ECCV 2020

Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches

Shuai Yang Zhangyang Wang Jiaying Liu Zongming Guo

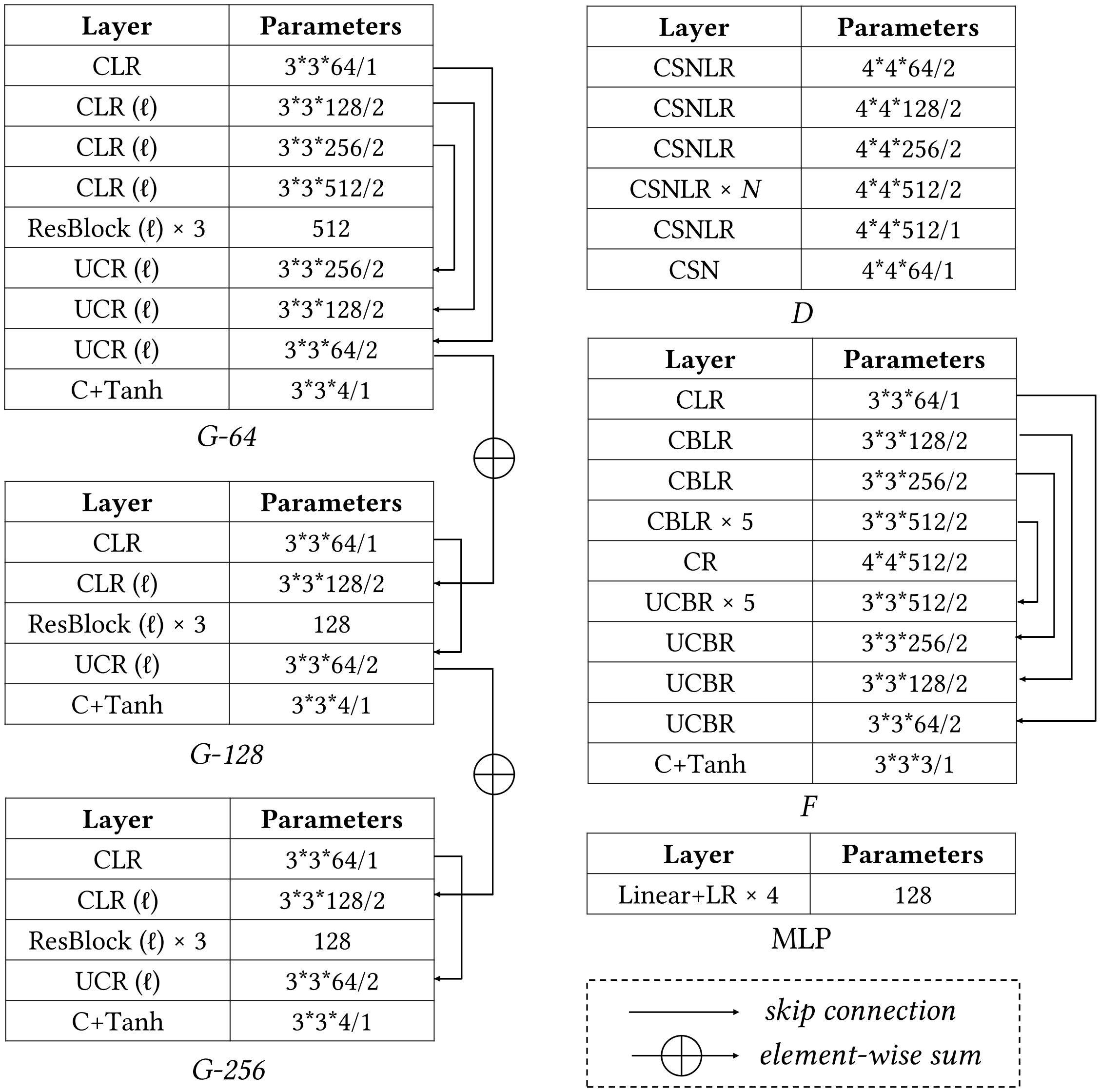
Supplementary Material

As supplementary material of our paper, we present the following contents:

- Detailed network architectures. (Fig. 1)
- Comparison with state-of-the-art methods. (Figs. 2−5)
- Comparison with ContextualGAN. (Fig. 6)
- Quantitative evaluation. (Tables 1−2, Fig. 7)
- Ablation study. (Figs. 8−11)
- Robust image synthesis on human-drawn sketches. (Figs. 12-14)
- Image editing performance without and with user guidance. (Fig. 15)
- Image editing performance on face attributes. (Fig. 16)
- Image editing performance on object removal. (Fig. 17)
- Image editing performance on deep plastic surgery. (Fig. 18)
- Image editing performance on pose and gender transfer. (Fig. 19)
- Image editing performance with spatially non-uniform refinement. (Fig. 20)

1. Detailed network architectures

Network architecture. Our generator G utilizes the fully convolutional Encoder-ResBlocks-Decoder architecture as in [3]. The discriminator D follows the SN-PatchGAN [15] for stable and fast training. Finally, we use pix2pix [1] as our edge-based baseline model F. F is trained with a discriminator whose architecture is based on D with an additional linear layer to map the output tensor to a score. The detailed network architectures are shown in Fig. 1, where "C" denotes a Convolution layer, "CSN" denotes a Convolution-SpectralNorm layer [7], "CB" denotes a Convolution-BatchNorm layer, the prefix "U" denotes an Upsampling layer, the suffix "R" and "LR" denote ReLU and LeakyReLU layers, respectively. Finally, we use "LayerName (ℓ)" to denote the convolutional layer is followed by the AdaIN layer for refinement level control. We use "$k * k * c/s$" to indicate that the convolutional layer has c filters with a spatial size of $k * k$ and stride s.



Layer	Parameters
CLR	3*3*64/1
CLR (ℓ)	3*3*128/2
CLR (ℓ)	3*3*256/2
CLR (ℓ)	3*3*512/2
ResBlock (ℓ) × 3	512
UCR (ℓ)	3*3*256/2
UCR (ℓ)	3*3*128/2
UCR (ℓ)	3*3*64/2
C+Tanh	3*3*4/1

G-64

Layer	Parameters
CLR	3*3*64/1
CLR (ℓ)	3*3*128/2
ResBlock (ℓ) × 3	128
UCR (ℓ)	3*3*64/2
C+Tanh	3*3*4/1

G-128

Layer	Parameters
CLR	3*3*64/1
CLR (ℓ)	3*3*128/2
ResBlock (ℓ) × 3	128
UCR (ℓ)	3*3*64/2
C+Tanh	3*3*4/1

G-256

Layer	Parameters
CSNLR	4*4*64/2
CSNLR	4*4*128/2
CSNLR	4*4*256/2
CSNLR × N	4*4*512/2
CSNLR	4*4*512/1
CSN	4*4*64/1

D

Layer	Parameters
CLR	3*3*64/1
CBLR	3*3*128/2
CBLR	3*3*256/2
CBLR × 5	3*3*512/2
CR	4*4*512/2
UCBR × 5	3*3*512/2
UCBR	3*3*256/2
UCBR	3*3*128/2
UCBR	3*3*64/2
C+Tanh	3*3*3/1

F

Layer	Parameters
Linear+LR × 4	128

MLP

Figure 1. Overview of network architectures.

Edge deformation and discarding. We use the sampling layer [9] to deform lines. We generate random offset maps via Gaussian noises and resample the input sketch based on the offset maps via the sampling layer. We randomly erase 0 to 3 rectangular regions with height and width in $[8, 24]$ to discard lines, which can improve the robustness.

Network training. The Adam optimizer is adopted with a fixed learning rate of 0.0002. We follow pix2pixHD [12] to use multi-scale generators and discriminators to gradually refine sketches from 64×64 to 128×128 to 256×256. Note that loss terms related to I_{out} are computed only in 256×256 resolution since we usually only have F for the target resolution available in real application. For the image resolution of 64, 128 and 256, the number N of "CSNLR" of discriminator D is set to 0, 1 and 2, respectively. For each resolution, we first train our network with $\ell = 1$ for 30 epoches, and then train with uniformly sampled $\ell \in [0, 1]$ for 200 epoches. The maximum allowable dilation radius is set to $R = 10$ for CelebA-HQ dataset [4] and $R = 4$ for CelebA dataset [5]. For all experiments, the weight for $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{perc}}$, $\mathcal{L}_G$ and $\mathcal{L}_{\text{D}}$ are 100, 1, 1 and 1, respectively. To calculate $\mathcal{L}_{\text{perc}}$, we use the conv2_1 and conv3_1 layers of the VGG19 [10] weighted by 1 and 0.5, respectively. For hinge loss, we set τ to 10 and 1 for G and F, respectively.

Training dataset. We use CelebA-HQ dataset [4] with edge maps extracted by HED edge detector [13] to train our model. We use PostprocessHED.m provided in pix2pix github page to simplify HED edges. The obtained binary lines are further smoothed by gaussian blur to simulate the real sketches. All images are resized to 256×256 pixels. We select the first 29K images for training and the remaining 1K images for testing. The masks are generated as the randomly rotated rectangular regions following [8], which teaches the network to handle all possible slopes to deal with arbitrarily shaped masks during testing. To make a fair comparison with ContextualGAN [6], we also train our model on CelebA dataset [5] preprocessed and provided by ContextualGAN [6], where images are of 64×64 size, 196K of which are for training and 1K for testing. In addition, we use handbag and shoe images provided by pix2pix [1] as non-human datasets.

Testing hand-drawn sketch dataset. We collected a human-drawn facial dataset with 30 challenging sketches for testing. We invited 20 users to draw facial sketches with mouses, Microsoft surface laptop, Wacom Bamboo Pen tablet and Ugee Pen tablet. Most users are amateurs. A total of 30 sketches are collected and are preprocessed through warping to align to the face images in CelebA-HQ dataset [4]. Specifically, we first obtain an average face of CelebA-HQ, and mark the eyes, nose and mouth with five points. Then users draw sketches given the five points as position reference. Finally, we manually align the sketches to the five points using the warping tool of PhotoShop. An overview of these sketches are shown in Fig. 12. We also test on 50 shoe images from Sketch dataset [11] in Figs. 13−14.

2. Comparisons with State-of-the-Art Methods

Face editing. Figs. 2–3 present the qualitative comparison on face editing with two state-of-the-art inpainting models: DeepFillv2 [15] and SC-FEGAN [2]. The released DeepFillv2 uses no sketch guidance, which means the reliability of the input sketch is set to zero ($\ell = \infty$). Despite being one of the most advanced inpainting models, DeepFillv2 sometimes fails to repair the fine-scale facial structures well, indicating the necessity of user guidance. SC-FEGAN, on the other hand, totally follows the inaccurate sketch and yields weird faces, due to unrealistic details contained in the rough sketches. Our method yields more natural and realistic facial details.

