

# Painterly Image Harmonization by Learning from Painterly Objects

Li Niu\*, Junyan Cao, Yan Hong, Liqing Zhang

MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University  
{ustcnewly, joy.c1, hy2628982280, lqzhang}@sjtu.edu.cn

## Abstract

Given a composite image with photographic object and painterly background, painterly image harmonization targets at stylizing the composite object to be compatible with the background. Despite the competitive performance of existing painterly harmonization works, they did not fully leverage the painterly objects in artistic paintings. In this work, we explore learning from painterly objects for painterly image harmonization. In particular, we learn a mapping from background style and object information to object style based on painterly objects in artistic paintings. With the learnt mapping, we can hallucinate the target style of composite object, which is used to harmonize encoder feature maps to produce the harmonized image. Extensive experiments on the benchmark dataset demonstrate the effectiveness of our proposed method. Dataset and code are available at <https://github.com/bcmi/ArtoPIH-Painterly-Image-Harmonization>.

## 1 Introduction

Compositing multiple regions from different images is a fundamental photo editing technique. However, when pasting the foreground object extracted from one image on another background image, there may exist style inconsistency between foreground and background. To address such style inconsistency, myriads of image harmonization methods (???) have been developed to match the illumination information between foreground and background. Nevertheless, they are less effective when the foreground object is extracted from a photographic image and the background is an artistic painting, because complex styles (color, texture, pattern, strokes) need to be matched between foreground and background. Image harmonization with photographic object and painterly background is called painterly image harmonization (?), which has only received limited attention.

The existing painterly image harmonization methods can be roughly divided into optimization-based methods (??) and feed-forward methods (??). Optimization-based methods update the composite object iteratively to minimize the designed losses, which is too slow for real-time application. In contrast, feed-forward methods pass the composite image

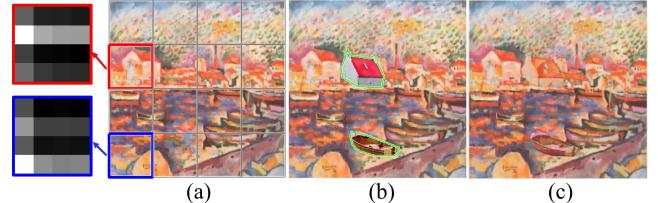


Figure 1: (a) The background image with two marked query patches (red and blue). We show the similarity maps of two query patches based on VGG-19 (?) features. (b) The composite image with two inserted foreground objects. (c) The ideal harmonized image.

through the network only once to produce a harmonized image, which is much more efficient than optimization-based methods. To name a few, (?) introduced AdaIN (?) to adjust the style of composite object. (?) explored harmonizing the composite image in both spatial domain and frequency domain. Some diffusion model based methods (??) for cross-domain image composition can also be applied to this task. However, the harmonized objects may still look unnatural on the painterly background.

One factor that hinders painterly image harmonization is the absence of ground-truth harmonized objects, but we can learn what an object in an artistic painting should be like based on the existent painterly objects in the artistic paintings. Therefore, this work explores learning from painterly objects, which is substantially overlooked by previous works (????). The key insight of our work is hallucinating the target style of composite object. Previous works (??) migrate the global style of painterly background to the composite object using AdaIN (?), during which the style is represented by the feature statistics (mean, variance) in pretrained VGG-19 (?) encoder. Nevertheless, the styles of different regions/objects in an artistic painting could vary considerably. As shown in Figure 1(a), we divide the whole image into 16 patches and extract the style vectors (mean/variance of VGG-19 encoder features) of all patches. We calculate their pairwise similarity based on style vectors. Given each query patch, we display a similarity map containing the similarities between this query patch and all patches. The patches with high similarity values have similar visual styles with

\*Corresponding author.

Copyright © 2024, Association for the Advancement of Artificial Intelligence ([www.aaai.org](http://www.aaai.org)). All rights reserved.

the query patch. According to the similarity map, we can see that the style vector has prominent locality property, that is, the style vectors of various regions in the same image could be dramatically distinct. Therefore, when compositing different objects (*e.g.*, house, boat) at different locations on the background, they should be harmonized to different target styles (see Figure 1(b) and (c)).

