HairCLIPv2: Unifying Hair Editing via Proxy Feature Blending
Supplementary Material

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

1. Implementation Details

For the generation of all optimization-based proxies, we set the learning rate to 0.01 and use the Adam [3] optimizer. For text proxy, the overall loss in the optimization process is defined as follows: $L^{text} = \lambda^{clip} L^{clip} + \lambda^{pose} L^{pose} + \lambda^{shape} L^{shape}$, where λ^{clip}, λ^{pose}, and λ^{shape} are set to 1, 200, and 1 respectively to make each loss balance. For the start point strategy for optimization of text proxy, we set $\psi = 0.3$ to ensure that the initial optimization starting point w^{init} is around the average face latent code w^{mean}. For reference proxy, The overall loss of hairstyle transfer is defined as follows: $L^{ref} = \lambda^{style} L^{style} + \lambda^{pose} L^{pose} + \lambda^{reg} L^{reg} + \lambda^{shape} L^{shape}$, where λ^{style}, λ^{pose}, λ^{reg}, and λ^{shape} are set to 2000, 200, 1, and 1 respectively to make each loss balanced.

For sketch proxy, the number of training iterations for the sketch2hair translation inverter T for local hairstyle editing is $500,000$. The training loss includes regular pixel-level L_2 loss $L^{mse} = ||I^{sketch} - G(T(S))||_2^2$, feature-level LPIPS [13] loss $L^{LPIPS} = ||F(I^{sketch}) - F(G(T(S)))||_2^2$, where S represents the hairstyle sketch input, I^{sketch} stands for the hair image corresponding to the hairstyle sketch S, T means the sketch2hair invertor to be trained, and F denotes the AlexNet [4] feature extractor. To provide more local supervision, we additionally use multi-layer face parsing loss L^{m_par} which provides more detailed knowledge by introducing multi-layer features from the pre-trained face parsing network:

$$L^{m\_par} = \sum_{i=1}^{5} (1 - cos(P_i(I^{sketch}), P_i(G(T(S))))), \quad (1)$$

where $P_i(I^{sketch})$ represents the feature corresponding to the i-th semantic level from the face parsing network P [5] of the hair image I^{sketch}. The overall training losses are as follows:

$$L^{sketch} = \lambda^{mse} L^{mse} + \lambda^{LPIPS} L^{LPIPS} + \lambda^{m\_par} L^{m\_par}, \quad (2)$$

where λ^{mse}, λ^{LPIPS}, and λ^{m_par} are set to 0.5, 0.8, and 1, respectively.

2. Quantitative Results

2.1. Editing Speed

We compare the editing runtime with competitive methods in Table 1. We are faster than baseline methods in hair transfer and sketch-based editing. For text-based editing, we are slower but with better editing quality and irrelevant attributes preservation. Moreover, only our method excels at the task of hair editing with arbitrary text.

Text	Ours(35.2)	TediGAN(28.0)	HairCLIP(**0.10**)
Transfer	Ours(**58.9**)	Barbershop(117.8)	SYH(136.8)
Sketch	Ours(**0.04**)	MichiGAN(0.42)	SketchSalon(0.14)

Table 1. Editing Runtime on 2080 Ti (seconds).

3. Ablation Analysis

3.1. Necessity of Balding Steps

We employ two key steps during the process of converting the input image into bald proxy: first, editing the latent code of the input image to obtain its balded latent code; second, performing feature blending between the balded features and the original features of the input image to preserve the irrelevant attributes from being modified as shown in Eq. 1 of the main text. To verify the necessity of these two steps,

[†] Wenbo Zhou is the corresponding author. Our code is available at https://github.com/wty-ustc/HairCLIPv2.

we perform experiments on the following two variants: (A). without balding, i.e., step 1 is skipped and Eq. 1 of the main text becomes $F_7^{bald} = F_7^{src}$; (B). without feature blending with original image after balding, i.e., Eq. 1 of the main text becomes $F_7^{bald} = G(w_{1-7}^{bald})$. The visual comparison results are shown in Figure 1. Since variant A does not employ the balding method to de-obscure, there are obvious artificial artifacts caused by blending the bald proxy features with the editing proxy features. Although the result of variant B looks relatively natural overall, the editing of the 1-dimensional latent code inevitably modifies other irrelevant attributes (background, identity, etc.). Combining the advantages of these two steps, our default setting achieves both the natural editing effect resulting from the balding operation to de-obscure while inpainting the hair area sensibly and the excellent irrelevant attribute preservation caused by feature blending.

