

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

Wonwoong Cho¹ Sungha Choi^{1,2} David Keetae Park¹ Inkyu Shin³ Jaegul Choo¹
¹Korea University ²LG Electronics ³Hanyang University

Abstract

Recently, unsupervised exemplar-based image-to-image translation, conditioned on a given exemplar without the paired data, has accomplished substantial advancements. In order to transfer the information from an exemplar to an input image, existing methods often use a normalization technique, e.g., adaptive instance normalization, that controls the channel-wise statistics of an input activation map at a particular layer, such as the mean and the variance. Meanwhile, style transfer approaches similar task to image translation by nature, demonstrated superior performance by using the higher-order statistics such as covariance among channels in representing a style. In detail, it works via whitening (given a zero-mean input feature, transforming its covariance matrix into the identity), followed by coloring (changing the covariance matrix of the whitened feature to those of the style feature). However, applying this approach in image translation is computationally intensive and error-prone due to the expensive time complexity and its non-trivial backpropagation. In response, this paper proposes an end-to-end approach tailored for image translation that efficiently approximates this transformation with our novel regularization methods. We further extend our approach to a group-wise form for memory and time efficiency as well as image quality. Extensive qualitative and quantitative experiments demonstrate that our proposed method is fast, both in training and inference, and highly effective in reflecting the style of an exemplar. Finally, our code is available at <https://github.com/WonwoongCho/GDWCT>.

1. Introduction

Since the introduction of image-to-image translation [16], in short, image translation, it has gained significant attention from relevant fields and constantly evolved propelled by the seminal generative adversarial networks [10]. The primary goal of image translation [16, 39] is to convert particular attributes of an input image in an

original domain to a target one, while maintaining other semantics. Early models for image translation required training data as paired images of an input and its corresponding output images, allowing a direct supervision. CycleGAN [39] successfully extends it toward unsupervised image translation [26, 2, 4, 39] by proposing the cycle consistency loss, which allows the model to learn the distinctive semantic difference between the collections of two image domains and translate the corresponding style without a direct pair-wise supervision.

Nonetheless, CycleGAN is still unimodal in that it can only generate a single output for a single input. Instead, image translation should be capable of generating multiple possible outputs even for a single given input, e.g., numerous possible gender-translated outputs of a single facial image. 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 an additional, exemplar image for a detailed guidance toward a desired style.

They both have two separate encoders corresponding to the content image (an input) and style image (an exemplar), and combine the content feature and style feature together to produce the final output. 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 is proven to perform well in image translation.

However, we hypothesize that matching only these two statistics may not reflect the target style well enough, ending up with the sub-optimal quality of image outputs on numerous occasions, as we confirm through our experiments in Section 4. That is, the interaction effects among variables, represented as the Gram matrix [8] or the covariance matrix [23], can convey critical information of the style, which

is agreed by extensive studies [8, 7, 23]. In response, to fully utilize the style information of an exemplar, we propose a novel method that takes into account such interaction effects among feature channels, in the context of image translation.

Our model is mainly motivated by whitening-and-coloring transformation (WCT) [24], which utilizes the pair-wise feature covariances, in addition to the mean and the variance of each single feature, to encode the style of an image. To elaborate, whitening refers to the normalization process to make every covariance term (between a pair of variables) as well as every variance term (within each single variable) as a unit value, with given an input whose each single variable is zero-meaned. This plays a role in removing (or neutralizing) the style. On the other hand, coloring indicates the procedure of matching the covariance of the style to that of the content feature, which imposes the intended style into an neutralized input image.

The problem when applying WCT in image translation is that its time complexity is as expensive as $O(n^3)$ where n is the number of channels of a given activation map. Furthermore, computing the backpropagation with respect to singular value decomposition involved in WCT is non-trivial [33, 15]. To address these issues, we propose a novel deep whitening-and-coloring transformation that flexibly approximates the existing WCT based on deep neural networks. We further extend our method into group-wise deep whitening-and-coloring transformation (GDWCT), which does not only reduce the number of parameters and the training time but also boosts the generated image quality [35, 12].

The main contribution of this paper includes:

- We present the novel deep whitening-and-coloring approach that allows an end-to-end training in image translation for conveying profound style semantics.
- We also propose the group-wise deep whitening-and-coloring algorithm to further increase the computational efficiency through a simple forward propagation, which achieves highly competitive image quality.
- We demonstrate the effectiveness of our method via extensive quantitative and qualitative experiments, compared to state-of-the-art methods.

2. Related Work

Image-to-image translation. Image-to-image translation aims at converting an input image to another image with a target attribute. Many of its applications exist, e.g., colorization [38, 5, 1, 37], super-resolution [6, 20], and domain adaptation [11, 22].

A slew of studies have been conducted in an unsupervised setting of image translation [39, 18, 26]. StarGAN [4] proposes a single unified model which can handle unsupervised image translation among multiple different domains.

Several studies [9, 40] focus on the limitation of earlier work in which they produce a single output given an input without consideration that diverse images can be generated within the same target domain. However, they are not without limitations, either by generating a limited number of outputs [9] or requiring paired images [40].

Recently proposed approaches [14, 21] are capable of generating multimodal outputs in an unsupervised manner. They work mainly based on the assumption that a latent image space could be separated into a domain-specific style space a domain-invariant content spaces. Following the precedents, we also adopt the separate encoders to extract out each of the content and style features.

Style transfer. Gatys et al. [7, 8] show that the pair-wise feature interactions obtained from the Gram matrix or the covariance matrix of deep neural networks successfully capture the image style. It is used for transferring the style information from a style image to a content image by matching the statistics of the style feature with those of the content. However, 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, alternative methods [30, 3, 17] have achieved a superior time efficiency through feed-forward networks approximating an optimal result of the iterative methods.

