

# Learning from Multi-domain Artistic Images for Arbitrary Style Transfer

Zheng Xu<sup>1†</sup>, Michael Wilber<sup>2</sup>, Chen Fang<sup>3</sup>, Aaron Hertzmann<sup>3</sup>, and Hailin Jin<sup>3</sup>

<sup>1</sup>University of Maryland, College Park <sup>2</sup>Cornell Tech, New York <sup>3</sup>Adobe Research, San Jose

## Abstract

We propose a fast feed-forward network for arbitrary style transfer, which can generate stylized image for previously unseen content and style image pairs. Besides the traditional content and style representation based on deep features and statistics for textures, we use adversarial networks to regularize the generation of stylized images. Our adversarial network learns the intrinsic property of image styles from large-scale multi-domain artistic images. The adversarial training is challenging because both the input and output of our generator are diverse multi-domain images. We use a conditional generator that stylized content by shifting the statistics of deep features, and a conditional discriminator based on the coarse category of styles. Moreover, we propose a mask module to spatially decide the stylization level and stabilize adversarial training by avoiding mode collapse. As a side effect, our trained discriminator can be applied to rank and select representative stylized images. We qualitatively and quantitatively evaluate the proposed method, and compare with recent style transfer methods. We release our code and model at [https://github.com/nightldj/behance\\_release](https://github.com/nightldj/behance_release).

## CCS Concepts

• **Computing methodologies** → Image manipulation; Non-photorealistic rendering;

## 1. Introduction

Image style transfer is a task that aims to render the content of one image with the style of another, which is important and interesting for both practical and scientific reasons. The style transfer techniques can be widely used in image processing applications such as mobile camera filters and artistic image generation. Furthermore, the study of style transfer often reveals the intrinsic property of images. Style transfer is challenging as it is difficult to explicitly separate and represent the content and style of an image.

In the seminal work of [GEB16], the authors represent content with deep features extracted by a pre-trained neural network, and represent style with second order statistics (i.e. the Gram matrix) of the deep features. They propose an optimization framework with the objective that the generated image has similar deep features to the given content image, and similar second order statistics to the given style image. The generated results are visually impressive, but the optimization framework is far too slow for real-time applications. Later works [JAF16; UVL17] train a feed-forward network to replace the optimization framework for fast stylization, with a loss similar to [GEB16]. However, they need to train a network for each style image and cannot generalize to unseen images. More recent approaches [HB17; LFY\*17] tackle arbitrary style transfer for

unseen content and style images, which still represent style with second order statistics of deep features. The second order statistics of style representation is originally designed for textures [GEB15], and style transfer is considered as texture transfer in previous methods.

Another line of research considers style transfer as conditional image generation, and apply adversarial networks to train an image to image translation network [IZZE17; TPW17; ZPIE17; HLBK18]. The trained image translation networks can transfer image from one domain to another domain, for example, from a natural image to sketch. However, they cannot be applied to arbitrary style transfer as the input images are from multiple domains.

In this paper, we combine the best of both worlds by adversarially training a single feed-forward network for arbitrary style transfer. We introduce several techniques to tackle the challenging problem of adversarial training from multi-domain data. In adversarial training, the generator (stylization network) and the discriminator are alternatively updated. Both our generator and discriminator are conditional networks. The generator is trained to fool the discriminator, as well as satisfy the content and style representation similarity to inputs. Our generator is built upon a state-of-the-art network for arbitrary style transfer [HB17], which is conditioned on both content image and style image, and uses adaptive instance normalization (AdaIN) to combine the two inputs. AdaIN shifts the mean and variance of the deep features of content image to match those

† contact: xuzh@cs.umd.edu

of the style image. Our discriminator is conditioned on the coarse domain categories, which is trained to distinguish the generated images with real images from the the same style category.

Comparing with previous arbitrary style transfer methods, our approach uses the discriminator to learn a data-driven representation for styles. The combined loss for our generator considers both instance-level information from style loss and category-level information from adversarial training. Comparing with previous adversarial training methods, our approach handles multi-domain inputs by using a conditional generator designed for arbitrary style transfer and a conditional discriminator. Moreover, we propose a mask module to automatically control the level of stylization by predicting a mask to blend the stylized features and the content features. Finally, we use the trained discriminator to rank and find the representative generated images in each style category. We release our code and model at [https://github.com/nightldj/behance\\_release](https://github.com/nightldj/behance_release)

## 2. Related work

**Style transfer.** We briefly review the neural style transfer methods, and recommend [JYF\*17] for a more comprehensive review. [GEB16] proposed the first neural style transfer method based on an optimization framework, which uses deep features to represent content and Gram matrix to represent style. The optimization framework was replaced by a feed forward network to achieve real-time performance in [JAF16; ULVL16; WOZW17]. [UVL17] showed that instance normalization is particularly effective for training a fast style transfer network. Other works focused on controlling spatial, color, and stroke for stylization [GEB\*17; FSDH16; JLY\*18], and exploring other style representation such as mean and variance [LWLH17], histogram [WRB17], patch-based MRF [LW16a], and patch-based GAN [LW16b]. Comparing with [GEB16], these fast style transfer methods sometimes compromise on the visual quality, and need to train one network for each style.

Various methods have been proposed to train a single feed forward network for multiple styles. [DSK17] proposed conditional instance normalization, which learned the affine parameter for each style image. [CYL\*17] learned the “style bank”, which contains several layers of filters for each style. [ZD17] proposed comatch layers for multi-style transfer. These methods only work with limited number of styles, and cannot apply to an unseen style image.

More recent approaches are designed for arbitrary style transfer, where both the content and the style inputs can be unseen images. [GLK\*17] extended conditional instance normalization (IN) by training a separate network to predict the affine parameter of IN. [FZ18] learned a meta network to predict filters in the transformation networks. [HB17] proposed adaptive instance normalization (AdaIN) that adjusts the mean and variance of content image to match those of the style image. [LFY\*17; LLL\*18] used feature whitening and coloring transforms (WCT) to match the statistics of the content image to the style image. [SLSW18] proposed feature decoration that generalizes AdaIN and WCT. Note that the optimization framework [GEB16] and path-based non-parametric methods (e.g., style swamp [CS16], deep image analogy [LYY\*17], and deep feature reshuffle [GCLY18]) can also be applied to arbitrary style transfer, but these methods can be much slower. [ZCZ18]

proposed to separate style and content and then combine them with bilinear layer, which requires a set of content and style images as input and has limited applications. Our approach is the first to explore adversarial training for arbitrary style transfer.