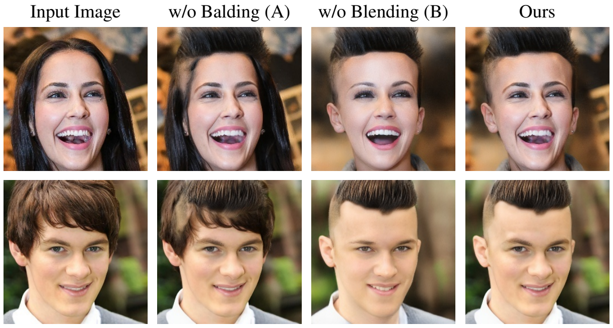

Figure 1. Ablation analysis on the necessity of balding steps. The text description is "*Mohawk Hairstyle*".

3.2. Robustness of Balding Technique

Our system uses HairMapper [10] in the first step of generating bald proxy to baldify the input image and thus remove the occlusion and facilitate feature blending with the editing proxy later. In Figure 2, we illustrate the results of the balding technique [10] and our method under extreme lighting, pose, and self-occlusion conditions. Obviously, the balding technique performs relatively robustly in most extreme conditions. In the case of the self-occlusion condition, the balding technique shows significant artifacts at the hand position, while our method is not affected because of the feature blending mechanism adopted in the second step of generating the bald proxy.

4. Limitations

Despite the unprecedented unification, our method has some limitations. For example, our method only focuses on image hair editing, and cannot handle facial hair or coherent video hair editing. Moreover, our method cannot perfectly transfer the reference color for some cases (e.g., slight color bias in Fig. 3), especially when the lighting is very different. Lastly, for some conditions our method still gets the proxy by optimization, thus real-time generation of all proxies is the future research direction.

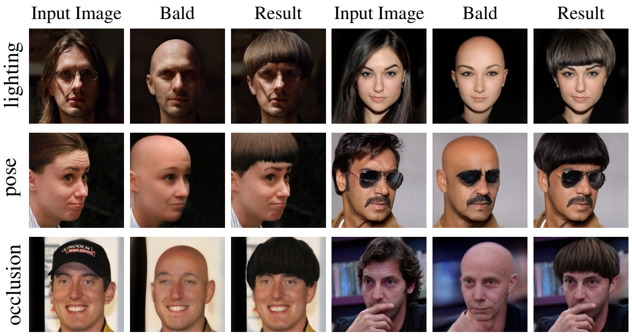

Figure 2. Ablation analysis on the robustness of balding technique. The text description is "*Bowl Cut Hairstyle*".

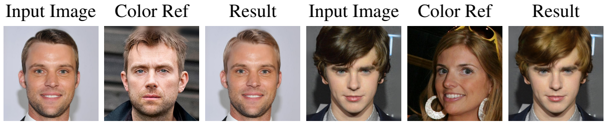

Figure 3. Failure cases.

5. More Qualitative Results

In Figures 4, 5, 6, 7, 8, and 9 we give more visual comparison results with other methods and our results for the comprehensive cross-modal conditional inputs.

References

[1] 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.

[2] 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.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[5] Ziwei Liu. https://github.com/switchablenorms/CelebAMask-HQ/tree/master/face_parsing. Accessed: Mar. 2023. [Online].