However, these models are incapable of transferring an unseen style from an arbitrary image. To alleviate the limitation, several approaches enable an unseen, arbitrary neural style transfer [13, 24, 25]. 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 [24] encodes the style as the feature covariance matrix, so that it effectively captures the rich style representation. Recently, a new approach [22] have approximated the whitening-and-coloring transformation as a one-time transformation through a single transformation matrix. Even though the idea of learning the transformation is similar to ours, the proposed networks are incapable of transferring the semantic style information, such as the translation between the cat and the dog because the existing approach can transfer only the general style, such as the color and the texture. Moreover, its settings of approximating the transformation is less rigorous than ours due to the lack of the regularization for ensuring the whitening-and-coloring transformation.

3. Proposed Method

This section describes our proposed model in detail, by first giving a model overview and by explaining the our proposed loss functions.

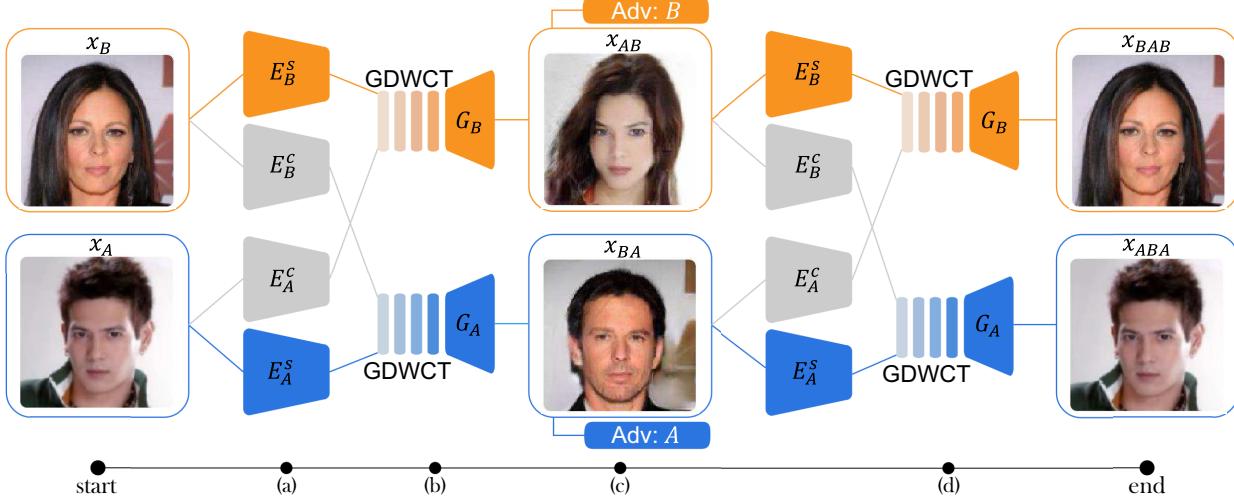


Figure 1: Overview of our model. (a) To translate from $\mathcal{A} \rightarrow \mathcal{B}$, we first extract the content feature $c_{\mathcal{A}}$ from the image $x_{\mathcal{A}}$ (i.e., $c_{\mathcal{A}} = E_{\mathcal{A}}^c(x_{\mathcal{A}})$) and the style feature $s_{\mathcal{B}}$ from the image $x_{\mathcal{B}}$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

3.1. Model Overview

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 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.,

$$\begin{aligned} \{c_{\mathcal{A}}, s_{\mathcal{A}}\} &= \{E_{\mathcal{A}}^c(x_{\mathcal{A}}), E_{\mathcal{A}}^s(x_{\mathcal{A}})\} & c_{\mathcal{A}} \in \mathcal{C}, s_{\mathcal{A}} \in \mathcal{S}_{\mathcal{A}} \\ \{c_{\mathcal{B}}, s_{\mathcal{B}}\} &= \{E_{\mathcal{B}}^c(x_{\mathcal{B}}), E_{\mathcal{B}}^s(x_{\mathcal{B}})\} & c_{\mathcal{B}} \in \mathcal{C}, s_{\mathcal{B}} \in \mathcal{S}_{\mathcal{B}}, \end{aligned}$$

where $\{E_{\mathcal{A}}^c, E_{\mathcal{B}}^c\}$ and $\{E_{\mathcal{A}}^s, E_{\mathcal{B}}^s\}$ are the 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.,

$$x_{\mathcal{A} \rightarrow \mathcal{B}} = f_{\mathcal{A} \rightarrow \mathcal{B}}(x_{\mathcal{A}}, x_{\mathcal{B}}) = G_{\mathcal{B}}(E_{\mathcal{A}}^c(x_{\mathcal{A}}), E_{\mathcal{B}}^s(x_{\mathcal{B}}))$$

$$x_{\mathcal{B} \rightarrow \mathcal{A}} = f_{\mathcal{B} \rightarrow \mathcal{A}}(x_{\mathcal{B}}, x_{\mathcal{A}}) = G_{\mathcal{A}}(E_{\mathcal{B}}^c(x_{\mathcal{B}}), E_{\mathcal{A}}^s(x_{\mathcal{A}})),$$

where $\{G_{\mathcal{A}}, G_{\mathcal{B}}\}$ are the generators for the corresponding domains.

As illustrated in Fig. 1, the group-wise deep whitening-and-coloring transformation (GDWCT), plays a main role in 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 $s_{\mathcal{B}}^{\text{CT}}$, and the mean of the style $s_{\mathcal{B}}^{\mu}$ as input and conduct a translation of $c_{\mathcal{A}}$ to $c_{\mathcal{A} \rightarrow \mathcal{B}}$, formulated as

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

where $s_{\mathcal{B}}^{\text{CT}} = \text{MLP}_{\mathcal{B}}^{\text{CT}}(s_{\mathcal{B}})$ and $s_{\mathcal{B}}^{\mu} = \text{MLP}_{\mathcal{B}}^{\mu}(s_{\mathcal{B}})$. 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 determine how much of the style to apply considering that the amount of the style information the networks require may vary, i.e., $c_{\mathcal{A} \rightarrow \mathcal{B}} = \alpha(\text{GDWCT}(c_{\mathcal{A}}, s_{\mathcal{B}}^{\text{CT}}, s_{\mathcal{B}}^{\mu})) + (1 - \alpha)c_{\mathcal{A}}$.