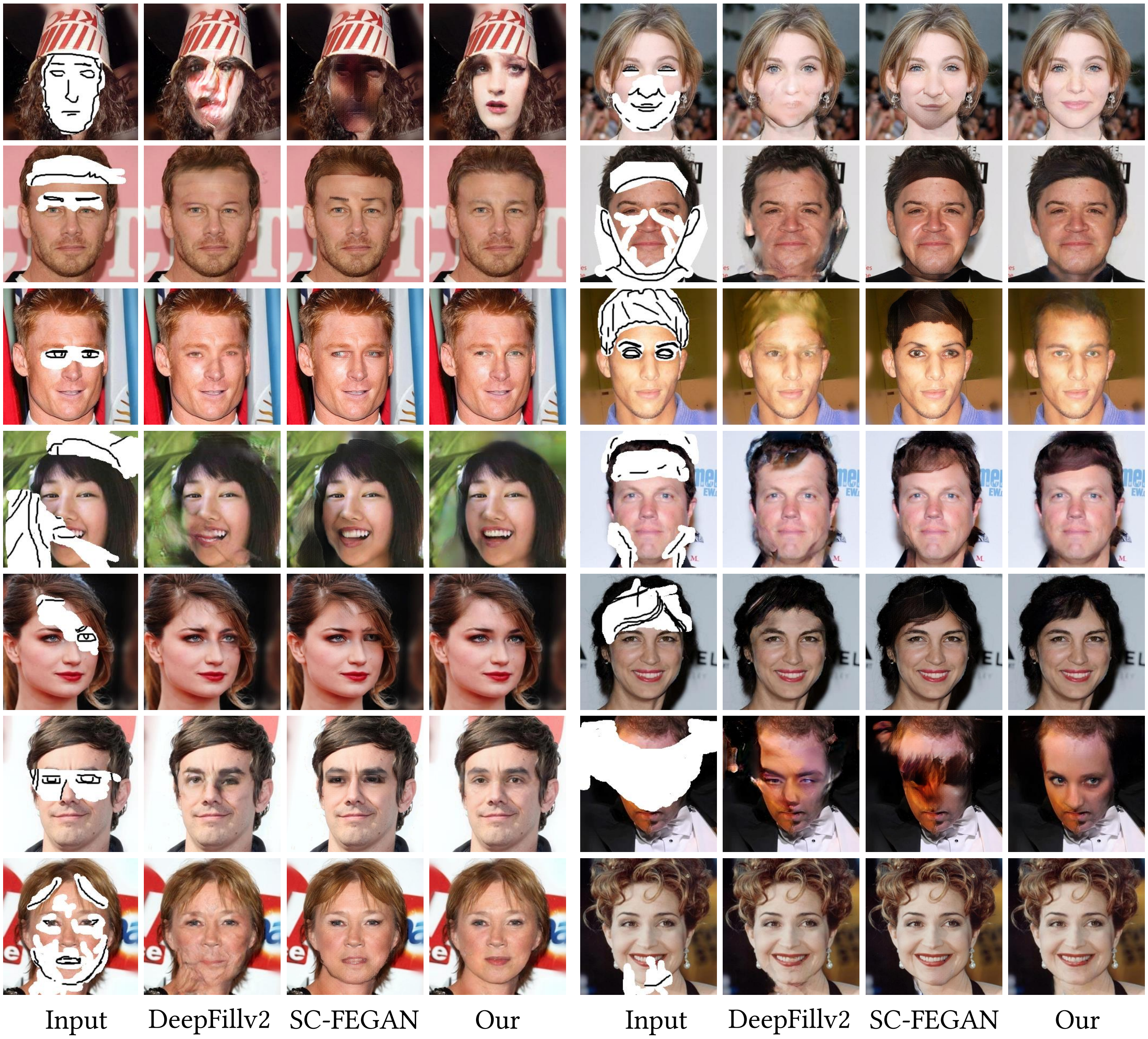


Figure 2. Comparison with state-of-the-art methods on face edting (Part I). For each group, from left to right: User inputs, results by DeepFillv2 [15], results by SC-FEGAN [2] and our results.

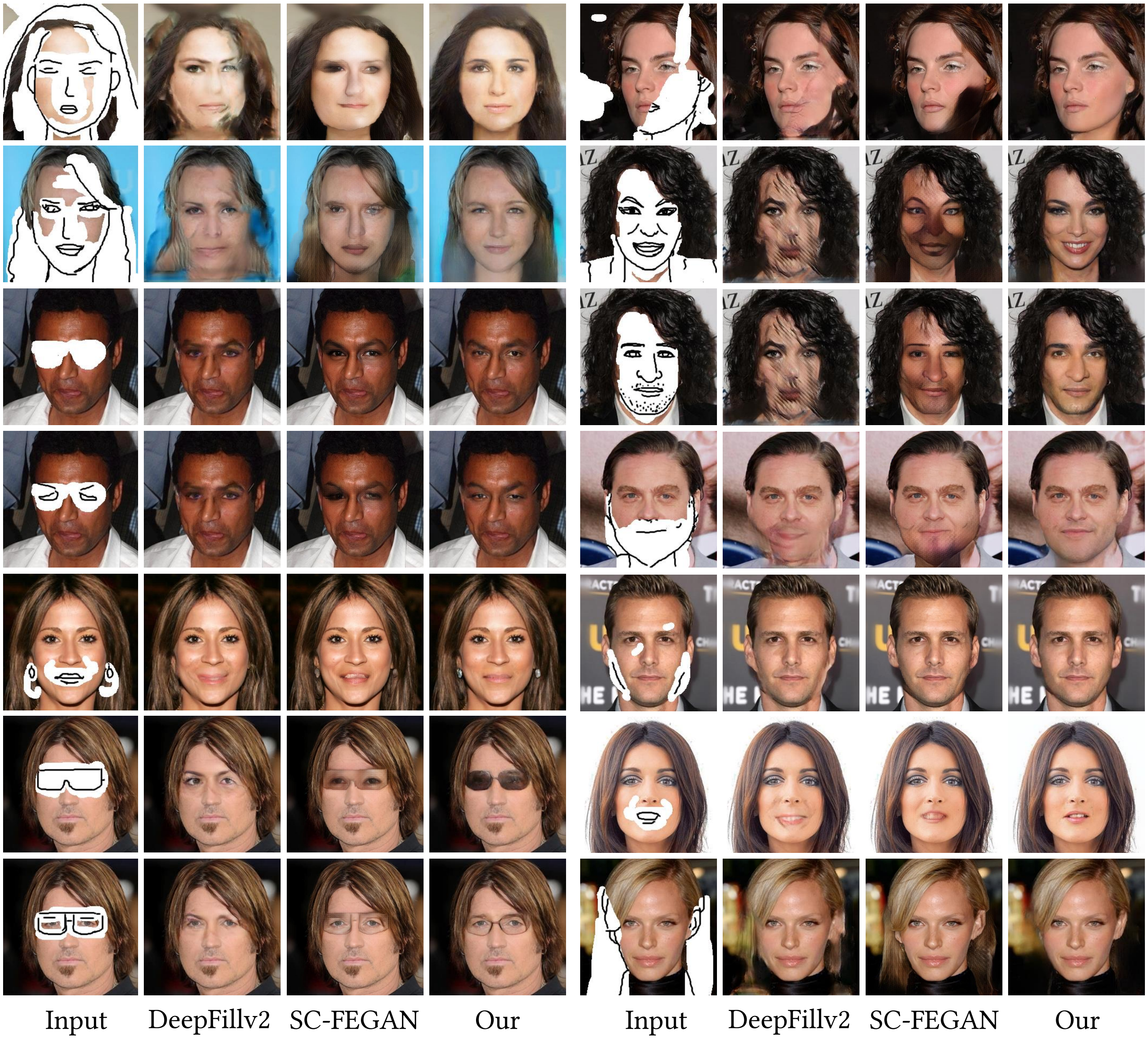


Figure 3. Comparison with state-of-the-art methods on face edting (Part II). For each group, from left to right: User inputs, results by DeepFillv2 [15], results by SC-FEGAN [2] and our results.

Face synthesis. Figs. 4–5 show the qualitative comparison on face synthesis with two state-of-the-art image-to-image translation models: BicycleGAN [16] and pix2pixHD [12]. As expected, both models synthesize facial structures that strictly match the inaccurate sketch inputs, producing poor results. Our model takes sketches as "useful yet flexible" constraints, and strikes a good balance between authenticity and consistency with the user guidance.

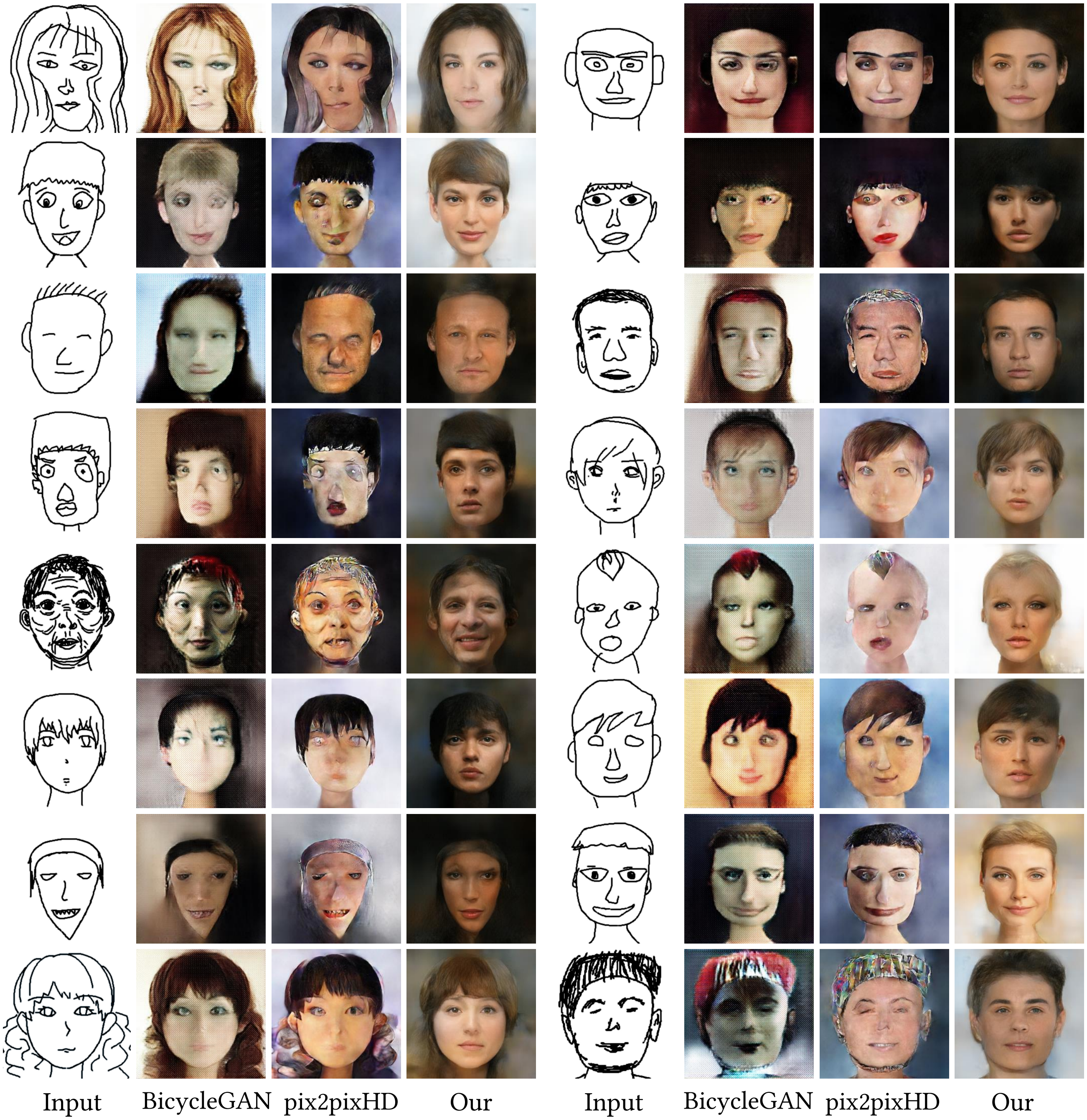


Figure 4. Comparison with state-of-the-art methods on face synthesis (Part I). For each group, from left to right: User inputs, results by BicycleGAN [16], results by pix2pixHD [12], and our results.

