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

A More Qualitative Results

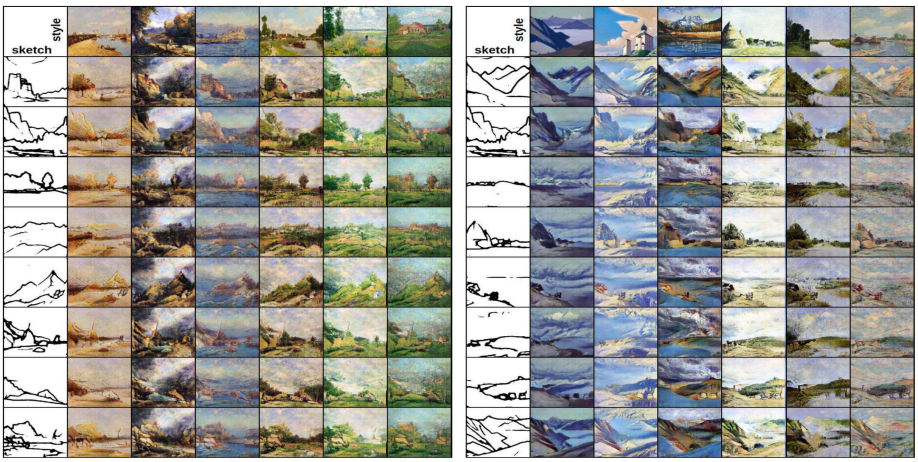

Fig. 1. Qualitative results from our model trained on genre **landscape**

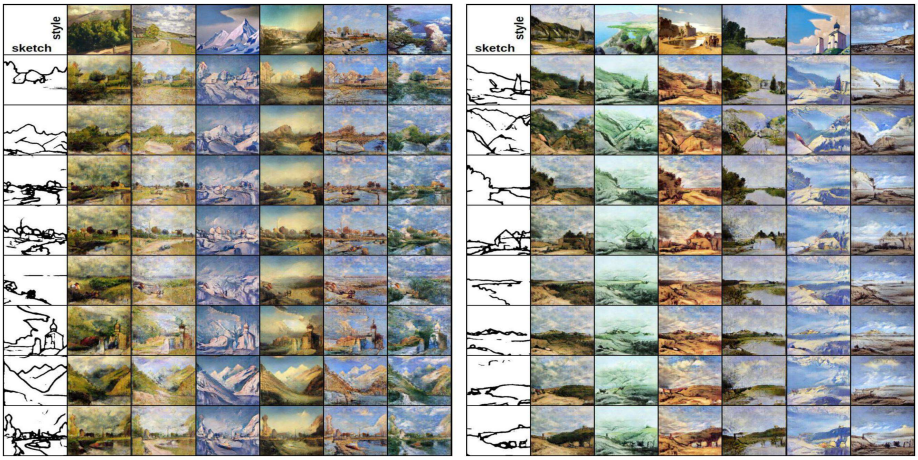

Fig. 2. Qualitative results from our model trained on genre **landscape**

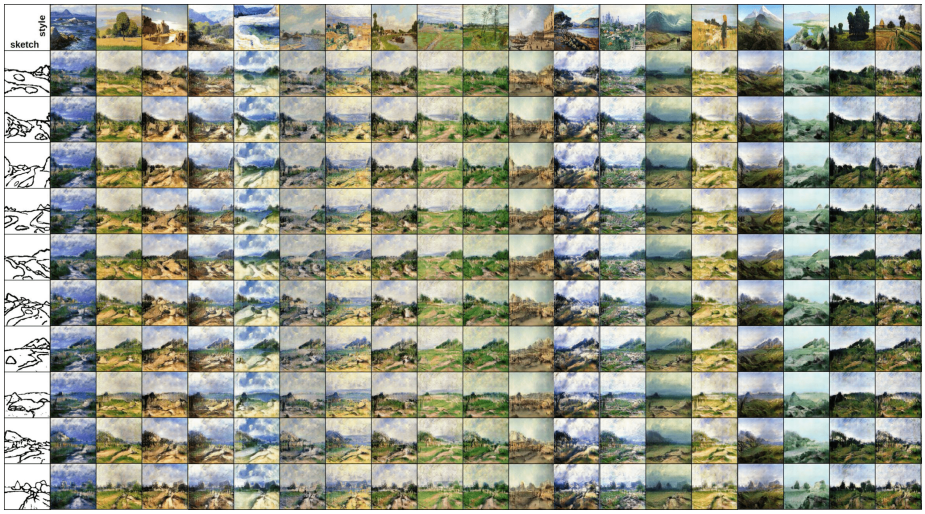

Fig. 3. Synthesizing with hand-draw sketches

Fig. 4. Synthesizing with hand-draw sketches

Fig. 5. Qualitative results from our model trained on genre **still-life**

B Model Components

B.1 Attention-based Residual Block

Description: We customize the convolution structure in G and D based on the residual blocks proposed by [1]. Specifically, we apply a channel-wise attention following [2] on the second convolution layer in each block. To our knowledge, we are the first to adopt such attention-based layer within residual blocks. This tweak brings significant image quality boost in our task from the convolution layers used in Pix2Pix and BicycleGAN [3,4] while maintains minimum extra computing cost. In sketch-to-image task, traditional convolution layers or residual convolutions suffer from fuzzy artifices and can hardly generate diverse colors. The proposed attention-based residual block largely improved such scenario for the baseline models. All the experimental results we present in this paper are based on this tweak. Further experiments are required to validate its effectiveness in general tasks, however, it is an orthogonal component that is beyond the discussion scope of this paper.

The tweak of the attention-based residual block is illustrated in Figure 6. It consists of two convolution layers and one fully-connected layer (linear layer). It takes two input, one is the feature-map f from the previous layer, and the other one is an style vector V_{style} from our feature extractor E. During training, the two convolution layers will compute a new feature-map f', while the linear layer will take V_{style} as input and output a channel-wise weight w_{style}. Unlike traditional residual block which directly add f' back to f, w_{style} will provide the weights to scale the values in each channel of f' before the add-back operation.

