

# 基于 FS 空间的人脸发型编辑

## 摘要

近年来，头发编辑取得了巨大的进步 [27, 29, 32]。早期的头发编辑方法使用绘制良好的草图 [25, 26] 或蒙版 [25, 31] 来指定编辑条件。尽管它们可以实现非常细粒度的局部控制，但交互方式低效。得益于 StyleCLIP [22] 的突破，基于隐空间重建的发型编辑的方法也逐渐流行，但是这种方法无法很好地保持无关属性，并且不支持草图或掩码指定的细粒度控件。在本文中，本文提出了 FS-CLIP，通过同时结合蒙版和 CLIP [30] 的方法，保留了发型良好的结构光影和面部的特征，以及在无关属性的保留上的优越性。

**关键词：**发型编辑；StyleGAN；FS latent space；

## 1 引言

头发编辑作为一个有趣且具有挑战性的问题，吸引了来自学术界和工业界的大量研究关注 [1–4]。在过去的几十年里，该领域取得了巨大的进步，实现了基于蒙版草图和 GAN inversion [5, 6] 的头发迁移，头发重建。在早期的蒙版和草图编辑条件可以启用细粒度的本地控件，到现在有将人脸整体映射到隐空间中进行重建，进而编辑发型属性。近年来，跨模态的视觉和语言表示学习取得了令人瞩目的突破 [7, 8, 30]，使得文本引导的图像表示学习成为可能。HairCLIP [9] 首个尝试在统一的框架内通过文本描述和参考图像支持发型编辑。尽管这种文本驱动和参考图像编辑头发的交互方式更加高效和人性化，但 HairCLIP 无法支持蒙版细粒度的控件，同时由于其头发编辑是通过隐空间的操作完成，受限于完全在隐空间中解耦不同属性是十分困难的，重建的过程中不可避免地会改变无关发型的属性，如人脸 ID 信息，背景等。尽管在 HairCLIPv2 [10] 中引入了 Barbershop [11] 的 FS 空间，可以实现较好的无关属性保留的结果，但是由于 HairCLIPv2 多次重建，最终效果参考原图依旧不是很理想，在发色编辑方面，由于其通过直接通过 StyleGAN [12] 空间的高层进行发色的编辑，偏离了对高层发型的光影，真实度以及发型的结构的保留。在本文提出了 FS-CLIP，在支持文本描述，参考图像，蒙版的编辑功能基础上，通过对 StyleGAN 中不同层更精细的控制，在保留不相关属性和发型的结构和真实性上都有很大的改进。

## 2 相关工作

### 2.1 使用 GANs 进行发型编辑

现今的发型编辑方法大致可以分为基于 Condition GANs [13] 和预训练的 StyleGAN。MichiGAN [14] 通过提取参考发型图像的方向图实现发型的转移。Barbershop 提出将人脸

映射到 FS 空间内进行发型转移。但是当原图像与发型参考图像的人脸的姿势和方向有较大差异时，这些方法往往会出现伪影和难以对齐的问题。随着多模态模型的发展,StyleCLIP 将 CLIP 与 StyleGAN 结合，利用 CLIP Loss 优化实现了用文本编辑发型； HairCLIP 在 StyleCLIP 基础上以解耦的方式分别对发型和发色进行编辑，实现了良好的发型编辑效果。但是由于 HairCLIP 编辑发型，需要将人脸映射到隐空间中进行重建，生成的图像人脸等不相关属性也会随之改变。因此此文中，本文通过引入 FS 空间，可以更好的保留不相关属性，同时保留文本和参考图像实现发型编辑的操作。

## 2.2 隐空间的嵌入和编辑

GAN inversion 通过将图片映射到 StyleGAN 的隐空间中，从而可以对图像进行多种下游图像编辑任务。GAN inversion 目前大致可以分为两类：一种是为了实现对真实图像的完美重建 [15, 16]，但往往使得图像在隐空间中的特征难以解耦，难以对属性进行单独的编辑；另一种则是将映射到更适合编辑的潜空间中 [16, 17, 28]，但是牺牲了重建质量。受到 Barbershop 提出的一种新颖的 GAN inversion 方法，该方法通过结合 StyleGAN 的特征空间与 W+ 空间 [16] 相结合，形成了 FS 空间，如图 1 所示。受到该方法的启发，本文通过将人脸特征和发型特征映射到 StyleGAN 的特征空间，通过真实图像优化特征空间 F，实现了保持人脸不相关属性不会改变，同时忠于发型提供的图像的结构特征。