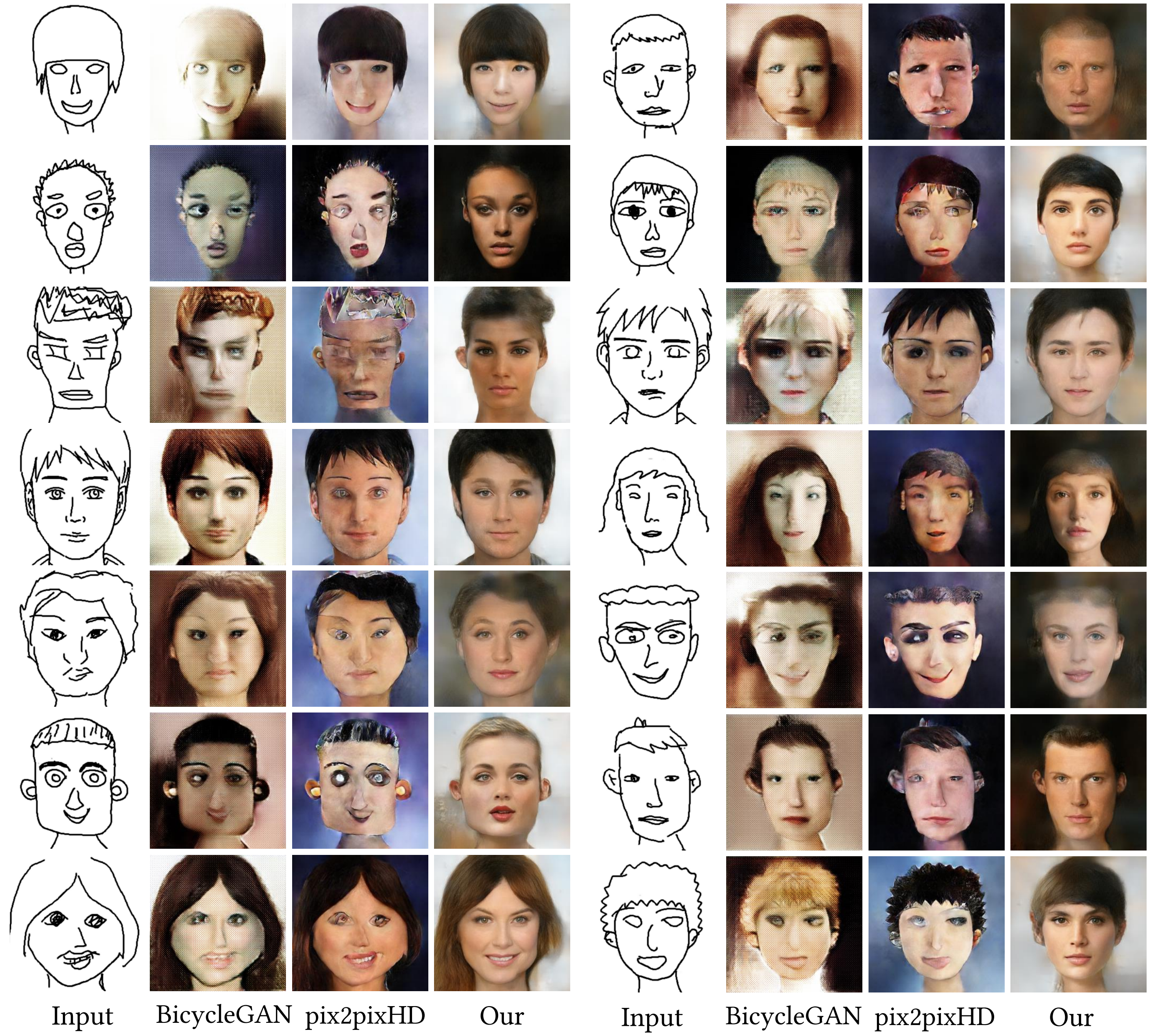


Figure 5. Comparison with state-of-the-art methods on face synthesis (Part II). For each group, from left to right: User inputs, results by BicycleGAN [16], results by pix2pixHD [12], and our results.

3. Comparison with ContextualGAN

Fig. 6 shows supplementary sketch-to-image translation results, where the results of ContextualGAN [6] are directly imported from the original paper. As can be seen, although realistic, ContextualGAN mainly searches from natural manifolds (with the sketch providing just a starting point) and produces differently from the sketch specification. Our method, by comparison, preserves some key attributes of the input sketch much better. We also find in the last two rows that ContextualGAN tends to generate attributes not appeared in the input such as hairs, while our method respects user input more. Hair will only be synthesized if it is drawn in the input.

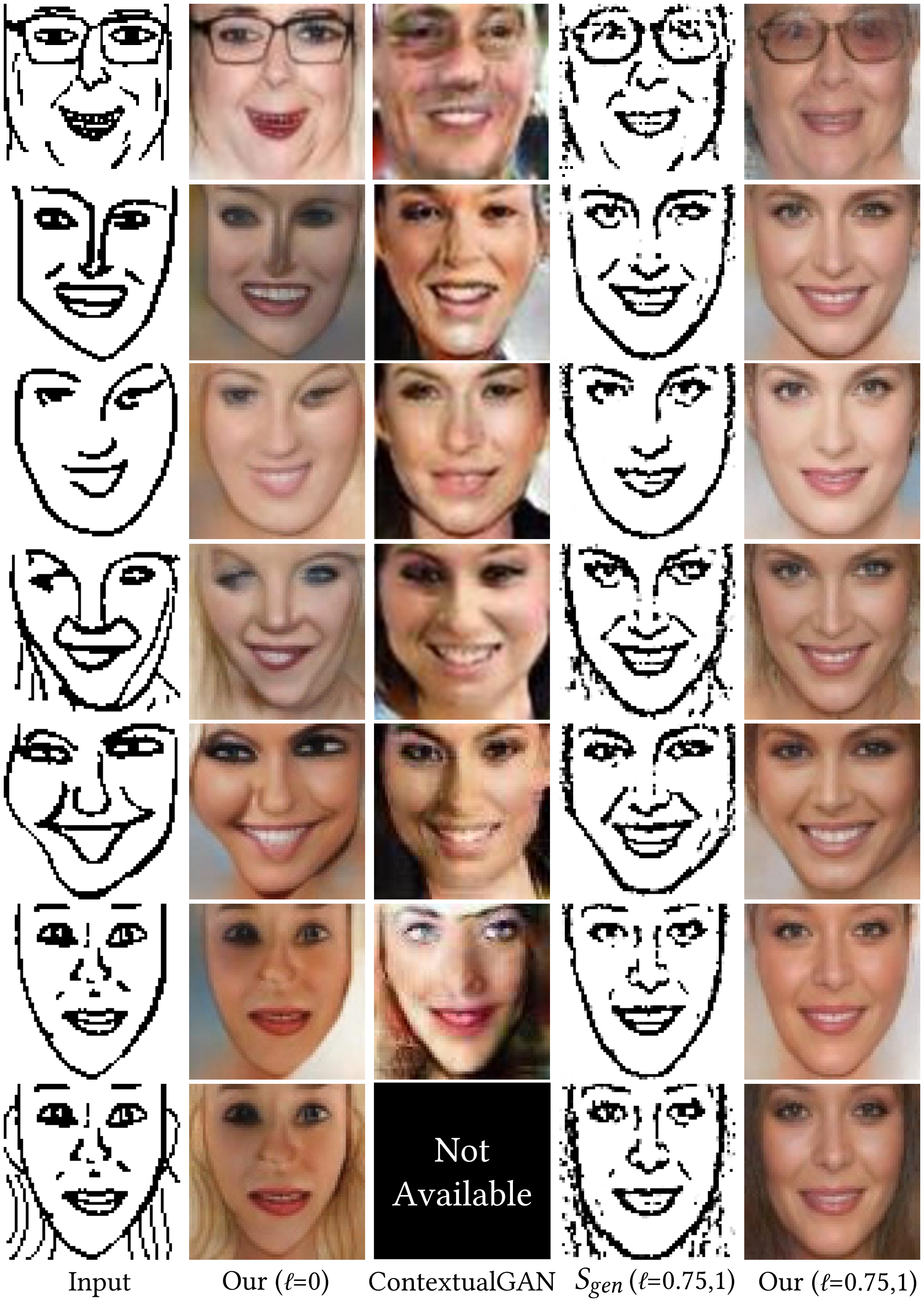


Figure 6. Comparison with ContextualGAN [6] on face synthesis. Top row uses $\ell = 0.75$, the following rows use $\ell = 1$.

4. Quantitative Evaluation

To better understand the performance of the compared methods, we perform user studies for quantitative evaluations. Participants are shown the 28 face editing and 38 face synthesis cases in Figs. 2–6 and Fig. 10(b) in the main paper. Each subject is asked to select one from the three results that best balances the sketch faithfulness with the output verisimilitude. To ensure the fairness, the orders of three methods randomly change every round. A total of 20 subjects participate in this study and a total of 1,160 selections are tallied. The preference ratio is used as the evaluation metrics. It is calculated by:

$$\text{preference ratio of Method } A = \frac{\text{The total number of times Method } A \text{ was selected}}{\text{The total selection number}}. \tag{1}$$

According to the definition, if Method A performs significantly better than all other methods, its mean preference ratio can reach 1.0. As shown in Table 1, for the task of face editing, the proposed method obtains the best average preference ratio of 0.730, while the average scores of DeepFillv2 [15] and SC-FEGAN [2] are 0.032 and 0.238, respectively. For the task of face synthesis, the proposed method achieves preference ratios above 0.5 in all cases, which means our method is steadily preferred by the users. The proposed method obtains the best average preference ratio of 0.945, while the average scores of BicycleGAN [16] and pix2pixHD [12] are 0.024 and 0.031, respectively. The proposed method and ContextualGAN [6] obtain 0.906 and 0.094, respectively. This user study quantitatively verifies the superiority of our method.

Table 1. User preference ratio of state-of-the-art methods. The best score in each row is marked in bold.

