An Unpaired Sketch-to-Photo Translation Model

Runtao Liu[1]* **Qian Yu**[2]*(✉), **Stella Yu**[2]
Peking University[1] University of California, Berkeley[2]
runtao219@gmail.com {qianyu1023,stellayu}@berkeley.edu

Abstract

Sketch-based image synthesis aims to generate a photo image given a sketch. It is a challenging task; because sketches are drawn by non-professionals and only consist of strokes, they usually exhibit shape deformation and lack visual cues, i.e., colors and textures. Thus translation from sketch to photo involves two aspects: shape and color (texture). Existing methods cannot handle this task well, as they mostly focus on solving one translation. In this work, we show that the key to this task lies in decomposing the translation into two sub-tasks, shape translation and colorization. Correspondingly, we propose a model consisting of two sub-networks, with each one tackling one sub-task. We also find that, when translating shapes, specific drawing styles affect the generated results significantly and may even lead to failure. To make our model more robust to drawing style variations, we design a data augmentation strategy and re-purpose an attention module, aiming to make our model pay *less* attention to distracted regions of a sketch. Besides, a conditional module is adapted for color translation to improve diversity and increase users' control over the generated results. Both quantitative and qualitative comparisons are presented to show the superiority of our approach. In addition, as a side benefit, our model can synthesize high-quality sketches from photos inversely. We also demonstrate how these generated photos and sketches can benefit other applications, such as sketch-based image retrieval.

1 Introduction

Human free-hand sketch[1] has been adopted as an expression and communication method since pre-historic times. It is highly expressive: although a sketch only consists of strokes, it is amazingly recognizable by human beings. On the other hand, sketching is intuitive and is easy to deliver. Without resorting to other complicated tools, a pen and a piece of paper allow a person to express the ideas in his/her mind. Thanks to the popularization of touch-screen devices, sketching has become even easier than before as people can directly create an image with their fingers. Given the above natures, sketch has been applied in many application scenarios, such as education and designing.

With sketch playing a more critical role in people's lives, sketch understanding attracts increasing attention from the computer vision community. Researchers have investigated this problem from various aspects and have achieved impressive progress, e.g., sketch recognition, sketch parsing/grouping, and sketch-based image/video retrieval. Now the most advanced recognition model can achieve nearly 80% classification accuracy on a sketch benchmark dataset; while an SBIR model knows how to match a sketch with photos having similar content(s). Recall the reaction when we see a sketch image, as we identify the object it represents, we also *picture* its original appearance. Here we want to ask, can a machine imagine the real look of an object represented by a sketch? In other words, given a specific sketch, is it possible for a model to synthesize a corresponding photo?

We call the task of generating a photo given a sketch as ***S**ketch-**B**ased **I**mage **S**ynthesis* (SBIS)[2]. It is a very challenging task. First of all, sketches are drawn by amateurs; therefore, they generally deform in shape. Besides, since people have various drawing styles, sketches could appear differently even they are corresponding to the same object. Second, sketches lack visual cues as they do not contain colors and most texture information. So translating a sketch to photo involves changes in two aspects, shape and color. However, existing image-to-image translation methods mostly focus on one of them. Works like (Isola et al. 2017; Zhu et al. 2017) have shown impressive performance in style transfer or object transfiguration among *photo* images. Works like (Isola et al. 2017; Zhu et al. 2017; Huang et al. 2018) can derive high-quality photo images from edge maps. Nevertheless, edge maps do not have the shape deformation problem as they are extracted from photos. The most relevant work is (Chen and Hays 2018). It focuses on synthesizing natural photos of multiple classes based on sketches. Unfortunately, the quality of synthesized photos is far from satisfactory, and the method relies on paired sketch/photo images and class labels.

In this work, we propose a sketch-to-photo translation model which, for the first time, can synthesize photos according to sketches without paired data. The key idea of our approach lies in disentangling shape and color trans-

*represents equal contribution. Qian Yu is the corresponding author.

[1]In the following paper, we use 'sketch' for short.

[2]In practice, the synthesized image can be with various visual formats, e.g., photo, cartoon, and so on. In this work, we focus on synthesizing photos.

lation, allowing the model to handle the task step by step. Specifically, given a sketch, the model first translates it to a grayscale photo, resolving shape distortion problem; then the generated grayscale photo is enriched with other visual information like colors. This idea is verified to be surprisingly effective. In addition, during shape translation, we notice that specific drawing styles have a significant impact on generated results, sometimes even leading to failed translations. For this problem, we adopt a data augmentation strategy and integrate an attention module into our model. To be specific, different from previous works which use attention module to emphasize particular regions in the original image, our proposed model is guided by the learned attention mask to *ignore* (or pay *less* attention to) distracted regions. Furthermore, we apply a conditional module in color translation step, aiming to give users more control over final synthesized photos.

We focus on single-category generation and choose *shoe* class for our task. *Shoes* is a representative fashion class; therefore, it has been used in many computer vision tasks, and there exists many shoes datasets (Yu et al. 2016; Parikh and Grauman 2011). Note that our model does not rely on sketch/photo pairs during training. Thus any sketch datasets containing shoe class can be used for our task.

In summary, our contribution is four-fold: 1. to our best knowledge, it is the first unsupervised sketch-to-photo translation model which can derive a photo from a human free-hand sketch; 2. we propose a simple but effective approach to tackle this task, where the key idea is decomposing the SBIS task into a sequential translations of shape and color; 3. We also introduce two data augmentation strategies and re-purpose an attention module to handle drawing style problem; 4. as a side benefit, the proposed model can also synthesize realistic sketches. Extensive experiments show the superiority of our approach against other baselines; we also demonstrate how the generated photos and sketches can benefit other sketch-related applications.