The different layers of a model focus on different information (e.g., the low-level feature captures a local fine pattern, whereas the high-level one captures a complicated pattern across 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 modules, our model can simultaneously reflect both the fine- and coarse-level style information.

3.2. Loss Functions

Following MUNIT [14] and DRIT [21], we adopt both the latent-level and the pixel-level reconstruction losses. First, we use 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 $x_{\mathcal{A} \rightarrow \mathcal{B}}$, i.e.,

$$\mathcal{L}_s^{\mathcal{A} \rightarrow \mathcal{B}} = \mathbb{E}_{x_{\mathcal{A} \rightarrow \mathcal{B}}, x_{\mathcal{B}}} [\|E_{\mathcal{B}}^s(x_{\mathcal{A} \rightarrow \mathcal{B}}) - E_{\mathcal{B}}^s(x_{\mathcal{B}})\|_1]$$

Second, we utilize 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 $c_{\mathcal{A} \rightarrow \mathcal{B}}$, i.e.,

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

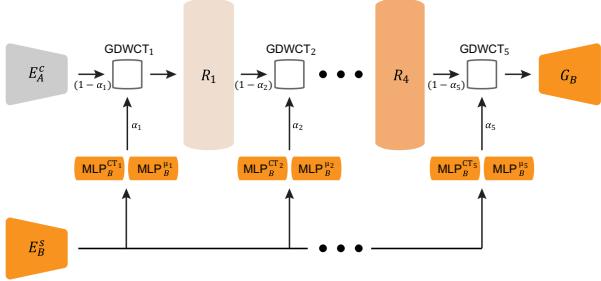


Figure 2: Image translation via the proposed GDWCT. We apply the style via multiple hops to apply the style from the low-level feature to the high-level feature.

Third, in order to guarantee the performance of our model through pixel-level supervision, we adopt the cycle consistency loss and the identity loss [39] to obtain a high-quality image, i.e.,

$$\begin{aligned}\mathcal{L}_{cyc}^{\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}} &= \mathbb{E}_{x_{\mathcal{A}}} [\|x_{\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}} - x_{\mathcal{A}}\|_1] \\ \mathcal{L}_i^{\mathcal{A} \rightarrow \mathcal{A}} &= \mathbb{E}_{x_{\mathcal{A}}} [\|x_{\mathcal{A} \rightarrow \mathcal{A}} - x_{\mathcal{A}}\|_1].\end{aligned}$$

Lastly, we use an adversarial loss for minimizing the discrepancy between the distribution of the real image and that of the generated image. In particular, we employ LS-GAN [28] as the adversarial method, i.e.,

$$\begin{aligned}\mathcal{L}_{D_{adv}}^{\mathcal{B}} &= \frac{1}{2} \mathbb{E}_{x_{\mathcal{B}}} [(D(x_{\mathcal{B}}) - 1)^2] + \frac{1}{2} \mathbb{E}_{x_{\mathcal{A} \rightarrow \mathcal{B}}} [(D(x_{\mathcal{A} \rightarrow \mathcal{B}}))^2] \\ \mathcal{L}_{G_{adv}}^{\mathcal{B}} &= \frac{1}{2} \mathbb{E}_{x_{\mathcal{A} \rightarrow \mathcal{B}}} [(D(x_{\mathcal{A} \rightarrow \mathcal{B}}) - 1)^2]\end{aligned}$$

To consider the opposite translation, similar to DRIT [21], our model is trained in both directions, ($\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}$) and ($\mathcal{B} \rightarrow \mathcal{A} \rightarrow \mathcal{B}$), at the same time. Finally, our full loss function is represented as

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

where \mathcal{L} without a domain notation indicates both directions between two domains, and 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 makes the covariance matrix of a given input into an identity matrix. Specifically, we first subtract the content feature $c \in \mathcal{R}^{C \times BHW}$ by its mean c^{μ} , where (C, B, H, W) represent the number of channels, batch size, height, and width, respectively. We then compute the outer product of the zero-meaned c along the BHW dimension.

Lastly, we obtain the covariance matrix $\Sigma_c \in \mathcal{R}^{C \times C}$ and factorize it via eigendecomposition, i.e.,

$$\Sigma_c = \frac{1}{BHW-1} \sum_{i=1}^{BHW} (c_i - c^{\mu})(c_i - c^{\mu})^T = Q_c \Lambda_c Q_c^T,$$

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

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

where c_w denotes the whitened feature. However, as pointed out in Section 1, eigendecomposition is not only computationally intensive but also difficult to backpropagate the gradient signal. To alleviate the problem, we propose the deep whitening transformation (DWT) approach such that the content encoder E_c can naturally encode the whitened feature c_w , i.e., $c_w = c - c^{\mu}$, where $E_c(c_w) = c$. To this end, we propose the novel regularization term that makes the covariance matrix of the content feature Σ_c as close as possible to the identity matrix, i.e.,

$$\mathcal{R}_w = \mathbb{E}[\|\Sigma_c - I\|_{1,1}]. \quad (2)$$

Thus, the whitening transformation in Eq. (1) is reduced to $c_w = c - c^{\mu}$ in DWT.

However, several limitations exist in DWT. First of all, estimating the full covariance matrix using a small batch of given data is inaccurate [12]. Second, performing DWT with respect to the entire channels may excessively throw away the content feature, compared to channel-wise standardization. We therefore improve DWT by grouping channels and applying DWT to the individual group.

Concretely, the channel dimension of c is re-arranged at a group level, i.e., $c \in \mathcal{R}^{G \times (C/G) \times BHW}$, where G is the number of groups. After obtaining the covariance matrix Σ_c in $\mathcal{R}^{G \times (C/G) \times (C/G)}$, we apply Eq. (2) along its group dimension. Note that group-wise DWT (GDWT) is the same with DWT during the forward phase, as shown in Fig. 3(a), because the re-arranging procedure is required for the regularization (2).