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

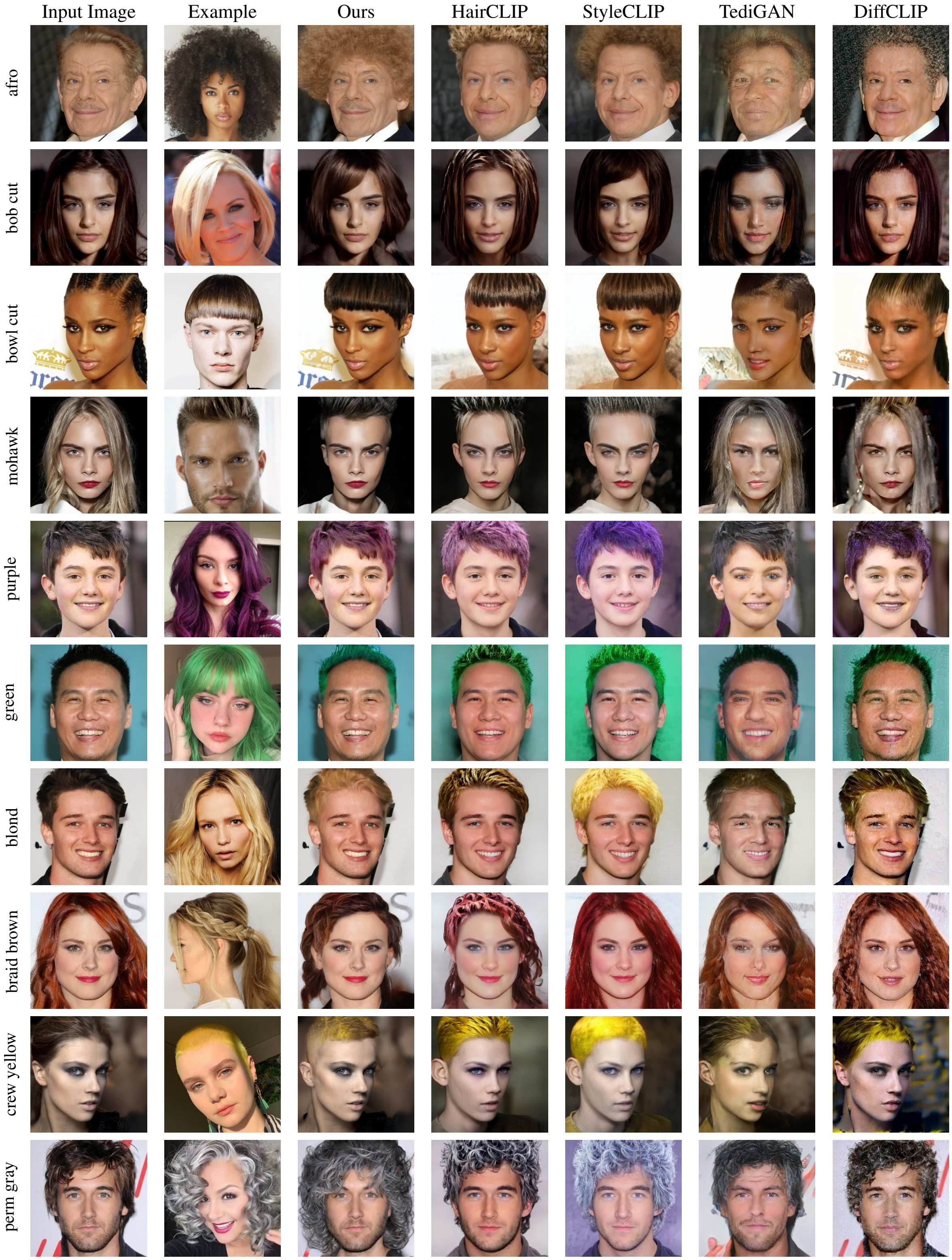

Figure 4. Visual comparison with HairCLIP [9], StyleCLIP-Mapper [6], TediGAN [11] and DiffusionCLIP [1]. The simplified text descriptions (editing hairstyle, hair color, or both of them) are listed on the leftmost side. We additionally provide an example image for each description for better comparison. Our approach demonstrates better editing effects and irrelevant attribute preservation (e.g., identity, background, etc.).

