask M with a semantic segmentation network and its orientation map O with the dense orientation estimation method presented in Sec. 4.1. Eventually, we have training pairs with the ground-truth portrait image I, and the network output $I' = \mathcal{G}(M, O, I, I)$ that is reconstructed from the conditions.

We adopt the following loss terms to train the network:

- **Chromatic loss.** The generated result should be as close as possible to the ground-truth, but strong pixel reconstruction supervision could also harm the generalization ability of the network. Therefore, instead of measuring full-color reconstruction error, we convert the images into the CIELAB color space and only measure the chromatic distance in a and b channels, to relax the supervision but still penalize color bias:

$$L_c = \left\| \mathrm{Lab}(I')_{\mathrm{ab}} - \mathrm{Lab}(I)_{\mathrm{ab}} \right\|_1 \tag{9}$$

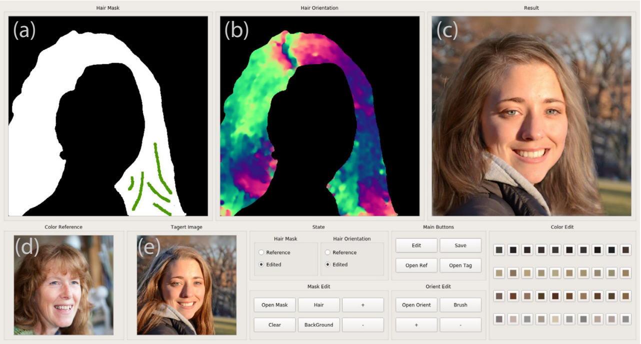

Fig. 3. **Our interactive hair editing system.** (e) shows the original input portrait; (a) is the current shape mask and painting interaction area; (b) is the visualization of current structure orientation map; (d) is the reference appearance photo selected by the user; and (c) is the generated result.

- **Structural loss.** We propose an additional structure loss to enforce the structural supervision, leveraging our differentiable orientation estimation layer. As described in Sec. 4.1, the orientation estimation layer outputs both orientation O' and confidence w', and the raw orientation output may not be accurate at every pixel. Therefore we formulate the loss as the weighted sum of orientation differences:

$$L_s = \left\| (O' - O) * w' * M \right\|_1 ; \tag{10}$$

- **Perceptual loss.** We also measure high-level feature loss with the pre-trained VGG19 model [Simonyan and Zisserman 2015], which is widely used in previous image translation problems[Chen et al. 2017a,b; Liao et al. 2017]:

$$L_p = \left\| \mathrm{VGG19}(I') - \mathrm{VGG19}(I) \right\|_1 ; \tag{11}$$

- **Adversarial loss.** A two-scale PatchGAN [Isola et al. 2017; Wang et al. 2018] discriminator $\mathcal{D}$ is used to match distributions between generated results and real portraits to enforce natural visual quality and boost local details;
- **Feature matching loss.** To achieve more robust training of GAN, we also adopt the discriminator feature matching loss [Wang et al. 2018].

In summary, the overall training objective can be formulated as:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} (\lambda_c L_c + \lambda_s L_s + \lambda_p L_p + \lambda_a L_a + \lambda_f L_f). \tag{12}$$

6 INTERACTIVE EDITING SYSTEM

Based on the proposed MichiGAN, we also build an interactive portrait hair editing system. Fig. 3 shows one screenshot of the user interface of the editing system. Within this system, users can control the shape (top left), structure (top middle), and the appearance (bottom left) separately or jointly, and see the result at the top right.

To facilitate user control over multiple approaches, we further propose two types of interaction modes, namely reference mode and painting mode.

Reference mode. One direct motivation for a user to manipulate the hair in a portrait, is to try hair attributes from another reference photo onto the target one. So we provide support to such an intuitive interaction approach by allowing the user to offer reference portraits for different attributes individually or jointly, and extracting network-understandable condition inputs to guide the generation.

Specifically, users are allowed to specify up to three references for shape (I_{shape}), structure (I_{str}), and appearance (I_{ref}), respectively. The system then calculates hair mask M from I_{shape}, orientation map O from I_{str}, and uses I_{ref} to extract the appearance features. These inputs are used to drive the generation of results.

When the target mask and orientation map come from different sources, the shape of the orientation may not fully cover the entire target shape. Therefore, we train an orientation inpainting network similar to the background one (Sec. 4.3) to complete the missing holes in the orientation map.

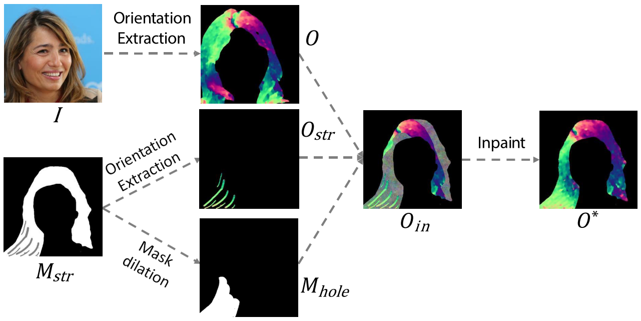

Fig. 4. **Guided orientation inpainting.** Given the current orientation map O extracted from the input image I and user strokes M_{str}, we get local hole mask M_{hole} by dilating M_{str} and calculate the guiding orientation O_{str} from M_{str}. By merging O, O_{str} and noise pattern in $M_{hole} - M_{str}$ together, we obtain the input O_{in} to the inpainting network, which eventually outputs the new orientation map O^* to update the generation result.

