Latent-Guided Exemplar-Based Image Re-Colorization

Wenjie Yang[1], Ning Xu, Yifei Fan[2,3]
[1]Shanghai Jiao Tong University, [2]Adobe, [3]Academy of Art University
136334913888@sjtu.edu.cn

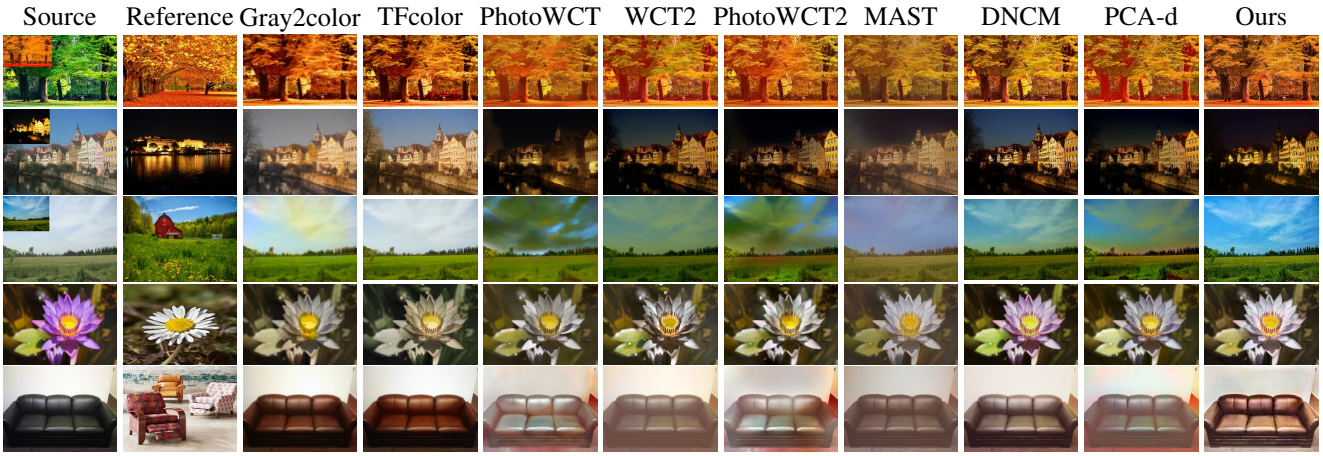

Figure 1. Qualitative comparison of re-colorization with RGB source as input. The sub-figure on the top-left of the source (if exists) is the ground-truth we annotated.

Abstract

Exemplar-based re-colorization transfers colors from a reference to a colored or grayscale source image, accounting for the semantic correspondences between the two. Existing grayscale colorization methods usually predict only the chromatic aberration while maintaining the source's luminance. Consequently, the result's color may diverge from the reference due to such luminance difference. On the other hand, global photorealistic stylization without segmentation cannot handle scenarios where different parts of the scene need different colors. To overcome this issue, we propose a novel and effective method for re-colorization: 1) We first exploit the spatial-adaptive latent space of SpaceEdit in the context of the re-colorization task and achieve re-colorization via latent maps prediction through a proposed network. 2) We then delve into SpaceEdit's self-reconstruct latent codes and maps to better characterize the global style and local color property, based on which we construct a novel loss to supervise re-colorization. Qualitative and quantitative results show that our method outperforms previous works by generating superior outputs with more consistent colors and global styles based on references.

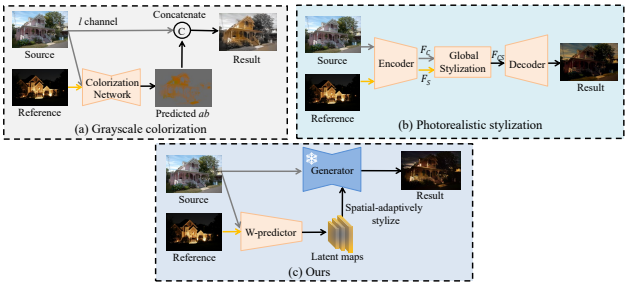

Figure 2. (a) Colorization methods predict *ab* channel only and reuse the *l* channel of source. (b) Photorealistic stylization methods perform global stylization on the features of the source according to reference. (c) Our method predict latent maps to spatial-adaptively stylize the source image.

1. Introduction

For a pair of content-related source and reference images, exemplar-based re-colorization reasonably transfers the color of the reference to the source. Traditional exemplar-based colorization methods primarily target grayscale source images. There is a growing need to re-colorize colored images by references, which is the focus

of our paper. Although, in theory, traditional grayscale colorization methods (e.g., [1, 6, 13, 23, 24]) also apply to colored image re-colorization by simply converting the source images from RGB to grayscale, their design and architectures often limit the performance. As Figure 2(a) shows, they only predict chromatic aberration (*ab* channels) while reusing the source image's luminance (l channel) for the output. As the l channel contains the most significant amount of content information of an image, improperly adjusting it may disrupt the source's content. Consequently, the output's color can become inconsistent with the reference due to luminance differences (i.e., the color of sky).

Another task related to re-colorization is photorealistic image stylization [14], which transfers a reference image's global style (e.g., color, saturation, and brightness) to a source image while keeping the fine-grained content of the source unchanged. Specifically, re-colorization can be interpreted as spatial-adaptive photorealistic image stylization without explicit segmentation, which is more challenging. Existing photorealistic stylization methods [3,4,7,12,14,25] all perform global stylization and cannot directly handle the scenes where different parts of an image need different stylization, thus may lead to unsatisfactory results, as shown in Figure 2(b). Thus, segmentation is usually needed to improve their performance under such scenarios.

