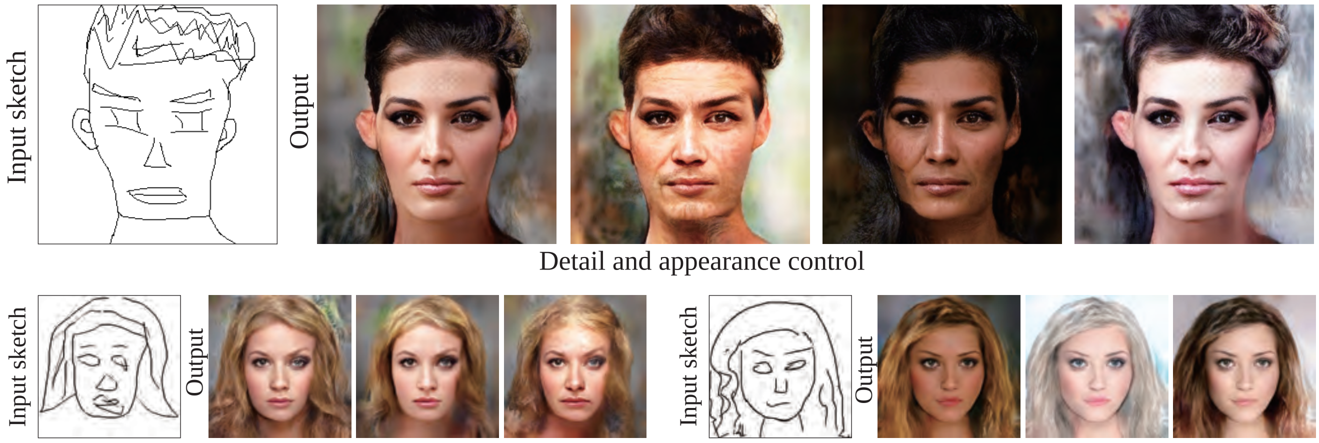
CGI2022 manuscript No.
(will be inserted by the editor)

Diversifying Detail and Appearance in Sketch-Based Face Image Synthesis

Takato Yoshikawa · Yuki Endo · Yoshihiro Kanamori



Detail and appearance control

Detail-only control

Appearance-only control

Fig. 1: Our method can generate diverse, photorealistic images from a single portrait sketch (top row). We can also controll the *detail* and *appearance* separately (bottom row).

Abstract Sketch-based face image synthesis has gained greater attention with the increasing realism of its output images. However, existing studies have overlooked the significance of output *diversity*: because sketches are inherently ambiguous, it would be desirable to have various output candidates for a single input sketch. In this paper, we explore synthesis of diverse face images from a single sketch by using a three-stage framework consisting of sketch refinement, detail enhancement, and appearance synthesis. Each stage uses supervised learning with neural networks. With this three-stage framework, we can separately control the *detail* (e.g., wrinkles and hair structures) and *appearance* (e.g., skin and hair colors) of output face images separately by using multiple latent codes. Quantitative and quantitative evaluations demonstrate that our method offers greater diversity in its output images than the state-of-the-art methods, while retaining the output realism.

Keywords Sketch-based image synthesis · Deep learning · GAN · Multimodal

1 Introduction

Sketching of human faces is an intuitive way to depict facial characteristics with rough strokes: it is done not only as a hobby but also in criminal investigation [31], for example. Digital sketching has become common with the availability of off-the-shelf devices

T. Yoshikawa
E-mail: tenten0727@icloud.com
University of Tsukuba

Y. Endo
E-mail: endo@cs.tsukuba.ac.jp
University of Tsukuba

Y. Kanamori
E-mail: kanamori@cs.tsukuba.ac.jp
University of Tsukuba

such as pen tablets and touch screens, which ignited active research on sketch-based image synthesis over the last decade [4]. Human faces have been a main subject of sketch-based image synthesis because of their ubiquity, and the recent deep learning techniques can "magically" transform even poor face sketches into photorealistic face images [34].

However, in the quest for realistic output face images, existing studies have overlooked an essential aspect of sketch inputs: because sketches are inherently ambiguous, there are many possibilities for the appearance of photorealistic outputs. For example, from the portrait sketches shown in Fig. 1, it is difficult to recognize what the skin and hair colors are and how many wrinkles are on the faces. Such output *diversity* is essential for artists to foster their creativity in sketch-based image synthesis. Nevertheless, most existing methods can obtain only a unimodal output from a single sketch.

In this paper, we propose a method for generating face images from hand-drawn portrait sketches while taking both realism and diversity into account. We achieve multimodal sketch-to-face translation by using a three-stage framework consisting of sketch refinement, detail enhancement, and appearance synthesis. Each stage uses supervised learning with neural networks. For the first stage, we adopt the sketch refinement network used in *Deep Plastic Surgery* (DPS) [34], which translates an input hand-drawn sketch into a sparse edge map that forms a facial outline. The second stage translates the sparse edge map into a dense edge map, and the third stage synthesizes a colorized image as the final output from the dense edge map.

Our key idea is to integrate controllability of *detail* (e.g., wrinkles and hair structures) and *appearance* (e.g., skin and hair colors) to diversify output face images separately in terms of each aspect. The controllability of detail and appearance is achieved in the second and third stages, respectively, by injecting latent codes sampled from a prior into multi-scale *adaptive instance normalization* (AdaIN) [9] layers. During training, the second and third stages learn the respective latent spaces for detail and appearance via a *Wasserstein auto-encoder* (WAE) [27], which achieves higher-quality multimodal outputs than common approaches with a *variational auto-encoder* (VAE) [13]. Note that the ground-truth (GT) dense maps are crucial for the supervised learning in the second stage; accordingly, we selected an edge detector [3] for photographs through a comparative experiment.

As demonstrated in Fig. 1, our method can generate a variety of face images from a single sketch while controlling the detail and appearance separately. Both qualitative and quantitative evaluations demonstrate that our method can generate diverse images while maintaining realism comparable to that of state-of-the-art methods for sketch-based image synthesis.

2 Related Work

2.1 Image-to-image (I2I) translation

Various I2I translation methods using deep learning have been proposed to translate images from a source domain to a target domain. The seminal work is *pix2pix* [10], a supervised framework that can translate various types of images (e.g., semantic masks and edge maps) into photographs and vice versa. That method achieved photorealistic image synthesis by using a conditional *generative adversarial network* (GAN) [22]. pix2pix inspired various follow-up studies on applications such as high-resolution images [30,12] and unsupervised learning [38,18,19,6].

General I2I translation frameworks [10,30,12,23,24] have been applied to sketch-based image synthesis via training with edge maps because they can easily be generated from photographs. However, edge-map-based approaches yield poor generalizability with respect to hand-drawn sketches because of the large domain gap.