Painting mode. It is also natural to let a user perform local and detailed editing to the image with a much higher degree of flexibility than the reference mode. To achieve so, we provide a stroke-based interaction interface to enable local modification to both shape and structure, and a color palette tool to control the hair appearance. Their underlying methods are introduced below:

- **Shape painter.** Users can change the binary hair mask with shape painter to augment or remove certain mask regions at will, similar to the interface of [Park et al. 2019]. The smooth boundaries of these edited regions may not match true hair boundary quality, but our fuzzy mask augmentation (Sec. 4.1) could still nicely capture these details.
- **Structure painter.** Structure interaction is a way more challenging than the shape, since orientation is dense and not intuitively understandable. Regarding that, we propose a guided structure editing method to enable user-friendly manipulation. Given the current orientation map O and a set of user guidance strokes M_{str}, within a certain local region M_{hole} around the strokes, we synthesize new orientation information, which should be compatible with both the stroke guidance and the outside regions. The hole mask M_{hole} is generated by dilating the thin mask of user strokes with a certain radius. And the guidance orientation O_{str} with the stroke is calculated with the known painting orders of line segments that form the stroke paths. We compose all the information into a partial orientation map O_{in} as:

$$O_{in} = O * (1 - M_{hole}) + O_{str} * M_{str} + N * (M_{hole} - M_{str}), \quad (13)$$

  where N represents a random noise pattern to hide the original orientation. We use an orientation inpainting network to synthesize new orientation inside the hole region M_{hole}, which is trained on paired data of ground-truth orientation maps and synthesized random user strokes traced on them.
- **Appearance picker.** Finding a specific good appearance reference could be a headache for users. In painting mode, we design a palette-like appearance picker tool to retrieve, cluster, and navigate references by annotating a single RGB hair color. This is enabled by embedding all our reference portraits (selected from our test dataset) into the color space by calculating the average hair color. Given an arbitrary target color, we do KNN search around it to find K closest references for the user to test. And once the user picks a reference, the system will instantly update the candidate list to contain references clustered around the selected one.

7 EXPERIMENTS

7.1 Implementation Details

Datasets. To generate data pairs for supervised training, we use the large-scale portrait dataset of Flickr-Faces-HQ (FFHQ) [Karras et al. 2019]. The whole dataset is divided into two parts: 56000 images for training and 14000 images for testing. The resolution of images is resized to 512×512.

Network Structures. We illustrate the structures of MichiGAN in Fig. 2. Specifically, the appearance module uses five consecutive downsampling partial convolutions followed by an instance average pooling to get the appearance vector. All these partial convolutions are masked by the appearance reference, with the same kernel size of 3×3 and increasing feature channels of 64/128/256/512/1024. Each partial convolution is followed by an instance normalization and a leaky ReLu activation. A mask transform block is then used to spatially duplicate the appearance feature vector to an appearance map (size 8×8), guided by the target hair mask. The shape and structure modules follow the same modulation networks in SPADE to denormalize each SPADE ResBlk in the backbone with target hair mask and orientation (after inpainting). The background module generates the background input via inpainting and noise-filling, and produces 4 scales of background features using a network that consists of four 3×3 convolution layers with feature channels of 64/128/256/512. These features are blended into the backbone guided with the background mask. Finally, the backbone generator accepts the appearance map as input, and goes through six upsampling SPADE ResBlks and one final convolution to generate the result. The feature

Method	pix2pixHD	SPADE	MaskGAN	MichiGAN	Baseline-NCGA	Baseline-NoS	Baseline-FB	MichiGAN
Resolution	512×512				256×256			
FID	21.076	21.627	49.596	10.697	31.637	31.530	30.842	21.024

Table 1. **Frecet inception distance (FID) for different generators and settings.** We use 5,000 images randomly selected from the testing set.

channels of these layers are 1024/1024/1024/512/256/128/64. The inpainting networks for both orientation and background follow the structure proposed by [Liu et al. 2018].

Training Parameters. We train all condition modules and the backbone generator jointly. The Adam optimizer [Kingma and Ba 2015] is used with a batch size of 8 and the total epoch number of 50. The learning rates for the generator and discriminator are set to 0.0001 and 0.0004, respectively. The structural loss weight λ_s is 10, while all other loss weights, λ_c, λ_p, λ_a, and λ_f are simply set to 1 (Sec. 5.2). To obtain the dilated target mask M' (Sec. 4.3), M is randomly expanded by $[0.03, 0.07]$ of the image width.

7.2 Comparison with SOTA

Semantic Image Generation. We compare MichiGAN with two semantic image synthesis baselines: pix2pixHD [Wang et al. 2018] and SPADE [Park et al. 2019]. Pix2pixHD only supports generation based on semantic masks. And SPADE allows user control over both the semantic mask and the reference style image, which is the current state-of-the-art GAN-based conditional image synthesis framework. We train both pix2pixHD and SPADE models using the implementations provided by their authors, and exactly the same data as we used to train MichiGAN. For a fair comparison, the hair mask of the original image is converted to the input semantic mask while the reference image is set to be the original input. Besides, since pix2pixHD and SPADE can not maintain the background unchanged, we blend their generated hair with the original background with the soft hair mask. In this reconstruction setting, the original image can be regarded as the ground truth of the desired result, since all the input conditions are derived from it.

