HairCLIPv2: Unifying Hair Editing via Proxy Feature Blending

Tianyi Wei[1], Dongdong Chen[2], Wenbo Zhou[1,†], Jing Liao[3],
Weiming Zhang[1], Gang Hua[4], Nenghai Yu[1]
[1]University of Science and Technology of China [2]Microsoft Cloud AI
[3]City University of Hong Kong [4]Xi'an Jiaotong University
{bestwty@mail., welbeckz@, zhangwm@, ynh@}ustc.edu.cn
{cddlyf@, ganghua@}gmail.com, jingliao@cityu.edu.hk

Figure 1. HairCLIPv2 supports hairstyle and color editing individually or jointly with unprecedented user interaction mode support, including text, mask, sketch, reference image, etc.

Abstract

Hair editing has made tremendous progress in recent years. Early hair editing methods use well-drawn sketches or masks to specify the editing conditions. Even though they can enable very fine-grained local control, such interaction modes are inefficient for the editing conditions that can be easily specified by language descriptions or reference images. Thanks to the recent breakthrough of cross-modal models (e.g., CLIP), HairCLIP is the first work that enables hair editing based on text descriptions or reference images. However, such text-driven and reference-driven interaction modes make HairCLIP unable to support fine-grained controls specified by sketch or mask. In this paper, we propose HairCLIPv2, aiming to support all the aforementioned interactions with one unified framework. Simultaneously, it improves upon HairCLIP with better irrelevant attributes (e.g., identity, background) preservation and unseen text descriptions support. The key idea is to convert all the hair editing tasks into hair transfer tasks, with editing conditions converted into different proxies accordingly. The editing effects are added upon the input image by blending the corresponding proxy features within the hairstyle or hair color feature spaces. Besides the unprecedented user interaction mode support, quantitative and qualitative experiments demonstrate the superiority of HairCLIPv2 in terms of editing effects, irrelevant attribute preservation and visual naturalness. Our code is available at https://github.com/wty-ustc/HairCLIPv2.

1. Introduction

Hair editing as an interesting and challenging problem has attracted a lot of research attention from both academia and industry. Over the past few decades, tremendous progress [50, 41, 31, 36] has been made in this field, enabling high-fidelity hair editing based on various types of user interactions or controls. In earlier hair editing methods [50, 41], commonly supported user editing conditions are sketches and masks, which can enable fine-grained local controls. But in real scenarios, many hair editing conditions can be specified by simpler interactions, e.g., text descriptions (e.g., "bowl cut hairstyle") and reference images.

Recently, cross-modal visual and language representation learning [29, 22, 43, 8, 35, 38, 45] has made remarkable breakthrough, which makes text-guided image manipu-

† Wenbo Zhou is the corresponding author.

	HairCLIP [36]	LOHO [24]	Barbershop [47]	HairNet [48]	SYH [20]	MichiGAN [31]	SketchSalon [41]	Ours
Aligned Hair Transfer	✔	✔	✔	✔	✔	✔	✗	✔
Unaligned Hair Transfer	✔	✗	✗	✔	✔	✗	✗	✔
Text	✔	✗	✗	✗	✗	✗	✗	✔
Mask	✗	✗	✔	✗	✔	✔	✗	✔
Sketch	✗	✗	✗	✗	✗	✔	✔	✔
Local Hairstyle Editing	✗	✗	✗	✗	✗	✔	✔	✔
Local Hair Color Editing	✗	✗	✗	✗	✗	✗	✔	✔

Table 1. Comparisons between our approach and mainstream hair editing methods in terms of available interaction modes and functionality. Only our method supports all interaction modes and enables both global and local hair editing.

lation possible. HairCLIP [36] presents the first attempt that supports hair editing via text description and reference image within one unified framework. Despite such text-driven and reference-driven interaction being more efficient and user-friendly, HairCLIP cannot support fine-grained controls like sketches and masks. Moreover, HairCLIP has two other limitations: 1) Since hair editing in HairCLIP is accomplished by pure latent code manipulation, it will inevitably alter other irrelevant attributes (e.g., identity, background) because fully decoupling different attributes in latent codes is difficult; 2) It struggles in yielding satisfactory results for text descriptions that differ significantly from training texts.

In this paper, we take a step forward and propose HairCLIPv2, a unified hair editing system that unprecedentedly supports all the aforementioned interaction modes, including the natural text/reference-driven interaction and fine-grained local interaction. In Table 1, we list the interaction modes and editing functionality supported by existing hair editing methods. Moreover, with fundamentally different editing mechanism design, HairCLIPv2 makes great improvement upon HairCLIP with better irrelevant attributes preservation and unseen text description support.

The key idea of HairCLIPv2 is *converting all the hair editing tasks into hair transfer tasks, and the editing conditions are converted into different transfer proxies accordingly*. Conceptually, it can be understood as "find proxy hair images that satisfy the editing conditions and transfer the corresponding attributes to the source image". Note that we use the StyleGAN latent code or feature corresponding to such proxies rather than use the proxy images explicitly.

More specifically, we first transform the input source image into the bald proxy, which inpaints the hair-covered regions (e.g., background, ears) with reasonable semantic attributes. This can help avoid the editing artifacts caused by occlusion when blending the source image with condition proxies. For different editing proxies, we define their generation as different tasks performed in StyleGAN according to their characteristics. Depending on the users' editing preferences, hair editing effects are then enforced upon the input image by blending the corresponding proxy features within the hairstyle feature space or hair color feature space. This is different from HairCLIP that achieves the editing effect by manipulating the 1-d latent codes. Such feature blending based editing naturally supports global and local hair editing by controlling the blending area to cover the entire hair area or part of it.

