Quality Metric Guided Portrait Line Drawing Generation From Unpaired Training Data

Ran Yi ⬤, Yong-Jin Liu ⬤, *Senior Member, IEEE*, Yu-Kun Lai ⬤, *Member, IEEE*, and Paul L. Rosin ⬤

Abstract—Face portrait line drawing is a unique style of art which is highly abstract and expressive. However, due to its high semantic constraints, many existing methods learn to generate portrait drawings using paired training data, which is costly and time-consuming to obtain. In this paper, we propose a novel method to automatically transform face photos to portrait drawings using unpaired training data with two new features; i.e., our method can (1) learn to generate high quality portrait drawings in multiple styles using a single network and (2) generate portrait drawings in a "new style" unseen in the training data. To achieve these benefits, we (1) propose a novel quality metric for portrait drawings which is learned from human perception, and (2) introduce a quality loss to guide the network toward generating better looking portrait drawings. We observe that existing unpaired translation methods such as CycleGAN tend to embed invisible reconstruction information indiscriminately in the whole drawings due to significant information imbalance between the photo and portrait drawing domains, which leads to important facial features missing. To address this problem, we propose a novel asymmetric cycle mapping that enforces the reconstruction information to be visible and only embedded in the selected facial regions. Along with localized discriminators for important facial regions, our method well preserves all important facial features in the generated drawings. Generator dissection further explains that our model learns to incorporate face semantic information during drawing generation. Extensive experiments including a user study show that our model outperforms state-of-the-art methods.

Index Terms—Face portrait, drawing, style transfer, unpaired image translation, generative adversarial network, quality metric

✦

1 INTRODUCTION

FACE portrait line drawing is a highly abstract and expressive art form, which compresses the rich information in human portraits into a sparse set of graphical elements (e.g., lines) and has high semantic constraints. Usually only skilled artists can generate delicate portrait line drawings and different artists have diverse styles. However, the hand-made drawing process is time consuming and challenging. Recently, a few state-of-the-art works develop elegant algorithms to automatically generate face portrait line drawings [1], [2], [3], which show some interesting progress on the aspect that artificial intelligence can learn to create human art. In this paper, we take a step forward by addressing the following problem: *Can artificial intelligence learn the artistic style space of face portrait line drawings and generate portrait drawings of "new styles" unseen in the training data?*

This challenging problem has not been studied in previous research, possibly due to two outstanding issues. First, artistic portrait line drawings (APDrawings) are quite different from the previously tackled image styles. Image style transfer has been a longstanding topic in computer vision. In recent years, inspired by the effectiveness of deep learning, Gatys *et al.* [4] introduced convolutional neural networks (CNNs) to transfer the style from a style image to a content image, and opened up the field of neural style transfer. Subsequently, generative adversarial networks (GANs) have achieved much success in solving image style transfer problems [5], [6]. However, existing methods are mainly applied to cluttered styles (e.g., oil painting style) where (1) a stylized image is full of fragmented brush strokes and (2) the requirement for the quality of each individual element is low. APDrawings are completely different, and generating them is very challenging because the style is highly abstract: it (1) only contains a sparse set of graphical elements, (2) is line-stroke-based and disables shading, and (3) has high semantic constraints. Therefore, previous texture-based style transfer methods and general image-to-image translation methods fail to generate good APDrawing results (Fig. 1).

The second outstanding issue is to use unpaired training data. The artistic style space of APDrawings contains diverse styles and collecting paired training data for each style is impossible. APDrawingGAN [1] and APDrawingGAN++ [2] are the only methods that explicitly deal with APDrawings by using a hierarchical structure. However, these methods require *paired* training data that is costly to obtain. Compared to paired training data, APDrawing generation

- Ran Yi is with BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China, and also with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: ranyi@sjtu.edu.cn.
- Yong-Jin Liu is with BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China. E-mail: liuyongjin@tsinghua.edu.cn.
- Yu-Kun Lai and Paul L. Rosin are with the School of Computer Science and Informatics, Cardiff University, CF10 3AT Cardiff, U.K. E-mail: {laiy4, rosinpl}@cardiff.ac.uk.
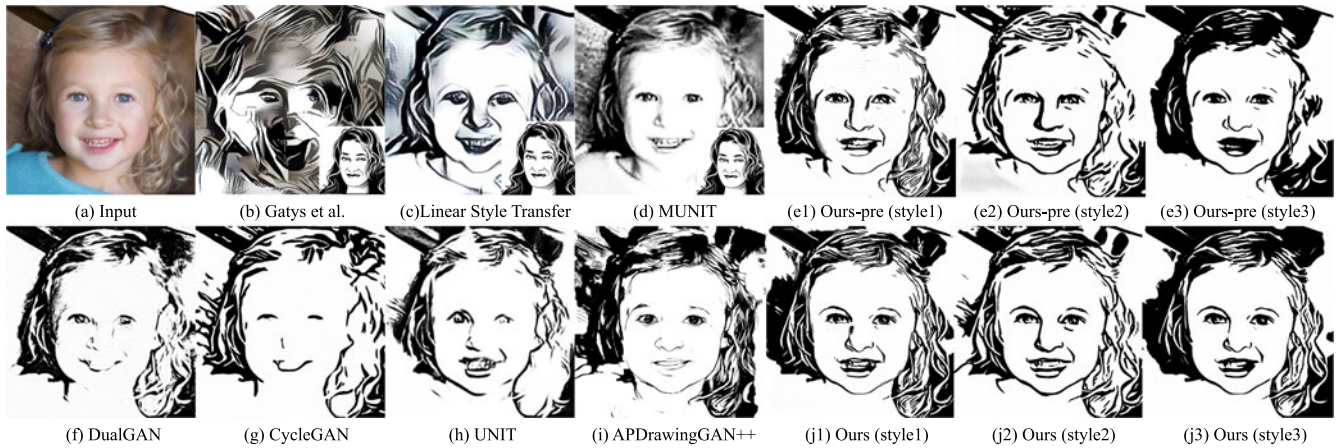
Fig. 1. Comparison with state-of-the-art methods: (a) input face photo; (b)-(c) style transfer methods: Gatys [4] and Linear Style Transfer [7]; (f)-(h) single-modal image-to-image translation methods: DualGAN [8], CycleGAN [6], UNIT [9]; (d) multi-modal image-to-image translation methods MUNIT [10]; (e) our previous conference version (Ours-pre) [3] that outputs three styles; (i) a portrait generation method APDrawingGAN++ [2] using paired training data; (j) our method. Note that our method only uses unpaired training data. Due to this essential difference, we only compare APDrawingGAN++ with our method in Appendix A5.2, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3147570.

learned from *unpaired* data is much more challenging. Previous methods for unpaired image-to-image translation [6], [8] use a cycle structure to regularize training. Although cycle consistency loss enables learning from unpaired data, we observe that when applying them to face photo to APDrawing translation, due to significant imbalance of information richness in these two data types (accurately recovering a photo from the corresponding line drawing is an impossible task), these methods tend to embed invisible reconstruction information indiscriminately in the whole APDrawing, causing a deterioration in the quality of the generated APDrawings, such as important facial features partially missing (Figs. 1f and 1g).

Our previous conference work [3] partially addressed the unpaired training data issue by proposing an asymmetric cycle structure and a truncation loss to prevent the model from embedding invisible features in the generated APDrawings. In this paper, we substantially improve upon [3], and propose a novel quality-metric-guided APDrawing generation model, which can generate (1) "better looking" APDrawings according to human perception, and (2) "new style" APDrawings other than the styles in the training data. Learning from unpaired data makes our model able to utilize diverse APDrawing styles from web data for training the style space. To exploit the natural

diversity of styles from web training images (see Fig. 2 for some examples), our model can (1) learn *multiple styles* (as well as the style space) of APDrawings from web data of mixed styles, (2) generate "new styles" unseen in the training data, and (3) control the output style using a code in the style space. The source code is available.[1]

In particular, we make the following contributions:

- We propose a novel quality metric for APDrawings by learning from human perception. The new quality metric is modeled by a regression network whose input is APDrawing alone and the output is a quality score.
- Based on the quality metric, we propose a quality loss that is consistent with human perception, and use it to guide the network toward generating better looking APDrawings.
- We generate APDrawings of "new styles" unseen in the training data by searching for a corresponding style code in the style space.
- To interpret our model, we dissect the generator by visualizing feature maps and comparing them to face semantics, validating that our generator learns to incorporate semantic face information during APDrawing generation.