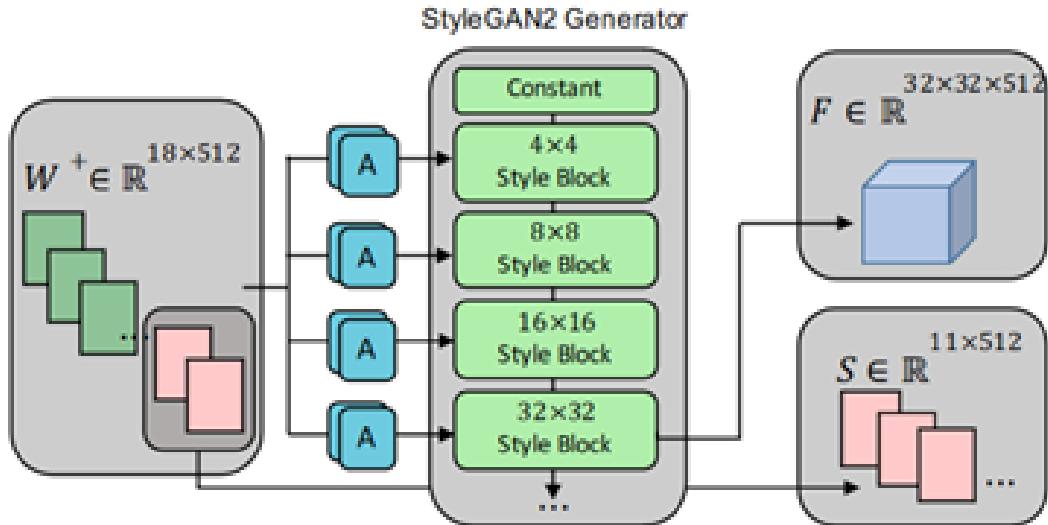


图 1. FS 与 W+ 潜在空间的关系

## 2.3 基于文本的发型编辑

随着跨模态视觉和语言表征的迅速发展，尤其是 CLIP 这种文本图像对训练的多模态语言模型，使得基于文本编辑图像成为可能。StyleCLIP 作为第一个将 StyleGAN 与引入到 CLIP，利用 CLIP 的损失来修正输入的潜向量，从而达到文本引导潜在向量向语义方向改变，实现了交互式的文本驱动的图像编辑操作。而 HairCLIP。但是 StyleCLIP 需要为每个特定的头发编辑描述训练一个单独的映射器网络，这对于实际应用来说并不灵活。对于 TediGAN [18]，提出了两种方法：TediGAN-A 将文本和图像分别编码到 StyleGAN 的隐空间中，通过风格混

合完成操作，解耦性较差，难以完成头发编辑；TediGAN-B 使用 CLIP 进行优化操作，以提供文本-图像相似性，但缺乏从大型数据集学习的知识，使得该过程不稳定且耗时。