2 Related Works

Sketch-based Image Synthesis In recent years, sketches have attracted increasing attention from the computer vision community. Thanks to machine learning, especially deep learning, people have achieved compelling progress in various sketch-related tasks, such as sketch recognition (Eitz, Hays, and Alexa 2012; Yu et al. 2015; Yu et al. 2017) and sketch-based image retrieval (Eitz et al. 2011; Hu, Barnard, and Collomosse 2010; Li et al. 2014; Yu et al. 2016; Sangkloy et al. 2016; Liu et al. 2017). However, sketch-based image synthesis is still underexplored. Before the popularization of deep learning, Sketch2Photo (Chen et al. 2009) and PhotoSketcher (Eitz et al. 2011) synthesize a photo image by composing objects of photos which are retrieved based on a given sketch. In recent image editing works (Bau et al. 2019; Portenier et al. 2018; Sangkloy et al. 2017; Yu et al. 2018), sketch is used to edit on a photo image. The most relevant work is sketchyGAN, which is the first deep-learning-based image synthesis work based on the free-hand sketch. It uses an encoder-decoder structure. During training, it requires paired sketch and photo images.

Beyond synthesizing a photo image based on a sketch, people have also investigated the inverse task of generating a sketch from a photo, like (Pang et al. 2018; Song et al. 2018). In this work, our proposed model shows the capability of translating between sketches and photos in both directions.

Generative Adversarial Networks (GANs) The GAN model (Goodfellow et al. 2014) has achieved impressive performance on various image generation(Mirza and Osindero 2014; Karras et al. 2017) and translation(Isola et al. 2017; Huang et al. 2018) tasks. The key of this model lies in the adversarial training between generator and discriminator. Each GAN model has a generator and a discriminator; the goal of the generator is to fool the discriminator by generating data indistinguishable from the real data, while the discriminator is trained to distinguish between real and fake data. In this work, we employ a GAN model to map an image from sketch domain to photo domain.

Image-to-Image Translation With the popularization of GAN models, image-to-image translation task has been widely explored in recent years. In Pix2Pix (Isola et al. 2017), a conditional GAN model is adopted to learn a mapping function from the source domain to the target. As a limitation, it requires paired data during training. To overcome this shortcoming, the cycleGAN model proposes a cycle consistency loss, allowing the model to get rid of the dependency on paired data. Based on the idea of cycle consistency, several models are proposed for the task of unpaired image-to-image translation, such as UNIT(Huang et al. 2018) and MUNIT(Huang et al. 2018). While these methods achieve impressive results, they are limited to the scenarios where the source and target data are well-aligned. In our case, a sketch image and a photo image are misaligned in shape and are also different in color distribution. We take the models mentioned above (i.e., Pix2Pix, cycleGAN, UNIT, and MUNIT) as baselines and show their performance in our task.

Besides, we also include a recent model UGATIT (Kim et al. 2019) as a baseline since it employs an attention module to emphasize domain-discriminative regions and intensively translate such regions. Similarly, the proposed model also has an attention module, but it serves the opposite goal. We will detail this in Section 3.2.

3 Our Approach

Our goal is to learn mapping functions between two domains S and P_R, i.e., sketches and RGB photos, given training samples $\{s_i\}_{i=1}^{N}$ where $s_i \in S$ and $\{p_j\}_{j=1}^{M}$ where $p_j \in P_R$, assuming no existence of paired images. Mapping a sketch to a photo image involves changes in two aspects: shape and color; therefore, the proposed model consists of two sub-networks, each tackling the translation in one aspect. We explain the architecture of the model in Section 3.1. In Section 3.2, we discuss the strategies proposed to handle a problem encountered in shape translation, namely that the basic shape translation network may fail when an input sketch exhibits specific drawing styles. Thus, we improve the model by integrating an attention module and introducing two data augmentation strategies.

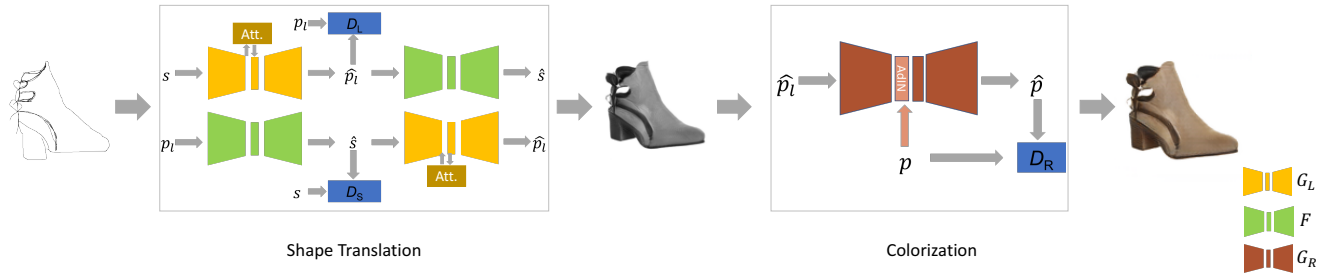

Figure 1: Our two step based sketch-to-photo translation model architecture.

3.1 Disentangle Shape and Color Translation

3.1.1 Shape Translation

Shape translation is to translate a sketch image to a grayscale image whose shape is faithful to a real object. The key to this step is removing the factor of color and forcing the network to focus on shape. Given a sketch image s and a photo image p, we first convert p from RGB color space to Lab space and obtain its grayscale version p_l, and then learn mapping functions between s and p_l. Our shape translation network is developed based on CycleGAN (Zhu et al. 2017). It learns two mapping functions, $G_L : S \rightarrow P_L$ and $F : P_L \rightarrow S$, and two domain discriminators D_L and D_S. The discriminator D_L aims to distinguish between p_l and $G_L(s)$, while D_S aims to distinguish between s and $F(p_l)$. The output of this step, $G_L(s)$, will be fed into the color translation network for further processing.