2 RELATED WORK

2.1 Neural Style Transfer

Inspired by the successes of CNNs in many visual perception tasks, Gatys *et al.* [4] proposed to use a pretrained CNN to transfer the style in an image to the content of another image in two steps. First, the content features and style features are extracted from images. Second, the content image is optimized by matching the style features from the style image while simultaneously maintaining the content features. In [4], the Gram matrix is used to measure style similarity. This method opens up the field of neural style
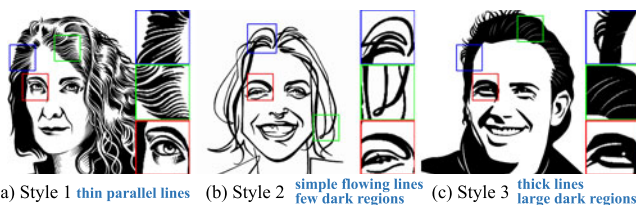

Fig. 2. Representative images of the three styles in our collected web portrait line drawing data. (a) The first style is from Yann Legendre and Charles Burns who use thin parallel lines to draw shadows. (b) The second style is from Kathryn Rathke who draws facial features using simple flowing lines and uses few dark regions. (c) The third style is from vectorportal.com where continuous thick lines and large dark regions are utilized. Close up views are presented alongside for better comparison of the three styles.

1. https://github.com/yiranran/QMUPD

transfer and many follow-up methods are proposed based on this.

Li and Wand [11] proposed to replace the Gram matrix by a Markov Random Field (MRF) regularizer for modeling the style. Stylized images are synthesized by combining MRF with CNN. To speed up the slow optimization process of [4], some methods (e.g., [12], [13]) use a feed-forward neural network to minimize the same objective function. However, these methods still suffer from the problem that each model is restricted to a single style. To simultaneously speed up optimization and maintain style flexibility as [4], Huang and Belongie [14] proposed adaptive instance normalization (AdaIN) to align the mean and variance of content features to those of style features. In these example-guided style transfer methods, the style is extracted from a single image, which is not as convincing as learning from a set of images to synthesize style (refer to GAN-based methods in Section 2.2).

In principle, some neural style transfer methods can generate images with styles unseen in the training data (e.g., [4]). However, these methods model style as texture, and are not suitable for our APDrawing style that has little texture.

2.2 GAN-Based Image-to-Image Translation

GANs [15] have achieved much progress in many vision applications, including image super-resolution [16], text-to-image synthesis [17], [18], and facial attribute manipulation [19], etc. Among these works, two unified GAN frameworks, Pix2Pix [5] and CycleGAN [6], have boosted the field of image-to-image translation.

Pix2Pix [5] was the first general image-to-image translation framework based on conditional GANs, and was later extended to high-resolution image synthesis [20]. Pix2Pix is trained by paired data. Recently, more works focus on learning from unpaired data, due to the difficulty of obtaining paired images in two domains. In this direction, two representative works are CycleGAN [6] and DualGAN [8], which make use of the cycle consistency constraint. This constraint enforces that the two mappings from domains A to B and from B to A — when applied consecutively to an image — revert the image back to itself. Different from enforcing cycle consistency at the image level, UNIT [9] tackles the problem by a shared latent space assumption and enforcing a feature-level cycle consistency. These methods work well for general image-to-image translation tasks. However, in the transformation from face photos to APDrawings, we observe that cycle consistency constraints lead to facial features partially missing in APDrawings, because the information between the source and target domains is imbalanced. In this paper, we relax the cycle consistency in the forward cycle (i.e., photo → drawing → photo) and propose additional constraints to avoid this problem. The NIR-to-RGB method in [21] adopts a very different type of asymmetry — it uses the same loss for the forward and backward cycles, and only changes the network complexity — and targets a different task from ours.

The aforementioned unpaired translation methods are also limited in the diversity of translation outputs. Unpaired data such as crawled web data often naturally contains multi-modal distributions (i.e., inconsistent styles). When knowing the exact number of modes and the mode each sample belongs to, the multi-modal image-to-image translation could be solved by treating each mode as a separate domain and using a multi-domain translation method [19], [22]. However, in many scenarios including our problem setting, this information is not available. MUNIT [10] deals with multi-modal image-to-image translation without knowing the mode each sample belongs to. It encodes an image into a domain-invariant content code and a domain-specific style code, and recombines the content code with the style code sampled from a target domain. Although MUNIT generates multiple outputs with different styles, it cannot generate satisfactory portrait line drawings with clear lines. By inserting style features into the generator and using a soft classification loss to discriminate modes in the training data, our network architecture proposed in this paper can produce multi-style outputs and generate better looking APDrawings than state-of-the-art methods.

3 QUALITY METRIC FOR APDRAWINGS

Most image-to-image translation methods guide generation towards the target domain using a discriminator which decides whether an image is a real target-domain image or not. When the target domain is APDrawings, we found it is not sufficient to decide whether such a drawing is real or fake; i.e., the generator needs further to be told the quality of the synthesized drawing. To the best of our knowledge, there lacks an optimization tool to encourage the network to generate *good looking* portrait line drawings. The perception of good APDrawings — e.g., fluent lines and avoiding messy lines on the face — can be easily concluded by a human, but has not been fully described in an optimization goal. Thus, we introduce a new quality metric for portrait line drawings by *learning from human preference*.

From previous user studies, we found that people can easily decide the quality of a portrait line drawing without knowing the original face photo. So our desired metric can be modeled by a regression network whose input is an APDrawing alone and the output is its quality score.

To obtain such a regression network to predict APDrawing quality, we first generate many APDrawings using existing methods and mix these generated drawings with real artist drawings. Then a user study is conducted to collect human preferences of these APDrawings. After calculating the quality score of each drawing based on human preference, we train a regression network to predict the quality score of an APDrawing.

3.1 Data Preparation

We use existing unpaired image-to-image generation methods including DualGAN [8], CycleGAN [6], UNIT [9], ComboGAN [22], DRIT [23] and our previous conference work [3] to learn the three target styles (as specified in Fig. 2). We then test the trained models on collected web face photos and generate portrait line drawings for the three target styles. The generated drawings are mixed with high-quality APDrawings for subsequent human evaluation. To facilitate the development of a good quality prediction model, we include drawings with diverse quality in the data (as shown in Fig. 3).

Fig. 3. Samples of collected (including generated and artist) portrait line drawings of target style 2 for quality metric model training. The drawings from top to bottom have decreasing quality.
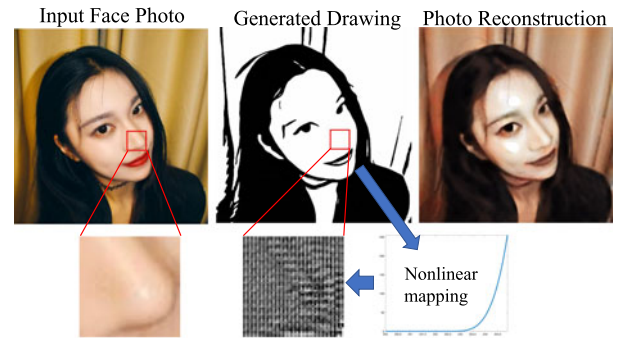

Fig. 4. CycleGAN reconstructs the input photo from generated drawings using a strict cycle-consistency loss, which can potentially embed invisible reconstruction information anywhere in the whole drawings. A nonlinear monotonic mapping of the gray values is applied in a local region around the nose to visualize the embedded reconstruction information.

3.2 Human Preference Collection and Ranking

Considering pairwise comparison is one of the most practical and reliable methods to compare different results, we design a user study based on pairwise comparison. From the comparison results, we compute a ranking. However, given n results, obtaining all $n(n-1)/2$ comparisons from a user study is time consuming and not feasible when n is large. We seek to utilize as few pairwise comparisons as possible to get a global ranking. The efficient ranking method in [24] is suitable in our application. It randomly conducts pairwise comparisons and estimates the score of an object as the relative difference of numbers of preceding items and succeeding items. By using this efficient ranking strategy, an average of $O(n\log n)$ pairwise comparisons are sufficient to recover the true ranking.

User Study Design. Noting that it is difficult to compare the quality of portrait line drawings of different styles due to the style distractions, we therefore only enable pairwise comparison between portrait line drawings of the same style. Then we adopt the above efficient ranking strategy and conduct three user studies based on pairwise comparison for three target styles. To simplify the answering process, each user is shown three portrait line drawings of the same style in a question and asked to rank the three drawings (the answer to each question equals to three pairwise comparisons). To balance the data amount and the effort for human evaluation, we randomly choose 250 drawings for each of the three target styles and collect $2,450\sim3,450$ question responses for each style.

Score and Ranking Calculation. We calculate the scores and global rankings for portrait line drawings of each style separately. To compute the relative difference of numbers of preceding items and succeeding items in [24], for each question answer, denote the ranking as $I_1 \prec I_2 \prec I_3$, then the score of I_1 decreases by 2 (0 preceding and 2 succeeding), the score of I_3 increases by 2 (2 preceding and 0 succeeding) and the score of I_2 stays unchanged. By summarizing all question responses for a style, we calculate the score for each drawing of this style and get a global ranking based on the score. The scores are then normalized to the range $[0.1,1]^2$ so that drawing scores of the three styles have the same range.

2. The lower bound is greater than zero since even the worst examples of training data are better than random.

3.3 Quality Metric Prediction

Given the portrait drawing data and the normalized quality score, we train a regression network to predict APDrawing quality. The regression network is based on the Inception v3 [25] architecture. It takes an APDrawing as input and outputs a quality value. We gather the drawing data of three target styles and train a unified prediction model M.