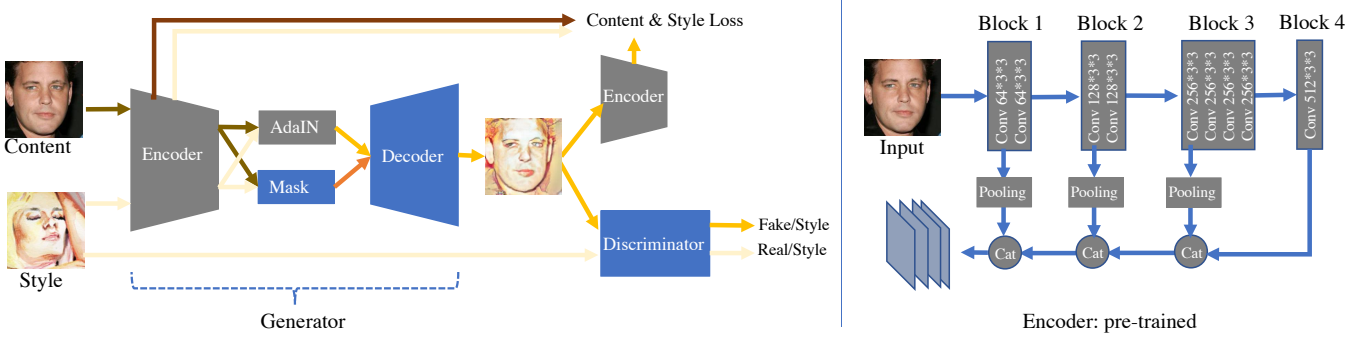
**Generative adversarial networks (GANs).** GANs have been widely studied for image generation and manipulation tasks since [GPM\*14]. [ELEM17] applied GANs to generate artistic images. [IZZE17] used conditional adversarial networks to learn the loss for image to image translation, which is extended by several concurrent methods [ZPIE17; KCK\*17; YZTG17; LBK17] that explored cycle-consistent loss when training data are unpaired. Later works improved the diversity of generated images by considering multimodality of data [ZZP\*17; ARS\*18; HLBK18]. Similar techniques have been applied to specific image to image translation tasks such as image dehazing [YXL18], face to cartoon [TPW17; RBG\*17] and font style transfer [AFK\*18]. These methods successfully train a transformation network from one image domain to another. However, they cannot handle multi-domain input and output images, and it is known to be difficult to generate images with large variance [CDH\*16; OOS17; MK18]. Our approach adopt conditional generator and discriminator to tackle the multi-domain input and output for arbitrary style transfer.

## 3. Proposed method

We use an encoder-decoder architecture as our transformation network, and use the convolutional layers of the pre-trained VGG net [SZ15; XYLS18] as our encoder to extract the deep features. We add skip connections and concatenate the features from different levels of convolutional layers as the output feature of the encoder. We adopt adaptive instance normalization (AdaIN) [HB17] to adjust the first and second order statistics of the deep features. Furthermore, we generate spatial masks to automatically adjust the stylization level. Our transformation network is a conditional generator inspired by the state-of-the-art network for arbitrary style transfer. Our network is trained with perceptual loss for content representation, Gram loss for style representation as in [GEB16; JAF16; ULVL16], as well as the adversarial loss to capture the common style information beyond textures from a style category. We show the proposed network in figure 1, and provide details in the following sections.

### 3.1. Network architecture

Our **encoder** uses the convolutional layers of the VGG net [SZ15] pre-trained on Imagenet large-scale image classification task [RDS\*15]. VGG net contains five blocks of convolutional layers, and we adopt the first three blocks and the first convolutional layer of the forth block. Each block contains convolutional layers with ReLU activation [KSH12], and the width (number of channels) and size (height and width) of the convolutional layers are shown in figure 1. There is a maxpooling layer of stride two between blocks, and the width of convolutional layer is doubled after the downsampling by maxpooling. We concatenate the features from the first convolutional layer of each block as the output of the encoder. These skip connections help to transfer style captured by both high-level and low-level features, as well as make the training easier by smoothing the loss surface of neural networks [LXT\*18].



**Figure 1:** Proposed network: (left) encoder-decoder as generator; (right) pre-trained VGG as encoder. The decoder architecture is symmetric comparing to encoder. We use the conventional texture loss based on pre-trained encoder features, and adversarially train mask module, decoder and discriminator.

Our **decoder** is designed to be almost symmetric to the VGG encoder, which has four blocks and between blocks are transposed convolutional layer for upsampling. We add LeakyReLU [HZRS15] and batch normalization [IS15] to each convolutional layer for effective adversarial training [RMC16]. The decoder is trained from scratch.

**Adaptive instance normalization (AdaIN)** has been shown to be effective for image style transfer [HB17]. AdaIN shifts the mean and variance of deep features of content to match style with no learnable parameters. Let  $x, y \in \mathbb{R}^{N \times C \times H \times W}$  represent the features of a convolutional layer from a minibatch of content and style images, where  $N$  is the batch size,  $C$  is the width of the layer (number of channels),  $H$  and  $W$  are height and width, respectively.  $x_{nchw}$  denotes the element at height  $h$ , width  $w$  of the  $c$ th channel from the  $n$ th sample, and adaIN layer can be written as,

$$A_{nchw}(x, y) = \sigma_{nc}(y) \left( \frac{x_{nchw} - \mu_{nc}(x)}{\sigma_{nc}(x)} \right) + \mu_{nc}(y) \quad (1)$$

where  $\mu_{nc}(x) = 1/HW \sum_{h,w=1}^{H,W} x_{nchw}$ ,  $\sigma_{nc}(x) = \sqrt{1/HW \sum_{h,w=1}^{H,W} (x_{nchw} - \mu_{nc}(x))^2 + \epsilon}$ ,  $\epsilon$  is a very small constant, and  $\mu_{nc}(x), \sigma_{nc}^2(x)$  represent the mean and variance for the  $c$ th channel of the  $n$ th sample of feature  $x$ .

The **mask** module in our network contains a few convolutional layers operated on the concatenation of content feature  $x$  and style feature  $y$ . The output is a spatial soft mask  $M(x, y) \in [-1, 1]^{N \times C \times H \times W}$  that has the same size as feature and each value is between  $-1$  and  $1$ . The generated mask  $M(x, y)$  is used to control the stylization level by linearly combine the adaIN feature  $A(x, y)$  and the original content feature  $s$  as the input of the decoder,

$$z = M(x, y) \times x + (1 - M(x, y)) \times A(x, y), \quad (2)$$

where the element-wise operations are used for combining these features.

Our **discriminator** is a patch-based network inspired by [IZZE17]. To handle the multi-domain images for arbitrary style transfer, our discriminator is conditioned on the style category labels. Inspired by AC-GAN [OOS17], our discriminator predicts the style category and distinguish the real image and fake image at the

same time. We also adopt the projection discriminator [MK18] to make sure the style category conditioning will not be ignored.

### 3.2. Adversarial training

We alternatively update the generator (mask module and decoder) and discriminator during training, and apply prediction optimizer [YSX\*18] to stabilize the training.