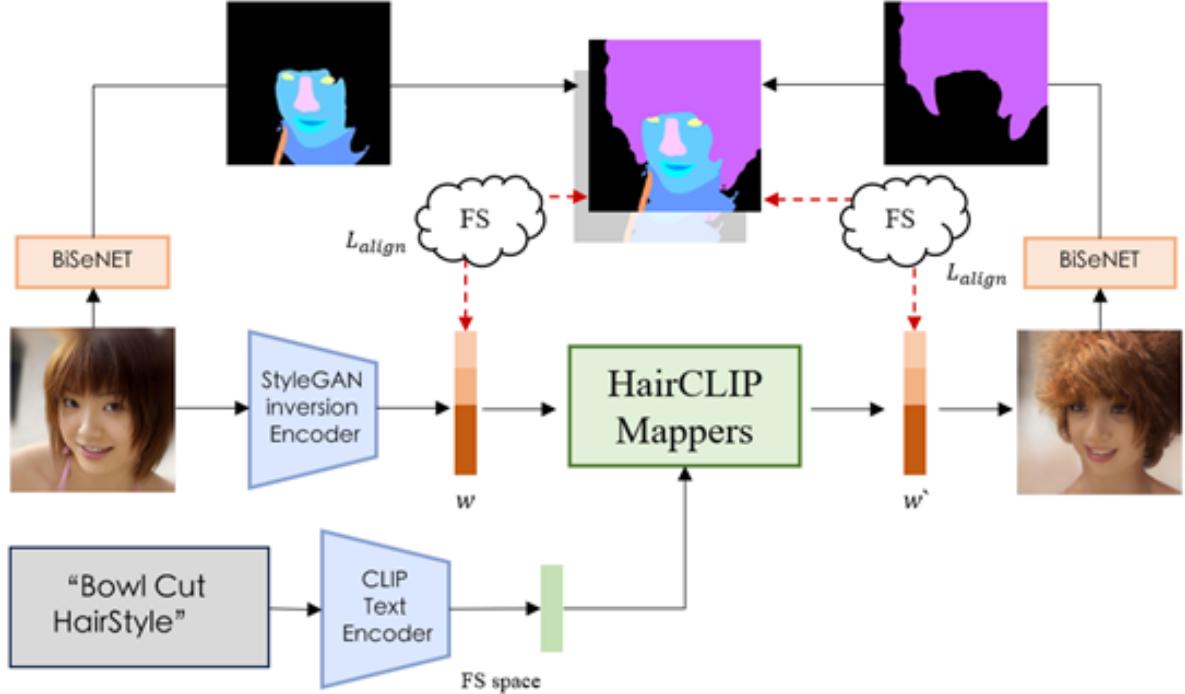


图 2. FS-CLIP 结构示意图

### 3 本文方法

#### 3.1 本文方法概述

本文的方法先将原图片通过一个轻量级的分割网络 BiSeNET [19] 将  $I_f$  人脸分割为脸部除发型区域获得蒙版  $M_f$ , 以及发型区域蒙版  $M_h$ , 将两个蒙版组合成蒙版  $M_b$ , 如图 3所示。再将目标图像输入到 StyleGAN inversion Encoder 获得隐空间的  $W+$  向量  $w$ 。 $w$  通过 HairCLIP Mappers 后生成  $w'$  向量, 将  $w$  和  $w'$  分别映射到 FS 空间中获得  $F$  向量与  $F'$  向量。再将  $F$  向量、 $F'$  向量分别与蒙版  $M_b$  通过  $L_s$  进行优化。使  $F$  向量之在  $W+$  空间中对齐属性的特征, 再将  $F$  向量与  $F'$  向量混合, 把获得  $F_{blend}$  而再把原图像的  $S$  向量组合成

$$C_{blend} = (F_{blend}, S)$$

, 最后将  $C_{blend}$  输入到 StyleGAN 中获得编辑好发型的图片  $I_c$ , 整体框架如图 2所示。

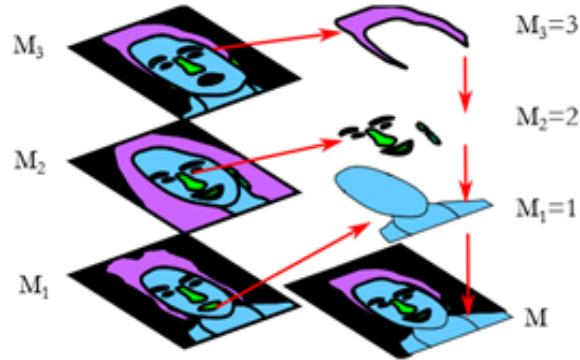


图 3. 生成目标掩码示意图

### 3.2 发型颜色函数定义

$$\mathcal{L}_{s\_mc} = \|\text{avg}(\mathbf{x}_M * P_h(\mathbf{x}_M)) - \text{avg}(\mathbf{x}_w * P_h(\mathbf{x}_w))\|_1$$

