Sketch-to-Art: Synthesizing Stylized Art Images From Sketches

Bingchen Liu[1,2], Kunpeng Song[1,2], Yizhe Zhu[2], and Ahmed Elgammal[1,2]

[1] Playform - Artrendex Inc., USA
{bingchen, kunpeng, elgammal}@artrendex.com
https://www.playform.io/
[2] Department of Computer Science, Rutgers University, USA
yizhe.zhu@rutgers.edu

Abstract. We propose a new approach for synthesizing fully detailed art-stylized images from sketches. Given a sketch, with no semantic tagging, and a reference image of a specific style, the model can synthesize meaningful details with colors and textures. Based on the GAN framework, the model consists of three novel modules designed explicitly for better artistic style capturing and generation. To enforce the content faithfulness, we introduce the dual-masked mechanism which directly shapes the feature maps according to sketch. To capture more artistic style aspects, we design feature-map transformation for a better style consistency to the reference image. Finally, an inverse process of instance-normalization disentangles the style and content information and further improves the synthesis quality. Experiments demonstrate a significant qualitative and quantitative boost over baseline models based on previous state-of-the-art techniques, modified for the proposed task (17% better Frechet Inception distance and 18% better style classification score). Moreover, the lightweight design of the proposed modules enables the high-quality synthesis at 512×512 resolution.

1 Introduction

Synthesizing fully colored images from human-drawn sketches is an important problem, with several real-life applications. For example, colorizing sketches following a specified style can significantly reduce repetitive works in storyboarding. Fruitful results have been achieved in applying deep learning to the art literature [1–3]. Most research works have focused on synthesizing photo-realistic images [4], or cartoonish images [5] from sketches. In this paper, we focus on rendering an image in a specific given artistic style based on a human-drawn sketch as input. The proposed approach is generic, however, what distinguish art images from other types of imagery is the variety of artistic styles that would affect how a sketch should be synthesized into a fully colored and textured image.

In the history of art, style can refer to an art movement (Renaissance style, Baroque style, Impressionism, etc.), or particular artist style (Cezanne style,

Fig. 1. Synthetic images from sketches with different styles. Upper panel: our approach synthesizes from different styles (art movements). The first row shows the reference images from each style, and the second row shows the generated images. Lower panel: our model synthesizes from specific artists' styles by taking paintings from the artists as reference.

Monet style, etc.), or a specific artwork style [6]. Style encompasses different formal elements of art, such as the color palette, the rendering of the contours (linear or painterly), the depth of the field (recessional or planer), the style of the brush strokes, the light (diffused or light-dark contrast), etc.

We propose a novel generation task: given an input sketch and a style, defined by a reference image, or an artist name, or a style category, we want to synthesize a fully colored and textured image in that style following the sketch, as shown in Figure 1. Previous works on synthesizing from sketches do not allow users to specify a style reference [7, 4]. We propose a new model to achieve this task, which takes the input sketch and sample reference image(s) from art-historical corpus, defining the desired style, to generate the results.

A sketch contains very sparse information, basically the main composition lines. The model has to guess the semantic composition of the scene and synthesize an image with semantic context implied from the training corpus. E.g., given a corpus of landscape art of different styles, the model needs to learn how different parts of the scene correlate with colors and texture given a choice of style. In the proposed approach, no semantic tagging is required for the sketches nor the style images used at both training and generation phases. The model implicitly infers the scene semantic.

The proposed approach is at the intersection between two different generation tasks: Sketch-to-image synthesis and Style transfer. Sketch-to-image synthesis focuses on rendering a colored image based on a sketch, where recent approaches focused on training deep learning models on a corpus of data, e.g., [7]. However, these approaches do not control the output based on a given style.

More importantly, in our case the reference image should only define the style not the content details. For example, the model should infer that certain plain area of the sketch is sky, trees, mountains, or grass; and infer details of these regions differently given the style image, which might not have the same semantic regions all together (see examples in Figure 1).

Fig. 2. Synthesized samples in 512×512 resolution. The first column are the input sketches hand-drawn by human, and the first row are the style reference images.

On the other hand, style transfer focuses on capturing the style of an (or many) image and transfer it to a content image [8]. However, we show that such models are not applicable to synthesize stylized images from sketches because of the lack of content in the sketch, i.e., the approach need to both transfer style from the reference image and infer content based on the training corpus.

The proposed model has three novel components, which constitute the technical contributions of this paper, based on a GAN [9] infrastructure,:

Dual Mask Injection. A simple trainable layer that directly imposes sketch constraints on the feature maps, to increase content faithfulness.

Feature Map Transfer. An adaptive layer that applies a novel transformation on the style image's feature map, extracting only the style information without the interference of the style images' content.

