Image-to-Image Translation via Group-wise Deep Whitening and Coloring Transformation

Wonwoong Cho[1] Sungha Choi[1,2] David Park[1] Inkyu Shin[3] Jaegul Choo[1]

[1]Korea University [2]LG Electronics [3]Hanyang University

Abstract

Unsupervised image translation is an active area powered by the advanced generative adversarial networks. Recently introduced models, such as DRIT or MUNIT, utilize a separate encoder in extracting the content and the style of image to successfully incorporate the multimodal nature of image translation. The existing methods, however, overlooks the role that the correlation between feature pairs plays in the overall style. The correlation between feature pairs on top of the mean and the variance of features, are important statistics that define the style of an image. In this regard, we propose an end-to-end framework tailored for image translation that leverages the covariance statistics by whitening the content of an input image followed by coloring to match the covariance statistics with an exemplar. The proposed group-wise deep whitening and coloring (GDWTC) algorithm is motivated by an earlier work of whitening and coloring transformation (WTC), but is augmented to be trained in an end-to-end manner, and with largely reduced computation costs. Our extensive qualitative and quantitative experiments demonstrate that the proposed GDWTC is fast, both in training and inference, and highly effective in reflecting the style of an exemplar.

1. Introduction

Since the introduction of image translation [16], it has gained significant attention from relevant fields, and constantly evolved into a more generic form propelled by the seminal generative adversarial networks [10]. The primary goal of image translation [16, 37] is to convert certain attributes of an input image in an original domain to a target domain, while maintaining a particular semantics. In its initial form [16], image translation was regarded as a demanding task in which the training data should consist of paired images for the purpose of a direct supervision. CycleGAN [37] successfully extends it toward the unsupervised image translation [25, 2, 4, 37] by introducing the cycle consistency loss, which allowed the model to learn the distinctive difference in the semantics from collection of two image domains and translate the corresponding style without a direct supervision. CycleGAN links an input image to a single output, but a limitation is that it overlooks the one-to-many nature of image translation. Image translation is multimodal, in other words, there are infinite possible answers for a single given input, such as in the translation between artistic style and real photo, or male and female.

Subsequently, two notable methods, DRIT [21] and MUNIT [14], have been proposed to address the multimodal nature of unsupervised image translation. They demonstrate that a slew of potential outputs could be generated given a single input image, based on either a random sampling process in the midst of translation or utilizing a target image as an exemplar for a detailed guidance toward a desired style.

They both have two separate encoders corresponding to the content and style image, and merge the content feature and style feature together before producing the final output. Concretely speaking, DRIT concatenates the encoded content and style feature vectors, while MUNIT exploits the adaptive instance normalization (AdaIN), a method first introduced in the context of style transfer. AdaIN matches two channel-wise statistics, the mean and variance, of the encoded content feature with the style feature, which was proven to perform well in image translation.

However, we observed that the style features of an exemplar computed by the two methods do not reflect the target style well enough, ending up with unnatural looking image outputs on numerous occasions, as our experiments demonstrate in Section 4. We hypothesize that the reason lies in the way in which the methods extract style from the exemplar. For instance, the way style feature is applied to the content in MUNIT, may fail to match the covariance of the style to that of the content. It is agreed by extensive studies [8, 7, 22] that the correlation represented as the Gram matrix or the covariance matrix has an outstanding capacity as a storage of the style. To fully exploit the style of an exemplar, we propose a novel method that takes into account correlations between feature channels, in the context of image translation.

Our model is mainly motivated by whitening and coloring transformation (WCT) [23], which utilizes the feature covariance space to encode the style of an image. To elaborate, whitening refers to the process to make every correlation between features to be zero. This has the effects of removing the style, because the covariance, as well as the mean and the variance of features, define the style of an image. On the other hands, coloring indicates the procedure of matching the covariance of the style to that of the content feature. Thus in high-level view, whitening erases the style and coloring wears the style of an exemplar. WCT demonstrates that the whitening and coloring transformation is more effective in capturing a high-level representation of the style than AdaIN [23].

The problem regarding WCT when applied to image translation is that the time complexity of solving the intricate whitening and coloring computation is as expensive as $O(n^3)$ with respect to a matrix in $\mathcal{R}^{n\times n}$. In addition, computing the backpropagation with respect to singular value decomposition (SVD) is non trivial [32, 15]. To settle the problems, we propose a novel deep whitening and coloring transformation (DWCT) which allows the existing method by a simple approximation based on deep neural networks. We further extend our method into group-wise deep whitening and coloring (GDWCT), which not only reduce the number of parameters, but also boost the performance [34, 12].

The main contribution of this paper includes:

- We introduce the novel deep whitening-and-coloring algorithm which allows an end-to-end training in image translation for rendering profound style semantics.
- We introduce the group-wise deep whitening-and-coloring algorithm to further increase the computation speed up to a simple forward propagation, and achieve highly competitive image quality.
- We demonstrate the effectiveness of our method via extensive experiments, in both qualitative and quantitative manners, with compared to state-of-the-art baselines.

2. Related Work

Image-to-image translation. Image-to-image translation aims at converting an image to a target domain. There are many sub-applications such as colorization [36, 5, 1], super-resolution [6, 20], domain adaptation [11], style transfer [8] and photo-realistic image synthesis [35, 24].

A slew of research have been conducted in the context of unsupervised image translation task [37, 18, 25]. These studies commonly involve methods to learn joint distribution between two different image domains without any pair of corresponding images, so that the model can conduct a translation without a direct supervision. StarGAN [4] proposes a single unified model which can deal with unsupervised image translation among multiple different domains. A few of studies [9, 38] focus on the limitation of earlier work in which they produce a single output given an input without consideration that diverse images can also be generated within the same target domain. However, they are not without limitations, either by generating limited number of outputs [9] or requiring paired images [38].

Recent studies [14, 21] suggest new methods capable of generating multimodal outputs in an unsupervised manner. These studies are conducted based on the presumption that a latent image space could be separated into a domain-specific style space and a domain-invariant content space. Following the precedents, we also adopt the separate encoders to extract out each of the content and style features.

Style transfer. The seminal studies by Gatys et al. [7, 8] show that the correlation among the feature channels obtained from deep neural networks successfully captures the image style. The correlation is captured by the gram-matrix or covariance matrix. It is used for transferring a style from a style image to a content image by matching the statistics of the style feature to those of the content. However, a drawback is that they require a time-consuming, iterative optimization process during the inference time involving multiple forward and backward passes to obtain a final result. To address the limitation of a slow optimization process, alternative methods [29, 3, 17] have achieved a superior time-efficiency through a feed-forward network approximating an optimal result of the iterative methods. However, the feed-forward based models are incapable of transferring an unseen style from an arbitrary image.

To alleviate the limitation, several approaches have been suggested to enable an arbitrary neural style transfer, which can transfer an unseen style [13, 23, 24]. AdaIN [13] directly computes the affine parameters from the style feature and aligns the mean and variance of the content feature with those of the style feature. WCT [23] encodes the style into the feature covariance matrix, so that it effectively captures a high-level representation of the style.

3. Proposed Method

In this section, we describe our proposed model in detail. We first provide a broad overview of the model and explain the details. The exact loss functions are formulated and illustrated in Section 3.2.