2.2 Sketch-based image synthesis

Training data can be a bottleneck in sketch-based image synthesis. Several sketch datasets are publicly available [25,36], but the quality and amount of data are not sufficient to obtain good generalizability. To compensate for the lack of data, Chen et al. used synthetic sketches augmented by edge maps to train a GAN that could handle various image classes [5]; however, that approach had insufficient generalizability and limited output realism. The output quality would be improved with more sketch data for training, but collecting such data is quite costly.

Instead of collecting more sketch data, several approaches have attempted to close the domain gap between edge maps and hand-drawn sketches. The contextual GAN [21] learns a joint distribution of edge maps and photos from jointly-paired sketch-photo images and retrieves the paired image whose sketch portion is the closest to the input sketch in the latent space. Later, that approach was specialized for face images [3]. Yang et al. proposed DPS [34], a sketch-based framework for face image editing with controllable sketch fidelity. During training, DPS first generates pseudo-sketches by deforming and dilating edge maps and then trains a refinement network that converts the pseudo-sketches

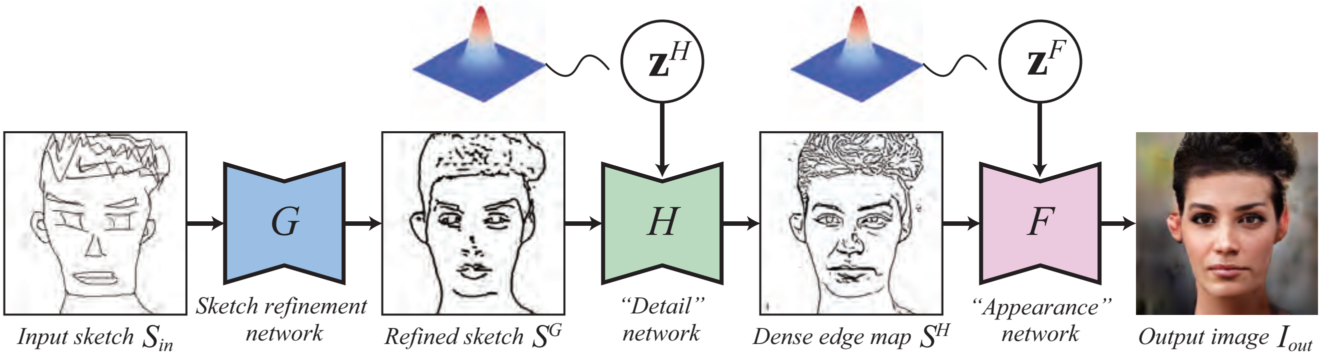


Fig. 2: Overview of our method's inference phase. G is the network to refine a rough input sketch [34]. H and F are networks to diversify the sketch's *detail* and *appearance*, respectively, with separate controllability via latent codes $\mathbf{z}^H$ and $\mathbf{z}^F$.

Table 1: Summary of the notations. Some notations might accompany subscripts "*gt*" and "*real*," which refer to ground-truth images and real samples from a normal distribution, respectively.

Notation	Description
S_{in}	Input sketch
S^G	Refined sketch
S^H	Dense edge map
I_{out}	Output Image
G	Sketch refinement network
H	Detail network
D_H	Detail discriminator
F	Appearance network
E_W	Encoder of WAE
E_D	Discriminator of WAE
$\mathbf{z}^H$	Latent code for detail control
$\mathbf{z}^F$	Latent code for appearance control
$\mathcal{L}_{rec}$	L_1 loss
$\mathcal{L}_{perc}$	Perceptual loss
$\mathcal{L}_G$, $\mathcal{L}_D$	Adversarial losses
$\mathcal{L}_{FM}$	Feature-matching loss
$\mathcal{L}_{GW}$, $\mathcal{L}_{DW}$	Adversarial losses for WAE

to the original edge maps. During inference, the refinement network transforms hand-drawn sketches to facial outlines, and DPS then obtains photorealistic outputs from the refined sketches by using the edge-to-photo translation model. Users can also specify the refinement degree for sketches. Li et al. [17] adopted a similar idea to DPS's pseudo-sketches for face image synthesis from freehand sketches. Their method differs from DPS in that they used an attention mechanism to control the refinement degree according to facial parts. Although those studies improved the realism of synthetic images, they did not consider output diversity sufficiently. In contrast, our method accounts for both realism and diversity in sketch-based image synthesis.

2.3 GAN output diversification

Diversification of images generated by GANs is a challenging problem in general I2I translation. For example, pix2pix [10] often suffers from mode collapse, in which we can obtain only a single plausible result even if we inject an additional noise input into the network. To solve this problem, Ghosh et al. achieved multimodal image synthesis by using multiple generators that have different modes [8]. Whereas that method can generate only a fixed number of images from a single input image, a VAE-GAN [14], which uses a variational auto-encoder (VAE) [13] together with a GAN, can generate a varying number of outputs. Moreover, a BicycleGAN [39] is a more sophisticated network that consists of a conditional version of a VAE-GAN and a latent regressor GAN. In addition to those general approaches, multimodal image synthesis has been tackled in specific tasks such as semantic image synthesis [23,16,20,7], in which the inputs are semantic masks.

Multimodal image synthesis has also been used for sketch inputs. The methods by Lee et al. [15] and Chen et al. [2] enable users to specify appearance according to a reference image. Yang et al. proposed a method for controlling output face images by specifying attributes (e.g., a facial expression, beard, etc.) [35]. Those methods can semantically manipulate the output image, but its diversity is limited to the range of given reference images or specific attributes. In contrast, our method allows users to control not only appearance but also detail in the latent space and to sample an unlimited number of candidates from a single sketch.

Tseng et al. [28] proposed a method that can diversify multiple factors similar to appearance and detail by learning a step-by-step generation process, in which it first generates flat color images from sketches and then adds edges to them. However, that method is not designed for hand-drawn sketch inputs, and its generalizability is limited. Even if input rough sketches were refined by a sketch refinement network [34] as in our method, their method would not obtain plausible results (see Section 4.3 for a comparison).

Very recently, Wang et al. [29] proposed a method that fine-tunes a pre-trained generator so that object shapes in the output images match a few sketch exemplars specified by the user. Their method yields photorealistic, diverse images. Given a single sketch input, however, their method is quite slow because it requires tens of thousands of iterations of backpropagation. In contrast, our feed-forward approach offers instant feedback.

3 Method

Our goal is to generate diverse face images from sketches by controlling detail and appearance separately. Fig. 2 shows an overview of our method's inference phase, and Table 1 summarizes the notations used in this paper. First, a sketch refinement network G converts a hand-drawn sketch input S_{in} to a refined sketch S^G. Next, a detail network H converts S^G to a dense edge map S^H. Finally, an appearance network F generates an output face image I_{out} from S^H. We control the output images of networks H and F by using latent codes $\mathbf{z}^H, \mathbf{z}^F \in \mathbb{R}^n$, where n is the number of dimensions. While $\mathbf{z}^H$ and $\mathbf{z}^F$ are randomly sampled from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in the inference phase, they are encoded by a WAE during training. We train each network separately in a supervised manner for stable training. We describe the network architectures in Section 3.1 and the training procedure in Section 3.2.