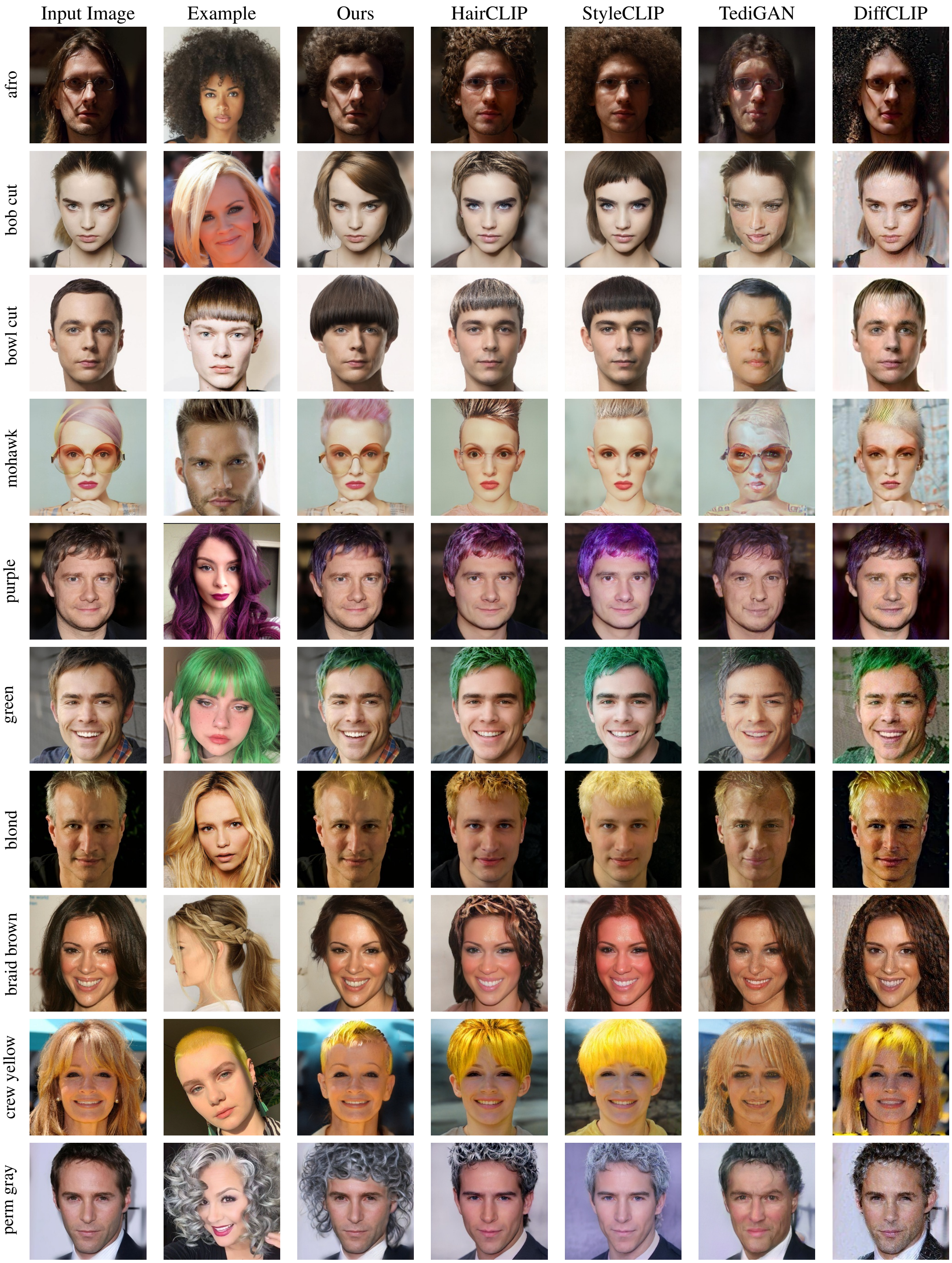

Figure 5. Visual comparison with HairCLIP [9], StyleCLIP-Mapper [6], TediGAN [11] and DiffusionCLIP [1]. The simplified text descriptions (editing hairstyle, hair color, or both of them) are listed on the leftmost side. We additionally provide an example image for each description for better comparison. Our approach demonstrates better editing effects and irrelevant attribute preservation (e.g., identity, background, etc.).