*To achieve the goal of target style hallucination, we learn a mapping module based on painterly objects, which maps background style and object feature to object style.* The object features are supposed to contain the useful conditional information (*e.g.*, color, semantics, local context) for adapting background style to object style. With the learnt mapping module, we attempt to hallucinate the target style of composite object based on its object feature and the background style. Intuitively, the mapping module answers the following question: “*given a painterly background, if its painter draws a specific object at this location, what should it look like?*”

However, one critical issue is the domain gap of object features between photographic objects and painterly objects, so that the mapping module trained on painterly objects cannot work well on composite objects. We have tried unsupervised manner like adversarial learning (?) to bridge the domain gap, but the performance is far from satisfactory. Therefore, we choose to annotate pairs of similar photographic objects and painterly objects. Given an artistic painting with a painterly object, we retrieve and annotate photographic objects which are similar to this painterly object, leading to abundant triplets (artistic painting, painterly object, photographic object). In each triplet, we refer to the painterly object (*resp.*, artistic painting) as the reference object (*resp.*, reference image) of the photographic object. We put the photographic object within the bounding box of reference object in the reference image, resulting in a composite image (see Figure 3). In this way, we can acquire abundant pairs of composite images and reference images, in which the composite object feature should be close to reference object feature and the hallucinated style of composite object should be close to reference object style. With the above assumptions, we design a novel network with mapping module, and train it using pairs of composite images and reference images.

Our major contributions can be summarized as follows. 1) We explore learning from painterly objects in the painterly image harmonization task, which has never been studied before. 2) We design a novel network with mapping module, which can hallucinate the target style of composite object based on the background style and object feature. 3) We will release our annotated reference images/objects, which would greatly benefit the future research of painterly image harmonization. 4) Extensive experiments on COCO (?) and WikiArt (?) demonstrate the effectiveness of our proposed method.

## 2 Related Work

### 2.1 Image Harmonization

Image harmonization aims to harmonize a composite image by adjusting foreground illumination to match background

illumination. In recent years, abundant deep image harmonization methods (???????) have been developed. For example, (???) proposed diverse attention modules to treat the foreground and background separately, or establish the relation between foreground and background. (????) directed image harmonization to domain translation or style transfer by treating different illumination conditions as different domains or styles. (???) decomposed an image into reflectance map and illumination map. More recently, (???????) used deep network to predict color transformation, striking a good balance between efficiency and effectiveness.

However, most image harmonization methods only adjust illumination and require ground-truth supervision, which is unsuitable for painterly image harmonization.

### 2.2 Painterly Image Harmonization

Different from Section 2.1, in painterly image harmonization, the foreground is a photographic object while the background is artistic painting. The goal of painterly image harmonization is adjusting the foreground style to match background style and preserving the foreground content. The existing methods (???) can be roughly categorized into optimization-based methods (??) and feed-forward methods (?). The optimization-based methods (??) iteratively optimize over the composite foreground to minimize the designed loss functions. The feed-forward methods (???) send the composite image through the harmonization network once and generate the harmonized image. Some diffusion model based methods (??) for cross-domain image composition can also harmonize the composite image.

Our proposed method belongs to feed-forward methods. Different from previous works (??), we explore learning from painterly objects to hallucinate the target styles of composite objects.

### 2.3 Artistic Style Transfer

Artistic style transfer (?????????????) targets at recomposing a content image in the style of a style image. Amounts of works (??) focus on global style transfer. Recently, some works (???) turn to fine-grained local style transfer by establishing local correspondences between content image and style image. For example, non-local attention mechanism was adopted in SANet (?) to attentively transfer style features to content features. AdaAttN (?) proposed novel attentive normalization by combining AdaIN (?) and SANet (?). StyTr<sup>2</sup> (?) applied transformer architecture to capture patch-wise features to solve content leak issue.

The motivation of local style transfer (????) is seeking for relevant regions in the style image for a given content image and transferring the corresponding local style. *Nevertheless, the existence of relevant regions is questionable, and the methods may not find the relevant regions accurately (see Section 4.2).*

## 3 Our Method

We suppose that a composite image  $I^c$  is obtained by pasting a photographic object on the painterly background. Painterly image harmonization aims to transfer the style