	Face Editing			Face Synthesis			Face Synthesis	
ID	DeepFillv2	SC-FEGAN	Ours	BicycleGAN	pix2pixHD	Ours	ContextualGAN	Ours
1	0.00	0.00	**1.00**	0.15	0.10	**0.75**	0.10	**0.90**
2	0.10	**0.45**	**0.45**	0.05	0.00	**0.95**	0.10	**0.90**
3	0.20	0.25	**0.55**	0.05	0.05	**0.90**	0.00	**1.00**
4	0.00	**0.50**	**0.50**	0.00	0.00	**1.00**	0.10	**0.90**
5	0.15	0.00	**0.85**	0.00	0.25	**0.75**	0.15	**0.85**
6	0.00	0.20	**0.80**	0.00	0.10	**0.90**	0.10	**0.90**
7	0.10	**0.65**	0.25	0.05	0.00	**0.95**	0.05	**0.95**
8	0.00	**0.65**	0.35	0.00	0.00	**1.00**	0.15	**0.85**
9	0.00	0.10	**0.90**	0.00	0.00	**1.00**		
10	0.00	**0.60**	0.40	0.00	0.00	**1.00**		
11	0.05	**0.60**	0.35	0.00	0.10	**0.90**		
12	0.00	0.00	**1.00**	0.00	0.00	**1.00**		
13	0.00	**0.60**	0.40	0.00	0.15	**0.85**		
14	0.00	**0.50**	**0.50**	0.00	0.00	**1.00**		
15	0.00	0.00	**1.00**	0.00	0.00	**1.00**		
16	0.00	0.00	**1.00**	0.00	0.00	**1.00**		
17	0.00	0.15	**0.85**	0.00	0.00	**1.00**		
18	0.00	**0.50**	**0.50**	0.05	0.00	**0.95**		
19	0.00	0.00	**1.00**	0.05	0.00	**0.95**		
20	0.05	0.10	**0.85**	0.00	0.00	**1.00**		
21	0.00	0.00	**1.00**	0.05	0.00	**0.95**		
22	0.05	0.45	**0.50**	0.00	0.00	**1.00**		
23	0.15	0.00	**0.85**	0.10	0.00	**0.90**		
24	0.00	0.10	**0.90**	0.05	0.00	**0.95**		
25	0.00	0.00	**1.00**	0.00	0.10	**0.90**		
26	0.00	0.05	**0.95**	0.05	0.05	**0.90**		
27	0.05	0.20	**0.75**	0.05	0.00	**0.95**		
28	0.00	0.00	**1.00**	0.00	0.00	**1.00**		
29				0.00	0.00	**1.00**		
30				0.00	0.10	**0.90**		
Average	0.032	0.238	**0.730**	0.024	0.031	**0.945**	0.094	**0.906**

We further conducted a quantitative experiment on simulated poor sketches, where the edge maps are distorted to simulate the drawing errors. Specifically, we randomly sample 100 test images from CelebA-HQ dataset [4] and distorted their edge maps with maximum allowable offset radius of 10 to form as poor sketch inputs. Then, we obtain the sketch-to-image translation results of BicycleGAN [16], pix2pixHD [12] and the proposed model under different refinement levels. Fig. 7 shows results on four simulated sketches. Finally, perceptual loss [3] is used to measure the difference between the generated image and the ground truth test image. The conv3_1 layer of the VGG19 [10] is used to calculate perceptual loss. Results are shown in Table 2. We achieve better results than BicycleGAN [16] and pix2pixHD [12]. In addition, our sketch refinement method successfully corrects some errors of the distorted sketches to make the generated results better approach the ground truth with large ℓ. The best score is obtained when $\ell = 0.6$. $\ell > 0.6$ does not bring additional gains because the goal of our sketch refinement method is to correct drawing errors rather than to approach the ground truth. Therefore, some structures might be better refined, but become inconsistent with the original edge maps.

Table 2. Perceptual loss on CelebA-HQ test sets.

Method	BicycleGAN	pix2pixHD	Ours					
			$\ell = 0$	$\ell = 0.2$	$\ell = 0.4$	$\ell = 0.6$	$\ell = 0.8$	$\ell = 1$
Score	447.6172	218.0837	201.6762	191.0261	183.7526	**179.6589**	180.2929	180.5574

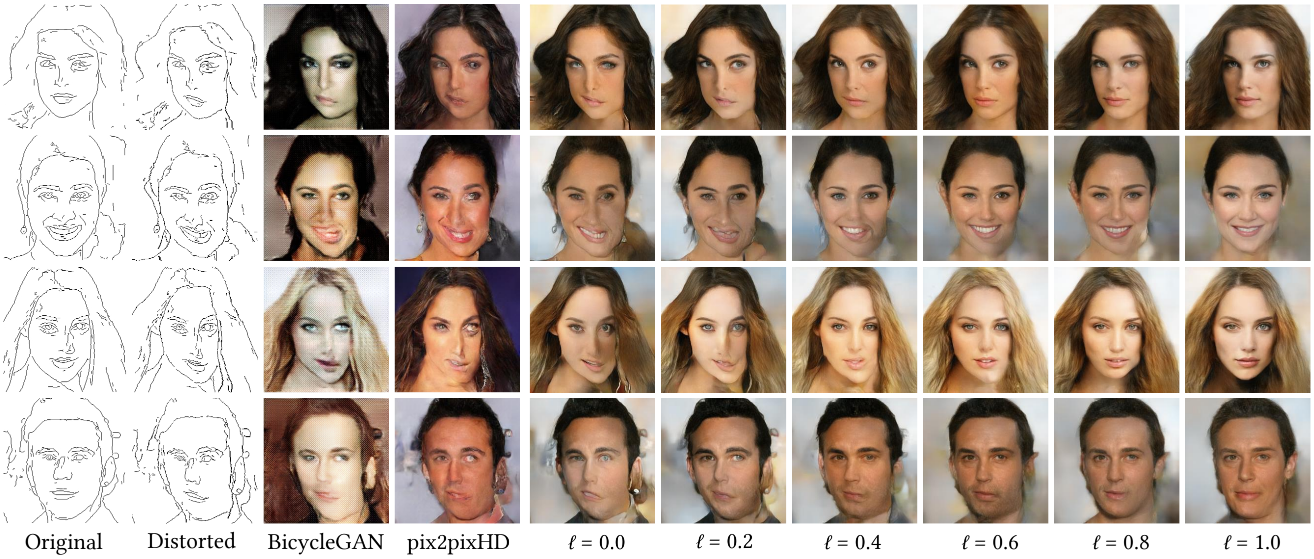


Figure 7. Visual comparison on simulated poor sketches from CelebA-HQ test sets.

5. Ablation Study

Rough sketch modelling. We examine the effectiveness of our dilation-based sketch modelling, which is the key of our sketch refinement. In Fig. 8, we perform a comparison between rough sketch data generation using dilation and gaussian blurring, both of which use a kernel radius of 21. No deformation and line discarding are applied. Although the blurring lines only provide rough information about the original edge, the network is powerful enough to recover it without any refinement. Through stricter edge dilation, the network begins to have the pressure to infer the extra facial details.

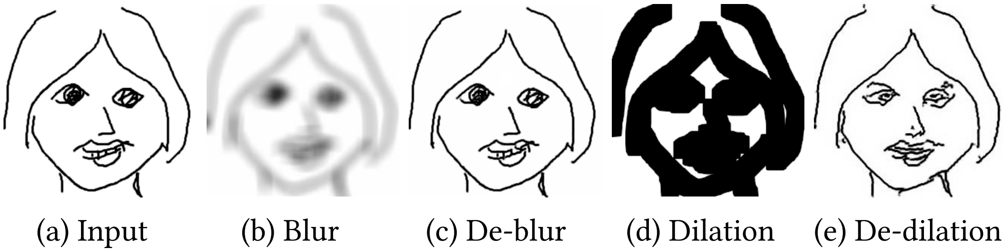

(a) Input (b) Blur (c) De-blur (d) Dilation (e) De-dilation

Figure 8. Effect of the dilation-based sketch modelling.

Training objective. To analyze our training objective, we design the following configurations:

- S2I: This baseline model is trained to directly map S_ℓ to I. With no intermediate sketch produced, it cannot serve as a plug-in for other edge-based models.
- S2S: This model is trained to map S_ℓ to S with adaptation to F. With no image produced, this model cannot be used independently of other edge-based models.
- S2SI: The proposed model. With both sketch and image produced, this model can be used independently or in combination with other edge-based models.

Fig. 9 displays the outputs of these models. The huge discrepancy between the rough sketch and the photo makes it hard to build their direct mapping, leading to the unrealistic results of S2I model. On the other hand, without the intermediate photos as perceptual guidance, S2S model fails to learn the correct structure from sparse lines, yielding unreasonable structures such as the small mouth. Our full model is guided by the complimentary constraints from both the edge and photo modalities, producing more natural results.

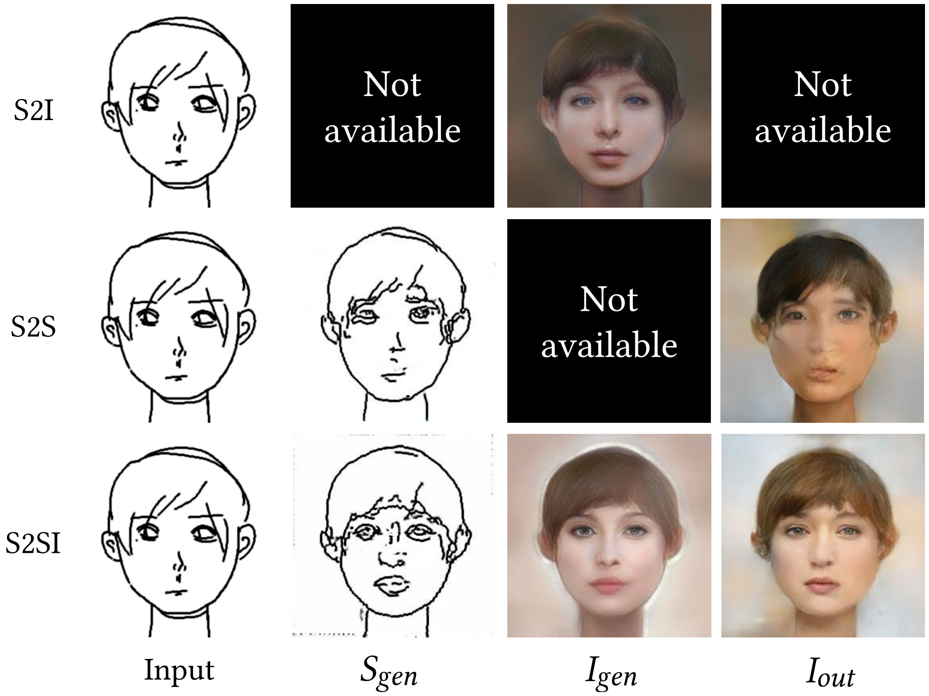


Figure 9. Effect of different training objectives.

