Locally Stylized Neural Radiance Fields
– Supplementary Material –

Hong-Wing Pang[1], Binh-Son Hua[2,3], and Sai-Kit Yeung[1]

[1]Hong Kong University of Science and Technology [2]Trinity College Dublin [3]VinAI Research, Vietnam

Abstract

In this supplemental document, we provide additional details and experiment results for our proposed method. In particular, we provide an in-depth description of the segmentation procedure used in our stylization procedure; a user study; and more qualitative results on the LLFF and the Replica dataset.

1. Segmentation procedure

In this section, we describe in greater detail the segmentation procedures used for creating scene regions and style regions. As mentioned in the main paper, our stylization method can be performed irrespective of the segmentation method used.

1.1. Scene regions

Given a set of training images $\{\hat{\mathbf{y}}\}$, we want to generate a set of segmentation maps $\{\hat{\mathbf{k}}\}$, where every pixel is classified into a finite set of C scene regions. The segmentation process should be unsupervised (i.e. C is not set to a fixed value, and is determined during segmentation), and can be applied to any arbitrary set of scenes.

To satisfy these requirements, we base our implementation on the segmentation procedure proposed by Kim et. al. [2]. In their method, a single image is passed into a CNN model producing a *response map* $\mathbf{r} \in \mathbb{R}^{H\times W\times Q}$, where Q is the upper bound of no. of scene regions. A segmentation map $\mathbf{c} \in \mathbb{R}^{H\times W}$ can be obtained from $\mathbf{r}$ by taking the argmax function.

This network is trained with a combination of the *similarity loss* $\mathcal{L}_{sim}$ and *continuity loss* $\mathcal{L}_{con}$. The similarity loss is defined as the sum of cross-entropies between response vector r_n and the target vector c_n:

$$\mathcal{L}_{sim}(\mathbf{r}) = -\sum_{n}\sum_{i=1}^{Q} c_{n,i}\log r_{n,i}, \tag{1}$$

where c_n is the one-hot vector in $\mathbf{c}$ corresponding to r_n. The continuity loss is defined as the sum of L1 distances between horizontally and vertically adjacent features in $\mathbf{r}$:

$$\mathcal{L}_{con}(\mathbf{r}) = \sum_{i=1}^{W-1}\sum_{j=1}^{H-1} \|r_{i+1,j} - r_{i,j}\|_1 + \|r_{i,j+1} - r_{i,j}\|_1. \tag{2}$$

It can be seen that $\mathcal{L}_{sim}$ encourages similar features in $\mathbf{r}$ to be grouped together and form a single cluster; $\mathcal{L}_{con}$ ensures spatial continuity of clusters and prevents the segmentation from being too fragmented. In general, the unique number of classes in $\mathbf{c}$ is initially high (i.e. close to Q), and gradually decreases over time as more and more feature vectors in $\mathbf{r}$ are clustered into the same region.

To extend this method to segmentation of multiple images simultaneously, we train the segmentation network by sampling a batch of B images during each iteration, instead of a single image. The loss values for each individual response maps $\{\mathbf{r}_1, \cdots, \mathbf{r}_B\}$ are computed and summed together. After the network is trained, we can run segmentation over all of $\{\hat{\mathbf{y}}\}$, obtain the set of C remaining active classes, and re-index the segmentation maps from 1 to C.

1.2. Style regions

Given a style image $\mathbf{s}$, we want to segment it into a set of S style regions $\{\mathbf{s}_j\}$, once again without explicit supervision. Unlike section 1.1, we only need to apply segmentation on a single image. We use the robust *Segment-Anything* method [3] as it has good performance outside real-life images, which is the case for style images in artistic style transfer. We use the official pretrained weights based on ViT-H [1].