Recently, StyleGAN [2, 8] has been widely used for image generating and editing tasks due to its capability to generate highly realistic images. Luo *et al.* [16] utilize StyleGAN for automatic colorization. However, their methods could only support close domain images (i.e., same category). In SpaceEdit [20], a conditional StyleGAN is proposed for open-domain image editing (which is also the generator we adopted). It is a conditional generator conditioned on pixel-level content. Specifically, it takes both image and latent code as input. The former provides content information, and the latter performs global color stylization. The output image has the same content but different color from the input. Inspired by SpaceEdit, we formulate the re-colorization as latent optimization (GAN inversion) of SpaceEdit. However, the solution is not straightforward. First, SpaceEdit's 1d latent space is for global style editing, and it has limitations in our task since re-colorization requires more accurate local-wise color transformation. Second, their latent optimization method only works on content-matched image pairs since L_1 loss is used. However, our task requires latent optimization on image pairs with different contents. To overcome those issues, we first extend the original 1d latent space to a 2d spatial-adaptive format, which could represent more complex color transformations. We then propose a W-predictor network to predict the latent maps for re-colorization. To improve the W-predictor's performance, we further delve into the properties of the self-reconstruction latent codes and maps, studying their representation of the global style and local color information. Based on the properties of self-reconstruction latent codes, we construct a novel loss called $\mathbf{w}_0$ loss to supervise re-colorization.

Another challenge for the re-colorization research is the lacking of the content-matched ground truth of the source images to calculate pixel-wise quantitative metrics. Previous works usually [13, 22] adopt histogram intersection similarity (HIS) to evaluate performance. However, HIS is a global statistic without spatial information and does not reflect local-level colorization accuracy. Bai *et al.* [1] generates augmented image pairs (e.g., cropping) as ground truth for evaluation. However, such pairs do not reflect actual use cases as the content of paired images are highly similar. To better support the research in this domain and evaluate relevant approaches more precisely, we offer ground truth for the Deep Photo Style Transfer (DPST) dataset [15], which is adjusted in Adobe Lightroom. With the upgraded DPST dataset, we further demonstrate that our solution outperforms previous works both qualitatively and quantitatively.

In summary, our contributions are three folds. First, we propose a novel solution for exemplar-based re-colorization based on spatial-adaptive SpaceEdit. In particular, we propose a network to predict latent maps for accurate colorization. second, we delve into the properties of self-reconstruction latent codes and maps to represent images' style and color features, based on which we construct a novel style loss for re-colorization. Finally, our method achieves state-of-the-art performance in transferring color and style between unmatched contents without segmentation maps. We also offer manually adjusted ground truth for the DPST dataset to facilitate more objective quantitative evaluations of the exemplar-based re-colorization.

2. Related Work

Exemplar-based Grayscale Image Colorization. Deep-learning-based methods [1, 6, 13, 23, 26, 29] utilize the features from a pretrained VGG [21] model to capture semantic correspondence between the pair of images. Some [1, 23] first pre-color source images coarsely into RGB and thus make correspondence more accurate. In [23], researchers utilize stylization networks and the Adaptive Instance Normalization (AdaIN) operation to stylize the features of the source image. However, the coarse results are usually unsatisfactory and full of artifacts due to the inaccurate correspondence between the source and reference. Consequently, such methods [1, 6, 23, 26] often require an extra colorization sub-network (usually a U-Net [19]) to refine the coarse results. Recently, other methods [1, 22] leverage transformers [22] for better performance.

Photorealistic Image Stylization. In general, existing stylization operations are global: PhotoWCT [12], WCT2 [25], and PhotoWCT2 [4] are based on the whitening and col-