3.1 Network architectures

In this section, we first describe the overall network architectures and then discuss the design choices for specific operations used in the networks. The sketch refinement network G, we adopt the encoder-decoder architecture used in DPS [34]. The networks H and F use the same encoder-decoder architecture; here we elaborate the case of the detail network H, as shown in Fig. 3. The encoder contains reflection padding and 3×3 convolution in the first two layers. These are followed by six *Encoder's Modulation Blocks.* As shown in the upper-right blue box in Fig. 3, each of these blocks consists of leaky ReLU activation, reflection padding, 3×3 convolution, and AdaIN [9]. Meanwhile, the decoder consists of six *Decoder's Modulation Blocks* (upper-right pink box in Fig. 3), each of which consists of ReLU activation, upsampling, reflection padding, 3×3 convolution, and AdaIN. In the last three layers, it also contains reflection padding, 3×3 convolution, and hyperbolic tangent. The network D_H is a discriminator that has the same architecture as DPS and discriminates between the output dense edge map S^H and GT S^H_{gt}. Finaly, the networks E_W and D_W are the respective encoder and discriminator of WAE [27]. We use the `resnet_128` architecture [39] for E_W and an architecture consisting of two linear layers for D_W.

Learning of detail and appearance latent spaces. As explained in Section 2.3, many existing techniques for multimodal image synthesis [14,23,7] have adopted a VAE [13] to diversify their output images. However, a VAE tends to generate blurry images, because its training enforces reconstraction of the same image from different latent codes that are sampled via the reparameterization trick. To solve this problem, we adopt a WAE [27], which makes the latent space's distribution match the prior via GAN-based learning. As shown in the upper left of Fig. 3, we encode the GT dense edge map S^H_{gt} into the latent code $\mathbf{z}^H$ via the encoder E_W. Then, by using the discriminator D_W, we compute the loss between $\mathbf{z}^H$ and $\mathbf{z}^H_{real}$, which is sampled from the prior distribution. Lastly, we inject the encoded latent codes $\mathbf{z}^H$ into network H and reconstruct S^H_{gt} as S^H. In our framework, the WAE learns latent spaces for details and appearance that match prior distributions and yield diverse outputs, with separate controllability of each factor.

Injection of detail and appearance information. To inject "style" information into image generation networks, AdaIN [9] is a popular choice in various techniques. AdaIN applies scaling and shifting to input feature maps after instance normalization:

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y), \tag{1}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote functions that compute the mean and standard deviation across the spatial locations of the feature maps x and y.

We use AdaIN to inject detail information into the detail network H by using the latent code $\mathbf{z}^H$ as input (and similarly for the appearance network F with $\mathbf{z}^F$). As shown in the upper-right boxes in Fig. 3, the two linear layers project $\mathbf{z}^H$ to scaling and shifting values that correspond to $\sigma(y)$ and $\mu(y)$ in Equation (1). To reflect detail information over multiple scales, we apply AdaIN repeatedly via the six modulation blocks in the encoder and the decoder.

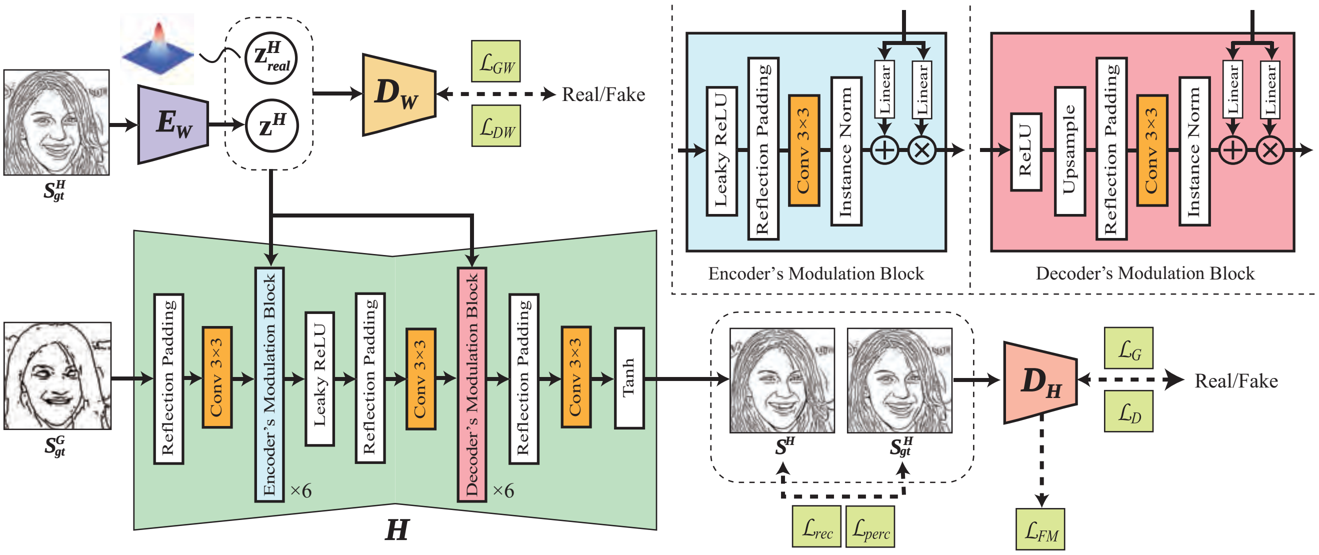


Fig. 3: Our network architecture and a training overview for network H (the same as network F except for $\mathcal{L}_{FM}$). The details of the *Encoder's/Decoder's Modulation Blocks* are illustrated in the upper-right boxes.

3.2 Training

We use a pre-trained network for G and separately train the networks H and F in a supervised manner. The sketch refinement network G is trained to translate synthetic sketches (i.e., deformed, discarded, and dilated edge maps) into original edge maps [34]. The detail network H learns translation from a sparse edge map S_{in} to a dense edge map S^G_{gt}. Meanwhile, the appearance network F learns translation from a dense edge map S^H_{gt} to a face photograph I_{gt}.

To prepare the GT sparse edge maps, we use holistically nested edge detection (HED) [33], because network G is also trained using HED-based edge maps. To prepare the GT dense edge maps, we adopt the approach in *DeepFaceDrawing* (DFD) [2,3], among several candidates, because it can extract visually plausible detail edges like wrinkles and hair structures, as demonstrated in our experiments (Section 4.4).