Directly applying the method results in a set of regions $\{\mathbf{s}_j\}$ which may overlap with each other. To fix this issue, we first sort the regions in decreasing order of size, and run the procedure in Algorithm 1. Here, $\{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$ is the list of regions from largest to smallest; $\mathbf{m} \in \mathbb{R}^{H\times W}$ is a binary map that keep tracks if a pixel has been assigned to a

Algorithm 1 Filtering overlapping style regions

$\mathbf{m} \leftarrow 0$
$\mathbf{k} \leftarrow -1$
$i \leftarrow 0$
for $\mathbf{s}_j$ in $\{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$ **do**
  if $\sum \mathbf{m}[\mathbf{s}_j]/|\mathbf{s}_j| \geq \lambda_t$ and $|\mathbf{s}_j|/\|\mathbf{s}\| \geq \lambda_m$ **then**
    $(\mathbf{m}[\mathbf{s}_j]) \leftarrow 1$
    $(\mathbf{k}[\mathbf{s}_j]) \leftarrow i$
    $i \leftarrow i + 1$
  end if
end for

style region; and $\mathbf{k} \in \mathbb{R}^{H \times W}$ is the segmentation map. The notation $\mathbf{m}[\mathbf{s}_j]$ and $\mathbf{k}[\mathbf{s}_j]$ represents the subset of pixels in $\mathbf{m}$ and $\mathbf{k}$ belonging to region $\mathbf{s}_j$. λ_t determines if the current $\mathbf{s}_j$ is overlapping with previous regions; λ_m ensures that regions too small are not considered. We set λ_t as 0.05 and λ_m as 0.004 (i.e. 0.4% of total image area).

After the procedure, each pixel of $\mathbf{k}$ should be given an integer value from $[-1, S-1]$, where 0 to $S-1$ indicates the S style regions. An index of -1 means that the pixel is not assigned to any region, and is not considered during stylization.

Last but not least, $\mathbf{k}$ is downscaled by nearest neighbor interpolation to match the dimensions of $\mathbf{f}_s$, the VGG features extracted from $\mathbf{s}$.

1.3. Fine vs. coarse regions

The values of C and S determines the level of fineness during the segmentation of scene and style regions. We provide an ablation experiment to experiment on the effect of using larger values of C and S on the stylization result.

The value of C can be indirectly controlled by modifying the number of iterations of training the unsupervised segmentation network; in general, by using a smaller number of iterations, the network output will contain a larger number of classes as it has not fully converged. For this experiment, we segment the style image $\mathbf{s}$ with the same procedure as scene regions, creating different segmentation maps with different values of S.

Figure 2 demonstrates segmentation results under three sets of scene regions and style regions. In the first example, the change from $C = 8, S = 14$ to $C = 19, S = 25$ results in a more varied stylization result; the walls and ceiling are dissected into smaller scene regions which are given different styles. However, "over-segmentation" of the style image $\mathbf{s}$ will only result in smaller style regions with similar patterns and colors, i.e. it will not create significant changes towards the stylization result, as demonstrated by the results of increasing C, S to $C = 41, S = 47$. A similar trend can be observed for the second example as well.

1.4. Injective and surjective mapping

When computing the mapping $\mathcal{M}$, our current method assumes that $C \leq S$, i.e. $\mathcal{M}$ is injective. Under this assumption, every scene region is matched to a unique style region, which prevents a single local style from dominating the stylization.

However, the computation of our style loss $\mathcal{L}_S$ can take in any arbitrary mapping function. For example, in the case where $C < S$, we can produce a *surjective mapping* where every style region has to be used for stylization.

To demonstrate this point, we provide an additional ablation experiment comparing between injective mapping and surjective mapping in Fig. 1. Under our default injective setting, we have $C = 4, S = 15$. By reducing the number of training iterations during the segmentation of scene regions, we can increase the number of C from 4 to 26. To generate a surjective mapping with 26 scene regions, we run our current Hungarian algorithm matching to obtain a bijective mapping for 15 scene regions; then run the algorithm again to match the remaining 9 scene regions.