## 4 复现细节

### 4.1 与已有开源代码对比

本次工作主要引用了 <https://github.com/wty-ustc/HairCLIPv2> 中的代码。下面是改进部分的关键代码。

```
config = Config(
    access_key_id=os.environ.get('ALIBABA_CLOUD_ACCESS_KEY_ID'),
    access_key_secret=os.environ.get(
        'ALIBABA_CLOUD_ACCESS_KEY_SECRET'),
    endpoint='imageseg.cn-shanghai.aliyuncs.com',
    region_id='cn-shanghai'
)
segment_hair_request = SegmentHairAdvanceRequest()
with io.BytesIO(binary_data) as image_buffer:
    segment_hair_request.image_urlobject = image_buffer
    runtime = RuntimeOptions()
    client = Client(config)
    response = client.segment_hair_advance(segment_hair_request,
                                             runtime)
    response_data = response.body
    image_url = response_data.data.elements[0].image_url
    height = response_data.data.elements[0].height
    width = response_data.data.elements[0].width
    x = response_data.data.elements[0].x
```

```

y = response_data.data.elements[0].y
original_width = 1024
original_height = 1024
if image_url:
    image_data = urlopen(image_url).read()
    image = Image.open(io.BytesIO(image_data))
    # 调整位置大小
    # 指定背景色（可以根据需求选择）
    background_color = (0, 0, 0) # 这里选择黑色
    # 创建一个黑色背景的RGB图像
    rgba_background = Image.new('RGBA', (original_width,
                                          original_height), background_color)
    image_array = np.array(image, dtype=np.uint8)
    rgba_background_array = np.array(rgba_background, dtype=np.uint8)
    )
    # 设置阈值a, Alpha通道值小于某个阈值a时，将RGB通道全部设置为零
    a = 255 # 你可以根据需要调整阈值
    # 将Alpha通道小于阈值的区域的RGB通道设置为零
    alpha_channel = image_array[:, :, 3]
    image_array[alpha_channel < a, :3] = 0
    image_array[alpha_channel >= a, :3] = 255
    rgba_background_array[y:y + height, x:x + width] = image_array
    image_pil = Image.fromarray(rgba_background_array, 'RGBA')
    result_image = Image.alpha_composite(Image.new('RGBA',
                                                   rgba_background.size, (0, 0, 0)), image_pil)
    result_image = np.array(result_image)
    result_image = result_image[:, :, 0]
    tensor_image = torch.from_numpy(result_image)
    tensor_image = tensor_image.unsqueeze(0).unsqueeze(0)
    tensor_image = tensor_image.to('cuda:0')
    hair_mask = (tensor_image == 255).float()
    return hair_mask

```

```

for i in pbar:
    optimizer.zero_grad()
    latent_in = torch.cat([latent_W_optimized[:, :6, :], 
                          latent_end], dim=1)
    img_gen, _ = self.generator([latent_in], input_is_latent=True, randomize_noise=False)
    img_gen_256, gen_hairmask_256 = self.gen_256_img_hairmask(

```

```

        img_gen)

hair_style_loss = self.transfer_loss_builder.style_loss(
    ref_img_256, img_gen_256, mask1=ref_hairmask_256, mask2=
    gen_hairmask_256)

delta_w_loss = self.delta_loss(latent_W_optimized[:, :6,
    :], latent_prev)

if painted_mask is not None:
    down_seg = self.seg(img_gen)[1]
    hair_mask_loss = self.mask_loss(down_seg, painted_mask)
    loss = self.style_lambda * hair_style_loss + self.
        delta_w_lambda * delta_w_loss + self.
        hair_mask_lambda * hair_mask_loss

else:
    loss = self.style_lambda * hair_style_loss + self.
        delta_w_lambda * delta_w_loss

latent_prev = latent_W_optimized[:, :6, :].clone().detach()
loss.backward()
optimizer.step()

pbar.set_description((f"ref_loss: {loss.item():.4f};"))

if (i % visual_interval == 0) or (i == (self.step-1)):
    with torch.no_grad():
        img_gen, _ = self.generator([latent_in],
            input_is_latent=True, randomize_noise=False)
        visual_list.append(process_display_input(img_gen))

return latent_in, visual_list

```



图 4. W+ 空间与 FS 空间重建效果图

## 5 实验结果分析

### 5.1 StyleGAN 重建空间实验

本文在 CelebA-HQ [20] 上进行实验，对 HairCLIP 在重建过程中使用的 e4e 模型重建的 W+ 空间和 Barbershop 提出的 FS 空间进行 inversion 实验。如图 3 中所示，三行图像由上倒下依次是原图像，通过 W+ 空间重建的图和通过 FS 空间重建的图。在第一列中，可以看到 W+ 空间对偏向于插画风格的图，重建后无法保留原来的风格，而是更偏向于更为真实的风格，而 FS 空间则可以更好的保留图像原来的风格。在第二列和第三列中，可以看出 W+ 重建的图像无法还原脸上的痣或者眉心处的装饰等面部的小众特征。这是由于 W+ 更偏向于生成数据集中较为大众的特征，就忽略这些小众的特征。同样的第四列脸部的光影以及第六列中眼睛的朝向等小众特征都难以重建。在第五列和第六列第七列中可以看到，第七列中 W+ 空间将帽子重建成了头发，第五列中的发饰甚至直接忽略了。FS 空间对于原图像发型中的装饰以及发型的结构和光影都达到了一个几乎还原的效果。在背景的保留方面，可以从第四列中看到 FS 空间的背景也几乎完美地保留下来了，这得益于 FS 空间增加了对隐空间对细节的表征能力，实现了对图像特征地空间控制。

