# DRB-GAN: A Dynamic ResBlock Generative Adversarial Network for Artistic Style Transfer

Wenju Xu[1], Chengjiang Long[2*], Ruisheng Wang[3], Guanghui Wang[4*]
[1]OPPO US Research Center, InnoPeak Technology Inc, Palo Alto, CA, USA
[2]JD Finance America Corporation, Mountain View, CA, USA
[3]Department of Geomatics Engineering, University of Calgary, Alberta, Canada
[4]Department of Computer Science, Ryerson University, Toronto, ON, Canada
xuwenju123@gmail.com, cjfykx@gmail.com, ruiswang@ucalgary.ca, wangcs@ryerson.ca

## Abstract

*The paper proposes a Dynamic ResBlock Generative Adversarial Network (DRB-GAN) for artistic style transfer. The style code is modeled as the shared parameters for Dynamic ResBlocks connecting both the style encoding network and the style transfer network. In the style encoding network, a style class-aware attention mechanism is used to attend the style feature representation for generating the style codes. In the style transfer network, multiple Dynamic ResBlocks are designed to integrate the style code and the extracted CNN semantic feature and then feed into the spatial window Layer-Instance Normalization (SW-LIN) decoder, which enables high-quality synthetic images with artistic style transfer. Moreover, the style collection conditional discriminator is designed to equip our DRB-GAN model with abilities for both arbitrary style transfer and collection style transfer during the training stage. No matter for arbitrary style transfer or collection style transfer, extensive experiments strongly demonstrate that our proposed DRB-GAN outperforms state-of-the-art methods and exhibits its superior performance in terms of visual quality and efficiency. Our source code is available at* https://github.com/xuwenju123/DRB-GAN.

## 1. Introduction

Artistic style transfer is to synthesize an image sharing structure similarity of the content image and reflecting the style of the artistic style. Here, artistic style implies the genre of paintings by the artist, and the artistic images refer to a set of images created by the same artist, and each image has a unique character. As shown in Figure 1, style image 1 and all the style images in style collection 1 are created by *Pablo Picasso*, in which the style includes color, brushstroke, form, or use of light. Therefore, an ideal artistic style transfer should be able to synthesize images with consistent style genre and also take the diverse artworks of the artist into account.

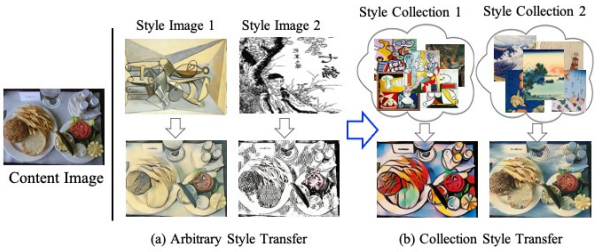



Figure 1. Examples of two types of artistic style transfer: (a) arbitrary style transfer and (b) collection style transfer. Note that the style image 1 and style collection 1 are from the artist *Pablo Picasso*, and the style image 2 and style collection 2 are from the artist *Ukiyo-e*. Our proposed DRB-GAN experimentally performs well on both arbitrary style transfer and collection style transfer.

To facilitate more efficient artistic style transfer, some prior works have explored arbitrary style transfer [17, 20], which heavily rely on only one arbitrary style image. Hence, they are not effective to produce a bunch of results that reflect the understanding of artistic style characterized by color, scale, and stroke size of the artistic work set. Some recent efforts on generative adversarial networks (GANs) [55, 41, 55, 41, 5, 36] have succeeded in collection style transfer, which considers each style image in a style collection as a domain. However, the existing collection style transfer methods only recognize and transfer the domain dominant style clues and thus lack the flexibility of exploring style manifold.

In this paper, we propose a Dynamic ResBlock Generative Adversarial Network (DRB-GAN) for artistic style transfer. As illustrated in Figure 2, it consists of a style encoding network, a style transfer network, and a style col-

*This work was supervised by Chengjiang Long and Guanghui Wang.

lection discriminative network. In particular, inspired by the ideas of DIN [20] and StyleGAN [23], we model the "*style code*" as the shared parameters for Dynamic Convolutions and AdaINs in dynamic ResBlocks, and design multiple Dynamic Residual Blocks (DRBs) at the bottleneck in the style transfer network. Note that each DRB consists of a Convolution Layer, a Dynamic Convolution [3] layer, a ReLU layer, an AdaIN [17] layer, and an instance normalization layer with a residual connection. Such treatment is to attentively adjust the shared parameters for Dynamic Convolutions and adaptively adjust affine parameters for AdaINs to ensure the statistic matching in bottleneck feature spaces between content images and style images.

We incorporate a fixed pretrained VGG encoder [43] and a learnable encoder as feature extractor in the style encoding network to capture style class-aware feature representation. The output style class-aware probabilities can be used as attention weights to attend to the style features for style code recalibration. As the style class-aware attention mechanism is learned via images in style collections, the output "*style code*" captures the underlying discriminative information in the style image collections. Note that our attention mechanism enforces a better clustering of the "*style codes*", which is different from previous class activation mapping based methods [25, 53] that aim at highlighting spatial regions.