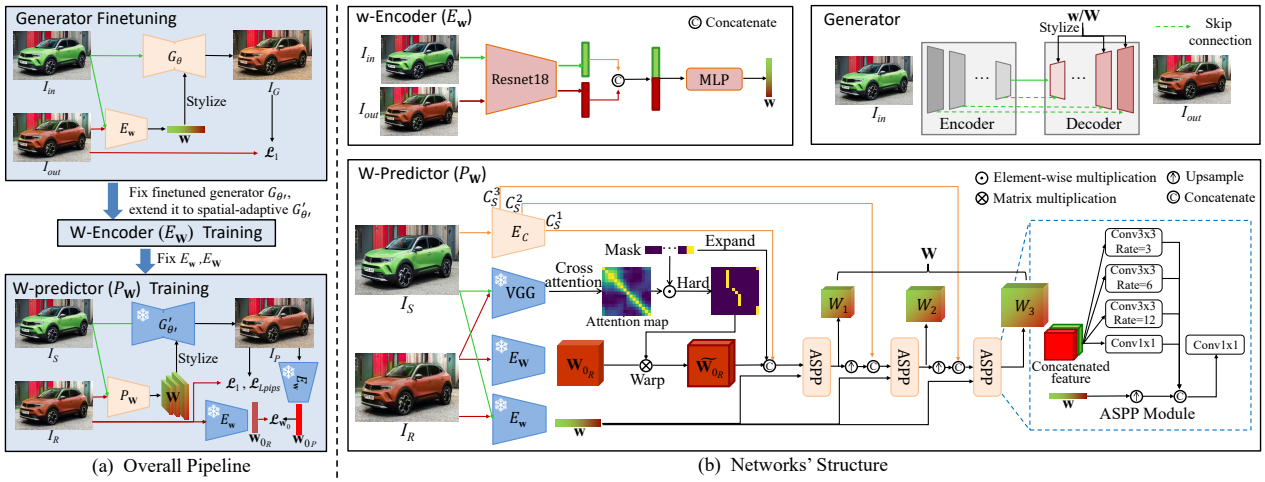

Figure 3. The left is the overall pipeline of our method. The right is the detailed structure of the generator, w-Encoder $E_{\mathbf{w}}$ and W-predictor $P_{\mathbf{W}}$. We first finetune generator (G_θ) and train w-Encoder ($E_{\mathbf{w}}$) jointly. After finetuning, $G_{\theta\prime}$ is fixed and extended to spatial-adaptive format $G'_{\theta\prime}$. $G_{\theta\prime}$ and $G'_{\theta\prime}$ have the same structure but takes as input 1d latent codes and 2d latent maps, respectively. We then train W-Encoder ($E_{\mathbf{W}}$) on fixed $G'_{\theta\prime}$. After that, both $E_{\mathbf{w}}$ and $E_{\mathbf{W}}$ are fixed and set as modules of W-predictor ($P_{\mathbf{W}}$). Finally, we train $P_{\mathbf{W}}$ by image pairs with identical contents. After training, $P_{\mathbf{W}}$ works on images with different contents.

oring transformation (WCT). PCA-d [3] utilizes PCA for style knowledge distillation. MAST [7] learns a global projection matrix $P \in \mathbb{R}^{C\times C}$ to transform the content features into the subspace of style features. DNCM [9] stylizes the source image by multiplying it with an adaptive color mapping matrix. LS-FT [5] introduces a transformation that enables controlling the balance between content preservatoin and style transfer. Among them, although MAST utilizes VGG features to establish the semantic mapping between two images, the global stylization still weakens the performance in scenes where different parts of an image need different stylization (shown in later experiments). Therefore, segmentation is required to ensure their reported performance.

Image Editing via the Latent Space of GANs. After witnessing the breakthrough latent-space disentanglement from StyleGAN2 [8], researchers have been exploring how to utilize the latent space of a pretrained generator for image manipulation. The authors of [18] train an encoder to inverse the image into latent codes. Luo *et al.* [16] utilizes latent optimization of StyleGAN [8] for automatic colorization. Stylemap [10] and SalS-GAN [27] exploit the spatial-adaptive latent space of StyleGAN2 for more flexible image editing. However, those methods only apply to close-domain images within the same object category. SpaceEdit [20] proposes a conditional StyleGAN2 that takes both an image and a latent code as input, which aims at global color style editing for open-domain images. It obtains the latent code representing the style transformation between two images by online latent optimization with the L_1 loss. Unfortunately, their latent optimization only works on pairs with identical contents. In addition, the stylization is global, as the latent space in their work is one-dimensional.

3. Method

3.1. Overview of method

Figure 3 shows the overall pipeline and networks' structure of our methods. In the following, we first introduce the SpaceEdit generator and our adaptation on it (i.e., finetuning and extension to 2d spatial-adaptive latent space) in Sec. 3.2. We then introduce self-reconstruct latent codes and maps and analyze their properties in Sec. 3.3, which benefits the latent map prediction. Finally, we elaborate on the detailed structure and training process of our proposed latent prediction network (W-predictor) in Sec. 3.4.

3.2. Generator structure and adaptation

3.2.1 Generator Structure

Our generator structure and initial weights follow that of SpaceEdit [20]. SpaceEdit aims for global color style editing (e.g., tone, saturation, and brightness). Specifically, the edited result should have a different color style but the same content as the input image. To satisfy the requirement, the authors proposed a conditional StyleGAN2 generator, whose structure is shown in the top-right of Fig 3. It takes both image and latent code as input. To preserve most fine-grained details of the input image, the authors add skip-connection from the encoder's layers to the de-

coder's corresponding layers. SpaceEdit is pretrained on the Discover[1] dataset, which contains 60k pairs of before- and after-adjustment photos.

3.2.2 Generator Finetuning

SpaceEdit's latent space is for global style editing. To obtain a better latent representation of color transformations in the re-colorization task, we first finetune the generator with image pairs of the Discover dataset. We do random color augmentation to simulate more color transformations. The finetune process is shown in Fig. 3. Specifically, a w-Encoder $E_{\mathbf{w}}$ takes as input an image pair (I_{in}, I_{out}) with the same content but different colors and predicts the latent code $\mathbf{w}$ representing the color transformation from I_{in} to I_{out} (i.e., green→red in Figure 2). The generator G then generates an output $I_G = G_\theta(I_{in}, E_{\mathbf{w}}(I_{in}, I_{out}))$ to match the target output I_{out}. We train generator G_θ and w-Encoder $E_{\mathbf{w}}$ jointly by minimizing the L_1 loss between I_G and I_{out}. Once finetuning is completed, both $G_{\theta'}$ and $E_{\mathbf{w}}$ are fixed, and we obtain a new latent space for re-colorization.

3.2.3 Spatial-adaptive latent space

After finetuning, we extend the latent space from 1d to spatial-adaptive 2d as in [27]. Previous works [10, 27] have revealed the advantage of 2d space over 1d space. The spatial information brings the potential to represent more complex color transformations. Specifically, our customized spatial-adaptive generator $G'_{\theta'}$ takes an image and a series of 2d latent maps $\{W^1, W^2, ..., W^n\}$ ($n = 20$ is the number of layers in the decoder) with a pyramid of various input resolutions. For each layer, the corresponding W^k is first resized to the same resolution as the input feature x, then converted to a style map m with the same resolution and channel as x by an affine transformation. Next, we perform spatial-adaptive modulation on an input feature x:

$$x'_{ij} = m_{ij} \cdot x_{ij} \tag{1}$$

where x' are the features after modulation, and i, j are channel and spatial indices of the features, respectively. After that, the colorization task is formulated as predicting optimal latent maps to re-colorize the source. We achieve this via a proposed network called W-predictor ($P_{\mathbf{W}}$).