**Generator update.** Our generator takes a content image and a style image as input, and outputs the stylized image. The generator is updated by minimizing the loss combined of adversarial loss  $\mathcal{L}_A$ , style classification loss  $\mathcal{L}_{DS}$ , content loss  $\mathcal{L}_c$  and style loss  $\mathcal{L}_s$ ,

$$\min_G \mathcal{L}_G = \mathcal{L}_A + \lambda_{DS} \mathcal{L}_{DS} + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s, \quad (3)$$

where  $\lambda_{DS}, \lambda_c, \lambda_s$  are hyperparameters for the weights of different losses. Let us denote the feature map of the  $l$ th layer in our encoder as  $x^{(l)}, y^{(l)}$ , the input content and style images as  $x^{(0)}, y^{(0)}$ , the generator network as  $G(\cdot, \cdot)$ , and the discriminator network as  $D(\cdot)$ . When the discriminator  $D(\cdot)$  is fixed, the output stylized images  $\hat{x} = G(x^{(0)}, y^{(0)})$  aim to fool the discriminator, and also be classified to same style category  $s$  as the input style image,

$$\begin{aligned} \mathcal{L}_A &= \mathbb{E}[\log \text{Prob}(\text{Real} | D(\hat{x}))], \\ \mathcal{L}_{DS} &= \mathbb{E}[\log \text{Prob}(s | D(\hat{x}))]. \end{aligned} \quad (4)$$

$\mathcal{L}_A$  and  $\mathcal{L}_{DS}$  are learned loss that capture the category-level style of images from the training data. We also use the traditional content and style loss based on deep features and Gram matrix,

$$\begin{aligned} \mathcal{L}_c &= \mathbb{E}[\|x^{(4)} - \hat{x}^{(4)}\|_1], \\ \mathcal{L}_s &= \mathbb{E}[\sum_{l=1}^4 \|\text{Gram}(y^{(l)}) - \text{Gram}(\hat{x}^{(l)})\|_1]. \end{aligned} \quad (5)$$

We use the deep feature from the forth block of pre-trained VGG net for content representation, and use the Gram matrix from all the blocks for style representation. We find  $\ell_1$  norm is more stable than  $\ell_2$  when combining with the adversarial loss.

**Discriminator update.** Our discriminator is conditioned on style category to handle the multi-domain generated images, inspired by [CDH\*16; OOS17; MK18; XHH18]. When the generator is fixed,



**Figure 2:** Benefits of adversarial training and mask module. We show the encoder-decoder network with adversarial training only, mask module only, and the combination of adversarial training and mask module. Mask module only does not improve the visual quality of generated images, which have artifacts and undesired textures. GAN only can generate collapsed images with corrupted eyes and noses.

the discriminator is adversarially trained to distinguish the generated images and the real style images,

$$\min_D \mathcal{L}_D = \hat{\mathcal{L}}_A + \lambda_{DS} \hat{\mathcal{L}}_{DS}, \quad (6)$$

where  $\hat{\mathcal{L}}_A = \mathbb{E}[\log \text{Prob}(\text{Fake}|D(\hat{x})) + \log \text{Prob}(\text{Real}|D(y))]$ , and  $\hat{\mathcal{L}}_{DS} = \mathbb{E}[\log \text{Prob}(s|D(\hat{x})) + \log \text{Prob}(s|D(y))]$ .

**Discriminator for ranking.** The adversarially trained discriminator characterizes the real style images, and hence can be used to rank the generated images. We rank the stylized images  $\hat{x}$  based on the likelihood score  $\text{Prob}(s|D(\hat{x})) * \text{Prob}(\text{Real}|D(\hat{x}))$ .

### 3.3. Ablation study

The encoder-decoder architecture and adaIN module have been shown to be effective in previous work [HB17]. We use visual examples to show the importance of mask module and adversarial training in the proposed method in figure 2. We present results from adversarially trained network without mask module, network with mask module but trained without adversarial loss, and the proposed method. When trained without adversarial loss, the network produces visually similar results with or without mask module as the network is over-parameterized.

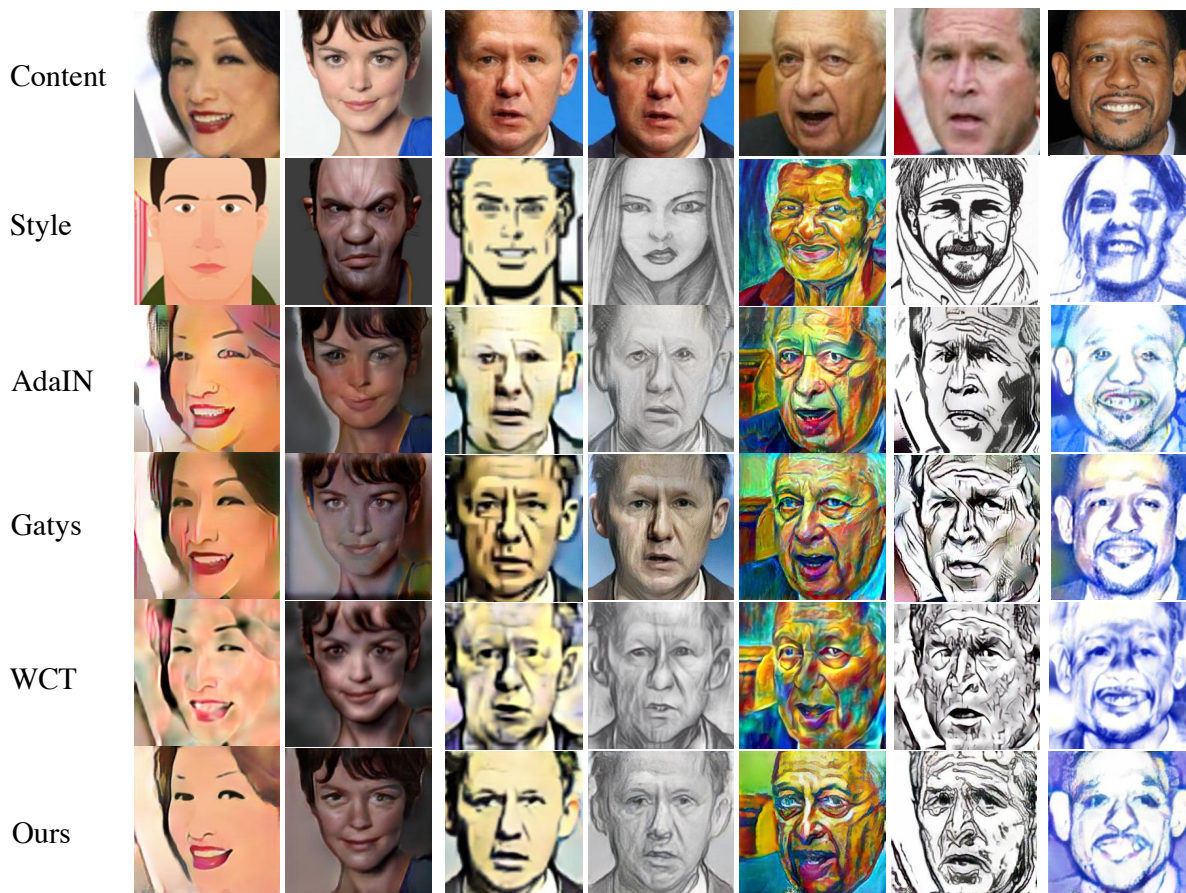
Our adversarial training significantly improves the visual quality of the generated images in general. The block effects and many other artifacts are removed through adversarial training, which makes the generated images look more “natural”. Moreover, the data-driven discriminator learns to distinguish foreground and background well; adversarial training cleans the background and adds more details to the foreground. Our mask module controls the

stylization level at different spatial location of the image, which significantly improves the stylization of salient components like eyes, nose and mouth of a face. The salient regions are repeatedly captured by the deep features from high-level layers, which can make them difficult to handle when adjusting the statistics of the features. By controlling the stylization level, the mask module prevents over-stylization of salient region, and also helps adversarial training by relieving the mode collapse of salient regions.