3.1.2 Colorization

Next, the proposed colorization network will map the generated grayscale image $G_L(s)$ to RGB photo domain. We first introduce a basic version to synthesize photos as real as possible and then explain an improved version which targets for generating images with more diversity.

Basic Network The basic colorization network adopts an encoder-decoder structure. We modify the network by adding an adversarial loss. A mapping function G_R will be learned in this step, and $f(.)$ and $g(.)$ is the encoder and decoder of G_R. Specifically, given an input image $p'_l = G_L(s)$, the model needs to output an image satisfying the following conditions: it should be (1) indistinguishable with a real photo p, and (2) as similar as possible to the input image in the Lab color space. A discriminator D_R is learned to distinguish between a fake image $G_R(p'_l)$ and a real image p.

Improved Network For a specific grayscale image, there are many colorization options. To make use of this flexibility, we treat the colorization task as a style transfer problem. It has been verified in (Huang and Belongie 2017) that the style of an image can be modified by changing the channel-wise statistics of its feature maps. Therefore, we keep the same structure of our basic model and modify it to accept a reference image based on which the output is generated. In this improved version, the generated grayscale image p'_l serves as a content image, and the reference image $p \in P_R$ as a style image. Following the Eq. 1, the mean and variance of the content image p'_l will be adjusted to match those of p.

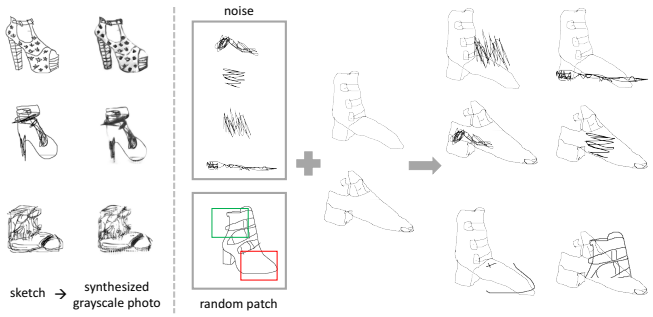

Figure 2: Left: failed examples. The first column shows the input sketches, the second column lists the generated grayscale photos. Right: examples of noise stroke masks and random patches, and input sketches with data augmentation.

$$AdaIN(f(p'_l), f(p)) = \sigma(f(p))(\frac{f(p'_l) - \mu(f(p'_l))}{\sigma(f(p'_l))}) + \mu(f(p)) \tag{1}$$

3.1.3 Model Training

During training, the shape translation network and colorization network are trained one by one. Similar to (Zhu et al. 2017), our full training objective of our shape translation network is,

$$\min_{G_L,F} \max_{D_L,D_S} \lambda_1 L_{GAN} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} \tag{2}$$

where the three terms $L_{GAN}, L_{cycle}, L_{identity}$ are adversarial loss, reconstruction loss, and identity loss as follows (here we list the loss for $S \rightarrow P_L$ only),

$$L_{GAN}^{S \rightarrow P_L} = \mathbb{E}_{p_l \sim P_L}\left[(D_L(p_l))^2\right] + \mathbb{E}_{s \sim S}\left[(1 - D_L(G_L(s)))^2\right] \tag{3}$$

$$L_{cycle}^{S \rightarrow P_L} = \mathbb{E}_{s \sim S}\left[|s - F(G_L(s))|_1\right] \tag{4}$$

$$L_{identity}^{p_l \rightarrow P_L} = \mathbb{E}_{p_l \sim P_L}\left[|p_l - G_L(p_l)|_1\right] \tag{5}$$

When training the colorization network, apart from the adversarial loss and reconstruction loss, a style loss L_s is used

to facilitate training and improve generation quality. For the improved network, a content loss L_c is added. Therefore, the loss functions of step 2 for the basic and the improved network are

$$\min_{G_R} \max_{D_R} \lambda_4 L_{GAN} + \lambda_5 L_{rec} + \lambda_6 L_s \tag{6}$$

$$\min_{G_R} \max_{D_R} \lambda_4 L_{GAN} + \lambda_5 L_{rec} + \lambda_6 (L_s + L_c) \tag{7}$$

$$L_{GAN}^{P_L \to P_R} = \mathbb{E}_{p \sim P_R} \left[(D_R(p))^2 \right] + \mathbb{E}_{p'_l \sim P_L} \left[\left(1 - D_R\left(G_R(p'_l)\right)\right)^2 \right] \tag{8}$$

The reconstruction loss[3] $L_{rec} = \|G_R(p'_l) - p'_l\|_1$ and the content loss is $L_c = \|g(t) - t\|_1$, where $t = AdaIN(f(p'_l), f(p))$ in the improved network. Style loss is shown in Eq. 9. $\phi_i(.)$ denotes a layer of a pre-trained model, e.g., VGG19 (Simonyan and Zisserman 2014). In implementation, we use $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ layers with equal weights to compute style loss. For the shape translation network, $\lambda_1 = 1.0$, $\lambda_2 = 10.0$ and $\lambda_3 = 0.5$. For the colorization network, we set λ_4 as 1.0, λ_5 as 10.0 (or 1.0), λ_6 as 100.0 (or 0.2) for the basic (or the improved) version.