Quantitatively, MichiGAN outperforms the competing methods by a large margin, as shown in Tab. 1. Our method achieves an FID score of 10.7, which is less than half of pix2pixHD and SPADE. It indicates the edited portrait images by our method have much better realism than these two baseline methods.

Qualitatively, overall, all the methods can produce plausible hair results while our results have the best visual quality. A set of visual comparison results are shown in Fig. 5. Taking a closer look at the details of synthesized hair, we can find that the results generated by pix2pixHD and SPADE models are coarse-grained and have some apparent artifacts, while our results are more fine-grained with the rich and delicate texture details.

In terms of controllability, MichiGAN performs much better than pix2pixHD and SPADE. Pix2pixHD can only generate hair that looks adaptive to the input hair mask but with uncontrollable color and structure. SPADE can also synthesize hair adaptive to the semantic mask. Compared to pix2pixHD, it has additional ability to transfer the style of the reference image to the synthesized hair. However, the transferred results look apparently different from the desired ones, even using the original image as the reference, which is unsatisfactory. Besides, SPADE can not control the structure of the synthesized hair either. Compared to pix2pixHD and SPADE, MichiGAN can not only produce hair adaptive to the semantic mask, but also precisely control the different attributes of the hair, including appearance and structure, so that we can produce desired hair that looks most similar to the references.

In addition, the hair produced by pix2pixHD and SPADE is not compatible with the background of the original image. Despite that the results are softly blended with the backgrounds using high-quality hair masks, we can still easily see the stitching artifacts around the hair boundaries, especially for some complicated backgrounds, which introduces difficulties when applying pix2pixHD and SPADE to interactive editing. In contrast, MichiGAN does not suffer from this problem, since we take the background of the original image as one of the input conditions to directly generate the entire image with both the desired hair and the fixed background.

Conditional Face Generation. We also compare MichiGAN with two recent conditional face generation methods: MaskGAN [Lee et al. 2019] and SC-FEGAN [Jo and Park 2019].

MaskGAN uses fine-grain facial semantic masks as the representation to manipulate face images and is able to transfer styles between different subjects, which is similar to SPADE [Park et al. 2019]. Therefore we conduct the comparison with MaskGAN in the same mask-conditioned generation experiments, as shown in Fig. 5 and Tab. 1. As can be seen from both qualitative and quantitative results, although MaskGAN can synthesize visually plausible hair, our method still achieves better visual quality while being able to preserve or control the structural details.

SC-FEGAN adopts free-form sketches for portrait generation in the target hole regions to control both structure and appearance, which is comparable to our stroke-based hair generation. For a fair comparison, we adopt the hair mask as the target hole region, and use a same set of user strokes painted with our interactive system. The strokes are converted to the sketch input to SC-FEGAN, together with the corresponding hair color samples. A set of visual results are shown in Fig. 6, which contains comparisons on two target images with different sketches and appearance references. As demonstrated by, although SC-FEGAN allows control over hair structure and color to some extent, the generated results are less realistic comparing with ours. This may be partially caused by the gap between synthetic sketches generated by an edge detector and real user strokes. Our method avoids such a gap by using the dense orientation map as a unified representation for the hair structure.

7.3 Qualitative Results

Our interactive hair editing system based on MichiGAN supports flexible manipulation of hair attributes, including shape, structure, and appearance, individually or jointly. These attributes can be

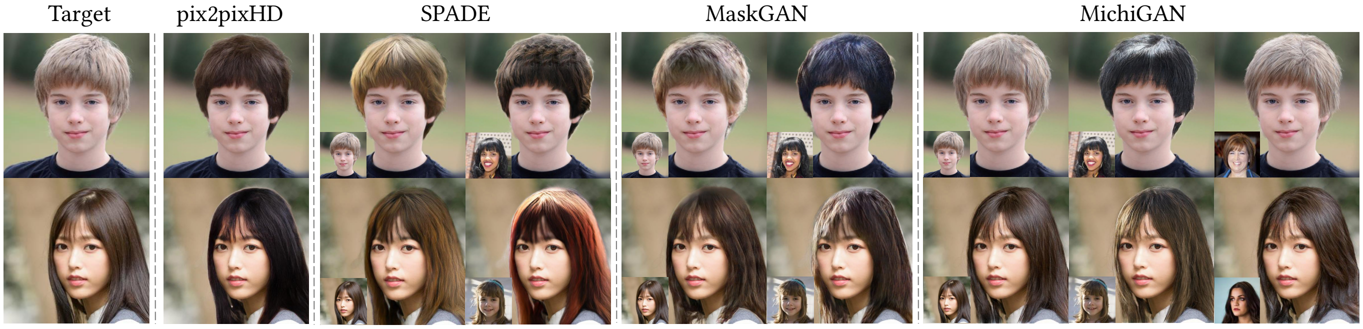

Fig. 5. **Visual comparison with pix2pixHD, SPADE, and MaskGAN.** Here, the hair mask of the original image is used as the input semantic mask for all methods. For SPADE and MaskGAN, a reference image is used to embed the appearance of the hair, which is shown in the left-bottom corner of the result image. For the proposed MichiGAN, we use the same appearance reference image as in SPADE/MaskGAN and take the original image as the orientation reference. In the last column, we also show the example of changing the structure of the hair using MichiGAN with another orientation reference image. Compared to these baselines, MichiGAN can produce reliable conditional results with higher visual quality for different condition inputs.

