StyleRF: Zero-shot 3D Style Transfer of Neural Radiance Fields
Supplementary Material

Kunhao Liu[1] Fangneng Zhan[2] Yiwen Chen[1] Jiahui Zhang[1]
Yingchen Yu[1] Abdulmotaleb El Saddik[3,5] Shijian Lu[1*] Eric Xing[4,5]

[1]Nanyang Technological University [2]Max Planck Institute for Informatics
[3]University of Ottawa [4]Carnegie Mellon University [5]MBZUAI

We refer to the project website (https://kunhao-liu.github.io/StyleRF/) for more results, comparisons with baselines, and application demonstrations. This document provides supplementary materials in 1) proof of deferred style transformation, 2) implementation details, 3) runtime comparisons, 4) user studies, and 5) limitations, more details to be found in the ensuing subsections.

1. Proof of Deferred Style Transformation

In this section, we provide proof that applying deferred style transformation (DST) on 2D feature maps is equivalent to applying style transformation on 3D points' features. This is also the full derivation of Eq. (9) in the main paper.

The DST on 2D feature maps can be mathematically formulated by:

$$F_{cs} = conv\left(T \otimes \bar{F}_c\right) \times \sigma(F_s) + w_{\mathbf{r}} \times \mu(F_s),$$

$$\text{where} \quad \bar{F}_c = \sum_{i=1}^{N} w_i \bar{F}_i, w_{\mathbf{r}} = \sum_{i=1}^{N} w_i, \mathbf{r} \in \mathcal{R}$$

w_i denotes the weight of sampled point i, $\bar{F}_i$ denotes the feature of sample i after sampling-invariant content transformation, $\mathcal{R}$ denotes the set of rays in each training batch, $conv$ is a 1×1 convolution layer without bias, and $\otimes$ denotes matrix multiplication.

Note that 1×1 convolution layer without bias is basically a matrix multiplication operation. We can thus move the scalar multiplication outside the $conv$ as follows:

$$\begin{aligned}
F_{cs} &= conv\left(T \otimes \sum_{i=1}^{N} w_i \bar{F}_i\right) \times \sigma(F_s) + \sum_{i=1}^{N} w_i \times \mu(F_s) \\
&= \sum_{i=1}^{N} w_i conv\left(T \otimes \bar{F}_i\right) \times \sigma(F_s) + \sum_{i=1}^{N} w_i \times \mu(F_s) \\
&= \sum_{i=1}^{N} w_i \left(\underbrace{conv\left(T \otimes \bar{F}_i\right) \times \sigma(F_s) + \mu(F_s)}_{\text{(i)}}\right) \\
&= \sum_{i=1}^{N} w_i \left(F_{\text{(i)}}\right)
\end{aligned}$$

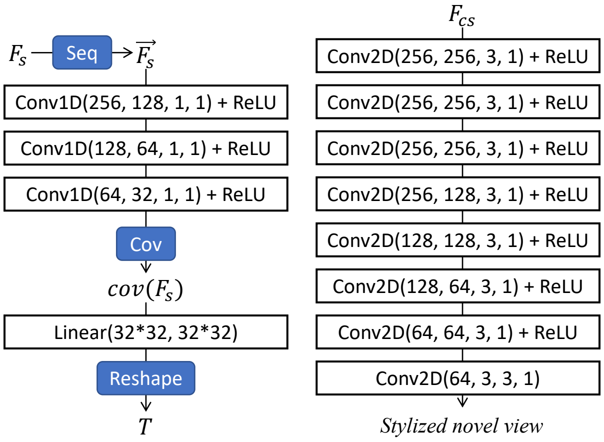
We can see that $\sum_{i=1}^{N} w_i \left(F_{\text{(i)}}\right)$ has the same form as the volume rendering (Eq. (1) in the main paper), where $F_{\text{(i)}}$ becomes the transformed feature of the sampled point. Hence, DST is equivalent to a transformation that is independently applied to each 3D point, which keeps the multi-view consistency.

2. Implementation Details

Style transformation matrix generation network. We present the architecture of the style transformation matrix T in Fig. 1(a). Following [7], we formulate T from the feature covariance $cov(F_s)$. Specifically, we apply a sequence of 1D convolution layers to the sequentialized style features $\overrightarrow{F_s}$, and then compute the feature covariance $cov(F_s)$. Finally, we apply a linear layer to the feature covariance to get the style transformation matrix T.

Estimation of the mean and variance. We adopted the running estimation technique to estimate the mean and variance. At training iteration $i (i > 0)$, the estimated statistics $\hat{x}_i$ is computed by $\hat{x}_i = mx_i + (1 - m)\hat{x}_{i-1}$, where m is the momentum which is set to $1e - 4$, x_i is the calculated batch statistics (i.e. mean and variance), and $\hat{x}_0 = x_0$.

[*] Shijian Lu is the corresponding author.