Intuition: We assume that different channels in the feature-map carry the information that corresponding to different artistic styles. If we have a weight vector that can control the attendance of each channel, i.e. mute some channels' value to zero and amplify some other channels' value, it will make the residual block easier to learn a diversified features. Also, the extra style vector provides more information during the generation process. The generative power is therefore largely increased.

During training, the extra linear layer in the residual block introduces almost none extra computing time. However, it makes the model much faster to converge (the model us able to generate meaningful images usually after just one epoch of training), and also results in much better final performance.

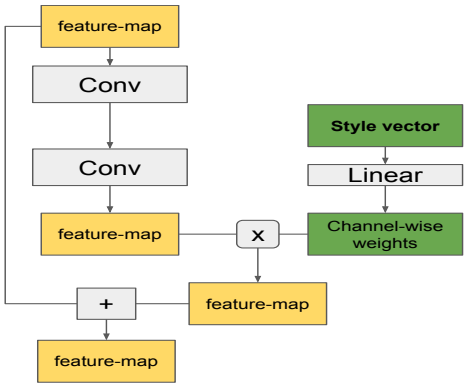

Fig. 6. Attention-based residual block

B.2 Image Gradient Matching Loss

A critical element of style is whether contours are linear (sharp) or painterly and whether planer areas are flat or textured. To capture these differences, we propose a Gradient Matching loss that integrates statistics about image gradient. Specifically, we match the gradient statistics of the generated image and their feature maps to the ones of the style image as a new training objective.

We use a patched version of gradient matching on image level and also on the conv-layer activation level. For a given input I and a target T where $(I, T) \in \mathbb{R}^{C \times H \times W}$, we first divide them spatially into 8×8 patches and compute the gradient loss within each patch, then we will average among all patches to get the final loss. To ignore the content differences between the generated image and target image, we match the mean and variance of the gradients in each patch instead of directly matching by pixels as follows:

$$\mathcal{L}_{gradient} = \frac{1}{n}\sum_{p=0}^{n}(||\mu(\nabla I_p) - \mu(\nabla T_p)||^2 + ||\sigma(\nabla I_p) - \sigma(\nabla T_p)||^2). \tag{1}$$

Table 1. FID for the gradient-matching loss.

	baseline (Pix2pix+MRU)	ours w/o G-loss	ours with G-loss	ours with Gram-matrix
mean	4.77	4.43	**4.28**	4.51
std	0.14	0.15	0.04	0.09

The intuition of using gradient matching to capture style features is similar to gram-matrix matching [5,6,7]. However, gram-matrix carries more of the color palette information and even some shape information, and usually leads to similar yet artificial textures at unwanted locations. Gradient matching preserves the diversity and is more representative on various style textures, such as the brush stroke styles and contour sharpness and fuzziness.