Coloring transformation (CT). CT matches the covariance matrix of the whitened feature with that of the style feature Σ_s , where Σ_s is the covariance matrix of the style feature. Σ_s is then decomposed into $Q_s \Lambda_s Q_s^T$, used for the subsequent coloring transformation. This process is written as

$$c_{cw} = Q_s \Lambda_s^{\frac{1}{2}} Q_s^T c_w, \quad (3)$$

where c_{cw} denotes the colored feature.

Similar to WT, however, CT has the problems of expensive time complexity and non-trivial backpropagation. Thus, We also replace CT with a simple but effective method that we call a deep coloring transformation

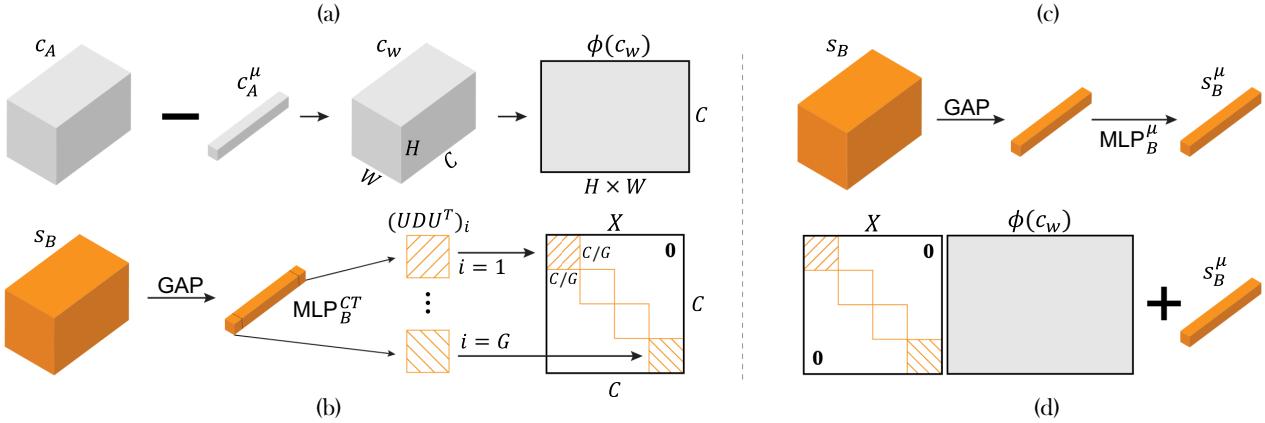


Figure 3: Details on the proposed GDWCT module. (a) The process for obtaining the whitened feature. Because the regularization (Eq. (2)) encourages the zero-mean content feature $c - c^\mu$ to be the whitened feature c_w , we just subtract the mean of the content feature c_A^μ from c_A . (b) The procedure of approximating the coloring transformation matrix (Section 3.3). (c) We obtain the mean of the style feature s_B^μ by forwarding it to the MLP layer MLP_B^μ . (d) Our module first multiply the whitened feature c_w with the group-wise coloring transformation matrix X . We then add it with the mean of the style s_B^μ .

(DCT). Specifically, we first obtain the matrix s^{CT} through $\text{MLP}^{\text{CT}}(s)$, where $s = E_s(x)$. We then decompose s^{CT} into two matrices by computing its column-wise L_2 norm, i.e., $s^{\text{CT}} = UD$, where the i -th column vector u_i of $U \in \mathcal{R}^{C \times C}$ is the unit vector, and $D \in \mathcal{R}^{C \times C}$ is the diagonal matrix whose diagonal entries correspond to the L_2 norm of each column vector of s^{CT} . We assume that those matrices UD is equal to two matrices in Eq. (3), i.e., $UD = Q_s \Lambda_s^{\frac{1}{2}}$.

In order to properly work as Q_s and $\Lambda_s^{\frac{1}{2}}$, U needs to be an orthogonal matrix, and every diagonal entry in the matrix D should be positive. To assure the conditions, we add the regularization for U to encourage the column vectors of U to be orthogonal, i.e.,

$$\mathcal{R}_c = \mathbb{E}_s [\|U^T U - I\|_{1,1}]. \quad (4)$$

The diagonal matrix D has its diagonal elements as the column-wise L_2 norm of s^{CT} , such that its diagonal entries are already positive. Thus, it does not necessitate additional regularization. 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, with the regularization Eq. (4), UD satisfies the entire conditions to be $Q_s \Lambda_s^{\frac{1}{2}}$. Finally, combining U and D , we simplify CT as

$$c_{cw} = UDU^T c_w. \quad (5)$$

However, approximating the entire matrix s^{CT} has an expensive computational cost (the number of parameters to estimate is C^2). Hence, we extend DCT to the group-wise DCT (**GDCT**) and reduce the number of parameters from C^2 to C^2/G , as the detailed steps are illustrated in Fig. 3(b). We first obtain the i -th matrix $\{UDU^T\}_i \in \mathcal{R}^{(C/G) \times (C/G)}$ for GDCT for $i = \{1, \dots, G\}$. We then form a block diagonal matrix $X \in \mathcal{R}^{C \times C}$ by arranging the matrices $\{UDU^T\}_{1, \dots, G}$. Next, as shown in Fig. 3(d), we com-

pute the matrix multiplication with X and the whitened feature c_w , thus Eq. (5) being reduced to

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

where ϕ denotes a reshaping operation $\phi : \mathcal{R}^{C \times H \times W} \rightarrow \mathcal{R}^{C \times HW}$. Finally, we add the new mean vector s^μ to the c_{cw} , where $s^\mu = \text{MLP}^\mu(s)$, as shown in Fig. 3(c). We empirically set $\lambda_w = 0.001$, $\lambda_c = 10$, and $G = 4, 8, 16$.

4. Experiments

This section describes the baseline models and the datasets. Implementation details as well as additional comparisons and results are included in the appendix.

4.1. Experimental Setup

Datasets. We evaluate GDWCT with various datasets including CelebA [27], Artworks [39] (Ukiyoe, Monet, Cezanne, and Van Gogh), cat2dog [21], Pen ink and Watercolor classes of the Behance Artistic Media (BAM) [34], and Yosemite [39] (summer and winter scenes) datasets.