## 4. Experiments

We qualitatively and quantitatively evaluate the proposed method with experiments. We extensively use the Behance dataset [WFJ\*17] for training and testing. Behance [WFJ\*17] is a large-scale dataset of artistic images, which contains coarse category labels for content and style. We use the seven media labels in Behance as style category: vector art, 3D graphics, comic, graphite, oil paint, pen ink, and water color. We create four subsets from the Behance images for face, bird, car, and building. Our face dataset is created by running a face detector on a subset of images with people as content label and contains roughly 15,000 images for each style. The other three are created by selecting the top 5000 ranked images of each media for the content, respectively. We add describable textures Dataset (DTD) [cimpoi14describing] as another style category to improve the robustness of our method. We add natural images as both content images and an extra style for each subset. Specifically, we use labeled faces in the wild (LFW) [HRBL07], the first 16,000 images of CelebA dataset [LLWT15], Caltech-UCSD birds dataset [WBM\*10], cars dataset [KSDF13], and Oxford building dataset [PCI\*07]. In total, we have nine style categories in our data. We split both content and style images into





**Figure 3:** Qualitative evaluation for style transfer: We shown examples of transferring photos to seven different styles. AdaIN and WCT will generate artifacts and undesired textures. Gatys’ results are more visually appealing, but the optimization is slow, and it is hard to choose the parameter to control stylization level. Our method efficiently generate clean and stylized images.

**Table 1:** Quantitative evaluation for style transfer. Our method is preferred by human annotators and outperforms baselines.

	vectorart	3D graphics	comic	graphite	oil paint	pen ink	water color	all
AdaIN [HB17]	0.2849	0.2029	0.2314	0.1277	0.3018	0.2151	0.2118	0.2199
WCT [LFY*17]	0.1134	0.1957	0.2066	<b>0.4754</b>	0.3350	0.2868	<b>0.4409</b>	0.3001
Ours	<b>0.6017</b>	<b>0.6014</b>	<b>0.5620</b>	0.3969	<b>0.3632</b>	<b>0.4981</b>	0.3473	<b>0.4800</b>

training/testing set, and use unseen testing images for our evaluation. The total number of training/testing images are 122,247 / 11,325 for face, 35,000 / 3,505 for bird, 36,940 / 3,700 for car, and 34,374 / 3,444 for building.

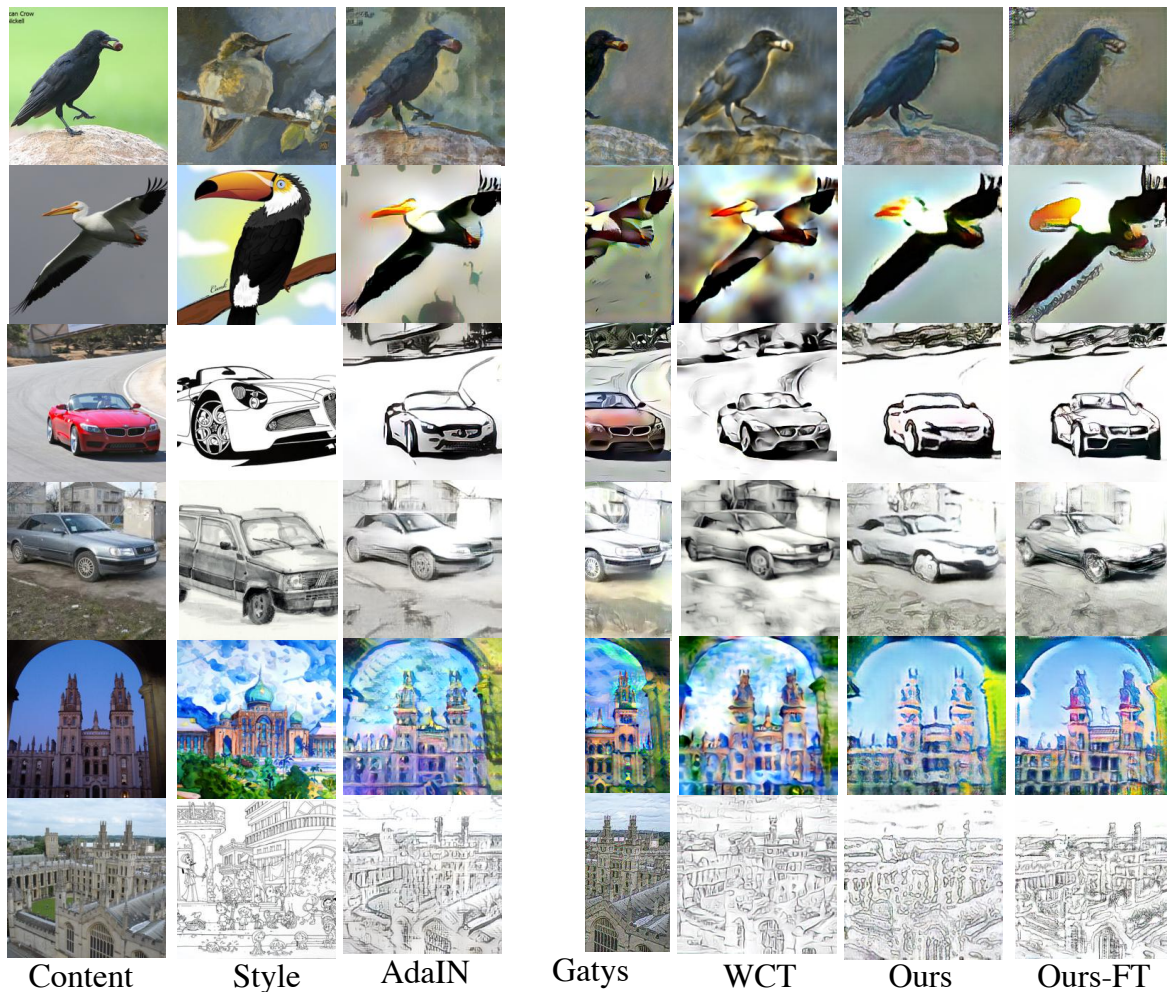
We train the network on face images, and then fine-tune it on bird, car, and building. We use Adam optimizer with prediction method [YSX\*18] with learning rate  $2e-4$  and parameter  $\beta_1 = 0.5, \beta_2 = 0.9$ . We train the network with batch size 56 for 150 epochs and linearly decrease the learning rate after 60 epochs. It takes about 8 hours to complete on a workstation with 4 GPUs. We set all weights in our combined loss (3) as 1 except for  $\lambda_s = 200$  for the style loss. The weights are chosen so that different compo-

nents of the loss have similar numerical scales. The training code and pre-trained model in Pytorch will be released.

