Supplementary for Painterly Image Harmonization via Bi-Transformation with Dynamic Kernels

Zhangliang Sun
szl21@mails.tsinghua.edu.cn
Hui Zhang
huizhang@tsinghua.edu.cn

School of Software
BNRist
Tsinghua University
Beijing, China

In the supplementary, we will introduce the experimental settings in Section 1. Then, we will provide more visual results in Section 4. Next, we will provide more results of ablation studies and hyper-parameter analyses in Section 2. Finally, we will discuss our limitations in Section 3.

1 Experimental Settings

1.1 Dataset

We conduct our experiments on COCO [4] and WikiArt [7]. COCO is a benchmark dataset with instance segmentation for 80 object categories, while WikiArt provides a wide variety of digital art styles for backgrounds. Following PHDNet [1], we randomly select an object from an image within the COCO dataset, ensuring that the object's size relative to the image falls within the range of 0.05 to 0.3. We extract this object from the COCO image using instance annotations, effectively creating a "foreground" object. Next, we randomly choose an image from the WikiArt dataset to serve as the background for our composite image. The objective is to intentionally create a composite image where the foreground and background elements do not visually harmonize, resulting in an inharmonious composition. Finally, 57,025 background images for training and 24,421 for testing are generated in this setting.

1.2 Implementation Details

Our decoder is a pretrained VGG-19 encoder, as clearly declared in Section 3.1. Our dynamic kernel generation module consists of **one** Inverted Bottleneck Module and an EfficientNet-B2 kernel generator. The Inverted Bottleneck Module comprises a convolution (conv) layer with a kernel size of one, followed by a depthwise convolution (conv) layer with a kernel size of three, and then another convolution (conv) layer with a kernel size of one. We employ Swish as the activation layer between every two convolution layers. Both the hidden layer and the output layer have 512 channels. The kernel generator outputs four dynamic kernels with sizes of 16x16, 16x16, 32x32 and 32x32, respectively. In the Bi-Transformation Module, each Transformation Module consists of 4 learnable matrices **P** and 4 learnable matrices

V. The sizes of **P** matrices are (64,16), (128,16), (256,32), and (512,32), respectively. The sizes of **V** matrices are (16,64), (16,128), (32,256), and (32,256), respectively. The structure of the decoder is symmetrical to the VGG-19 encoder.

1.3 Baseline Details

We employ the official implementation of the baselines. We set strength = 0.7 in CDC, and we use the default setting in PHDiffusion [6](with Res).

1.4 User Study Settings

We conduct a user study to compare our methods with previous state-of-the-art (SOTA) painterly image harmonization methods since there are no established evaluation metrics for this task. To do this, we randomly select 150 content images from COCO [4] testset and 150 style images from WikiArt [7] testset to generate 150 composite images for the user study. These were used to generate 150 composite images for the user study. We excluded 4 images where the foreground regions were significantly smaller than the background regions, leaving us with 146 composite images and their corresponding masks.

2 Results of Ablation Studies and Hyper-parameter Analyses

Ablation Studies. More visual results of the ablation studies is provided in Figure 1. The meanings of all the ‘ablation’ options are the same as in Section 4.3.

Hyper-parameter Analyses. We show more visual results of transforming different feature maps within our method in Figure 2. Additionally, we provide more visual examples in Figure 3 when we use kernels of different sizes. “Small kernels” represents the use of an 8x8 kernel for each feature map, “medium kernels” represents the use of a 16x16 kernel for each feature map, and “large kernels” represent the use of a 32x32 kernel for each feature map. Usually, our method with “large kernels” generates better results.

3 Limitations

Although our DKTNet can generally produce visually satisfactory results, there are some instances of failure. In some cases, our DKTNet overly emphasizes the global style of the entire background image, resulting in unusual harmonized images. One potential solution is to employ attention mechanisms or other techniques to generate pixel-wise dynamic kernels. For example, Figure 4 illustrates some failure cases of our method, where the bodies of the horse and the man are segmented into several parts, making them challenging to identify. We will explore this problem in the future.