Refinement level control. We compare the proposed style-based conditioning with label concatenation and controllable resblock [14] in Figs. 10−11. Label concatenation yields stacking lines like those in draft sketches. By comparison, controllable resblock interpolates the resblock features at two extremes to achieve level control, which generates cleaner lines but still rough facial details. Our style-based conditioning surpasses controllable resblock in adaptive channel-wise control, which provides strongest results in both well-structured sketches and realistic photos.

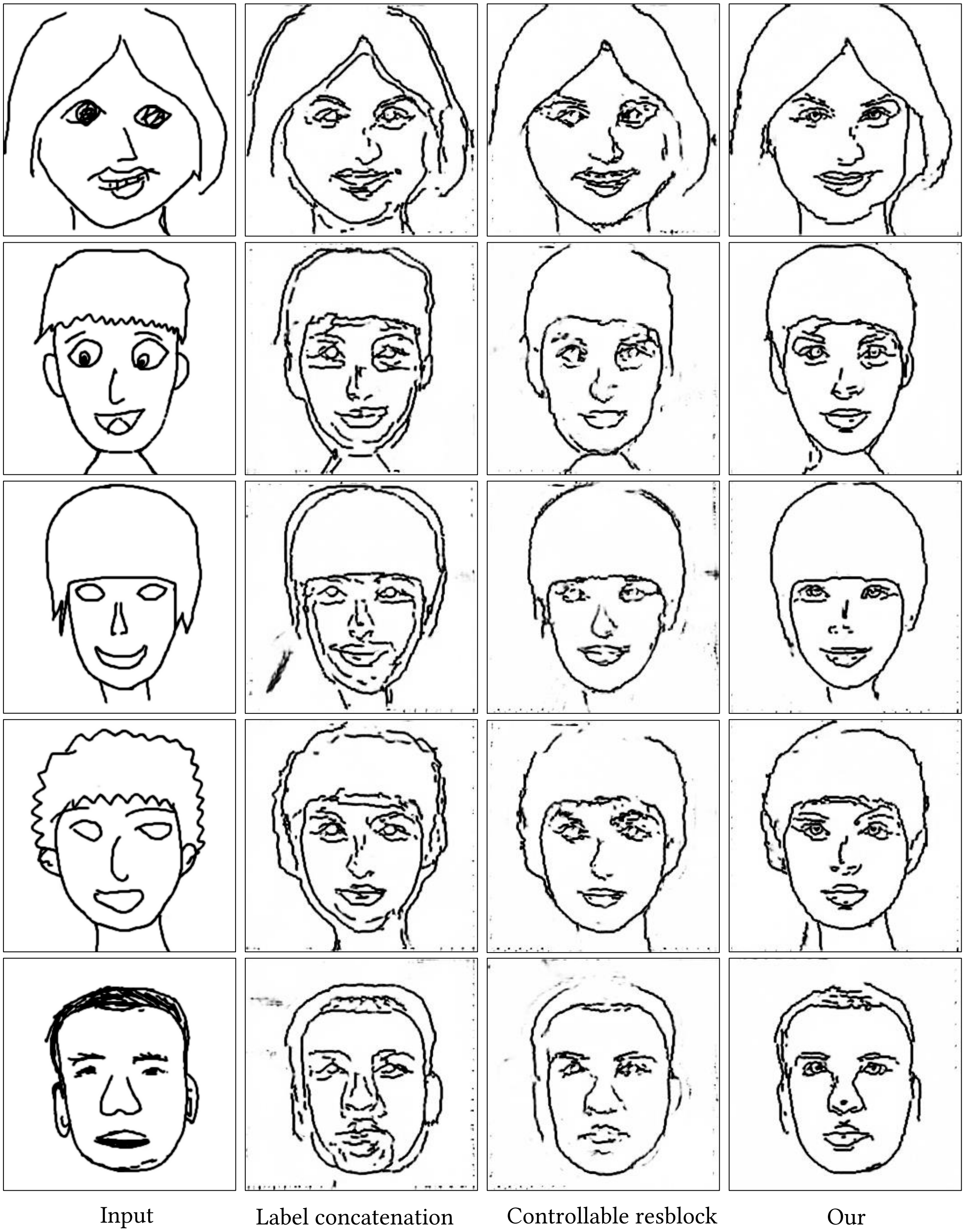


Figure 10. Visual comparison of the sketch refinement on label conditioning with the maximal refinement level $\ell = 1$.

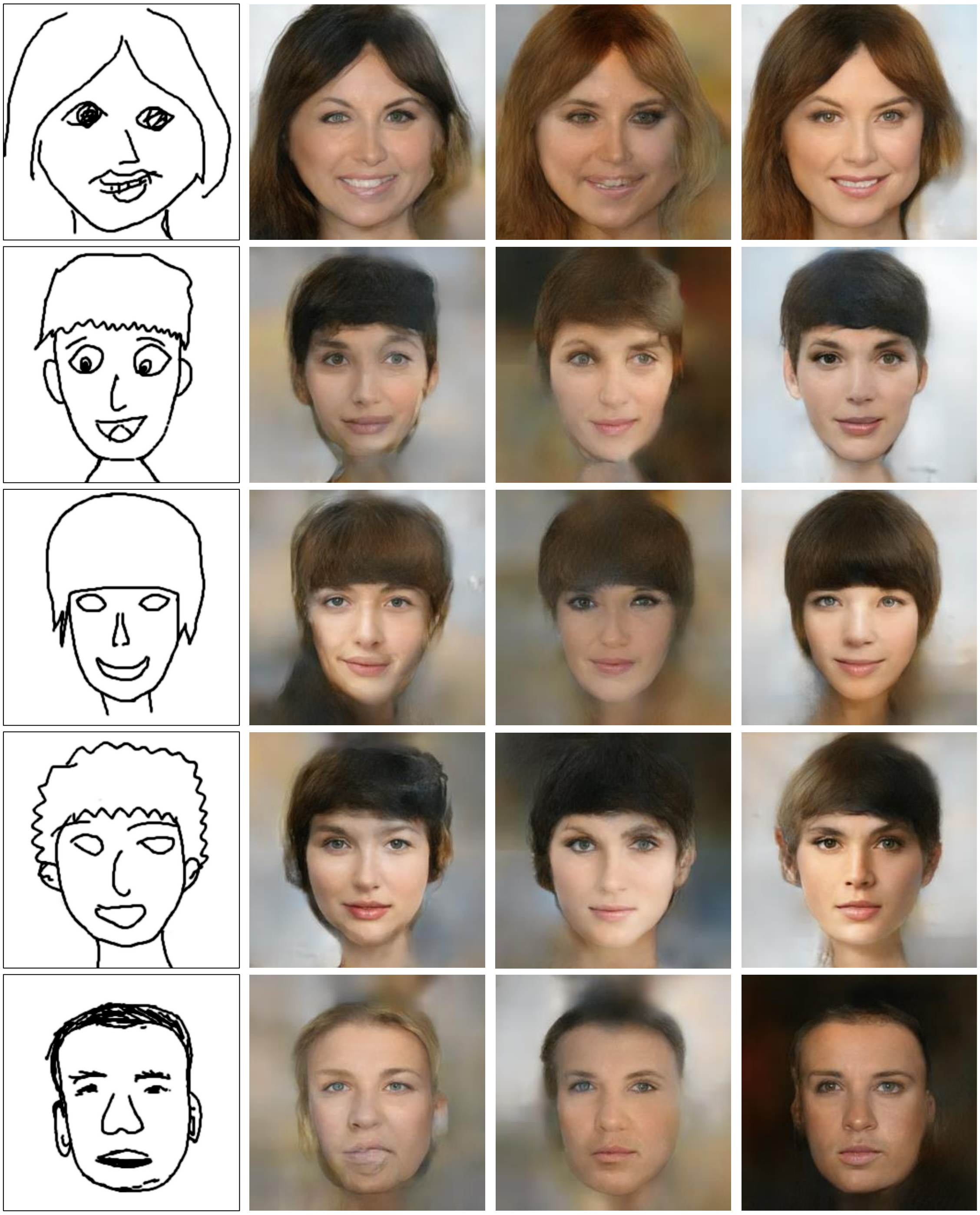


Figure 11. Visual comparison of the face synthesis on label conditioning with the maximal refinement level $\ell = 1$.

6. Robust Image Synthesis on Human-Drawn Sketches

Fig. 12 shows image synthesis results on our collected human-drawn facial dataset. As can be seen, these sketches demonstrate huge variety among different users and our model produces plausible results, showing its generalizability and robustness.

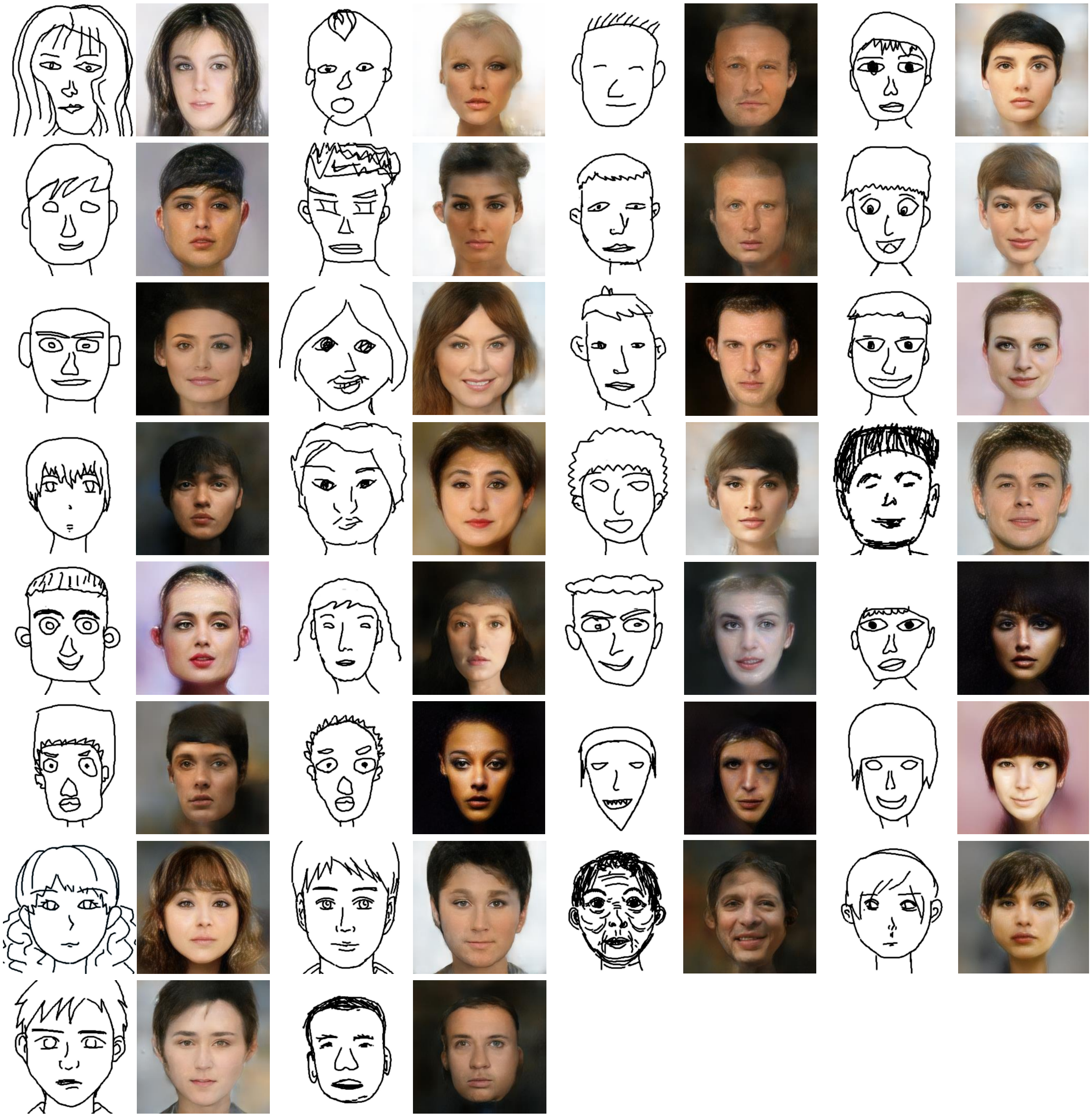


Figure 12. An overview of our collected hand-drawn sketches and our results.