We compare with arbitrary style transfer methods, the optimization framework of neural style transfer (Gatys) [GEB16], and two state-of-the-art methods, adaptive instance normalization (AdaIN) [HB17] and feature transformation (WCT) [LFY\*17]. Note that our approach, AdaIN and WCT apply feed-forward network for style transfer, which are much faster than Gatys method.

#### 4.1. Evaluation of style transfer

We qualitatively compare our approach with previous arbitrary style transfer methods, and present some results in figure 3. We



**Figure 4:** Qualitative evaluation for general objects. This task is more difficult for our GAN-based method because the training data is more noisy, especially for bird images with large diversity. Our method can generate clean background, detailed foreground, and better stylized strokes.

**Table 2:** Quantitative evaluation for style transfer of building. Different methods are competitive for different styles. The overall performance of our method is better.

	vectorart	3D graphics	comic	graphite	oil paint	pen ink	water color	all
AdaIN [HB17]	0.2119	0.2703	0.3089	0.3260	0.2778	<b>0.3944</b>	0.3654	0.3203
WCT [LFY*17]	<b>0.4503</b>	<b>0.4865</b>	<b>0.3740</b>	0.1547	<b>0.4383</b>	0.2310	0.1731	0.3145
Ours	0.3377	0.2432	0.3171	<b>0.5193</b>	0.2840	0.3746	<b>0.4615</b>	<b>0.3652</b>

show seven pairs of content and style images from our face dataset, and the style images are from testing set of vector art, 3D graphics, comic, graphite, oil paint, pen ink, and water color, respectively. For Gatys method [GEB16], we tune the weight parameter, and select the best visual results from either Adam or BFGS as optimizer. For AdaIN [HB17] and WCT [LFY\*17], we use their released best models. The content and style images are from the separate testing set that have not been seen for our approach and the baseline methods.

Gatys method [GEB16] is sensitive to parameter and optimizer setting. We may get results that are not stylized enough even after parameter tuning due to the difficulty of optimization. AdaIN [HB17] often over-stylizes the content image, creates undesirable artifacts, and sometimes changes the semantic of the content image. WCT [LFY\*17] suffers from severe block effect and artifacts. The previous methods all create texture-like artifacts because of the texture-based style representation. For example, the stylized images of baselines in the first column of figure 3 have stride arti-





Figure 5: Qualitative evaluation for style ranking.

facts. Our approach generate more visually appealing results with clean background, vivid foreground, and more consistent with the style of the input.

We conduct user study on Amazon Mechanical Turk, and present quantitative results in table 1. We compare with the two recent fast style transfer methods in this study. We randomly select 10 content images and 10 style images from each Behance style category to generate 700 testing pairs. For each pair, we show the stylized images by our approach, AdaIN [HB17], and WCT [LFY\*17], and ask 10 users to select the best results. We remove the unreliable results that are labeled too soon, and show preference (click) ratio for different style categories. WCT [LFY\*17] performs well on graphite and water color, where the style images themselves are visually not “clean”. Our approach achieves the best results in the other five categories and is overall the most favorable.

#### 4.2. Evaluation of style transfer for general objects

We evaluate the performance of the proposed approach on general objects beyond face. Specifically, we test for bird, car, and building. In figure 4, we show the stylized images generated by our network trained on face (Ours), as well as fine-tuned for each object (Outs-FT). Our network trained on face generalizes well, and generates images look comparable, if not better than, the baseline methods. Fine-tuning on bird does not help the performance. The adversarial training may be too difficult for bird because the given training style images are noisy and diverse. Fine-tuning on car and building brings more details to the foreground object of our generated images. The training images of car and building are also noisy and diverse, but these objects are more structured than bird. We show more results on our performance on general object tasks in the supplementary material.

We conduct the user study for building images and report results in table 2. Our approach achieves good results for graphite and water color because of the clean background in our generated images.

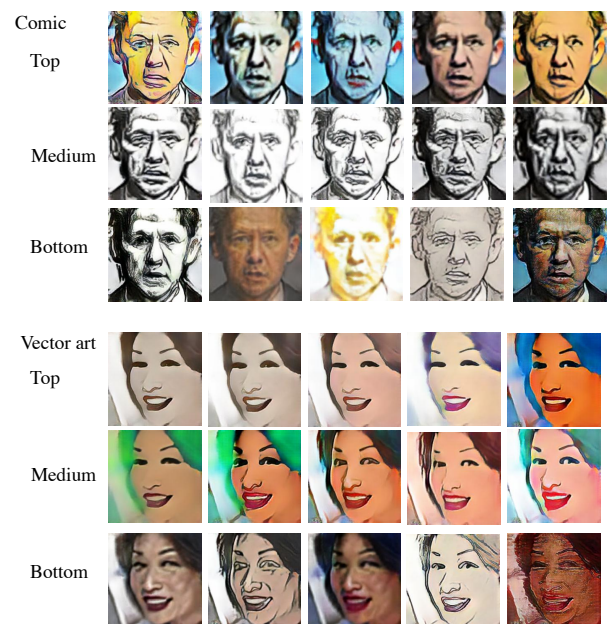
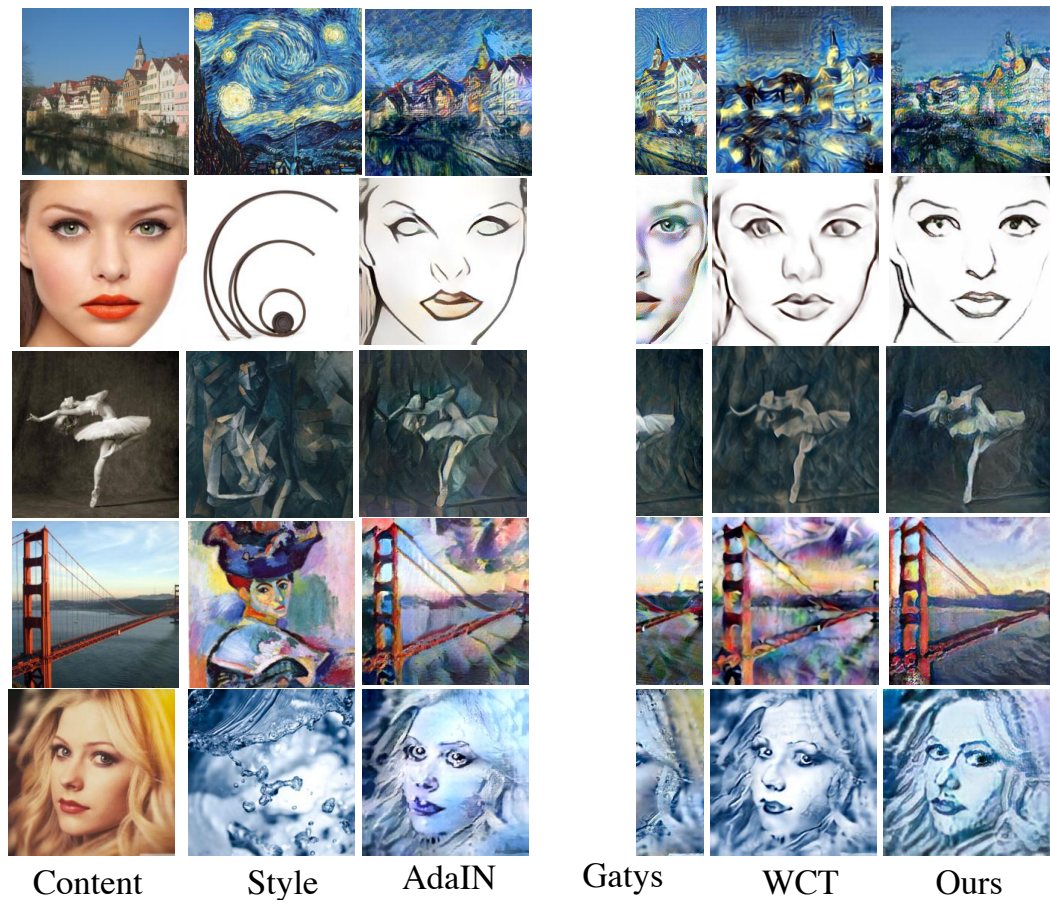