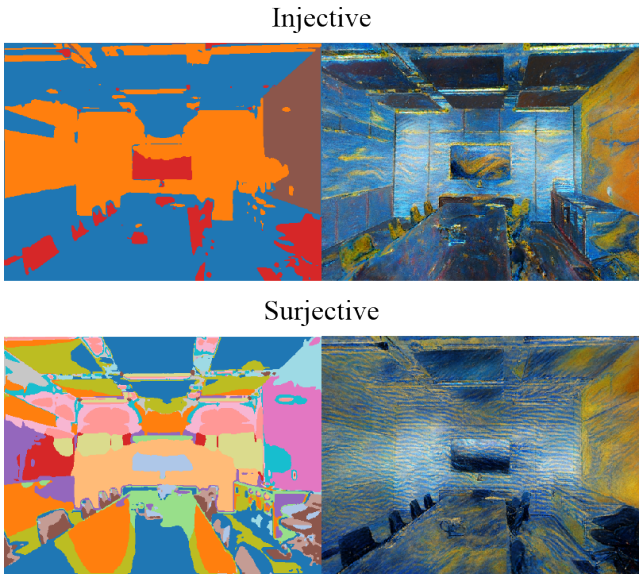

Figure 1. Injective vs. surjective mapping comparison.

In this example, we observe that the increase in scene regions means a more diverse stylization result; for example, the chairs are no longer constrained to be stylized in the same style as the ceiling. Nevertheless, many scene regions similar in nature are stylized similarly. This arises from the fact that multiple scene regions are now assigned to the same style region, or style regions that have similar patterns and appearances.

One future direction to extended our current method is to improve the automatic matching algorithm such that any general mapping $\mathcal{M}$ can be considered as a candidate.