3.1. Model Setup

Let $x_{\mathcal{A}} \in \mathcal{X}_{\mathcal{A}}$ and $x_{\mathcal{B}} \in \mathcal{X}_{\mathcal{B}}$ denote images from two different image domains, $\mathcal{X}_{\mathcal{A}}$, $\mathcal{X}_{\mathcal{B}}$, respectively. Inspired by MUNIT [14] and DRIT [21], we assume that the image x

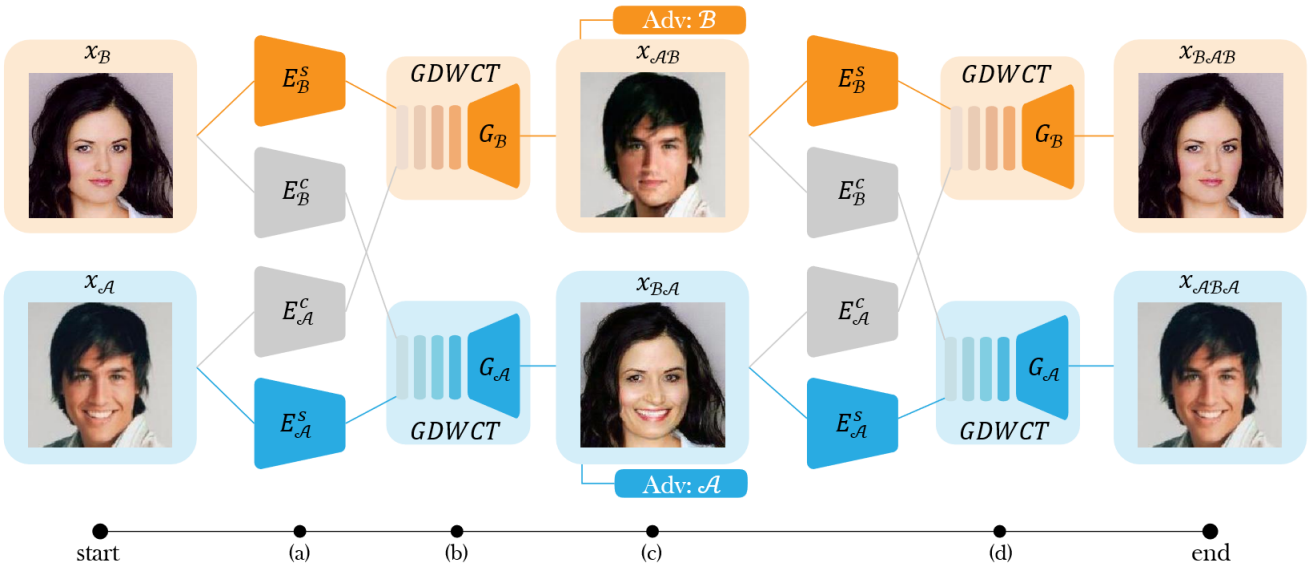

Figure 1: Overview of our model. (a) To translate from $\mathcal{A} \rightarrow \mathcal{B}$, First we extract the content feature $c_{\mathcal{A}}$ from the image $x_{\mathcal{A}}$ (i.e., $c_{\mathcal{A}} = E^c_{\mathcal{A}}(x_{\mathcal{A}})$) and the style feature $s_{\mathcal{B}}$ from the image $x_{\mathcal{B}}$ (i.e., $s_{\mathcal{B}} = E^s_{\mathcal{B}}(x_{\mathcal{B}})$). (b) The obtained features are combined in GDWCT module while forwarding through the generator $G_{\mathcal{B}}$. (c) The discriminator $D_{\mathcal{B}}$ classifies whether the input $x_{\mathcal{AB}}$ is a real image of domain $\mathcal{B}$ or not. (d) Similar to the procedures from (a) to (c), the generator $G_{\mathcal{B}}$ generates the reconstructed image $x_{\mathcal{BAB}}$ by combining the content feature $c_{\mathcal{BA}}$ and the style feature $s_{\mathcal{AB}}$.

can be decomposed into the domain-invariant content space $\mathcal{C}$ and the domain-specific style spaces $\{\mathcal{S}_{\mathcal{A}}, \mathcal{S}_{\mathcal{B}}\}$, i.e.,

$$\begin{aligned} \{c_{\mathcal{A}}, s_{\mathcal{A}}\} &= \{E^c_{\mathcal{A}}(x_{\mathcal{A}}), E^s_{\mathcal{A}}(x_{\mathcal{A}})\} \qquad c_{\mathcal{A}} \in \mathcal{C}, s_{\mathcal{A}} \in \mathcal{S}_{\mathcal{A}} \\ \{c_{\mathcal{B}}, s_{\mathcal{B}}\} &= \{E^c_{\mathcal{B}}(x_{\mathcal{B}}), E^s_{\mathcal{B}}(x_{\mathcal{B}})\} \qquad c_{\mathcal{B}} \in \mathcal{C}, s_{\mathcal{B}} \in \mathcal{S}_{\mathcal{B}}, \end{aligned} \tag{1}$$

where $\{E^c_{\mathcal{A}}, E^c_{\mathcal{B}}\}$ and $\{E^s_{\mathcal{A}}, E^s_{\mathcal{B}}\}$ are content and style encoders for each domain, respectively. Our objective is to generate the translated image by optimizing the functions $\{f_{\mathcal{A}\rightarrow\mathcal{B}}, f_{\mathcal{B}\rightarrow\mathcal{A}}\}$ of which $f_{\mathcal{A}\rightarrow\mathcal{B}}$ maps the data point $x_{\mathcal{A}}$ in the original domain $\mathcal{X}_{\mathcal{A}}$ to the point $x_{\mathcal{A}\rightarrow\mathcal{B}}$ in the target domain $\mathcal{X}_{\mathcal{B}}$, reflecting a given reference $x_{\mathcal{B}}$, i.e.,

$$\begin{aligned} x_{\mathcal{A}\rightarrow\mathcal{B}} &= f_{\mathcal{A}\rightarrow\mathcal{B}}(x_{\mathcal{A}}, x_{\mathcal{B}}) = G_{\mathcal{B}}(E^c_{\mathcal{A}}(x_{\mathcal{A}}), E^s_{\mathcal{B}}(x_{\mathcal{B}})) \\ x_{\mathcal{B}\rightarrow\mathcal{A}} &= f_{\mathcal{B}\rightarrow\mathcal{A}}(x_{\mathcal{B}}, x_{\mathcal{A}}) = G_{\mathcal{A}}(E^c_{\mathcal{B}}(x_{\mathcal{B}}), E^s_{\mathcal{A}}(x_{\mathcal{A}})), \end{aligned} \tag{2}$$

where $\{G_{\mathcal{A}}, G_{\mathcal{B}}\}$ are the generators for the corresponding domains. The overview of our model is illustrated in Fig. 1.

Our proposed method, the group-wise deep whitening and coloring transformation (GDWCT) plays a main role of applying the style feature s to the content feature c inside the generator G. Concretely, GDWCT takes the content feature $c_{\mathcal{A}}$, the matrix for coloring transformation $S_{\mathcal{B}}$, and the mean of the style $\mu^s_{\mathcal{B}}$ as input, and conduct a translation of $c_{\mathcal{A}}$ to $c_{\mathcal{A}\rightarrow\mathcal{B}}$, formulated as

$$c_{\mathcal{A}\rightarrow\mathcal{B}} = \text{GDWCT}(c_{\mathcal{A}}, S_{\mathcal{B}}, \mu^s_{\mathcal{B}}), \tag{3}$$

where $S_{\mathcal{B}} = \text{MLP}^{\text{CT}}_{\mathcal{B}}(s_{\mathcal{B}})$ and $\mu^s_{\mathcal{B}} = \text{MLP}^{\mu}_{\mathcal{B}}(s_{\mathcal{B}})$. Note that MLP denotes a multi-layer perceptron composed of several linear layers with a non-linear activation after each layer. Additionally, we set a learnable parameter α such that the networks can decide how much of the style to apply considering that the amount of style the networks require may vary, i.e., $c_{\mathcal{A}\rightarrow\mathcal{B}} = \alpha(\text{GDWCT}(c_{\mathcal{A}}, S_{\mathcal{B}}, \mu^s_{\mathcal{B}})) + (1-\alpha)c_{\mathcal{A}}$.