3.3. Self-reconstruct latent codes and maps

Before elaborating on the W-predictor, one interesting question to consider is: can we find a proper representation for global color style via the latent space of SpaceEdit? In the original StyleGAN, the latent codes represent the style (i.e., pose, expression, and appearance in face generation) of generated images. In SpaceEdit, however, the latent codes represent the color transformation (i.e., green→red). Fortunately, we found that the **self-reconstruct latent codes** can represent the global color style. Specifically, an image's self-reconstruction code is a 1d latent code $\mathbf{w}_0$ with which the generator can reconstruct the image itself (i.e., $I = G_{\theta'}(I, \mathbf{w}_0)$). It can be obtained from our trained w-Encoder by feeding two identical images as $\mathbf{w}_0 = E_{\mathbf{w}}(I, I)$. A critical difference between our trained self-reconstruction code and the one in SpaceEdit [20] is that theirs came from iteratively optimizing L_1 loss online: $\mathbf{w}_0 = \arg\min_{\mathbf{w}} \mathcal{L}_1(I, G_\theta(I, \mathbf{w}))$. Moreover, they only used it for style interpolation.

Figure 4. Clustered images according to $\mathbf{w}_0$ and color histogram, respectively. The values below are the corresponding similarity scores. Our $\mathbf{w}_0$ better characterizes global styles.

We reveal that the self-reconstruct code $\mathbf{w}_0$ represents the global style by clustering images according to the cosine similarity of $\mathbf{w}_0$. Figure 4 illustrates that images with similar global styles (i.e., low key) have closer self-reconstruct codes, indicating that $\mathbf{w}_0$ can effectively characterize global styles. To explain this, we argue that $\mathbf{w}_0$ only ensures existing colors in the image are correctly mapped to themselves while ignoring the missing colors. We verify the hypothesis by reconstructing 1k images with its own $\mathbf{w}_0$ or $\mathbf{w}_0$ from other images and calculate the mean absolute error (MAE) at a scale of 255. The error is 1.89 with their own $\mathbf{w}_0$ and 2.85 with ones from others. Moreover, we find that for reconstructing an individual image with $\mathbf{w}_0$ from another image, the error is lower when those images have a similar style to the individual image, as shown in supplementary.

Based on the properties of $\mathbf{w}_0$, we propose a novel $\mathbf{w}_0$ **loss** $\mathcal{L}_{\mathbf{w}_0}$ to measure the similarity of global style between reference image I_R and predicted image I_P:

$$\mathcal{L}_{\mathbf{w}_0}(I_R, I_P) = \|\mathbf{w}_{0_R} - \mathbf{w}_{0_P}\|_1 \tag{2}$$

in which $\mathbf{w}_{0_R}$ and $\mathbf{w}_{0_P}$ are the self-reconstruction codes of I_R and I_P, respectively. We use $\mathcal{L}_{\mathbf{w}_0}(I_P, I_R)$ as one of the training losses for W-predictor to force I_P's global style close to I_R's.

Compared to $\mathbf{w}_0$, the color histogram used in previous colorization work [13, 22] appears less accurate for representing global styles because it is low-level statistical data and is weighted by the area of different parts, as shown in Fig. 4. Later experiments also show the advantage of our $\mathbf{w}_0$ loss over color histogram loss [13, 22].

[1] https://lightroom.adobe.com/learn/discover

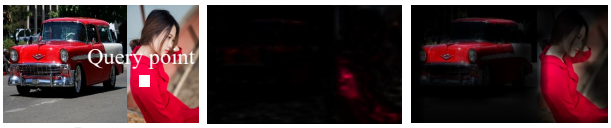

Figure 5. Comparison of the heatmaps of VGG and $\mathbf{W}_0$ features (brighter means higher similarity). Our $\mathbf{W}_0$ features better capture local color.

We further extend $\mathbf{w}_0$ to a spatial-adaptive 2d format $\mathbf{W}_0$ (i.e.,$I = G'_{\theta'}(I, \mathbf{W}_0)$) and train a W-Encoder network $E_{\mathbf{W}}$ to predict $\mathbf{W}_0$ for an input image I. The W-Encoder $E_{\mathbf{W}}$ has a ResNet-18 backbone and outputs the feature map $\mathbf{W}_0$ (with resolution 1/16) from layer3. $\mathbf{W}_0$ is then resized to different resolution to constitute latent maps $\{W^1, W^2, ..., W^n\}$ of $G'_{\theta'}$. We calculate the L_1 loss between I and the reconstructed image $G'_{\theta'}(I, E_{\mathbf{W}}(I))$ to train the W-Encoder $E_{\mathbf{W}}$, during which the generator $G'_{\theta'}$ is fixed, as Fig 3 shows.

We now demonstrate that $\mathbf{W}_0$ can characterize the color features of local regions by visualizing the pairwise element self-similarity heatmap of $\mathbf{W}_0$ for an image I, which contains various types of objects in the same color. For a query point i on I, its heatmap H is calculated by

$$H(j) = \langle \mathbf{W}_0(i), \mathbf{W}_0(j) \rangle \tag{3}$$

in which $\langle \cdot, \cdot \rangle$ is the cosine similarity. Figure 5 shows that regions with similar colors have closer $\mathbf{W}_0$s, indicating $\mathbf{W}_0$ characterizes the color of local regions. Therefore, $\mathbf{W}_0$ can be used to regulate the color transfer of local regions.