Since the quality metric model behavior is learned from human evaluation, the predicted score can help guide the drawing generator toward better quality. Furthermore, it can also be used to choose which trained version of the photo-to-APDrawing generator to use, e.g., when multiple versions are trained using different hyper-parameters.

4 NETWORK ARCHITECTURE AND OPTIMIZATION

4.1 Overview

With the aid of trained quality metric model (Section 3), in this section, we propose a GAN model with a novel asymmetric cycle structure, that transforms face photos to high-quality APDrawings, only using unpaired training data. Let $\mathcal{P}$ and $\mathcal{D}$ be the face photo domain and the APDrawing domain, and no pairings need to exist between these two domains. Our model learns a function Φ that maps from $\mathcal{P}$ to $\mathcal{D}$ using training data $S(p) = \{p_i|i=1,2,\ldots,n_p\} \subset \mathcal{P}$ and $S(d) = \{d_j|j=1,2,\ldots,n_d\} \subset \mathcal{D}$. n_p and n_d are the numbers of training photos and APDrawings. Our model consists of (1) two generators, i.e., a generator G transforming face photos to APDrawings, and an inverse generator F transforming APDrawings back to face photos, and (2) two discriminators, i.e., $D_{\mathcal{D}}$ responsible for discriminating generated drawings from real drawings, and $D_{\mathcal{P}}$ responsible for discriminating generated photos from real photos.

The information in the APDrawing domain is much less than in the face photo domain. For example, in the cheek region, there are many color variations in the original photo but the cheek is usually drawn completely white (i.e., no lines are included) in an APDrawing. As illustrated in Fig. 4, enforcing a strict cycle-consistency loss like in Cycle-GAN [6] on the reconstructed and input photos will cause the network to embed reconstruction information in very small variations in the generated APDrawings (i.e., color changes that are invisible to the eye but can make a difference in network calculation); a similar phenomenon was
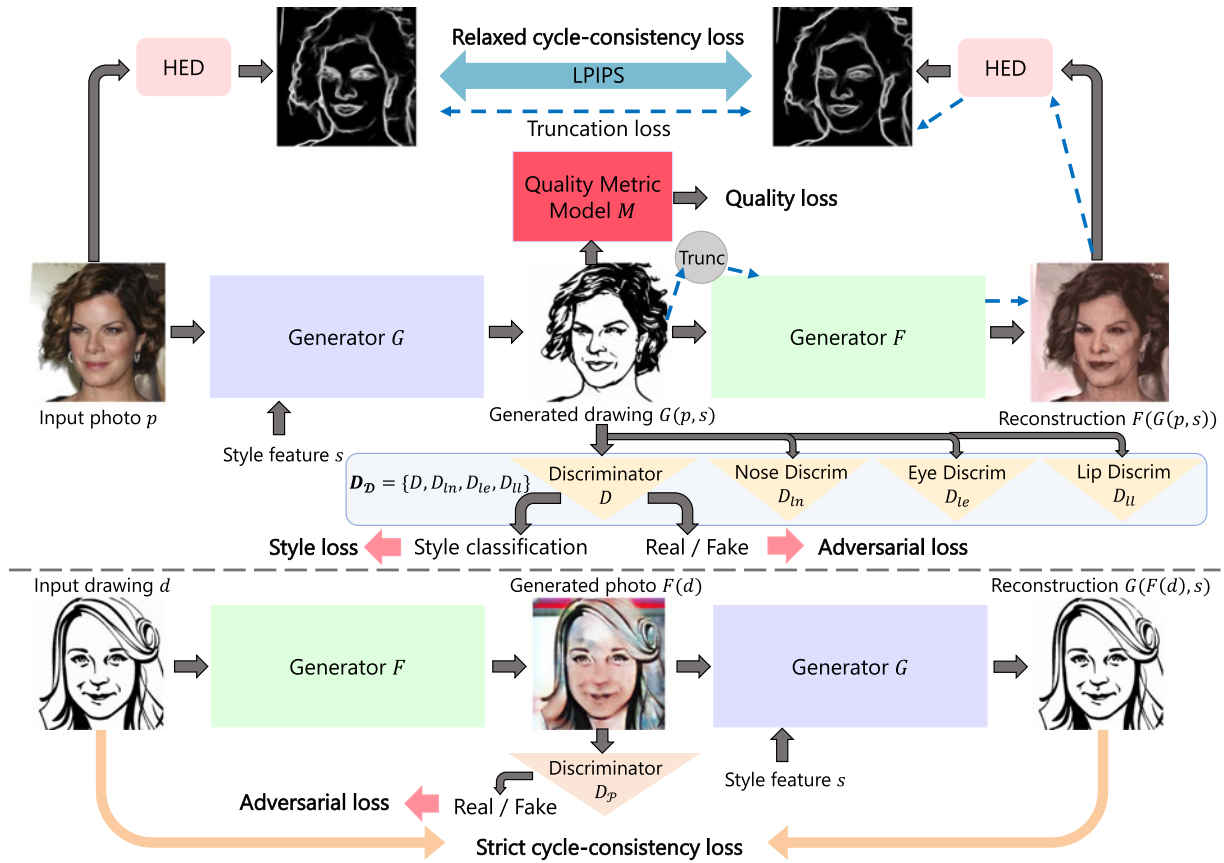
Fig. 5. Our GAN model uses an asymmetric cycle structure, which consists of a photo to drawing generator G, a drawing to photo generator F, a drawing discriminator $D_{\mathcal{D}}$ and a photo discriminator $D_{\mathcal{P}}$. We use a relaxed cycle-consistency loss between reconstructed face photo $F(G(p,s))$ and input photo p, while enforcing a strict cycle-consistency loss between reconstructed drawing $G(F(d))$ and input drawing d. We also introduce local drawing discriminators D_{ln}, D_{le}, D_{ll} for the nose, eyes and lips and a truncation loss. Our model deals with multi-style generation by inserting a style feature vector into the generator and adding a style loss. A quality loss based on the quality metric model (Section 3) further encourages generation of "good looking" APDrawings. The detailed architecture is illustrated in Appendix A2, available in the online supplemental material.

observed in [26]. Embedding reconstruction information in very small variations achieves a balance between cycle-consistency loss and GAN loss in CycleGAN; from the generated drawing $G(p)$, a face photo similar to the input photo p can be successfully reconstructed because of small color changes, while at the same time $G(p)$ tries to be similar to real drawings and be classified as real by the discriminator. However, in APDrawing generation, embedding invisible reconstruction information indiscriminately in the whole drawing will put a strong restriction on the objective function optimization. Moreover, it will allow important facial features to be partially missing in the generated drawings.

To address this problem, our model utilizes a novel idea: although cycle consistency constraints are useful to regularize training, we are only interested in the one way mapping from photos to APDrawings. Therefore, unlike CycleGAN, we do not expect or require the inverse generator F to reconstruct a face photo exactly as the input photo (which is a near impossible task). Instead, our proposed model is *asymmetric* in that we use a relaxed cycle-consistency loss between $F(G(p))$ and p, where only edge information is enforced to be similar, while a strict cycle-consistency loss is enforced on $G(F(d))$ and d. By tolerating the reconstruction information loss between $F(G(p))$ and p, the objective function optimization has enough flexibility to recover all important facial features in APDrawings. A truncation loss

is further proposed to enforce the embedded information to be visible, especially around the local area of the selected edges where relaxed cycle-consistency loss works. Furthermore, local drawing discriminators for the nose, eyes and lips are introduced to enforce the significance of their existence and ensure quality for these regions in the generated drawings. By integrating these techniques, our model can generate high-quality APDrawings with complete facial features.

Another benefit of our model is to generate multi-style APDrawings. The APDrawing data we collected from the Internet contains a variety of styles, of which only some are tagged with author/source information. We select representative styles from the collected data (Fig. 2), and train a classifier for the collected drawings. Then a learned representation is extracted as a style feature vector and inserted into the generator to control the generated drawing style. An additional style loss is introduced to optimize for each style.

Different from our previous conference work [3], we further improve the APDrawing quality and alleviate unwanted artifacts by utilizing the trained quality metric model (Section 3). A human-perception-consistent quality metric loss is proposed to guide the network toward generating good looking APDrawings.