Table 1 shows the effectiveness of the gradient matching loss, we also compared it with the loss that matches the gram matrix of the images in the same manner. From our experiments on data from other domains, including photo-realistic human faces and fashion apparels, this image gradient matching loss does not improve the result. It is due to the fact that images from those domains do not have a comparable gradient variance as arts with various styles. Instead, they share the same gradient statistics over the whole dataset, thus gradient matching has no effect.

B.3 The Style and Content Disentanglement

Our Dual-Mask Injection (DMI) and Instance De-Normalization (IDN) modules are designed to strengthen the content faithfulness of the generated images to the given sketch, which further lead to a better content and style separation capability. To quantitatively show the effectiveness of the proposed modules, we took the edge maps of a set of images as input sketches, paired them with random style images and then extracted the new edge maps from the generated images. In the end, regardless of the style differences, the edge maps from the original image and the generated image should match as much as possible. And a better matched edge map indicates a better content faithfulness.

Table 2. Content faithfulness comparison for DMI and IDN

	baseline	with DMI only	with IDN only	with DMI and IDN
L1 ($\times 1000$) $\downarrow$	4.1	1.8	2.7	1.3
PDAR $\downarrow$	0.23	0.13	0.21	0.11

As shown in Table 2, we compute the L1-distance and the Pixel Disagreement Ratio to evaluate the edge map consistency between the input "sketch" and the output stylized image. DMI makes the biggest performance boost and IDN also contributes to a better style and content separation.

B.4 Comparison between FMT and AdaIN

Adaptive Instance Normalization approaches the style transfer task by shifting the source image's feature statistics to match those of the target image. However, while AdaIN is effective for the stylization task of one or a few style images, it has one undesired side-effect when applied on models that are trained on large corpus of images of various styles. As shown in Figure 7, the model trained with AdaIN exhibits artifact patterns that do not belong to any style images. Such defect caused by AdaIN is also discovered in [**?**], where the authors hypothesis the reason as the Generator deliberately seeking to "fake" a region of high signals in the feature-maps in order to compliment the normalized matching requirements of AdaIN.

Table 3. Style transfer comparison between AdaIN and FMT

	AdaIn at 16,32,64	FMT as 16, AdaIn at 32,64	FMT at 16, 32, AdaIn at 64	FMT at 16,32,54
Gram matrix L2 ($\times 1000$) $\downarrow$	2.35 ± 1.4	2.14 ± 0.9	1.83 ± 1.2	1.47 ± 0.9

Unlike AdaIN, our proposed Feature-Map-Transformation (FMT) does not apply any force to manipulating the features generated by the Generator. Instead, it provides the "style" information by concatenating a supplementary feature map from the style images, making the generating process unlikely to have the defect. According to our experiments, no such defect is discovered in our model trained with FMT. Moreover, the style transfer performance is also boosted according to the style classification experiment in the main paper when replacing the AdaIN with FMT. Here in Table 3 we provide a more dedicated experiment by measuring the L2 distance between the gram matrix of the generated images and the style images, over 10000 generated samples. We first use AdaIN on all layers of feature-map at resolution of $16^2, 32^2, 64^2$, then we gradually replace AdaIN by FMT. At each replacement, we can see a more consistent gram matrix between the generated images and the style images, indicating a better aligned style statistics.

Fig. 7. Defects caused by AdaIN. For each sketch, the first row shows the generated images with AdaIN applied on early feature-maps, the second row are samples from model which replace AdaIN with FMT.

B.5 More Ablation Samples for IDN

Unlike DMI and FMT, IDN improves the overall generative quality by enabling the Discriminator with the ability to capture the essential content and style features. It is hard to intuitively and selectively point out one direction IDN mainly focuses on. So apart from the FID score, we put more generated samples to qualitatively demonstrate the benefits of it. According to our observation, we conclude two major aspects of IDN's contribution: 1. less artifacts(notice the sky part) and more re-fined image synthesis, and 2. more accurate color consistency to the referential style image and no color-shift effects (notice the 3rd, 6th, 8th style).

