Supplementary Materials

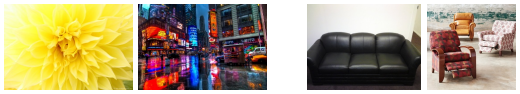

Figure 1. Pairs with ambiguous semantic correspondence in DPST. For each pair, the left is source, the right is reference.

Figure 2. Some annotation examples for DPST. For each group, the top-left, top-right and bottom are source, reference, and ground-truth, respectively.

A. More details of Annotation on DPST

We adopt a commonly-used public dataset DPST [5] for annotation instead of cherry-picking image pairs with better results. The original DPST has 59 pairs of images. To ensure the objectiveness of our annotation, we first discard some pairs that have ambiguous semantic correspondence, which may lead to multiple colorization solutions, such as the pairs in Figure 1. The rest 53 pairs have definite semantic correspondence, leading to definite optimal solutions. In total, we provide ground-truth annotations at three levels: *basic*, *global*, and *local*. At the basic level, adjustments are made using only the Light and Color sliders. At the global level, all sliders in the Edit panel are allowed. As for the local level, segmentation is introduced to achieve local edits. The leveled ground truth annotations could be used to evaluate various stylization and colorization tasks. In this work, we evaluate and compare methods based on annotations at the local level, as also mentioned in the main paper. Figure 2 shows more examples of our annotations at the local level. Note that the annotations are isolated from algorithm design: our method is settled beforehand, and the annotators were unaware of our method during annotation, which assures the objectiveness of our annotation.

B. More training details

- Generator finetuning: We use image pairs from the Discover dataset to finetune the generator and train w-Encoder jointly. We use the ColorJitter function of PyTorch with brightness = contrast = saturation = (0.5, 2), and hue = (-0.2, 0.2) for random color augmentation on discover to enlarge the color difference of image pairs and simulate more color transformations. The batch size is 8. It takes nearly 25k steps and 6 hours to finetune.

- W-Encoder training: Any image dataset without annotation can be used to train W-Encoder since it does not require paired images. For convenience, we use the Discover dataset. The batch size is set as 16. It takes nearly 6k steps and 2 hours to train W-Encoder.

C. More experiments

C.1. User Study

We also conduct a user study for different methods' recolorization results (with RGB source as input) on DPST datasets. To ensure the objectiveness of our user study, we first discard one pair (i.e., the left pair of Figure 1, since the contents of the images in this pair are extremely mismatched, which makes it hard for users to judge). The right pair of Figure 1 is kept, although the semantic correspondence is uncertain (i.e., one-to-many), but the color of the sofa in a desirable result should lie in the colors of the sofas in reference. Then we conduct our user study on all 58 pairs instead of cherry-picked image pairs. We select 5 representative previous methods with relatively stronger performance (i.e., Gray2color [4], PhotoWCT [3], WCT2 [7], MAST [2] and PhotoWCT2 [1]) as well as our method for the user study. Moreover, to lighten users' workload and make results more reliable, we randomly divide 58 pairs into two subsets, each containing 29 pairs. We repeat the dividing operation 7 times and get 14 subsets in total. The subsets are then presented to 14 users (one subset for each). The users are asked to choose the best one from the results of different methods, which are presented anonymously in a random order, based on the following criterion: (1) The proximity of global style and color of semantic-related regions between reference and result (2) The degree of artifacts. The former is our top priority. Finally, we calculate the user-assigned scores of the results from different methods. As reported in Table 1, our method also outperforms previous methods by a large margin.

C.2. More analysis of self-reconstruct latent code

As mentioned in Sec 3.3 of the main paper, 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. An example is shown in Figure 4. For images with a similar style (i.e., setting sun) as the source, their $\mathbf{w}_0$ is more similar to the source. Additionally, when using their $\mathbf{w}_0$ to reconstruct the source, the reconstruction error is lower, further demonstrating the relation between $\mathbf{w}_0$ and global style.

Table 1. Statistics of the user study on DPST results (%)

Methods	Gray2color	PhotoWCT	WCT2	MAST	PhotoWCT2	Ours
Score ↑	1.7	8.1	8.1	2.7	24.9	**54.5**