The four networks in our model are trained in an adversarial manner [15]: (1) the two discriminators $D_{\mathcal{D}}$ and $D_{\mathcal{P}}$

are trained to maximize the probability of assigning correct labels to real and synthesized drawings and photos, and (2) meanwhile the two generators G and F are trained to minimize the probability of the discriminators assigning the correct labels. The loss function $L(G, F, D_{\mathcal{D}}, D_{\mathcal{P}})$ contains six types of loss terms: adversarial loss $L_{adv}(G, D_{\mathcal{D}}) + L_{adv}(F, D_{\mathcal{P}})$, relaxed cycle consistency loss $L_{relaxed\sim cyc}(G, F)$, strict cycle consistency loss $L_{strict\sim cyc}(G, F)$, truncation loss $L_{trunc}(G, F)$, style loss $L_{style}(G, D_{\mathcal{D}})$, and quality loss based on the quality metric model $L_{quality}(G)$. Then the function Φ is optimized by solving the minimax problem with the loss function

$$\min_{G,F} \max_{D_{\mathcal{D}}, D_{\mathcal{P}}} L(G, F, D_{\mathcal{D}}, D_{\mathcal{P}})$$
$$= (L_{adv}(G, D_{\mathcal{D}}) + L_{adv}(F, D_{\mathcal{P}}))$$
$$+ \lambda_1 L_{relaxed\sim cyc}(G, F) + \lambda_2 L_{strict\sim cyc}(G, F)$$
$$+ \lambda_3 L_{trunc}(G, F) + \lambda_4 L_{style}(G, D_{\mathcal{D}}) + \lambda_5 L_{quality}(G). \quad (1)$$

The network architectures for G, F, $D_{\mathcal{D}}$ and $D_{\mathcal{P}}$ are introduced in Section 4.2. The detailed design of six loss terms are presented in Section 4.3. An overview of our model is illustrated in Fig. 5.

4.2 Network Architecture

Our GAN model consists of (1) a generator G and a drawing discriminator $D_{\mathcal{D}}$ for face photo to APDrawing translation, and (2) another generator F and a photo discriminator $D_{\mathcal{P}}$ for the inverse APDrawing to photo translation. Considering information imbalance between the face photo in $\mathcal{P}$ and the APDrawing in $\mathcal{D}$, we design different architectures for $(G, D_{\mathcal{D}})$ and $(F, D_{\mathcal{P}})$.

4.2.1 Face Photo to APDrawing Generator G

The generator G takes a face photo p and a style feature s as input, and outputs an APDrawing $G(p, s)$ whose style is specified by s.

Style Feature Vector s. We first train a classifier C (based on VGG-19 [27]) that classifies APDrawings into three styles (Fig. 2), using tagged web drawing data. Then we extract the output of the last fully-connected layer and use a softmax layer to calculate a 3-dimensional vector as the style feature for each drawing (including untagged ones).

Network Structure. G is an encoder-decoder with residual blocks [28] in the middle. It starts with a flat convolution and two down convolution blocks to encode face photos and extract useful features. Then the style feature is mapped to a 3-channel feature map and inserted into the network by concatenating it with the feature map of the second down convolution block. An additional flat convolution is used to merge the style feature map with the extracted feature map. Afterwards, nine residual blocks of the same structure are used to construct the content feature and transfer it to the target domain. Then the output drawing is reconstructed by two up convolution blocks and a final convolution layer.

4.2.2 Drawing Discriminator $D_{\mathcal{D}}$

The drawing discriminator $D_{\mathcal{D}}$ has two tasks: 1) discriminating generated APDrawings from real ones; and 2) classifying an APDrawing into three selected styles, where a real

drawing d is expected to be classified into the correct style label (given by C), and a generated drawing $G(p, s)$ is expected to be classified into the style specified by the 3-dimensional style feature s.

For the first task, to enforce the existence of important facial features in the generated drawing, in addition to a global discriminator D that examines the full drawing, we add three local discriminators D_{ln}, D_{le}, D_{ll} to focus on discriminating local drawings around the nose, eyes and lip respectively. The inputs to these local discriminators are masked drawings, where masks are obtained from a face parsing network [29]. Finally $D_{\mathcal{D}}$ consists of D, D_{ln}, D_{le} and D_{ll}.

Network Structure. The global discriminator D is based on PatchGAN [5] and modified to have two branches. The two branches share three down convolution blocks. Then one branch D_{rf} includes two flat convolution blocks to output a prediction map of real/fake for each patch in the drawing. The other classification branch D_{cls} includes more down convolution blocks and outputs probability values for three style labels. Local discriminators D_{ln}, D_{le}, D_{ll} also adopt the PatchGAN structure.

4.2.3 APDrawing to Face Photo Generator F and Photo Discriminator $D_{\mathcal{P}}$

The generator F in the inverse direction takes an APDrawing d as input and outputs a face photo $F(d)$. It adopts an encoder-decoder architecture with nine residual blocks in the middle. Photo discriminator $D_{\mathcal{P}}$ discriminates generated face photos from real ones, and also adopts the PatchGAN structure.

4.3 Loss Function

There are six types of losses in our loss function (Eq. (1)). We explain them in detail as follows.

Adversarial Loss. The adversarial loss judges discriminator $D_{\mathcal{D}}$'s ability to assign correct labels to real and synthesized drawings. It is formulated as

$$L_{adv}(G, D_{\mathcal{D}}) = \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{d \in S(d)}[\log D(d)]$$
$$+ \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{p \in S(p)}[\log (1 - D(G(p, s)))], \quad (2)$$

where s is randomly selected from the style feature vectors of APDrawings in the training data $S(d)$ for each p. As $D_{\mathcal{D}}$ maximizes this loss and G minimizes it, this loss drives the generated drawings to become closer to real drawings.

We also adopt an adversarial loss for the photo discriminator $D_{\mathcal{P}}$ and the inverse mapping F

$$L_{adv}(F, D_{\mathcal{P}}) = \mathbb{E}_{p \in S(p)}[\log D_{\mathcal{P}}(p)]$$
$$+ \mathbb{E}_{d \in S(d)}[\log (1 - D_{\mathcal{P}}(F(d)))]. \quad (3)$$

Relaxed Forward Cycle-Consistency Loss. As previously mentioned, we observe that there is much less information in domain $\mathcal{D}$ than information in domain $\mathcal{P}$. We do not expect the result from $p \to G(p, s) \to F(G(p, s))$ to be pixel-wise similar to p. Instead, we only expect the edge information in p and $F(G(p, s))$ to be similar. We extract edges from p and $F(G(p, s))$ using HED [30], and evaluate the similarity
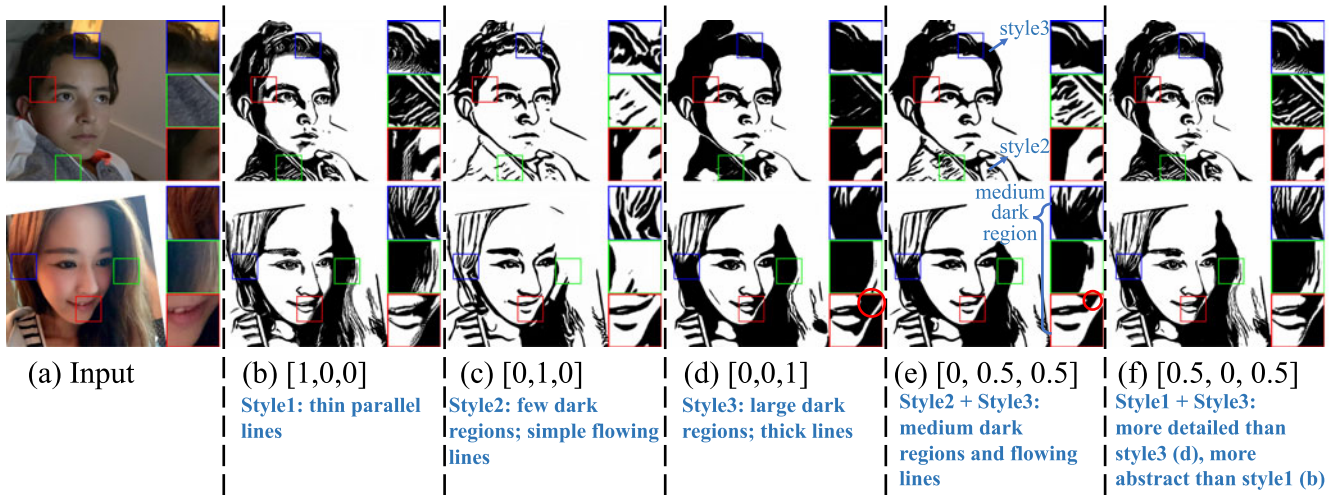
Fig. 6. Results of interpolating between style feature vectors: (a) input photos, (b)-(d) results of three target styles, (e)-(f) results of interpolating target styles. Close-up views are shown by the side.

of edges by the LPIPS perceptual metric proposed in [31]. Denote HED by H and the perceptual metric by L_{lpips}. The relaxed cycle-consistency loss is formulated as

$$L_{relaxed\sim cyc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(G(p, s))))]. \tag{4}$$

Strict Backward Cycle-Consistency Loss. On the other hand, the information in the generated face photo is adequate to reconstruct the APDrawing. Therefore, we expect the result from $d \to F(d) \to G(F(d), s(d))$ to be pixel-wise similar to d, here $s(d)$ is the style feature of d. The strict cycle-consistency loss in the backward cycle is then formulated as

$$L_{strict-cyc}(G, F) = \mathbb{E}_{d \in S(d)}[||d - G(F(d), s(d))||_1]. \tag{5}$$

Truncation Loss. The truncation loss is designed to prevent the generated drawing from hiding information in small values. It is in the same format as the relaxed cycle-consistency loss, except that the generated drawing $G(p, s)$ is first truncated to 6 bits[3] to ensure that encoded information is clearly visible, and then fed into F to reconstruct the photo. More specifically, we first scale the intensities to the range $[0, 64]$, truncate the fractional part, and then scale back. Denote the truncation operation as $T[\cdot]$, the truncation loss is formulated as

$$L_{trunc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(T[G(p, s)])))]. \tag{6}$$

At the beginning of training process, the weight for the truncation loss is kept low, otherwise it would be hard to optimize the model. The weight gradually increases as the training progresses.