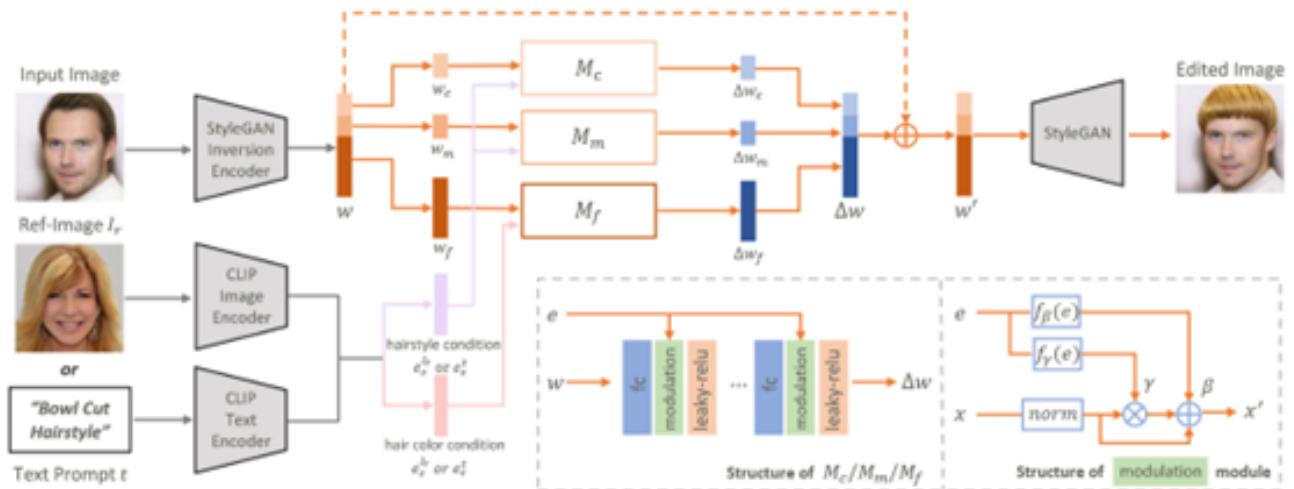


图 5. HairCLIP 结构图

### 5.2 HairCLIP 发型编辑实验

HairCLIP 受开创性工作 StyleCLIP 的启发，利用预训练的 StyleGAN 强大的合成能力，旨在学习一个额外的映射器网络来实现头发编辑功能。更具体地说，给定要编辑的真实图像，本文首先使用 StyleGAN 反演方法 e4e 来获取 W+ 空间中的潜在代码  $w$ ，然后使用映射器网络根据  $w$  预测潜在代码变化  $\Delta w$  和编辑条件（包括发型条件  $e_s$  和头发颜色条件  $e_c$ ）。最后，修改后的潜在代码  $w = w + \Delta w$  将被反馈到预训练的 StyleGAN 中以获得目标编辑结果。整体流程如图 5 所示如图 6 所示，HairCLIP 由于只收集了 44 个发型文本描述和 12 个头发颜色描述进行训练，所以对于 CLIP Text Encoder 会将未在这 44 个发型描述内和 12 种颜色内的发型编辑效果不是很理想。可以看到图，其中左侧是目标图像通过 e4e 重建后的图像，右侧为参考发型图像，中间是生成图像。可以看到基本上编辑发型任务都失败了。所以本文采用转移发型的方法，就可以很好地避免这个问题。在图 9 中，是通过输入文本进行发色的编辑，右

侧为原图，左侧为编辑后的图。可以看到均取得了良好的效果，但值得注意的是，在右边的两个大图中，可以看到图片包括背景和人脸的整体的颜色都被改变了。这是由于改变 W+ 空间中较高层的颜色特征时，发型的颜色和其他不相关区域的颜色难以解耦，因此本文中采用蒙版的方法，对于不相关区域进行蒙版操作，可以更好的保持不相关属性的保留度。