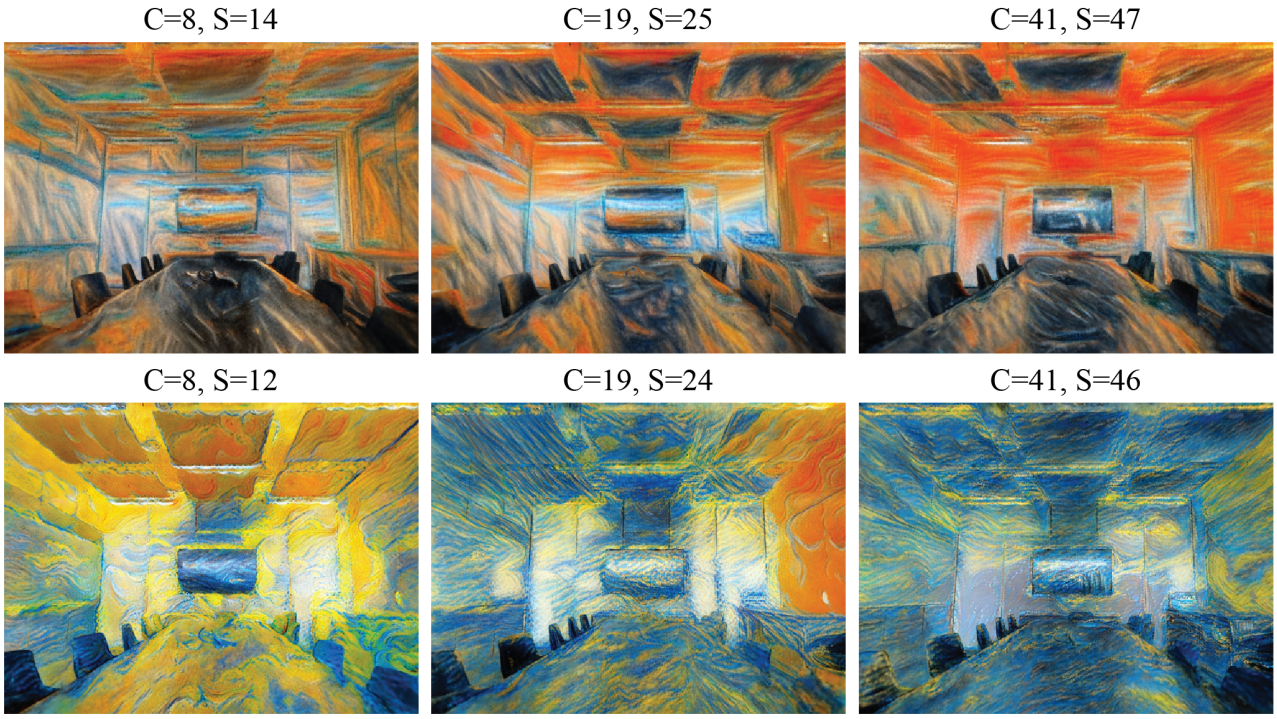

Figure 2. Stylization results by using segmentation maps of different degree of fineness.

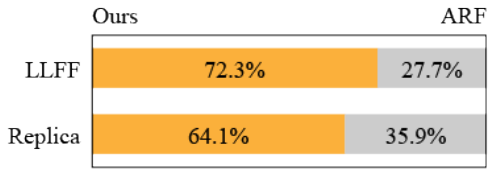

Figure 3. User study results.

2. User study

We conduct a user study to verify the performance of our proposed method compared with ARF [6]. The study consists of 20 questions presented in random order; each question consists of two images rendered from a scene stylized by our method; as well as the two ARF-generated images with the same scene and camera pose. The order of the two choices are randomized. In addition, the corresponding ground truth training image and the style images are also shown. The user is asked to select the choice that preserves the content of the ground truth image, and simultaneously appears similar to that of the style image. Out of the 20 questions, 12 of them correspond to images rendered from the `trex`, `room` and `fern` scenes from the LLFF [4] dataset; and 8 of them correspond to images rendered from the `office3` and `frl_apartment3` scenes in the Replica dataset [5].

We collected a total of 23 replies and the percentage of picking each method is summarized in Figure 3. The study shows that on average our method is picked at a higher percentage than ARF on both the LLFF and Replica datasets.

3. Further qualitative results

We show in this section the results of simultaneously training the stylization of multiple styles within a single hash grid. Figure 4 shows the `fern` scene from LLFF stylized in four distinct styles. Figure 5 compares the difference between with and without regional matching.

We provide further qualitative results for the LLFF dataset in Figure 6, and for the Replica dataset in Figure 7. Both figures illustrate that our method is less likely to transfer similar, repetitive patterns to the NeRF scene. This is especially the case in low-frequency regions, e.g. concrete walls and bare surfaces.

Our algorithm for matching content-style regions work by assuming that the regions can be paired up in a meaningful sense. However, even in cases where there is little to no correlation between regions from the NeRF scene and style image, our method is able to transfer local patterns on the scene.

Finally, we provide two further examples of modifying the pairing between content and style regions in Figure 8. We demonstrate that our method can achieve diverse and customizable stylization results via adjusting the pairing.

Figure 4. Multiple styles rendered from the same model.