However, the different layers of a model focus on different information (e.g., the low-level feature captures a fine pattern, whereas the high level feature captures a complicated pattern appeared in a wide area). We thus add our GDWCT module in each residual block R_i of the generator $G_{\mathcal{B}}$ as shown in Fig. 2. By injecting the style information across multiple hops via a sequence of GDWCT module, the model is set to reflect both the fine- and coarse-level style information simultaneously.

3.2. Loss

Following MUNIT [14] and DRIT [21], we adopt both the latent-level reconstruction loss and the pixel-level reconstruction loss. First, we set the style consistency loss between two style features $(s_{\mathcal{A}\rightarrow\mathcal{B}}, s_{\mathcal{B}})$, so that it encourages the model to reflect the style of the reference image $s_{\mathcal{B}}$ to the translated image $x_{\mathcal{A}\rightarrow\mathcal{B}}$.

$$\mathcal{L}^{\mathcal{A}\rightarrow\mathcal{B}}_s = \mathbb{E}_{x_{\mathcal{A}\rightarrow\mathcal{B}}, x_{\mathcal{B}}} \left[\|E^s_{\mathcal{B}}(x_{\mathcal{A}\rightarrow\mathcal{B}}) - E^s_{\mathcal{B}}(x_{\mathcal{B}})\|_1\right] \tag{4}$$

Second, we set the content consistency loss between two content features $(c_{\mathcal{A}}, c_{\mathcal{A}\rightarrow\mathcal{B}})$ to enforce the model to maintain the content feature of the input image $c_{\mathcal{A}}$ after being translated $c_{\mathcal{A}\rightarrow\mathcal{B}}$.

$$\mathcal{L}^{\mathcal{A}\rightarrow\mathcal{B}}_c = \mathbb{E}_{x_{\mathcal{A}\rightarrow\mathcal{B}}, x_{\mathcal{A}}} [\|E^c_{\mathcal{B}}(x_{\mathcal{A}\rightarrow\mathcal{B}}) - E^c_{\mathcal{A}}(x_{\mathcal{A}})\|_1] \tag{5}$$

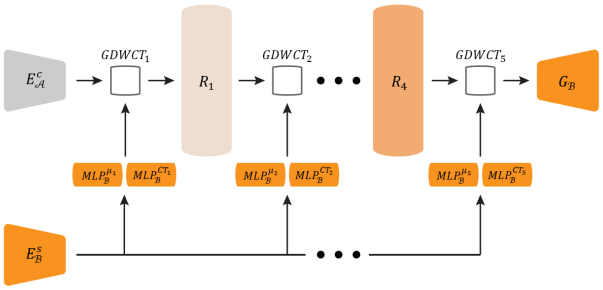

Figure 2: Overall process of the proposed GDWCT module. We apply the style via multiple hops for applying style from the low-level feature to the high-level feature.

Third, because style and content consistency loss do not guarantee the performance of image reconstruction, we adopt the cycle consistency loss and identity loss [37] to obtain a high quality image. i.e.,

$$\mathcal{L}_{cyc}^{\mathcal{A}\to\mathcal{B}\to\mathcal{A}} = \mathbb{E}_{x_{\mathcal{A}}}\left[\|x_{\mathcal{A}\to\mathcal{B}\to\mathcal{A}} - x_{\mathcal{A}}\|_1\right] \quad (6)$$

$$\mathcal{L}_{i}^{\mathcal{A}\to\mathcal{A}} = \mathbb{E}_{x_{\mathcal{A}}}\left[\|x_{\mathcal{A}\to\mathcal{A}} - x_{\mathcal{A}}\|_1\right]. \quad (7)$$

Lastly, we use an adversarial loss for minimizing the distance between the distribution of the real image and the generated image. In concrete, we employ LSGAN [27] as the adversarial method. i.e.,

$$\mathcal{L}_{D_{adv}}^{\mathcal{B}} = \frac{1}{2}\mathbb{E}_{x_{\mathcal{B}}}[(D(x_{\mathcal{B}})-1)^2] + \frac{1}{2}\mathbb{E}_{x_{\mathcal{A}\to\mathcal{B}}}[(D(x_{\mathcal{A}\to\mathcal{B}}))^2] \quad (8)$$

$$\mathcal{L}_{G_{adv}}^{\mathcal{B}} = \frac{1}{2}\mathbb{E}_{x_{\mathcal{A}\to\mathcal{B}}}[(D(x_{\mathcal{A}\to\mathcal{B}})-1)^2] \quad (9)$$

Note that there exists the opposite translation. That is, similar to DRIT [21], the model is trained on both ways, $(\mathcal{A}\to\mathcal{B}\to\mathcal{A})$, $(\mathcal{B}\to\mathcal{A}\to\mathcal{B})$ at the same time. Finally, our full loss is as follows:

$$\begin{aligned}\mathcal{L}_D &= \mathcal{L}_{D_{adv}}^{\mathcal{A}} + \mathcal{L}_{D_{adv}}^{\mathcal{B}} \\ \mathcal{L}_G &= \mathcal{L}_{G_{adv}}^{\mathcal{A}} + \mathcal{L}_{G_{adv}}^{\mathcal{B}} + \lambda_{latent}(\mathcal{L}_s + \mathcal{L}_c) + \\ &\quad \lambda_{pixel}(\mathcal{L}_{cyc} + \mathcal{L}_i^{\mathcal{A}\to\mathcal{A}} + \mathcal{L}_i^{\mathcal{B}\to\mathcal{B}})\end{aligned} \quad (10)$$

where $\mathcal{L}$ without a domain notation implies both ways and λ is the hyperparameter such that we empirically set $\lambda_{latent} = 1$, $\lambda_{pixel} = 10$.

3.3. Group-wise Deep Whitening and Coloring Transformation

For concise expression, we omit the domain notation unless needed, such as $c = \{c_{\mathcal{A}}, c_{\mathcal{B}}\}$, $s = \{s_{\mathcal{A}}, s_{\mathcal{B}}\}$, etc.

Whitening transformation (WT). WT is a linear transformation that transforms a covariance matrix of a given input into the identity matrix. Specifically, we first subtract the content feature $c \in \mathcal{R}^{C\times BHW}$ by its mean μ_c, where (C, B, H, W) represent the number of channels, batch size, height, and width, respectively. We then compute the outer product of $\bar{c}$ along the BHW dimension, where $\bar{c} = c - \mu_c$. Lastly, we factorize the covariance matrix $\bar{c}\bar{c}^T \in \mathcal{R}^{C\times C}$ via eigendecomposition. i.e.,

$$\bar{c}\bar{c}^T = \frac{1}{N-1}\Sigma_{i=1}^{N}(c_i-\mu_c)(c_i-\mu_c)^T = Q_c\Lambda_c Q_c^T, \quad (11)$$

where N denotes (BHW), $Q_c \in \mathcal{R}^{C\times C}$ is the orthogonal matrix containing the eigenvectors and $\Lambda_c \in \mathcal{R}^{C\times C}$ indicates the diagonal matrix whose each diagonal element is the eigenvalue corresponding to each column vector of Q_c. The whitening transformation is defined as follows:

$$c_w = Q_c\Lambda_c^{-\frac{1}{2}}Q_c^T\bar{c}, \quad (12)$$

where c_w denotes the whitened feature. In other words, the content feature $\bar{c}$ is rotated by Q_c^T, scaled by $\Lambda_c^{-\frac{1}{2}}$ and rotated back to the original basis.