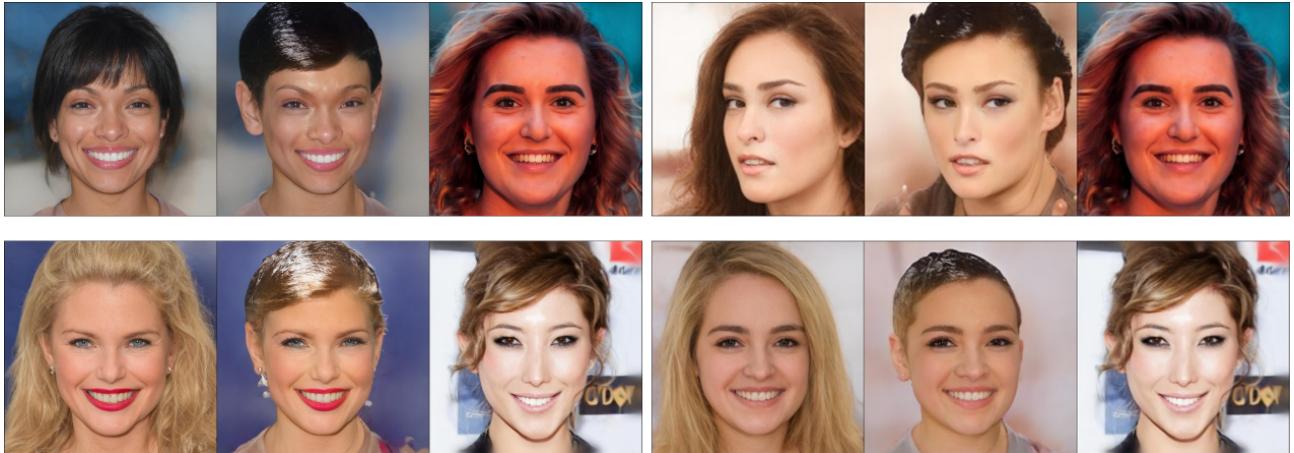


图 6. 用 HairCLIP 通过参考图像编辑发型

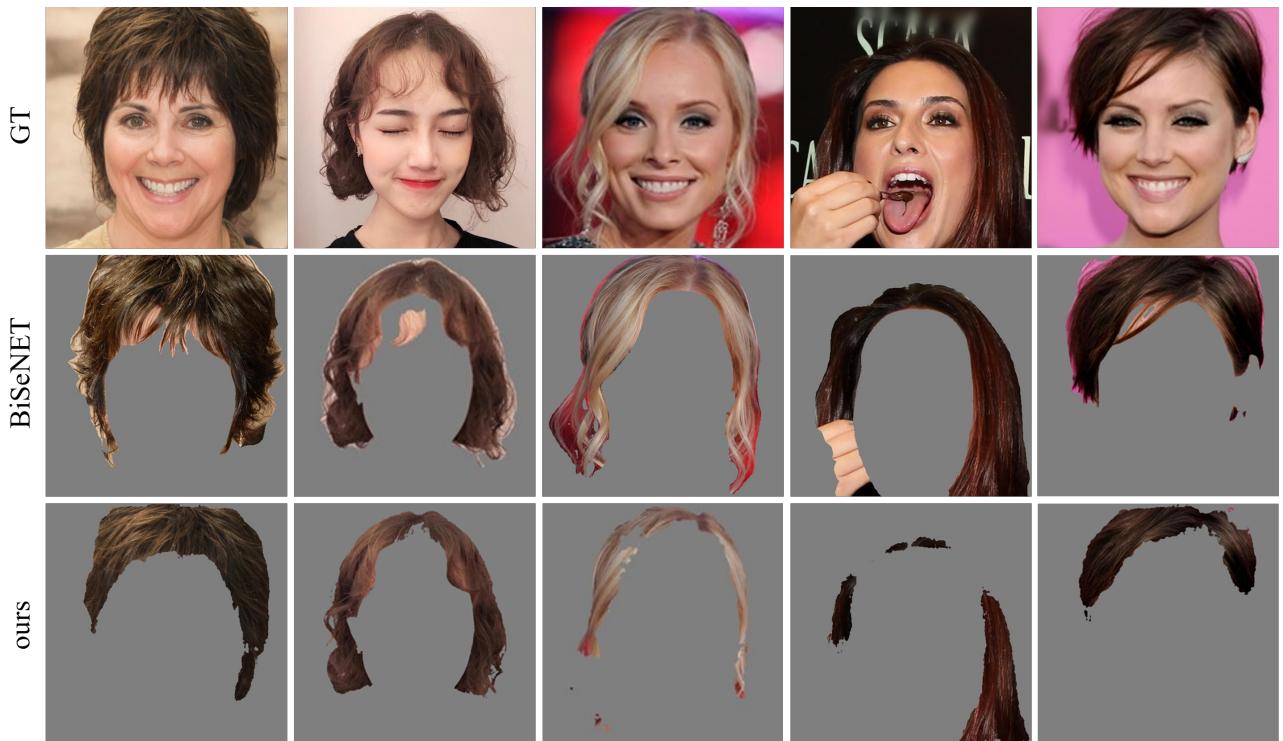


图 7. 本文蒙版方法与 BeSiNET 方法比较

而 HairCLIPv2 中，虽然同样的使用了 FS 空间来尝试保留人脸的 ID 信息，但是由于 HairCLIPv2 将发型编辑分为三个阶段，需要通过 FS 空间对图像进行三次重建并进行优化，这将使得图像的 ID 等不相关的属性还是难以保留。而本文的整个方法对图片只通过 FS 空间进行一次重建和优化。如图 10 所示，在第二行中 HairCLIPv2 的方法的眼睛部分与原图像相比变化很大。如表 1 所示，IDS [21] 用来衡量脸部区域的身份相似度，而 PSNR [23] 和 SSIM [24] 用来衡量编辑前后图片非头发区域交并区域的相似度。不同于 HairCLIP 中与重建后的图像

进行对比，本文采用 HairCLIPv2 中的将编辑后的图像与原图像进行计算，得到本文提出的方法在三个指标中均优于 baseline 的方法。并且由于 HairCLIPv2 将发色编辑作为一个优化任务，通过 CLIP 的 color Loss，或者发型蒙版区域的颜色均值优化获得发色，但是这样将不可避免地损失 W+ 空间高层次中包含发型特征，比如光影，发型层次等特征信息。