$$\begin{aligned} L_s = &\sum_{i=1}^{K} \|\mu(\phi_i(g(t))) - \mu(\phi_i(p))\|_2 \\ &+ \sum_{i=1}^{K} \|\sigma(\phi_i(g(t))) - \sigma(\phi_i(p))\|_2 \end{aligned} \tag{9}$$

3.2 Deal with Drawing Style Problem

Due to their free-hand nature, sketches could show different drawing styles. Although the shape translation network proposed in Section 3.1 can successfully translate sketch images in most cases, we noticed that it might fail to translate sketches with specific drawing styles. As shown in Fig.2, the network directly 'copy' the input as an output or modify it slightly. We can see these failure cases share a common characteristic that they contain dense and irregular strokes (we name such sketches as *complex* sketch). We presume that the network confuses a *complex* sketch with a grayscale image due to the dense strokes, thus doing little work on the input.

Motivated by our observation, we tackle this problem by introducing two data augmentation strategies and incorporating an attention module. We will explain the details next and demonstrate their effectiveness in Section 4.2.2.

Data Augmentation From the data point of view, increasing the diversity of training data allows the model to 'see' more possibilities. As a result, the model can be more robust when dealing with drawing style variations. Therefore, we synthesize *complex* sketches by randomly applying noise strokes on original sketches (as indicated in Fig.2). The noise stroke masks are obtained from training samples[4]. We formed a noise set consisting of 42 stroke noise masks. They will be randomly sampled and applied to the input sketch during training.

The second data augmentation strategy shares a similar idea with the first one but generalizes to broader cases. Mainly, when feeding a sketch image into the network, a random patch is extracted from another sketch and then applied on it to form a new one. Note that the reconstruction loss is computed between the reconstructed and the *original* sketch. Therefore, our model is trained to ignore the distracting noise and extract useful information from a composed sketch.

Re-purposed Attention Module As explained before, complex sketches with dense strokes may confuse the network and lead to a failed translation. Hence, activation of such regions should be suppressed. We introduce an attention module and use it as a detector to locate the dense stroke region(s). During training, the attention module will generate an attention mask A, then this mask is used to re-weight the feature map. Different from existing works, we compute the final feature map as $f_{final}(s) = (1 - A) * f(s)$. In our implementation, the attention module consists of two $1 * 1$ convolutional layers and is inserted after the final down-sampling layer of the encoder.

[3]The synthesized photo $G_R(p'_l)$ needs to be converted to Lab color space first for computing L_{rec}.

[4]The original dataset provides data in SVG format, so we can manually extract particular stroke(s).

Table 1: Qualitative comparison on ShoeV2 dataset. '$*$' represents using paired data during training.

Model	FID	Quality	Diversity
Pix2Pix*	65.09	40.0	0.071
CycleGAN	74.60	27.15	0
UNIT	117.34	26.67	0
MUNIT	98.67	20.94	**0.246**
UGATIT	76.37	34.30	0.116
Ours (w/o ref.)	**50.56**	**50.0**	0
Ours (with ref.)	-	-	0.180

4 Experiments

4.1 Implementation

Dataset We use the ShoeV2 dataset (Yu et al. 2016) for training and evaluation. It contains 6,648 sketches and 2,000 photos, and each photo has three or more corresponding sketches drawn by different individuals. Note that although paired data are available, we do not use them during training. Compared with other existing sketch datasets, like QuickDraw (Ha and Eck 2017), Sketchy (Sangkloy et al. 2016), and TU-Berlin (Eitz, Hays, and Alexa 2012), this dataset not only contains both sketch and photo images, but its sketches include sufficient fine-grained details. That means the synthesized photos should reflect these details, which is a more challenging setting than synthesizing photos from coarser sketches. We use 5982/1800 sketch/photo images for training, while the rest are used for testing.

Baselines We choose several image-to-image translation models as baselines and provide quantitative and qualitative comparisons in Section 4.2.