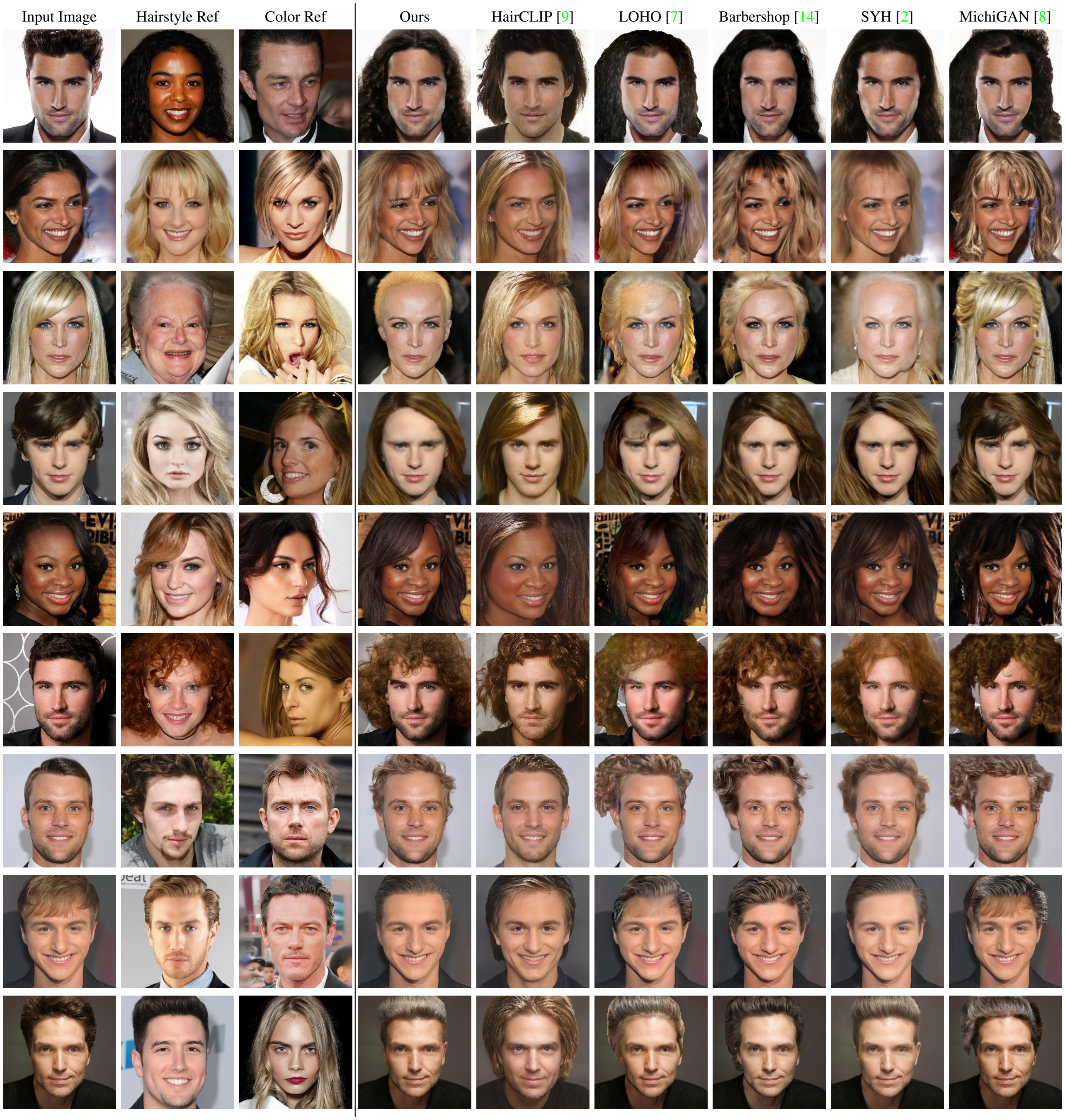

Figure 6. Visual comparison with HairCLIP [9], LOHO [7], Barbershop [14], SYH [2] and MichiGAN [8] on hair transfer. Only our method and SYH can accomplish unaligned hair transfer while keeping irrelevant attributes unmodified.