F_s
Seq
$\vec{F_s}$
Conv1D(256, 128, 1, 1) + ReLU
Conv1D(128, 64, 1, 1) + ReLU
Conv1D(64, 32, 1, 1) + ReLU
Cov
$cov(F_s)$
Linear(32*32, 32*32)
Reshape
T
F_{cs}
Conv2D(256, 256, 3, 1) + ReLU
Conv2D(256, 256, 3, 1) + ReLU
Conv2D(256, 256, 3, 1) + ReLU
Conv2D(256, 128, 3, 1) + ReLU
Conv2D(128, 128, 3, 1) + ReLU
Conv2D(128, 64, 3, 1) + ReLU
Conv2D(64, 64, 3, 1) + ReLU
Conv2D(64, 3, 3, 1)
Stylized novel view

(a) Style transformation matrix (b) Decoder

Figure 1. **Model architecture.** We present the architectures of the style transformation matrix generation network in (a) and the CNN decoder in (b). The formatting of convolution layers is (number of input channels, number of output channels, kernel size, stride) and the formatting of linear layers is (number of input channels, number of output channels). Seq denotes sequentialization, Cov denotes covariance matrix.

CNN decoder. We present the architecture of the CNN decoder in Fig. 1(b). Unlike [4, 9], we do not use a standard U-Net [11] decoder that first downsamples and then upsamples the feature maps. According to our experiments, the standard U-Net decoder introduces clear multi-view inconsistency as the downsample-upsample operations increase the receptive field of the decoder. Thus we construct the decoder only using a sequence of 2D convolution layers without downsample-upsample operations.

Training settings. We train our model on a single RTX A5000 GPU. We use style images from WiKiArt [10] as the training data, which contains about $80,000$ samples. In the feature grid training stage (first stage), we train the model for 25K iterations using the Adam optimizer [6]. The learning rate of the feature grid is set to 0.02 and the learning rate of the decoder is set to 1e-4. We further apply a TV regularization to the feature grid to encourage the smoothness of the feature grid which improves the stylization quality clearly. In the stylization training stage (second stage), we also train the model for 25K iterations using the Adam optimizer. We freeze the feature grid and finetune the decoder with the learning rate of 1e-5 and train the stylization module with the learning rate of 1e-4. We set $C = 256$ and $C' = 32$. During training, we randomly sample rays from all training rays while calculating the first term of Eq.(10) and randomly crop patches from a random training view while calculating the last two terms.

Method	First-stage training	Second-stage training	Inference (per frame)
Hyper [2]	~4 days	~2 days	50s
Ours	1h 58min	3h 10min	18s

Table 1. Runtime comparisons.

Method	Consistency	Stylization
AdaIN [5]	0.02	0.05
CCPL [13]	0.07	0.08
ReReVST [12]	0.26	0.20
LSNV [4]	0.13	0.17
Hyper [2]	0.10	0.03
Ours	0.42	0.47

Table 2. **User study.** We present the user preference scores of StyleRF and the state-of-the-art baselines. Best score, second best score and thrid best score are in red, orange and yellow respectively.

3. Runtime Comparisons

Tab. 1 shows quantitative comparisons with another NeRF-based zero-shot style transfer method Hyper [7] on training and inference time. All the evaluations are performed on a single NVIDIA RTX A5000 GPU with 24G of memory and tested on 4 scenes of LLFF [32] dataset.

4. User Study

We perform a user study to compare our method to the 3D zero-shot style transfer baselines. The user study involved 30 participants with different occupations, ages, and races. Specifically, we show each user a series of stylization results, including a video of original scene, a style image and corresponding videos stylized by our StyleRF and baselines. The users then choose one stylized video that better matches the given style image and one video that has better multi-view consistency. In total, we provide 30 scene-style pairs, which are randomly divided into 6 groups. Each user is asked to rate a random group. The results in Tab. 2 demonstrate the superiority of the proposed StyleRF.

5. Limitations

Despite the superior stylization quality and generalizability to new styles, StyleRF has two limitations. First, StyleRF has to be trained for every 3D scene since NeRF [8] is a per-scene-per-model method. This limitation could be mitigated by incorporating generalizable NeRF models such as [14] for better generalizability to new 3D scenes. Second, the current StyleRF does not support 360° unbounded scenes. We can extend StyleRF with multi-sphere images like [3] or use coordinate parameterization like [1, 15] for supporting 360° unbounded scenes.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[2] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1475–1484, 2022. 2

[3] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[4] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13869–13878, 2021. 2

[5] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[7] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019. 1

[8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[9] Fangzhou Mu, Jian Wang, Yicheng Wu, and Yin Li. 3d photo stylization: Learning to generate stylized novel views from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16273–16282, 2022. 2

[10] K. Nichol. Painter by numbers, 2016. 2

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[12] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. Consistent video style transfer via relaxation and regularization. *IEEE Trans. Image Process.*, 2020. 2

[13] Zijie Wu, Zhen Zhu, Junping Du, and Xiang Bai. Ccpl: Contrastive coherence preserving loss for versatile style transfer. In *European Conference on Computer Vision*, pages 189–206. Springer, 2022. 2

[14] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 2

[15] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2