However, as pointed out in Section 1, eigendecomposition is not only slow to compute, but also tricky to back-propagate the gradient signal. To alleviate the problem, we introduce the deep whitening transformation (DWT) approach such that the learned content encoder E_c^* can naturally encode the whitened feature c_w. i.e., $c_w = c^* - \mu_{c^*}$, where $E_c^*(x_c) = c^*$. To this end, we introduce a novel regularization term that makes the covariance matrix $\bar{c}\bar{c}^T$ as close as possible to the identity matrix. i.e.,

$$\mathcal{R}_w = \mathbb{E}_{\bar{c}}\left[\|\bar{c}\bar{c}^T - I\|_1\right]. \quad (13)$$

Note that if the loss from Eq. 13 becomes zero, it can be easily derived that $c_w = c^* - \mu_{c^*}$. That is, (1) from Eq. 13, $\bar{c}\bar{c}^T = I$, then from Eq. 11, $Q_c\Lambda_c Q_c^T = I$, (2) where Q_c can be an arbitrary orthogonal matrix, but Λ_c has a unique solution of the identity matrix I, so that (3) from Eq. 12, $c_w = Q_c\Lambda_c^{-\frac{1}{2}}Q_c^T(c^*-\mu_c^*) = I(c^*-\mu_c^*)$, thus the whitening transformation in Eq. 12 being reduced to

$$c_w = c - \mu_c, \quad (14)$$

However, there are several limitations in DWT. First of all, estimating a full covariance matrix with a given data in small batch is inaccurate owing to a lack of sample numbers [12]. Second, conducting DWT with respect to the entire channels may lose more information than the standardization, so that the identity of the content feature can be lost. We therefore improve DWT by grouping channels and applying DWT to the individual group.

Concretely, the channel dimension of $\bar{c}$ is re-arranged at a group level, i.e., $\bar{c} \in \mathcal{R}^{G\times C/G\times BHW}$, where G is the number of groups. After obtaining the covariance matrix $\bar{c}\bar{c}^T$ in $\mathcal{R}^{G\times C/G\times C/G}$, we apply Eq. 13 along its group dimension. Note that during the forwarding phase, group-wise DWT (**GDWT**) is the same with DWT as shown in Fig. 3(a).

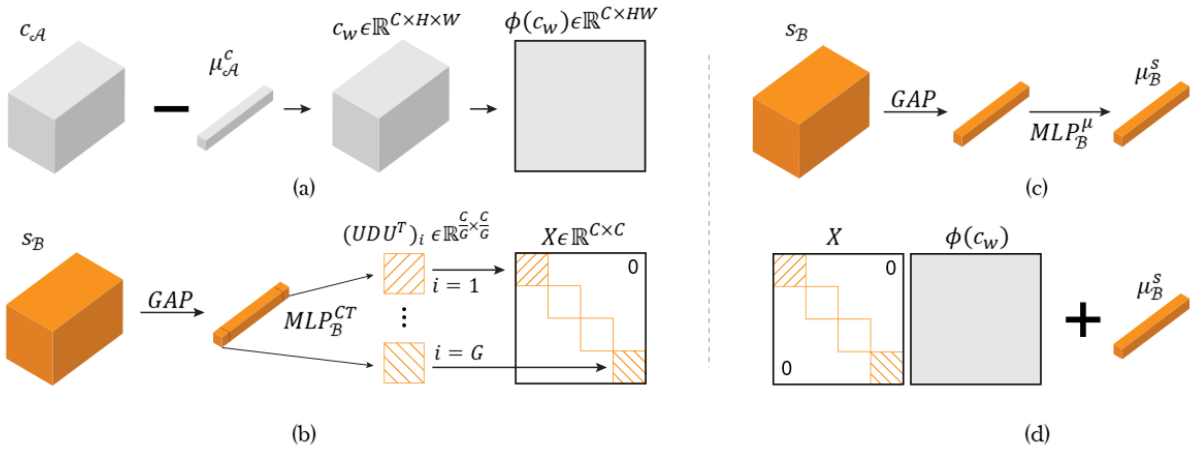

Figure 3: Details on the proposed GDWCT module. (a) For obtaining the whitened feature, we subtract the mean of the content feature $\mu^c_{\mathcal{A}}$ from $c_{\mathcal{A}}$. (b) The procedure of approximating coloring transformation matrix. Details are described in Section 3.3. (c) We obtain the mean of the style feature $\mu^s_{\mathcal{B}}$ by forwarding into the MLP layer $MLP^{\mu}_{\mathcal{B}}$. (d) As the final form of our method, we first compute the matrix multiplication between the coloring transformation matrix GDCT and the whitened feature c_w. We then add it to the mean of the style $\mu^s_{\mathcal{B}}$.

Coloring transformation (CT). CT matches the covariance matrix of the whitened feature $c_w c_w^T$ to that of the style feature $\bar{s}\bar{s}^T$, where $\bar{s} = s - \mu_s$. The covariance matrix $\bar{s}\bar{s}^T$ is then decomposed into $Q_s\Lambda_s Q_s^T$, used for the subsequent coloring transformation. This process is written as

$$c_{cw} = Q_s\Lambda_s^{\frac{1}{2}}Q_s^T c_w, \tag{15}$$

where c_{cw} denotes the colored feature.

For the same reason as WT, we also replace CT to a simple but effective method that we call a deep coloring transformation (DCT). Specifically, we obtain the matrix S through $\text{MLP}^{\text{CT}}(s)$, where $s = E_s(x)$. we then decompose S into two matrices by computing its column-wise L_2 norm, i.e., $S = UD$, where $U \in \mathcal{R}^{C\times C}$ is the square matrix of which the i-th column vector u_i is the unit vector and D is the diagonal matrix in $\mathcal{R}^{C\times C}$ whose diagonal entries are L_2 norm of each u_i. Note that we assume that our optimal MLP^{CT*} first encodes two matrices in Eq. 15. i.e., $\text{MLP}^{\text{CT}*}(s) = U^*D^* = Q_s\Lambda_s^{\frac{1}{2}}$.

In order to properly work as Q_s and $\Lambda_s^{\frac{1}{2}}$, U needs to an orthogonal matrix, and the matrix D should be positive definite. To assure the conditions, we set a regularization for U to encourage the column vectors of U to be orthogonal, written as

$$\mathcal{R}_c = \mathbb{E}_s\ [\|U^TU - I\|_1]. \tag{16}$$

Because the diagonal matrix D has its diagonal elements as the column-wise L_2 norm of S, it is already positive definite. Meanwhile, U^* becomes the orthogonal matrix if U accomplishes the orthogonality, because each column vector u_i of U has a unit L_2 norm. That is, U^*D^* satisfies the entire conditions to be $Q_s\Lambda_s^{\frac{1}{2}}$. Finally, recombining U and D, we simplify CT as

$$c_{cw} = UDU^Tc_w. \tag{17}$$

However, approximating the entire matrix S has an expensive computational cost (the number of parameters we have to predict is C^2). Therefore, we extend DCT to the group-wise DCT (**GDCT**), and reduce the number of parameters from C^2 to C^2/G. The specific steps are illustrated in Fig. 3(b). We first acquire the i-th matrix $\{UDU^T\}_i \in \mathcal{R}^{C/G\times C/G}$ for GDCT, where $i = \{1, ..., G\}$. We then build up a block diagonal matrix $X \in \mathcal{R}^{C\times C}$ by arranging the matrices $\{UDU^T\}_{1,...,G}$. Next, as shown in Fig. 3(d), we compute the matrix multiplication with X and the whitened feature c_w, so that Eq. 17 is converted into

$$c_{cw} = X\phi(c_w), \tag{18}$$

where ϕ denotes a reshaping operation $\phi : \mathcal{R}^{C\times H\times W} \rightarrow \mathcal{R}^{C\times HW}$. The regularization in Eq. 16 is applied G times in every iteration. Finally, we add the shifting parameter μ_s to the c_{cw}, where $\mu_s = \text{MLP}^{\mu}(s)$ as illustrated in Fig. 3(c). Note that we set the coefficients $\lambda_w = 0.001, \lambda_c = 10$ for the experiments.