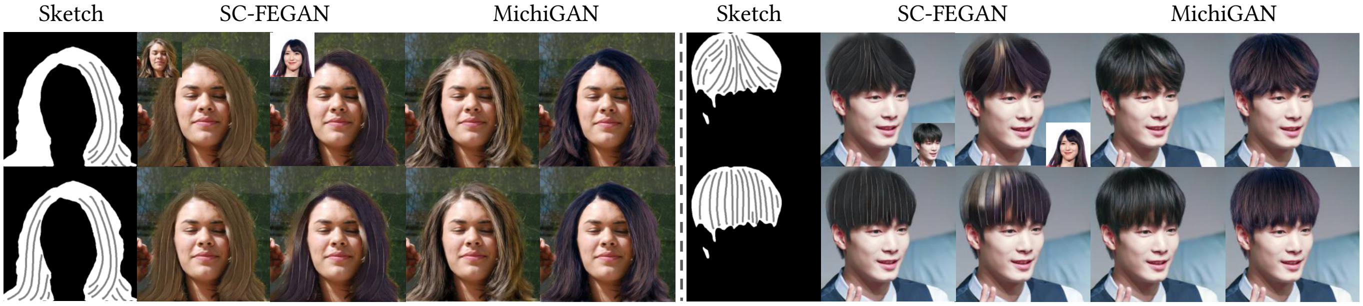

Fig. 6. **Visual comparison with SC-FEGAN.** We use two different user sketches to generate the hair for each target. Two results are generated for each sketch with different appearance references shown in the insets of the SC-FEGAN results, including the image itself (left) and another style reference (right).

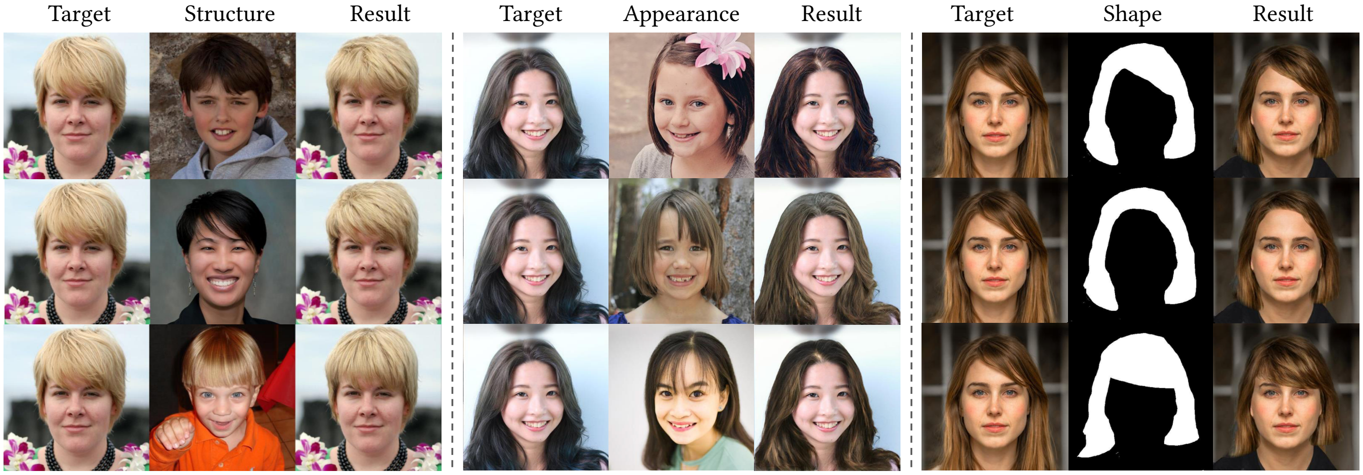

Fig. 7. **Reference-based hair editing results with a single attribute adjusted.** It shows that our MichiGAN disentangles these visual attributes well without the interference of each other, thus enabling editing each attribute individually.

conveniently specified in two ways, either by giving a reference image or interactively modifying original ones with painters, which corresponds to two modes of the system: reference-based editing and painting-based editing.

Reference-based results. We first show our reference-based hair editing results on in-the-wild portrait images. In this mode, the user first loads an original image into our system and then can simply adjust an attribute by providing another reference image. For example, once a reference image is given for the structure attribute, the orientation map will be extracted automatically by our system, and an image with a new hair structure will be generated based on the orientation condition. It is the same for appearance and shape. Figure 7 show the results with each attribute adjusted individually, while Figure 8 shows the results with multiple attributes adjusted jointly. It can be observed that MichiGAN disentangle these visual hair attributes well, without the interference of each other. And all these attributes are naturally embedded in generating a new image.

Painting-based results. In this part, we show painting-based hair editing results, where the attributes are modified by the user with an interactive painting interface (Fig. 3). Compared to the reference-based mode, this painting-based mode is more natural and flexible for the user to perform some local and detailed edits. Two examples are given in Fig. 10. The user first uses a brush tool to modify the mask image, thus the shape of the hair is changed accordingly. Then several strokes are added to indicate the orientation of hair, and a new structure of hair is generated to follow the orientations. Finally, the user changes the appearance of the hair by specifying some fancy colors. Please refer to the accompanying video for a demonstration. Since our interface is user-friendly and the generated result is intermediately fed back to the user, editing hair with our system is quite convenient. The user study shows that 15-30 seconds on average is spent for an amateur to edit one example.