Instance De-Normalization. A reverse procedure of Instance Norm [10] on the discriminator to effectively disentangle the style and content information.

2 Related Work

Image-to-Image Translation: I2I aims to learn a transformation between two different domains of images. The application scenario is broad, including object transfiguration, season transfer, and photo enhancement. With the generative power of GAN [9], fruitful advances have been achieved in the I2I area. Pix2pix [11] established a common framework to do the one-to-one mapping for paired images using conditional GAN. Then a series of unsupervised methods such as CycleGAN [12–15] were proposed when paired data is not available. Furthermore, multi-modal I2I methods are introduced to simulate the real-world many-to-many mapping between image domains such as MUNIT and BicycleGAN [16–21]. However, those I2I methods can not generate satisfying images from the coarse sketch's domain, nor can they adequately reproduce the styles from the reference image, as will be shown in the experiments.

Neural Style Transfer: NST transfers textures and color palette from one image to another. In [8], the problem was formulated by transforming the statistics of multi-level feature maps in CNN layers in the form of a Gram Matrix. The follow-up works of NST have multiple directions, such as accelerating the transfer speed by training feed-forward networks [22, 23], and capturing multiple styles simultaneously with adaptive instance normalization [24, 10]. However, little attention has been paid on optimizing towards artistic style transfer from a sketch, nor on improving the style transfer quality regarding the fine-grained artistic patterns among the various NST methods [25].

Sketch-to-image Synthesis: Our task can be viewed from both the I2I and the NST perspectives, while it has its unique challenges. Firstly, unlike datasets such as horse-zebra or map-satellite imagery, the sketch-to-painting dataset is heavily unbalanced and one-to-many. The information in the sketch domain is ambiguous and sparse with only few lines, while in the painting's domain is rich and diverse across all artistic styles. Secondly, "style transfer" approaches focus mainly on color palettes and "oil painting like" textures but pays limited attention to other important artistic attributes such as linear vs. painterly contours, texture boldness, and brush stroke styles. Thirdly, it is much harder for NST methods to be semantically meaningful, as the optimization procedure of NST is only between few images. On the contrary, with a sketch-to-image model trained on a large corpus of images, learning semantics is made possible (the model can find the common coloring on different shapes and different locations across the images) and thus make the synthesized images semantically meaningful.

To our knowledge, there are few prior works close to our task. AutoPainter [26] propose to do sketch-to-image synthesis using conditional GANs, but their model is designed towards cartoon images. ScribblerGAN [7] achieves user-guided sketch colorization but requires the user to provide localized color scribbles. SketchyGAN [4] is the state-of-the-art approach for multi-modal sketch-to-image synthesis, focusing on photo-realistic rendering of images conditioned on object categories. Furthermore, [5, 27, 28] accomplish the conditioned colorization with a reference image, but they only optimize towards datasets that lack style and content variances. None of those works is specifically geared towards artistic images with the concern of capturing the significant style variances. In contrast, Figure 1 and Figure 2 demonstrate the ability of our model to capture the essential style patterns and generate diversely styled images.

3 Methods

In this section, we provide an overview of our model and introduce the detailed training schema. Then three dedicated components will be described.

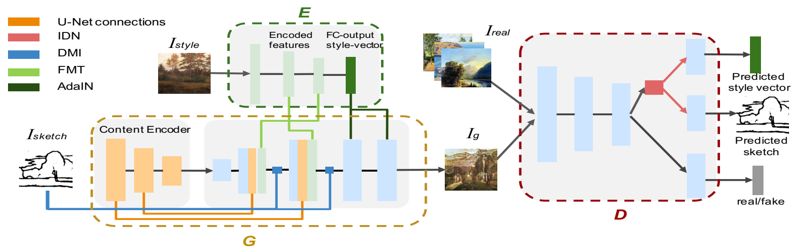

Fig. 3. Overview of the model structure. G adopts a U-Net structure [29]. It takes the features of the reference image I_{style} from E, runs style-conditioned image synthesis on the input sketch image I_{sketch}, and outputs I_g. D takes as input an image (alternatively sampled from real and generated images) and gives three outputs: a predicted style vector, a predicted sketch, and a real/fake signal of the image

As shown in Figure 3, our model consists of three parts: a generator G, a discriminator D, and a separately pre-trained feature-extractor E. Either an unsupervised Auto-Encoder or a supervised DNN classifier, such as VGG [30] can work as the feature-extractor. During the training of G and D, E is fixed and provides multi-level feature-maps and a style vector of the reference image I_{style}. The feature-maps serve as inputs for G, and the style vector serves as the ground truth for D. We train G and D under the Pix2pix [11] schema. In the synthesis phase, the input style is not limited to one reference image. We can always let E extract style features from multiple images and combine them in various ways (averaging or weighted averaging) before feeding them to G. Apart from the following three modules, our model also includes a newly designed attentional residual block and a patch-level image gradient matching loss, please refer to the appendix for more details.