Baseline methods. We exploit MUNIT [14], DRIT [21], and WCT [24] as our baselines because those methods are the state-of-the-art in image translation and style transfer, respectively. MUNIT and DRIT utilize different methods when applying the style into the content from GDWCT. MUNIT leverages AdaIN [13] while DRIT is based on concatenation of the content and the style features. Meanwhile, WCT applies the whitening-and-coloring transformation to the features extracted from the pretrained encoder, in order to transfer the style into the content image.

4.2. Quantitative Analysis

We compare the performance of our model with the baselines with user study and classification accuracy.



Figure 4: Qualitative comparisons based on Artworks dataset [39].

User study. We first conduct a user study using CelebA dataset [27]. The participants of the user study was instructed 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 in an exemplar. Each user evaluated 60 sets of image comparisons, choosing one among four candidates within 30 seconds per comparison. We informed the participants of the original and the target domains for every run, e.g., Male to Female, so that they can understand 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 → 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.

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 indicates that the model learns deterministic patterns to be represented in the target domain. The classification results are reported in Table 2. For the classification, we adopted the pretrained Inception-v3 [29] 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 Smile.

	MUNIT	DRIT	WCT	GDWCT
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
Smile → Non-Smile	5.56	30.35	1.35	64.44
Non-Smile → Smile	2.30	22.25	2.25	73.33

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

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

Table 2: Comparison of the classification accuracy (%).

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.

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

Table 3: Comparison of the inference time. Tested with the image size 256×256 on a NVIDIA Titan XP GPU, and averaged over 1,000 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. 9. We believe our work gives another dimension of an opportunity to translate the 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 Artworks dataset (Ukiyoe) [39]. We train two identical models different only in the number of hops, a single hop (GDWTC₁) or multi-hops (GDWTC_{1–5}). In Fig. 5, the rightmost image (GDWTC_{1–5}) has the style that agrees with the detailed style given in the leftmost image. The third image (GDWTC₁) 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



Figure 5: Comparison between single- and multi-hops.

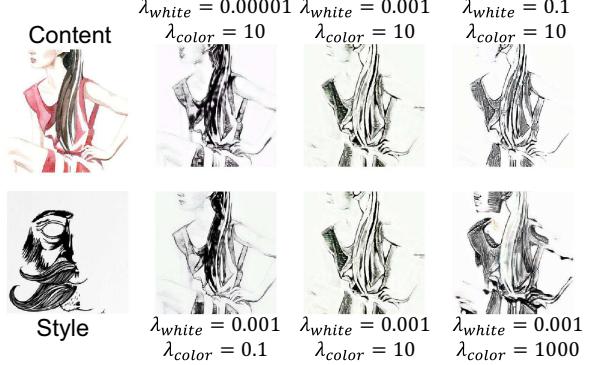


Figure 6: Visualization of the regularization influences.

of GDWCT₁, but is clearly rendered on GDWTC_{1–5}. The difference comes from a capacity of the multiple hops on a stylization, which covers both fine and coarse style [24].

Effects of regularization. We verify the influences of the regularizations \mathcal{R}_w and \mathcal{R}_c 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 [34] dataset. The images in Fig. 6 illustrates the results of (Water color \rightarrow Pen ink). 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 transformation. 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 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.

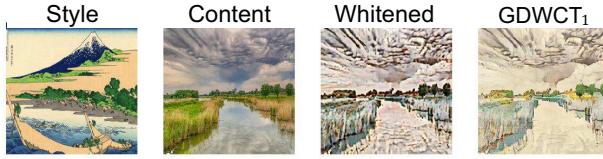


Figure 8: Visualization of whitening transformation that makes the content feature lose the original information.

Comparison on face attribute translation. We compare GDWCT with the baselines using CelebA dataset with the image size of 216×216 . The results are shown in Fig. 7. Two columns from the left of each macro column denote a content image and a style image (exemplar), respectively, while the other columns indicates outputs of compared models. Each row of each macro column illustrates the different target attribute. Our model shows a superior performance in overall attribute translation, because our model drastically but suitably applies the style compared to the baselines. For example, In case of (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 Smile and Bangs, the outputs of MUNIT show less capacity than ours in transferring the style as shown in (Smile \rightarrow Non-Smile), (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. In case of (Non-Smile \rightarrow Smile), DRIT applies the style only into a mouth but ours converts both eyes and mouth. Meanwhile, as seen in overall cases of WCT, it cannot perform image translation because it does not learn to transfer the semantic style.

5. Conclusion

In this paper, we propose a novel framework, group-wise deep whitening-and-coloring transformation (GDWCT) for



Figure 7: Comparison with the baseline models on CelebA dataset; (a) Smile \rightarrow Non-Smile (b) Non-Smile \rightarrow Smile (c) Male \rightarrow Female (d) Female \rightarrow Male (e) Bang \rightarrow Non-Bang (f) Non-Bang \rightarrow Bang