To show the superiority of HairCLIPv2, we conduct extensive comparisons. In addition to more complete user interaction modes support, HairCLIPv2 also shows obvious advantages in terms of manipulation accuracy, irrelevant attribute preservation, and visual naturalness. Some interactive editing examples are provided in Figure 1. Our contributions can be summarized as below:

- We present a fresh perspective for hair editing tasks and propose a novel hair editing paradigm that unifies various types of editing into the form of proxy hair transfers. We achieve all editing effects with the feature blending mechanism, which not only alleviates the editing pressure on each proxy but also enables excellent irrelevant attribute preservation.

- We dedicately design the proxy generation for different conditions based on their own special properties, e.g., for the text proxy, the decoupled proxy design and optimization starting point selection strategy help us achieve better editing effects and arbitrary text support; for the sketch proxy, we achieve local hairstyle editing support for the first time within the StyleGAN-based framework by formalizing its generation as the image translation task and incorporating insights of semantic layering in StyleGAN.

- Our system pushes the interactions of hair editing to a new level, supporting arbitrary text, mask, reference image, sketch and their combinations, and enabling both global and local hair editing, which has never been realized before.

2. Related Work

Generative Adversarial Networks. Since being invented, GANs have made considerable progress in terms of training strategies [18, 34], loss functions [3, 4, 9], and network

structures [25, 12, 10, 30, 32]. In the field of image synthesis, a series of works called StyleGAN [17, 18, 15, 16] represents the cutting edge of GANs. Given its promising semantically decoupled latent space [7, 26] and high-quality image synthesis abilities, the pre-trained StyleGAN has become the preferred choice for performing image editing. In this paper, we choose StyleGAN2 to develop our framework, which is consistent with other hair editing methods [36, 47, 20, 21, 40, 24] to be compared.

Latent Space Embedding and Editing. As the bridge connecting the pre-trained StyleGAN and other downstream editing tasks, GAN inversion aims to yield the ideal embedding of the real image in the latent space. Based on the application purposes, we roughly classify the GAN inversion methods into two categories: methods [33, 46, 49] suitable for editing and methods [2, 23, 37] for better reconstruction. The former methods project the real image into the embedding subspace more suitable for editing at the expense of reconstruction. Among them, e4e [33] has become the most popular method for editing tasks [36, 21, 39] performed in the latent space. The latter approaches [2, 23, 37] aim to achieve the perfect reconstruction of the real image. However, limited by the representation capability of the latent space, all these methods cannot achieve the perfect reconstruction. To address this issue, Barbershop [47] proposes a novel inversion method, which additionally introduces a feature space $\mathcal{F}$ of StyleGAN combined with the latent space $\mathcal{S}$ to form a new embedding space $\mathcal{FS}$. Inspired by this, we decouple the editing task from the reconstruction task by blending editing proxy features in the feature space to achieve a unified hair editing system that supports a wide range of interactions.

Hair Editing Using GANs. Existing hair editing methods can be roughly categorized as conditional GANs [41, 31, 13] based and pre-trained StyleGAN based [36, 21, 40, 24, 47, 20, 48, 28]. As a pioneering work of hair transfer, MichiGAN [31] accomplishes hairstyle transfer by extracting the orientation map of the reference image. Barbershop [47] performs hair transfer within their proposed $\mathcal{FS}$ embedding space. But these methods often struggle when large pose differences exist between source and target image. Recently, some improvements [20, 48] on Barbershop make pose unaligned hair transfers possible. Our framework is also compatible with pose unaligned hair transfers and additionally offers more interaction modes. SketchHairSalon [41] enables local editing of hairstyle and hair color by using colored sketches as the input to the conditional translation network. Unlike them, we show for the first time that the StyleGAN-based framework can also perform local hair editing with sketches as the condition.

Benefiting from the development of cross-modal models [29, 22], text-guided hair editing [36, 21, 40] has become the new trend. StyleCLIP [21] and TediGAN [40] utilize CLIP loss to perform hair editing in an optimized manner. However, since the embedding of real image deviates from the original suitable editing latent space, these methods will fail for some cases. HairCLIP [36] alleviates the problem by training a hair mapper on a large-scale dataset, but struggles in yielding good results for descriptions that differ significantly from the training text. Moreover, none of these methods can preserve the irrelevant attributes well. In this work, we present a new perspective to enable text-guided hair editing methods. By decoupling the editing task from the reconstruction, we can better preserve the irrelevant attributes while enabling high-quality hair manipulation via arbitrary text descriptions. More importantly, there is no prior work that supports so many interaction modes and functionality as we offer.

3. Proposed Method

3.1. Preliminaries

StyleGAN [18] can synthesize photorealistic images with a progressive upsampling network consisting of 18 layers. Its $\mathcal{W} \subsetneq \mathbb{R}^{512}$ latent space exhibits good semantic decoupling properties [7, 26], thus enabling various editing tasks. To perform semantic editing while ensuring the reconstruction quality, some inversion methods [1, 23, 37] extend the original $\mathcal{W}$ space to $\mathcal{W}+$ space, which is defined as a cascade of 18 different latent codes w_i from the $\mathcal{W}$ space, i.e., $\mathcal{W}+ \subsetneq \mathbb{R}^{18\times512}$.