3.1 Reinforcing Content: Dual Mask Injection

In our case, the content comes in the form of a sketch, which only provides sparse compositional constraints. It is not desirable to transfer composition elements from the style image or training corpus into empty areas of the sketch that should imply textured areas. Typically when training a generator to provide images with diverse style patterns, the model tends to lose faithfulness to the content, which results in missing or excrescent shapes, objects, and ambiguous contours (especially common in NST methods). To strengthen the content faithfulness, we introduce Dual-Mask Injection layer (DMI), which directly imposes the sketch information on the intermediate feature-maps during the forward-pass in G.

Given the features of a style image, in the form of the conv-layer activation $f \in \mathbb{R}^{C\times H\times W}$, we down sample the binary input sketch to the same size of f and use it as a feature mask, denoted as $M_s \in [0,1]^{H\times W}$. The proposed DMI layer will first filter out a *contour-area feature* f_c and a *plain-area feature* f_p by:

$$f_c = M_s \times f, \quad f_p = (1 - M_s) \times f, \tag{1}$$

where $\times$ is element-wise product. For f_c and f_p, the DMI layer has two sets of trainable weights and biases, $w_c, b_c, w_p, b_p \in \mathbb{R}^{C\times1\times1}$, that serve for a **value relocation** purpose, to differentiate the features around edge and plain area:

$$f'_c = w_c \times f_c + b_c, \quad f'_p = w_p \times f_p + b_p \tag{2}$$

Finally, the output feature maps of DMI will be $f' = f'_c + f'_p$.

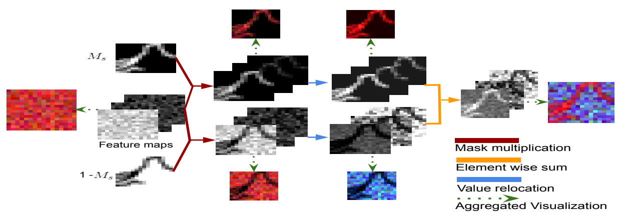

Fig. 4. Forward flow of Dual-Mask Injection layer, we aggregate the first three channels in feature-map as an RGB image for visualization purpose

A real-time forward flow of the DMI layer is shown in Figure 4. Notice that, when $w = 1$ and $b = 0$, the output feature will be the same as the input, and we set the weights and bias along the channel dimension so that the model can learn to impose the sketch on the feature-maps at different degrees on different channels. By imposing the sketch directly to the feature-maps, DMI layer ensures that the generated images have correct and clear contours and compositions. While DMI serves the same purpose as the masked residual layer (MRU) in SketchyGAN [4], it comes with almost zero extra computing cost, where MRU requires three more convolution layers per unit. In our experiments, our model is two times faster in training compared with SketchyGAN, while yields better results on art dataset. Moreover, the lightweight property of DMI enables our model to achieve great performance on 512×512 resolution, while SketchyGAN was studied only on 128×128 resolution.

3.2 Disentangling Style and Content by Instance De-Normalization

To better guide G, D is trained to adequately predict an style latent representation of the input image as well as the content sketch that should match with the input image. Ideally, a well-disentangled discriminator can learn a style representation without the interference of the content information, and retrieve the content sketch regardless of the styles.

Several works have been done in specifically disentangling style and content [31, 32], but they only work around the generator side, using AdaIN or Gram-matrix to separate the factors. To train D to effectively disentangle, we propose the Instance De-Normalization (IDN) layer. IDN takes the feature-map of an image as input, then reverses the process of Instance-Normalization [24, 10] to produces a style vector and a content feature map. In the training phrase of our model, IDN helps D learn to predict accurate style-vectors and contents of the ground truth images, therefore, helps G to synthesis better.

In AdaIN or IN, a stylized feature-map is calculated by:

$$f_{styled} = \sigma_{style} \times \frac{(f - \mu(f))}{\sigma(f)} + \mu_{style}, \tag{3}$$

where $\sigma(\cdot)$ and $\mu(\cdot)$ calculate the variance and mean of the feature map f respectively. It assumes that while the original feature map $f \in \mathbb{R}^{C \times H \times W}$ contains the content information, some external μ_{style} and σ_{style} can be collaborated with f to produce a stylized feature map. The resulted feature map possesses the style information while also preserves the original content information.