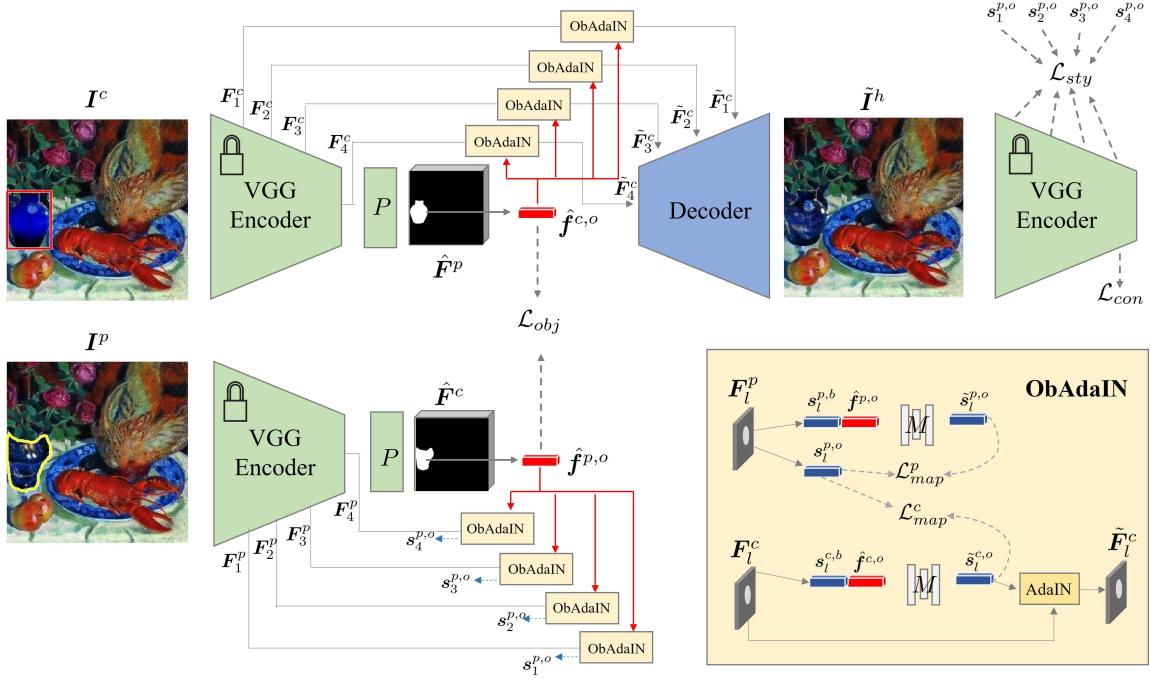


Figure 2: The illustration of our network structure. Given a pair of composite image  $I^c$  and painterly image  $I^p$ , we pass them through pretrained VGG-19 (?) encoder and projection module  $P$  to get object features  $\hat{f}^{c,o}$  and  $\hat{f}^{p,o}$  respectively. We insert our designed ObAdaIN modules into skip connections and bottleneck, which harmonize the encoder feature maps of  $I^c$ . The harmonized encoder feature maps are delivered to the decoder to produce the harmonized image  $\tilde{I}^h$ . In the ObAdaIN module in the  $l$ -th layer, the mapping module  $M_l$  learns a mapping from background style  $s_l^{p,b}$  (*resp.*,  $s_l^{c,b}$ ) and object feature  $\hat{f}^{p,o}$  (*resp.*,  $\hat{f}^{c,o}$ ) to object style  $\tilde{s}_l^{p,o}$  (*resp.*,  $\tilde{s}_l^{c,o}$ ) for painterly (*resp.*, composite) object.

from painterly background to composite object while preserving the content of composite object, resulting in a harmonized image  $\tilde{I}^h$ . To learn from painterly objects in artistic paintings, we first build a training set with pairs of composite images  $I^c$  and reference images  $I^p$ . Then, we design a network which could be trained on paired training images. Our network is based on U-Net (?) structure with skip connections, with our designed ObAdaIN module inserted into the bottleneck and skip connections. The ObAdaIN module aims to hallucinate the target style of composite object. We name our constructed training data as Arto (**A**rtistic **O**bject) dataset and our method as ArtoPIH (**P**ainterly **I**mage **H**armonization). Next, we will introduce our Arto dataset in Section 3.1 and our ArtoPIH in Section 3.2.

### 3.1 Training Data Preparation

Based on 57,025 artistic paintings in the training set of WikiArt (?), we use off-the-shelf object detection model (?) pretrained on COCO (?) dataset to detect objects in artistic paintings. Despite the domain gap between artistic paintings and photographic images, we detect 34,570 painterly objects. For each painterly object, we first retrieve 100 photographic objects with similar appearance and semantics automatically (see details in the Supplementary).