Fig. 8. Generated images without IDN module

Fig. 9. Generated images with IDN module

C Qualitative Results on Other Image Domains

Even though focused on art, our model is generic and can be applied to other sketch-to-image tasks. Below we show the results of our model trained on apparel and human face data (the apparel dataset is from kaggle: https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset, and the human face dataset is FFHQ: https://github.com/NVlabs/ffhq-dataset). Note that since these datasets do not have the artistic style variances that we are interested in, we do not think the power of the proposed modules, especially FMT, can be adequately reflected. And we do not use the image gradient matching loss because there is no texture patterns that we want the model learn from these datasets. However, our model does show the state-of-the-art performance in general sketch-to-image tasks. Most importantly, it shows the evidence that the model learns semantics from the training corpus, as pointed out in Figure 11 and Figure 12.

Fig. 10. Qualitative results from our model trained on apparel. For the three sets of images, the first column is the style images, and the first row is the sketches. Individually, for the right-most set, we input random art images as the style image which the model never saw before. And the model is still able to get the correct shapes and colors.

Fig. 11. Qualitative results from our model trained on human face. Note that how the glasses in the sketches are successfully drawn on the generated images even when there is no glasses in the style image; and how the moustache is correctly removed when there is moustache in the style images but is not indicated in the sketches.

D Synthesis from multiple style images

While our model is trained to take one sketch and one style image as input, there are many ways to take advantage of the model and syntheses artful image from sketches. And we believe those applications indicate the potential of our model in the field of creative art creation.

Taking a set of style images, we can let E extract their features, and manipulate the set of features such as averaging them, or combine them with a given ratio, to get the summarized features. Then feed the summarized features to G for the generation. Examples can be found in Figure 13 and Figure 14. They show that our model is able to produce high quality generation and manipulate the mixed style patterns well, thus generate meaningful images.

Even-though trained on pair sketch/image, our model works consistently with human hand-draw sketches. Actually, we find that the extracted sketch from paintings are fairly similar to simple lines that human can draw. The generations from human-drawn sketches can be found in Figure 3 and Figure 4.

E Discussions and Limitations

Our discussions on the sketch to art task and some limitations in current model can be found in Figure 15, Figure 16.

Fig. 12. Qualitative results from our model trained on human face. We use random images from other domains as the style image for our model, including Pokemon, paintings, and shoes, which are not seen by the model during training. Note how the model still able to draw the essential face even when the style images do not have a face at all, showing the ability of the model learning semantics from the training data. It also shows that our model has a firm content faithfulness to the sketch.