Style Loss. It is introduced to help G generate multiple styles with different style features. Denote the classification branch in $D_{\mathcal{D}}$ as D_{cls}. The style loss is formulated as

$$L_{style}(G, D_{\mathcal{D}}) = \mathbb{E}_{d \in S(d)}[-\sum_c p(c)\log D_{cls}(c|d)]$$
$$+ \mathbb{E}_{p \in S(p)}[-\sum_c p'(c)\log D_{cls}(c|G(p, s))]. \tag{7}$$

3. Generally the intensity of a digital image is stored in 8 bits.

For a real drawing d, $p(c)$ is the probability over style label c given by classifier C, $D_{cls}(c|d)$ is the predicted softmax probability by D_{cls} over c. We multiply the term by the probability $p(c)$ in order to take into account those real drawings that may not belong to a single style but lie between two styles, e.g., with softmax probabilities [0.58,0.40,0.02]. For generated drawing $G(p, s)$, $p'(c)$ denotes the probability over style label c and is specified by style feature s, and $D_{cls}(c|G(p, s))$ is the predicted softmax probability over c. This classification loss drives D_{cls} to classify a drawing into the correct style and drives G to generate a drawing close to a given style feature.

Quality Loss Based on the Quality Metric Model. It is designed for generating "good looking" APDrawings. The quality metric model M (described in Section 3) predicts a quality score of an APDrawing by how consistent it is with human perception, where better looking drawings get higher prediction scores (in the range [0.1,1]). We then define the quality loss $L_{quality}$ as

$$L_{quality}(G) = \mathbb{E}_{p \in S(p)}[1 - M(G(p, s))]. \tag{8}$$

5 NEW STYLE GENERATION

In this section, we propose a solution to address the challenging problem of how to generate high-quality APDrawings of "new styles" unseen in the training data. In our multi-style generation setting, different style feature vectors lead to different style outputs. The three target styles correspond to vectors $[1, 0, 0], [0, 1, 0], [0, 0, 1]$, respectively. An interesting question is *what results other style feature vectors would generate and whether some style feature vectors could generate new styles unseen in the training data.*

More specifically, the three target styles we used here are representative styles of portrait line drawings, as shown in Fig. 2. These three styles vary in line thickness, arrangement and dark region ratio, etc. The key features in the three styles are: 1) the drawings of style1 often use thin parallel lines to draw shadows, 2) the drawings of style2 use simple flowing lines and few dark regions, and 3) the drawings of style3 use thick lines and large dark regions.
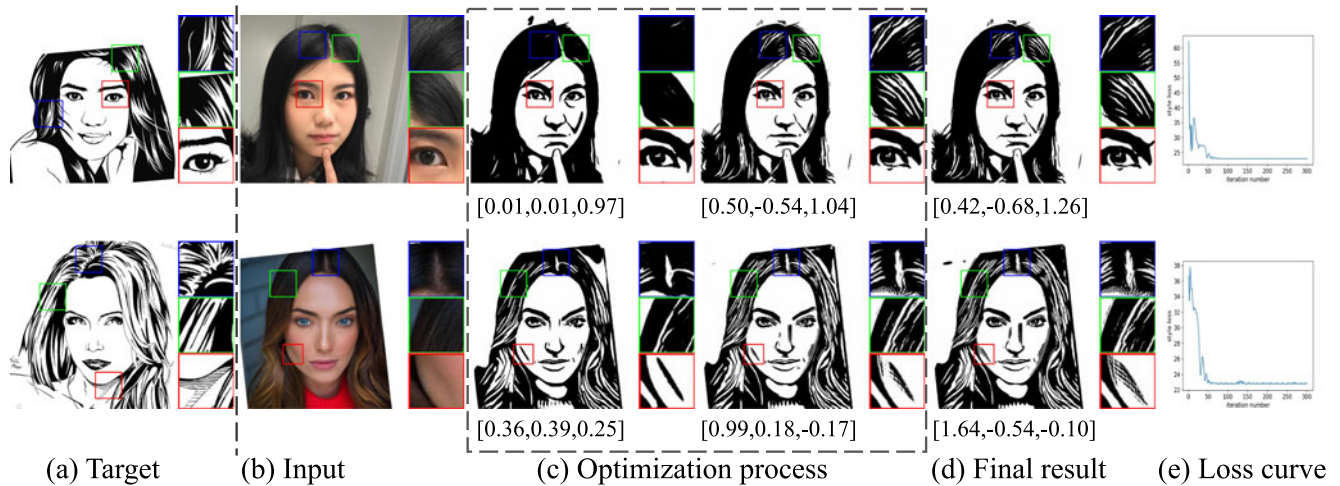
[0.01,0.01,0.97] [0.50,-0.54,1.04] [0.42,-0.68,1.26]

[0.36,0.39,0.25] [0.99,0.18,-0.17] [1.64,-0.54,-0.10]

(a) Target   (b) Input   (c) Optimization process   (d) Final result   (e) Loss curve

Fig. 7. Examples of "new" style generation. Given a target "new" style portrait drawing (i.e., style unseen in training data) in (a), we find a proper style feature vector that generates APDrawings similar to the target, by optimizing a histogram based style loss. The optimization process is shown in (c) and the final generated APDrawing is shown in (d). The style loss change during optimization is shown in (e). Style feature vectors used for generation are shown under each generated APDrawing. Close-up views are shown by the side.

By interpolating between the style feature vectors, we observe that the generated results show a combination of target styles. As shown in Fig. 6e, the results of style feature vector $[0, 0.5, 0.5]$ exhibit a combination of styles 2 and 3, i.e., medium dark regions and flowing lines; in other words, less dark regions compared to (d) and more compared to (c). The result of vector $[0.5, 0, 0.5]$ in Fig. 6f shows a combination of styles 1 and 3, i.e., a combination of thin parallel lines and dark region shadows; in other words, hair regions are more detailed (parallel lines) than (d) and more abstract than (b). Close-up views are also provided for better comparison.

Next we explore whether the network can generate some "new" styles[4] unseen in the training data. Given a "new" style APDrawing d_{target} as a reference, we use the trained APDrawing generator G to look for a style feature vector s in the style space that generates APDrawings most similar in style to the unseen target d_{target}. The best style feature vector s^* is found by optimizing the style distance between the generated APDrawing guided by this vector and the target d_{target}. Denote the loss term to measure style distance as L_{style}. The problem is formulated as

$$s^* = \arg\min_s L_{style}(G(p, s), d_{target}), \qquad (9)$$

where p is a face photo in the training data. Examples of "new" style generation and the corresponding "new" style targets are presented in Figs. 7a, 7b, 7c, and 7d.

To model the style similarity, we explored existing style losses [32] including Gram-matrix-based loss [4] and histogram-based loss [33]. We found histogram loss is better for measuring line drawing style differences. Given the generated APDrawing A and the target style APDrawing B, histogram matching is performed to match feature activations of A to feature activations of B. We use VGG-19 [27] to extract features and take five feature activations for style representation ('conv1_1', 'conv2_1', 'conv3_1', 'conv4_1',

'conv5_1'). Denote the ith feature activation as O_i, and the jth channel in O_i as O_{ij}, we compute the normalized histogram for O_{ij} of A and match it to the normalized histogram for O_{ij} of B, thus obtaining the remapped activations. The process is repeated for each channel in O_i. Denote the ith remapped activations as $R_i(O_i, A, B)$. The histogram loss is defined as

$$L_{histogram}(A, B) = \sum_{i=1}^{5} ||O_i(A) - R_i(O_i, A, B)||. \qquad (10)$$

Then we set L_{style} in Eq. (9) to $L_{histogram}$. We randomly initialize the style feature vector and use an Adam optimizer with learning rate 0.05 to optimize the vector. Some examples of the optimization process and results are shown in Figs. 7c and 7d.

6 GENERATOR DISSECTION

Our model can successfully learn to generate good looking APDrawings in multiple styles using a single network, and can generate APDrawings of "new styles" unseen in the training data. To better interpret our model, we explore the semantic meaning of convolution layers in the APDrawing generator G by visualizing feature maps and analyzing their relation to face semantics. Following [34], we measure the spatial agreement between thresholded feature map and facial part segmentation with intersection-over-union (IoU). For a convolution layer unit u and a facial part region r (e.g., upper lip, left eye, etc.), denote the feature map of u as F_u, the upsampled feature map as $F_u^\uparrow$, and the facial part region r's segmentation[5] as S_r. The IoU is calculated as

$$IoU_{u,r} = \frac{\mathbb{E}_p|(F_u^\uparrow > t_{u,r}) \cap S_r(p)|}{\mathbb{E}_p|(F_u^\uparrow > t_{u,r}) \cup S_r(p)|}$$

$$t_{u,r} = \arg\max_t \frac{I(F_u^\uparrow > t; S_r(p))}{H(F_u^\uparrow > t; S_r(p))}, \qquad (11)$$

---

4. Here, a "new" style portrait drawing means that the style is not one of the three target styles and unseen in the training data.