Hairstyle Transfer Validation. We also validate the realism of our results by transferring the hairstyle from one image to another on a same subject. As shown in Fig. 9, we collect two images with different hairstyles of a same person, and use one image as the source of hair conditions, and the other one as the target to generate a hairstyle transfer result. Our method achieves realistic results with both appearance and structure similar to the ground truth photo.

7.4 User Study

To further evaluate the realism to real human perception, a simple user study is conducted on Amazon Mechanical Turk. We prepare two image sets, each has 200 images. One contains real portrait photos, and the other contains results generated by our method with condition inputs from randomly selected test images. During the study, we mix both sets and randomly show one image at a time to a user, and ask the user to subjectively judge whether this image is real or fake. Each image has been evaluated by 3 users. Each user spends 3 seconds on each image on average. The final fooling rates of our results and real images are 64.67% and 81.10%, respectively.

7.5 Ablation Study

In order to achieve plausible results, we dedicatedly design the proposed framework with several key ingredients, ranging from the network architecture design to the newly introduced objective functions. But considering the space limitation and training cost, only three key ablation analysis experiments are conducted, and the image resolution is set to 256 × 256.

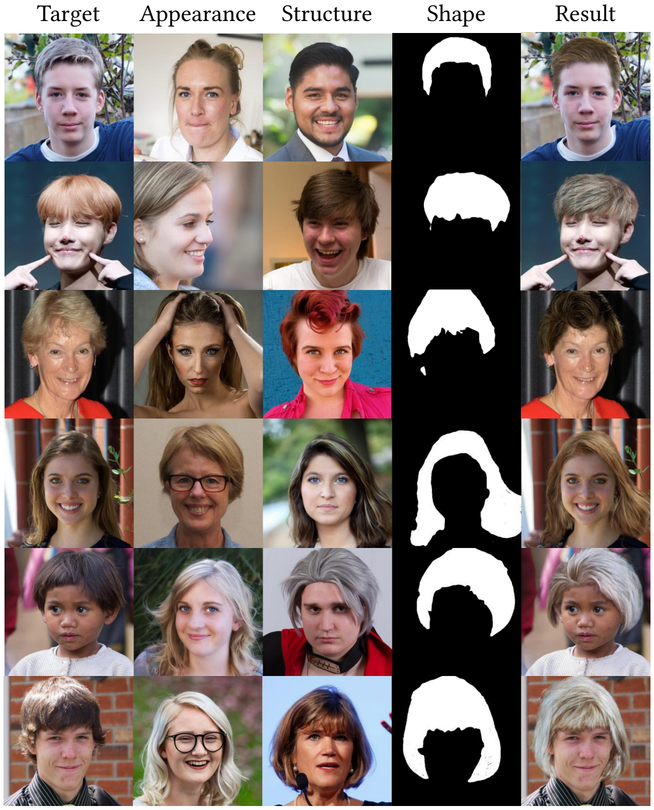

Fig. 8. **Reference-based hair editing results with multiple attributes adjusted jointly.** By disentangling appearance, shape, structure and background, our MichiGAN enables editing multiple attributes together and each attribute is from one specific reference image.

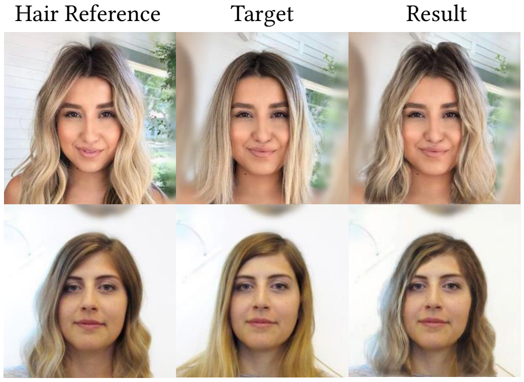

Fig. 9. **Hairstyle Transfer Validation.** We collect two photos of a same person with different hairstyles (left and middle). The result (right) is generated by using one image (left) as the reference that provides hair conditions, and the other one (middle) as the target that provides the background.

Partial Convolution and Instance-wise Average Pooling. When adopting the reference image as the appearance guidance, we only want the appearance condition module to absorb the appearance

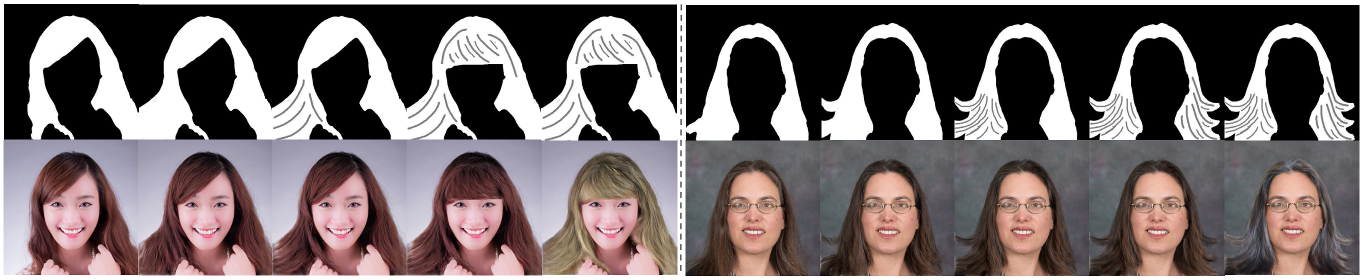

Fig. 10. **Hair editing results with our interactive system.** The users use a brush tool to modify the hair mask to change the shape, paint orientation strokes to modify the hair structure, or change the appearance of the hair by picking an alternative color.