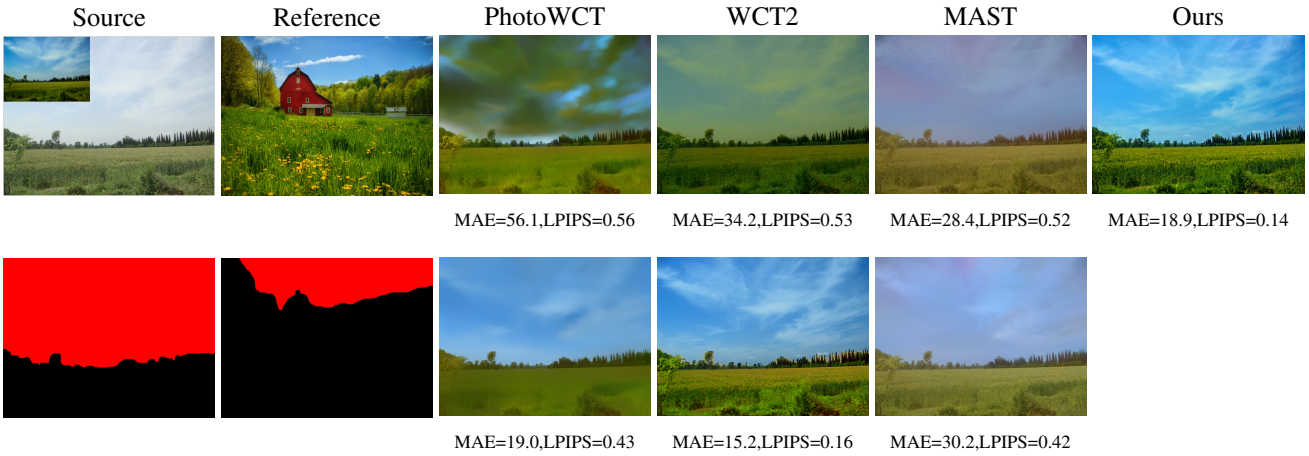

Figure 3. Results of photorealistic stylization methods without and with segmentation. The 1st row contains results without segmentation, and the sub-figure on the top-left of the source is our annotated ground truth. The 2nd row shows results of photorealistic stylization methods with segmentation.

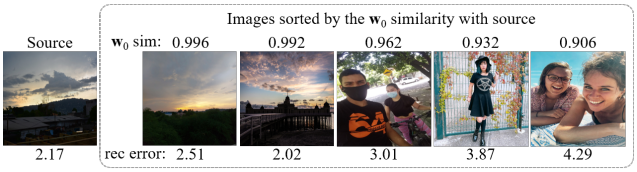

Figure 4. Reconstructing an individual image (source) with $\mathbf{w}_0$ from another image: the values above each image are the cosine similarities between its $\mathbf{w}_0$ and the source's $\mathbf{w}_0$, and the value below is the reconstruction error (at a scale of 255) of source by the $\mathbf{w}_0$ from that image.

Table 2. Quantitative ablation study of training dataset.

Training Dataset	COCO	Discover	COCO+Discover
MAE↓	28.91	27.8	**27.34**
LPIPS ↓	0.2561	0.2267	**0.2264**

C.3. Influence of training dataset

Finally, we conduct quantitative ablation for the training dataset of the W-predictor. As shown in Table 2, Discover has a more significant influence on the performance of the W-predictor. We argue the reason is that the image pairs of Discover contain more various color transformations. With COCO only to train, our methods still achieve competitive performance.

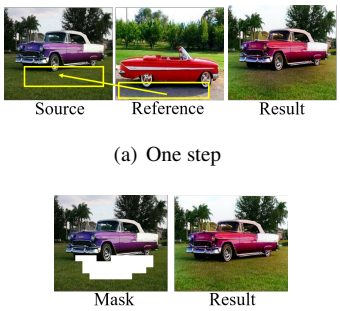

Figure 5. Comparison between one step dense correspondence and two step.

C.4. Comparison to photorealistic stylization works with segmentation

As mentioned in the main paper, previous photorealistic stylization works require segmentation maps to ensure performance. As shown in Figure 3, under such difficult pairs, their performance is ensured only with the help of segmentation.

C.5. Failure case and further improvement

In our work, the semantic correspondences between source and reference are based on the cross-attention of high-level VGG features, which may be inaccurate. Although the random mask strategy we adopted during train-

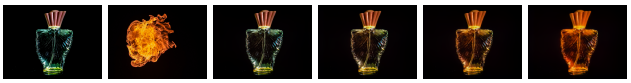

Figure 6. Result of interpolation.

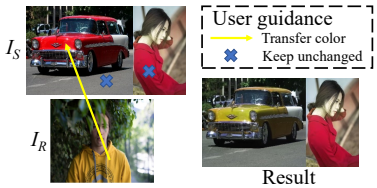

Figure 7. User-guided colorization with our method.

ing can help significantly alleviate the effect of inaccurate correspondence, our result is less satisfying in a few scenes a severe correspondence mistake exists. As shown in 5 (a), the color of the shadow under the car of reference is transferred to the grass under the car of the source by mistake due to the inaccurate correspondence by the VGG feature. Instead of utilizing a more accurate model for correspondence to improve. Here we reveal the two-step scheme proposed in the user-guided colorization has the potential to improve the performance of our method under such scenes. In detail, we decompose the one-step cross-attention into two steps: *sparse to dense* and *coarse to fine*. For the first step, we manually mask some false corresponded regions of the source (i.e., the grass under the car) and only calculate cross-attention for other regions. Then for the second step, we use self-attention to fill the masked regions. In detail, the warped color feature $\widetilde{\mathbf{W}_{0_R}}$ is calculated by