However, the retrieved photographic objects are very

noisy, so we ask annotators to manually remove those dissimilar photographic objects. After filtering, we have 33,294 painterly objects associated with similar photographic objects. Each painterly object has an average of 9.83 similar photographic objects, leading to a total of 327,181 triplets (artistic painting  $I^p$ , painterly object, photographic object). In each triplet, we refer to the painterly object (*resp.*, artistic painting) as the reference object (*resp.*, reference image) of the photographic object. As illustrated in Figure 3, we put the photographic object within the bounding box of reference object in the reference image, producing a composite image  $I^c$  with foreground mask  $M^c$ . Therefore, we have in total 327,181 training image pairs  $\{I^c, I^p\}$ .

### 3.2 Our Network Structure

We employ the encoder and decoder structures in (?) as our backbone, in which the encoder is pretrained VGG-19 (?) and the decoder is symmetric structure of encoder. We only use the layers up to *ReLU-4\_1* of VGG-19 as our encoder, which are fixed to extract multi-scale encoder features. The encoder is divided into four encoder layers, which are up to *ReLU-1\_1*, *ReLU-2\_1*, *ReLU-3\_1*, and *ReLU-4\_1* respectively. We add skip connections for the first three encoder layers.

We feed composite image  $I^c$  into the encoder to extract the feature map  $F_l^c$  of  $l$ -th layer for  $l \in \{1, 2, 3, 4\}$ . Sim-

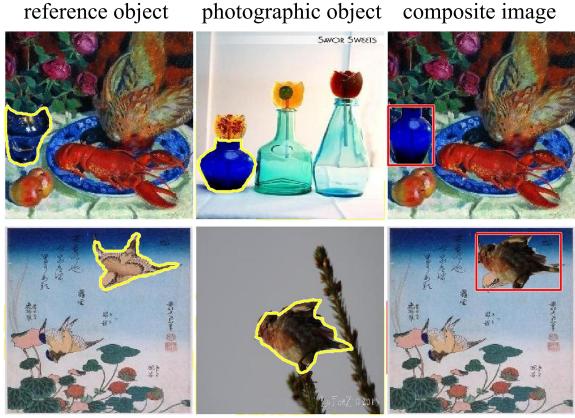


Figure 3: The illustration of constructing training image pairs  $\{I^p, I^c\}$ . Given a photographic object (outlined in yellow), we have a reference object (outlined in yellow) with similar color and semantics in the reference image  $I^p$ . We put the photographic object within the bounding box of reference object, resulting in a composite image  $I^c$ .

ilarly, we also feed its reference image  $I^p$  with reference object into the encoder to extract the feature map  $F_l^p$  of  $l$ -th layer for  $l \in \{1, 2, 3, 4\}$ . Previous works on painterly image harmonization (??) usually apply vanilla Adaptive Instance Normalization (AdaIN) (?) to the encoder feature maps, by aligning the feature statistics (mean, standard deviation) of foreground feature map with those of background feature map. As the feature statistics represent the style information, the composite object would be rendered with the background style.

However, the background style may not be the ideal target style for the composite object, considering the diversified local styles within the same image and the unique property (e.g., color, semantics, local context) of composite object. In this work, we attempt to hallucinate the ideal target style of composite object by considering background style and object information. To achieve this goal, we design a novel Object-aware AdaIN (ObAdaIN) module to replace the vanilla AdaIN operation.

**Domain-Invariant Object Feature:** Our ObAdaIN module requires the object feature of foreground object, which contains the necessary conditional information for adapting background style to object style.

Given a training image pair  $\{I^c, I^p\}$ , the composite object in  $I^c$  is photographic object but the reference object in  $I^p$  is painterly object, so their object features  $\hat{f}^{c,o}$  and  $\hat{f}^{p,o}$  belong to two different domains. As shown in Figure 2, we first project feature maps  $F_4^p$  and  $F_4^c$  to a common domain through a projection module  $P$ , yielding  $\hat{F}^p$  and  $\hat{F}^c$  respectively. Then, we perform average pooling within the foreground region on  $\hat{F}^p$  (resp.,  $\hat{F}^c$ ) to get the object feature  $\hat{f}^{p,o}$  (resp.,  $\hat{f}^{c,o}$ ).