Here, we reverse the process for a stylized feature map f'_{style} by: first predict μ_{style} and σ_{style}, $(\mu_{style}, \sigma_{style}) \in \mathbb{R}^{C \times 1 \times 1}$ from f'_{style}, then separate them out from f'_{style} to get $f_{content}$ which carries the style-invariant content information. Formally, the IDN process is:

$$\mu'_{style} = Conv(f'_{style}), \quad \sigma'_{style} = \sigma(f'_{style} - \mu'_{style}), \tag{4}$$

$$f_{content} = \frac{(f'_{style} - \mu'_{style})}{\sigma'_{style}}, \tag{5}$$

where $Conv(\cdot)$ is 2 conv layers. Note that unlike in AdaIN where μ_{style} can be directly computed from the known style feature f_{style}, in IDN the ground truth style feature is unknown (we don't have f_{style}), thus we should not naively compute the mean of f'_{style}. Therefore, we use conv layers to actively predict the style information, and will reshape the output into a vector as μ'_{style}. Finally, we concatenate μ'_{style} and σ'_{style} to predict the style-vector with MLP, and use conv layers on $f_{content}$ to predict the sketch. The whole IDN process can be trained end-to-end with the target style-vector and target-sketch. Unlike other disentangling methods, we separate the style and content from a structural perspective that is straightforward while maintaining effectiveness.

3.3 Reinforcing Style: Feature Map Transformation

To approach the conditional image generation task, previous methods such as MUNIT [16] and BicycleGAN [17] use a low-dimensional latent vector of I_{style} extracted from some feature extractors E as the conditioning factor. However, we argue that such vector representation carries limited information in terms of style details. Therefore, it is more effective to directly use feature-maps as a conditional factor and information supplier.

Nevertheless, the image feature-maps in CNN usually carry both the style information and strong content information. Such content information can be problematic and is undesired. For instance, if the style image is a house while the input sketch implies a lake, we do not want any shape or illusion of a house

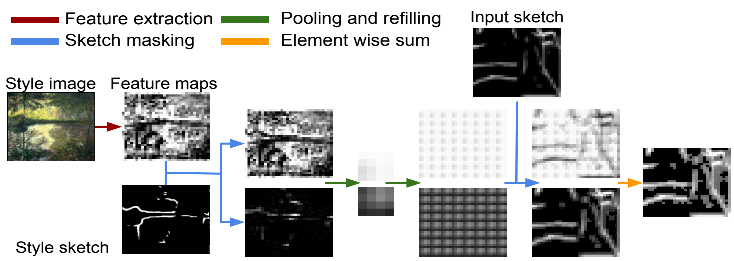

Fig. 5. Process flow of Feature-Map Transformation

within the lake region in the synthesized image. To get rid of the content information, while keeping the richness of the style features, we propose the Feature Map Transformation (FMT). FMT takes the input sketch I_{sketch}, the sketch of the style image $I_{style-sketch}$, and the feature map of the style image f_{style} as inputs, and produces transformed feature-maps f_t. f_t only preserves the desired style features of the style image and discards its content structure. Note that $I_{style-sketch}$ is extracted using simple edge detection methods and f_{style} comes from the feature-extractor E, that are both easy to achieve.

The proposed FMT is a fixed formula without parameter-training. The procedure is illustrated in Figure 5 with five steps. In step 1, we use $I_{style-sketch}$ as a mask to filter f_{style} and get two sets of features, i.e., f^c_{style} that only have the features around the contours and f^p_{style} with features on plain areas. In step 2, we apply a series of max-pooling and average-pooling to this filtered yet sparse feature values to extract a 4×4 feature-map for each part, namely $f^{c'}_{style}$ and $f^{p'}_{style}$. In step 3, we repeatedly fill the 4×4 $f^{c'}_{style}$ and $f^{p'}_{style}$ into a f^c_t and a f^p_t with the same size of f_{style}. In step 4, we use I_{sketch} as a mask in the same manner to filter these two feature maps, and get $f^{c\prime}_t$ and $f^{p\prime}_t$ that have the features of I_{style} but in the shape of I_{sketch}. Finally in step 5, we add the results to get $f_t = f^{c\prime}_t + f^{p\prime}_t$ as the output of the FMT layer. We then concatenate f_t to its corresponding feature-maps in G for the synthesis process.

The pooling operation collects the most distinguishable feature values along spatial channel in f_{style}, then the repeat-filling operation expands the collected global statistics, finally the masking operation makes sure the transformed feature-map will not introduce any undesired content information of the style image. FMT provides accurate guidance to the generation of fine-grained style patterns in a straightforward manner, and unlike AdaIN [24] and Gram-matrix [8] which require higher-order statistics. In practice, we apply FMT from the 16×16 to 64×64 feature maps. FMT contains two max-pooling layers and one avg-pooling layer. These layers have 5×5 kernel and stride of 3, which give us a reasonable receptive field to get the highlighted style features. We first use max-pooling to get the most outstanding feature values as the style information, then use the average-pooling to summarize the values into a "mean feature" along the spatial dimensions. The average-pooling can help smooth out some local bias (peak values) and get a more generalized style representation.