With the "*style code*" from the style encoding network, multiple DRBs can adaptively proceed the semantic features extracted from the CNN encoder in the style transfer network then feed them into the spatial window Layer-Instance Normalization (SW-LIN) decoder to generate synthetic images. Specially, we borrow the idea of local feature normalization from [28] and design an SW-LIN function that dynamically combines the local channel-wise and layer-wise normalization with a learnable parameter in each decoder block. With the spatial window constraint, our SW-LIN is able to flexibly shift the mean and variance in the feature spaces. As a consequence, our SW-LIN decoder can avoid the possible artifacts and retain the capability to synthesize high-resolution stylization.

As the "*style code*" captures the underlying discriminative information in the style encoding network, it is easy to apply the learned style network on each style image in a style collection and get a set of style codes. We can apply the weighted average on the style codes to obtain the "*collection style code*" and then feed it into the style transfer network to conduct the collection style transfer. Moreover, our discriminative network takes several style images sampled from the target style collection of the same artist as references to ensure consistency in the feature space. Together with the perception supervision, our well-designed discriminator provides good guidance for our DRB-GAN to own abilities for both arbitrary style transfer and collection style transfer gradually, shrinking their gap at the training stage. With extensive experiments, we have demonstrated the effectiveness of our proposed DRB-GAN on both arbitrary style transfer and collection style transfer.

Several aspects distinguish our work from previous style transfer models [41, 28, 44]. First of all, our DRB-GAN introduces a novel prototype for artistic style transfer, in which "*style code*" is modeled as the shared parameters for Dynamic ResBlocks, connecting both the style encoding network and the style transfer network, to shrink the gap between arbitrary style transfer and collection style transfer in a unified model. Second, we introduce a style class-aware attention mechanism for style code recalibration and then employ well-designed multiple dynamic ResBlocks to integrate the style code and the extracted semantic feature to realize artistic style transfer when generating high-quality synthetic images. Last but not least, the discriminative network makes full use of style images sampled from the target collection as a reference which enforces our DRB-GAN's ability for collection style transfer. Together with perception supervision, the ability for arbitrary style transfer can be well preserved and improved at the training stage.

Both quantitative and qualitative experiments demonstrate the effectiveness and efficiency of the proposed DRB-GAN, as well as its superior performance in artistic style transfer, regardless of arbitrary or collection style transfer.

## 2. Related Work

**Generative Adversarial Networks (GANs)** [11] have been successfully applied to visual recognition [14, 37, 15, 39, 38, 16], object detection [19], image generation [45, 12, 23, 48, 47], image translation [56, 29, 25], shadow removal [6, 46, 52, 51], image captioning [1, 7], *etc*. These GAN models are trained to minimize the discrepancy between distributions of the training data and unobserved generations. Our GAN model is designed with a special discriminator that judges the generated images by taking similar images from the target collection as a reference.

**Arbitrary style transfer.** Gatys *et al*. [10] for the first time takes a pre-trained neural network to optimize synthesized images. However, this method inefficiently searches for a numerical solution in pixel space. To address this, recent approaches rely on a learnable neural network to match the statistical information in feature space. The earliest Per-Style-Per-Model (PSPM) algorithms train a single model for one particular style image[21]. While the Multiple-Style-Per-Model (MSPM) algorithms are proposed [50, 49, 2] to use one model for multiple style images. For instance, the StyleBank [2] uses one single model to incorporate feature representations of multiple style images. Recently, the Arbitrary-Style-Per-Model (ASPM) algorithms [32, 4, 30] are proposed to transfer arbitrary new styles in one unified model. MetaNet [42] introduces a pa-

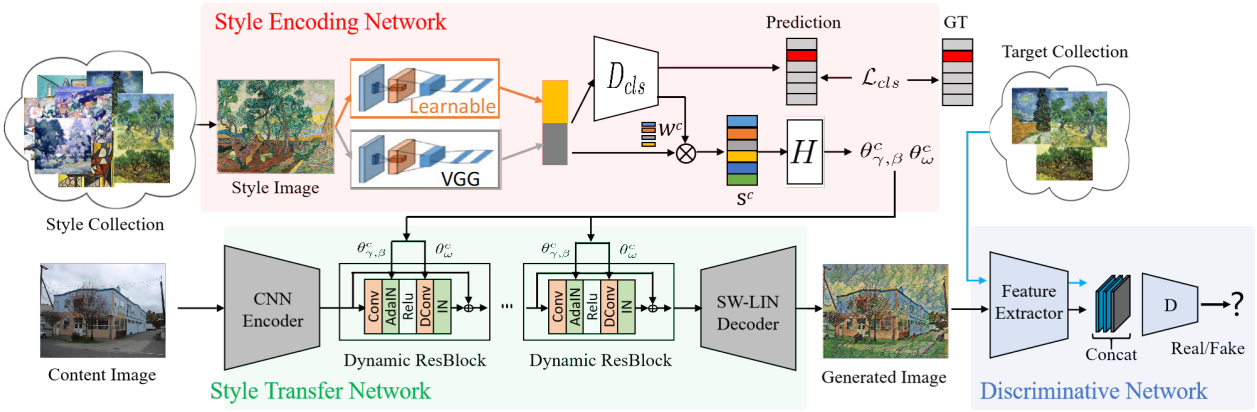



Figure 2. An overview of the proposed DRB-GAN, which consists of a style encoding network, a style transfer network, and a discriminative network. The style code modeled as the shared parameter for dynamic ResBlocks is the output from the style encoding network, with a combination of a pre-trained VGG encoder and a learnable encoder as the feature extractor. A style class-aware attention mechanism is employed to recalibrate the style code. The final style code is then fed into the style transfer network designed as an encoder-decoder structure with multiple well-designed dynamic ResBlocks.