Finally, we compare the heatmap of $\mathbf{W}_0$ to that of low-level VGG features (i.e., $relu2\_2$), which is often used to build context loss [17] for color transfer in previous work. As shown in Figure 5, the heatmap of VGG features only focuses on the cloth without considering the car, which has the same red color as the query. As VGG is trained for classification, the color information is dominated by semantic and texture information.

3.4. Latent maps prediction by W-predictor

Given a pair of *content-related* but *pixel-unaligned* source I_S and reference I_R, our goal is to predict an optimal $\mathbf{W}$ that re-colorizes I_S as $I_P = G'_{\theta'}(I_S, \mathbf{W})$. The result I_P should satisfy the following requirements: 1) Semantic-related regions in I_P and I_R should have close colors, 2) I_P and I_R should have close global style, and 3) I_P shall have no artifact. Compared to 1d latent code prediction, 2d latent map prediction is more challenging as the predicted latent maps should also meet a continuity constraint to ensure that adjacent patches of an object with similar colors are transformed similarly and smoothly; otherwise, there will be severe artifacts, as shown in later experiments. To achieve that, we propose a novel W-predictor network $P_{\mathbf{W}}$ to predict the optimal $\mathbf{W}$ for I_S and I_R: $\mathbf{W} = P_{\mathbf{W}}(I_S, I_R)$. The structure of $P_{\mathbf{W}}$ is shown in Figure 3. In subsequent sections, we elaborate on the detailed design and training strategy of $P_{\mathbf{W}}$.

3.4.1 Multi-source feature extraction

Different from previous works [4, 25] that use a shared model to extract the color feature of I_S and I_R, we use different models for I_S and I_R separately. Specifically, we use E_C^S (ResNet-18) to extract multi-level color features $\{C_S^1, C_S^2, C_S^3\}$ of I_S (with resolution 1/4, 1/8 and 1/16 of I_S) and fixed $E_{\mathbf{W}}$ to extract a single-level color feature $\mathbf{W}_{0_R}$ (with 1/16 of I_R) of I_R. The last section has proved that $\mathbf{W}_{0_R}$ characterizes the local color feature. By design, the reference's feature $\mathbf{W}_{0_R}$ only contains color information, while the source's features $\{C_S^1, C_S^2, C_S^3\}$ also carry fine-grained semantics such as texture and boundaries for better predicting $\mathbf{W}$.

In addition to color feature maps of I_S and I_R, we use $E_{\mathbf{w}}$ to extract latent code $\mathbf{w}$ from I_S to I_R, which can represent the global style difference between the two and help better transfer the global style.

3.4.2 Hard sparse semantic correspondence

We extract vgg features (i.e., $relu5\_1$) with a pretrained VGG16 model to find semantic correspondence between I_S and I_R. Let F_S and F_R denote the vgg features of I_S and I_R, respectively. After feature extraction, F_S, $\mathbf{W}_{0_R}$, and C_S^3 have the same size (1/16 of the original image). We first calculate the cross attention matrix A_C, a pairwise cosine similarity matrix of F_S and F_R.

$$A_C(i, j) = \langle F_S(i), F_R(j) \rangle \tag{4}$$

where i and j are the spatial indices of F_R and F_S.

To increase the task difficulty during training, we propose a **random semantic masking** method to mask some patches of the reference image randomly. Specifically, we mask the cross attention matrix A_C instead of directly zeroing out patches of I_R, which might be unrealistic. we achieve this by element-wise multiplying each row of A_C with the random mask M (1× HW) as $A_C = A_C \odot M$. Such random masking makes correspondence less accurate and forces the model to combine the information from neighboring patches to predict the required color transformation for the masked patch, increasing the model's robustness on content-different pairs during testing. The masking is disabled at test time to enable more accurate dense correspondence.

To avoid the smoothing and blurring effect (i.e., averaged colors) mentioned in [7], we further convert the masked A_C

to a hard binary matrix A_h via **hard activation** operation, which only preserves the location j with the highest similarity while zeroing out weights of other locations.

$$A_h(i,j) = \begin{cases} 1, & j = \arg\max_k A_C(i,k) \\ 0, & \text{else} \end{cases}. \quad (5)$$

3.4.3 Multi-level latent maps prediction

To spatially align the color features from the source and reference, we matrix multiply $\mathbf{W}_{0_R}$ by A_h to obtain a warped color feature $\widetilde{\mathbf{W}_{0_R}}$, which is expected to contain both reference's color and source's spatial information. The features $\widetilde{\mathbf{W}_{0_R}}, C_S^1$ and mask M are first concatenated channel-wise and then fed into the first ASPP together with $\mathbf{w}$ to predict W_1. We replace the globally averaged feature in ASPP with our $\mathbf{w}$ to better guide the global style transfer. The output features of the first ASPP are concatenated with C_S^2 and then fed into the second ASPP together with $\mathbf{w}$ to predict W_2. The same operation is repeated for the third ASPP to predict W_3. After that, we predict multiple-level latent maps $\mathbf{W} = \{W_1, W_2, W_3\}$ with resolution 1/16, 1/8 and 1/4, respectively, of the original image. These latent maps are connected to different layers of the generator's decoder for stylization: inspired by [18], W_1 is resized to fill $W^1 - W^9$, W_2 to $W^{10} - W^{12}$, and W_3 to $W^{13} - W^{20}$). Multiple-level latent maps help colorization at finer scales, and our customized ASPP module is used to increase the receptive field.