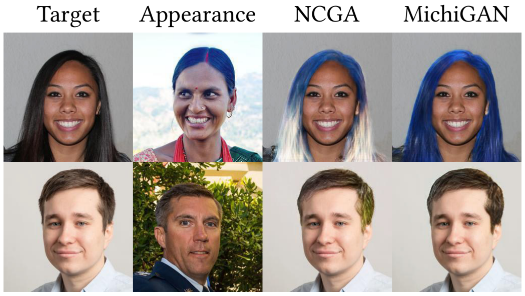

Fig. 11. **Comparisons with the baseline model using normal convolution and global average pooling.** Baseline-NCGA tend to absorb the background color into the appearance features while our method does not.

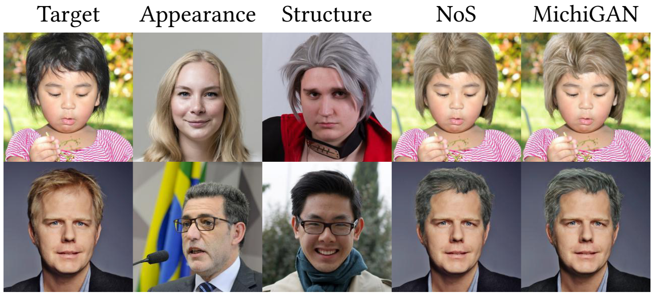

Fig. 12. **Comparisons with the baseline model without structural loss.** It can be seen that Baseline-NoS cannot adopt the reference hair orientation in the final generated target hair.

information from the hair region but not the background region. But this is non-trivial because most existing operators in modern neural networks do not naturally support it. For example, the normal convolutional layer processes the pixels within a regular grid for all the possible sliding windows. Considering the hair region is often of an irregular shape, directly applying these normal operators in the appearance module will inevitably introduce the background information into the encoded appearance features. Based on this observation, we replace all the normal convolutional layers and average pooling layers with partial convolutional layers [Liu et al. 2018] and instance-wise average pooling [Wang et al. 2018] respectively.

To show its necessity, we conduct an ablation experiment that adopts normal convolutional and global average pooling layers, denoted as "Baseline-NCGA". Obviously, from Fig. 11, we can find that the generated hair of the baseline model has significantly introduced the background color and becomes very unrealistic. Specifically, in these two cases, the target hair is polluted by the white color from the background sky and the green color from the background tree, respectively. By contrast, our method only adopts the reference hair color in the final generated results. As shown in Tab. 1, our method also achieves a better FID score than Baseline-NCGA.

Necessity of Structural Loss. To enable structural hair control, besides the dense orientation map as the condition input, we also leverage one differentiable orientation layer in the network and add an extra structural loss as training supervision. In this ablation experiment, we are curious about whether the structural loss is necessary. To verify it, we still feed the orientation map into the network but ignore the structural loss term in the training objective. As shown in Tab. 1, our method performs better than this baseline in terms of the FID score. In Fig. 12, we provide two typical examples where both the appearance and orientation guidance are given by a reference image. It can be seen that, though the baseline method can adopt the appearance from one reference image, it cannot enable the structure control at all. Therefore the hair orientation in its results is very orderless and makes the overall result not realistic enough. From this experiment, two conclusions can be drawn: 1) Hair structure control is not only a feature but also a key component for realistic hair generation; 2) Without explicit structural supervision, it is difficult for the generative network to generate sensible hair orientation, let alone keeping the reference orientation.

Mask-Guided Feature Blending. Though the main goal of our method is foreground hair editing, how to keep the original background unchanged and blend it with edited hair without artifacts is still very crucial. In order to address this problem, we choose to adopt a background condition module and fuse the background feature with the foreground feature in a mask-guided way in this paper. To show the necessity of the mask guidance, we further compare our method with the naive baseline that directly blends the foreground and background features with the mask guidance. In Fig.13, two representative examples are provided. Obviously, this naive blending way will introduce the background feature into the foreground region, and cannot preserve the reference appearance and orientation well. The better FID score shown in Tab. 1 further verifies the necessity of our design.

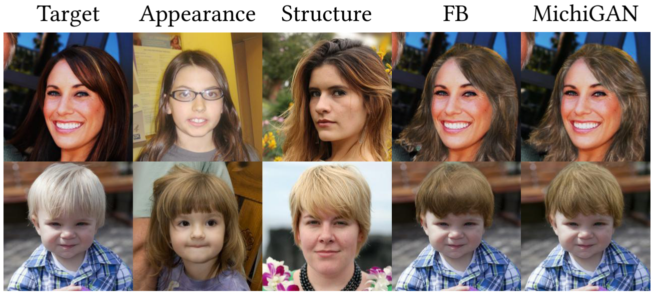

Fig. 13. **Comparisons with the baseline that uniformly blends the foreground and background feature without the mask guidance.** Compared to our method, Baseline-FB cannot preserve the reference appearance and orientation well due to the interference of the background.