3.4 Objective Functions

Besides the original GAN loss, auxiliary losses are adopted in our model. Specifically, when the input is a real image, we train D with a *style loss* and a *content loss*, which minimize the *MSE* of the predicted style vector and sketch with the ground-truth ones respectively. Meanwhile, G aims to deceive D to predict the same style vector as the one extracted from I_{style} and output the same sketch as I_{sketch}. These auxiliary losses strengthen D's ability to understand the input images and let G generate more accurate style patterns while ensuring the content faithfulness. During training, we have two types of input for the model, one is the paired data, where the I_{sketch} and I_{style} are from the same image, and the other is randomly matched data, where I_{sketch} and I_{style} are not paired. We also have a reconstruction *MSE* loss on the paired data.

Formally, D gives three outputs: $S(I)$, $C(I)$, $P(I)$, where $S(I)$ is the predicted style vector of an image I, $C(I)$ is the predicted sketch, and $P(I)$ is the probability of I being a real image. Thus, the loss functions for D and G are:

$$\begin{aligned}\mathcal{L}(D) = & \mathbb{E}[\log(P(I_{real}))] + \mathbb{E}[\log(1 - P(G(I_{sketch}, I_{style})))] \\ & + MSE(S(I_{real}), E(I_{real})) + MSE(C(I_{real}), I_{real-sketch}), \qquad (6) \\ \mathcal{L}(G) = & \mathbb{E}[\log(P(G(I_{sketch}, I_{style})))] + MSE(C(G(I_{sketch}, I_{style})), I_{sketch}) \\ & + MSE(S(G(I_{sketch}, I_{style})), E(I_{style})). \qquad (7)\end{aligned}$$

and the extra loss for G: $MSE(G(I_{sketch}, I_{style}), I_{style})$ is applied when the inputs are paired. I_{real} is randomly sampled real data and $I_{real-sketch}$ is its corresponding sketch, and I_{sketch} and I_{style} are randomly sampled sketches and referential style images as the input for G.

4 Experiments

We first show comparisons between our model and baseline methods, and then present the ablation studies. The code to reproduce all the experiments with detailed training configurations are included in the supplementary materials, along with a video demonstrating our model in real time. A website powered by the proposed model is available online at: https://www.playform.io, where people can synthesize 512×512 images with their free-hand drawn sketches.

Dataset: Our dataset is collected from Wikiart [33] and consists of 10k images with 55 artistic styles (e.g., impressionism, realism, etc.). We follow the sketch creation method described by [4] to get the paired sketch for each painting. We split the images into training and testing set with a ratio of $9:1$. All the comparisons shown in this section were conducted on the testing set, where both the sketches and the art images were unseen to the models.

Metrics: We use **FID** [34] and a **classification accuracy** for the quantitative comparisons. FID is a popular image generation metric that provides a perceptual similarity between two sets of images. For our task, we generate I_g from all the testing sketches using the same I_{style}, and compute the FID between I_g and the images from the same style. We repeat this process for all style images. We further employ a more direct metric to compute the style classification accuracy of I_g, which leverages a VGG model pre-trained on ImageNet [35] and fine-tuned on art for style classification. Such model is more invariant to compositions, and focuses more on the artistic style patterns. We record how many of I_g are classified correctly as the style of I_{style}, which reflects how well the generator captures the style features and translates them into I_g. The style-classification accuracy for VGG is 95.1%, indicating a trustworthy performance.

4.1 Comparison to Baselines:

MUNIT [16] (unsupervised) and BicycleGAN [17] (supervised) are the two state-of-the-art I2I models that are comparable to our model for their ability to do conditional image translation. SketchyGAN [4] is the latest model that is dedicated to the sketch-to-image task. Pix2pix [11] is the fundamental model for the I2I process. In this section, we show comparisons between our model and the aforementioned models. We also include the results from the classical NST method by [8] as a representative of that class of methods. For SketchyGAN (noted as "Pix2pix+MRU" since its main contribution is MRU) and Pix2pix, we adopt their model components but use our training schema to enable them the style-conditioned sketch-to-image synthesis.