Figure 6: Ranking stylized images by our discriminator.

For the other categories, our results are comparable with baselines. Our overall performance is still the best.

#### 4.3. Evaluation for style ranking

We apply the trained discriminator to rank the generated images for a style category. Figure 5 show the top five generated images by stylizing with all the testing images in comic style. The stylized images are generated by our network, and ranked by our discriminator, a style classifier, and random selection, respectively. The



**Figure 7:** Qualitative evaluation for style transfer on texture-centric cases in previous papers. Our method generates stylized images with clean background, which are visually competitive to the previous methods that targeted only on texture transfer.

style classifier use the same network architecture as our discriminator and training data as our method. The hyper parameters are tuned to achieve the best style classification accuracy on the separate validation dataset, which makes the style classifier a strong baseline. Our generator network produced good results, and even random selected images look acceptable. The top selected results of our discriminator are more diverse, and more consistent to the comic style because of the adversarial training.

Figure 6 shows more ranked images by our discriminator at top, in the middle, and at the bottom for two content images stylized by images from two categories. The top ranked results are more visually appealing, and more consistent with the style category.

Finally, we conduct user study to compare the ranking performance of our discriminator and the baseline classifier. We generated images by stylizing ten content images with all the testing images for each of the seven Behance styles, and rank the 70 sets of results. We comparing the rank of each generated image by discriminator and classifier, and select five images that are ranked higher by our discriminator, and five images that are ranked higher by the baseline classifier. We show the ten images to ten users and ask them to select five images for each set. The preference ratio of our

discriminator is 0.5068 comparing to 0.4932 of classifier. We beat a strong baseline in a highly subjective and challenging evaluation.

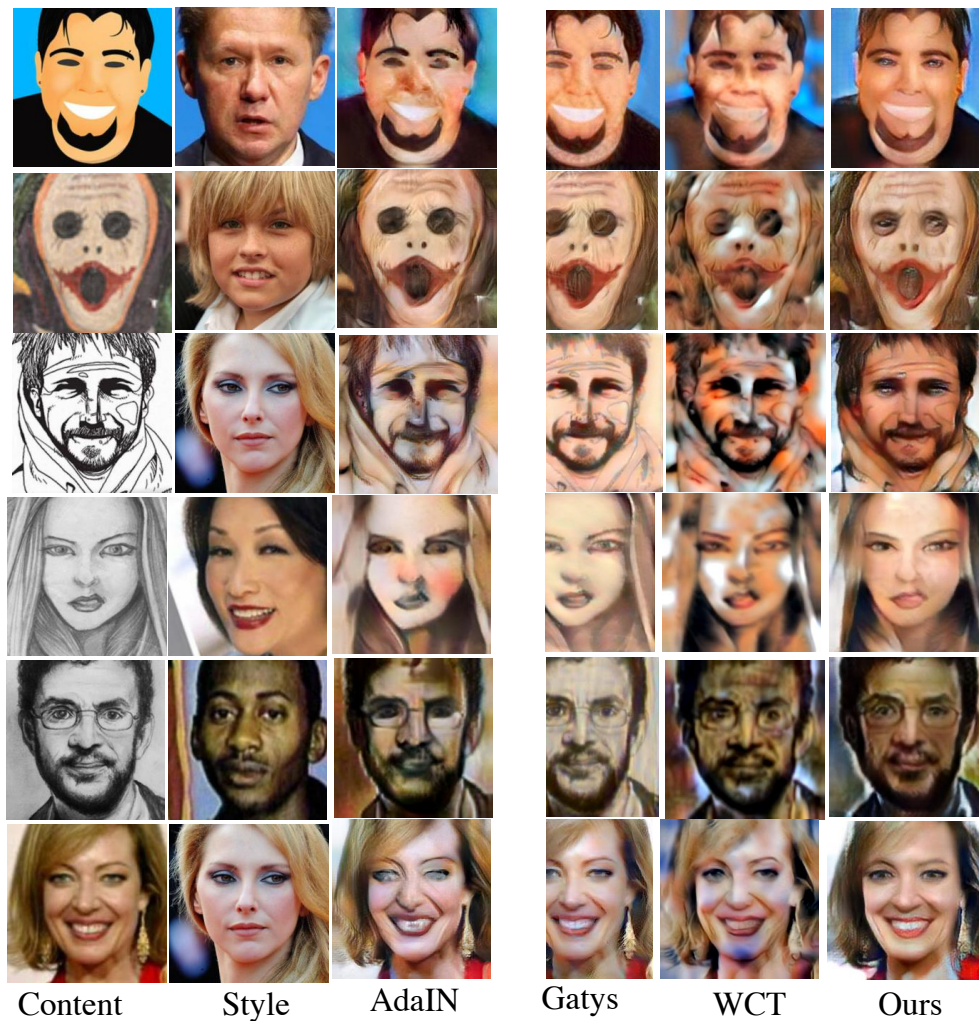
## 5. Supplemental experiments

In this section, we present supplemental experiments to show some side effect of the proposed method. We first demonstrate our method can be applied to previous style transfer test cases which focus on transferring textures of the style image. We then show that the proposed method can be applied to destylization and generate images look more realistic than baselines.