4. Experiments

In this section, we describe in detail the baseline models and the datasets we used. The qualitative and quantitative analyses are then reported in Section 4.2 and Section 4.3, respectively. Implementation details and additional results are included in the supplementary material.

Figure 4: Quality comparisons based on the Artworks dataset [37].

4.1. Implementation Details

Datasets. We evaluate our model with a variety of datasets including CelebA [26] dataset, Artworks [37] dataset (Ukiyoe, Monet, Cezanne and Van Gogh), cat2dog dataset [21], Pen ink and Watercolor classes of the Behance Artistic Media (BAM) dataset [33], and Yosemite [37] (Summer and Winter scenes).

MUNIT & DRIT. Both MUNIT [14] and DRIT [21] disentangle the latent space by decomposing the image to the domain invariant content and domain specific style features. However, when combining the content feature with the style feature, MUNIT exploits the adaptive instance normalization [13] to match the first order statistics (mean, variance) of the content feature to those of the style feature, whereas DRIT concatenates the content and style features, and let the model learn how to combine.

WCT. In order to transfer the style into the content image, WCT [23] applies the whitening and coloring transformation to the features extracted from the encoder. WCT extracts the content and style features from the same encoder, so that the definition of the style in WCT is different from ours. In concrete, the style in our model is a task-specific style extracted from the style encoder (i.e., in attributes translation, the style is defined as the attribute to translate during training).

4.2. Quantitative results

We compare the performance of our model with the baseline models based on user preference and classification accuracy.

User Study. We first conduct an user study using the CelebA dataset [26]. The initial motivation of our user study was to measure user preferences on outputs produced by GDWCT and the baseline models with a focus on the quality of an output, and the rendering of the style given as an exemplar. Each user evaluated 60 set of image comparisons, choosing one from four candidates with 30 seconds limit per comparison. We informed the participants with the original and the target domains for every run, e.g., Male to Female, so they knew exactly which style in an exemplar is of interest. Table 1 summarizes the result. It is found that the users prefer our model to other baseline models on five out of six class pairs. In the translation of Female $\Rightarrow$ Male,

because DRIT consistently generates a facial hair in all translation, it may obtain the higher score than ours. The superior measures demonstrate that our model produces visual compelling images. Furthermore, the result indicates that our model reflect the style from the exemplar better than other baselines, which justifies that matching entire statistics including a covariance would render style more effectively.

Q. Which one do you prefer? (%)				
	MUNIT [14]	DRIT [21]	WCT [23]	GDWCT (ours)
Male ⇒ Female	4.41	42.25	10.12	**44.52**
Female ⇒ Male	7.78	**48.89**	4.44	38.89
Bang ⇒ Non Bang	3.35	42.20	3.37	**51.10**
Non Bang ⇒ Bang	6.67	18.89	4.45	**71.15**
Smiling⇒ Non Smiling	5.56	30.35	1.35	**64.44**
Non Smiling⇒ Smiling	2.30	22.25	2.25	**73.33**

Table 1: Comparisons on the user preference. Numbers indicate the percentage of preference on each class.

Classification Accuracy. A well-trained image translation model would generate outputs that are classified as an image from the target domain. For instance, when we translate a female into male, we measure the classification accuracy in the gender domain. A high accuracy indicate that the model learns deterministic patterns to be represented in the target domain. We report the classification results on translated images in Table 2. For the classification, we adopted the pretrained Inception-v3 [28], and fine-tuned on CelebA dataset. Our model records competitive average on the accuracy rate, marginally below DRIT on Gender class, and above on Bangs and Smiling.

	MUNIT	DRIT	WCT	Ours
Gender	30.10	**95.55**	28.80	92.65
Bangs	35.43	66.88	24.85	**76.05**
Smiling	45.60	78.15	32.08	**92.85**
Avg.	37.04	80.19	28.58	**87.18**

Table 2: Comparison of the classification accuracy (%) in the target domain. Tested with the image size of 216x216.

Inference Time. The superiority of GDWCT also lies in the speed at which outputs are computed in the inference stage. Table 3 shows that our model is as fast as the existing image translation methods, and has the capacity of rendering rich style information as of WCT. The numbers represent the time taken to generate one image.

Figure 5: Comparison between single hop and multi-hops of the style.

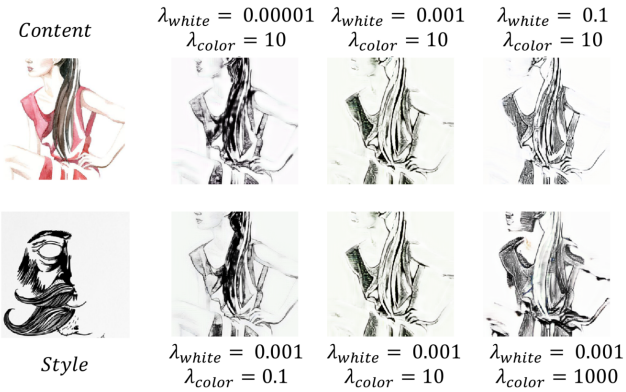

Figure 6: Regularization on whitening and coloring transformation.

Figure 7: Visualization of the effects on whitening transformation. Through the whitening step, the content image lose the original information.

	MUNIT	DRIT	WCT	Ours
Runtime (sec)	0.0419	0.0181	0.8324	0.0302

Table 3: Comparison of the inference time. Tested with the image size 256x256 on a NVIDIA Titan XP GPU, and averaged over 1000 trials.

4.3. Qualitative results

In this section, we analyze the effects of diverse hyperparameters and devices on the final image outputs.

Stylization Comparisons. We conduct qualitative analyses by a comparison with the baseline models on Fig. 4. Each row represents different classes, and the leftmost and

the second columns are content and the exemplar style, respectively. Across diverse classes, we observe consistent patterns for each baseline model. First, MUNIT tends to keep the object boundary, leaving not much room for style to get in. DRIT shows results of high contrast, and actively transfer the color. WCT is more artistic in the way it digests the given style, however at times losing the original content to a large extent. Our results transfer object colors as well as the overall mood in the style, while not overly blurring details. We provide additional results of our model in Fig. 8. We believe our work gives another dimension of an opportunity to translate image at one's discretion.