Since the composite object and its reference object have similar appearance and semantics, we push close  $\hat{f}^{p,o}$  and

$\hat{f}^{c,o}$  using the following loss:

$$\mathcal{L}_{obj} = \|\hat{f}^{p,o} - \hat{f}^{c,o}\|^2. \quad (1)$$

By pulling close the object features of similar objects from two domains, we enforce the extracted object features to be domain-invariant.

**ObAdaIN Module:** With extracted domain-invariant object features, our ObAdaIN module learns a mapping from background style and object feature to object style. The architecture of ObAdaIN is shown in Figure 2.

By taking the reference image  $I^p$  and the  $l$ -th encoder layer as an example, we have the feature map  $F_l^p$ . We extract the background style vector  $s_l^{p,b} = [\mu_l^{p,b}, \sigma_l^{p,b}]$ , in which  $\mu_l^{p,b}$  (resp.,  $\sigma_l^{p,b}$ ) is channel-wise mean (resp., standard deviation) of background feature map.

We concatenate the background style vector  $s_l^{p,b}$  with object feature  $\hat{f}^{p,o}$ , which are sent to a mapping module  $M_l$  to generate the object style vector  $\tilde{s}_l^{p,o}$ . The above process can be represented by  $\tilde{s}_l^{p,o} = M_l(s_l^{p,b}, \hat{f}^{p,o})$ . For the reference object in  $I^p$ , we can obtain its ground-truth style vector  $s_l^{p,o}$  by calculating the channel-wise mean and standard deviation of foreground feature map. We use  $s_l^{p,o}$  to supervise  $\tilde{s}_l^{p,o}$  as follows,

$$\mathcal{L}_{map}^p = \|\tilde{s}_l^{p,o} - s_l^{p,o}\|^2. \quad (2)$$

Compared with  $I^p$ , its paired composite image  $I^c$  has similar foreground object, the same background, and the same foreground placement (bounding box). Therefore, we assume that the ideal target style of composite object should be close to reference object style  $s_l^{p,o}$ . Specifically, given  $I^c$ , we extract its background style vector  $s_l^{c,b}$  and object feature  $\hat{f}^{c,o}$  in a similar way to  $I^p$ . Then, we can have  $\tilde{s}_l^{c,o} = M_l(s_l^{c,b}, \hat{f}^{c,o})$ , which is supervised by  $s_l^{p,o}$ :

$$\mathcal{L}_{map}^c = \|\tilde{s}_l^{c,o} - s_l^{p,o}\|^2. \quad (3)$$

Next, we use the hallucinated style vector  $\tilde{s}_l^{c,o} = [\tilde{\mu}_l^{c,o}, \tilde{\sigma}_l^{c,o}]$  to harmonize the composite image  $I^c$ . By denoting the foreground (resp., background) feature map in  $F_l^c$  as  $F_l^{c,o}$  (resp.,  $F_l^{c,b}$ ), we apply AdaIN to adjust  $F_l^{c,o}$  as follows,

$$\tilde{F}_l^{c,o} = \tilde{\sigma}_l^{c,o} \frac{F_l^{c,o} - \mu(F_l^{c,o})}{\sigma(F_l^{c,o})} + \tilde{\mu}_l^{c,o}, \quad (4)$$

in which  $\mu(\cdot)$  (resp.,  $\sigma(\cdot)$ ) means calculating the mean (resp., standard deviation) of the specified feature map. The adjusted foreground feature map  $\tilde{F}_l^{c,o}$  is combined with the unchanged background feature map  $F_l^{c,b}$  to form the harmonized feature map  $\tilde{F}_l^c$ .

The harmonized feature maps of all encoder layers are sent to the decoder to produce the harmonized image  $I^h$ . Similar to (?), we also adopt blending layer to combine the output image  $I^h$  and the composite image  $I^c$  with predicted soft blending mask, giving rise to the final harmonized image  $\tilde{I}^h$ .

For the harmonized image  $\tilde{I}^h$ , we expect its object style to match the reference object style  $s_l^{p,o}$ , and thus design the style loss accordingly:

$$\begin{aligned}\mathcal{L}_{sty} = & \sum_{l=1}^4 \|\mu\left(\phi_l(\tilde{I}^h) \circ M^c\right) - \mu_l^{p,o}\|^2 \\ & + \sum_{l=1}^4 \|\sigma\left(\phi_l(\tilde{I}^h) \circ M^c\right) - \sigma_l^{p,o}\|^2,\end{aligned}\quad (5)$$

where each  $\phi_l(\cdot)$  denotes the  $l$ -th encoder layer in VGG-19 (?) encoder,  $\circ$  means element-wise product.