$\mathcal{FS}$ **Embedding Space** is proposed by Barbershop [47], which is designed to increase the representational capability of embedding space for details and enable the spatial control of image features. The new $\{F_7, S\}$ latent code replaces the first 7 layers of $\mathcal{W}+$ latent code with the $32 \times 32 \times 512$ features F_7 of style-block 7 of StyleGAN G, i.e., $F_7 \in \mathcal{F} \subsetneq \mathbb{R}^{32\times32\times512}$, $S = [w_8, \cdots w_i, \cdots w_{18}], w_i \in \mathcal{W}$.

CLIP [22] is a multi-modality model pretrained on web-scale image-text pairs. It can well measure the semantic similarity between given image and text.

3.2. Overview

Since the $\mathcal{FS}$ embedding space [47] is proposed, performing seamless feature blending in $\mathcal{F}$-space has become the de facto standard for many hair transfer works [47, 20, 48], because it can encode the spatial information and preserve local details. On the other hand, many hair editing efforts [36, 21, 40, 39] choose to perform editing in the $\mathcal{W}+$ space, despite unsatisfactory reconstruction, because it can encode rich disentangled semantics [7, 26]. Considering that $\mathcal{F}$-space is expressive and enables realistic feature integration results while $\mathcal{W}+$ space is editable, we therefore wonder "*Can we enjoy the best features of both spaces to facilitate the hair editing task?*". To achieve this goal, we formulate the hair editing tasks as the hair transfer tasks.

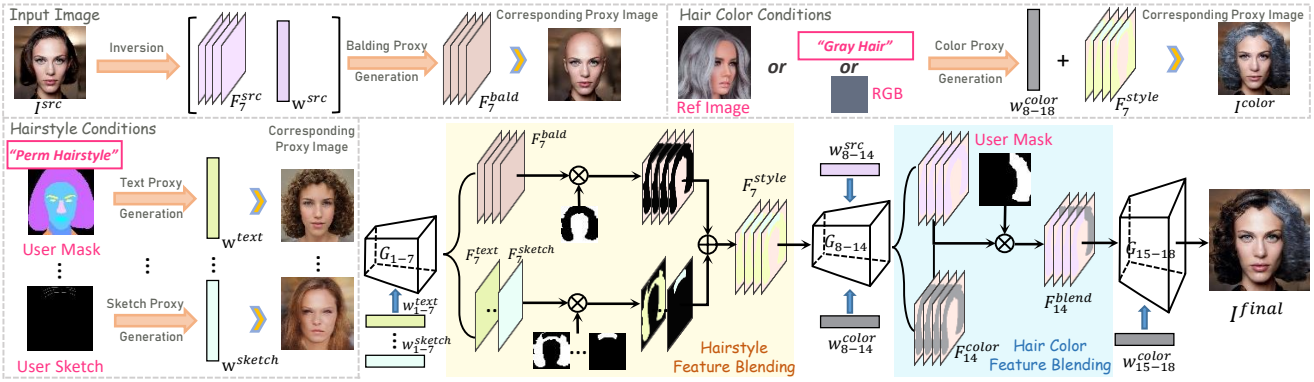

Figure 2. Overview of HairCLIPv2: Example with hairstyle description text, sketch, mask and hair color RGB values as conditional inputs. Corresponding proxy images are just for better understanding. We complete the hair editing by converting different conditions into different proxies and achieve editing effects by blending them in StyleGAN feature spaces.

Specifically, we convert all editing conditions (e.g., text, reference image, sketch) into different proxies in the $\mathcal{W}+$ space, and accomplish hair editing by seamless proxy feature blending in the feature spaces of StyleGAN. The proxy features of different conditions are obtained with tailored methods based on the condition characteristics.

Following the design in HairCLIP [36], we edit hairstyle and hair color sequentially by blending editing proxy features in the early and later StyleGAN feature space respectively. In detail, as shown in Figure 2, we choose to perform proxy feature blending of hairstyle and hair color on feature spaces $\mathcal{F}^{style} \subsetneq \mathbb{R}^{32\times32\times512}$ and $\mathcal{F}^{color} \subsetneq \mathbb{R}^{256\times256\times128}$, which correspond to the features of 7-th and 14-th style-block in StyleGAN respectively. And users can select various interactions (global or local, single or combined) to edit hairstyles and hair color individually or jointly.

3.3. Converting Input Image to Bald Proxy

Given a source image I^{src} to be edited, we obtain its latent code w^{src} in $\mathcal{W}+$ space and feature F_7^{src} in $\mathcal{FS}$ space by II2S [49] and FS embedding algorithms [47], respectively. Balding proxy is then generated to inpaint the hair-covered region with reasonable semantic attributes (e.g., background, ears, face, etc.), which can avoid editing artifacts due to occlusion when blending the original image with different editing proxies.

Balding Proxy. To remove the occlusion from the hair area of the source image, we bald it using HairMapper [39]: $w^{bald} = \mathbb{B}(w^{src})$. $\mathbb{B}$ denotes HairMapper, which completes the balding editing operation in $\mathcal{W}+$ space on the latent code w^{src} to yield the latent code w^{bald} corresponding to the balded source image. Since editing in $\mathcal{W}+$ space inevitably gets other irrelevant attributes modified, we circumvent this issue by blending bald feature with source image feature in $\mathcal{F}^{style}$ space:

$$F_7^{bald} = G(w_{1-7}^{bald}) \times M^{bald} + F_7^{src} \times (1 - M^{bald}), \quad (1)$$

where G stands for StyleGAN, $G(w_{1-7}^{bald})$ represents the bald feature in $\mathcal{F}^{style}$ space. M^{bald} is the binary mask indicating the hair and ear regions of the source image, which is obtained by the facial parsing network BiseNET [42] and downsampled to 32×32. With the guidance of M^{bald}, the irrelevant attributes region in F_7^{bald} can continue to be preserved in following proxy feature blending.