- **CycleGAN** CycleGAN (Zhu et al. 2017) is a bidirectional

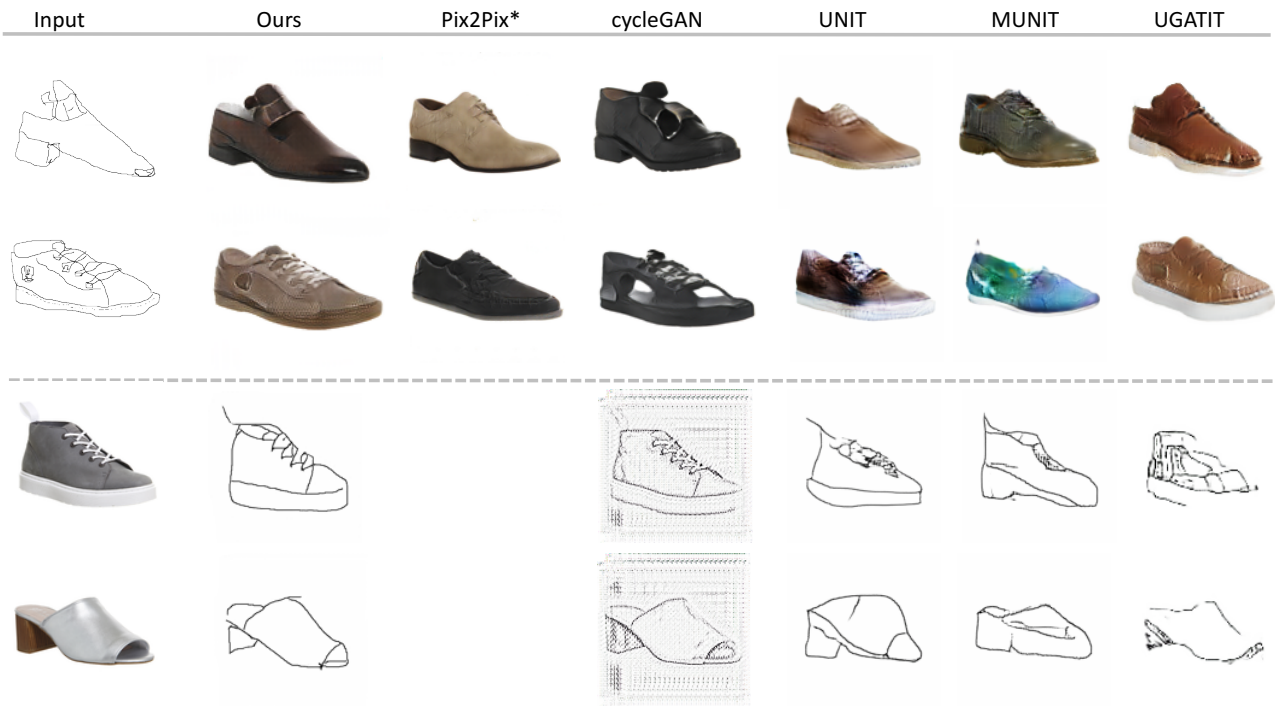

Figure 3: Comparison of the results generated by different methods. Upper: sketch to photo. Bottom: photo to sketch. From the left to right: input sketch/photo images, ours, Pix2Pix, cycleGAN, UNIT, MUNIT, UGATIT trained on ShoeV2 dataset.

Figure 4: Generated photos with different reference images as condition.

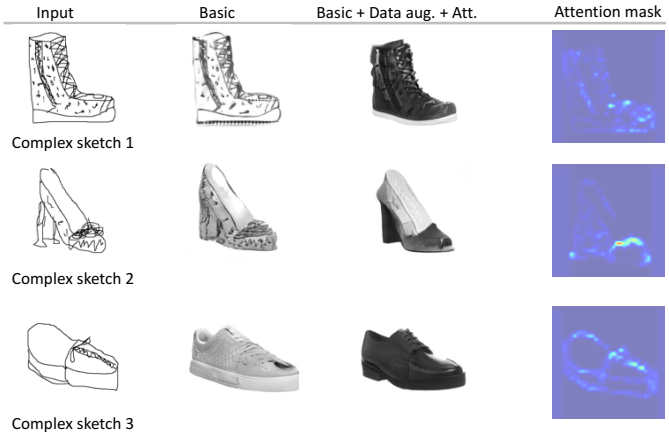

Figure 5: Effects of attention module and data augmentation strategies. Our attention module successfully highlights the dense and complex part of the input sketch.

unsupervised image-to-image translation model. In our approach, it serves the primary network for shape translation (as explained in Section 3.1). We take it as a baseline by training it to translate a sketch to an RGB photo image directly.

- **UNIT** The UNIT (Liu, Breuel, and Kautz 2017) model consists of two VAE-GANs with a shared latent space. Different from cycleGAN, it uses a multi-scale discriminator and shares the weights of high-level layers between two encoders and decoders respectively.
- **MUNIT** MUNIT (Huang et al. 2018) is an unsupervised model which can generate multiple outputs given an input image. It assumes that the image representation can be decomposed into a content code and a style code. Our approach shares a similar idea with MUNIT model. However, our disentangle strategy is proposed considering the unique characteristics of sketches, thus is more specific and suitable for sketch-to-photo translation task.
- **UGATIT** UGATIT (Kim et al. 2019) is a model incorporating an attention module and a learnable normalization function. The attention module is designed to help the model focus on the domain-discriminative regions which distinguish the source and target domain, such that the generated results quality can be improved.
- **Pix2Pix** Pix2Pix (Isola et al. 2017) is a *directional* generative model which requires paired images of two domains during training. We include this model as a baseline as we want to see how is the performance when paired data are used during training.

Training Details We train our shape translation network for 700 epochs and colorization network for 200 epochs. The initial learning rate is set to be 0.0002, and the input image size is 128*128. We use Adam optimizer with a batch size of 1. Following the practice suggested in cycleGAN, we train the first 100 epochs at the same learning rate and then linearly decrease the rate to zero until the maximum epoch. For data augmentation, we add noise stroke masks and random patches to the input sketches at the rate of 20% and 30% respectively. The random patch size is 50*50.

We train the baseline models with their default settings. During our experiments, we find that cycleGAN is very sensitive to the initialization and easy to collapse. Thus we train the model six[5] times and report its average performance.

Evaluation Metrics We use three metrics to evaluate the performance of the proposed approach and baselines: user study, FID, and LPIPS distance.

- **Human Preference** We perform a human perceptual study to evaluate the similarity and realism of results produced by different methods. As introduced in the beginning, a human can picture the real appearance of an object represented by a sketch. So we can evaluate which method can generate photos that are more consistent with the human imagination. Thus we adopt the approach introduced in (Wang et al. 2018). For each comparison, an input sketch and its corresponding generated photos from two methods (one is the proposed method, and the other is a baseline method) are shown to a user at the same time, and then the user needs to choose which one is closer to his/her expectation. We sample 200 pairs for each comparison and ask five individuals to answer each question.
- **Fréchet Inception Distance** Fréchet Inception Distance measures the distance between generated samples and real samples by their statistics of activation distributions in a pre-trained Inception-v3 pool3 layer. It could evaluate quality and the diversity simultaneously. Lower FID value indicates more similar generated and real samples.
- **Learned Perceptual Image Patch Similarity** LPIPS evaluates the distance between two images. Similar to (Huang et al. 2018) and (Zhu et al. 2017), we utilize this metric to evaluate the diversity of the outputs generated by different methods.