We use content loss (?) to ensure that the foreground content is maintained:

$$\mathcal{L}_{con} = \left\| \phi_4(\tilde{I}^h) - \phi_4(I^c) \right\|^2, \quad (6)$$

in which  $\phi_4(\cdot)$  has been defined in Eqn. 5.

The total loss can be written as

$$\mathcal{L}_{total} = \mathcal{L}_{obj} + \lambda(\mathcal{L}_{map}^p + \mathcal{L}_{map}^c) + \mathcal{L}_{sty} + \mathcal{L}_{con}, \quad (7)$$

in which the hyper-parameter  $\lambda$  is empirically set as 10.

## 4 Experiments

### 4.1 Datasets and Implementation Details

We conduct experiments on COCO (?) and WikiArt (?). The details are left to Supplementary. Our network is implemented using Pytorch 1.10.0 and trained using Adam optimizer with learning rate of  $1e-4$  on ubuntu 20.04 LTS operation system, with 128GB memory, Intel(R) Xeon(R) Silver 4116 CPU, and one GeForce RTX 3090 GPU. For the encoder and decoder structure, we follow (?). For  $P$  module, we use one residual block (?). For  $M_l$  module in the  $l$ -th encoder layer, we stack three ResMLP Layers (?), in which the intermediate dimension is equal to the dimension of style vector in the  $l$ -th layer.

### 4.2 Comparison with Baselines

We compare with two groups of baselines: artistic style transfer methods and painterly image harmonization methods. Since the standard image harmonization methods introduced in Section 2.1 only adjust the illumination statistics and require ground-truth images as supervision, they are not suitable for painterly image harmonization.

For the first group of baselines, we first use artistic style transfer methods to stylize the entire photographic image according to the painterly background, and then paste the stylized photographic object on the painterly background. We compare with typical or recent style transfer methods: SANet (?), AdaAttN (?), StyTr2 (?), QuantArt (?), INST (?). For the second group of baselines, we compare with SDEdit (?), CDC (?), E2STN (?), DPH (?), PHDNet (?).

**Visualization Results:** The comparison with the first group of baselines is shown in the upper part in Figure 4. We can see that the harmonized objects generated by our method look more natural and compatible with the background. In

Method	B-T score	Time(s)	FLOPs(G)
SANet	-0.365	0.0097	43.32
AdaAttN	-0.535	0.0115	49.64
StyTr2	0.149	0.0504	39.74
QuantArt	0.336	0.1031	133.34
Inst	-0.762	2.2996	3378.43
SDEdit	-0.654	2.1321	3164.52
CDC	0.204	2.3427	3299.81
E2STN	-0.152	0.0079	29.28
DPH	0.340	55.24	-
PHDNet	0.485	0.0321	158.41
ArtoPIH	0.953	0.0258	40.03

Table 1: The comparison between different methods.

row 1, one interesting observation is that the vases harmonized by different methods have different colors. One possible reason is that SANet, AdaAttN, and StyTr2 seek for the relevance between composite object and different regions in the painterly background, and transfer the local style of relevant region to the composite object. However, there may not exist relevant regions in the painterly background. Even if relevant regions exist, they may not accurately attend to the relevant regions. In contrast, we hallucinate the target style based on background style and object information. Our harmonized vase maintains the object color and also has background texture, while the baseline results are over-smooth without compatible textures or losing some structure details (*e.g.*, vase neck). In row 2, the harmonized faces from baseline methods all have corrupted structures, while our method preserves the facial structure well. This might be because that our model learns from abundant painterly person in artistic paintings and the hallucinated target style is able to preserve the facial structure.

The comparison with the second group of baselines is shown in the lower part in Figure 4. In row 1, our harmonized dog has more harmonious color and compatible texture with the background, while the baseline results are corrupted or overdark. In row 2, our harmonized fruits have more suitable color and textures, and thus look more visually pleasant than the baseline results. More visualization results can be found in Supplementary.

**User Study:** Following (?), we also conduct user study to compare different methods. We randomly select 100 content images from COCO and 100 style images from WikiArt to generate 100 composite images. We compare the harmonized results generated by 10 baselines and our ArtoPIH.