rameter network to generate network parameters based on the style image. AdaIN [17] and DIN [20] methods employ conditional instance normalization to dynamically generate the affine parameter in the instance normalization layer. The CST [44] introduces a unified model for conditional style transfer. In a different manner, Li *et al*. [31] utilizes the whitening and coloring transforms (WCT) on the style features. However, these style transfer algorithms consider one individual style image as one style. This assumption ignores the concept of the artistic collection of an artist. Instead, our model takes novel Dynamic ResBlocks to efficiently deal with the artistic style transfer task.

**Collection style transfer.** The collection style transfer methods [8, 5, 35] work on image collections or domains. They are built upon GANs to map inputs into a different domain. We refer to this type of method as a Per-Domain-Per-Model (PDPM) algorithm. For instance, CycleGAN [55] takes one generator to translate images into another domain and uses another generator to translate them back for cyclic consistency. The AST [41] extends the GAN-based model for high-resolution artistic style transfer. The CSD [28] introduces a content transformation block to preserve the content structure in the synthetic images. Recent work [18, 36] propose to handle the multiple domain translation task. In contrast, our proposed method models the "*collection style code*" in a dynamic way to facilitate efficiency in handling multiple domains. We, therefore, refer to our mother as a Multiple-Domain-Per-Model (MDPM) method.

## 3. Proposed Approach

As illustrated in Figure 2, our DRB-GAN consists of three networks, *i.e.*, a style encoding network and a style transfer network to formulate the image generator $G$, a discriminative network as the discriminator $D$ to ensure the generated images with desired style consistent with style images in the collection. Let us use subscripts $c$ to indicate the $c$-th style. Given a content image $x \in X$ and an arbitrary style image $y^c \in Y$ randomly sampled from $N$ different style image collections, our goal is to transfer the content image with the generator $G$ to produce a desired synthetic image $\tilde{x}^c$, ensuring a consistent style with the style image $y^c$ via the discriminator $D$.

### 3.1. Style Encoding Network with Style Class-Aware Attention

As shown in Figure 2, we model the style code as the shared hyper-parameters for Dynamic ResBlocks which is designed to integrate dynamic convolution (DConv) and Adaptive Instance Normalization Normalization (AdaIN) [17] in a residual structure [13]. The style encoding network is to generate style code from the style image for the style transfer network on the content image.

**Attention guided feature extractor.** We introduce an architecture of style encoding by concatenating the features from a pre-trained VGG encoder and a learnable encoder. The parameters in the learnable encoder are updated while those in the VGG encoder are fixed. Since the fixed VGG network is pre-trained on the COCO dataset [34], it has seen many images with various textures, thereby it has a global property and strong generalization ability for in-the-wild textures. Considering the gap between the COCO dataset and others, it is difficult for the network with a fixed encoder to fit such a complex model. Therefore we introduce a learnable encoder as complementary to the fixed VGG encoder to extract the subtle variations in style.

Inspired by the class activation mapping (CAM) [53], we take a classification weight to recalibrate our encoded style feature, denoted as $F_s$. The attention mechanism is based on an auxiliary classifier $D_{cls}$ trained to predict the style classification probability $w^c$, which is used to the likelihood of the input style image belonging to the $c_{th}$ category. Then,

the style encoding is recalibrated as

$$s^c = \omega^c F_s, \tag{1}$$

The recalibrated style encoded feature is then fed into the weight generation module $H$ designed as multi-layer perceptions (MLPs) to determine parameter values for Dynamic ResBlocks.

**Style code generation for arbitrary style transfer.** Given the recalibrated style encoding, this module is to generate the parameters as "*style code*" in dynamic ResBlocks, which can be written as:

$$\{\theta^c_\omega, \theta^c_{\gamma,\beta}\} = \{H_\omega(s^c), H_{\gamma,\beta}(s^c)\}, \tag{2}$$

where $s^c$ is the style feature, $H_\omega(\cdot)$ is a MLP used to generate filter weights $\theta^c_\omega$ for Dynamic Convolution [3] Layers. Another MLP $H_{\gamma,\beta}(\cdot)$ creates the affine parameters $\theta^c_{\gamma,\beta}$ in the AdaIN [17] layers.

**Weighted averaging strategy for collection style transfer**. We introduce a weighted averaging strategy to extend arbitrary style encoding for collection style transfer. Specifically, we calculate the "*collection style code*" as a weighted mean of the "*style codes*" based on several representative paintings of the same artist and the corresponding weights $\pi$. For the $k$-th style image in the collection, the weight $\pi_k$ is determined by the similarity between the style image and the query content image. Therefore, we can formulate the "*collection style code*" as

$$\{\bar{\theta}^c_\omega,\ \bar{\theta}^c_{\gamma,\beta}\} = \{\frac{1}{K}\sum_{k=0}^{K}\pi_k\theta^c_{\omega_k},\ \frac{1}{K}\sum_{k=0}^{K}\pi_k\theta^c_{\gamma_k,\beta_k} | c \sim N\},$$

where $K$ is the number of style images used to calculate the mean of the generated weights at test stage, and $c$ indicates the target style domain. Our experimental results show that our weighted averaging strategy produces impressive results on collection style transfer task.