Loss function. To train the detail network H, we use the following loss function:

$$\mathcal{L} = \mathcal{L}_{DPS} + \lambda_{FM}\mathcal{L}_{FM} + \lambda_{GW}\mathcal{L}_{GW} + \lambda_{DW}\mathcal{L}_{DW}, \quad (2)$$

where λ_{FM}, λ_{GW}, and λ_{DW} are weight coefficients. We also use $\mathcal{L}$ to train the appearance network F, but without $\mathcal{L}_{FM}$, because it does not change the results. We explain each term as follows. First, $\mathcal{L}_{DPS}$ captures the loss for the edge-to-photo network in DPS [34]:

$$\mathcal{L}_{DPS} = \lambda_{rec}\,\mathcal{L}_{rec} + \lambda_{perc}\,\mathcal{L}_{perc} + \lambda_G\,\mathcal{L}_G + \lambda_D\,\mathcal{L}_D, \quad (3)$$

where λ_{rec}, λ_{perc}, λ_G, and λ_D are weight coefficients. Here, $\mathcal{L}_{rec}$ is the L_1 loss between the GT images and the network outputs:

$$\mathcal{L}_{rec} = \mathbb{E}\left[\left\|S^H - S^H_{gt}\right\|_1\right]. \quad (4)$$

$\mathcal{L}_{perc}$ is a perceptual loss [11] that evaluates the semantic similarity between images:

$$\mathcal{L}_{perc} = \mathbb{E}\left[\sum_i \lambda_i \left\|\Phi_i(S^H) - \Phi_i(S^H_{gt})\right\|_2^2\right], \quad (5)$$

where $\Phi_i(x)$ is the feature map of x in the i-th layer of VGG19 [26], and λ_i is the weight at each layer. Lastly, $\mathcal{L}_G$ and $\mathcal{L}_D$ are the adversarial losses with the hinge loss:

$$\mathcal{L}_G = -\mathbb{E}\big[D_H(S^H)\big]\,, \quad (6)$$

$$\mathcal{L}_D = \mathbb{E}\big[\mathrm{ReLU}(\tau + D_H(S^H))\big] + \mathbb{E}\big[\mathrm{ReLU}(\tau - D_H(S^H_{gt}))\big]\,, \quad (7)$$

where τ is a constant. In addition to $\mathcal{L}_{DPS}$, we introduce two more loss functions in our framework. First, we use the feature-matching loss [14] between edge maps:

$$\mathcal{L}_{FM} = \mathbb{E}\left[\sum_j \left\|D_H^{(j)}(S^H) - D_H^{(j)}(S^H_{gt})\right\|_1\right], \quad (8)$$

where $D_H^{(j)}(x)$ is the feature map of x in the j-th layer of the discriminator D_H. This loss function can reduce the occurrence of artifacts in the output of network H. Finally, we introduce the adversarial losses for WAE:

$$\mathcal{L}_{GW} = -\mathbb{E}\big[D_W(E_W(S_{gt}^H))\big]\,, \tag{9}$$

$$\mathcal{L}_{DW} = -(\mathbb{E}\big[D_W(\mathbf{z}_{real}^H)\big] - \mathbb{E}\big[D_W(E_W(S_{gt}^H))\big]), \tag{10}$$

where $\mathbf{z}_{real}^H \in \mathbb{R}^n$ is a vector sampled randomly from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

4 Experiments

4.1 Datasets

To train our framework, we used 28,000 images in the CelebAMask-HQ dataset [12,40]. We extracted sparse and dense edge maps from the face images in the dataset by using the HED [33] and DFD [2,3] approaches. We resized all images to 256 × 256 pixels. The test inputs, for our qualitative and quantitative evaluation were 30 hand-drawn sketches provided as part of DPS [34].

4.2 Implementation details

We used the Python language and the PyTorch library to implement our method. The networks were trained on a PC equipped with an NVIDIA RTX A4000. The training time for both networks H and F was about 13 hours for training of 20 epochs with 28,000 images. The test time was 0.07 seconds to obtain one output image from a single hand-drawn sketch via the trained model.

For optimization, we used Adam with the momentum term $\beta = (0.5, 0.999)$ and set the learning rates to 0.0002. We set the batch size to 4. In all experiments, we set the weights λ_{rec}, λ_{perc}, λ_G, λ_D, λ_{FM}, λ_{GW}, and λ_{DW} to 100, 1, 1, 1, 10, 10000, and 1, respectively. For the perceptual loss $\mathcal{L}_{perc}$, we used the first layer of `conv2` and the first layer of `conv3` of VGG19 [26], which were weighted by 1 and 0.5, respectively. For the hinge loss, constant τ was set to 10. The number of dimensions, n, of the latent codes was set to 8. For network G, we fixed the sketch refinement parameter [34] as $l = 1$ (i.e., full refinement) in our experiments, because the test input sketches were quite coarse and smaller l values yielded poor results.

Please watch the accompanying video for our interactive demo.

4.3 Comparison with existing methods

We compared our method with SPADE [23], ArtEditing [28], and DPS [34]. Because DPS only supports unimodal results, we compared both unimodal and multimodal results. Unimodal results of the multimodal methods (including ours) were obtained by feeding the mean vector of the prior distribution (i.e., a zero vector) as a latent code. The multimodal results were generated from randomly sampled latent codes. Because SPADE and ArtEditing were trained with edge maps and thus could not handle rough hand-drawn sketches, we also applied G to the inputs of those methods: we denote the resulting methods as SPADE+G and ArtEditing+G.

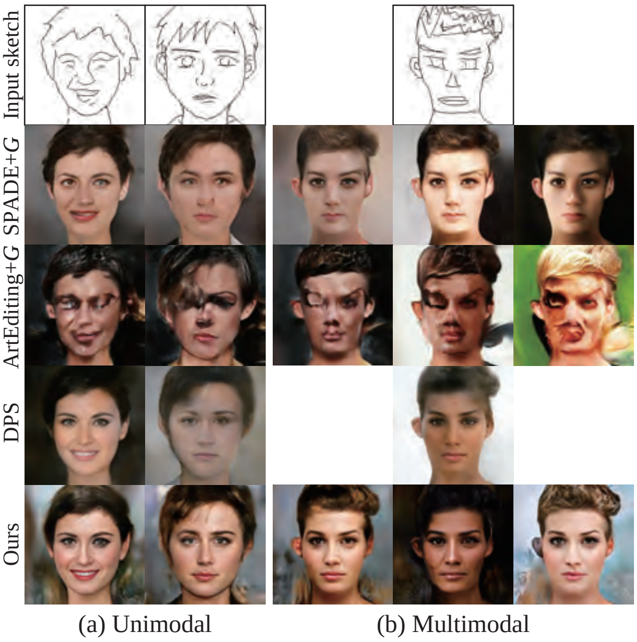
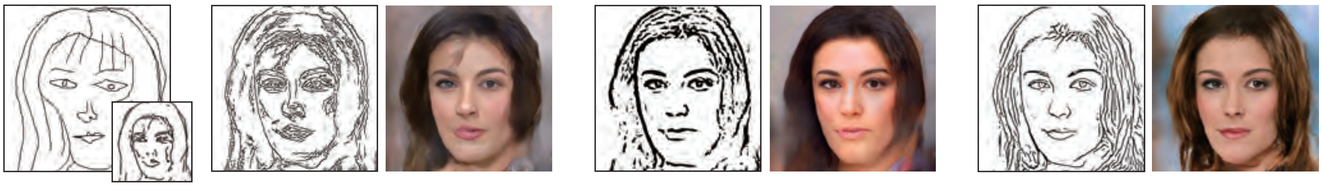


Fig. 4: Qualitative comparison of results from SPADE [23], ArtEditing [28], DPS [34], and our method. The designation "+G" indicates that the sketch inputs of SPADE and ArtEditing were refined by network G [34]. From the (a) unimodal results (generated from zero vectors for the multimodal methods) and (b) multimodal results (generated from random vectors), we can see that our method outperformed the existing methods in terms of diversity while retaining realism comparable to that of the state-of-the-art method.