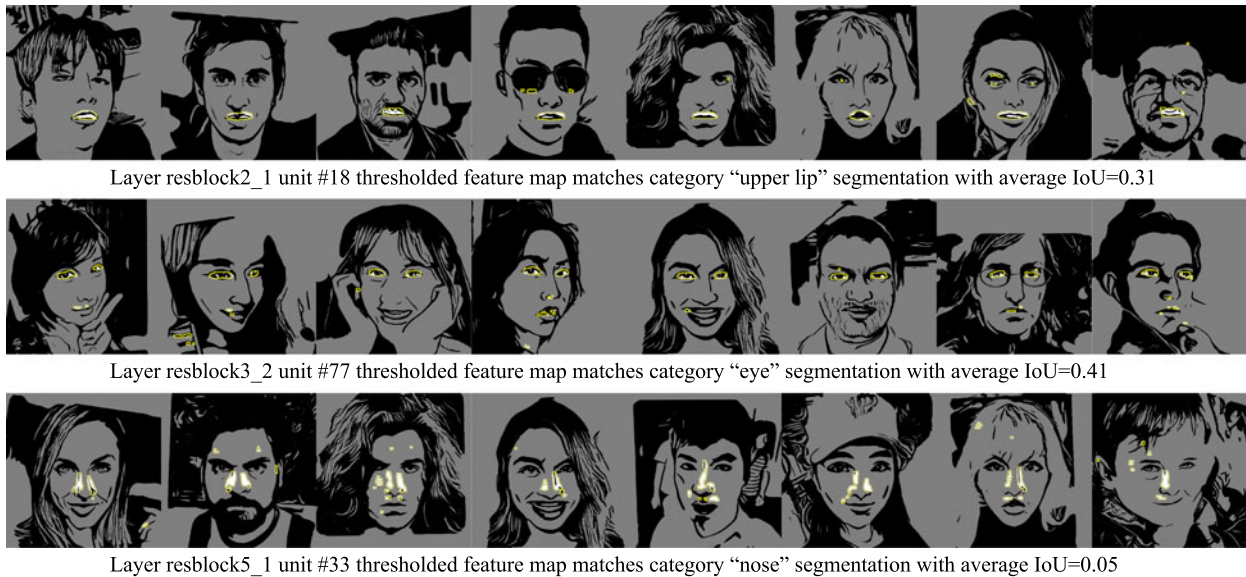5. A face parsing network [29] is used for facial part segmentation.

Layer resblock2_1 unit #18 thresholded feature map matches category "upper lip" segmentation with average IoU=0.31

Layer resblock3_2 unit #77 thresholded feature map matches category "eye" segmentation with average IoU=0.41

Layer resblock5_1 unit #33 thresholded feature map matches category "nose" segmentation with average IoU=0.05

Fig. 8. Visualizing interpretable units. The rows from top to bottom show units that best match "upper lip", "eye", and "nose" categories, respectively. The IoU is measured over the full test set of 154 images. For each unit, eight images with top IoU are shown, and the masks of thresholding the upsampled feature map ($F_u^\uparrow > t_{u,r}$) are outlined in yellow.

where p denotes face photos sampled from the face photo domain $\mathcal{P}$, the IoU is measured over a test set of face photos. The map $F_u^\uparrow > t_{u,r}$ produces a binary mask by thresholding the upsampled feature map at a fixed level $t_{u,r}$. S_r is a binary mask in which the foreground contains pixels belonging to the facial part region r. The threshold $t_{u,r}$ is computed by maximizing the information quality I/H where H is the joint entropy and I is the mutual information [35].

We use $IoU_{u,r}$ to find the facial parts related to each convolution layer unit and label each unit with the facial part that best matches it. The units with $\max_r IoU_{u,r} > 0.05$ are called "interpretable" units. Fig. 8 shows some examples of interpretable units with different labels. Among 5,505 convolution units in our generator G, 594 of them are interpretable units, showing that face semantic information is learned and incorporated during APDrawing generation.

7 EXPERIMENTS

We implemented our model in PyTorch. All experiments are performed on a computer with a Titan Xp GPU. The parameters in Eq. (1) are $\lambda_1 = 5 - \frac{4.5i}{N}$, $\lambda_2 = 5$, $\lambda_3 = \frac{4.5i}{N}$, $\lambda_4 = 1$, $\lambda_5 = 0.5 \cdot \mathbf{1}_{\{i > 100\}}(i)$, where $\mathbf{1}_A$ is the indicator function, i is the current epoch number, and N is the total epoch number ($N = 300$). We apply the quality loss after 100 epochs so that the model can learn a proper drawing first and is then optimized towards better quality.

7.1 Experiment Setup

Data. We collect face photos and APDrawings from the Internet and construct (1) a training corpus of 798 face photos and 625 delicate portrait line drawings, and (2) a test set of 154 face photos. Among the collected drawings, (1) 84 are labeled with artist *Charles Burns*, 48 are labeled with artist *Yann Legendre*, 88 are labeled with artist *Kathryn Rathke*, and 212 are from the website *vectorportral.com*, and (2) others have no tagged author/source information. We observed that both Charles Burns and Yann Legendre use similar thin

parallel lines to draw shadows, and so we merged drawings of these two artists into style1. We choose the drawings of Kathryn Rathke as style2 and the drawings of vectorportral as style3. Styles 1 and 2 have distinctive features: Kathryn Rathke uses flowing lines but few dark regions, while vectorportral uses thick lines and large dark regions. All the training images are resized and cropped to 512×512 pixels.

Training Process. It includes two steps: (1) training classifier C and (2) Training our model. We first train a style classifier C (Section 4.2.1) with the tagged drawings and data augmentation (including random rotation, translation and scaling). To balance the number of drawings in each style, we take all drawings from the first and second styles, but only part of the third style in the training stage of C, in order to achieve balanced training for different styles. In the second step, we use the trained classifier to obtain style feature vectors for all 625 drawings. We further augment training data using synthesized drawings. Training our network with the mixed data of real and synthesized drawings results in high-quality generation for all three styles; see Figs. 9, 10, and 11 for some examples, where our results of styles 1, 2, 3 are generated by feeding in style feature vectors $[1, 0, 0], [0, 1, 0], [0, 0, 1]$, respectively.

7.2 Comparisons

We compare our method with two state-of-the-art neural style transfer methods: Gatys [4], LinearStyleTransfer [7], and six unpaired image-to-image translation methods: DualGAN [8], CycleGAN [6], UNIT [9], MUNIT [10], ComboGAN [22] and DRIT [23].

For the two neural style transfer methods, i.e., Gatys [4] and LinearStyleTransfer [7], their inputs are a content image (face photo) and a style image (one of the collected artist line drawings). For the six unpaired image-to-image translation methods, i.e., DualGAN [8], CycleGAN [6], UNIT [9], MUNIT [10], ComboGAN [22] and DRIT [23], we retrained each comparison model using our training set, which

Fig. 9. Comparison with two state-of-the-art neural style transfer methods, i.e., Gatys [4] and LinearStyleTransfer [7].

consists of 978 photos, and both 625 collected real drawings and 353 synthesized drawings.

Comparisons with neural style transfer methods are shown in Fig. 9. Gatys' method fails to capture APDrawing styles because it uses the Gram matrix to model style as texture, but APDrawings have little texture. LinearStyleTransfer produces visually better results, although they are still not desired APDrawings: the generated drawings have many thick lines and they are produced in a rough manner.

Compared to these example-guided style transfer methods, our method learns from a set of APDrawings and generates delicate results for all three styles.

Comparisons with single-modal unpaired image-to-image translation methods are shown in Fig. 10. DualGAN and CycleGAN are both based on strict cycle-consistency loss. This causes a dilemma in photo-to-APDrawing translation: either a generated drawing looks like a real drawing (i.e., close to binary, containing large uniform regions) which cannot properly reconstruct the original photo, or a generated drawing has good reconstruction with grayscale changes but which does not look like a real drawing. Meanwhile, compared to CycleGAN, DualGAN is more grayscale-like, less abstract and worse in line drawing style. UNIT adopts feature-level cycle-consistency loss, which less constrains the results at the image level, making the face appear deformed. In comparison, our results both preserve face structure and have good image and line quality.