Specifically, for each composite image, we obtain 11 harmonized outputs and use every two images from these 11 images to construct image pairs. With 100 composite images, we construct 5500 image pairs. Then, we invite 100 participants to watch one image pair each time and choose the better one. Finally, we collect 550,000 pairwise results and calculate the overall ranking of all methods using Bradley-Terry (B-T) model (?). The B-T score results are summarized in Table 1, in which our method achieves the highest B-T score.

**Efficiency Analyses:** For efficiency comparison, we re-



Figure 4: In the upper part, we compare with style transfer baselines SANet (?), AdaAttn (?), StyTr2 (?), QuantArt (?), INST (?). In the lower part, we compare with painterly image harmonization baselines SDEdit (?), CDC (?), E2STN (?), DPH (?), PHDNet (?).

port the inference time and FLOPs. We evaluate with the image size  $256 \times 256$  and the inference time is averaged over 100 test images on a single RTX 3090 GPU. The optimization-based method DPH is time-consuming, due to iteratively updating the input composite image. Diffusion-based methods are much slower than the other feed-forward methods, which limits their real-world applicability. Our method is relatively efficient compared with the competitive baselines (??).

### 4.3 Hallucinated Object Style

To intuitively demonstrate the effectiveness of our hallucinated object styles, we sample some photographic objects in the test set and find their reference painterly objects in the test set, similar to the process of constructing training image pairs (see Section 3.1). Specifically, given a photographic object and a painterly object in a painterly image, we can obtain a composite image as illustrated in Figure 3, followed by calculating the  $L_2$  distance between the composite object feature  $\hat{f}^{c,o}$  and painterly object feature  $\hat{f}^{p,o}$ . For each photographic object, we retrieve its nearest painterly object as reference object and the corresponding painterly image as reference image. In Figure 5, we show several examples of photographic objects and the retrieved reference objects. It can be seen that the photographic objects have similar color and semantics with their reference objects, which verifies the

effectiveness of our learnt domain-invariant object features.

Based on the composite image composed by the photographic object and its reference image, we further adjust the shape of photographic object to fully cover the reference object in the reference image, to exclude the influence of reference object. Then, based on our trained model, we apply three types of style vectors to the composite object to get the harmonized images. The first one is the background style vector  $s_l^{c,b}$ . The second one is the style vector  $s_l^{p,o}$  of reference object. The third one is our hallucinated style vector  $\hat{s}_l^{c,o}$ . We apply the above three style vectors to the foreground feature map using AdaIN operation. The obtained harmonized images are denoted as “BG”, “RO”, and “Ours” respectively, as shown in Figure 5. The harmonized results “BG” and “RO” differ a lot, which demonstrates the huge gap between background style and object style. Directly applying background style may destroy the original color of composite object or bring in noticeable artifacts.

Since the composite object and reference object have similar object information and background style, the target style  $\hat{s}_l^{c,o}$  of composite object is expected to approach the reference object style  $s_l^{p,o}$ . Based on Figure 5, the harmonized results “Ours” are close to “RO”, which proves that our model can hallucinate ideal target styles.