Number of Hops on Style. As we previously discussed in Fig. 2, the proposed GDWTC could be applied in multi-hops. We demonstrate the effects of the different number of hops on the style. To this end, we use the Artworks dataset (Ukiyoe) [37]. We train two identical models different only in the number of hops, a single hop (GDWTC$_1$) or multi-hops (GDWTC$_{1\sim5}$). In Fig. 5, the rightmost image (GDWTC$_{1\sim5}$) has the style that agrees with the detailed style given in the leftmost image. The third image (GDWTC$_1$) follows the overall color pattern of the exemplar, but with details less transferred. For example, the writing in the background has not been transferred to the result of GDWCT$_1$, but is clearly rendered on GDWTC$_{1\sim5}$. The difference comes from a capacity of the multiple hops on a stylization, which covers both fine and coarse style [23].

Regularization on Whitening and Coloring. We verify the influences of the regularizations $\mathcal{R}_w$ and $\mathcal{R}_c$ having on the final image output. Intuitively, a higher λ_w will strengthen the whitening transformation, erasing the style more, because it encourages the covariance matrix of the content feature to be closer to the identity matrix. Likewise, a high value of λ_c would result in a diverse level of style, since the intensity of the style applied during coloring increases as the eigenvectors of the style feature gets closer to orthogonal.

We use two classes, Watercolor and Pen Ink, of BAM [33] dataset. The images in Fig. 6 illustrates the results of (watercolor $\rightarrow$ penink). Given the leftmost content and style as input, the top row shows the effects of gradually increasing value of λ_w. A large λ_w leads the model to erase textures notably in the cloth and hair. It proves our presumption that the larger w is, the stronger the effects of the whitening is. Meanwhile, the second row shows the effects of different coloring coefficient λ_c. The cloth of the subjects shows a stark difference, gradually getting darker, applying the texture of the style more intensively.

Visualization of Whitening and Coloring. We visualize the whitened feature to visually inspect the influence of the proposed group-wise deep whitening transformation on the content image. We also use a sample from the Artworks dataset. For visualization, we forward the whitened feature into the networks without coloring transformation. The third image from the left shows the whitening effects. It is evident that in the image, detailed style regarding the color and texture are erased from the content image. Notably, the reeds around the river, and the clouds in the sky are found to be whitened in color, being ready to be stylized. On the other hand, the rightmost image stylizes given the whitened image via the group-wise deep coloring transformation. It reveals that the coloring transformation properly applies the exemplar style, which is in a simpler style with monotonous color than that of the content image.

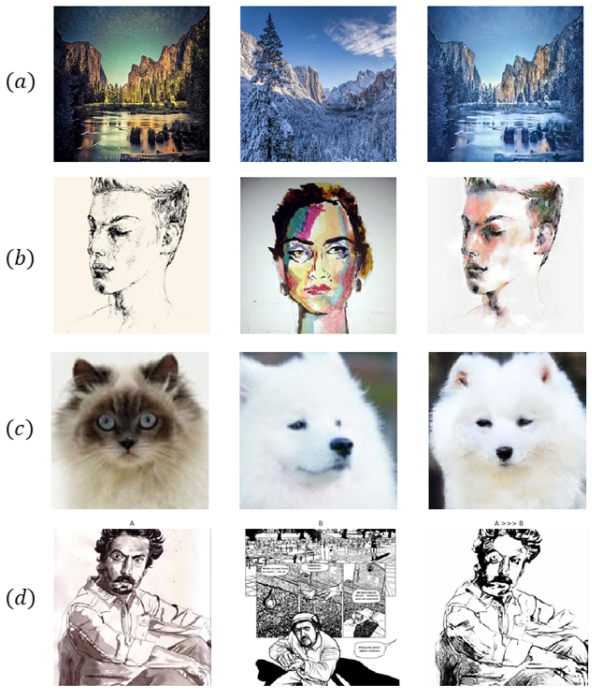

Figure 8: Results on various dataset; (a) Yosemite (b) BAM (Water Color $\Rightarrow$ Pen Ink) (c) Cat2dog (d) BAM (Pen Ink $\Rightarrow$ Water Color)

5. Conclusion

In this paper, we propose a novel framework, group-wise deep whitening and coloring transformation (GDWCT) for an improved stylization capability. Our experiments demonstrate that our work produces competitive outputs in image translation as well as style transfer domains, having a majority of real users agree that our model successfully reflects the given exemplar style. We believe this work bears the potential to enrich relevant academic fields with the novel framework and practical performances.

6. Appendix

In this section, we supplement our paper by reporting additional information. First of all, we describe the implementation details of our networks in subsection 6.1. We then qualitatively compare our model with the baseline models on CelebA dataset in subsection 6.2. Finally, we report extra results on CelebA dataset in subsection 6.3.

6.1. Implementation

Content encoder. The content encoders $\{E^c_{\mathcal{A}}, E^c_{\mathcal{B}}\}$ are composed of a few strided convolutional (conv) layers and four residual blocks. The size of the output activation is in $\mathcal{R}^{256\times\frac{H}{4}\times\frac{W}{4}}$. Note that we use the instance normalization [30] along the entire layers in E_c in order to flatten the content feature [14, 13].

Style encoder. The style encoders $\{E^s_{\mathcal{A}}, E^s_{\mathcal{B}}\}$ consist of several strided conv layers with the output size in $\mathcal{R}^{256\times\frac{H}{16}\times\frac{W}{16}}$. Lastly, after being global average pooled, the style feature s is forwarded into the MLP_{CT} and MLP_{μ}. We use the group normalization [34] in E_s to match the structure of s with MLP_{CT} by grouping the highly correlated channels in advance.

Multi layer perceptron. Each of $\{\text{MLP}^{\text{CT}}_{\mathcal{A}}, \text{MLP}^{\text{CT}}_{\mathcal{B}}\}$ and $\{\text{MLP}^{\mu}_{\mathcal{A}}, \text{MLP}^{\mu}_{\mathcal{B}}\}$ is composed of several linear layers. The input dimension of MLP_{CT} is depends on the number of group. Specifically, the partial style feature in $\mathcal{R}^{\frac{C}{G}}$ is forwarded as the input feature and the output size is the square of the input dimension. On the other hand, both of the input and output dimension of MLP_{μ} is same with the number of channels, 256.

Generator. The generators $\{G_{\mathcal{A}}, G_{\mathcal{B}}\}$ are made of four residual blocks and several sequence of upsampling layer with strided conv layer. Note that GDWCT is applied in the process of forwaring G.

Discriminator. The discriminators $\{D_{\mathcal{A}}, D_{\mathcal{B}}\}$ are in the form of multi-scale discriminators [31]. The size of the output activations are in $\mathcal{R}^{4\times4}, \mathcal{R}^{8\times8}, \mathcal{R}^{16\times16}$.

Training details. To be concrete on our training details, we use Adam optimizer [19] with $\beta_1 = 0.5, \beta_2 = 0.999$. We also set a learning rate of 0.0001 for both generators and discriminators. Other settings are different depending on the dataset. In CelebA, we apply a batch size of eight with the image size of 216 (we first resize the initial (178×218) to (216×264.5). We then conduct the center-crop by (216×216)). We also train 500,000 iterations with a decaying rate of 0.5 from 100,000 on every 50,000 iterations. On the other hand, in all other datasets, We train the model with the batch size of two and the image size of 256 (we first resize each image by 286 then we apply the random-crop by 256). Note that we set 200,000 iterations for training and apply decaying rate of 0.5 from 100,000 on every 10,000 iterations. All the experiments are trained using a single NVIDIA TITAN Xp GPU for 3 days with the group size of 8.