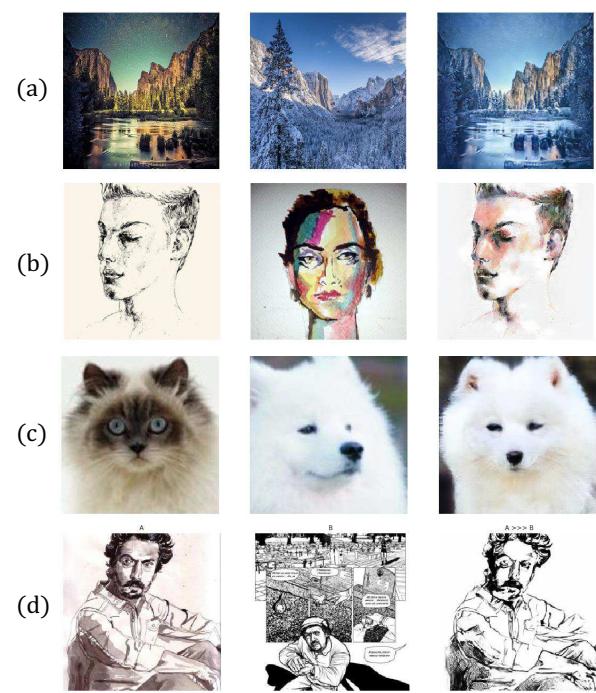
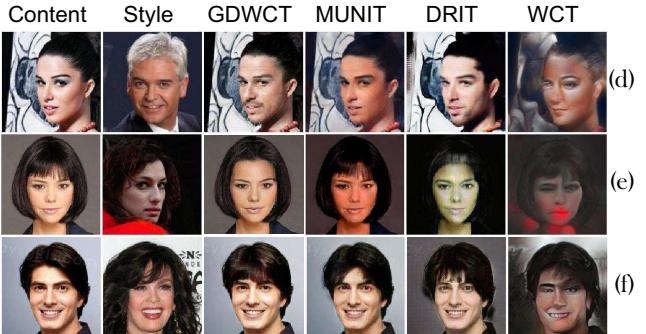


Figure 9: Results on various dataset; (a) Yosemite (b) BAM (Pen Ink \rightarrow Water Color) (c) Cat2dog (d) BAM (Water Color \rightarrow Pen Ink)

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.

Acknowledgement. This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF2016R1C1B2015924). Jaegul Choo is the corresponding author.



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 provide a discussion on the effects of the number of groups with qualitative results in subsection 6.2. Third, we qualitatively and quantitatively compare our model with the baseline models on CelebA dataset in subsection 6.3. Finally, we report extra results on CelebA dataset in subsection 6.4.

6.1. Implementation

Content encoder. The content encoders $\{E_A^c, E_B^c\}$ are composed of a few strided convolutional (conv) layers and four residual blocks. The size of the output activation map is in $\mathcal{R}^{256 \times \frac{H}{4} \times \frac{W}{4}}$. Note that we use the instance normalization [31] along with the entire layers in E_c in order to flatten the content feature [14, 13].

Style encoder. The style encoders $\{E_A^s, E_B^s\}$ consist of several strided conv layers with the output size in $\mathcal{R}^{256 \times \frac{H}{16} \times \frac{W}{16}}$. After the global average pooling, the style feature s is forwarded into the MLP^{CT} and MLP^{μ} . We use the group normalization [35] 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 $\{MLP_A^{\text{CT}}, MLP_B^{\text{CT}}\}$ and $\{MLP_A^{\mu}, MLP_B^{\mu}\}$ is composed of several linear layers. The input dimension of MLP^{CT} 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 the same with the number of channels, 256.

Generator. The generators $\{G_A, G_B\}$ 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 $\{D_A, D_B\}$ are in the form of multi-scale discriminators [32]. The size of the output activations are in $\mathcal{R}^{4 \times 4}$, $\mathcal{R}^{8 \times 8}$ and $\mathcal{R}^{16 \times 16}$.

Training details. We use the Adam optimizer [19] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ with a learning rate of 0.0001 for all generators and discriminators. Other settings are chosen differently based on the experimented dataset. In CelebA, we apply a batch size of eight with the image size of (216×216) . The original image size (178×218) is resized to (216×264.5) , followed by the center-crop to be

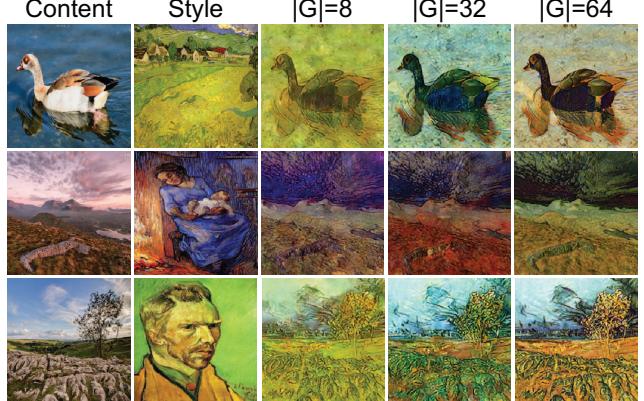


Figure 10: Effects of the number of groups.

(216×216) . Corresponding models are trained for 500,000 iterations with a decaying rate of 0.5 applied from the 100,000th iteration in every 50,000 iterations. In all other datasets, We train the model with the batch size of two and the image size of (256×256) . Note that we first resize each image up to 286, then perform a random cropping. We set 200,000 iterations for the training and apply the decaying rate of 0.5 from 100,000th iterations in every 10,000 iterations. All the experiments are trained using a single NVIDIA TITAN Xp GPU for three days. The group size is empirically set among 4, 8, 16.

6.2. Effects of Different Number of Groups

We discuss and conduct additional experiments on the effects of different group sizes.

First of all, the number of groups, $|G|$, is closely related to the number of model parameters to represent the style statistics of a given exemplar. Specifically, the number of model parameters is equivalent to $C^2/|G|$, where C and $|G|$ represent the numbers of channels and groups, respectively, as discussed in Section 3.3. Thus, increasing $|G|$ has the effect of reducing the model size, i.e., the number of parameters.

We also conduct a qualitative experiment to show the effects of $|G|$ on the final output. As shown in Fig. 10, a small value of $|G|$ tends to focus on the low-level information of the style. For example, those results with $|G| = 8$ in the third column mainly reflect the colors of the exemplar, while those results with $|G| = 64$ in the rightmost column do not necessarily maintain the color tone. We additionally observe that the style becomes more distinguishable across different objects in a resulting image as $|G|$ increases, such that the color of the duck in the first row becomes more distinct from the background as $|G|$ gets larger. We believe it is ascribed to the fact that larger $|G|$ shows the better capability in capturing contrast between objects in the image.



Figure 11: Qualitative comparison on attribute translation. Tested with image size of 216×216 .

Although we attempted to rigorously figure out the effects of $|G|$ on our method, however, through several experiments, $|G|$ sometimes shows inconsistent patterns, so that the generalization of the effects of $|G|$ is vague and difficult. Thus, as a future work, it is required to explore in-depth the influences the number of groups gives rise to.