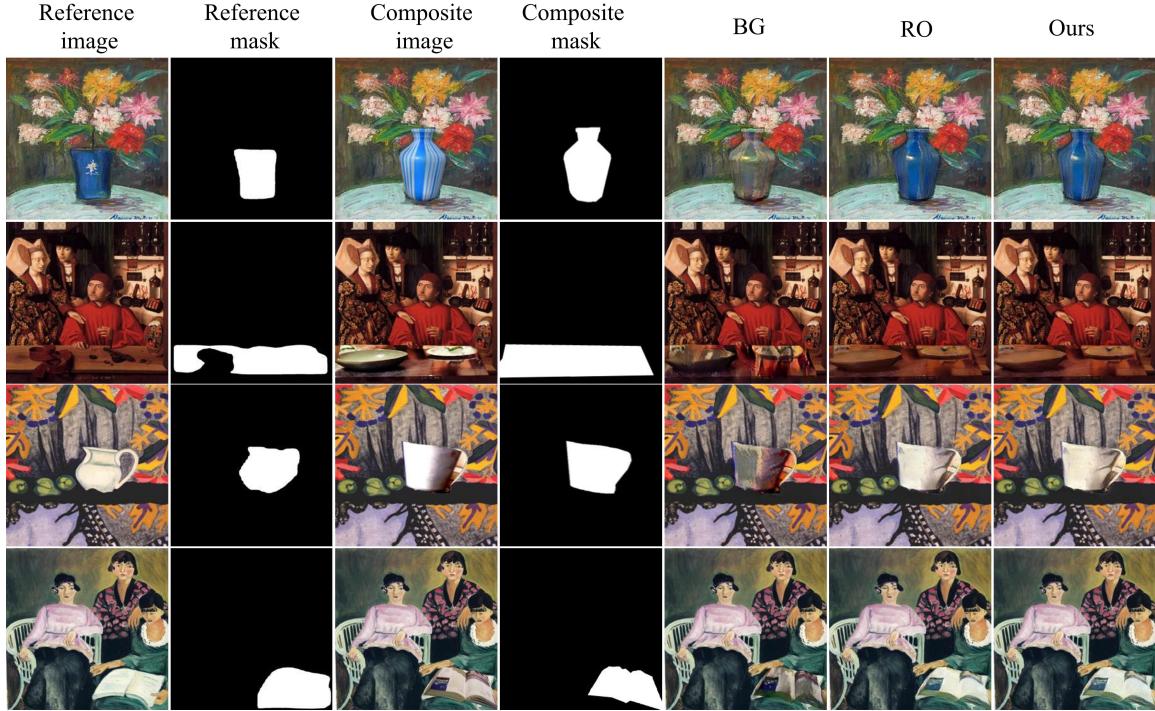


Figure 5: From left to right, we show the reference image, the mask of reference object, the composite image, the mask of composite object, and the harmonized results obtained using different style vectors. “BG” uses background style vector, “RO” uses reference object style vector, and “Ours” uses our hallucinated style vector.

#### 4.4 More Results in Supplementary

In the supplementary, we will provide ablation study results, more visual comparison with baselines, and discussion on failure cases. We will also show the results beyond COCO (?) dataset to demonstrate the generalization ability of our method across different datasets and different object categories.

### 5 Conclusion

In this work, we have explored learning from painterly objects for painterly image harmonization. Based on the annotated pairs of composite images and reference painterly images, we have succeeded in hallucinating the target style of composite object, leading to visually pleasing harmonization results. Extensive experiments on the benchmark dataset have proved the advantage of our proposed ArtoPIH.

### Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grant No. 62076162), the Shanghai Municipal Science and Technology Major/Key Project, China (Grant No. 2021SHZDZX0102, Grant No. 20511100300).

Optio laborum libero odit adipisci sequi ab temporibus recusandae dolores similiqe maiores, culpa deleniti consectetur a?Voluptatum iure labore, excepturi nemo incident doloremque consectetur.Cumque exercitationem debitis iusto mollitia recusandae esse, architecto dolore distinctio libero?Inventore distinctio placeat amet, voluptate nos-

trum facilis, eos libero eius iste, laborum atque amet fuit expedita animi sed qui accusamus eos nobis.Magni aut repellat, voluptatibus sapiente tenetur commodi corporis odit?Itaque ipsa doloremque voluptas blanditiis at nobis optio, est voluptatum accusantium mollitia illo, fugit et totam omnis iure ullam optio corrupti inventore quod quas, placeat quaerat officia laboriosam debitibus?Laboriosam est nobis sed sint nostrum ullam at iste officiis, nemo neque exercitationem culpa quia, minima magni dolorem magnam quaerat delectus ipsa officia quas alias labore tempora, sequi possimus libero blanditiis dolores corrupti, totam quaerat illo hic maxime repudianda quas.Eveniet ipsum ea nobis numquam, quo fugit sequi harum.Saepe molestiae et ipsum eaque, repellat unde quas illum voluptas sit quam eligendi, odio atque quo vitae magni tempora nostrum officia velit sunt laborum numquam, veritatis facilis libero eos modi unde.Quia doloremque doloribus officiis quibusdam quas at, odit pariatur asperiores libero nesciunt consequatur quisquam esse error quae cumque ipsam, similique voluptate nobis quaerat dolore, a doloribus voluptatum reiciendis totam velit enim aperiam cum repudianda sint, ab a esse impedit veritatis rerum illo praesentium voluptas.Temporibus error eveniet incident consectetur eius, architecto consequuntur ea quidem ex adipisci dolores voluptas, laboriosam excepturi quaerat deserunt rerum.