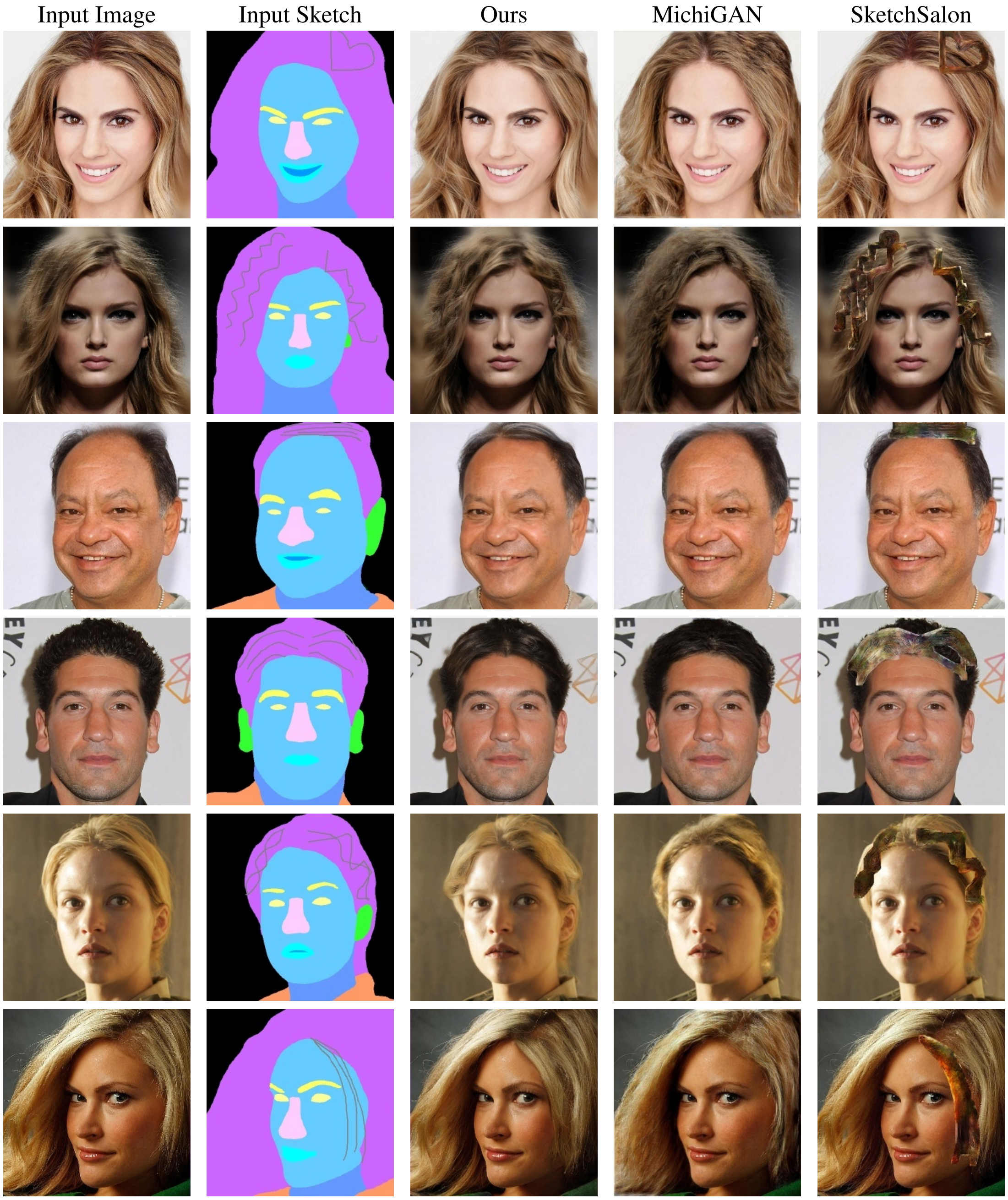

Figure 7. Qualitative comparison with MichiGAN [8] and SketchSalon [12] on sketch-based local hair editing. We provide sketches drawn in the facial parsing map for better visualization.