3.4. Hairstyle Editing

Below we elaborate on how to generate the proxy for different hairstyle conditions.

Text Proxy. We formalize text proxy generation as the editing task done in $\mathcal{W}+$ space based on the CLIP loss guidance. Unlike prior works [36, 21, 40], our text proxy generation process is free from the pressure of irrelevant attribute preservation, and allows us to select a more suitable starting point for the optimization process, which leads to better editing effects. In order to ensure both the optimal editing effect and the diversity of editing results, we choose to sample a random point around the mean face latent code as the optimization starting point for our text proxy latent code w^{text}. In detail, we adopt the truncation trick of StyleGAN as $w^{init} = w^{mean} + \psi(w^{random} - w^{mean})$, where w^{mean} is the mean face latent code and w^{random} is sampled randomly. By setting a small value of ψ, we ensure that the initial optimization starting point w^{init} is around the average face latent code w^{mean}. We will show the benefits of this initialization strategy in the ablation analysis. We adopt the CLIP loss with transformation augmentations [5] to perform the text-guided hairstyle editing while reducing the disturbance caused by adversarial examples:

$$L^{clip} = \frac{1}{N}\sum_{i=1}^{N}(1 - cos(E_i(A_i(G(w^{text}))), E_t(st))), \quad (2)$$

where A_i represents i-th transformation augmentation, N denotes the number of augmentations ($N = 4$ by default),

$cos(\cdot)$ means cosine similarity, $G(w^{text})$ means the editing result for each pass in the optimization process, E_i and E_t stands for the CLIP image encoder and text encoder respectively, and st refers to the user-supplied text description. Besides, pose alignment loss L^{pose} is utilized to ensure that the face shape and pose of text proxy are consistent with the source image to ease the subsequent feature blending:

$$L^{pose} = \frac{1}{N_k}||E_p(I^{src}) - E_p(G(w^{text}))||_2^2, \quad (3)$$

where E_p represents the 3D keypoint extractor [6] and N_k denotes the number of keypoints. Optionally, the shape loss L^{shape} is added to constrain the shape of the generated hair according to whether the user provides the hair region mask. We then obtain text proxy feature F_7^{text} using the optimized w^{text}: $F_7^{text} = G(w_{1-7}^{text})$.

Reference Proxy. Given a hairstyle reference image I^{sr}, we generate the reference proxy by performing the unaligned hairstyle transfer task. I^{sr} is first inverted by the II2S [49] embedding algorithm into the $\mathcal{W}+$ space to get w^{ref}, which is served as the starting point for hairstyle transfer. During the transfer process, we expect to keep the original hair structure of the reference image while ensuring its pose and facial shape to be consistent with the source image. Thus, L^{pose} is imposed between $G(w^{ref})$ and I^{src}. In addition, a style loss L^{style} based on the gram matrix [11] is used to ensure that the hairstyle structure of I^{sr} remains unchanged during the alignment process:

$$L^{style} = \frac{1}{4}\sum_{i=1}^{4}||\mathcal{G}_i(VGG_i(I^{sr} \times M^{rh})) - \mathcal{G}_i(VGG_i(G(w^{ref}) \times M^{gh}))||_2^2, \quad (4)$$

where $\mathcal{G}_i(\gamma_i)$ represents the gram matrix calculated on the i-th layer features and a total of 4 layers of features are extracted, i.e.,$\{relu1\_2, relu2\_2, relu3\_3, relu4\_3\}$ of VGG [27]. M^{rh} and M^{gh} are hair masks for the reference image and the generated image of each round during optimization process predicted by BiseNET [42]. For the same purpose, the L_2 norm of the manipulation magnitude in the latent space is utilized during optimization:

$$L^{reg} = ||w_t^{ref} - w_{t-1}^{ref}||_2^2, \quad (5)$$

where w_t^{ref} and w_{t-1}^{ref} represent the latent code of the current step and the previous step, respectively. Similar to the text proxy, L^{shape} is optionally added to allow the user to customize the shape of the hair. We then obtain reference proxy feature F_7^{ref} using the optimized w^{ref}: $F_7^{ref} = G(w_{1-7}^{ref})$.

Sketch Proxy. Enabling sketch-based local hairstyle editing within our framework is nontrivial. It is hard to find suitable losses to constrain the local hairstyle structure to conform to the sketch given by the user. To circumvent this problem, we innovatively formalize the synthesis of sketch proxy as an image translation task based on StyleGAN. Utilizing the sketch-hair dataset created by SketchHairSalon [41], we train a sketch2hair inverter T, which is based on E2Style [37] and aims to find the most appropriate latent code in $\mathcal{W}+$ space to accurately translate a given sketch to the corresponding hair structure. The training loss consists of regular pixel-level L_2 loss, feature-level LPIPS [44] loss and multi-layer face parsing loss [37] which is introduced to provide more local supervision. During the training process, we randomly remove a portion of the strokes to make our sketch2hair inverter adapt to a variety of sketch inputs from fine to coarse, e.g., even just one stroke. Given a local hairstyle sketch S, our sketch proxy features F_7^{sketch} are synthesized by pre-trained sketch2hair T and StyleGAN G:

$$w^{sketch} = T(S), \quad F_7^{sketch} = G(w_{1-7}^{sketch}). \quad (6)$$

Proxy Feature Blending. For text and reference image based hairstyle condition, we perform global blending in $\mathcal{F}^{style}$ space:

$$F_7^{global} = F_7^{tr} \times M^{global} + F_7^{bald} \times (1 - M^{global}), \quad (7)$$