3.4.4 Optimization

To train the W-predictor ($P_{\mathbf{W}}$), we adopt three losses: L_1 loss ($\mathcal{L}_1$) , Lpips loss [28] ($\mathcal{L}_{Lpips}$) and the proposed $\mathbf{w}_0$ loss ($\mathcal{L}_{\mathbf{w}_0}$) between I_P and I_R. L_1 loss provides the most fine-grained supervision, Lpips loss helps mitigate the difference between I_R and I_P in human perception, and $\mathbf{w}_0$ loss forces the global style of I_P close to I_R. The overall loss for optimization is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_{Lpips} \mathcal{L}_{Lpips} + \lambda_{\mathbf{w}_0} \mathcal{L}_{\mathbf{w}_0} \quad (6)$$

in which λ_1, λ_{Lpips} and $\lambda_{\mathbf{w}_0}$ are the weights for different loss terms.

4. Experiments

4.1. Implementation details

4.1.1 Default settings

The default size of images in our experiments is 256× 256. The channel dimension for latent code/map is set as 512. We use 1 NVIDIA RTX A4000 for experiments. The optimizer we used is the Adam optimizer [11]. During training, the parameters for the Adam optimizer are set as $\alpha = 0.0025, \beta_1 = 0, \beta_2 = 0.99$.

4.1.2 Training Details

We elaborate the training details of W-predictor. The details of generator finetuning and W-Encoder training are documented in the supplementary. We use two types of paired data to train W-predictor: (1) Image pairs from the Discover dataset and (2) synthesized image pairs augmented from the COCO dataset. With the help of segmentation annotations in COCO, we perform random local and global color augmentation to generate image pairs with different colors. The method and parameters for color augmentation are the same as in generator finetuning. The weights $\lambda_{\mathbf{w}_0}, \lambda_{Lpips}, \lambda_1$ in training loss (equation 6) are all set as 1. The batch size is set as 8. It takes nearly 60k steps and 12 hours to train W-predictor. To establish a curriculum that controls the training process from easy to difficult, a maximal mask ratio α is initialized as 0 and increases gradually during training until it reaches the upper bound of 0.95. We randomly select a mask ratio from [0, α] at each training step.

4.2. Comparison with previous methods

4.2.1 Qualitative comparison

We compare our method to previous work, including grayscale colorization methods Gray2color [13], TFcolor [24] and photorealistic stylization methods such as PhotoWCT [12], WCT2 [25], MAST [7], PhotoWCT2 [4], DNCM [9] and PCA-d [3]. For a fair comparison, segmentation is not used in photorealistic stylization methods because our method does not require segmentation maps.

Figure 1 shows the qualitative re-colorization results (with RGB source). For grayscale colorization methods (i.e., Gray2color and TFcolor), the RGB sources are first converted to grayscale. Compared with results from previous methods, ours have two significant advantages: (1) superior consistency with the reference's color and global styles and (2) higher quality with the slightest artifacts and more details, both of which benefit from our spatial-adaptive latent space in representing more complex color transformation and the effectiveness of our W-predictor. For results from Gray2color and TFcolor, the color and global styles are not consistent with reference as it keeps luminance unchanged (especially for the 2nd and 4th rows, where the luminance of the two colors has a huge difference). Photorealistic stylization methods can achieve satisfying results in simpler scenes where different regions share a similar style and color (the 1st row). However, for more difficult scenes (e.g., the 3rd row), most of them suffer from severe artifacts. Without an explicit segmentation map, the same color transformation (i.e., get greener) is applied to both the sky and grass. Finally, results from MAST

Table 1. Quantitative comparison of previous methods and our method. The first group is grayscale colorization, "Ours (gray)" means our method with grayscale source inputs. The second group is re-colorization with RGB sources as input.

Methods	Gray2color	TFcolor	Ours (gray)	PhotoWCT	WCT2	MAST	PhotoWCT2	DNCM	PCA-d	Ours
MAE↓	60.64	63.2	**28.52**	35.92	33.93	40.03	32.98	33.64	31.44	**27.34**
LPIPS↓	0.3311	0.3642	**0.2574**	0.3418	0.3478	0.3169	0.2712	0.2455	0.2663	**0.2264**
Delta E↓	22.82	23.44	**16.71**	18.49	24.72	17.97	18.57	27.58	20.55	**17.20**
PSNR↑	12.4228	11.64	**17.38**	15.2565	16.0117	15.0397	16.197	16.74	16.72	**17.928**

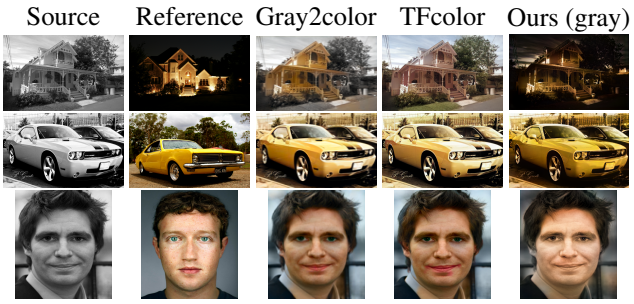

Figure 6. Qualitative comparison of grayscale colorization. "Ours (gray)" is our method with grayscale sources as inputs.

look hazy and inconsistent with references, though it applies VGG to find semantic correspondence, demonstrating the limitation of global stylization operations. Only with additional segmentation maps can they reach a similar performance as ours, which is presented in the supplementary.