We further test on 50 shoe sketches from the Sketch dataset [11]. As shown in Figs. 13-14, our method can effectively refine the shoe structures.

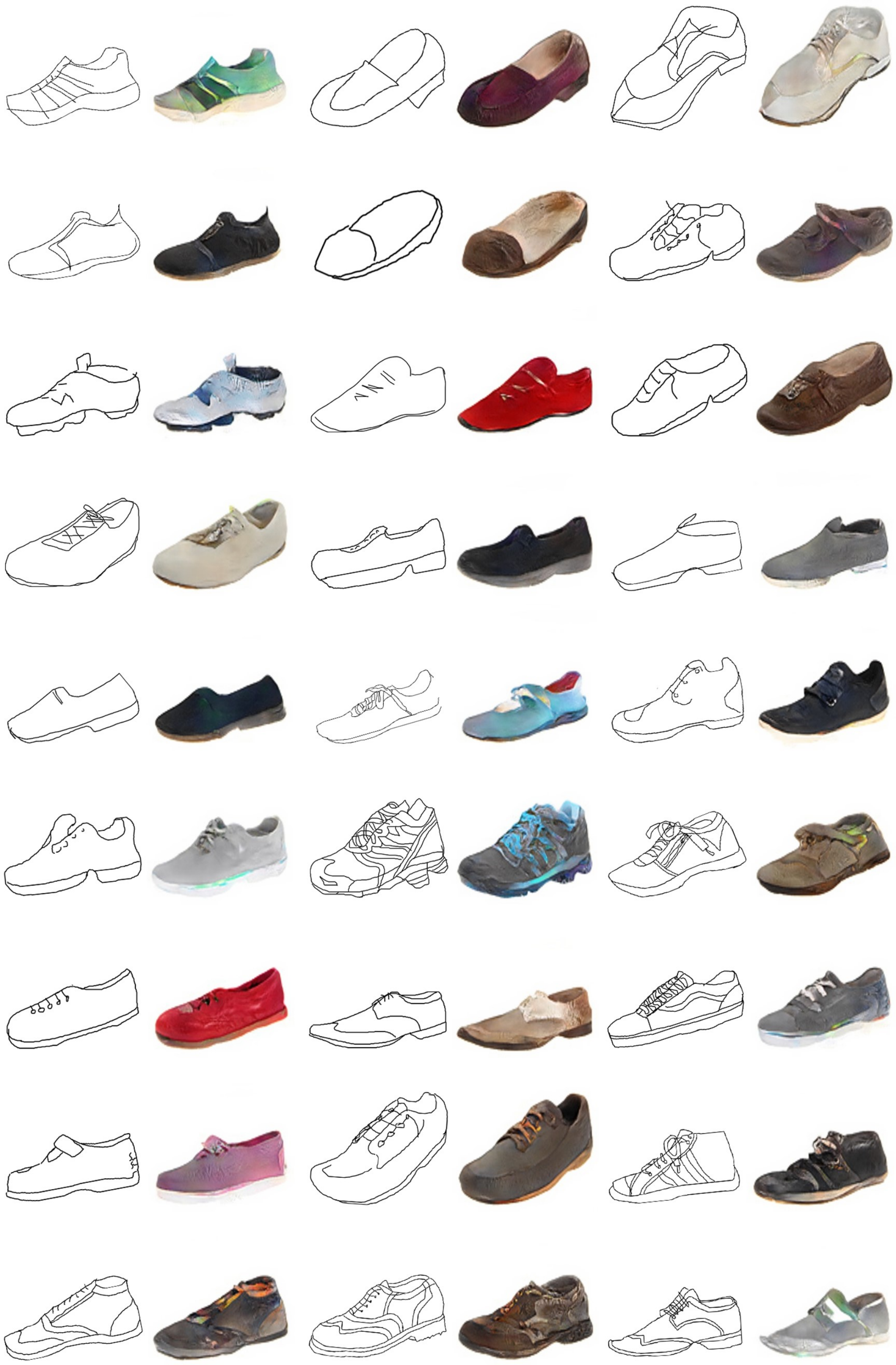


Figure 13. An overview of our image synthesis results on Sketch dataset [11] (Part I).

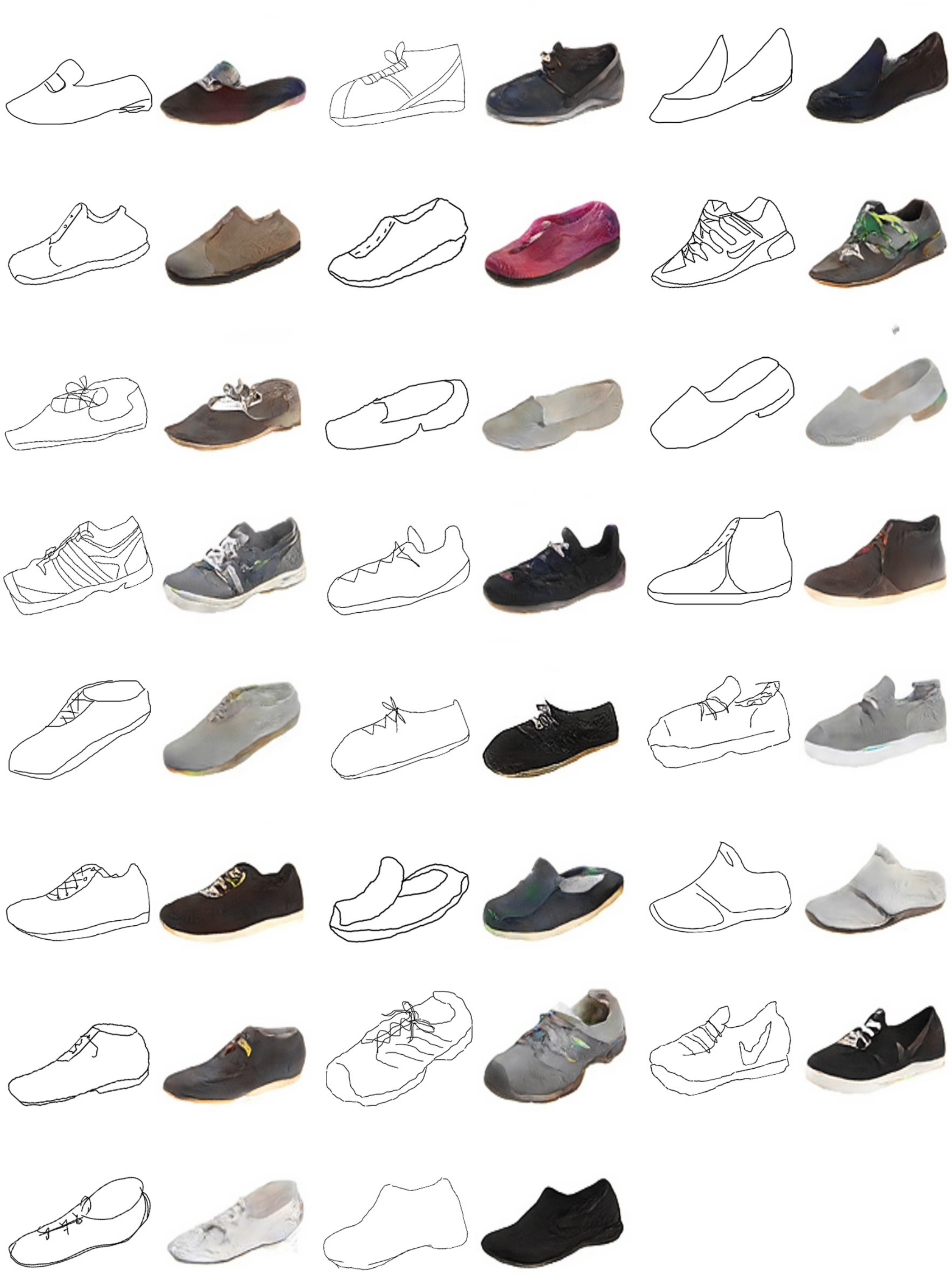


Figure 14. An overview of our image synthesis results on Sketch dataset [11] (Part II).

7. Image Editing Performance without and with User Guidance

In Fig. 15, we show the effects of user sketch guidance. Thanks to the line discarding process, our method can infer and complete the missing facial structures when using a large refinement level even without the user sketch inputs as shown in the first example. In the second and third rows, we show that our method can adjust the outputs based on the user inputs, therefore enabling user-customized editing. In the last two rows, we show an example of the necessity of the user guidance. In the absence of large facial regions, our method cannot well infer the large-scale facial structures even under a large refinement level. Through additional user guidance, the results are drastically improved.