### 3.2. Style Transfer Network with Dynamic ResBlocks

Our style transfer network contains a CNN Encoder to down-sample the input, multiple dynamic residual blocks, and a spatial window Layer-Instance Normalization (SW-LIN) decoder to up-sample the output. With different "*style code*" parameters, the style transfer network converts the content image into different styles.

**Dynamic ResBlock.** As the core of the transfer network, each dynamic ResBlock is composed of a Convolution Layer, a Dynamic Convolution [3] layer, a ReLU layer, an AdaIN [17] layer, and an instance normalization layer with a residual structure. It is designed to integrate the advantages of both dynamic convolution and adaptive instance normalization layer in the residual block. Note that all of the parameters in layers are dynamically generated from the style encoding network.

**SW-LIN Decoder.** The GAN model tends to produce artifacts in the generated samples [24], which significantly degrades its applications. Many researchers attempt to replace the instance normalization function with the layer normalization function in the decoder modules to remove the artifacts. After studying these normalization operations, we observe that instance normalization normalizes each feature map separately, thereby potentially destroying any information found in the magnitudes of the features relative to each other. While layer normalization operation normalizes the feature map together, thereby potentially demolishing each feature map as the representation of the style. Therefore, we equip the decoder blocks with spatial window Layer-Instance Normalization (SW-LIN) function which dynamically combine these normalization functions with learnable parameter $\rho$:

$$\text{SW-LIN}(\gamma, \beta, \rho) = \gamma(\rho\phi^c_{sw} + (1-\rho)\phi^l_{sw}) + \beta, \tag{3}$$

where $\gamma, \beta$ are learnable parameters, and $\phi^c_{sw}, \phi^l_{sw}$ are channel-wise, layer-wise normalized features, respectively. Specially, the statistic mean and variance are obtained across a window of spatial location instead of the whole input tensor $\mathbf{h}$. Formally, we get

$$\phi_{sw} = \frac{\mathbf{h} - E_{x_i \in sw}[\mathbf{h}(x_i)]}{\sqrt{Var_{x_i \in sw}[\mathbf{h}(x_i)]}}. \tag{4}$$

The SW-LIN function helps our decoder to flexibly normalize the features. As a result, without modifying the model architecture or hyper-parameters, our SW-LIN decoder can remove the artifacts and retain the capability to synthesize high-resolution stylization.

### 3.3. Discriminative Network for Arbitrary and Collection Style Transfer

Simply answering a real or fake question is not enough to provide correct supervision to the generator which aims at both individual style and collection style. To address this issue, we introduce a novel conditional discriminator with a collection of style images, which encourages the generated images to maintain the texture in any style genre.

As illustrated in Figure 2, our collection discriminator takes the generated images and several style images sampled from the target style collection as input. The feature extraction part generates a feature map for each image and we concatenate them channel-wisely. Then a small network with three Convolution layers is used to assess the quality based on the concatenated feature map. Unlike the discriminator of conditional GANs taking a category label as an additional input besides the generated image, our discriminator takes a collection of style images as a reference instead to ensure the style consistency at the feature space.

Working together with the perceptual supervision (see Equation 7) between a selected style image and the corresponding generated stylization image, our discriminator provides good guidance to train the generator for both arbitrary and collection style transfer. Consequently, the gap between the arbitrary and collection style transfer has been shrunk smoothly at the training stage.

### 3.4. Objective Functions

The objective loss function $\mathcal{L}$ is formulated with adversarial loss $\mathcal{L}_{adv}$, perceptual loss $\mathcal{L}_{per}$, and style classification loss $\mathcal{L}_{cls}$ as flollows,

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{per}\mathcal{L}_{per} + \lambda_{cls}\mathcal{L}_{cls} \tag{5}$$

where $\lambda_{per}$ and $\lambda_{cls}$ are the weight parameters.

**Adversarial loss** $\mathcal{L}_{adv}$ is designed to distinguish between the patches from synthesized images and those from a group of style images belonging to the same style collection, *i.e.*,

$$\begin{aligned}\mathcal{L}_{adv} &= E_{y^c, y_i^c \sim Y, c\sim N}[-\log D(y^c, \{y_i^c\}_{i=0}^M)] \\ &+ E_{\tilde{x}^c\sim G(x), y_j^c \sim Y, c\sim N}[-\log(1 - D(\tilde{x}^c, \{y_j^c\}_{j=0}^M))],\end{aligned} \tag{6}$$

where $M$ indicates the number of style images used by the collection discriminator at each iteration. in our experiment, we find setting $M = 2$ is enough to obtain decent performance.