6.2. Additional Comparison

We compare our method with the baseline models using CelebA dataset whose chosen attributes for translation are **Gender** (Male $\leftrightarrow$ Female), **Bangs** (Bang $\leftrightarrow$ Non-Bang) and **Smiling** (Smiling $\leftrightarrow$ Non-Smiling). For the comparison, we first pre-process the entire data following subsection 6.1. Then we construct three datasets, each composed of two other domains, for the corresponding attribute (e.g., Male and Female). Lastly, we separate each dataset into a training set and a test set with a ratio of 9:1. The results are showed in Fig. 9. Two columns from the left denote a content image and a style image (exemplar) respectively while each of other columns indicates an output of each model whose name is written on the top. Each row shows a different translation case whose attributes before and after conversion are on the leftmost.

Our model shows a superior performance in overall attribute translation, because our model drastically but suitably applies the style comparing with the baseline models. For example, In case of the (male $\rightarrow$ female) translation, our model generates an image with long hair and make-up, the major patterns of the woman. However, each generated image from MUNIT and DRIT wears only light make-up with incomplete long hair. Meanwhile, in both translation cases of Smiling and Bangs, our model shows an outstanding performance comparing with the baseline models. Specifically, the outputs of MUNIT show less capacity than ours in transferring the style as shown in (Smiling $\rightarrow$ Non Smiling), (Non Bang $\rightarrow$ Bang) and (Bang $\rightarrow$ Non Bang), because MUNIT matches only mean and variance of the style to those of the content when conducting a translation. On the other hand, DRIT conducts unnatural translation (two rows from the bottom) comparing with ours. Furthermore, in case of (Non Smiling $\rightarrow$ Smiling), DRIT applies the style only into a mouth but ours converts both eyes and mouth.

Note that we exclude WCT when analyzing the results due to the fact that WCT performs style transfer in overall attribute translation case.

6.3. Extra results.

Finally, we present the extra results of our model in Fig. 10, 11, 12, each translated attribute is written on the top of the macro column. All of the outputs in those figures are generated by the unseen data. Through the results, we

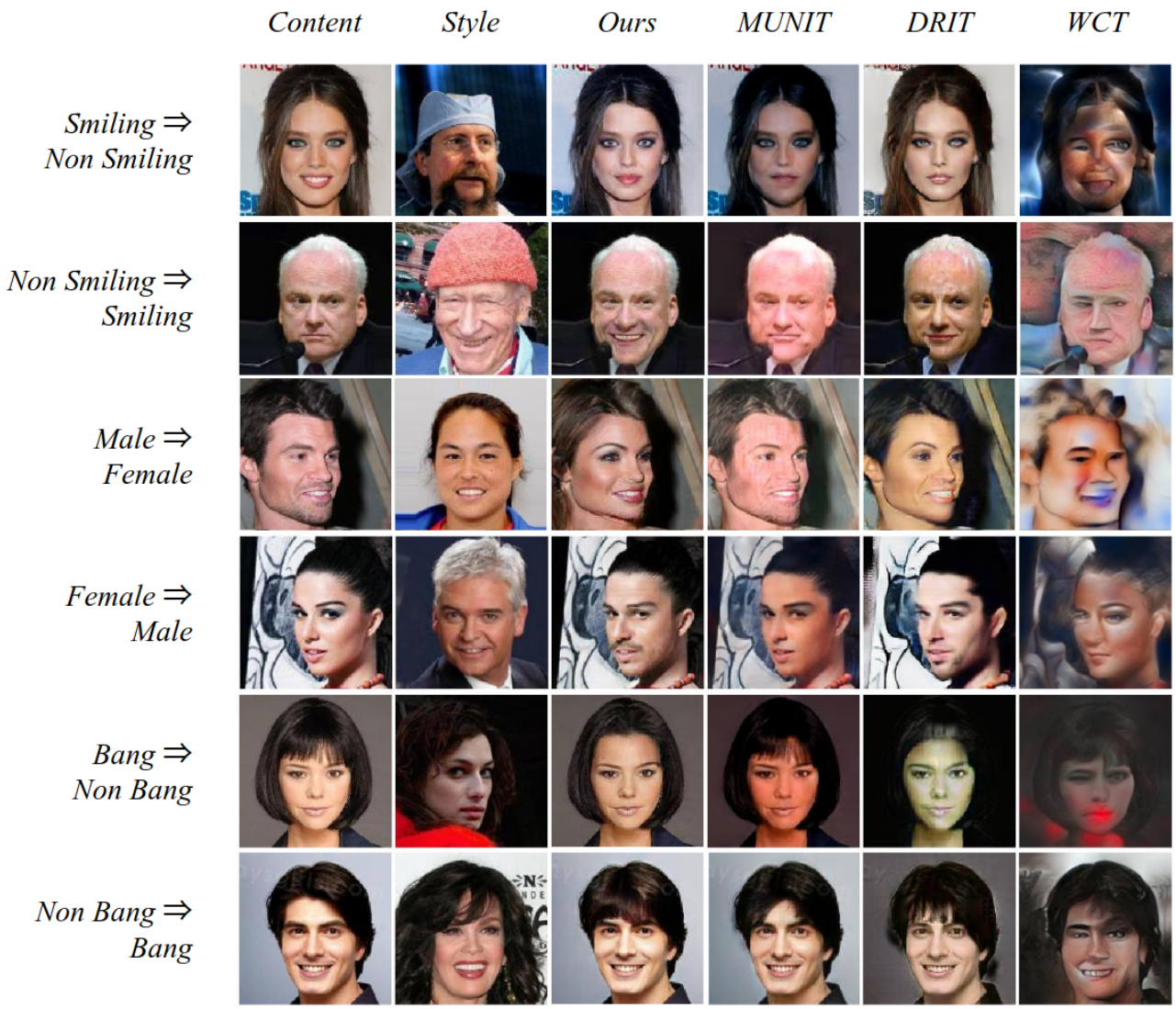

Figure 9: Comparison with the baseline models on CelebA dataset. Our model shows the superior performance on every attribute comparing with other models.

verify a superior performance of our model.

Figure 10: Extra results on CelebA dataset.