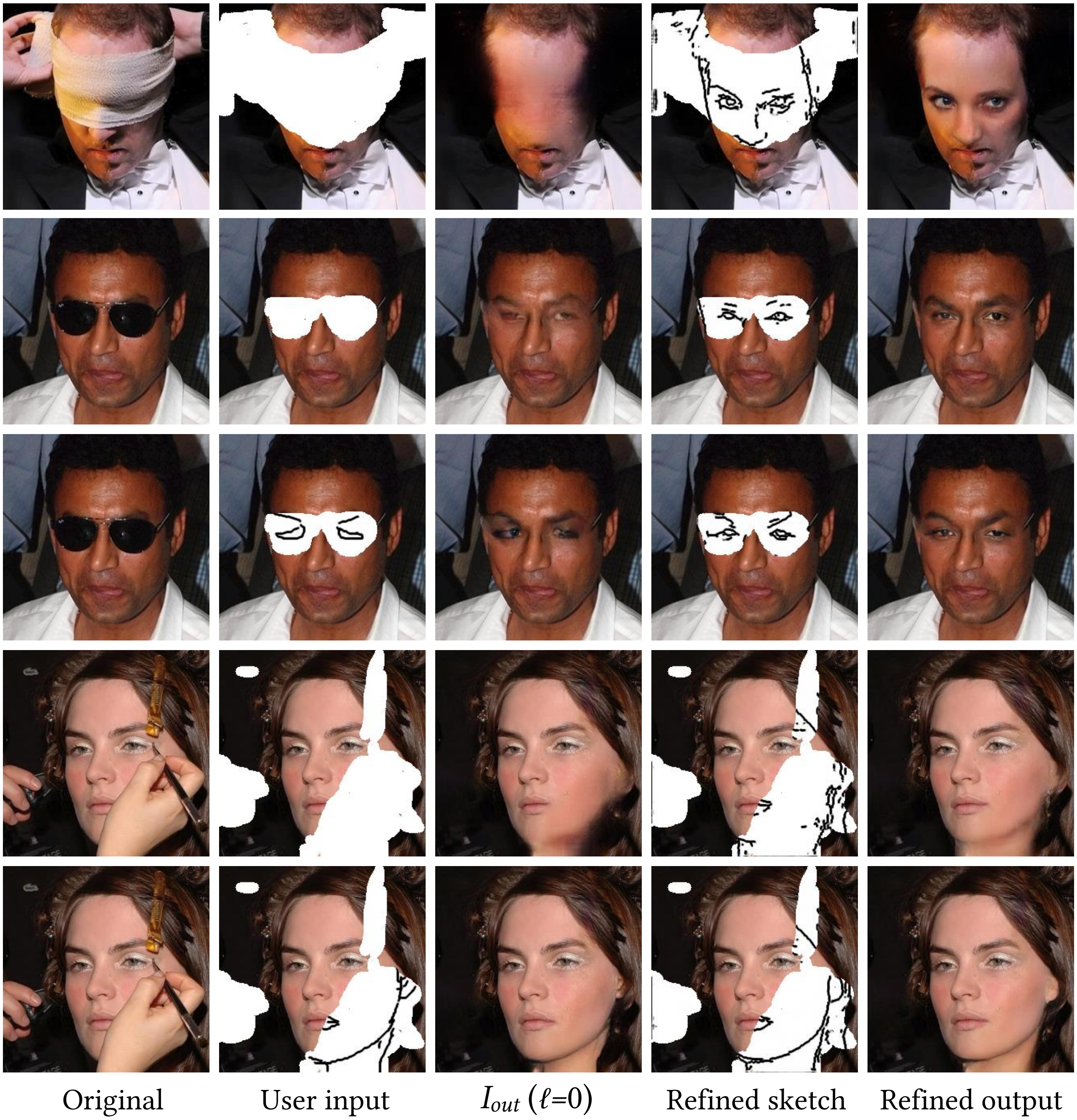


Figure 15. Image editing results with and without user sketch guidance.

8. Image Editing Performance on Faces

Fig. 16 presents our results on face editing to wear earrings, change hairstyles, change expressions, and wear glasses.

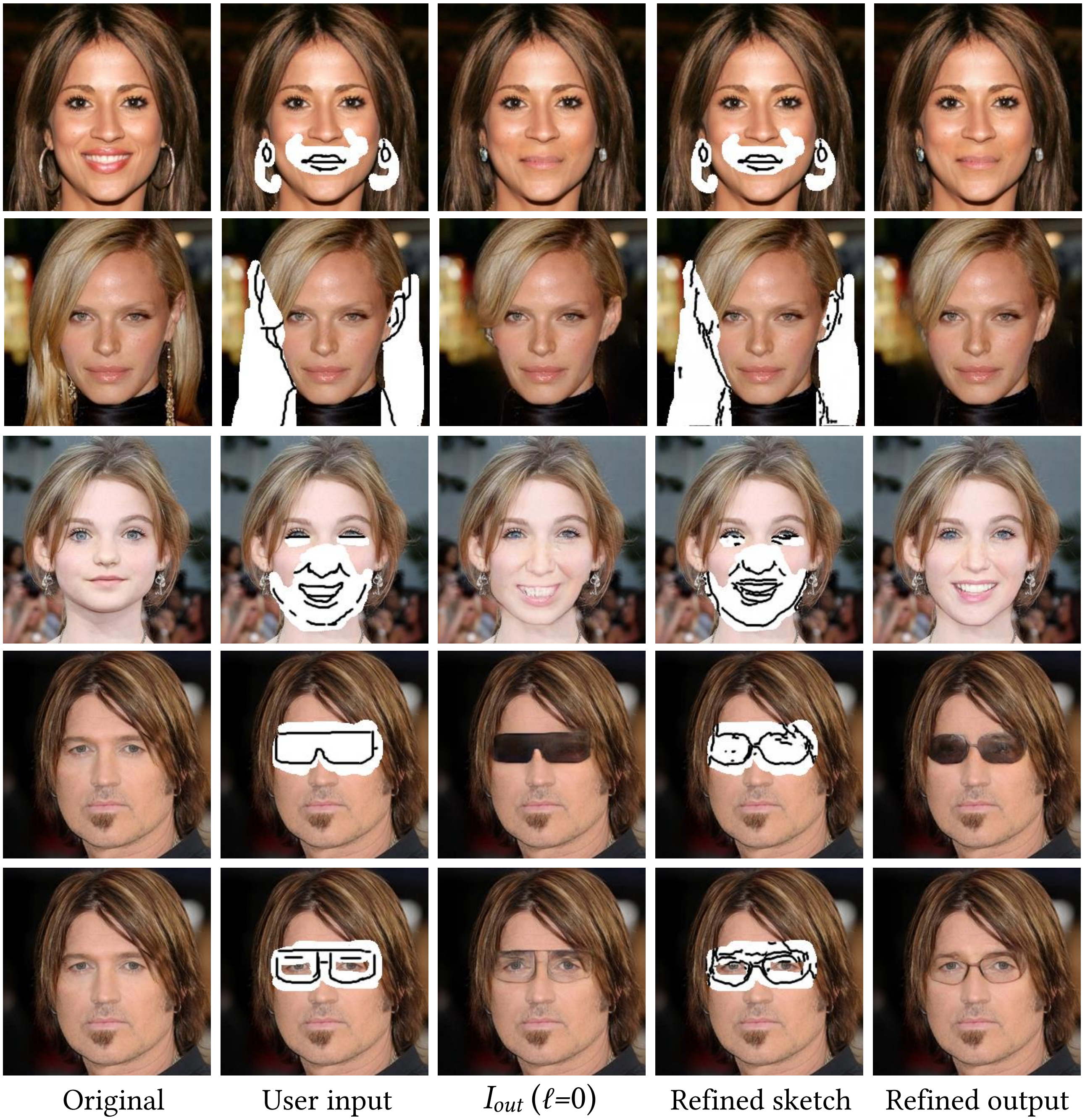


Figure 16. Image editing results for facial editing.

9. Image Editing Performance on Object Removal

As a special model of image inpainting, our model is inherently able to remove undesired objects, where the user guidance empowers the model to handle extremely complex scenes. As shown in Fig. 17, the masks, bangs, signatures, leaves and mustaches are successfully removed.

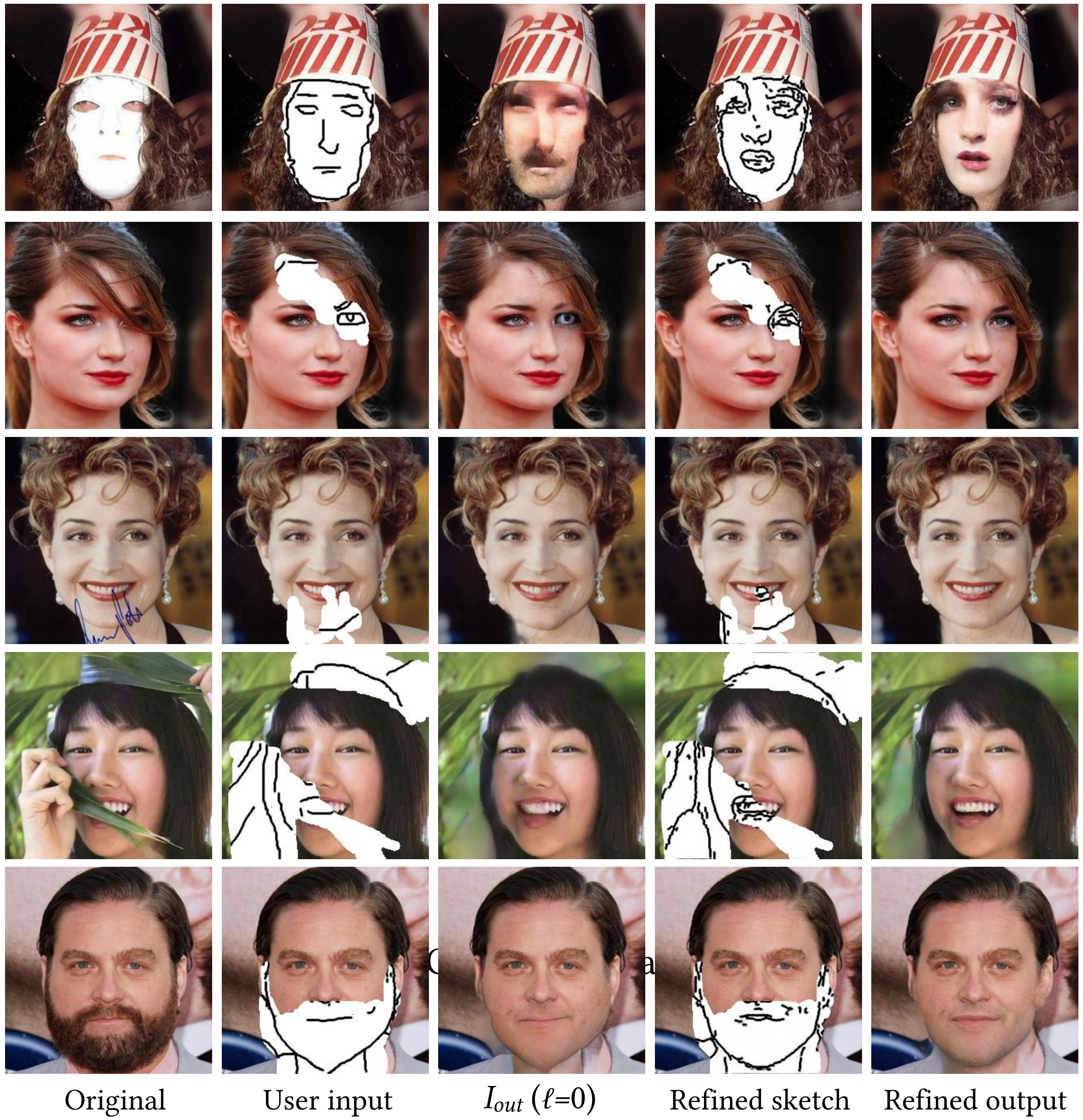


Figure 17. Image editing results for object removal.

10. Image Editing Performance on Deep Plastic Surgery

Fig. 18 presents our results on deep plastic surgery. Users can purposely perform “plastic surgery” digitally, such as widening eyes, clearing nevus, removing wrinkles, and lifting the eye corners. Alternatively, amateurs can intuitively edit the face with fairly coarse sketches to provide a general idea, such as face-lifting and bangs, and our model will tolerate the drawing errors and suggest a suitable “surgery” plan.