[5]Among the six trials, this model collapsed within 150 epochs for five times while only one time it can be trained over 150 epochs without collapse.

4.2 Results

4.2.1 Experimental Results

Quantitative Results Table 1 compares the quantitative results of our model with baseline models. We can have the following observations: (1) the proposed model achieves the best results among all methods in FID and user study. Compared with baselines, the proposed model can produce photos with higher fidelity, reflecting the fine-grained details indicated in original sketches; additionally, they are more aligned with human perception. (2) Comparing with CycleGAN, our model performs more stably and can generate results with more diversity. Also, it outperforms MUNIT by a large margin, demonstrating the superiority of our disentanglement strategy.

Qualitative Results Figure 3 compares the outputs generated by different methods. Aligned with quantitative results showed in Table 2, our proposed model significantly outperforms other baselines. Precisely, our model can generate not only realistic photos but also high-quality sketches. Apart from being realistic, the generated photos of our method can keep the fine-grained details indicated in input sketches. In contrast, outputs of the baseline models fail to keep such details.

Qualitative Results with Conditions In Fig. 4, we show examples of the generated results when reference images are available. The first and second column displays the input sketches and the synthesized grayscale images after shape translation; the first row shows four reference images. It is clear to see that our improved colorization network can translate a sketch to various RGB photos with different images as guidance. In addition, considering the styles present in the training set is limited, and there is a correlation between shoe style and colors/textures, the synthesized results may become unrealistic when the color/texture of the reference image is not typical.

4.2.2 Deal with Complex Sketch

In Section 3.2, we discuss the problem encountered during shape translation and further introduce two data augmenta-

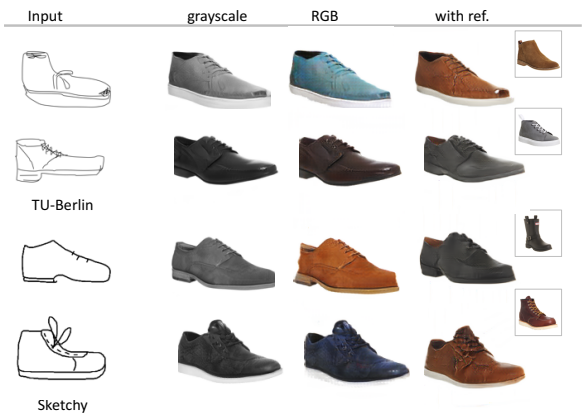

Figure 6: Generated photos based on sketches from different datasets.

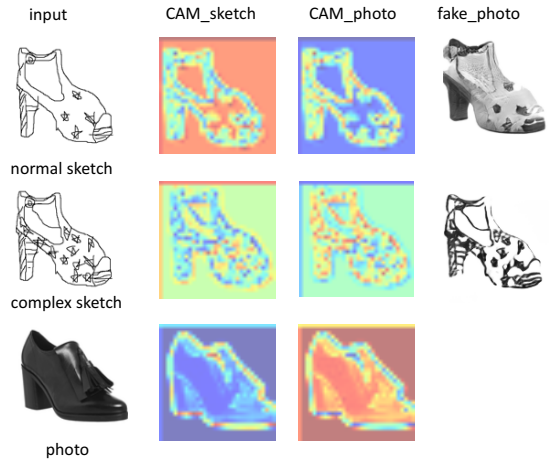

Figure 7: Examples of CAM visualizations.

tion strategies and an attention module to handle the problem. In Fig. 5, we display three examples. Each of them has compact strokes which we presume may cause translation failure of the basic shape translation network. We compare the translation results of the two variants, i.e., basic and improved version. The last column shows the attention mask learned by the newly introduced attention module. It is clear to see that after applying the data augmentation strategies and incorporating the attention module, our final shape translation network can handle complex sketches better.

It would be useful for us to understand why the basic model fails to translate *complex* sketches. Inspired from (Zhou et al. 2016), we train an auxiliary domain classifier and obtain Class Activation Maps (CAM) for analysis. To be specific, given the basic shape translation network, a binary classifier is added after the last downsampling layer. The classifier consists of two fully-connected layers, and its gradients do not back-propagate to the generator during training. We visualize domain-specific CAM of three examples: a typical sketch, a *complex* sketch, and a grayscale photo. Through comparing their CAM images in Fig. 7, we can see that the CAM_sketch and CAM_photo of a complex sketch are similar, and the background of CAM_photo of the complex sketch is more activated than that of the normal sketch. This observation verifies our assumption that it is these compact strokes that confuse the generator and lead to translation failure.

4.2.3 Test on sketches from other datasets

To measure the generalization ability of the trained model, we test it on sketches from other datasets. Compared with sketches in ShoeV2 dataset, shoe sketches in Sketchy and TU-Berlin are coarser than ShoeV2 due to different data collection pipelines. We directly test our trained model on sketches randomly sampled from these two datasets. The results are shown in Fig. 6.

4.3 Application: Sketch-based Image Retrieval

As a side benefit, the proposed model can generate realistic sketches. To the best of our knowledge, it is the first model which can handle photo and sketch generation at the same time. Because sketch-generation is not the focus of this work, we did not compare them with other existing works.