**Perceptual loss** $\mathcal{L}_{per}$ is employed to compute style losses at multiple levels and content loss between the style image and the generated stylization image with a pretrained VGG network in a way similar to the prior work [33, 9], *i.e.*,

$$\mathcal{L}_{per} = \lambda_c\mathcal{L}_c + \lambda_s\mathcal{L}_s, \tag{7}$$

where the style loss is calculated by matching the mean and standard deviation of the style features:

$$\mathcal{L}_s = \mathbb{E}_{l\sim N_l}((\mu_{y^c}^l - \mu_{\tilde{x}^c}^l)^2 + (Gram_{y^c}^l - Gram_{\tilde{x}^c}^l)^2),$$

where $N_L$ is the number of involved layers (in this work, we use the $Relu_{12}$, $Relu_{22}$, $Relu_{33}$, $Relu_{43}$ and $Relu_{51}$ layers in the VGG network and that of feature maps in the $l_{th}$ layer). In addition, $\mu$ and is the mean and Gram represents the Gram matrix of the corresponding feature map. We take advantage of the content loss to preserve the structural similarity between the content image and the synthesized image. The content loss is the Euclidean distance between the target features and the features of the output image.

$$\mathcal{L}_c = E_{x\sim X, c\sim N}||\phi(x) - \phi(\tilde{x}^c)||, \tag{8}$$

where $\phi$ represents the features of $Relu_{41}$ layers. At the training stage, the style image is randomly sampled with the target domain label. We then transfer the content image to the target style domain so as to learn all the mappings across multiple artistic style domains.

**Style classification loss** $\mathcal{L}_{cls}$ is exploited for the auxiliary classification $D_{cls}$ in the style encoding network to ensure that the style class prediction is correct.

$$\mathcal{L}_{cls} = E_{y\sim Y, c\sim N}(-\log D_{cls}(c|y^c)). \tag{9}$$

## 4. Experiment

**Implementation details.** Our model is implemented by PyTorch [40]. It is trained on 624,777 content images from the Place365 [54] dataset, and eleven artistic painting collections from WikiArt [22] database. We adopt Adam [26] as the optimization solver. The learning rate is set to 0.0001. We train our model for $600,000$ iterations with a batch size of 1. The training images are augmented by random rotation and horizontal flipping, and then resized and randomly cropped to $768 \times 768$ resolution. Note that during testing, our model is able to work on images of arbitrary size. We empirically set $\lambda_c = 1$, $\lambda_s = 0.02$, $\lambda_{cls} = 1$, and $\lambda_{vgg} = 1$.

**Baseline methods.** The goal of our DRB-GAN model is to generate high-resolution artistic stylizations. We compare our DRB-GAN with state-of-the-art methods, *i.e.*, instance style transfer method like Gatys *et al*. [10], arbitrary style transfer methods including AdaIN [17], CST [44] and MetaNet[42], and collection style transfer methods like AST [41] and CycleGAN [55]). For a fair comparison, we deploy all competing methods for style transfer on images of size $768 \times 768$, unless otherwise specified. Note that we use the public released code by authors and train the models on the same training data.

**Evaluation metrics.** We use deception rate to evaluate how well the target style characteristics are transferred to generated images. The deception rate is calculated as the percentage predicted by a pretrained artist classification network for the correct artist. We also conduct human perceptual study to assess the quality of the stylization results in terms of content preservation and style consistency to the target style.

### 4.1. Arbitrary Style Transfer

#### 4.1.1 Qualitative Evaluation

**Image stylization.** We present qualitative results of different style transfer methods in Figure 3 for comparison. All the results are obtained on images of high resolution. The comparison shows the outperformance of our DRB-GAN in terms of visual quality. These images in Figure 3 (j) contain no artifacts in the regions, and most importantly, they preserve the structural similarity of the content images. On the contrary, the algorithms AdaIN and MetaNet fail to generate sharp details and fine strokes. We also observe non-negligible artificial structures in those images obtained by AdaIN, Gatys, and CycleGAN. And the models CSD [28]

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3. Performance comparison on stylized results from different models. From left to right are (a) content image, (b) style image, the results of (c) CSD, (d) AST, (e) Gatys, (f) CycleGAN, (g) AdaIN, (h) MetaNet, (i) CST, and (j) DRB-GAN, respectively.

and AST [41] fail to transfer the Kandinsky style (third row) to the content image.

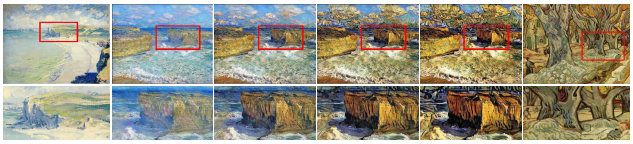


Figure 4. Interpolation results between given style samples of Monet (first column) and van Gogh (last column ). Magnified regions (row 2) show that our method mimics not only colors but also contours and textures specific to the style.

**Style interpolation.** To make smooth transitions between different style images, we operate linear interpolation on dynamically generated weights. The interpolated weights $\{\tilde{\theta}_\omega,\ \tilde{\theta}_{\gamma,\beta}\}$ are obtained as

$$\{\tilde{\theta}_\omega,\ \tilde{\theta}_{\gamma,\beta}\} = \{\alpha\theta_\omega^{c_1} + (1-\alpha)\theta_\omega^{c_2},\ \alpha\theta_{\gamma,\beta}^{c_1} + (1-\alpha)\theta_{\gamma,\beta}^{c_2} | c_1, c_2 \sim N\} \tag{10}$$

where $\alpha$ is the interpolation factor between 0 and 1, $c_1$ and $c_2$ represent style domains. Figure 4 demonstrates a smooth transition between Monet and van Gogh style with magnified details. Our model captures subtle variations between these two styles.