$$\widetilde{\mathbf{W}_{0_R}} = Hard(A_S \odot M.T)\widehat{\mathbf{W}_{0_R}}, \widehat{\mathbf{W}_{0_R}} = Hard(M \odot A_C)\mathbf{W}_{0_R} \quad (1)$$

in which A_C, A_S and M ($HW \times 1$) are cross-attention matrix, self-attention matrix and mask, respectively. Note that we use high-level VGG feature (i.e., $relu5\_1$) to calculate cross-attention while using low-level VGG feature (i.e., $relu3\_1$) to calculate self-attention, as the former mainly characterizes high-level semantic information such as category, while the latter mainly characterizes low-level semantic information such as texture. The texture of two objects with the same category from two images may be different, but the texture of different regions of an object itself is usually consistent. As shown in Figure 5 (b), the result is improved. Our future study will recognize the regions that tend to get inaccurate correspondence automatically and then mask them during cross-attention.

C.6. User-guided colorization

As colorization for foreground objects relies on the correspondence from VGG features, our methods will not transfer colors for semantically unrelated regions (e.g., clothes to cars). Fortunately, we offer a user-guided version of our method for such cases. As illustrated in Figure 7, one can assign correspondence manually by clicking the matched regions to force color transferring at will. Figure 7 shows the user-guided colorization result of transferring the clothes' orange in I_R to the car in I_S while keeping the colors of the clothes and floor in I_S unchanged. In detail, we provide two types of user guidance: 1) color transferring between regions of source (I_S) and reference (I_R), and 2) keeping the color of the source's regions unchanged. The former is recorded by a sparse binary cross-attention matrix A_C, while the latter is recorded by a binary diagonal matrix A_U. Specifically, we set $A_C(i,j) = 1$ when a user specifies a color transferring between I_R's ith position and I_S's jth position. In addition, $A_U(i,i) = 1$ if the user chooses to keep I_S's ith position unchanged. Accordingly, we alter the $\widetilde{\mathbf{W}_{0_R}}$ calculation to two steps:

$$\widetilde{\mathbf{W}_{0_R}} = Hard(A_S \odot c)\widehat{\mathbf{W}_{0_R}}, \quad \widehat{\mathbf{W}_{0_R}} = A_S\mathbf{W}_{0_R} + A_U\mathbf{W}_{0_S} \quad (2)$$

where $Hard()$ is the hard activation operation (equation 5 in the main paper), $c = \text{sum}(A_C + A_U, 1)$ is the indicator of assigned patches in I_S, and A_S is the self-attention matrix of I_S calculated by $A_S(i,j) = \langle F_S(i), F_S(j) \rangle$. The first step assigns correspondence according to the user's annotations and obtains a sparse warped feature $\widehat{\mathbf{W}_{0_R}}$. In the second step, following the continuity constraint that adjacent patches of an object should be transformed similarly, we propagate assigned patches' color to semantic-related regions and obtain the final dense $\widetilde{\mathbf{W}_{0_R}}$ by warping $\widehat{\mathbf{W}_{0_R}}$ with a hard-masked self-attention matrix.

Figure 8. Online stylization results.

C.7. Online stylization

Since our proposed $\mathbf{w}_0$ loss could measure the global style difference between two images, it provides us with an alternative solution to transfer global style in an online fashion. In such cases, there is no need to train W-predictor; instead, we iteratively optimize the 1d latent code $\mathbf{w}$ or 2d latent maps $\mathbf{W}$ to minimize the $\mathbf{w}_0$ loss between I_P and I_R. As shown in Figure 8, both $\mathbf{w}$ and $\mathbf{W}$ can help transfer global styles, which verifies the effectiveness of our $\mathbf{w}_0$ loss. However, the result with $\mathbf{W}$ has severe artifacts because the continuity constraint of $\mathbf{W}$ is not guaranteed in online fashion.

Figure 9. Visualization of different level maps of $\mathbf{W}$.

C.8. Colorization interpolation

Following SpaceEdit [6], we also design and implement an interface that let users adjust the strength of colorization by interpolating the latent code:

$$\mathbf{W}' = \mathbf{W}_{0_S} + \alpha(\mathbf{W} - \mathbf{W}_{0_S}) \tag{3}$$

where $\mathbf{W}_{0_S}, \mathbf{W}, \mathbf{W}'$ are the self-reconstruct latent maps of source, predicted latent maps, and interpolated latent maps, respectively. α is used to control the strength. The colorization results for different α's are shown in Fig. 6. As α increases, the strength of colorization also increases.

C.9. Visualization of $\mathbf{W}$ predicted by $P_{\mathbf{W}}$

We visualize different levels $\{W_1, W_2, W_3\}$ of $\mathbf{W}$ via PCA in Figure 9. From W_1 to W_3, the colorization outputs become increasingly fine-grained, which reveals the reason that multi-level $\mathbf{W}$ outperforms single map $\mathbf{W}$.

References

[1] 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. 1

[2] 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. 1

[3] 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. 1

[4] 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. 1

[5] 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. 1

[6] 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. 4

[7] 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. 1