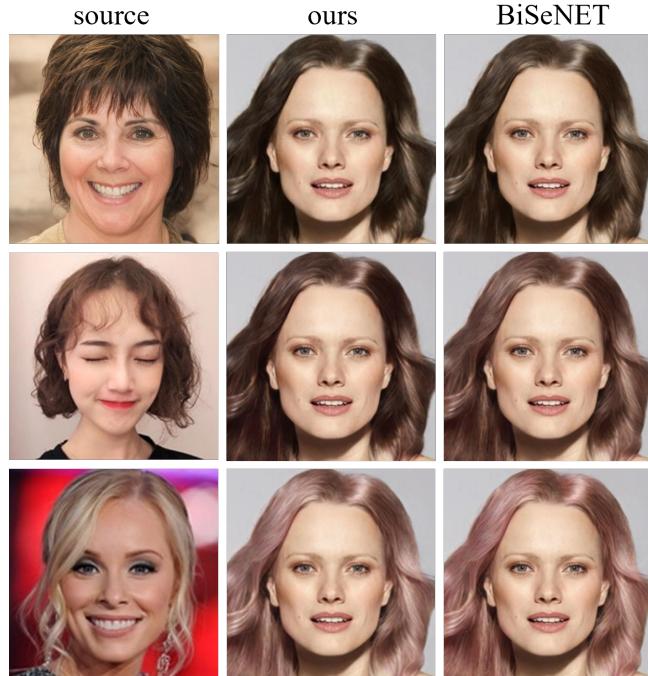


图 8. 本文与 HairCLIPv2 用参考图像修改发型颜色结果对比

在图 2 的视觉效果比较中，HairCLIPv2 对于发型光影和结构上难以忠于目标图像，发型地结构上比较模糊，同时光影的层次也没有那么清晰。而本文的方法就能很好地还原出原图像的光影和发型的结构。而 TediGAN 在所有文本条件的任务中都失败了。这是因为使用 GAN 的优化方法由于缺乏从大型数据集中学习到的知识，导致生成不稳定。而 HairCLIP 对人脸的保持方面由于使用的是 e4e 的反演方法，并且没有做相关优化，所以在不相关属性的保持上效果不是很理想。



图 9. 用 HairCLIP 通过文本编辑发型颜色

Method	IDS	PSNR	SSIM
Ours	<b>0.90</b>	<b>31.6</b>	<b>0.92</b>
HairCLIPv2	0.84	29.5	0.91
HairCLIP	0.45	21.6	0.74
StyleCLIP	0.43	19.6	0.72
TediGAN	0.16	22.5	0.74