6.3. Additional Comparison Results

In order to further validate the performances of our method, we additionally compare our method against ELEGANT [36], a recently proposed approach that focuses on facial attribute translation and exploits the adversarial loss. As shown in Fig. 11, qualitative results show that our method performs better than ELEGANT in terms of intended attribute translation. For instance, in the first row, our method generates more luxuriant bangs than the baseline method when translating from ‘Non-Bang’ to ‘Bang’. Better results are also found in the Smile attribute, which shows the results closer to the given style. The person in the last row is translated to a female of a high quality with regard to the eyes. ELEGANT encodes all target attributes in a disentangled manner in a single latent space and substitutes a particular part of the feature from one image to another. Since ELEGANT neither decomposes a given image into the content and the style nor matches the statistics of the style to that of the content, it shows worse performances in properly reflecting the style image than our proposed model.

Furthermore, we also show the outstanding performance of our method in a quantitative manner, as illustrated in Table 4. In all cases, our model achieves a higher classification accuracy by a large margin.

6.4. Extra Results

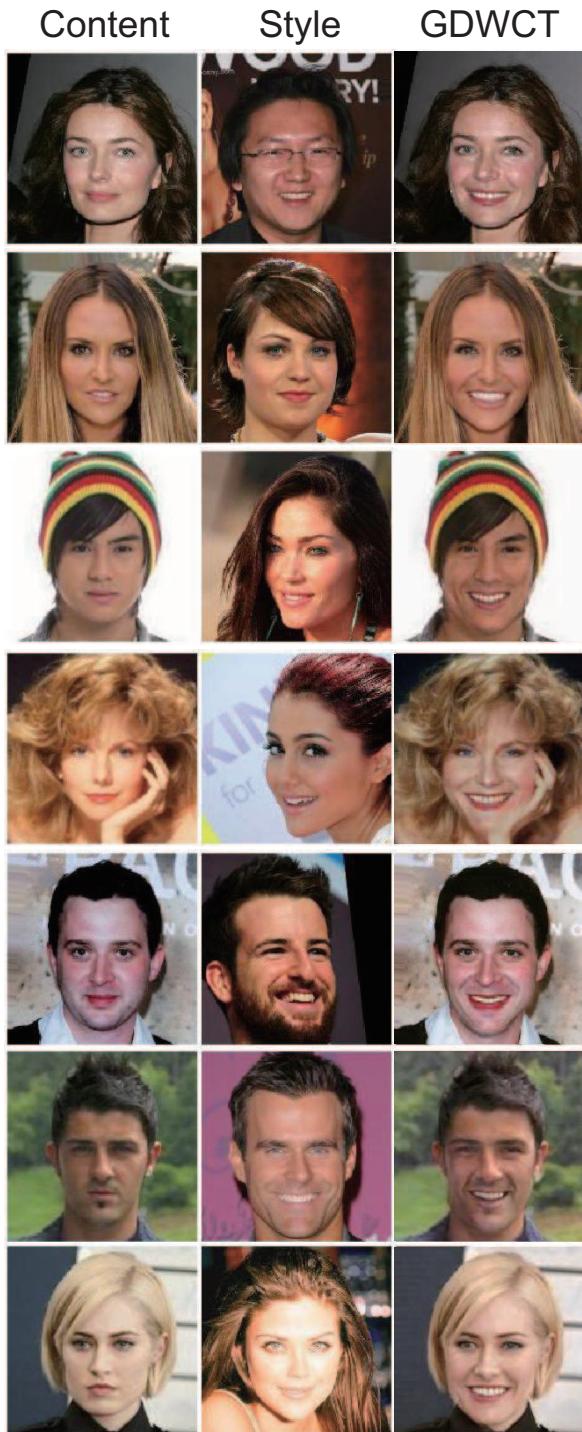
Finally, we present the extra results of our model in Fig. 12, 13, 14. Each translated attribute is written on the

	Gender	Bangs	Smile	Avg.
ELEGANT	77.15	61.73	70.88	69.92
Ours	92.65	76.05	92.85	87.18

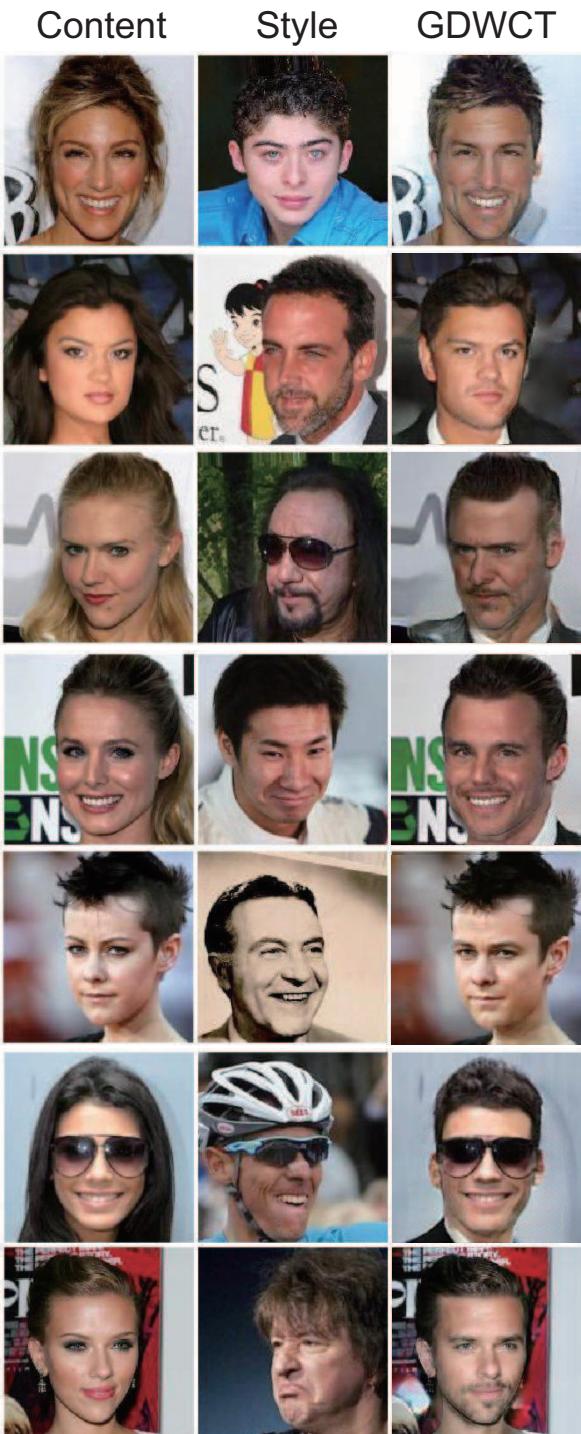
Table 4: Classification accuracy in percentages.