### 5.1. Examples for general style transfer

In figure 7, we evaluate on test cases from previous style transfer papers. The style images have rich texture information, and the content images vary from face to building. Our network is trained on our face dataset described in section 4. Our network generalizes well and produces comparable results, if not better than, comparing with baselines. Particularly, our approach often generates clean background without undesired artifacts.





**Figure 8:** *Qualitative evaluation for destylization.*

## 5.2. Destylization

We show that if we also use artistic images as content images during training, the exact same architecture can be used to destylize images (figure 8). Destylization is a difficult task because we only use one network to destylize diverse artistic images. The training also becomes much more difficult as the number of pairs increase square to the samples. Though there is still room to improve, our adversarial training and network architecture look promising in limited training time. The last row in 8 also suggests our network can transfer style of photorealistic images, which is difficult for the baselines.

## 6. Conclusion and discussion

We propose a feed-forward network that uses adversarial training to enhance the performance of arbitrary style transfer. We use both conditional generator and conditional discriminator to tackle multi-domain input and output. Our generator is inspired by the recent

progress in arbitrary style transfer, and our discriminator is inspired by the recent progress in generative adversarial networks. Our approach combines the best of both worlds. We propose a mask module that helps in both adversarial training and style transfer. Moreover, we show that our trained discriminator can be used to select representative stylized image, which has been a long-standing problem.

Previous style transfer and GAN-based image translation methods only target on one domain, such as transferring the style of oil paint, or transforming from natural images to sketches. We systematically study the style transfer problem on a large-scale dataset of diverse artistic images. We can train one network to generate images in different styles, such as comic, graphite, oil paint, water color and vector art. Our approach generates more visually appealing results than previous style transfer methods, but there is still room to improve. For example, transferring image into 3D graphics with the arbitrary style transfer network is still challenging.

## References

- [AFK\*18] AZADI, SAMANEH, FISHER, MATTHEW, KIM, VLADIMIR, et al. "Multi-Content GAN for Few-Shot Font Style Transfer". *CVPR* (2018) 2.
- [ARS\*18] ALMAHAIRI, AMJAD, RAJESWAR, SAI, SORDONI, ALESSANDRO, et al. "Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data". *ICML* (2018) 2.
- [CDH\*16] CHEN, XI, DUAN, YAN, HOUTHOOFT, REIN, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". *NIPS*. 2016 2, 3.
- [CS16] CHEN, TIAN QI and SCHMIDT, MARK. "Fast patch-based style transfer of arbitrary style". *arXiv preprint arXiv:1612.04337* (2016) 2.
- [CYL\*17] CHEN, DONGDONG, YUAN, LU, LIAO, JING, et al. "Stylebank: An explicit representation for neural image style transfer". *CVPR*. 2017 2.
- [DSK17] DUMOULIN, VINCENT, SHLENS, JONATHON, and KUDLUR, MANJUNATH. "A learned representation for artistic style". *ICLR* (2017) 2.
- [ELEM17] ELGAMMAL, AHMED, LIU, BINGCHEN, ELHOSEINY, MOHAMED, and MAZZONE, MARIAN. "CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms". *arXiv preprint arXiv:1706.07068* (2017) 2.
- [FSDH16] FRIGO, ORIEL, SABATER, NEUS, DELON, JULIE, and HELLIER, PIERRE. "Split and match: Example-based adaptive patch sampling for unsupervised style transfer". *CVPR*. 2016, 553–561 2.
- [FZ18] FALONG SHEN, SHUICHENG YAN and ZENG, GANG. "Neural Style Transfer Via Meta Networks". *CVPR*. 2018 2.
- [GCLY18] GU, SHUYANG, CHEN, CONGLIANG, LIAO, JING, and YUAN, LU. "Arbitrary Style Transfer with Deep Feature Reshuffle". *CVPR* (2018) 2.
- [GEB\*17] GATYS, LEON A, ECKER, ALEXANDER S, BETHGE, MATTHIAS, et al. "Controlling perceptual factors in neural style transfer". *CVPR*. 2017 2.
- [GEB15] GATYS, LEON, ECKER, ALEXANDER S, and BETHGE, MATTHIAS. "Texture synthesis using convolutional neural networks". *NIPS*. 2015 1.
- [GEB16] GATYS, LEON A, ECKER, ALEXANDER S, and BETHGE, MATTHIAS. "Image style transfer using convolutional neural networks". *CVPR*. 2016 1, 2, 5, 6.
- [GLK\*17] GHIASI, GOLNAZ, LEE, HONGLAK, KUDLUR, MANJUNATH, et al. "Exploring the structure of a real-time, arbitrary neural artistic stylization network". *BMVC* (2017) 2.
- [GPM\*14] GOODFELLOW, IAN, POUGET-ABADIE, JEAN, MIRZA, MEHDI, et al. "Generative adversarial nets". *NIPS*. 2014 2.
- [HB17] HUANG, XUN and BELONGIE, SERGE. "Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization". *CVPR*. 2017, 1501–1510 1–7.
- [HLBK18] HUANG, XUN, LIU, MING-YU, BELONGIE, SERGE, and KAUTZ, JAN. "Multimodal Unsupervised Image-to-Image Translation". *ECCV* (2018) 1, 2.
- [HRBL07] HUANG, GARY B., RAMESH, MANU, BERG, TAMARA, and LEARNED-MILLER, ERIK. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07–49. University of Massachusetts, Amherst, Oct. 2007 4.
- [HZRS15] HE, KAIMING, ZHANG, XIANGYU, REN, SHAOQING, and SUN, JIAN. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". *ICCV*. 2015, 1026–1034 3.
- [IS15] IOFFE, SERGEY and SZEGEDY, CHRISTIAN. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *ICML*. 2015, 448–456 3.
- [IZZE17] ISOLA, PHILLIP, ZHU, JUN-YAN, ZHOU, TINGHUI, and EFROS, ALEXEI A. "Image-to-image translation with conditional adversarial networks". *CVPR* (2017) 1–3.
- [JAF16] JOHNSON, JUSTIN, ALAHI, ALEXANDRE, and FEI-FEI, LI. "Perceptual losses for real-time style transfer and super-resolution". *ECCV*. Springer. 2016, 694–711 1, 2.
- [JLY\*18] JING, YONGCHENG, LIU, YANG, YANG, YEZHOU, et al. "Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields". *ECCV* (2018) 2.
- [JYF\*17] JING, YONGCHENG, YANG, YEZHOU, FENG, ZUNLEI, et al. "Neural style transfer: A review". *arXiv preprint arXiv:1705.04058* (2017) 2.
- [KCK\*17] KIM, TAEKSOO, CHA, MOONSU, KIM, HYUNSOO, et al. "Learning to discover cross-domain relations with generative adversarial networks". *ICML* (2017) 2.
- [KSDF13] KRAUSE, JONATHAN, STARK, MICHAEL, DENG, JIA, and FEI-FEI, LI. "3D Object Representations for Fine-Grained Categorization". *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013 4.
- [KSH12] KRIZHEVSKY, ALEX, SUTSKEVER, ILYA, and HINTON, GEOFFREY E. "Imagenet classification with deep convolutional neural networks". *NIPS*. 2012, 1097–1105 2.
- [LBK17] LIU, MING-YU, BREUEL, THOMAS, and KAUTZ, JAN. "Unsupervised Image-to-Image Translation Networks". *NIPS* (2017) 2.
- [LFY\*17] LI, YIJUN, FANG, CHEN, YANG, JIMEI, et al. "Universal style transfer via feature transforms". *NIPS*. 2017, 385–395 1, 2, 5–7.
- [LLL\*18] LI, YIJUN, LIU, MING-YU, LI, XUETING, et al. "A Closed-form Solution to Photorealistic Image Stylization". *ECCV* (2018) 2.
- [LLWT15] LIU, ZIWEI, LUO, PING, WANG, XIAOGANG, and TANG, XIAOOU. "Deep Learning Face Attributes in the Wild". *ICCV*. 2015 4.
- [LW16a] LI, CHUAN and WAND, MICHAEL. "Combining markov random fields and convolutional neural networks for image synthesis". *CVPR*. 2016, 2479–2486 2.
- [LW16b] LI, CHUAN and WAND, MICHAEL. "Precomputed real-time texture synthesis with markovian generative adversarial networks". *ECCV*. Springer. 2016, 702–716 2.
- [LWLH17] LI, YANGHAO, WANG, NAIYAN, LIU, JIAYING, and HOU, XIAODI. "Demystifying neural style transfer". *IJCAI* (2017) 2.
- [LXT\*18] LI, HAO, XU, ZHENG, TAYLOR, GAVIN, et al. "Visualizing the loss landscape of neural nets". *NeurIPS*. 2018, 6391–6401 2.
- [LYY\*17] LIAO, JING, YAO, YUAN, YUAN, LU, et al. "Visual attribute transfer through deep image analogy". *ACM (TOG)* 36.4 (2017), 120 2.
- [MK18] MIYATO, TAKERU and KOYAMA, MASANORI. "cGANs with projection discriminator". *ICLR* (2018) 2, 3.
- [OOS17] ODENA, AUGUSTUS, OLAH, CHRISTOPHER, and SHLENS, JONATHON. "Conditional image synthesis with auxiliary classifier gans". *ICML* (2017) 2, 3.
- [PCI\*07] PHILBIN, J., CHUM, O., ISARD, M., et al. "Object Retrieval with Large Vocabularies and Fast Spatial Matching". *CVPR*. 2007 4.
- [RBG\*17] ROYER, AMÉLIE, BOUSMALIS, KONSTANTINOS, GOUWS, STEPHAN, et al. "XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings". *arXiv preprint arXiv:1711.05139* (2017) 2.
- [RDS\*15] RUSSAKOVSKY, OLGA, DENG, JIA, SU, HAO, et al. "Imagenet large scale visual recognition challenge". *IJCV* (2015) 2.
- [RMC16] RADFORD, ALEC, METZ, LUKE, and CHINTALA, SOUMITH. "Unsupervised representation learning with deep convolutional generative adversarial networks". *ICLR* (2016) 3.
- [SLSW18] SHENG, LU, LIN, ZIYI, SHAO, JING, and WANG, XIAOGANG. "Avatar-Net: Multi-scale Zero-shot Style Transfer by Feature Decoration". *CVPR* (2018) 2.