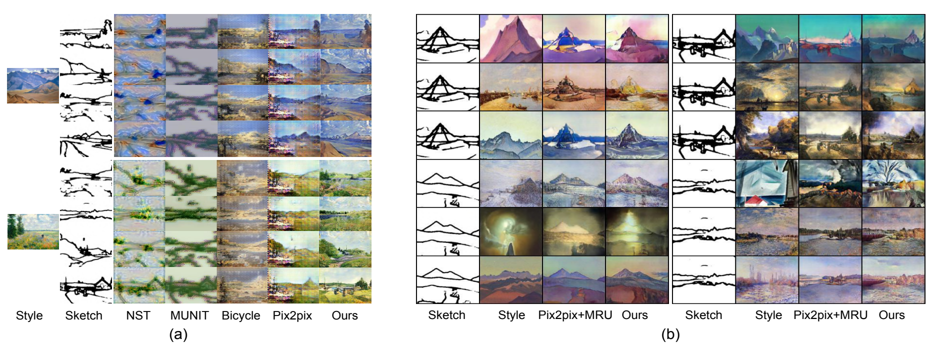

Fig. 6. Qualitative comparison to baseline models

All the tested models are trained on images with 128×128 resolution due to the compared models' capacity restrictions (our model can easily upscale to 512×512). We make sure all the compared models have a similar generative capacity by having a similar amount of weights in their respective generators. Except for Pix2pix, all the compared models have more total parameters than ours and require a longer time to train. We tested multiple hyper-parameter settings on baseline models and reported the highest figures.

Qualitative results are shown in Figure 6. Except "Pix2pix+MRU", all the other methods can hardly handle the art dataset and do not produce meaningful results. Due to the limited information in sketches, NST and MUNIT can hardly generate meaningful images, BicycleGAN only generates images with blurry edges and fuzzy colors, and Pix2pix consistently gets undesirable artifacts. Notice that, models like MUNIT and BicycleGAN do work well on datasets with simple sketches, where the images share one standard shape (only cats, shoes) and are well registered with white background, and without any artistic features involved (semantically diversified, different color palettes and texture). In contrast, images in art dataset are much more complicated with multiple objects and different compositions, which result in bad performance for these previous models and show the effectiveness of our proposed components.

Table 1. Quantitative comparison to baseline models

	NST	MUNIT	Pix2pix	BicycleGAN	Pix2pix+MRU	Ours
FID ↓	6.85	7.43 ± 0.08	7.08 ± 0.05	6.39 ± 0.05	5.05 ± 0.13	**4.18** ± 0.11
Classification Score ↑	0.182	0.241 ± 0.012	0.485 ± 0.012	0.321 ± 0.009	0.487 ± 0.002	**0.573** ± 0.011

The quantitative results, shown in Table 1, concur with the qualitative results. It is worth noticing that, while "Pix2pix+MRU" generates comparably visual-appealing images, our model outperforms it by a large margin especially in terms of style classification score, which indicates the superiority of our model in translating the right style cues from the style image into the generated images.

Comparison to MRU from SketchyGAN: Since SketchyGAN is not proposed for the exemplar-based s2i task, the comparison here is not with SketchyGAN as it is, but rather with a modified version with our training schema to suit our problem definition. While the "Pix2pix+MRU" results look good from a sketch colorization perspective, they are not satisfactory from the artistic style transfer perspective compared with the given style images (eg., texture of flat area, linear vs painterly). As shown in Figure 6-(b), "Pix2pix+MRU" tends to produce dull colors on all its generations, with an undesired color shift compared to the style images. In contrast, our results provide more accurate color palette and texture restoration. Apart from the quantitative result, "Pix2pix+MRU" is outperformed by our model especially in terms of the fine-grained artistic style features, such as color flatness or fuzziness and edge sharpness.

4.2 Ablation study

We perform ablation studies to evaluate the three proposed components using a customized Pix2pix+MRU model as the baseline. More experiments can be

Table 2. Ablation study of the proposed components

	baseline	with DMI	with FMT	with 2B	with IDN
FID ↓	4.77 ± 0.14	4.84 ± 0.09	4.43 ± 0.15	4.73 ± 0.09	**4.18**± 0.11
Classification Score ↑	0.485 ± 0.012	0.507 ± 0.007	0.512 ± 0.009	0.479 ± 0.013	**0.573** ± 0.011

found in the appendix, including a quantitative evaluation of the content and style disentanglement performance of our model, an effectiveness analysis of an image gradient matching loss for better texture generation, and a more detailed comparison between AdaIN and the proposed FMT for style transfer.

In this section, we show both the cumulative comparisons and individual benefits for each modules. When evaluating FMT, we replace the AdaIN layer on lower level image features with FMT, and show better performance of FMT compared to AdaIN. IDN changes the structure of the discriminator, so when validating the effectiveness of IDN, we add an extra baseline model which has a discriminator naively predicting the sketch and style-vector (noted as "with 2B") from its two separate convolution branches.