Fig. 4 shows a qualitative comparison. As can be seen in the results, although SPADE+G could generate relatively plausible images, they looked a little too smooth, and the multimodal results had relatively small variations. Meanwhile, ArtEditing+G could not obtain realistic results because of its limited generalizability to hand-drawn sketches. DPS [34] tended to produce blurry images and was limited to unimodal outputs. In contrast, our method generated clearer images with richer variations (e.g., laugh lines and skin colors) because of the diversified detail and appearance in our framework. Appendix A shows additional qualitative



S_{in},S^G (a) Canny (b) XDoG (c) DFD

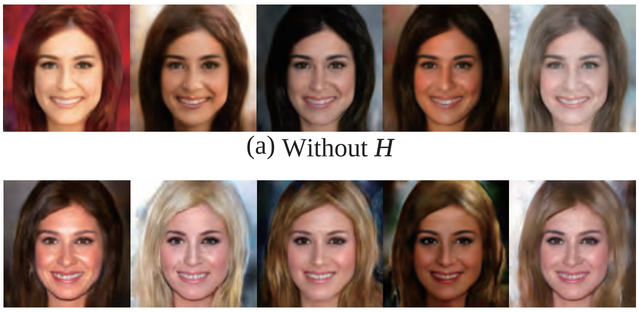
Fig. 5: Comparison of edge detectors for a dense edge map S^H and the corresponding final output I_{out}. The (a) Canny and (b) XDoG [32] edge maps could not represent detail appropriately, and their final outputs were not realistic. Meanwhile, the DFD [2, 3] edge map could depict detail such as hair structures, which is reflected in the final output.

Table 2: Quantitative results for diversity (based on LPIPS) and realism (via a user study). For multimodal outputs, 12 subjects evaluated results generated from zero vectors (scores outside parentheses) and from random vectors (inside parentheses). The caption of Fig. 4 indicates the compared methods.

	Diversity ↑	Realism ↑
SPADE+G	0.265	2.20 (2.29)
ArtEditing+G	0.350	1.13 (1.07)
DPS	n/a	**3.40** (n/a)
Ours	**0.373**	**3.40** (2.74)

comparisons, which also included a BicycleGAN [39] and pix2pixHD [30].

We also conducted experiments for quantitative evaluation of diversity and realism. For diversity, we calculated an evaluation metric by using LPIPS [37] as follows. For each input sketch, we generated 10 multimodal outputs and averaged the LPIPS value for each pair among all their combinations. We applied this process to 30 input sketches and averaged the resultant values to obtain the diversity score, where a higher score indicates greater diversity. For realism, we conducted a user study. We also considered numerical metrics such as FID and SSIM, but we could not use them because GT photographs corresponding to the test sketches were not available to compute those metrics. In the user study, we asked 12 subjects to score the results of each method in a range of 1 to 5 (with 1 and 5 indicating the least and most realistic results, respectively). We averaged the scores given by all the subjects to obtain the realism score. Note that the evaluation scores for the multimodal methods could differ depending on the latent codes sampled from the prior. For fair comparison, we showed the subjects two types of results: one generated with a zero vector, and the other with random latent codes.



(a) Without H

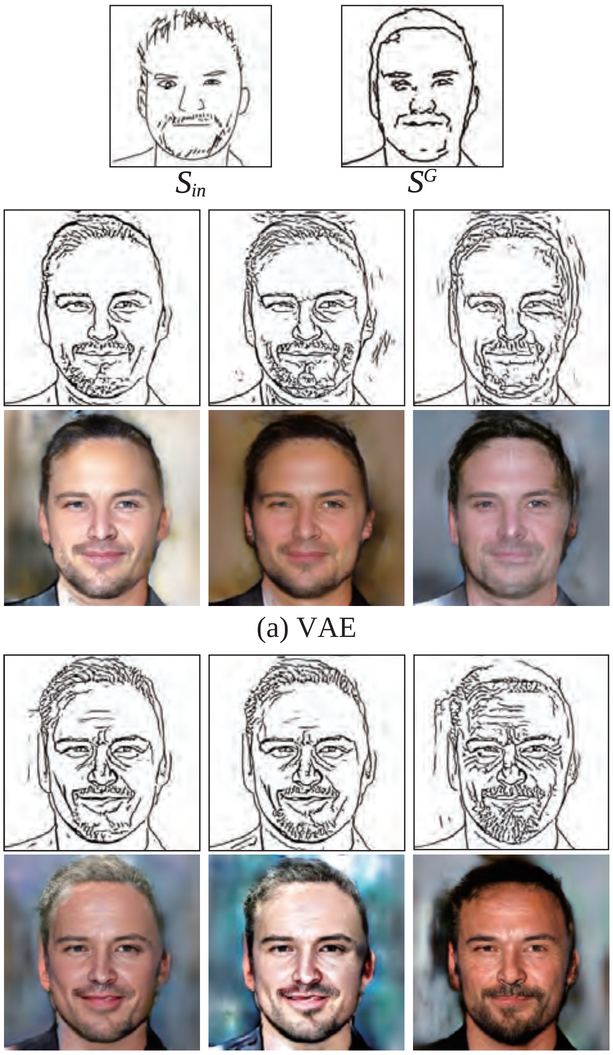
(b) With H

Fig. 6: Comparison of images (a) without and (b) with the detail network H. Whereas the results without H were blurry, those with H were clearer and more diverse.

Table 2 summarizes the quantitative comparison of the diversity and realism scores. The realism scores outside or inside parentheses represent those that were obtained from a zero vector or random latent codes, respectively. First, the diversity score (LPIPS) of our method was the highest among all the methods, which indicates that it yielded the most diverse images from the sketches. The realism scores showed that our method was on par with DPS [34] in the unimodal setting. As for the multimodal setting, our method achieved the highest realism score (i.e., the value in parentheses) among all the multimodal methods. In summary, our method yielded more diversity than the existing multimodal methods while maintaining high realism comparable to that of DPS.

4.4 Ablation study

We also conducted ablation studies to validate our design choices regarding the detail network H, GT dense edge maps, WAE, and feature-matching loss.