#### 4.1.2 Quantitative Evaluation

**Style transfer deception rate.** To quantitatively measure the performances of different models, we take the deception rate metric used by AST [41]. The deception rate is the correct rate of stylized images that were recognized by a pre-trained network as the target styles. We take the same way used in CSD [27] to measure the style transfer deception rate and report the mean deception rate in Table 1. As we can see, our method DRB-GAN achieves 0.573, which significantly outperforms other baseline methods. As a comparison, the mean accuracy of the network on real images of the artists from Wikiart is 0.626.

**Human perceptual study.** We also operate human study on the performance of different approaches. Specifically, we show each participant with 700 groups of images. Each group consists of stylized images generated by different methods based on the same content and style images. We ask the participants to choose one image that most realistically reflects the target style. The style score is computed as the frequency and a specific method is chosen as the best in the group. The content score is provided by the participants to evaluate the structural similarity of the content image. Human perceptual studies are reported in Table. 1. We can see that our method obtains the best scores, proving the superior performance of our model in terms of both style transfer and structure preservation.

Table 1. Quantitative comparison with state-of-the-art methods. Average inference time and GPU memory consumption, measured on a Titan XP GPU, for different methods with a batch size of 1 and an input image of $768 \times 768$. The column "model" is for the category of the style transfer method. For the content and style score, higher values indicate better performance. Scores are averaged over ten different styles.

| Method | GPU Time (sec) | GPU memory (MiB) | Model | Deception rate | Human studies Content score | Human studies Style score |
|---|---|---|---|---|---|---|
| Wikiart test | | | | 0.626 | - | - |
| Gatys *et al.* | 200 | 3887 | PSPM | 0.251 | 67.1% | 0.127 |
| AdaIN | 0.16 | 8872 | ASPM | 0.061 | 43.6% | 0.019 |
| WCT | 5.22 | 10720 | ASPM | 0.023 | 39.2% | 0.013 |
| PatchBased | 8.70 | 4159 | ASPM | 0.063 | 53.4% | 0.043 |
| Johnson | 0.06 | 671 | ASPM | 0.080 | 38.5% | 0.021 |
| CycleGAN | 0.07 | 1391 | PDPM | 0.130 | 43.2% | 0.012 |
| AST | 0.07 | 1043 | PDPM | 0.450 | 63.9% | 0.312 |
| DRB-GAN | 0.08 | 1324 | MDPM | **0.573** | **72.2%** | **0.453** |

**Speed and memory.** The comparison on time and memory consumption are also listed in Table. 1. We observe that our approach has comparable speed and modest demand on GPU memory. Thanks to the weight generation module that creates a smooth manifold structure, our model can perform flexible style transfer via interpolation and model averaging, which significantly improves the efficiency of our model as a Multiple-Domain-Per-Mode (MDPM) algorithm. On the contrary, other models lack the capability of performing diverse or collection artistic style transfer.

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

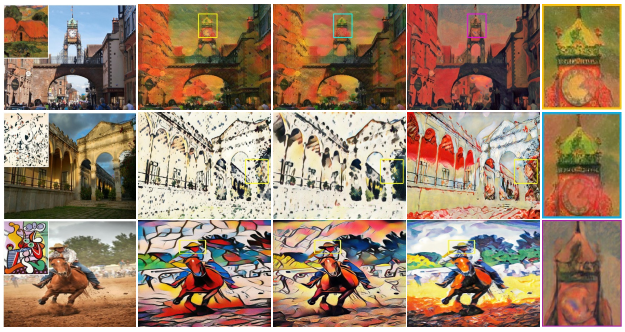
Figure 5. Ablation study of our DRB-GAN. From left to right are (a) content image, (b) style image, the results of (c) DRB-GAN, (d) w/ layer normalization layer in decoder, (e) w/ instance normalization layer in decoder, (f) w/o VGG encoder; (g) w/o $\mathcal{L}_{cls}$; (h) w/o $\mathcal{L}_{adv}$; (i) w/ AdaIN ResBlk, instead of our Dynamic ResBlk, and (j) are the zoom-in details from three regions marked in the top row, respectively.

Table 2. Quantitative comparison of different methods. SD stands for style distance metric; DS represents deception score.

| Setting | Arbitrary Style (SD↓) | Collection style (DS↑) | | | |
|---|---|---|---|---|---|
| | | K=2 | 5 | 10 | 20 |
| AdaIN | 263.4 | 0.066 | 0.045 | 0.013 | 0.011 |
| MetaNet | 271.8 | 0.032 | 0.026 | 0.023 | 0.020 |
| DRB-GAN | **241.2** | **0.576** | **0.580** | **0.581** | **0.583** |

(a) (b) (c) (d) (e)

Figure 6. Performance comparison in terms of collection style transfer. (a) Style collection, (b) and (c) are the result of our DRB-GAN, and (d) and (e) are the results of AST.



(a) (b) (c) (d) (e)

Figure 7. Performance comparison in terms of collection discriminator. (a) content/style images, (b) DRB-GAN w/ collection discriminator, (c) DRB-GAN w/ conditional discriminator, (d) CST and (e) are the zoom-in details from three regions marked in the top row, respectively.