In Table 2, the proposed components are cumulatively added to the model. FID and classification scores consistently show that each of the proposed components contributes to the generation performance.

FMT and IDN bring the most significant boost in FID and Classification, respectively. It is worth noticing how the added two branches on the discriminator (with 2B) hurt the performance and how IDN reverses that drawback and further boosts the performance. We hypothesis that naively adding two branches collected conflicting information during training (content and style) and made the models harder to converge. In contrast, IDN neatly eliminates the conflict thanks to its disentangling effect, and takes advantage of both the content and style information for a better generation performance.

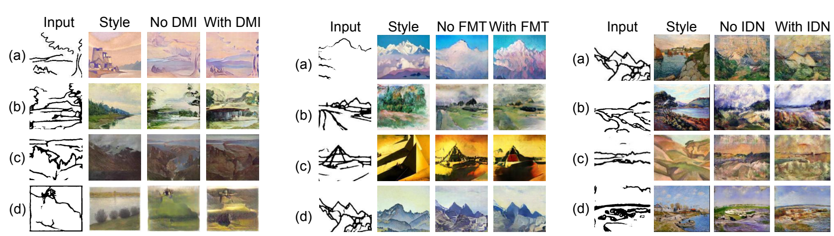

Fig. 7. Qualitative comparisons of DMI, FMT and IDN

Qualitative Comparisons: In Figure 7, the left panel shows how DMI boosts the content faithfulness to the input sketches. In row (a), the model without DMI misses the sketch lines on the top and right portion, while DMI redeems all the lines in I_{sketch}. In row (b), I_g without DMI is affected by I_{style} which has a flat and empty sky and misses the lines on the top portion of I_{sketch}. In contrast, the model with DMI successfully generates trees, clouds, and mountain views following the lines in I_{sketch}. In row (c), I_g without DMI totally messes up the shapes in the mid area in I_{sketch}, while, in the one with DMI, all the edges are correctly shaped with clear contrast.

The middle panel in Figure 7 shows how FMT helps translate the style from the input style images. In row (a), FMT helps generate the correct colors and rich textures in the mountain as in I_{style}. In row (b), I_g with FMT inherits the smoothness from I_{style} where the colors are fuzzy and the edges are blurry, while removing FMT leads to sharp edges and flat colors. When I_{style} is flat without texture, FMT is also able to resume the correct style. Row (c) and row (d) demonstrate that when I_{style} is clean and flat without fuzziness, I_g without FMT generates undesired textures while FMT ensures the accurate flat style.

The right panel in Figure 7 shows how IDN helps maintain a better color palette and generally reduce the artifacts. In row (a) and (b), there are visible artifacts in the sky of the generated images without IDN, while IDN greatly reduces such effects. In row (b) and (c), the images without IDN shows undesired green and purple-brown colors that are not in the given style images, and IDN has better color consistency. In row (c) and (d), there are clear color-shifts in the generated images, which then been corrected in the model with IDN. Please refer to the Appendix for more qualitative comparisons.

4.3 Human Evaluation

We conduct human survey to validate our model's effectiveness compared to the "Pix2pix+MRU" baseline model. The survey is taken by 100 undergraduate students to ensure the quality. In each question, the user is presented with a sketch, a style image and the generated images from baseline (Pix2pix+MRU) and our model (option letters are randomly assigned to reduce bias) and asked: "Which image, a or b, better captures the style in colorizing the sketch?". We collected 1000 results and 63.3% selects our model as the better performer. 13.3% selects "hard to tell". Only 23.4% prefer the baseline model.

4.4 Qualitative Results on Multi Domains

While focused on art, our model is generic to other image domains with superior visual quality than previous models. On more commonly used datasets CelebA and Fashion Apparel, our model also out-performs the baselines and shows the new state-of-the-art performance. Figure 8 shows the results on multiple image domains. In figure (c), the glasses in the sketches are successfully rendered in the generated images even when there is no glasses in the style image. Similarly, the moustache is correctly removed when there is moustache in the style images but is not indicated in the sketches. It's worth noticing that the gender of generated images follows the sketch instead of style. These results show clear evidence that the model learns semantics of input sketch from the training corpus.