8 CONCLUSION

To conclude, we present MichiGAN, a conditional hair generation network for portrait manipulation applications. Different from existing conditioned face generation methods, our method provides user control over every major hair visual factor, which is achieved by explicitly disentangling hair into four orthogonal attributes, including shape, structure, appearance, and background. For each of them, we modulate the image generation pipeline with a specially designed condition module that processes user inputs with respect to the unique nature of the corresponding hair attribute. All these condition modules are integrated with the backbone generator to form the final end-to-end network, which allows fully-conditioned hair generation from various user inputs. Based on it, we also build an interactive portrait hair editing system that enables straightforward manipulation of hair by projecting intuitive user inputs to well-defined condition representations.

Despite the superior performance of our MichiGAN, there still exist several limitations to be explored in the future (Fig. 14):

- The target shape mask, either painted or from a reference, should be aligned reasonably well with the face. The mismatched mask can result in unnatural hair shape. It would be of great help if we can automatically warp the hair shape according to different target face poses. We plan to investigate this topic in future research.
- The appearance module generally works well on hair images with a globally consistent appearance. However, spatially-varying hair appearance might be smeared out in the results.
- Even with our guided orientation inpainting, the guidance stroke should still be naturally compatible with the shape. Extreme stroke paths may cause unexpected structural issues since they can hardly be found in the synthetic training pairs.
- We randomly dilate/erode hair shape masks and shrink the background masks during training to enforce a certain ability to synthesize natural hair boundaries from binary masks. However, the results can still be less satisfactory when hair shapes are dramatically changed, since we do not explicitly handle boundary matting. In the future, we are interested in combining MichiGAN with the latest learning-based alpha-matting methods to further increase the quality.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their constructive feedback, Mingming He and Jian Ren for helpful discussions, Nvidia Research for making the Flickr-Faces-HQ (FFHQ) dataset, and all Flickr users for sharing their portrait photos under the Creative Commons License. This work was supported in part by Exploration Fund Project of University of Science and Technology of China, YD3480002001 and Hong Kong ECS grant #21209119, Hong Kong UGC.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images into the StyleGAN Latent Space?. In *ICCV 2019*. 4431–4440.

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.

Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter N. Belhumeur, and Shree K. Nayar. 2008. Face Swapping: Automatically Replacing Faces in Photographs. *ACM Trans. Graph.* 27, 3 (2008), 39.

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH 1999*. 187–194.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *ICLR 2019*.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Trans. Vis. Comput. Graph.* 20, 3 (2014), 413–425.

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-Quality Hair Modeling from A Single Portrait Photo. *ACM Trans. Graph.* 34, 6 (2015), 204:1–204:10.

Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. 2016. AutoHair: Fully Automatic Hair Modeling from A Single Image. *ACM Trans. Graph.* 35, 4 (2016), 116:1–116:12.

Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. 2013. Dynamic Hair Manipulation in Images and Videos. *ACM Trans. Graph.* 32, 4 (2013), 75:1–75:8.

Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. 2012. Single-View Hair Modeling for Portrait Manipulation. *ACM Trans. Graph.* 31, 4 (2012), 116:1–116:8.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017a. Coherent Online Video Style Transfer. In *ICCV 2017*. 1114–1123.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2017b. StyleBank: An Explicit Representation for Neural Image Style Transfer. In *CVPR 2017*. 2770–2779.

Kevin Dale, Kalyan Sunkavalli, Micah K. Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video Face Replacement. *ACM Trans. Graph.* 30, 6 (2011), 130.

Carl Doersch. 2016. Tutorial on Variational Autoencoders. *CoRR* abs/1606.05908 (2016).

Ohad Fried, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. 2016. Perspective-Aware Manipulation of Portrait Photos. *ACM Trans. Graph.* 35, 4 (2016), 128:1–128:10.

Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 2019. 3D Guided Fine-Grained Face Manipulation. In *CVPR 2019*. 9821–9830.

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 *NeurIPS 2014*. 2672–2680.

Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. 2018. Deep Exemplar-Based Colorization. *ACM Trans. Graph.* 37, 4 (2018), 47:1–47:16.

Seunghoon Hong, Xinchen Yan, Thomas S. Huang, and Honglak Lee. 2018. Learning Hierarchical Semantic Image Manipulation through Structured Representations. In *NeurIPS 2018*. 2713–2723.

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2014. Robust Hair Capture Using Simulated Examples. *ACM Trans. Graph.* 33, 4 (2014), 126:1–126:10.

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2015. Single-View Hair Modeling Using A Hairstyle Database. *ACM Trans. Graph.* 34, 4 (2015), 125:1–125:9.

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

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-Image Translation. In *ECCV 2018*, Vol. 11207. 179–196.

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

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

Neel Joshi, Wojciech Matusik, Edward H. Adelson, and David J. Kriegman. 2010. Personal Photo Enhancement Using Example Images. *ACM Trans. Graph.* 29, 2 (2010),