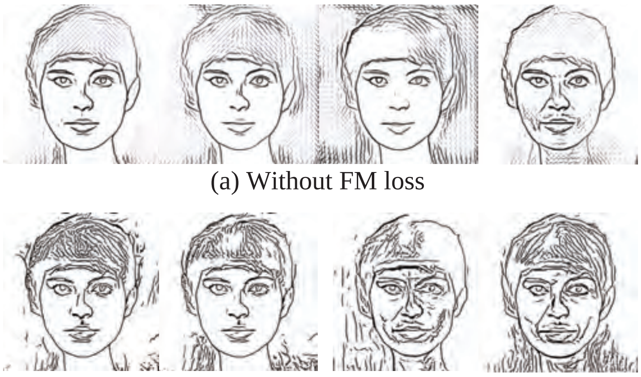


(a) VAE

(b) WAE

Fig. 7: Comparison of images generated using (a) a VAE or (b) a WAE, showing the dense edge map S^H (top rows) and the final output I_{out} (bottom rows). The images generated with the WAE had more diverse detail (e.g., wrinkles) in S^H and appearance (e.g., skin color) in I_{out} than those generated with the VAE.

Validation of detail network H. Fig. 6 shows results generated with and without network H. In the model without H, network F was trained to transform an HED edge map S_{gt}^G to a GT photograph I_{gt} directly, without generating a dense edge map. The comparison revealed that the diversity of the model without H deteriorated in terms of both detail and appearance (particularly in the hair regions), because the diversity of both factors was controlled with only a single latent code. Even worse, the model without H lost the separate controllability of detail and appearance.



(a) Without FM loss

(b) With FM loss

Fig. 8: Comparison of refined dense edge maps obtained (a) without and (b) with the feature-matching (FM) loss. The FM loss was useful to reduce grid-like artifacts like those in (a).

Validation of GT dense edge maps. To generate GT dense edge maps for training the detail network H, we evaluated three edge detectors: the Canny edge detector [1], the XDoG [32] operator, and DFD [2,3]. Fig. 5 shows that the edges detected by the Canny and XDoG techniques were denser than the input sketch, but they did not depicted hair structures (i.e., detail) appropriately. Consequently, the corresponding face outputs lost realism. In contrast, we can see that the DFD edge maps depict the detail well, and the final output was visually plausible while capturing hair structures, for example.

Validation of WAE. We compared two network variants with a WAE [27] and a VAE [13]. Fig. 7 shows that the WAE obtained more diverse and clearer results in terms of detail and appearance than the VAE. For example, the WAE could capture various skin and hair colors and high-frequency components such as beards and wrinkles, whereas the VAE could not handle them appropriately because of its drawback described in Section 3.1. Accordingly, the WAE is more suitable than the VAE for our framework.

Validation of feature matching loss. Fig. 8 shows results obtained with and without using the feature-matching (FM) loss in training network H. Without the FM loss, the network could not learn edge features well, which caused grid-like artifacts in the refined dense edge maps. In contrast, by using the FM loss, we could reproduce plausible hair structures and wrinkles without noticeable artifacts by having the network learn to match the discriminator features of the generated dense edge maps and GT dense edge maps.

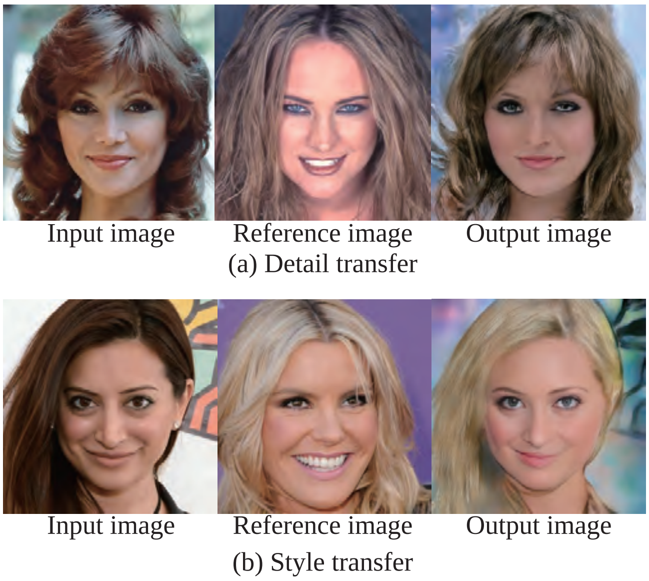


Fig. 9: Application of our method to transfer (a) detail and (b) appearance from a reference image.

4.5 Application

As shown in Fig. 9, our method can also transfer detail or appearance from a reference image to a target image, as follows. First, we extract a sparse edge map from an input image by using the HED algorithm. Next, we feed the extracted edge map to our model without network G. Instead of randomly sampling a latent code from the prior, we extract a latent code from the reference image by using the encoder E_W. By injecting the encoded latent code into networks H and F, we can transfer the hair structures and skin and hair colors in the reference image to the target image.

5 Conclusions

In this paper, we have proposed a three-stage framework that can generate diverse face images from a single hand-drawn sketch. After refining a rough sketch input by using a sketch refinement network [34], our framework diversifies the *detail* (e.g., wrinkles and hair structures) and *appearance* (e.g., skin and hair colors) of the output face images with separate controllability of each factor. The controllability stems from injection of latent codes into encoder-decoder networks via our AdaIN modulation blocks. The latent space for each factor is learned via a WAE instead of a VAE, unlike common techniques for multimodal image synthesis. We validated our method through qualitative and quantitative comparisons with state-of-the-art methods. Specifically, our method generated more diverse and photorealistic images, as it achieved the best diversity and realism scores based on LPIPS and a user study. We also conducted ablation studies and validated the effectiveness of the detail network H, GT dense edge maps, WAE, and FM loss used in our framework.

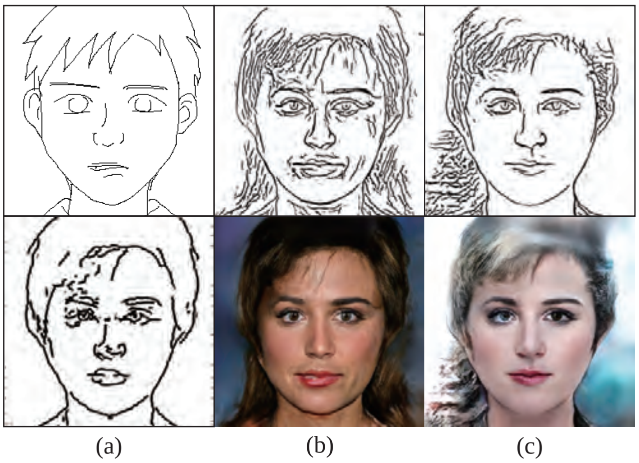


Fig. 10: Failure cases in the proposed framework, showing (a) the input sketch at the top and the refined sparse edge map at the bottom, and (b, c) the corresponding refined dense edge maps (top) and final outputs (bottom). With inappropriate combinations of random latent codes, our method sometimes (b) adds unnecessary edges in the background or (c) colorizes hair regions unnaturally.