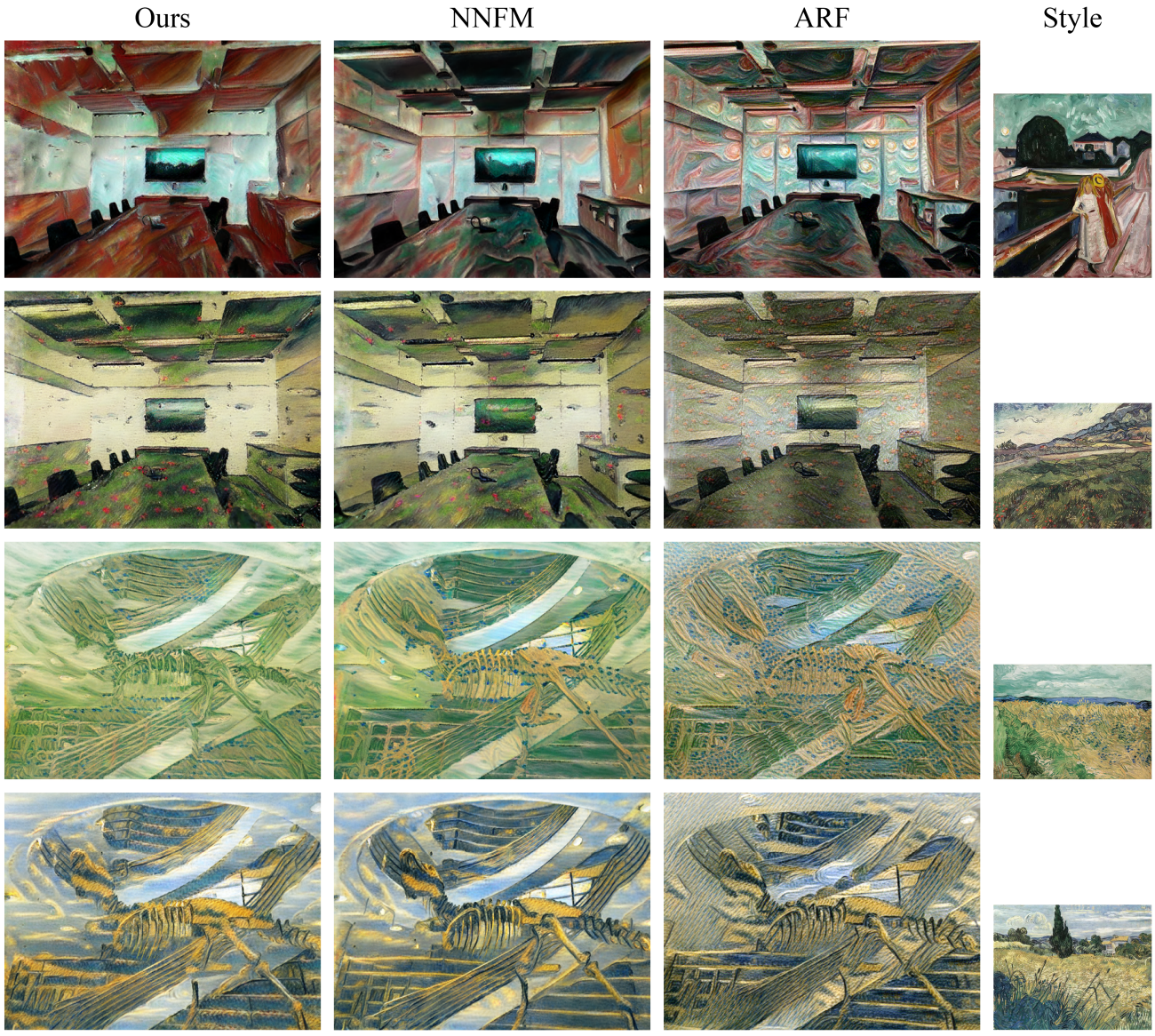

Figure 5. Comparison with and without regional matching. Left column uses regional matching; middle column do not use regional matching, i.e. reduces to the NNFM loss which looks for closest feature vector across entire image; Right column shows result from ARF.