Comparisons with unpaired image-to-image translation methods that can deal with multi-modal or multi-domain translation are shown in Fig. 11. Results show that MUNIT does not capture the APDrawing styles and the results are more similar to a pencil drawing with shading and many gray regions. ComboGAN fails to capture all three representative styles, which performs slightly better on styles 2 and 3 than style 1. DRIT also fails to capture all three representa-
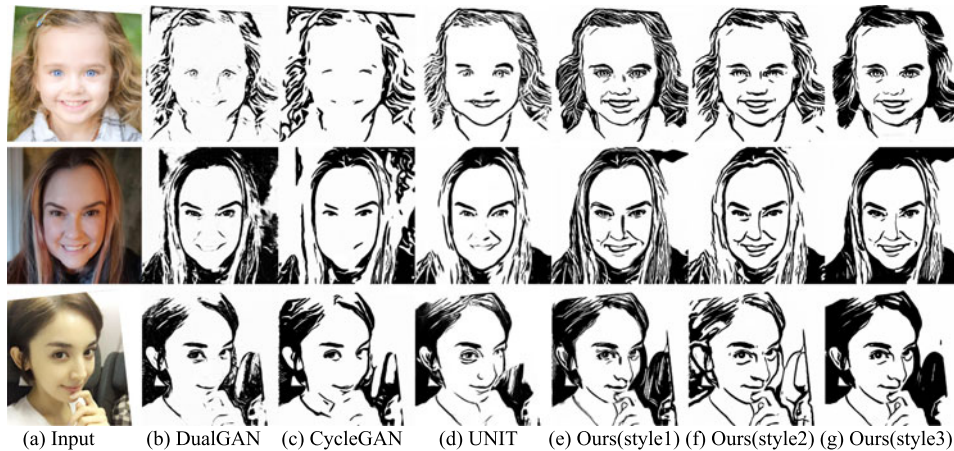

Fig. 10. Comparison with three single-modal unpaired image-to-image translation methods: DualGAN [8], CycleGAN [6], UNIT [9]. All methods are trained using the same training corpus with both real and synthesized drawings.
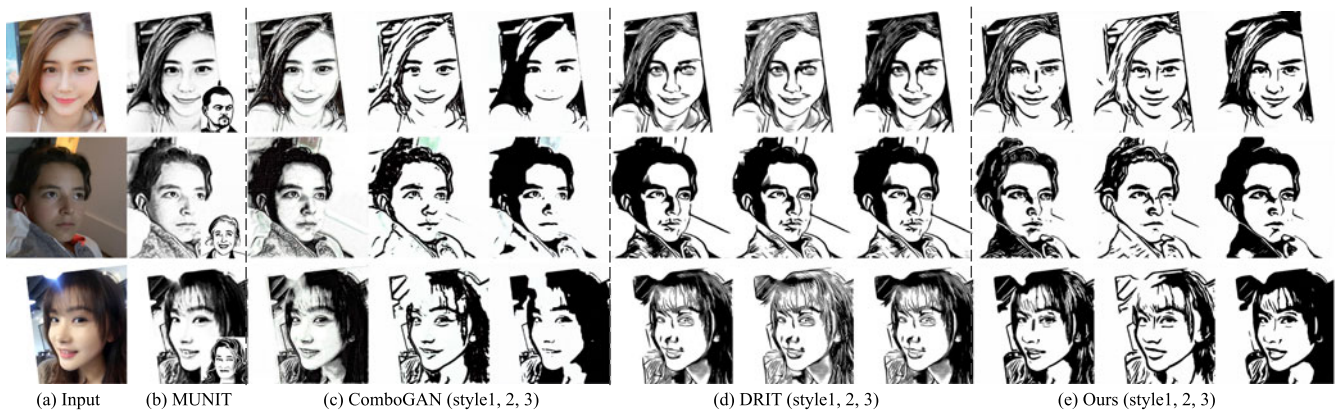

Fig. 11. Comparison with three unpaired image-to-image translation methods that can deal with multi-modal or multi-domain translation: MUNIT [10], ComboGAN [22], DRIT [23]. All methods are trained using the same training corpus with both real and synthesized drawings.

	ComboGAN	CycleGAN	Ours-pre	Ours
Rank1	23.7%	8.1%	23.1%	**45.1%**
Rank2	12.2%	8.9%	46.4%	32.6%
Rank3	35.1%	27.9%	21.9%	15.1%
Rank4	29.0%	55.2%	8.6%	7.2%

The ith row shows the percentages of different methods (ComboGAN [22], CycleGAN [6], our conference version (Ours-pre) [3] and Ours) being ranked as the ith among four methods.

tive styles; the results of all three styles look similar and only approximate the target styles 1 and 3. In comparison, our method generates distinctive results for three styles and reproduces the characteristics for each style well.

7.3 Quantitative Evaluation

User Study. Considering the artistic merits of portrait line drawings, we conduct a user study to compare our method with CycleGAN [6], ComboGAN [22] and our conference version (Ours-pre) [3]. LinearStyleTransfer, Gatys, Dual-GAN and UNIT are not included because of their lower visual quality. MUNIT and DRIT are not included because they obviously do not capture the target styles. We randomly sample 60 groups of images/drawings from the test set: 20 for style1 comparison, 20 for style2 and 20 for style3. Before the test, the participants went through some practice examples, and were given guidelines about the standard of good portrait line drawings. During the test, participants (1) were shown a photo, a real drawing (the style reference) and 4 generated drawings at a time, and (2) were asked to sort 4 results from best to worst. 54 participants attended the user study and 3,240 votes were collected in total.

Results of the percentages of each method ranked as 1st, 2nd, 3rd and 4th are summarized in Table 1. Our method ranks the best with 45.1% of the votes, which is higher than the other methods, i.e., ComboGAN, CycleGAN and our conference version, which rank the best in 23.7%, 8.1% and 23.1% of the votes. The average rank of our method is 1.84, lower compared to CycleGAN's 3.30, ComboGAN's 2.69 and our conference version's 2.16. We then conduct analysis of variance (ANOVA) between our method and each of other methods on the percentage of being ranked best by individual users. Pairwise ANOVA results are shown in Table 2. All of the p-values are $\ll 0.01$, justifying that the rejection of the null hypothesis and the differences between the means of our method and each of other three methods (ComboGAN, CycleGAN or our conference version) are statistically significant. A test boxplot of four methods is shown in Fig. 12. These results demonstrate that our

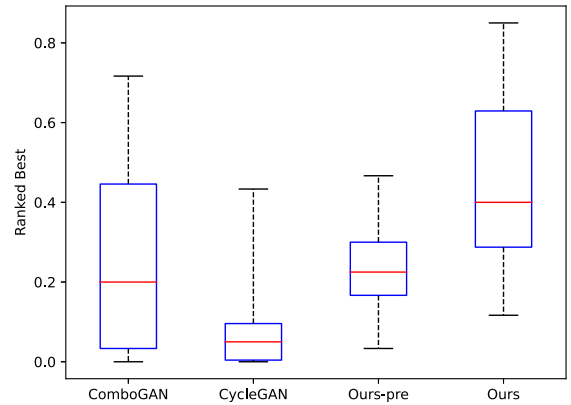| Pairwise comparison	Ranked Best
Ours vs. ComboGAN	p=3.57e-7
Ours vs. CycleGAN	p=1.32e-22
Ours vs. Ours-pre	p=2.09e-11

Fig. 12. Test boxplot [36] of four methods. In each box, the central red line indicates the median, and the bottom and top blue edges of the box indicate the 25% and 75% percentiles respectively. The dashed black line extends to the extreme data points.

method outperforms other methods. All generated drawings evaluated in the user study are presented in the appendix, available in the online supplemental material.

GAN Metric Evaluation. We adopt the Fréchet Inception Distance (FID) [37] to evaluate the similarity between the distributions of two drawing sets — one is the set of generated APDrawings for one style and the other is the set of collected true drawings for this style — where lower FID indicates better similarity. By changing the input style feature vector, we transform all face photos in the test set into three styles of APDrawings. The FID values between the set of generated APDrawings of each style and the collected drawings of the corresponding style are computed and summarized in Table 3. The results show that compared with the other multi-modal generation methods (MUNIT [10], ComboGAN [22], DRIT [23] and our conference version [3]), our method has lower FID on all three styles, indicating our method generates a closer distribution to the distribution of true drawings.

Quality Metric Model Evaluation. We apply the trained quality metric model M on generated drawings of different methods, and the quality scores are listed in Table 4. The score for each method is averaged on the test set. Our method achieves the highest score, indicating that our generated results have the best perceptual quality according to the trained metric model.

More Test Results. In addition to photos collected from Internet, we also test our method on photos from the CelebA-Mask-HQ Dataset [38]. The results are summarized in Appendix A5.3, available in the online supplemental material.

Methods	Style1 ↓	Style2 ↓	Style3 ↓
MUNIT [10]	194.3	267.4	242.8
ComboGAN [22]	184.9	144.1	141.4
DRIT [23]	82.8	135.0	119.9
Ours-pre [3]	88.3	139.0	108.2
Ours	**81.2**	**114.3**	**89.7**

The FID values are computed between the set of generated APDrawings of each style and the collected true drawings of the corresponding style.