where $F_7^{tr} \in \{F_7^{text}, F_7^{ref}\}$ and M^{global} is the binary mask corresponding to the hair region of F_7^{tr}. Optionally, the sketch-based local hairstyle editing is applied:

$$F_7^{style} = F_7^{sketch} \times M^{local} + F_7^{global} \times (1 - M^{local}), \quad (8)$$

where M^{local} is obtained by downsampling the user input sketch S to 32×32 after dilation. A natural concern is the artifacts brought by the mismatch between hair features within M^{local} and other hair features. But thanks to the semantic layering characteristics of StyleGAN, the resulting image shows consistent tones as these hair features will be modulated by the later layers. Our framework allows users to only edit the hairstyle: $I^{style} = G(F_7^{style}, w_{8-18}^{src})$, by skipping the following hair color editing.

3.5. Hair Color Editing

We achieve hair color editing by performing proxy feature blending in $\mathcal{F}^{color}$ space. By choosing to use the feature F_7^{src} or F_7^{style}, we allow to edit only hair color or both hair style&hair color. Below, we use F_7^{style} as the example.

Color Proxy. We initialize the color proxy with F_7^{style} and $w_{8-18}^{color} = w_{8-18}^{src}$, and set w_{10-13}^{color} to be optimizable. The loss L^{color} in the optimization process consists of L^{modal} and L^{bg}, where L^{modal} can be defined as L^{clip} or the average color L_2 loss of the hair region depending on the hair color condition types (text, reference image, RGB values), and L^{bg} is defined as follows:

$$L^{bg} = ||(I^{style} - I^{color}) \times (M^{n-hair})||_2^2, \quad (9)$$

Methods	IDS↑	PSNR↑	SSIM↑
Ours	**0.84**	**29.5**	**0.91**
HairCLIP [36]	0.45	21.6	0.74
StyleCLIP [21]	0.43	19.6	0.72
TediGAN [40]	0.16	22.5	0.74
DiffCLIP [19]	0.71	26.8	0.86

Table 2. Quantitative comparison for irrelevant attributes preservation. IDS denotes identity similarity, PSNR and SSIM are calculated at the intersected non-hair regions before and after editing.

where $I^{color} = G(F_7^{style}, w_{8-18}^{color})$, M^{n-hair} is the mask of the non-hair region intersection between I^{style} and I^{color}.

Proxy Feature Blending. Even with the L^{bg} constraint, we find non-hair regions are often inevitably modified because of imperfect semantic decoupling of the $\mathcal{W}+$ space. We solve this by performing proxy feature blending in $\mathcal{F}^{color}$ space, which also naturally supports local hair color editing:

$$F_{14}^{blend} = G(F_7^{style}, w_{8-14}^{color}) \times M^{color} + G(F_7^{style}, w_{8-14}^{src}) \times (1 - M^{color}), \quad (10)$$

where M^{color} is the hair area mask or a local editing area mask drawn by the user. We set F_{14}^{blend} and w_{15-18}^{color} to be optimizable to further perform the optimization. In the optimization process, we use the L_{blend} loss, which consists of L_2 loss and LPIPS loss to constrain I^{final} to be similar to I^{color} inside M^{color} and similar to I^{style} outside M^{color} simultaneously. The final edited image is synthesized as follows: $I^{final} = G(F_{14}^{blend}, w_{15-18}^{color})$.

4. Experiments

Implementation details of our approach are provided in the supplementary material. For all compared methods, we use their official codes or pre-trained models.

4.1. Quantitative and Qualitative Comparison

Comparison with Text-Driven Hair Editing Methods. We compare HairCLIPv2 with leading text-driven hair editing methods on the CelebA-HQ [14] testset ($2,000$ images) and follow the evaluation settings of HairCLIP. For HairCLIP [36] and StyleCLIP [21] ("Mapper" version), we first invert using e4e [33] to obtain the latent code for a given real image before performing the editing. For DiffusionCLIP [19], we finetune a model for each text description. For both TediGAN [40] and our method, the number of optimization iterations is set to 200. As shown in Figure 3 and Table 2, our method accomplishes satisfactory hair editing effects with better naturalness while maximizing the preservation of irrelevant attributes. It is worth noting that, even though HairCLIP and StyleCLIP also have pretty good hair editing capabilities, they cannot preserve the irrelevant attributes very well such as background, identity and clothes. Our method also demonstrates better preservation of the original hair structure when editing only the hair color.

For arbitrary hair editing word scenarios, the only methods that are instantly feasible without retraining the model are HairCLIP, StyleCLIP ("Optimization" version), and TediGAN. As shown in Figure 4, our method perform much better at such cases. In contrast, HairCLIP can only produce plausible results for text ("*Curly Short Hairstyle*") similar to the training texts, while all other methods struggle to produce reasonable editing effects.

Comparison with Hair Transfer Methods. We compare with state-of-the-art methods on hair transfer tasks. Among the $2,000$ images of the CelebA-HQ testset, the first 666 are set as the input images, the middle 666 are set as the hairstyle reference images, and the last 666 are set as the hair color reference images. As shown in Figure 5, when the hairstyle reference image is broadly aligned with the input image (first row), most methods yield plausible results. However, when not aligned (second row), only our method and SYH [20] are able to perform a more consistent hair transfer, which is achieved by introducing the pose alignment loss during the transfer to ensure that the facial shape and pose of the reference image are consistent with the source image. Compared to SYH, we achieve comparable hair transfer results, but support text, sketch, and other interactions beyond hair transfer.