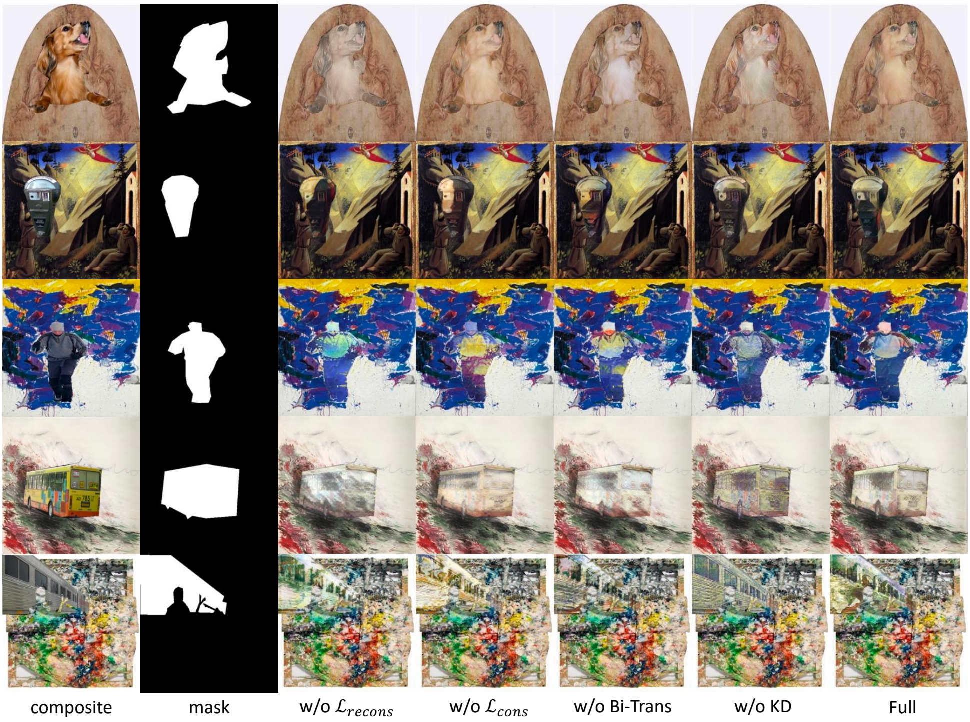

Figure 1: More visual results of ablation studies.

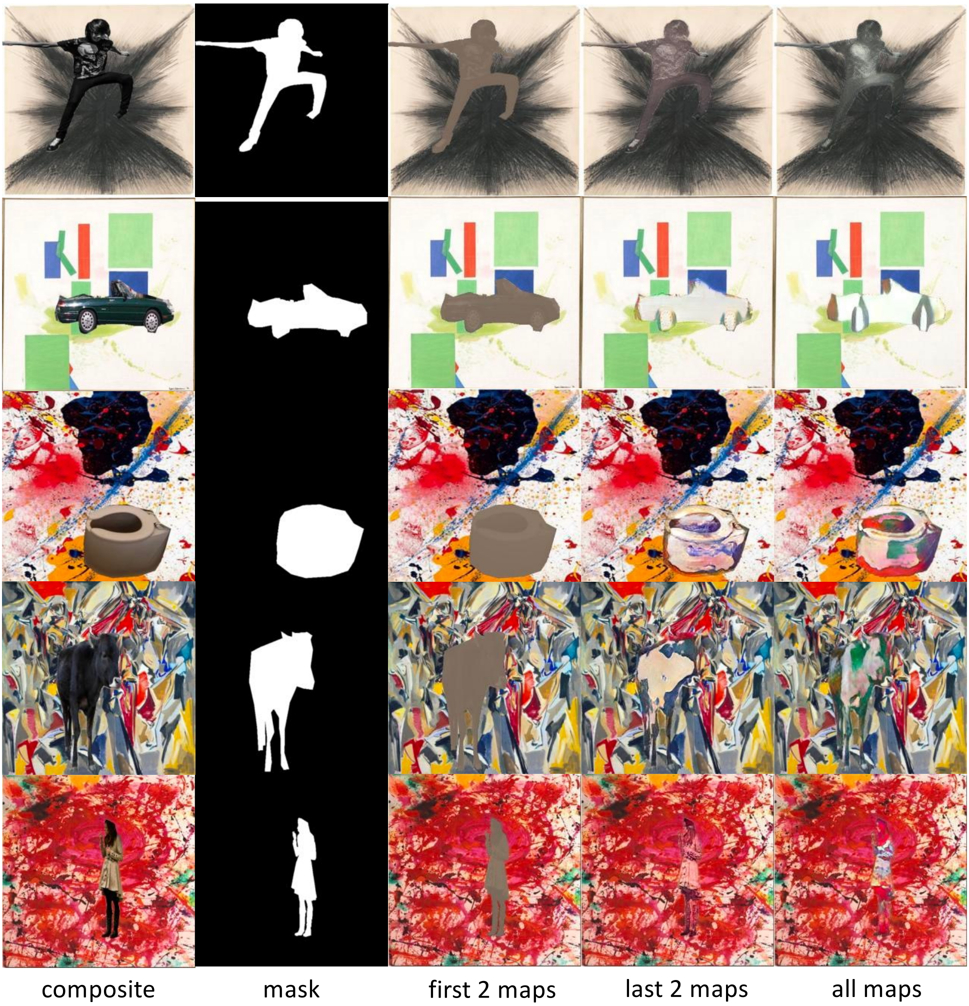

Figure 2: More visual results when we transform different feature maps.