表 1. 不相关属性保留的定量比较

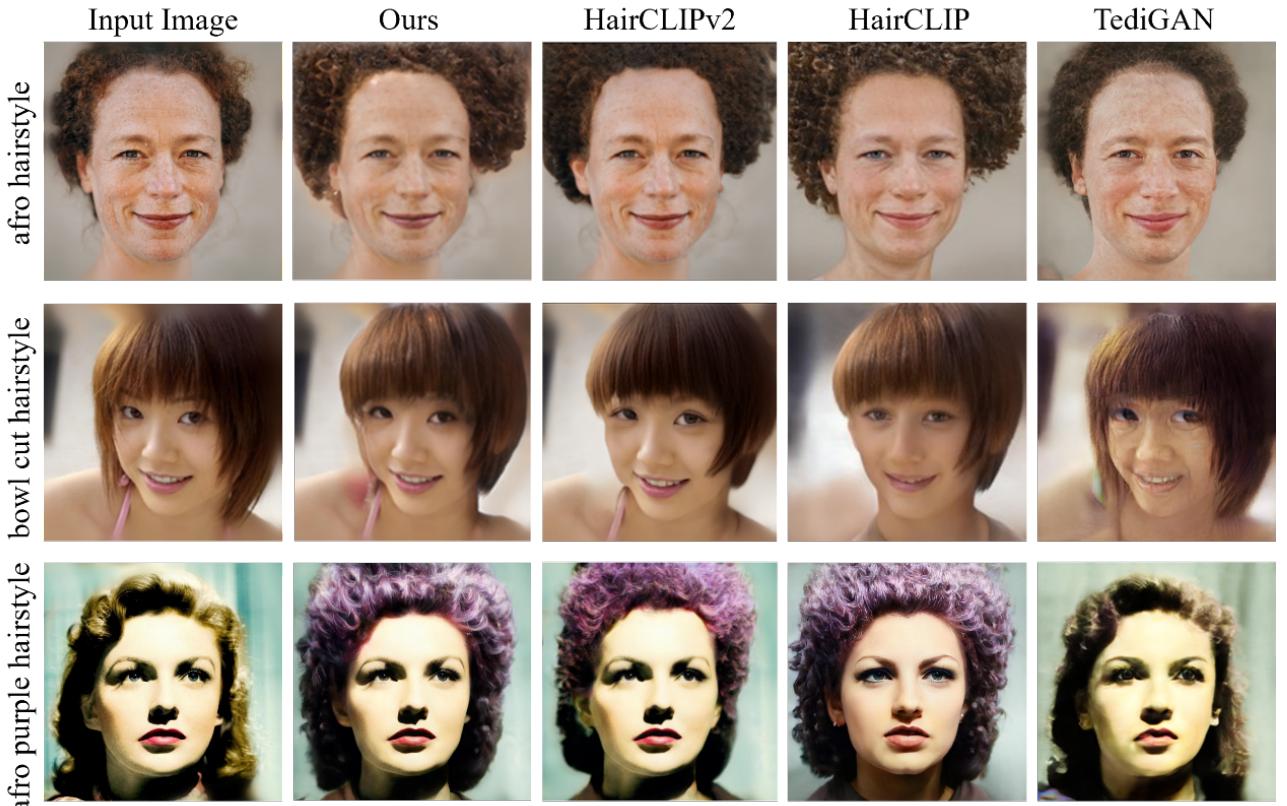


图 10. 本文方法与 HairCLIPv2,HairCLIP,TediGAN 方法比较

## 6 总结与展望

本文提出了一个将 HairCLIP 与 FS 空间统一的一个发型编辑方法 FS-CLIP，它利用 HairCLIP 的文本输入获取适应人脸姿势和方向的发型，结合 FS 空间对于人脸分割蒙版拼接发型和人脸其他区域的无缝连接以及良好的重建图像的特点，将 HairCLIP 生成的发型与目标人脸进行拼接，最后生成出保持目标图像的人脸信息和其他无关属性，以及忠于原发型的结构光影的图片。同时本文设计了更适合参考发色编辑的蒙版，使得定性的实验表明本文的颜色编辑的效果更接近参考图像的发色。定性和定量比较的实验表明，本文的方法在无关属性的保留方面优于竞争方法。

## 参考文献

- [1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.
- [2] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 5744–5753, 2019.
- [3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.
- [5] Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8385–8393, 2020.
- [6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [7] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.
- [8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.
- [9] Yuge Huang, Yuhang Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5901–5910, 2020.
- [10] Ali Jahanian, Lucy Chai, and Phillip Isola. On the ”steerability” of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.
- [11] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13799–13808, 2021.

- [12] 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.
- [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. *arXiv preprint arXiv:1710.10196*, 2017.
- [15] 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.
- [16] 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.
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [18] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.
- [20] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *arXiv preprint arXiv:2005.07728*, 2020.
- [21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.
- [22] 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.
- [23] 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.

- [24] 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.
- [25] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.
- [26] 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. *arXiv preprint arXiv:2010.16417*, 2020.
- [27] 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.
- [28] 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.
- [29] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *arXiv preprint arXiv:2109.07874*, 2021.
- [30] 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.
- [31] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *arXiv preprint arXiv:2106.01505*, 2021.
- [32] 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.