### 4.2. Collection Style Transfer

We denote our DRB-GAN model for collection style transfer as DRB-GAN@K, where $K$ is the number of painting images from the same artist used to create the averaged transformer network. We test a range of $K$ values, including 2, 5, 10, and 20. Especially, DRB-GAN is equivalent to DRB-GAN@1 for arbitrary style transfer. In Table 2, we demonstrate the deception score obtained with these different $K$ values. A larger $K$ leads to more improvements, and the largest performance boost comes from $K = 2$ to $K = 5$. The stylizations of DRB-GAN@20 is provided in Figure 6. As we can see, in comparison to the AST, the results of DRB-GAN@20 are better with sharper details and reflect the dominant clue of the artistic style. This also can be observed from Figure 8 (f) and (j).

### 4.3. Ablation Study

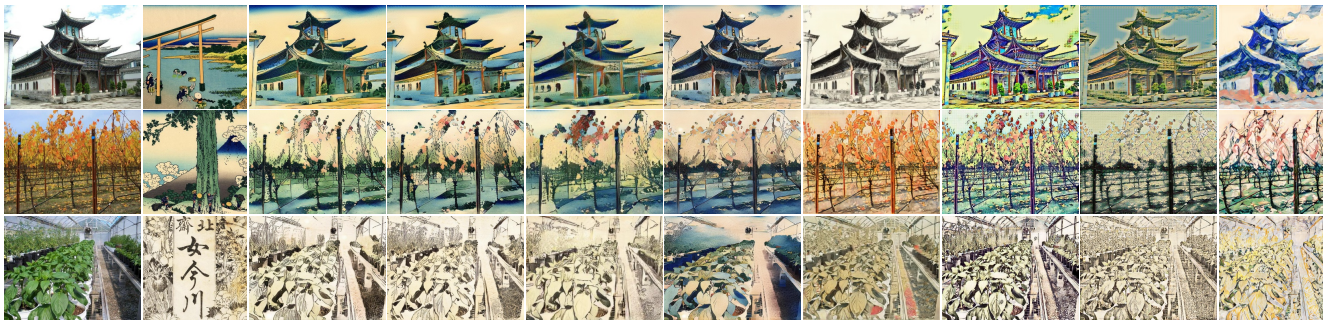
On arbitrary style transfer task, we perform the ablation study by removing or replacing each part of the network to validate the effectiveness of the proposed network and summarize the results in Figure 5 and Table 3.

As we can see, without using the layer-instance normalization function in the decoder, the model either produce degraded stylizations when the layer normalization function is used in the decoder (see Figure 5 (d)), or create artifacts in the stylizations when using instance normalization (see Figure 5 (e)). The results validate our claims that instance normalization normalizes each feature map separately, thereby potentially destroying the information in feature maps. Note that layer normalization operation normalizes the feature map together, thereby potentially demolishing each feature map as the representation of the style. Training without VGG encoder opts to capture the dominant style clues (see Figure 5 (f)) without subtle details.

Table 3. Quantitative analysis of ablation study. DS: deception score, CS: content score, and SS: style score.

| Method | DS↑ | CS↑ | SS↑ |
|---|---|---|---|
| DRB-GAN w/ IN Decoder | 0.561 | 42.1% | 0.007 |
| DRB-GAN w/ LN Decoder | 0.568 | 54.7% | 0.102 |
| DRB-GAN w/ AdaIN ResBlk | 0.552 | 36.2% | 0.006 |
| DRB-GAN w/o VGG Encoder | 0.570 | 66.3% | 0.228 |
| DRB-GAN w/o $\mathcal{L}_{cls}$ | 0.571 | 70.2% | 0.273 |
| DRB-GAN w/o $\mathcal{L}_{adv}$ | 0.542 | 32.2% | 0.001 |
| DRB-GAN | **0.573** | **72.2%** | **0.383** |

Moreover, we observe that training without the attention recalibration module ($D_{cls}$) causes slight degradation on stroke size variations. One possible reason is that the attention recalibration module works to shorten the gap between the specific style pattern of one individual style image and the overall style clues of the domain (see Figure 5



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 8. Performance comparison on stylized results from different models trained on images with different resolutions. From left to right are (a) content image, (b) style image, the results of (c) DRB-GAN×768, (d) DRB-GAN×512, (e) DRB-GAN×256, (f) DRB-GAN@20×512, (g) CycleGAN×256, (h) MetaNet×512, (i) StyleBank×512, and (j) AST×512, respectively. The number in blue indicates the size of the smaller edge of the original image.

(a) Content (b) DaVinci (c) (d) (e) Chikanobu (f) (g) (h) Seurat (i) (j)

Figure 9. Qualitative evaluation of our method for previously unseen styles. It can be observed that the generated images are consistent with the provided target style, (c),(f),(i), showing the good generalization capabilities of the approach. And it can also be observed that our model shows good performance on collection style transfer, (d),(g),(j).

(d)). Adversarial training is essential to improve the visual quality of the generated images (see Figure 5 (e)). We successfully combine adversarial and perceptual supervision to obtain high-quality style transfer. Finally, we demonstrate the advantage of our Dynamic Resblk over the AdaIN Resblk (see Figure 5 (f)).