Comparison with Local Hair Editing Methods. In terms of sketch-based local editing, we compare with the SOTA methods MichiGAN [31] and SketchSalon [41]. MichiGAN [31] uses user-drawn sketches to modify the orientation map to accomplish local hair editing. SketchSalon [41] trains a sketch-to-hair conditional translation network, with an additional soft alpha matte used to facilitate more natural blending. To generalize SketchSalon to local editing, we utilize the same mask as our method instead of a soft alpha matte, and the input sketch is colored as the average color within the mask area. As shown in Figure 6, MichiGAN struggles to perform satisfactory local editing and the reconstruction of other non-editing hair areas is slightly worse. Even ignoring the obvious blending artifacts, the local hair texture generated by SketchSalon is not in harmony with the surrounding hair. Compared to these two methods, our approach not only achieves satisfactory local editing but also better maintains the non-editing regions.

Comparison with Cross-Modal Hair Editing Methods. To the best of our knowledge, the only method that supports multimodal conditions to complete hairstyle and hair color editing is HairCLIP [36]. As the comparison shown in Figure 7, our approach not only perfectly prevents irrelevant attributes (identity, background, etc.) from being modified, but also achieves higher-quality editing effects. Moreover,

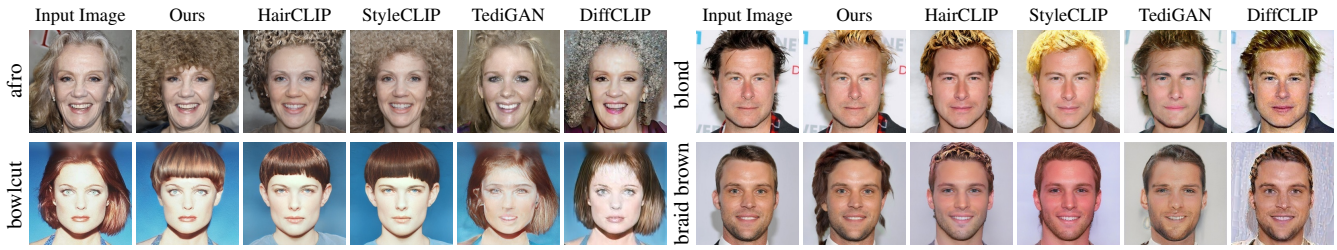

Figure 3. Visual comparison with HairCLIP [36], StyleCLIP-Mapper [21], TediGAN [40] and DiffusionCLIP [19]. The simplified text descriptions (editing hairstyle, hair color, or both of them) are listed on the leftmost side. Our approach demonstrates better editing effects and irrelevant attribute preservation (e.g., identity, background, etc.).

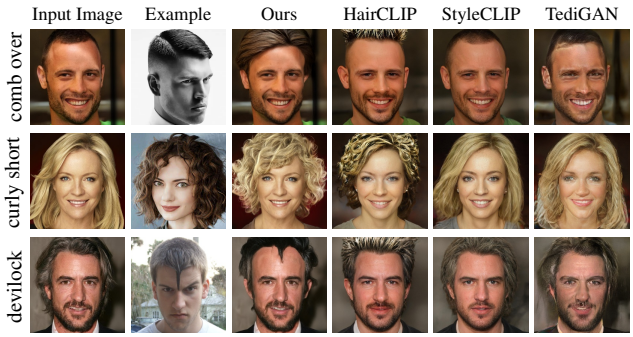

Figure 4. Visual comparison with HairCLIP [36], StyleCLIP-Optimization [21] and TediGAN [40] under any description setting. We additionally provide an example image for each description for better comparison.

HairCLIP allows only text and reference image while our method additionally supports sketch, mask, and RGB values. More diverse and comprehensive interactive editing results are shown in Figure 1 and supplementary materials.

User Study. For the above four types of comparisons, we recruit 20 volunteers with computer vision-related research backgrounds to execute a comprehensive user study. We randomly selected 20 groups of results from each experiment to form 80 test samples in total. The order of the different methods in each test sample is randomly shuffled. For each test sample, volunteers are asked to select the best option in terms of manipulation accuracy, irrelevant attribute preservation, and visual naturalness, respectively. As shown in Table 3, our method outperforms the baseline methods for most cases, except comparable results to Barbershop [47] and SYH [20] in the hair transfer setting. But our irrelevant attribute preservation performs best because of our hair color feature space blending mechanism, as demonstrated in Figure 5. It is worth mentioning that our goal is not to improve the performance of hair transfer, but to design a unified system that supports various hair editing and hair transfer tasks. Therefore, performing comparably with the state-of-the-art methods on the hair transfer task is acceptable.

4.2. Ablation Analysis

Importance of Initialization Strategy for Text Proxy Optimization. To justify our optimization starting point strategy for text proxy, we ablate three different optimization starting points: mean latent code, random sampling from $\mathcal{W}$ space, and inverted latent code of the input image within $\mathcal{W}+$ space. All other settings remain the same. As illustrated in Figure 8, the first two settings are more likely to complete high-quality editing, while the last one fails because the starting point deviates from the more suitable editing region. This may explain why StyleCLIP ("Optimization" version) and TediGAN struggle in performing hairstyle editing. Our strategy of randomly initializing the latent code around the mean within $\mathcal{W}$ space enjoys both good editability and diversity of the generated results.

Superiority of Sketch Proxy Generation Design. To generate sketch proxy, we ablate another two possible ways. The first way is to constrain the orientation field of the sketch proxy to be similar to that of user's sketch in the drawing region with the orientation loss used in MichiGAN[31]. The second way is to first obtain the image corresponding to the sketch through the conditional translation network (for simplicity, we choose sketch2hair here), and then constrain the sketch proxy to be similar to it in the drawing area by LPIPS [44] loss during the optimization process. The comparison is shown in Figure 9. Only our method and the LPIPS optimization-based version are feasible. However, our method requires only a single feed forward, which is more efficient.