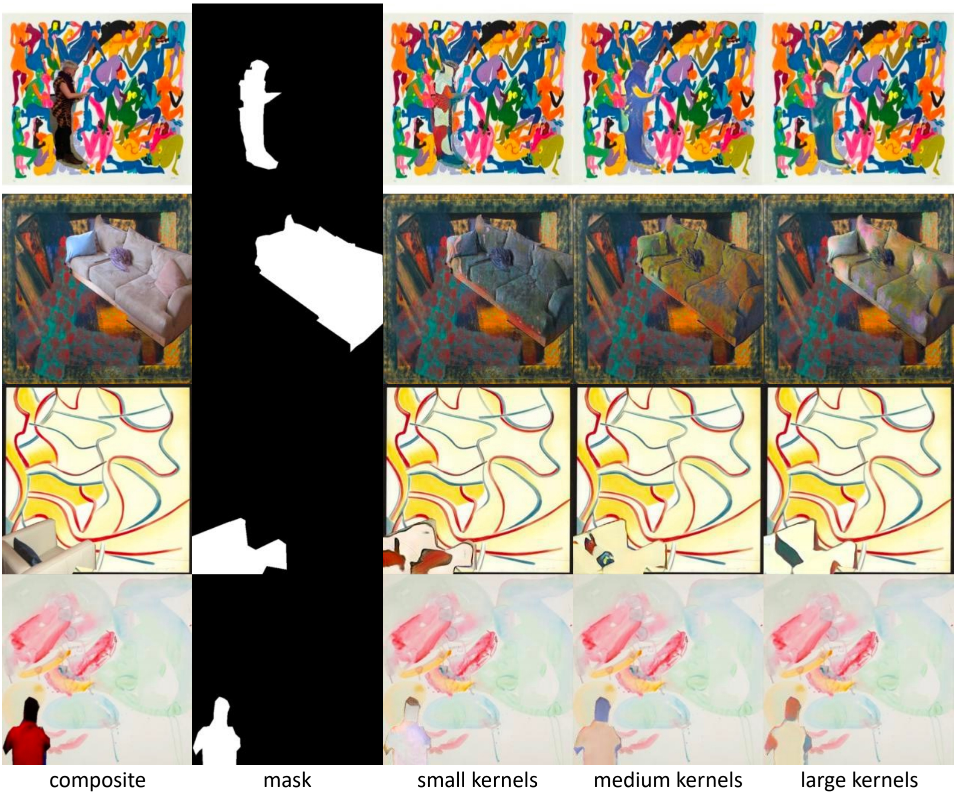

Figure 3: More visual results when we use kernels of different sizes.

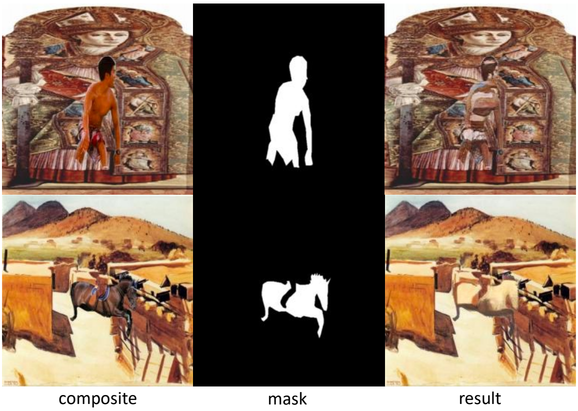

Figure 4: Some failure cases of our method.

4 Visual results

We select strong baselines AdaIN [3], AdaAttN [5], SANet [8], StyTr2 [2], PHDiffusion [6] and PHDNet [1] to make comparisons with our DKTNet. We provide more visual examples comparing our method with the baselines in Figure 5. Our DKTNet can strike a great balance between the style and content.

References

[1] Junyan Cao, Yan Hong, and Li Niu. Painterly image harmonization in dual domains. In *AAAI*, volume 37, pages 268–276, 2023.