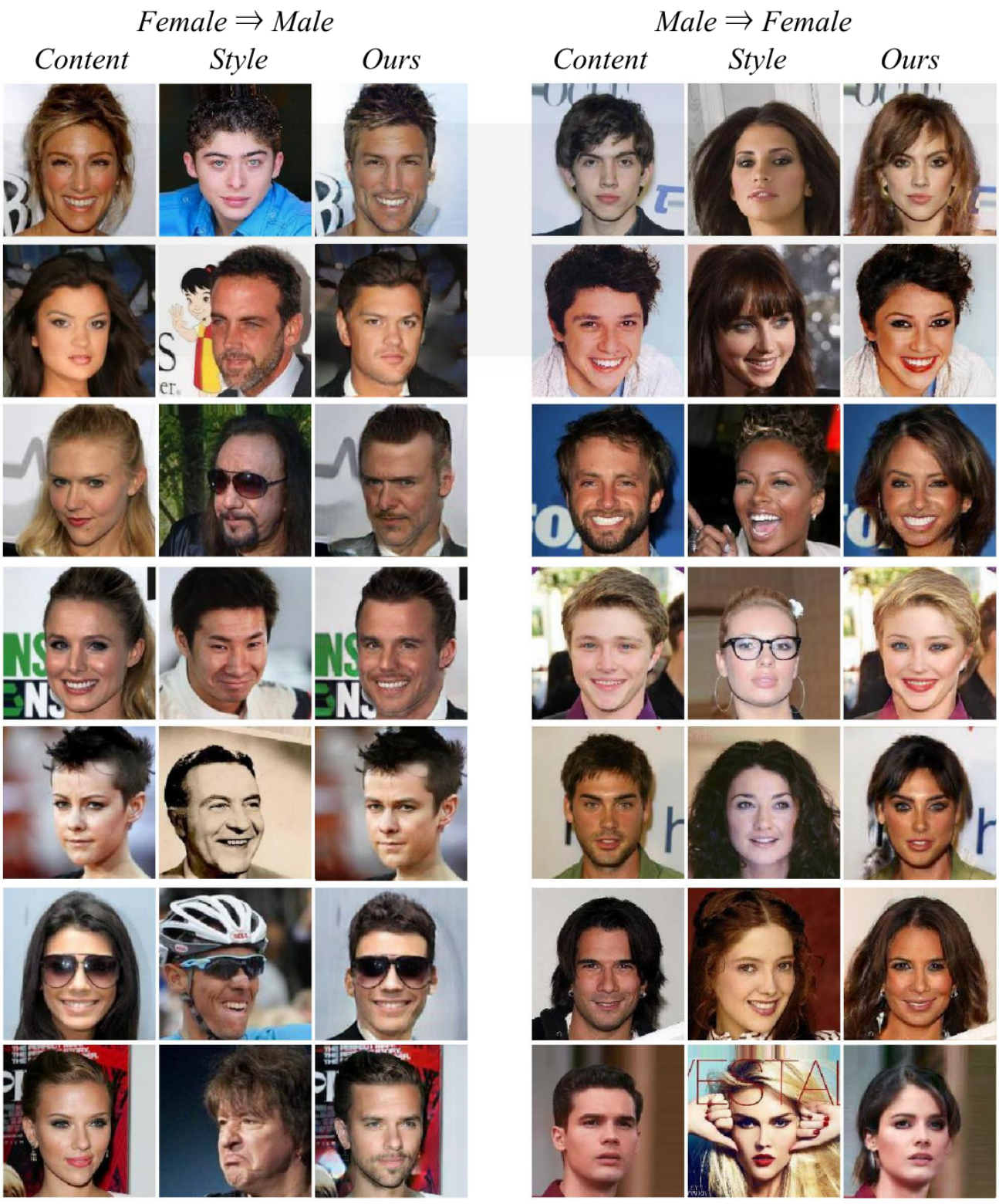

Figure 11: Extra results on CelebA dataset.

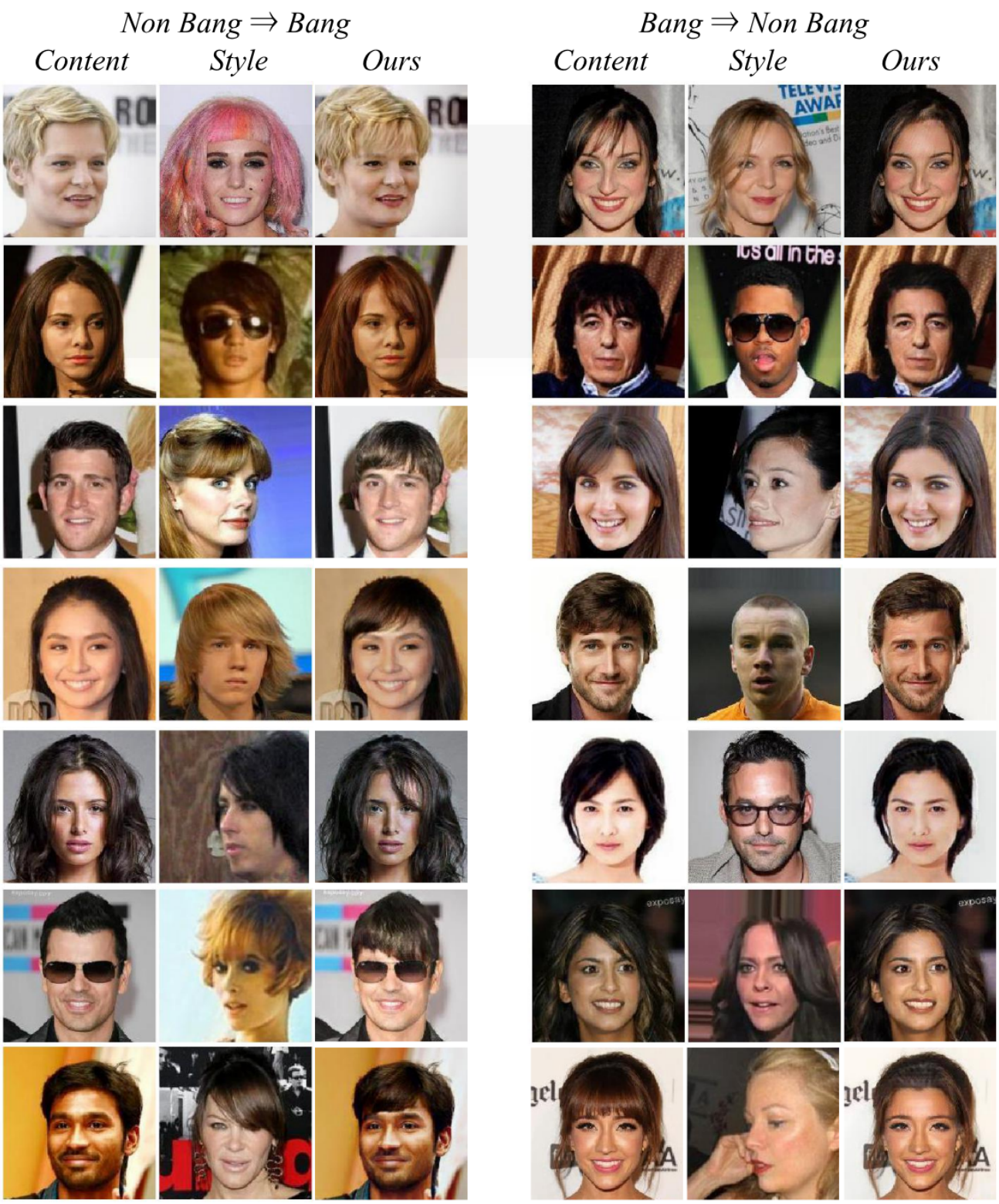

Figure 12: Extra results on CelebA dataset.

References

[1] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo. Coloring with words: Guiding image colorization through text-based palette generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 431–447, 2018.

[2] H. Chang, J. Lu, F. Yu, and A. Finkelstein. PairedCycleGAN: Asymmetric style transfer for applying and removing makeup. In *CVPR 2018*, June 2018.

[3] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stylebank: An explicit representation for neural image style transfer. In *Proc. CVPR*, volume 1, page 4, 2017.

[4] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, and D. A. Forsyth. Learning diverse image colorization. In *CVPR*, pages 2877–2885, 2017.

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[7] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[8] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[9] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. Torr, and P. K. Dokania. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv:1704.02906*, 1(4), 2017.

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

[11] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[12] L. Huang, D. Yang, B. Lang, and J. Deng. Decorrelated batch normalization, 2018.

[13] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017.

[14] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[15] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2965–2973, 2015.

[16] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017.

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[18] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[19] D. Kinga and J. B. Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, 2015.

[20] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[21] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[22] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Diversified texture synthesis with feed-forward networks. In *Proc. CVPR*, 2017.

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017.

[24] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. *arXiv preprint arXiv:1802.06474*, 2018.

[25] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[26] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[29] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[30] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[32] X. Wei, Y. Zhang, Y. Gong, J. Zhang, and N. Zheng. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[33] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] Y. Wu and K. He. Group normalization. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[35] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017.

[36] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[38] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 465–476. Curran Associates, Inc., 2017.