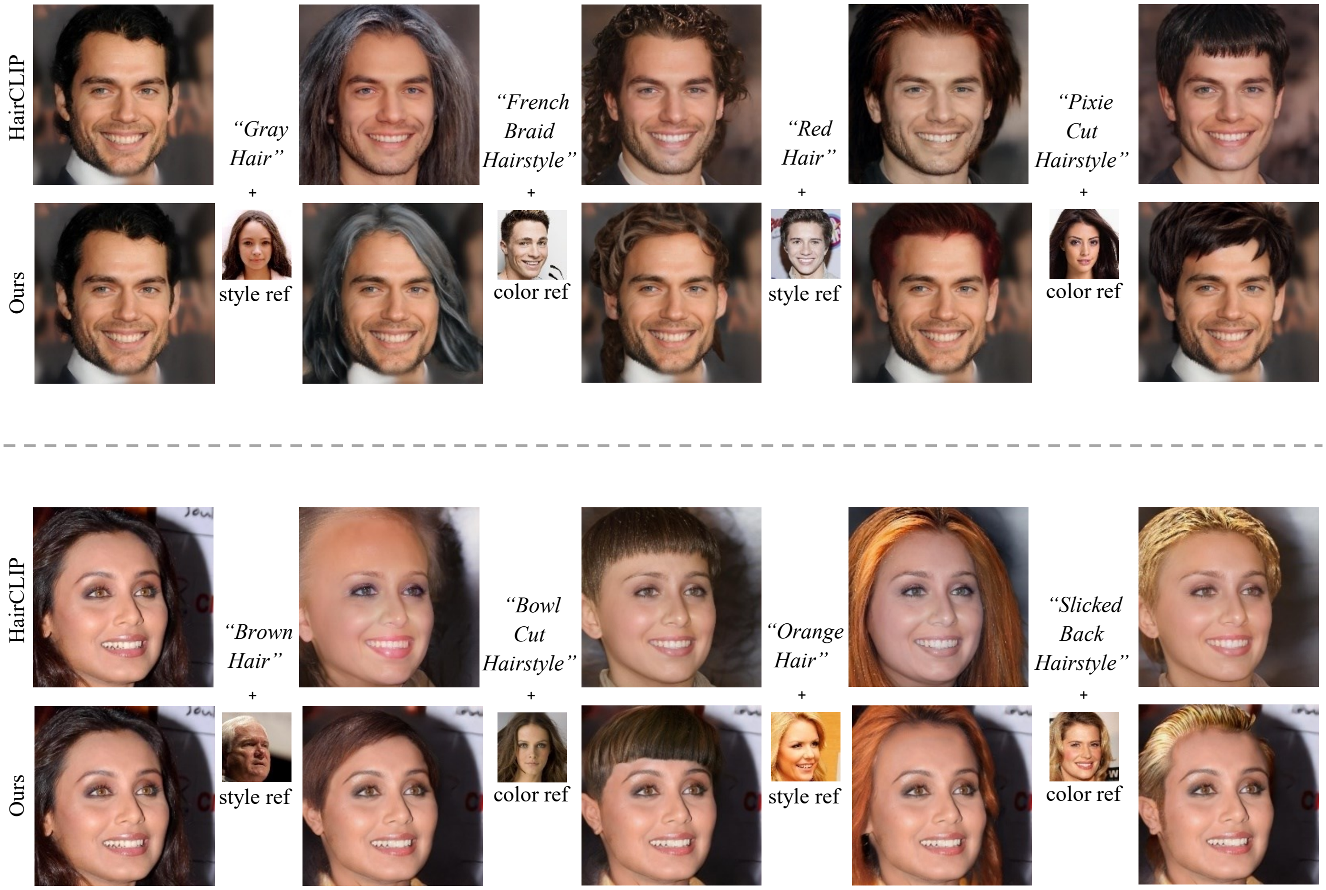

Figure 8. Qualitative comparison with HairCLIP on cross-modal conditional input setting. Our approach exhibits better editing effects and excellent preservation of irrelevant attributes. The first column are the input images.