Although our method focuses on re-colorization for colored images, it can also be applied to grayscale images colorization as Gray2color [13] and TFcolor [24] to make a more fair comparison. Figure 6 shows that our result with grayscale source input still outperforms previous grayscale colorization methods by achieving more consistent color and style with reference, which verifies the advantage and necessity of changing luminance.

4.2.2 Quantitative comparison

To make a more objective comparison, we conduct a quantitative evaluation of different methods based on the provided annotations of DPST. We adopt four metrics to evaluate: mean absolute error (MAE), LPIPS, PSNR, and Delta E. LPIPS evaluates the difference between output images and human perception. Delta E quantifies the difference between the color of the result and the ground truth. The quantitative result is shown in Table 1, in which the methods in the first group are with grayscale sources as input, and the methods in the second group are with RGB sources as input. Results show that our methods outperform previous methods on these metrics by a large margin. Our results with grayscale sources are slightly inferior to those with RGB sources. We believe the reason is that the chromatic aberration of the source is vital to provide more details and establish more accurate correspondence.

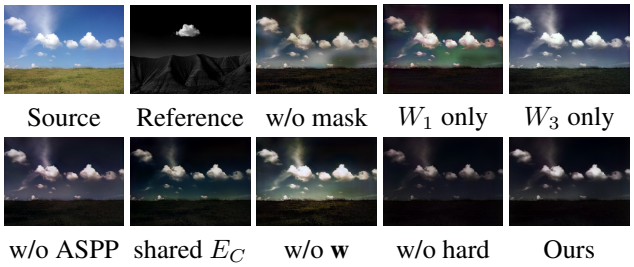

Figure 7. Qualitative ablation study on W-predictor's design and components. *w/o mask*: training without random semantic masking. W_1 or W_3 only: predicting only a single latent map with 1/16 or 1/4 of the original image's resolution and resizing it according to the required input resolution of the generator's decoder layers. *w/o ASPP*: replacing ASPP modules in W-predictor with $3\times$ 3 convolution layers, which reduces the reception. *shared* E_C: $E_{\mathbf{w}}$ is not used, instead, shared model E_C is used for source and reference. *w/o* **w**: $E_{\mathbf{w}}$ module is not used, and the feature **w** is replaced by the global average of the concatenated features (i.e., the other input) in each ASPP module. *w/o hard*: The hard activation operation is replaced by $softmax$.

4.3. Ablation Study

4.3.1 Structure of W-predictor

We first evaluate the effectiveness of the crucial design and components of the W-predictor by dropping or varying one at a time. The qualitative results are shown in Figure 7. Specifically, training without random semantic masking leads to discontinuity in outputs as the model struggles to handle pairs with unmatched contents. The W_1 only configuration results in block artifacts due to insufficient resolution from W_1 while W_3 only configuration lead to the inconsistent global style, which verifies the advantage of multi-layer latent maps for **W**. Using a shared model for source and reference or replacing hard activation by $softmax$ leads to inaccurate colorization at the local region (i.e., cloud). In addition, for W-predictor without feature **w**, styles of different regions in output are no longer consistent. Replacing ASPP module by $3\times$ 3 convolution layers also leads to inconsistent global style due to the reduced reception. We also conduct a quantitative ablation study in Table 2. Among different settings, "Ours" achieves the best

Table 2. Quantitative ablation study. The meaning of each setting is the same as that in Figure 7 and 8.

Methods	w/o mask	W_1 only	W_3 only	w/o ASPP	shared E_C	w/o $\mathbf{w}$	w/o hard	w/o $\mathcal{L}_{\mathbf{w}_0}$	$\mathcal{L}_{histogram}$	Ours
MAE↓	33.73	29.92	35.28	29.19	28.49	29.42	29.94	30.06	29.71	**27.34**
LPIPS↓	0.2871	0.2681	0.2570	0.2378	0.2412	0.2466	0.241	0.2541	0.2343	**0.2264**
DeltaE ↓	17.69	17.78	19.75	17.74	17.21	18.09	17.3	18.39	17.86	**17.2**
PSNR↑	15.74	17.1	16.24	17.67	17.41	17.32	17.2	17.34	17.18	**17.93**

Source Reference w/o $\mathcal{L}_{\mathbf{w}_0}$ $\mathcal{L}_{histogram}$ Ours

Figure 8. Qualitative ablation study on training loss of W-predictor. *w/o* $\mathcal{L}_{\mathbf{w}_0}$: training without the proposed $\mathbf{w}_0$ loss. $\mathcal{L}_{histogram}$: Replace $\mathbf{w}_0$ loss by color histogram loss.

Table 3. Reconstruct error for different generators.

Generator	G_θ	$G_{\theta'}$	$G'_{\theta'}$
Error(MAE)↓	11.2	8.68	**5.06**

qualitative and quantitative performance, which verifies the advantages of our design and the proposed components.

4.3.2 Training loss for W-predictor

We then evaluate the effectiveness of our proposed $\mathbf{w}_0$ loss. We also compare it to the histogram loss used in previous work. The quantitative result is shown in Figure 8, removing proposed $\mathbf{w}_0$ loss or replacing it with color histogram loss will lead to inconsistent global style between result and reference. And the quantitative ablation is also shown in Table 2. "Ours" achieves the best qualitative and quantitative performance, which verifies the effectiveness of our proposed loss.