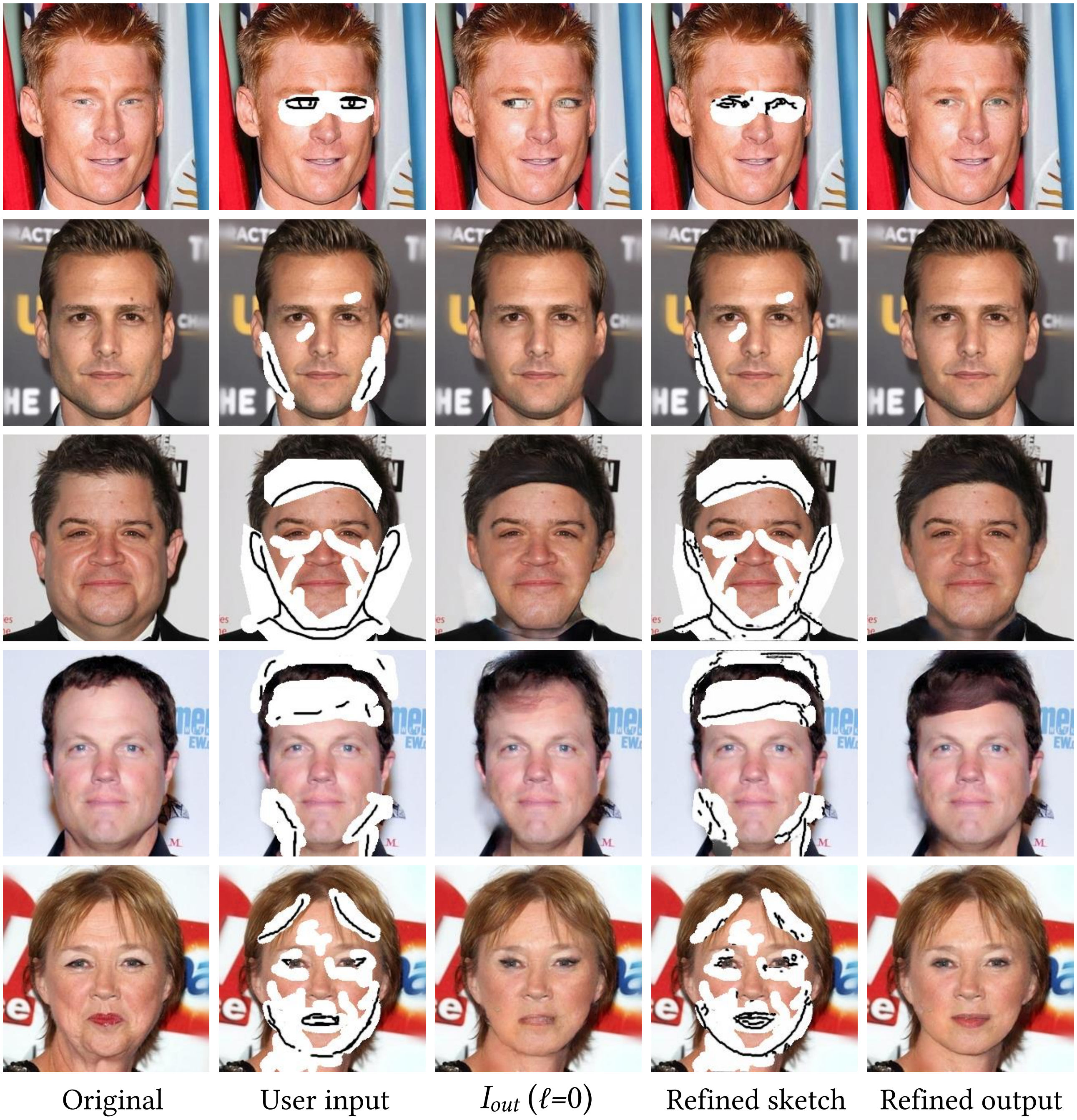


Figure 18. Image editing results for deep plastic surgery.

11. Image Editing Performance on Pose and Gender Transfer

Fig. 19 shows two challenging examples where a large portion of the face is masked out, which enables us to make some big adjustments such as modifying the pose and the gender.

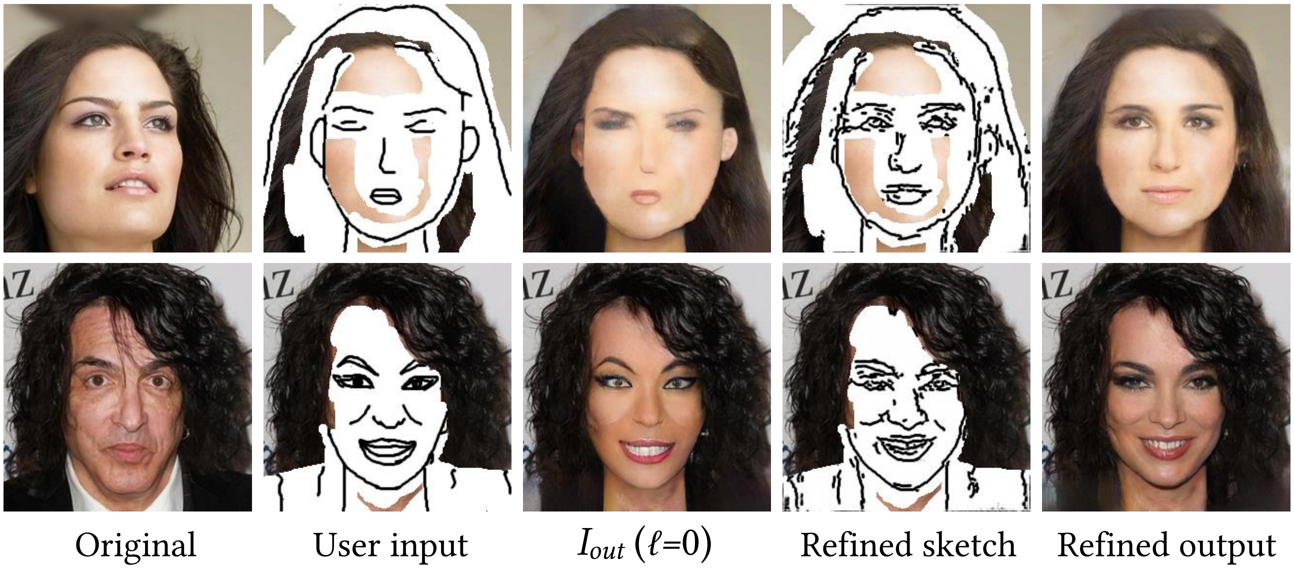


Figure 19. Image editing results for pose and gender transfer.

12. Image Editing Performance with Spatially Non-Uniform Refinement

Hand-drawn sketches can demonstrate huge variety among different users. Furthermore, even for the same user, the accuracy of the structure can vary within one sketch, demanding spatially non-uniform sketch refinement for more flexible controllability. Our model can be easily extended to spatially non-uniform refinement through multi-step refinement. As shown in Fig. 20, the user first uses a low ℓ on the whole mask region to better comply with the structure guidance of the nose, mouth and beard. Then, user can edit the mask and further improve the verisimilitude of the eye region with a high ℓ.

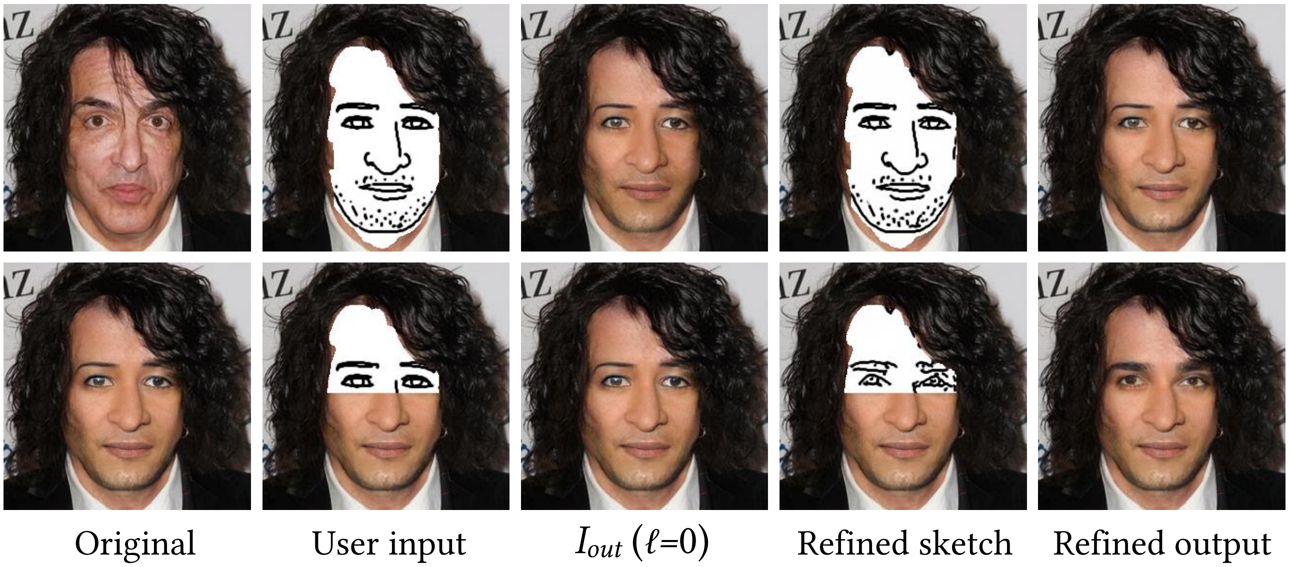


Figure 20. Image editing results with spatially non-uniform refinement level control.

References

[1] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 2, 3

[2] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proc. Int'l Conf. Computer Vision*, 2019. 4, 5, 9

[3] Justin Johnson, Alexandre Alahi, and Fei Fei Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conf. Computer Vision*, pages 694–711, 2016. 2, 10

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *Proc. Int'l Conf. Learning Representations*, 2018. 3, 10

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 3

[6] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual gan. In *Proc. European Conf. Computer Vision*, pages 205–220, 2018. 3, 8, 9

[7] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *Proc. Int'l Conf. Learning Representations*, 2018. 2

[8] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics*, 37(4):99, 2018. 3

[9] A. Recasens, P. Kellnhofer, S. Stent, W. Matusik, and A. Torralba. Learning to zoom: A saliency-based sampling layer for neural networks. In *ECCV*, pages 51–66, 2018. 3

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int'l Journal of Computer Vision*, 115(3):211–252, 2015. 3, 10

[11] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 35(4):119, 2016. 3, 15, 16

[12] Ting Chun Wang, Ming Yu Liu, Jun Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018. 3, 6, 7, 9, 10

[13] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 1395–1403, 2015. 3

[14] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Proc. Int'l Conf. Computer Vision*, 2019. 12

[15] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proc. Int'l Conf. Computer Vision*, 2019. 2, 4, 5, 9

[16] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017. 6, 7, 9, 10