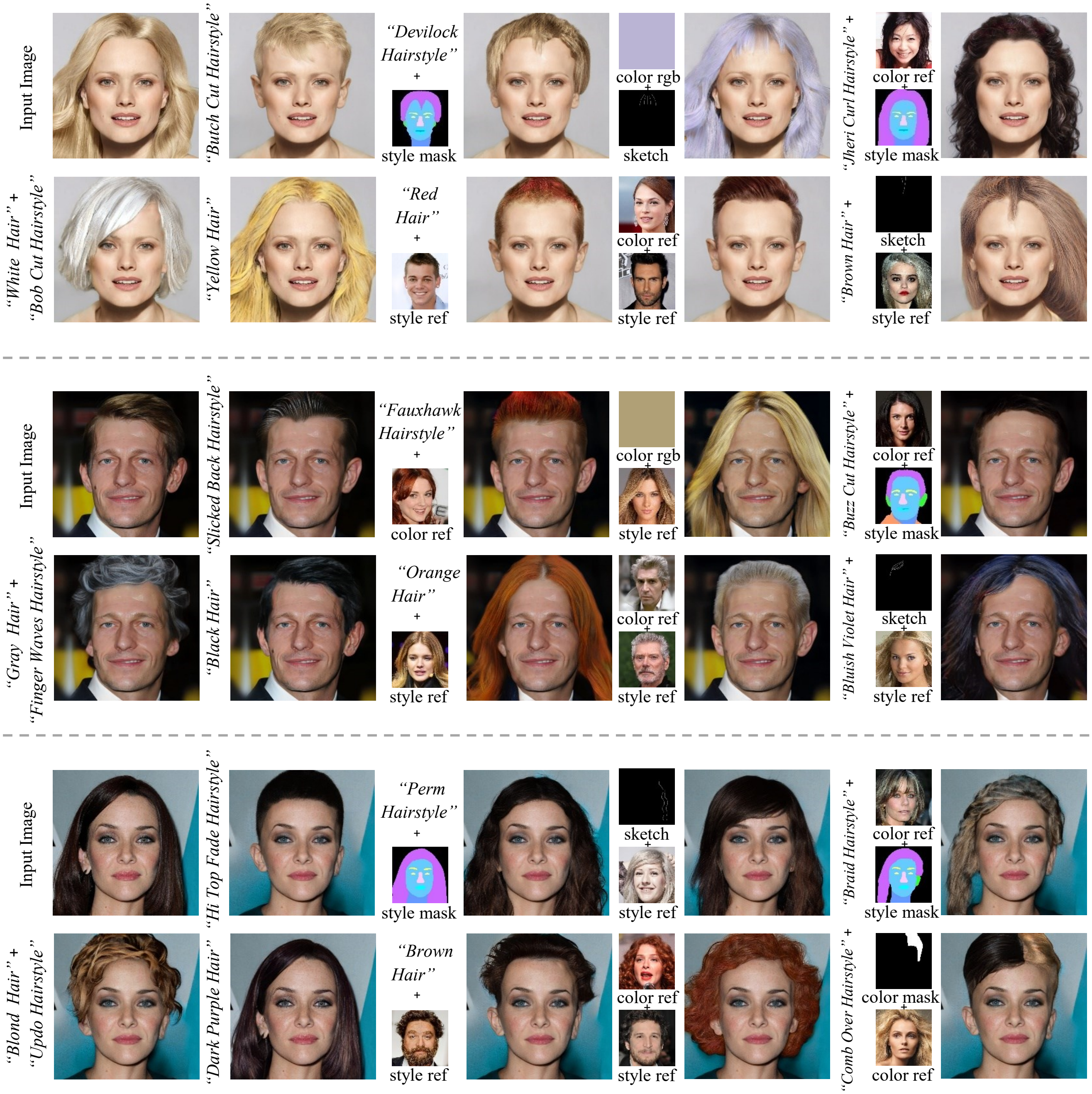
Input Image
"Butch Cut Hairstyle"
"Devilock Hairstyle" + style mask
color rgb + sketch
"Jheri Curl Hairstyle" + color ref + style mask
"White Hair" + "Bob Cut Hairstyle"
"Yellow Hair"
"Red Hair" + style ref
color ref + style ref
"Brown Hair" + sketch + style ref
Input Image
"Slicked Back Hairstyle"
"Fauxhawk Hairstyle" + color ref
color rgb + style ref
"Buzz Cut Hairstyle" + color ref + style mask
"Gray Hair" + "Finger Waves Hairstyle"
"Black Hair"
"Orange Hair" + style ref
color ref + style ref
"Bluish Violet Hair" + sketch + style ref
Input Image
"Hi Top Fade Hairstyle"
"Perm Hairstyle" + style mask
sketch + style ref
"Braid Hairstyle" + color ref + style mask
"Blond Hair" + "Updo Hairstyle"
"Dark Purple Hair"
"Brown Hair" + style ref
color ref + style ref
"Comb Over Hairstyle" + color mask + color ref

Figure 9. HairCLIPv2 supports hairstyle and color editing individually or jointly with unprecedented user interaction mode support, including text, mask, sketch, reference image, etc.