Feature Blending vs. Latent Code Blending. To demonstrate the superiority of proxy feature blending, we compare it with the alternative scheme based on the linear combination of latent codes. In detail, we initialize an interpolation factor for the latent code of the global editing proxy, the local sketch proxy, and the input image, respectively. In the optimization process, we optimize these three interpolation factors so that the generated image corresponding to the interpolated latent code is similar to the proxies or input im-

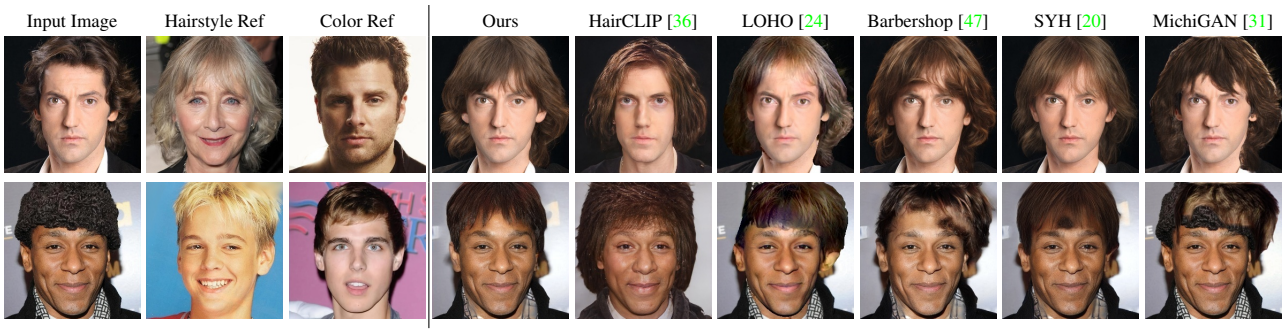

Figure 5. Visual comparison with HairCLIP [36], LOHO [24], Barbershop [47], SYH [20] and MichiGAN [31] on hair transfer. Only our method and SYH can accomplish unaligned hair transfer while keeping irrelevant attributes unmodified.

	Text-Driven					Hair Transfer						Sketch-Based			Cross-Modal	
Metrics	Ours	[36]	[21]	[40]	[19]	Ours	[36]	[24]	[47]	[20]	[31]	Ours	[31]	[41]	Ours	[36]
Accuracy	**41.5%**	32.3%	22.5%	1.0%	2.8%	28.0%	2.3%	4.8%	**29.3%**	28.5%	7.3%	**76.8%**	22.0%	1.3%	**82.8%**	17.3%
Preservation	**81.0%**	5.3%	3.3%	0.3%	10.3%	**32.8%**	2.8%	8.3%	15.3%	26.3%	14.8%	**62.0%**	33.3%	4.8%	**94.0%**	6.0%
Naturalness	**46.8%**	25.5%	22.0%	1.8%	4.0%	26.5%	9.5%	2.5%	22.3%	**35.0%**	4.3%	**60.5%**	38.0%	1.5%	**65.3%**	34.8%

Table 3. User study on text-driven image manipulation, hair transfer, sketch-based local hair editing and cross-modal hair editing methods. Accuracy denotes the manipulation accuracy for given conditional inputs, Preservation indicates the ability to preserve irrelevant regions and Naturalness denotes the visual realism of the manipulated image.

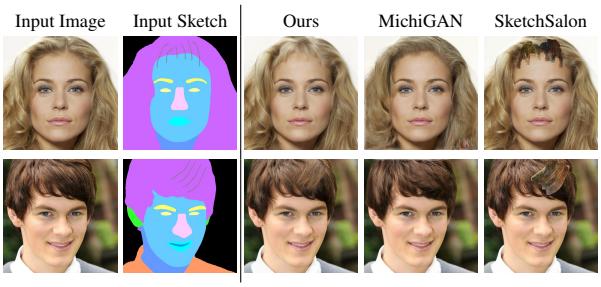

Figure 6. Comparison with MichiGAN [31] & SketchSalon [41] on sketch-based local hair editing. We provide sketches in the facial parsing map for better visualization.

Figure 7. Qualitative comparison with HairCLIP on cross-modal conditional input. Our approach shows better editing effects & excellent preservation of irrelevant attributes.

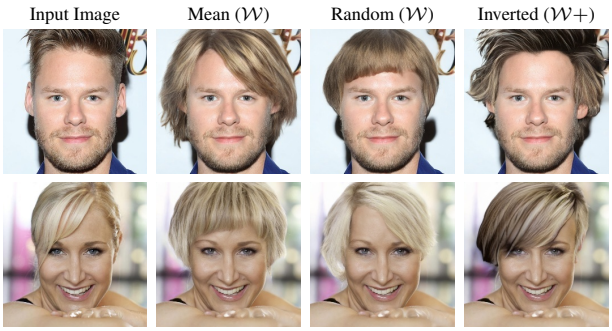

Figure 8. Ablation of different starting point settings for text proxy. The text description is "*Bob Cut Hairstyle*".