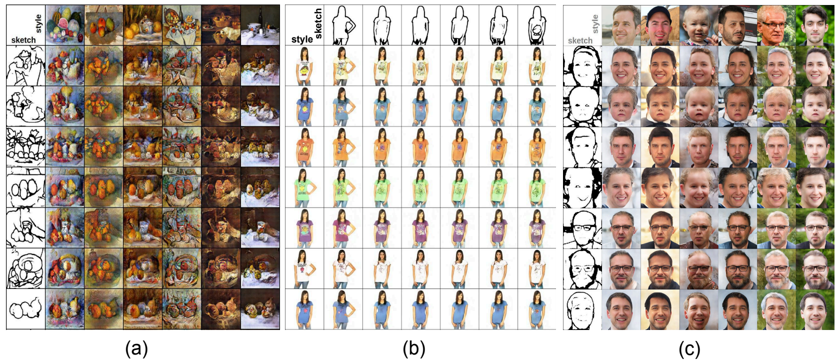

Fig. 8. Synthesize on still-life painting, apparel, and portrait at 512^2 resolution.

5 Discussions

The model yields consistent performance when we try different input settings for the synthesis, including using multiple reference style images rather than one. During the experiments, we also discovered some interesting behaviors and limitations that worth further research. For instance, learning from the corpus can be a good thing for providing extra style cues apart from the style image, however, it may also cause conflicts against it, such as inaccurate coloring and excess shapes. It is worth study on how to balance the representation the model learns from the whole corpus and from the referential style image, and how to take advantage of the knowledge from the corpus for better generation. We sincerely guide the readers to the Appendix for more information.

6 Conclusion

We approached the task of generating artistic images from sketch while conditioned on style images with a novel sketch-to-art model. Unlike photo-realistic datasets, we highlighted and identified the unique properties of artistic images and pointed out the different challenges they possess. Respectively, we proposed methods that can effectively addressed these challenges. Our model synthesizes images with the awareness of more comprehensive artistic style attributes, which goes beyond color palettes, and for the first time, identifies the varied texture, contours, and plain area styles. Overall, our work pushes the boundary of the deep neural networks in capturing and translating various artistic styles, and makes a solid contribution to the sketch-to-image literature.

Acknowledgement At https://create.playform.io/sketch-to-image, demo of the model in this paper is available. The research was done while Bingchen Liu, Kunpeng Song and Ahmed Elgammal were at Artrendex Inc.

References

1. Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M.: Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. arXiv preprint arXiv:1706.07068 (2017)
2. Elgammal, A., Mazzone, M., Liu, B., Kim, D., Elhoseiny, M.: The shape of art history in the eyes of the machine. arXiv preprint arXiv:1801.07729 (2018)
3. Kim, D., Liu, B., Elgammal, A., Mazzone, M.: Finding principal semantics of style in art. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), (IEEE) 156–163
4. Chen, W., Hays, J.: Sketchygan: Towards diverse and realistic sketch to image synthesis. In: CVPR. (2018) 9416–9425
5. Zhang, L., Ji, Y., Lin, X., Liu, C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In: ACPR. (2017) 506–511
6. Schapiro, M., Schapiro, M., Schapiro, M., Schapiro, M.: Theory and philosophy of art: Style, artist, and society. Volume 4. George Braziller New York (1994)
7. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: CVPR. (2017) 5400–5409
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR. (2016) 2414–2423
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 2672–2680
10. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017) 1125–1134
12. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017) 2223–2232
13. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NIPS. (2017) 700–708
14. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR. (2018) 8789–8797
15. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. In: ICCV. (2017) 2849–2857
16. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV. (2018) 172–189
17. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: NIPS. (2017) 465–476
18. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented cycleGAN: Learning many-to-many mappings from unpaired data. In: ICML. (2018) 195–204
19. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: ECCV. (2018) 35–51
20. Liu, B., Song, K., Zhu, Y., de Melo, G., Elgammal, A.: Time: Text and image mutual-translation adversarial networks. arXiv preprint arXiv:2005.13192 (2020)
21. Zhu, Y., Min, M.R., Kadav, A., Graf, H.P.: S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6538–6547

22. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016) 694–711
23. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. In: ECCV. (2018)
24. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV. (2017) 1501–1510
25. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M.: Neural style transfer: A review. arXiv preprint arXiv:1705.04058 (2017)
26. Liu, Y., Qin, Z., Luo, Z., Wang, H.: Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. arXiv preprint arXiv:1705.01908 (2017)
27. Zhang, L., Li, C., Wong, T.T., Ji, Y., Liu, C.: Two-stage sketch colorization. In: SIGGRAPH Asia 2018 Technical Papers. (2018) 261
28. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR. (2019) 2337–2346
29. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. (2015) 234–241
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
31. Kazemi, H., Iranmanesh, S.M., Nasrabadi, N.: Style and content disentanglement in generative adversarial networks. In: WACV. (2019) 848–856
32. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR. (2019) 4401–4410
33. : Wikiart. (https://www.wikiart.org/)
34. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS. (2017) 6626–6637
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115** (2015) 211–252