- [SZ15] SIMONYAN, KAREN and ZISSERMAN, ANDREW. “Very deep convolutional networks for large-scale image recognition”. *ICLR* (2015) 2.
- [TPW17] TAIGMAN, YANIV, POLYAK, ADAM, and WOLF, LIOR. “Unsupervised cross-domain image generation”. *ICLR* (2017) 1, 2.
- [ULVL16] ULYANOV, DMITRY, LEBEDEV, VADIM, VEDALDI, ANDREA, and LEMPITSKY, VICTOR S. “Texture Networks: Feed-forward Synthesis of Textures and Stylized Images.” *ICML*. 2016, 1349–1357 2.
- [UVL17] ULYANOV, DMITRY, VEDALDI, ANDREA, and LEMPITSKY, VICTOR. “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis”. *CVPR*. 2017 1, 2.
- [WBM\*10] WELINDER, P., BRANSON, S., MITA, T., et al. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010 4.
- [WFJ\*17] WILBER, MICHAEL J., FANG, CHEN, JIN, HAILIN, et al. “BAM! The Behance Artistic Media Dataset for Recognition Beyond Photography”. *ICCV*. Oct. 2017 4.
- [WOZW17] WANG, XIN, OXHOLM, GEOFFREY, ZHANG, DA, and WANG, YUAN-FANG. “Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer”. *CVPR*. 2017, 5239–5247 2.
- [WRB17] WILMOT, PIERRE, RISSER, ERIC, and BARNES, CONNELLY. “Stable and controllable neural texture synthesis and style transfer using histogram losses”. *arXiv preprint arXiv:1701.08893* (2017) 2.
- [XHH18] XU, ZHENG, HSU, YEN-CHANG, and HUANG, JIAWEI. “Training student networks for acceleration with conditional adversarial networks”. *BMVC* (2018) 3.
- [XYLS18] XU, ZHENG, YANG, XITONG, LI, XUE, and SUN, XI-AOSHUAI. “Strong Baseline for Single Image Dehazing with Deep Features and Instance Normalization”. *BMVC* (2018) 2.
- [YSX\*18] YADAV, ABHAY, SHAH, SOHIL, XU, ZHENG, et al. “Stabilizing Adversarial Nets With Prediction Methods”. *ICLR* (2018) 3, 5.
- [YXL18] YANG, XITONG, XU, ZHENG, and LUO, JIEBO. “Towards perceptual image dehazing by physics-based disentanglement and adversarial training”. *AAAI*. 2018 2.
- [YZTG17] YI, ZILI, ZHANG, HAO, TAN, PING, and GONG, MINGLUN. “DualGAN: Unsupervised Dual Learning for Image-To-Image Translation”. *CVPR*. 2017, 2849–2857 2.
- [ZCZ18] ZHANG, YEXUN, CAI, WENBIN, and ZHANG, YA. “Separating Style and Content for Generalized Style Transfer”. *CVPR* (2018) 2.
- [ZD17] ZHANG, HANG and DANA, KRISTIN. “Multi-style generative network for real-time transfer”. *arXiv preprint arXiv:1703.06953* (2017) 2.
- [ZPIE17] ZHU, JUN-YAN, PARK, TAESUNG, ISOLA, PHILLIP, and EFROS, ALEXEI A. “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”. *CVPR*. 2017, 2223–2232 1, 2.
- [ZZP\*17] ZHU, JUN-YAN, ZHANG, RICHARD, PATHAK, DEEPAK, et al. “Toward multimodal image-to-image translation”. *NIPS*. 2017, 465–476 2.