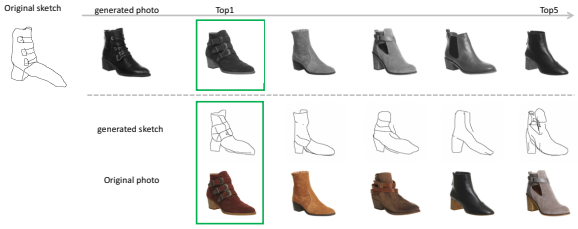

Figure 8: Examples of retrieval results using generated photo and sketch. Top: using the generated photo as a proxy of the query sketch for retrieval, i.e., fake photo → real photo; Bottom: using the generated sketches as proxy of candidate photos for retrieval, i.e., real sketch → fake sketch.

Here we demonstrate how generated photos or sketches can further benefit sketch-based image retrieval (SBIR). In SBIR, a sketch image is used as a query to search for photos which represent same or similar contents. The biggest challenge of this task lies in the large domain gap. We assume that translating the query and gallery to the same domain should help. Therefore, we do two experiments: (1)translate a sketch to a photo and then find its nearest neighbour(s) in the gallery; (2) translate gallery photos to sketches, and then find the nearest sketches for the query sketch. We use an ImageNet pre-trained ResNet18 as a feature extractor for photo-to-photo retrieval; while we further fine-tune this network on TU-Berlin dataset and use it to extract features from (synthesized) sketches.

Figure 8 shows the retrieval results. It is clear to see that even without using any supervision, the retrieved results are still acceptable. In the second experiment, we achieve the accuracy of 37.2%/65.2% at top5/top20 respectively. These results are higher than the results of *sketch to edge map*, which are 34.5%/57.7%.

Conclusion

In this work, we focus on the task of sketch-to-photo image translation. For the first time, an unsupervised model is proposed for this task, which can generate photos with high-fidelity and diversity. The key idea of our proposed method is disentangling the task into shape and color translation. This is motivated by the fact that sketches are generally sparse in visual cues and often exhibit deformation. In the future, we will further investigate the reason(s) why the generative model fails to translate some sketches. As the first step, we introduced an attention module which processes all sketches. However, it is expected to distinguish *complex* sketches with normal ones, and use different strategies to process them. Beyond the single-class setting, we will explore the multi-class setting in our future work.

References

[Bau et al. 2019] Bau, D.; Strobelt, H.; Peebles, W.; Wulff, J.; Zhou, B.; Zhu, J.-Y.; and Torralba, A. 2019. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)* 38(4):59.

[Chen and Hays 2018] Chen, W., and Hays, J. 2018. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9416–9425.

[Chen et al. 2009] Chen, T.; Cheng, M.-M.; Tan, P.; Shamir, A.; and Hu, S.-M. 2009. Sketch2photo: internet image montage. In *ACM Transactions on Graphics (TOG)*.

[Eitz et al. 2011] Eitz, M.; Richter, R.; Hildebrand, K.; Boubekeur, T.; and Alexa, M. 2011. Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications*.

[Eitz, Hays, and Alexa 2012] Eitz, M.; Hays, J.; and Alexa, M. 2012. How do humans sketch objects? In *ACM Transactions on Graphics (TOG)*.

[Eitz et al. 2011] Eitz, M.; Hildebrand, K.; Boubekeur, T.; and Alexa, M. 2011. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG* 17(11):1624–1636.

[Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

[Ha and Eck 2017] Ha, D., and Eck, D. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

[Hu, Barnard, and Collomosse 2010] Hu, R.; Barnard, M.; and Collomosse, J. 2010. Gradient field descriptor for sketch based retrieval and localization. In *ICIP*.

[Huang and Belongie 2017] Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 1501–1510.

[Huang et al. 2018] Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 172–189.

[Isola et al. 2017] Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

[Karras et al. 2017] Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[Kim et al. 2019] Kim, J.; Kim, M.; Kang, H.; and Lee, K. 2019. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *CoRR* abs/1907.10830.

[Li et al. 2014] Li, Y.; Hospedales, T.; Song, Y.-Z.; and Gong, S. 2014. Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*.

[Liu et al. 2017] Liu, L.; Shen, F.; Shen, Y.; Liu, X.; and Shao, L. 2017. Deep sketch hashing: Fast free-hand sketch-based image retrieval. *arXiv preprint arXiv:1703.05605*.

[Liu, Breuel, and Kautz 2017] Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, 700–708.

[Mirza and Osindero 2014] Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[Pang et al. 2018] Pang, K.; Li, D.; Song, J.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2018. Deep factorised inverse-sketching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 36–52.

[Parikh and Grauman 2011] Parikh, D., and Grauman, K. 2011. Relative attributes. In *2011 International Conference on Computer Vision*, 503–510. IEEE.

[Portenier et al. 2018] Portenier, T.; Hu, Q.; Szabo, A.; Bigdeli, S. A.; Favaro, P.; and Zwicker, M. 2018. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics (TOG)* 37(4):99.

[Sangkloy et al. 2016] Sangkloy, P.; Burnell, N.; Ham, C.; and Hays, J. 2016. The sketchy database: Learning to retrieve badly drawn bunnies. In *SIGGRAPH*.

[Sangkloy et al. 2017] Sangkloy, P.; Lu, J.; Fang, C.; Yu, F.; and Hays, J. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5400–5409.

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Song et al. 2018] Song, J.; Pang, K.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2018. Learning to sketch with shortcut cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 801–810.

[Wang et al. 2018] Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. High-resolution image synthesis and semantic manipulation with conditional gans.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8798–8807.

[Yu et al. 2015] Yu, Q.; Yang, Y.; Song, Y.; Xiang, T.; and Hospedales, T. 2015. Sketch-a-net that beats humans. In *BMVC*.