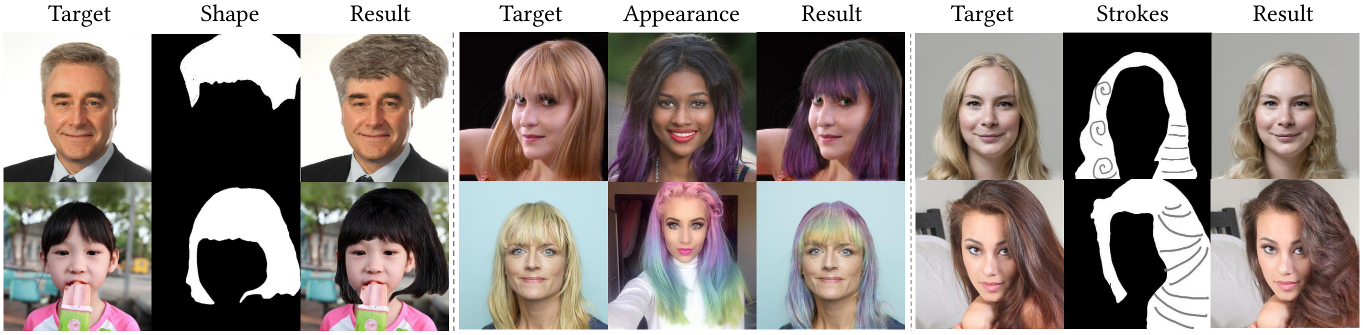

Fig. 14. **Three possible limitations of MichiGAN**: 1) we need the target mask to be aligned reasonably well with the face; 2) we cannot capture the spatially varying appearance well; 3) the painted stroke paths cannot be too extreme.

12:1–12:15.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR 2018*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR 2019*. 4401–4410.

Ira Kemelmacher-Shlizerman. 2016. Transfiguring Portraits. *ACM Trans. Graph.* 35, 4 (2016), 94:1–94:8.

Ira Kemelmacher-Shlizerman, Supasorn Suwajanakorn, and Steven M. Seitz. 2014. Illumination-Aware Age Progression. In *CVPR 2014*. 3334–3341.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2019. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. *CoRR* abs/1907.11922 (2019).

Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. 2008. Data-Driven Enhancement of Facial Attractiveness. *ACM Trans. Graph.* 27, 3 (2008), 38.

Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G. Shapiro, and Ira Kemelmacher-Shlizerman. 2018. Video to Fully Automatic 3D Hair Model. *ACM Trans. Graph.* 37, 6 (2018), 206:1–206:14.

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.

Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image Inpainting for Irregular Holes Using Partial Convolutions. In *ECCV 2018*, Vol. 11215. 89–105.

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 2019*. 10550–10559.

Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013. Structure-Aware Hair Capture. *ACM Trans. Graph.* 32, 4 (2013), 76:1–76:12.

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

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2017. Unrolled Generative Adversarial Networks. In *ICLR 2017*.

Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014).

Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. 2019. Deep Face Normalization. *ACM Trans. Graph.* 38, 6 (2019), 183:1–183:16.

Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis with Auxiliary Classifier GANs. In *ICML 2017*, Vol. 70. 2642–2651.

Sylvain Paris, Will Chang, Oleg I. Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. 2008. Hair Photobooth: Geometric and Photometric Acquisition of Real Hairstyles. *ACM Trans. Graph.* 27, 3 (2008), 30.

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

Haonan Qiu, Chuan Wang, Hang Zhu, Xiao Zhu, Jinjin Gu, and Xiaoguang Han. 2019. Two-Phase Hair Image Synthesis by Self-Enhancing Generative Model. *Comput. Graph. Forum* 38, 7 (2019), 403–412.

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative Adversarial Text to Image Synthesis. In *ICML 2016*, Vol. 48. 1060–1069.

Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D Hair Synthesis Using Volumetric Variational Autoencoders. *ACM Trans. Graph.* 37, 6 (2018), 208:1–208:12.

Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. 2017. Neural Face Editing with Intrinsic Image Disentangling. In *CVPR 2017*. 5444–5453.

Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015*.

Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. 2019. FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery. In *CVPR 2019*. 6490–6499.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder Variational Autoencoders. In *NeurIPS 2016*. 3738–3746.

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.

Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *CVPR 2016*. 2387–2395.

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 2018*. 8798–8807.

Yang Wang, Lei Zhang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. 2009. Face Relighting from A Single Image under Arbitrary Unknown Lighting Conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 11 (2009), 1968–1984.

Lingyu Wei, Liwen Hu, Vladimir G. Kim, Ersin Yumer, and Hao Li. 2018. Real-Time Hair Rendering Using Sequential Adversarial Networks. In *ECCV 2018*. 105–122.

Yanlin Weng, Lvdi Wang, Xiao Li, Menglei Chai, and Kun Zhou. 2013. Hair Interpolation for Portrait Morphing. *Comput. Graph. Forum* 32, 7 (2013), 79–84.

Fei Yang, Jue Wang, Eli Shechtman, Lubomir D. Bourdev, and Dimitris N. Metaxas. 2011. Expression Flow for 3D-Aware Face Component Transfer. *ACM Trans. Graph.* 30, 4 (2011), 60.

Shunyu Yao, Tzu-Ming Harry Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, Bill Freeman, and Josh Tenenbaum. 2018. 3D-Aware Scene Manipulation via Inverse Graphics. In *NeurIPS 2018*. 1891–1902.

Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. 2019. Semantics Disentangling for Text-to-Image Generation. In *CVPR 2019*. 2327–2336.

Han Zhang, Tao Xu, and Hongsheng Li. 2017b. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *ICCV 2017*. 5908–5916.

Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. 2017a. A Data-Driven Approach to Four-View Image-Based Hair Modeling. *ACM Trans. Graph.* 36, 4 (2017), 156:1–156:11.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the Continuity of Rotation Representations in Neural Networks. In *CVPR 2019*. 5745–5753.

Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. 2018. HairNet: Single-View Hair Reconstruction Using Convolutional Neural Networks. In *ECCV 2018*. 249–265.

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