Limitations and future work. Our method has the following limitations, which we plan to address in our future work. First, as shown in Fig. 10(b), the detail network H sometimes adds unnecessary edges in the background. We may be able to solve this problem by feeding a background mask into the network to restrict the area for refinement. Second, as shown in Fig. 10(c), certain inappropriate combinations of the latent codes for detail and appearance caused unnatural colors and degraded the output realism. We attribute this defect to the separate training procedures for our networks, which cause error accumulation in step-by-step inference. While we adopted this approach to stabilize the training, we plan to develop a framework for stable end-to-end training of the whole network.

References

1. Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)

2. Chen, S., Liu, F., Lai, Y., Rosin, P.L., Li, C., Fu, H., Gao, L.: DeepFaceEditing: deep face generation and editing with disentangled geometry and appearance control. ACM Trans. Graph. **40**(4), 90:1–90:15 (2021)
3. Chen, S., Su, W., Gao, L., Xia, S., Fu, H.: DeepFaceDrawing: deep generation of face images from sketches. ACM Trans. Graph. **39**(4), 72 (2020)
4. Chen, T., Cheng, M., Tan, P., Shamir, A., Hu, S.: Sketch2photo: internet image montage. ACM Trans. Graph. **28**(5), 124 (2009)
5. Chen, W., Hays, J.: SketchyGAN: Towards diverse and realistic sketch to image synthesis. In: CVPR 2018, pp. 9416–9425 (2018)
6. Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR 2018, pp. 8789–8797 (2018)
7. Endo, Y., Kanamori, Y.: Diversifying semantic image synthesis and editing via class- and layer-wise VAEs. Comput. Graph. Forum **39**(7), 519–530 (2020)
8. Ghosh, A., Kulharia, V., Namboodiri, V.P., Torr, P.H.S., Dokania, P.K.: Multi-agent diverse generative adversarial networks. In: CVPR 2018, pp. 8513–8521 (2018)
9. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV 2017, pp. 1510–1519 (2017)
10. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR 2017, pp. 5967–5976 (2017)
11. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV 2016, vol. 9906, pp. 694–711 (2016)
12. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR 2018 (2018)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR 2014 (2014)
14. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: ICML 2016, vol. 48, pp. 1558–1566 (2016)
15. Lee, J., Kim, E., Lee, Y., Kim, D., Chang, J., Choo, J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In: CVPR 2020, pp. 5800–5809 (2020)
16. Li, K., Zhang, T., Malik, J.: Diverse image synthesis from semantic layouts via conditional IMLE. In: ICCV 2019, pp. 4219–4228 (2019)
17. Li, Y., Chen, X., Yang, B., Chen, Z., Cheng, Z., Zha, Z.: DeepFacePencil: Creating face images from freehand sketches. In: ACMMM 2020, pp. 991–999 (2020)
18. Liu, M., Breuel, T.M., Kautz, J.: Unsupervised image-to-image translation networks. In: NeurIPS 2017, pp. 700–708 (2017)
19. Liu, M., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: ICCV 2019, pp. 10,550–10,559 (2019)
20. Liu, X., Yin, G., Shao, J., Wang, X., Li, H.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In: NeurIPS 2019, pp. 568–578 (2019)
21. Lu, Y., Wu, S., Tai, Y., Tang, C.: Image generation from sketch constraint using contextual GAN. In: ECCV 2018, vol. 11220, pp. 213–228 (2018)
22. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR **abs/1411.1784** (2014)
23. Park, T., Liu, M., Wang, T., Zhu, J.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR 2019, pp. 2337–2346 (2019)
24. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: A stylegan encoder for image-to-image translation. In: CVPR 2021, pp. 2287–2296 (2021)
25. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. Graph. **35**(4), 119:1–119:12 (2016)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR 2015 (2015)
27. Tolstikhin, I.O., Bousquet, O., Gelly, S., Schölkopf, B.: Wasserstein auto-encoders. In: ICLR 2018 (2018)
28. Tseng, H., Fisher, M., Lu, J., Li, Y., Kim, V.G., Yang, M.: Modeling artistic workflows for image generation and editing. In: ECCV 2020, vol. 12363, pp. 158–174 (2020)
29. Wang, S.Y., Bau, D., Zhu, J.Y.: Sketch your own GAN. In: ICCV 2021 (2021)
30. Wang, T., Liu, M., Zhu, J., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: CVPR 2018, pp. 8798–8807 (2018)
31. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 1955–1967 (2009)
32. Winnemöller, H., Kyprianidis, J.E., Olsen, S.C.: Xdog: An extended difference-of-gaussians compendium including advanced image stylization. Comput. Graph. **36**(6), 740–753 (2012)
33. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV 2015, pp. 1395–1403 (2015)
34. Yang, S., Wang, Z., Liu, J., Guo, Z.: Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In: ECCV 2020, vol. 12360, pp. 601–617 (2020)
35. Yang, Y., Hossain, M.Z., Gedeon, T., Rahman, S.: S2FGAN: Semantically Aware Interactive Sketch-to-Face Translation. CoRR **abs/2011.14785** (2020)
36. Yu, Q., Liu, F., Song, Y., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: CVPR 2016, pp. 799–807 (2016)
37. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR 2018, pp. 586–595 (2018)
38. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV 2017, pp. 2242–2251 (2017)
39. Zhu, J., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: NeurIPS 2017, pp. 465–476 (2017)
40. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: SEAN: image synthesis with semantic region-adaptive normalization. In: CVPR 2020, pp. 5103–5112 (2020)

A Additional Qualitative Comparisons

Figs. 11 and 12 show additional qualitative comparisons for 30 test input sketches that were also used in our user study. The compared methods were a BicycleGAN [39], pix2pixHD [30], SPADE [23] with the sketch refinement network G [34], ArtEditing [28] with G, and DPS [34]. The results obtained by the compared methods often had blurry, unnatural faces, whereas our results were plausible and diverse.