[2] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *CVPR*, pages 11326–11336, 2022.

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[5] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *ICCV*, pages 6649–6658, 2021.

[6] Lingxiao Lu, Jiangtong Li, Junyan Cao, Li Niu, and Liqing Zhang. Painterly image harmonization using diffusion model. In *MM*, pages 233–241, 2023.

[7] Kiri Nichol. Painter by numbers, 2016. URL https://kaggle.com/competitions/painter-by-numbers/overview.

[8] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, pages 5880–5888, 2019.

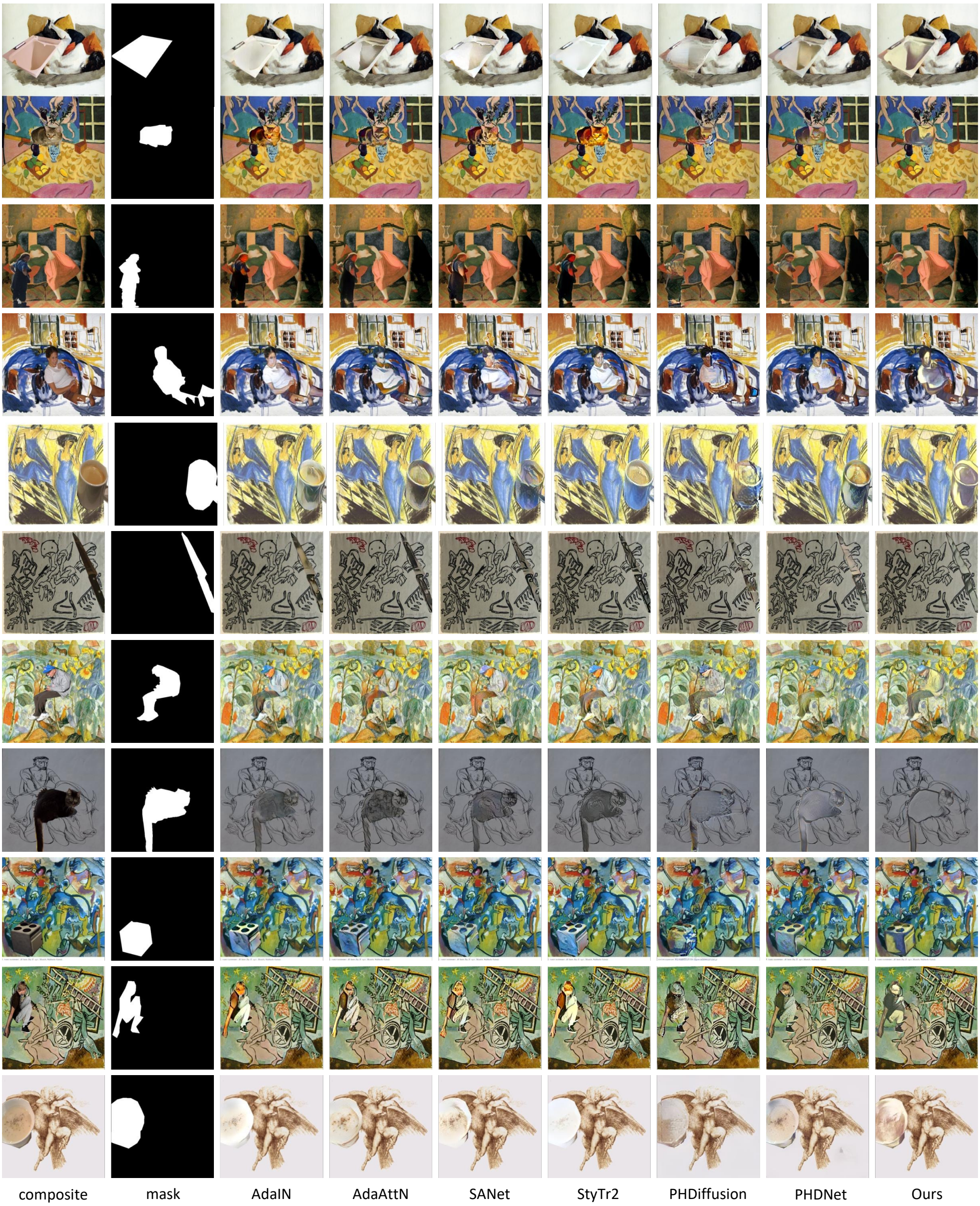

Figure 5: Additional visual comparisons with baselines.