**Effect of collection discriminator.** In Figure 7, we demonstrate the effect of our collection discriminator. Comparing to conditional GAN whose discriminator takes a category label as an additional input, our DRB-GAN produces better stylized images. However, the CST fails to maintain the style consistency between the synthesized images and target style images.

### 4.4. Discussions

**Influence of image resolution**. Our DRB-GAN is efficient in high-resolution image style transfer. It is also robust to perform style transfer on images with different resolutions. To illustrate the influence of image resolutions, we show the qualitative comparison in Figure 8. It demonstrates that our model creates consistent stylizations on different image resolutions with slight variations. The results are much better than other baseline models on the same image resolution.

**Effectiveness of unseen styles**. We apply our DRB-GAN to handle the unseen styles for both arbitrary style transfer and collection style transfer tasks. The visualization results are provided in Figure 9. Apparently, these results strongly demonstrate the robustness of our proposed method.

## 5. Conclusion

We have presented the DRB-GAN for artistic style transfer. In our model, "*style codes*" is modeled as the shared parameters, for Dynamic ResBlocks connecting both the style encoding network and the style transfer network to shrink the gap between arbitrary style transfer and collection style transfer in one single model. The proposed attention mechanism and discriminative network make full use of style information in target style images and thus encourage our model's ability for artistic style transfer. Extensive experimental results clearly demonstrate the remarkable performance of our proposed DRB-GAN model in generating synthetic style images with better quality than the state-of-the-art.

## References


[1] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, and Qi Ju. Improving image captioning with conditional generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8142–8150, 2019. 2

[2] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2

[3] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 4

[4] Wonwoong Cho, Sungha Choi, David Keetae Park, Inkyu Shin, and Jaegul Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3

[6] Bin Ding, Chengjiang Long, Ling Zhang, and Chunxia Xiao. Argan: Attentive recurrent generative adversarial network for shadow detection and removal. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[7] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao. Dual graph convolutional networks with transformer and curriculum learning for image captioning. In *Proceedings of the ACM International Conference on Multimedia*, 2021. 2

[8] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 3

[9] Yuanbin Fu, Jiayi Ma, Lin Ma, and Xiaojie Guo. Edit: Exemplar-domain aware image-to-image translation. *arXiv preprint arXiv:1911.10520*, 2019. 5

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2, 5

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 3

[14] Tao Hu, Chengjiang Long, and Chunxia Xiao. A novel visual representation on text using diverse conditional gan for visual recognition. *IEEE Transactions on Image Processing*, 30:3499–3512, 2021. 2

[15] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *IEEE International Conference on Computer Vision*. IEEE, 2013. 2

[16] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active visual recognition from crowds: A distributed ensemble approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):582–594, 2018. 2

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 3, 4, 5

[18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[19] Ashraful Islam, Chengjiang Long, Arslan Basharat, and Anthony Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[20] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. Dynamic instance normalization for arbitrary style transfer. In *AAAI Conference on Artificial Intelligence*, 2020. 1, 2, 3

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[22] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 5

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 4

[25] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019. 2

[26] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[27] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 6

[28] Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Bjorn Ommer. A content transformation block for image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 5

[29] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, 2016. 2

[30] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[31] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, 2017. 3

[32] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[33] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 5

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014. 3

[35] Alexander Liu, Yen-Chen Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. *arXiv preprint arXiv:1809.01361*, 2018. 3

[36] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 3

[37] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2

[38] Chengjiang Long and Gang Hua. Correlational gaussian processes for cross-domain visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[39] Chengjiang Long, Gang Hua, and Ashish Kapoor. A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing. *International Journal of Computer Vision*, 116(2):136–160, 2016. 2

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[41] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the European Conference on Computer Vision*, 2018. 1, 2, 3, 5, 6

[42] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 5

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[44] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. 2020. 2, 3, 5

[45] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. 2

[46] Jinjiang Wei, Chengjiang Long, Hua Zou, and Chunxia Xiao. Shadow inpainting and removal using generative adversarial networks with slice convolutions. *Computer Graphics Forum*, 38(7):381–392, 2019. 2

[47] Wenju Xu, Shawn Keshmiri, and Guanghui Wang. Toward learning a unified many-to-many mapping for diverse image translation. *Pattern Recognition*, 93:570 – 580, 2019. 2

[48] Wenju Xu, Keshmiri Shawn, and Guanghui Wang. Adversarially approximated autoencoder for image generation and manipulation. *IEEE Transactions on Multimedia*, 2019. 2

[49] Zheng Xu, Michael Wilber, Chen Fang, Aaron Hertzmann, and Hailin Jin. Learning from multi-domain artistic images for arbitrary style transfer. *arXiv preprint arXiv:1805.09987*, 2018. 2

[50] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017. 2

[51] Ling Zhang, Chengjiang Long, Qingan Yan, Xiaolong Zhang, and Chunxia Xiao. Cla-gan: A context and lightness aware generative adversarial network for shadow removal. *Computer Graphics Forum*, 39(7), 2020. 2

[52] Ling Zhang, Chengjiang Long, Xiaolong Zhang, and Chunxia Xiao. Ris-gan: Explore residual and illumination with generative adversarial networks for shadow removal. In *AAAI Conference on Artificial Intelligence*, 2020. 2

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2, 3

[54] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 5

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 1, 3, 5

[56] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017. 2