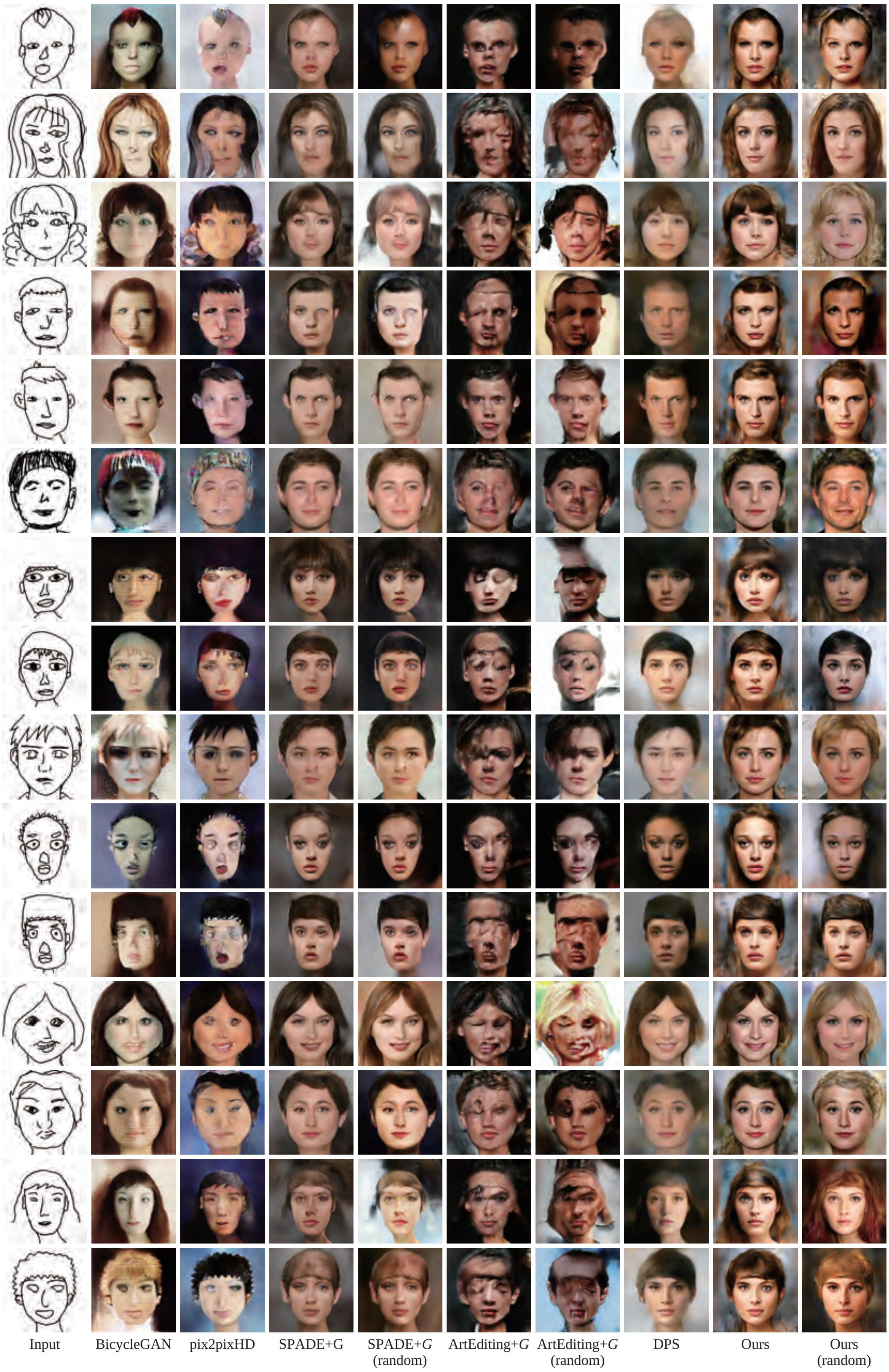


Fig. 11: Additional qualitative comparison. From left to right: the input sketches, and the results obtained by BicycleGAN [39], pix2pixHD [30], SPADE [23] with the sketch refinement network G [34], ArtEditing [28] with G, DPS [34], and our method.

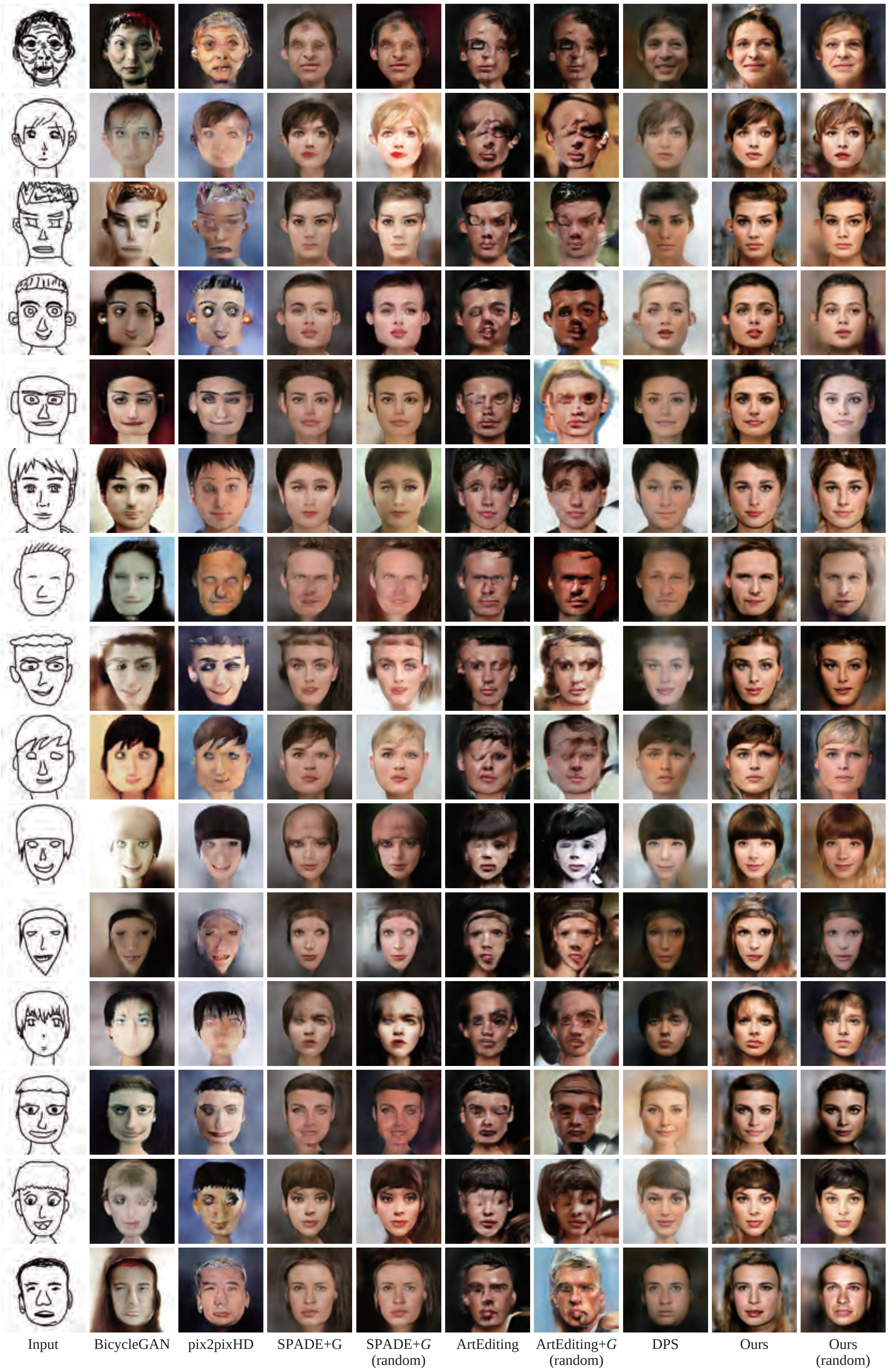


Fig. 12: Additional qualitative comparison obtained with the same settings as in Fig. 11 but on 15 different sketch inputs.