[Yu et al. 2016] Yu, Q.; Liu, F.; Song, Y.-Z.; Xiang, T.; Hospedales, T. M.; and Loy, C.-C. 2016. Sketch me that shoe. In *CVPR*.

[Yu et al. 2017] Yu, Q.; Yang, Y.; Liu, F.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2017. Sketch-a-net: A deep neural network that beats humans. *IJCV* 122(3):411–425.

[Yu et al. 2018] Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2018. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*.

[Zhou et al. 2016] Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

[Zhu et al. 2017] Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.

Appendix

In this appendix, more qualitative results are provided. We first show generated photo images, including grayscale photos, RGB photos, and RGB photos guided by reference images (Fig. 9 and Fig. 10). Then examples of generated sketches (Fig. 11) are illustrated. Finally, given each sketch can be treated as a list of strokes, we manipulated a sketch in SVG format to produce some intermediate sketches (accumulative sketches), and then generate corresponding photos (see Fig. 12).

Figure 9: Generated photo images using the proposed model. From left to right: input sketch, generated grayscale photo (shape translation), generated RGB photo (basic colorization), generated RGB photo condition on reference photo (improved colorization), two nearest neighbors of the generated RGB photo. To retrieve the nearest neighbours, we employed an ImageNet pre-trained ResNet18 to extract the deep feature of each image, and computed L2 distance for comparison.

Figure 10: More generated results. From left to right: input sketch, generated grayscale photo (shape translation), generated RGB photo (basic colorization), generated RGB photo condition on reference photo (improved colorization), two nearest neighbors of the generated RGB photo.

Figure 11: Generated sketch images using the proposed model.

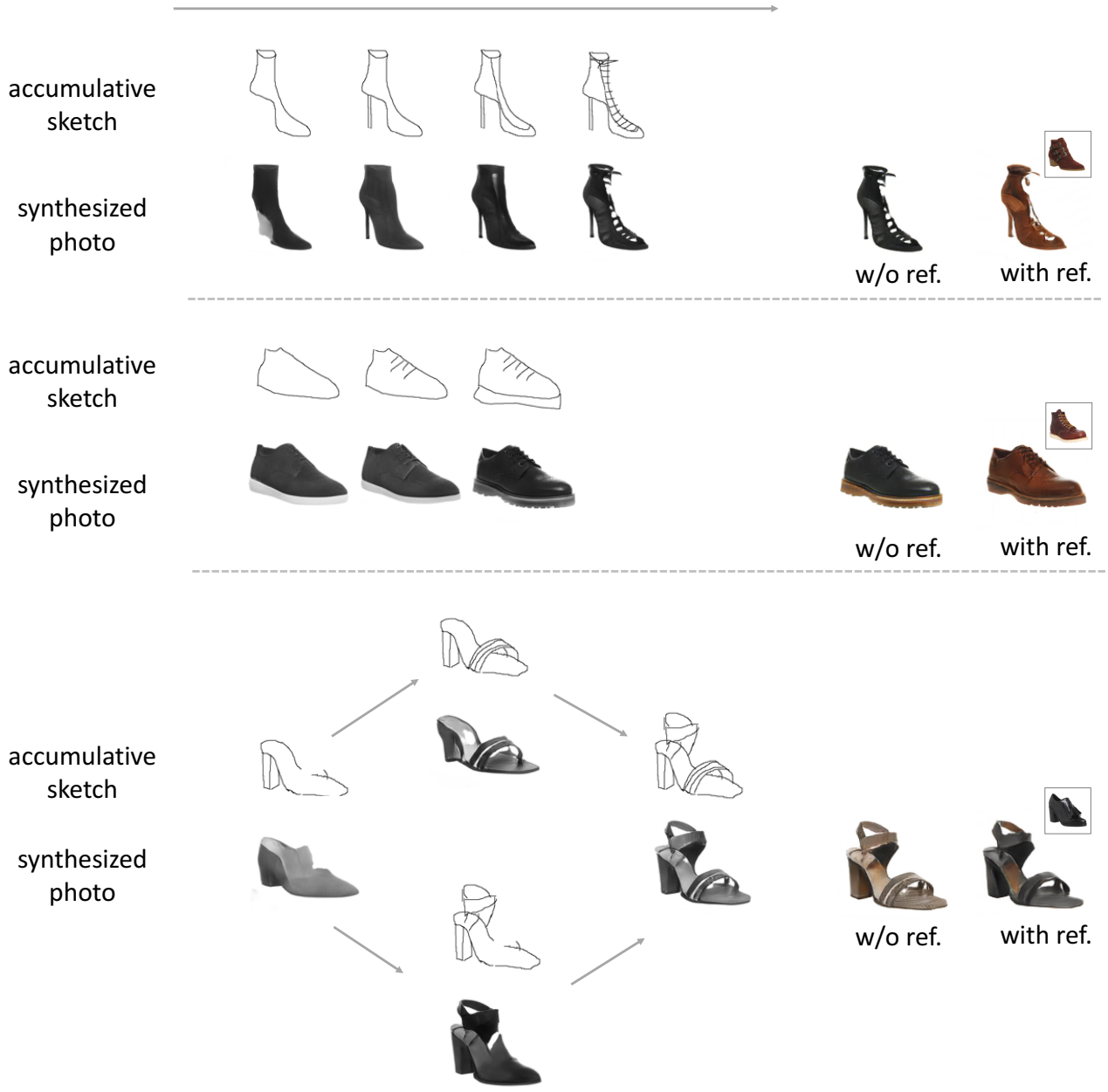

Figure 12: Generated grayscale photo images corresponding to accumulative sketches. The last two images of each example are the final outputs. Accumulative sketches are obtained by manipulating images in SVG format.