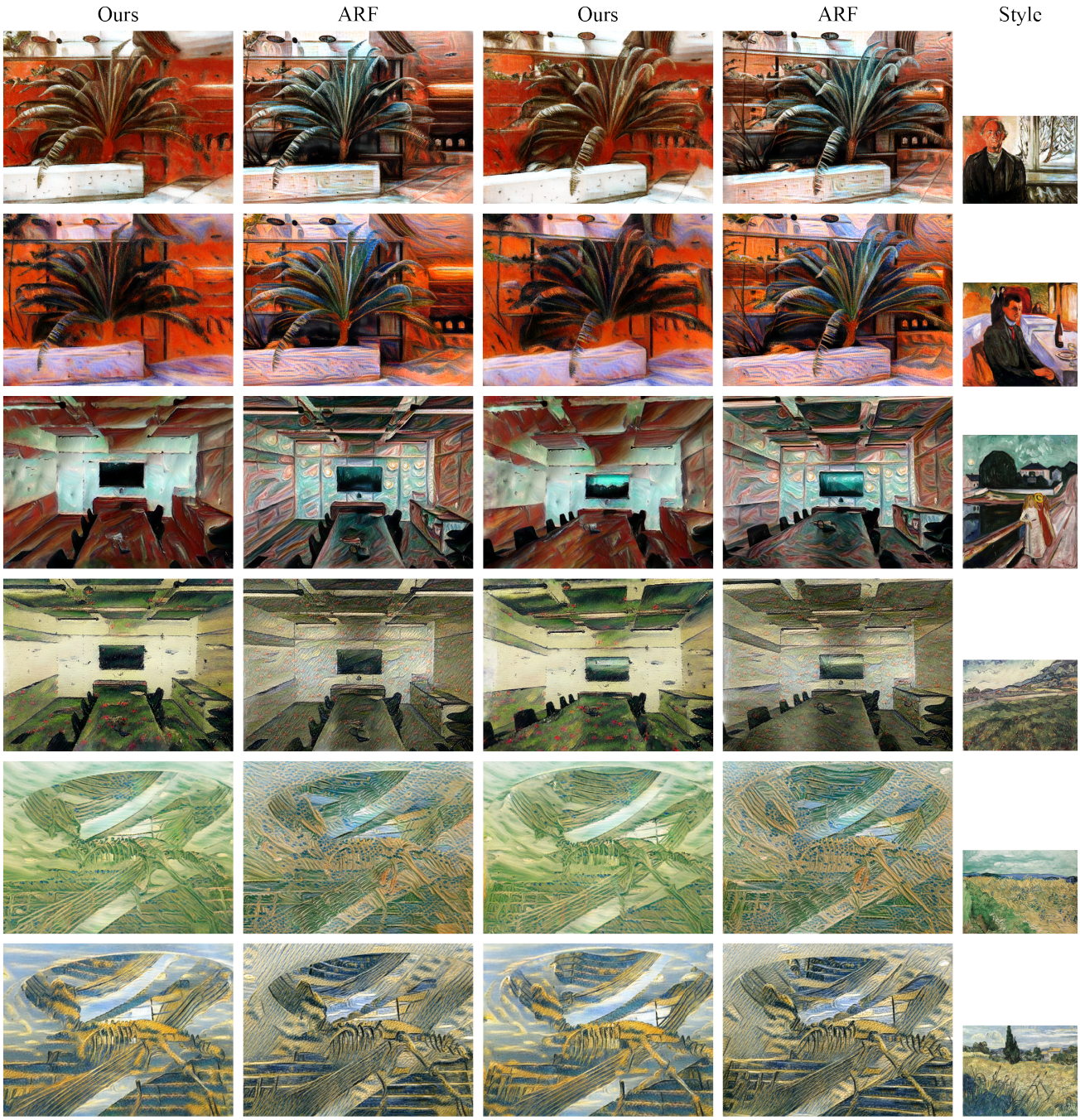

Figure 6. Further qualitative comparison on LLFF dataset.

Figure 7. Further qualitative comparison on Replica dataset.