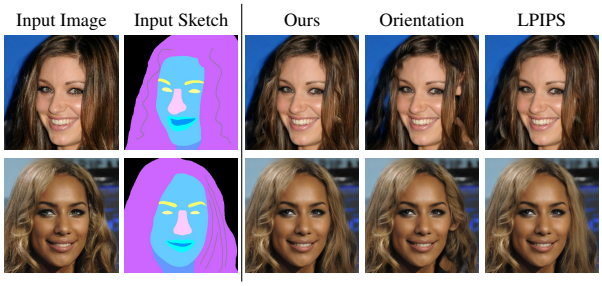

Figure 9. Ablation on sketch proxy generation design. We provide sketches drawn in the facial parsing map for better visualization.

age within the corresponding region. In Figure 10, we use the hairstyle reference image as an example to generate the global proxy. It is obvious that our scheme accomplishes global editing and local editing while perfectly keeping the

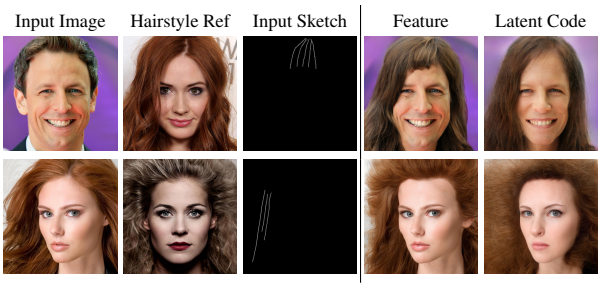

Figure 10. Ablation of feature blending vs. latent code blending. Feature blending can enable global editing, local editing, and irrelevant attributes preservation simultaneously.

irrelevant properties unmodified, while latent code blending does not perform either one well.

5. Conclusion and Discussion

In this paper, we propose a unified hair editing system HairCLIPv2, which presents the first attempt that supports both simple text/reference image interaction and fine-grained local interactions. It innovatively converts all hair editing tasks into hair transfer tasks, with the corresponding editing conditions converted into transfer proxies. It can not only achieve high-quality hair editing results, but also well preserve the irrelevant attributes from being modified. In the future, we will study how to use feed-forward networks to generate all the proxies. Also, it is worthy to generalize the proposed framework to support generic natural images.

6. Acknowledgement

This work was supported in part by the Natural Science Foundation of China under Grant U20B2047, 62072421, 62002334, 62102386 and 62121002, the Fundamental Research Funds for the Central Universities under Grant WK5290000003, Key Research and Development program of Anhui Province under Grant 2022k07020008. This work was also partly supported by Shenzhen Key Laboratory of Media Security, and the Opening Fund of Key Laboratory of Cyberculture Content Cognition and Detection, Ministry of Culture and Tourism. This work was also partially supported by a GRF grant (Project No. CityU 11216122) from the Research Grants Council (RGC) of Hong Kong. Thank Yi Yin for her help in this work.

References

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

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020.

[3] Abdul Fatir Ansari, J. Scarlett, and Harold Soh. A characteristic function approach to deep implicit generative modeling. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7476–7484, 2020.

[4] Martín Arjovsky, Soumith Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[5] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[6] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[7] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020.

[8] Xiaoyi Dong, Yinglin Zheng, Jianmin Bao, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2023)*, 2023.

[9] Ruili Feng, Jie Xiao, Kecheng Zheng, Deli Zhao, Jingren Zhou, Qibin Sun, and Zheng-Jun Zha. Principled knowledge extrapolation with GANs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 6447–6464. PMLR, 17–23 Jul 2022.

[10] Ruili Feng, Deli Zhao, and Zheng-Jun Zha. Understanding noise injection in gans. In *International Conference on Machine Learning*, pages 3284–3293. PMLR, 2021.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[12] Ishaan Gulrajani, F. Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[13] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1745–1753, 2019.

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[15] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

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

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

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

[19] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[20] Taewoo Kim, Chaeyeon Chung, Yoonseo Kim, Sunghyun Park, Kangyeol Kim, and Jaegul Choo. Style your hair: Latent optimization for pose-invariant hairstyle transfer via local-style-aware hair alignment. *arXiv preprint arXiv:2208.07765*, 2022.

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[23] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[24] Rohit Saha, Brendan Duke, Florian Shkurti, Graham W Taylor, and Parham Aarabi. Loho: Latent optimization of hairstyles via orthogonalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1984–1993, 2021.

[25] Edgar Schönfeld, B. Schiele, and A. Khoreva. A u-net based discriminator for generative adversarial networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8204–8213, 2020.

[26] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Xinhui Song, Chen Liu, Youyi Zheng, Zunlei Feng, Lincheng Li, Kun Zhou, and Xin Yu. Hairstyle editing via parametric controllable strokes. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[29] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019.

[30] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Bin Liu, Gang Hua, and Nenghai Yu. Diverse semantic image synthesis via probability distribution modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[31] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)*, 39(4):95–1, 2020.

[32] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, Gang Hua, and Nenghai Yu. Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

[34] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.

[35] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[36] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. Hairclip: Design your hair by text and reference image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18072–18081, 2022.

[37] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. E2style: Improve the efficiency and effectiveness of stylegan inversion. *IEEE Transactions on Image Processing*, 31:3267–3280, 2022.

[38] Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *ICML*, 2023.

[39] Yiqian Wu, Yong-Liang Yang, and Xiaogang Jin. Hairmapper: Removing hair from portraits using gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4227–4236, 2022.

[40] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[41] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.

[42] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[43] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[45] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *International Conference on Machine Learning (ICML 2023)*, 2023.

[46] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.

[47] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *ACM Transactions on Graphics (TOG)*, 40(6):1–13, 2021.

[48] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Hairnet: Hairstyle transfer with pose changes. In *European Conference on Computer Vision*, pages 651–667. Springer, 2022.

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

[50] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.