top of the macro column. All of the outputs in those figures are generated by the unseen data. Through the results, we verify a superior performance of GDWCT.

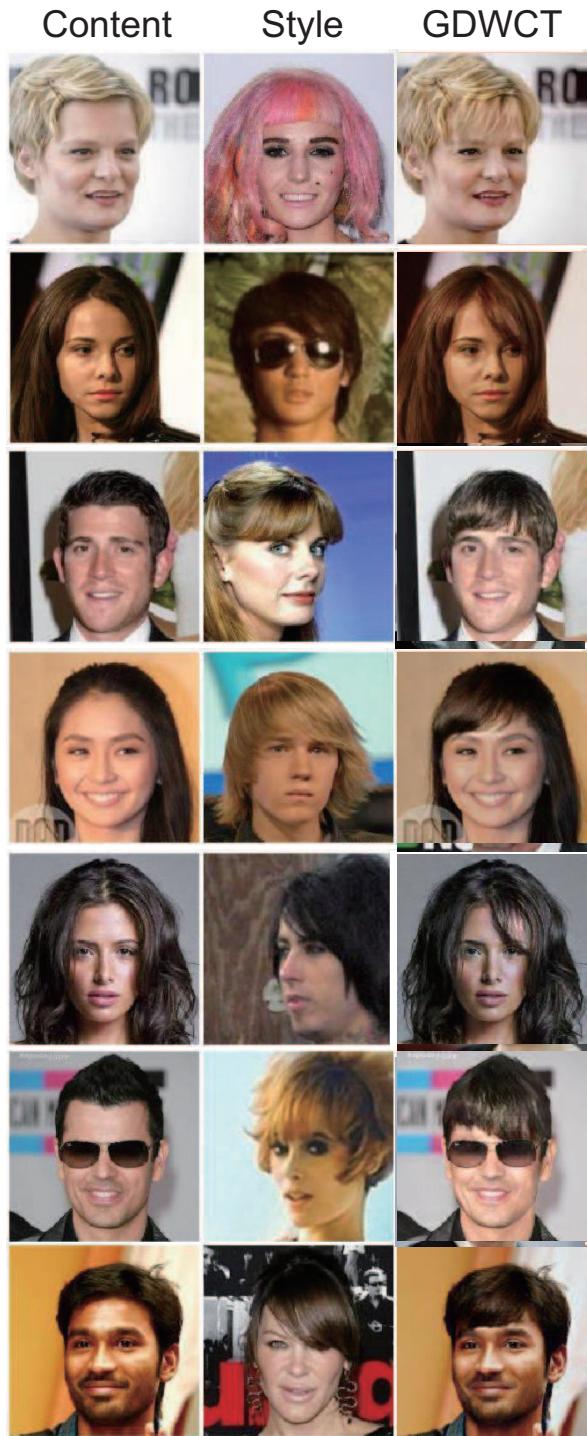
Non-Smile \Rightarrow Smile



Female \Rightarrow Male



Non-Bang \Rightarrow Bang



Bang \Rightarrow Non-Bang



Figure 14: Extra results on CelebA dataset.

References

- [1] Hyojin Bahng, Seungjoo Yoo, Wonwoong Cho, David Kee-tae Park, Ziming Wu, Xiaojuan Ma, and Jaegul Choo. Coloring with Words: Guiding image colorization through text-based palette generation. In *ECCV*, 2018.
- [2] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. PairedCycleGAN: Asymmetric style transfer for applying and removing makeup. In *CVPR*, 2018.
- [3] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *CVPR*, 2017.
- [4] Yunje 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 *CVPR*, 2018.
- [5] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David A Forsyth. Learning diverse image colorization. In *CVPR*, 2017.
- [6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014.
- [7] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015.
- [8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.
- [9] Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. Multi-agent diverse generative adversarial networks. In *CVPR*, 2018.
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.
- [11] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.
- [12] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *CVPR*, 2018.
- [13] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.
- [14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.
- [15] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, 2015.
- [16] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- [17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [18] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.
- [19] D Kinga and J Ba Adam. A method for stochastic optimization. In *ICLR*, 2015.
- [20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.
- [21] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.
- [22] Xuetong Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. In *CVPR*, 2019.
- [23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *CVPR*, 2017.
- [24] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NIPS*, 2017.
- [25] Yijun Li, Ming-Yu Liu, Xuetong Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018.
- [26] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.
- [27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.
- [28] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.
- [29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- [30] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture Networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016.
- [31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance Normalization: The missing ingredient for fast stylization, 2016.
- [32] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.
- [33] Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *ECCV*, 2018.
- [34] Michael J. Wilber, Chen Fang, Hailin Jin, Aaron Hertzmann, John Collomosse, and Serge Belongie. BAM! the behance artistic media dataset for recognition beyond photography. In *ICCV*, 2017.
- [35] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.
- [36] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. ELEGANT: Exchanging latent encodings with gan for transferring multiple face attributes. In *ECCV*, 2018.
- [37] Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo. Coloring With Lim-

- ited Data: Few-shot colorization via memory-augmented networks. In *CVPR*, 2019.
- [38] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.
- [39] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017.
- [40] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017.