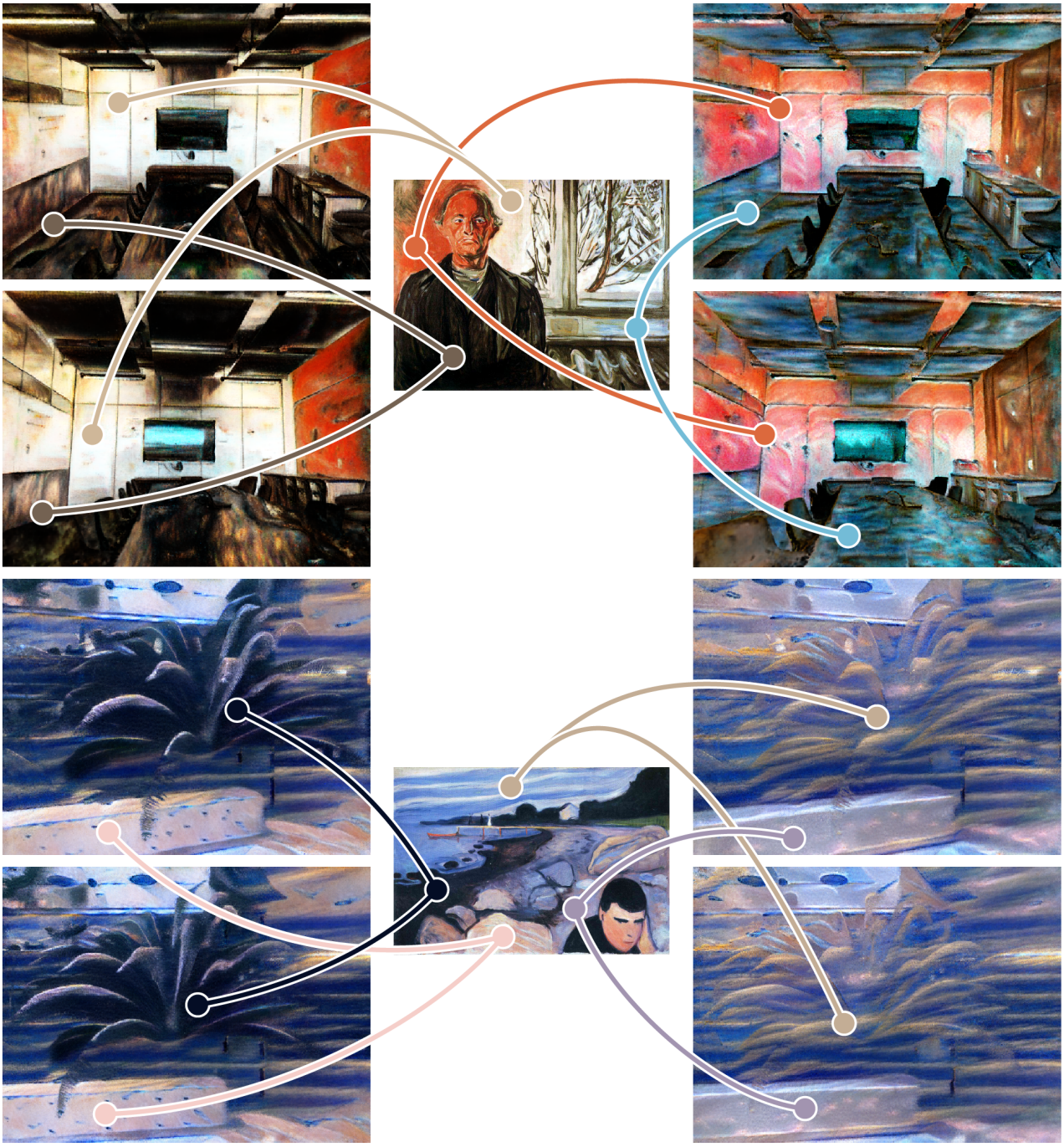

Figure 8. Effect of modifying the pairing between content and style regions. In each example, two content regions have been mapped to two different style regions in the style image (middle column), leading to two completely different stylization results (left and right columns).

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations*, 2021. 1

[2] Wonjik Kim, Asako Kanezaki, and Masayuki Tanaka. Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transaction on Image Processing*, 2020. 1

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 1

[4] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 3

[5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3

[6] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 3