TABLE 4
The Scores Predicted by Quality Metric Model M on the Results of Different Methods

Methods	Gatys	LST	DualGAN	CycleGAN	UNIT
Quality score	0.37	0.35	0.35	0.35	0.36
Methods	MUNIT	ComboGAN	DRIT	Ours-pre	Ours
Quality score	0.33	0.40	0.44	0.45	**0.51**

The score for each method is averaged on the test set. Higher quality score indicates better quality.

Fig. 13. Ablation study: (a) input photos, (b) results of removing relaxed cycle-consistency loss (i.e., using L_1 loss) and removing local discriminators, (c) results of removing relaxed cycle-consistency loss, (d) results of removing local discriminators, (e) results of removing HED in calculating relaxed cycle-consistency loss, (f) our results.

7.4 Ablation Study

We perform an ablation study based on the four key ingredients of our model: (1) relaxed cycle consistency loss, (2) the quality loss based on the quality metric model, (3) local discriminators, and (4) HED edge extraction. Results show that they are all essential to our model. In Appendix A4, available in the online supplemental material, three more ablation studies are presented: the first focuses on the style feature vector and style loss, the second focuses on the truncation loss, and the third focuses on how face region information is utilized in the discriminator.

As shown in Fig. 13b, without relaxed cycle consistency loss and local discriminators, facial features are often missing, e.g., the nose is missing in all three rows, and eye details are missing in the first and third rows. Removing only relaxed cycle consistency loss (Fig. 13c) preserves more facial feature regions (e.g., the nose in the second row) when compared to Fig. 13b, but some parts (e.g., the nose in the third row) are still missing compared to our method (Fig. 13f). Removing only local discriminators (Fig. 13d) produces few missing parts: although the results are much better than Fig. 13b in terms of preserving facial structure, some facial features are not drawn in the desired manner, i.e., some black regions or shadows (that are usually drawn near facial boundaries or hair) appear near the nose. When both relaxed cycle consistency loss and local discriminators are used, results (Fig. 13f) preserve all facial feature regions and no undesired black regions or shadows appear in faces. These results show that both relaxed cycle consistency loss and local discriminators help to preserve facial feature regions and are complementary to each other : (1) the relaxed cycle loss works in a more global and general way, it alleviates the need to hide information and helps preserve outlines (since lines are more easily missing in nose, eyes and lips regions, their effects on these regions are more visible), and (2) as a comparison, the local discriminators work

Fig. 14. Ablation study on quality loss: (a) input photos, (b) results of removing quality loss, (c) our results. Artifacts are highlighted in red boxes.

TABLE 5
Fréchet Inception Distance (FID) of the Ablation Studies

Methods	Style1 ↓	Style2 ↓	Style3 ↓	Avg Δ
w/o $L_{relaxed}$, w/o D_{l*}	102.9	126.8	105.1	16.53
w/o $L_{relaxed}$	88.0	132.9	107.0	14.23
w/o D_{l*}	89.4	142.3	101.4	15.97
w/o HED	84.1	114.8	103.4	5.70
w/o $L_{quality}$ [3]	88.3	139.0	108.2	16.77
Ours	**81.2**	**114.3**	**89.7**	/

The FID values are computed between the set of generated APDrawings of each style and the collected true drawings of the corresponding style.

in a local way, dedicated to eyes, nose and lips, improving drawings and eliminating artifacts in these local regions.

As shown in Fig. 13e, without HED edge extraction in the relaxed cycle consistency loss calculation (i.e., calculating LPIPS perceptual similarity between the input and reconstructed photo), the lines are often discontinuous or blurred, e.g., the nose outlines in the second row are discontinuous (upper right), and the noses in the first and third rows are blurred and messy. In comparison, our results have clear, sharp and continuous lines, demonstrating that using HED edge extraction helps the model to generate clearer and more complete lines.

As shown in Fig. 14b, without the quality loss, the results contain more artifacts including undesired dark regions and parallel lines on the face (highlighted in red boxes). In comparison, our results in Fig. 14c are cleaner and of better quality.

The quantitative evaluation of the above ablation studies are reported in Table 5. FID scores of our results are lower (better) in all three styles than the ablated versions. We further compute the average difference between our FID and each ablated version, shown as "Avg Δ".

Contributions of Each Component. (1) Qualitative and quantitative results show that cycle consistency loss and local discriminators are complementary to each other, and work together to better preserve facial features:

- without both components, the average Δ is larger than without a single component;
- without a single component, the average Δ is also large, indicating these two components contribute largely to the final results.

(2) In addition, the visual differences between results of removing the quality loss and ours are easily visible, and the quantitative difference is also large, indicating the quality loss helps remove undesirable artifacts and improves quality. (3) Compared to these three components, HED itself has smaller impact on the final results.

8 CONCLUSION

In this paper, we propose a method for high quality APDrawing generation using asymmetric cycle mapping. Our method can learn multi-style APDrawing generation from web data of mixed styles using an additional style feature vector input and a soft classification loss. In particular, our method makes use of unpaired training data and improves upon [3] in the following four aspects: (1) a novel

quality metric for APDrawings is proposed; (2) based on the quality metric, a new quality loss that is consistent with human perception is introduced to guide the model toward better looking drawings; (3) a "new" style APDrawing generation mechanism is proposed; and (4) the model is dissected by visualizing feature maps and exploring face semantics. Experiments and a user study demonstrate that our method can (1) generate high quality and distinctive APDrawing results for the styles in training data and new unseen styles, and (2) outperforms state-of-the-art methods.

REFERENCES

[1] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 735–10 744.

[2] R. Yi, M. Xia, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Line drawings for face portraits from photos using global and local structure based GANs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3462–3475, Oct. 2021.

[3] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Unpaired portrait drawing generation via asymmetric cycle mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8214–8222.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[7] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast image and video style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3804–3812.

[8] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2868–2876.

[9] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[10] X. Huang, M. Liu, S. J. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 179–196.

[11] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2479–2486.

[12] J. Johnson, A. Alahi, and L. Fei-Fei , "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[13] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1349–1357.

[14] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1510–1519.

[15] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[16] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[17] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1060–1069.

[18] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5908–5916.

[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.

[20] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.

[21] H. Dou, C. Chen, X. Hu, and S. Peng, "Asymmetric CycleGan for unpaired NIR-to-RGB face image translation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 1757–1761.

[22] A. Anoosheh, E. Agustsson, R. Timofte, and L. V. Gool, "ComboGAN: Unrestrained scalability for image domain translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 896–8967.

[23] H. Lee, H. Tseng, J. Huang, M. Singh, and M. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 35–51.

[24] F. L. Wauthier, M. I. Jordan, and N. Jojic, "Efficient ranking from pairwise comparisons," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 109–117.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[26] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a master of steganography," *CoRR*, vol. abs/1712.02950, pp. 1–6, 2017.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[29] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan, "Mask-guided portrait editing with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3431–3440.

[30] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[31] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[32] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, "Neural style transfer: A review," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 11, pp. 3365–3385, 2020.

[33] P. Wilmot, E. Risser, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," *CoRR*, vol. abs/1701.08893, pp. 1–14, 2017.

[34] D. Bau *et al.*, "GAN dissection: Visualizing and understanding generative adversarial networks," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–19.

[35] D. R. Wijaya, R. Sarno, and E. Zulaika, "Information quality ratio as a novel metric for mother wavelet selection," *Chemometrics Intell. Lab. Syst.*, vol. 160, pp. 59–71, 2017.

[36] R. V. Hogg and J. Ledolter, *Engineering Statistics*. New York, NY, USA: MacMillan, 1987.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

[38] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5548–5557.
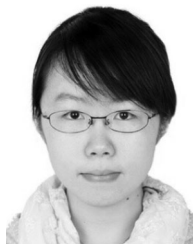
Ran Yi received the BEng and PhD degrees from Tsinghua University, Beijing, China, in 2016 and 2021, respectively. She is currently an assistant professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include computer vision, computer graphics, and machine intelligence.

Yong-Jin Liu (Senior Member, IEEE) received the BEng degree from Tianjin University, Tianjin, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. He is currently a professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computational geometry, computer graphics, and computer vision. For more information, please visit https://cg.cs.tsinghua.edu.cn/people/~Yongjin/Yongjin.htm

Yu-Kun Lai (Member, IEEE) received the BS and PhD degrees in computer science from Tsinghua University, Beijing, China, in 2003 and 2008, respectively. He is currently a professor with the School of Computer Science and Informatics, Cardiff University, U.K. His research interests include computer graphics, computer vision, geometric modeling, and image processing. For more information, please visit https://users.cs.cf.ac.uk/Yukun.Lai/

Paul L. Rosin is currently a professor with the School of Computer Science and Informatics, Cardiff University, U.K. Previous posts include lecturer with the Department of Information Systems and Computing, Brunel University London, U.K., research scientist with the Institute for Remote Sensing Applications, Joint Research Centre, Ispra, Italy, and lecturer with the Curtin University of Technology, Perth, Australia. His research interests include low level image processing, performance evaluation, shape analysis, facial analysis, cellular automata, non-photorealistic rendering, and cultural heritage. For more information, please visit http://users.cs.cf.ac.uk/Paul.Rosin/

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.