4.3.3 Effectiveness of latent space

To verify the advantage of our generator adaptation (i.e., finetuning and spatial-adaptive extension). We conduct the experiment to evaluate the effectiveness of the latent space of different generators (i.e., original SpaceEdit generator G_θ, finetuned generator $G_{\theta'}$ and spatial-adaptive finetuned generator $G'_{\theta'}$). We reconstruct 1,000 color-augmented image pairs (which are generated with the same method during finetuning but on the validation set of Discover) with different generators. The reconstruction follows the online scheme in SpaceEdit [20]. And we calculate the mean reconstruct error (MAE) for all image pairs, as shown in Table 3. A lower reconstruct error means the generator has a stronger latent space to represent more accurate color transformations. The finetuned generator $G_{\theta'}$ achieves lower

Source Reference $\mathbf{W}_{0_R}$ $\mathbf{w}'(\alpha=1)$ $\mathbf{w}'(\alpha=2)$ Both

Figure 9. Effectiveness of $\mathbf{W}_{0_R}$ and $\mathbf{w}$ in W-predictor.

MAE than the original G_θ. By converting $G_{\theta'}$ to spatial-adaptive format $G'_{\theta'}$, the MAE is reduced further, which verifies the advantages of our generator adaptation.

4.3.4 Effectiveness of features in W-predictor

We evaluate the effectiveness of used features $\mathbf{W}_{0_R}$ and $\mathbf{w}$ in W-predictor separately. To evaluate the effect of $\mathbf{W}_{0_R}$, we replace $\mathbf{w}$ by self-reconstruct code $\mathbf{w}_{0_S}$ of I_S to eliminate its effect. As the "$\mathbf{W}_{0_R}$" of Figure 9 shows, the color of the foreground (cloth) is correctly transferred. However, the global style (contrast, brightness) in the background is not so close to I_R. To evaluate the effect of $\mathbf{w}$, we replace $\mathbf{W}_{0_R}$ by self-reconstruct map $\mathbf{W}_0$ of I_S (i.e.,$\mathbf{W}_{0_S}$) to eliminate its effect. We also try to enlarge the effect of $\mathbf{w}$ by interpolation with $\mathbf{w}_{0_S}$ (i.e., $\mathbf{w}' = \mathbf{w}_{0_S} + \alpha(\mathbf{w} - \mathbf{w}_{0_S})$, and α is to used to control the strength). As "$\mathbf{w}'(\alpha = ...)$" of Figure 9 shows, with α increasing, the global style is closer and closer to I_R; however, the color of the cloth does not change. When combining $\mathbf{W}_{0_R}$ and $\mathbf{w}$, both the local color and global style of the result are consistent with I_R.

5. Conclusion

In this paper, we exploit the idea of exemplar-based image re-colorization via spatially-adaptive SpaceEdit. We design a novel and effective network to predict proper latent maps to colorize accurately. We then delve into the property of latent self-reconstruction code in conditional StyleGAN and utilize it to construct novel losses to improve performance further. Finally, our method achieves SOTA performance among both colorization and photorealistic stylization methods.

Acknowledgement. The paper is supported in part by the National Natural Science Foundation of China (No. 62325109, U21B2013, 61971277).

References

[1] Yunpeng Bai, Chao Dong, Zenghao Chai, Andong Wang, Zhengzhuo Xu, and Chun Yuan. Semantic-sparse colorization network for deep exemplar-based colorization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 505–521. Springer, 2022.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Tai-Yin Chiu and Danna Gurari. Pca-based knowledge distillation towards lightweight and content-style balanced photorealistic style transfer models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7844–7853, June 2022.

[4] Tai-Yin Chiu and Danna Gurari. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2868–2877, 2022.

[5] Tai-Yin Chiu and Danna Gurari. Line search-based feature transformation for fast, stable, and tunable content-style control in photorealistic style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 249–258, January 2023.

[6] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Trans. Graph.*, 37(4), jul 2018.

[7] Jing Huo, Shiyin Jin, Wenbin Li, Jing Wu, Yu-Kun Lai, Yinghuan Shi, and Yang Gao. Manifold alignment for semantically aligned style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14861–14869, 2021.

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

[9] Zhanghan Ke, Yuhao Liu, Lei Zhu, Nanxuan Zhao, and Rynson WH Lau. Neural preset for color style transfer. *arXiv preprint arXiv:2303.13511*, 2023.

[10] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 852–861, 2021.

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

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

[13] Peng Lu, Jinbei Yu, Xujun Peng, Zhaoran Zhao, and Xiaojie Wang. *Gray2ColorNet: Transfer More Colors from Reference Image*, page 3210–3218. Association for Computing Machinery, New York, NY, USA, 2020.

[14] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4990–4998, 2017.

[16] Xuan Luo, Xuaner Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, and Steven M Seitz. Time-travel rephotography. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[17] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[18] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2287–2296, June 2021.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[20] Jing Shi, Ning Xu, Haitian Zheng, Alex Smith, Jiebo Luo, and Chenliang Xu. Spaceedit: Learning a unified editing space for open-domain image color editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19730–19739, June 2022.

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

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[23] Zhongyou Xu, Tingting Wang, Faming Fang, Yun Sheng, and Guixu Zhang. Stylization-based architecture for fast deep exemplar colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[24] Wang Yin, Peng Lu, Zhaoran Zhao, and Xujun Peng. Yes," attention is all you need", for exemplar based colorization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2243–2251, 2021.

[25] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[26] Bo Zhang, Mingming He, Jing Liao, Pedro V. Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar-based video colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Lingyun Zhang, Xiuxiu Bai, and Yao Gao. Sals-gan: Spatially-adaptive latent space in stylegan for real image embedding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5176–5184, 2021.

[28] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] Hengyuan Zhao, Wenhao Wu, Yihao Liu, and Dongliang He. Color2embed: Fast exemplar-based image colorization using color embeddings. *arXiv preprint arXiv:2106.08017*, 2021.