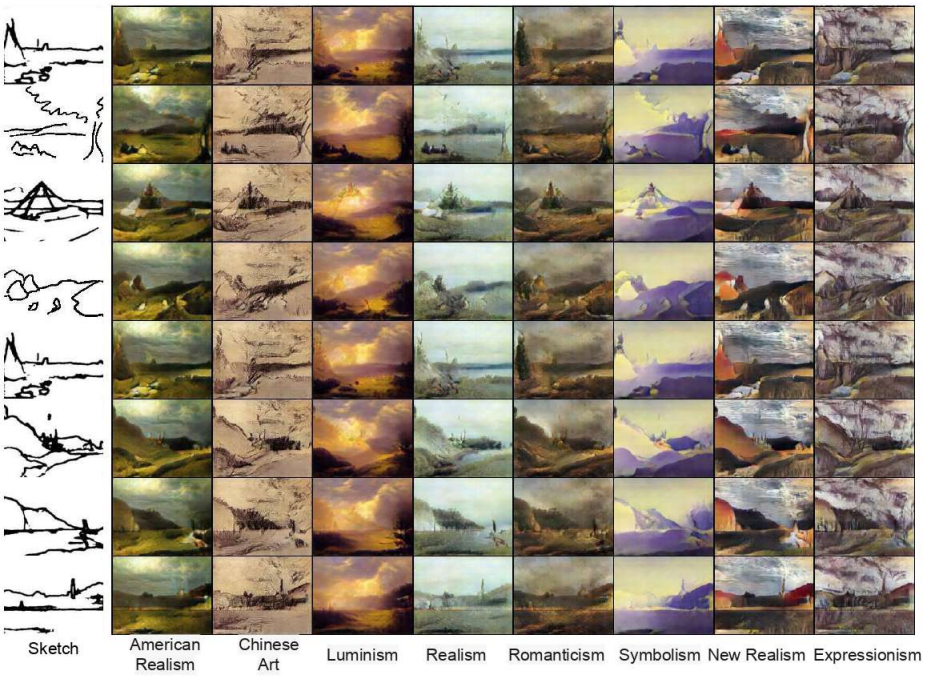

Fig. 13. Synthesizing by averaging 8 images from the same style: We take 8 paintings from the same style, and average their extracted features from the feature extractor E, then use the result features for the image generation process.

Fig. 14. Synthesizing by averaging 8 images from the same artists: We do the same operation as in Figure 13 but for the same artists rather than same styles.

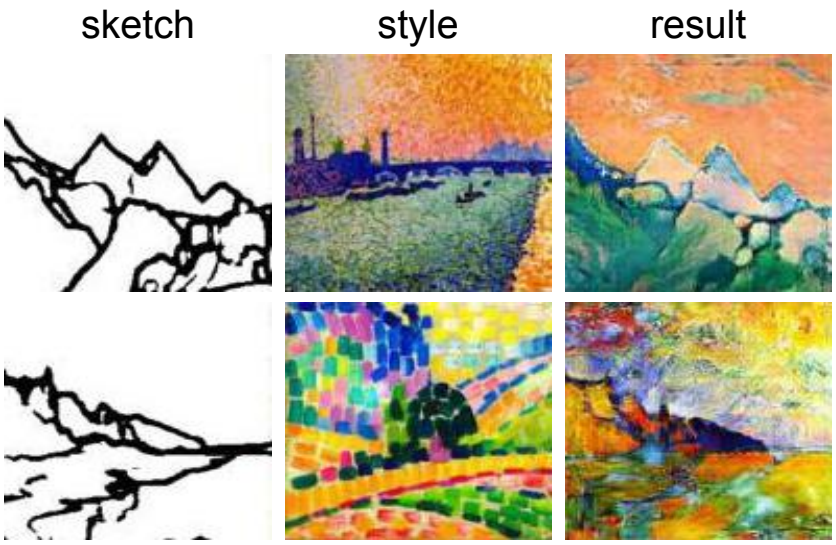

Fig. 15. Limitations: While our model gains marked progress in synthesizing artistic images from sketches, some style patterns are hard to be captured and well-represented by the current model, e.g., the pointillism style and some styles with large patches of colors. Our model has a relatively inconsistent performance on pointillism style. In row 1 where I_{style} has an intense pointy texture over the whole composition, while I_g is trying to imitate the pointy technique around the sky area, the result shows more of flatness across the whole image. Style with large patches of color is the one on which our model has a generally unsatisfying performance. As shown in row 2, I_g can hardly reproduce the neatly arranged color patches in I_{style}, even though it achieves the correct color palette. We believe some dedicated style recognition and generation methods can be developed for these two styles.

Fig. 16. Effect of reference image vs. corpus: During the training of the model, we assume that the model is able to learn some global knowledge from the whole corpus in synthesizing the images. This may contain some semantic features and more abstract style patterns. For example, in row 1, there is a house in I_{sketch} but I_{style} is a grassland with no color indication for the house. However, the model is still able to properly colorize the house with black roof and window. Similarly in row 2, despite that there is no mountain in I_{style}, the model surprisingly generates the mountain with a red color tune which is not appeared from the referential style image. Learning from the corpus can be a good thing for providing extra style cues apart from I_{style}, however, it may also cause conflicts against I_{style}, 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 hoe to take advantage of the knowledge from the corpus for better generation.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 770–778 3
2. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. (2018) 7132–7141 3
3. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017) 1125–1134 3
4. 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 3
5. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR. (2016) 2414–2423 5
6. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. In: ECCV. (2018) 5
7. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M.: Neural style transfer: A review. IEEE transactions on visualization and computer graphics (2019) 5