Stylizing 3D Scene via Implicit Representation and HyperNetwork

(Supplementary Materials)

Pei-Ze Chiang*[1,3] Meng-Shiun Tsai*[1,3] Hung-Yu Tseng[2] Wei-Sheng Lai[2] Wei-Chen Chiu[1,3]
[1]National Yang Ming Chiao Tung University, Taiwan [2]University of California, Merced
[3]MediaTek-NCTU Research Center, Taiwan

A. Detailed Model Architecture

A.1. Neural Radiance Fields

The detailed architecture of our NeRF-based scene representation is illustrated in Figure 1. Basically, F^{base}, F^{geo}, and F^{app} are built upon 5-layer, 2-layer, and 5-layer multilayer perceptrons (MLPs) respectively, where we use a ReLU activation function between every adjacent layer except for the first two layers of F^{app}.

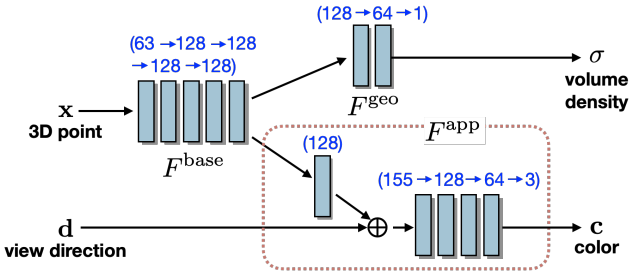

Figure 1: **The detailed architecture of the neural radiance fields model used for our scene representation.** As following [11], both 3D location $\mathbf{x}$ and viewing direction $\mathbf{d}$ are firstly transformed into positional embeddings before being utilized by F^{base} and F^{app} respectively. Note that the numbers of neurons for all the layers in an MLP are sequentially provided within the corresponding blue bracket shown in the figure.

A.2. Style Variational Autoencoder (style-VAE)

Instead of merely having our stylization effective for a limited number of styles which are seen during hypernetwork learning, we aim to achieve the **universal stylization** thus being able to render scene images with arbitrary and unseen styles in the inference/testing time. To this end, we propose the style variational autoencoder (style-VAE) to regularize the distribution of style latent vectors into a normal distribution, and expect that the hypernetwork Ψ is generalizable to the unseen styles.

The architecture of our style-VAE is illustrated in Figure 2, where we extend from the well-known AdaIN framework [3] of image style transfer to additionally equip a variational autoencoder (VAE [5]) into the space of style features f_s extracted from style reference images S by an ImageNet-pretrained VGG-19 encoder E [8]. Basically, such variation autoencoder is composed of an encoder E_{vae} and a decoder D_{vae} (both built upon MLPs), where E_{vae} encodes $\{\mu(f_s), \Sigma(f_s)\}$ (i.e. the mean and standard deviation of style feature f_s) into a Gaussian distribution $\mathcal{N}(\rho, \xi)$ and D_{vae} decodes from $\epsilon \sim \mathcal{N}(\rho, \xi)$ to produce the reconstructed $\{\hat{\mu}(f_s), \hat{\Sigma}(f_s)\}$. Note that, we use ρ obtained from $E_{vae}(\{\mu(E(S)), \Sigma(E(S))\})$ as the style latent vector z_S of the style reference image S. Afterwards, $\{\hat{\mu}(f_s), \hat{\Sigma}(f_s)\}$ are used to perform the adaptive instance normalization on the content feature f_c (extracted from the content image by E) for realizing the image style transfer on the content image as what the typical AdaIN framework does.

The learning objective of our style-VAE simply includes the loss functions for training the typical AdaIN and VAE frameworks, i.e. the content and style losses from AdaIN [3], as well as the reconstruction loss (between $\{\mu(f_s), \Sigma(f_s)\}$ and $\{\hat{\mu}(f_s), \hat{\Sigma}(f_s)\}$) and the KL-divergence loss on $\mathcal{N}(\rho, \xi)$ from VAE [5]. In particular, the KL-divergence loss regularizes the distribution of latent style vectors to follow the normal distribution. While having a sufficient number of training styles for learning the hypernetwork Ψ, we expect that the latent style vectors of unseen styles which are projected into the same latent distribution can be well handled by the hypernetwork thus producing the plausible W^{app} for driving the scene stylization. Please note that, as the style-VAE aims for learning to extract the latent style vectors from the style reference images where none of its objective functions is related to the stylization part, it thus can be learnt beforehand and kept fixed during training our scene stylization model.

∗ Both authors contributed equally to the paper

For more architecture details of the E_{vae} and D_{vae} used in our style-VAE, they are both 3-layer MLPs with having a ReLU activation function between every adjacent layer. The numbers of neurons for all the layers in E_{vae} are sequentially $(1024 \rightarrow 1024 \rightarrow 1024)$. The output of E_{vae} is a 1024-dimensional vector, in which its first half and the second half are ρ and ξ respectively (i.e. both are 512-dimensional vectors). And the numbers of neurons for all the layers in D_{vae} are sequentially $(512 \rightarrow 1024 \rightarrow 1024)$.

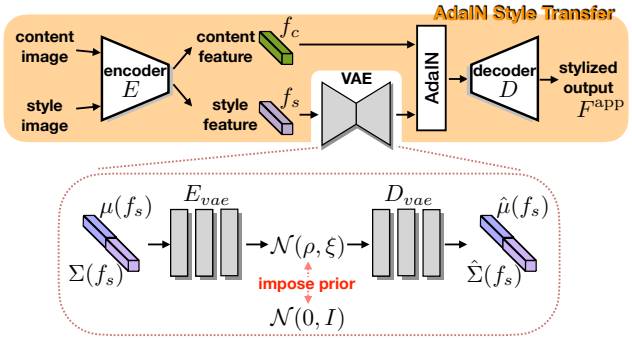

Figure 2: **Illustration of pre-training of the the style variational autoencoder (style-VAE) model.** Built upon the AdaIN framework [3] of image style transfer, a variational autoencoder [5] learns to project the style features f_s into the latent style vectors, where the distribution of latent style vectors is regularized to follow a normal distribution for the purpose of enabling universal stylization.

A.3. Hypernetwork

As illustrated in Figure 3(b) of the main paper, given a style latent vector z_S extracted from the style reference image S, the hypernetwork Ψ (built upon an MLP in our work) learns to regress from z_S to produce the weights W^{app} for updating F^{app}, in which the updated F^{app} (denoted as $\tilde{F}^{\text{app}}$) is then able to predict the color value emitted from 3D location $\mathbf{x}$ toward view direction $\mathbf{d}$, and the resultant images of view synthesis after running volume rendering are expected to have the style as S. Please note that, the idea of hypernetwork (i.e. regressing the network parameters via another network) was originally proposed in [2] but we innovatively adopt it here for the task the 3D scene stylization.

For more architecture details of the hypernetwork Ψ, it maps the style latent vector z_S (which is 512-dimensional) to the weights W^{app} of F^{app}. The weights w_l^{app} of each layer l of F^{app} is the output of a separate sub-hypernetwork Ψ_l, where each sub-hypernetwork Ψ_l is a 3-layer MLP. The numbers of neurons for all the layers in Ψ_l are sequentially $(512 \rightarrow 64 \rightarrow |w_l^{\text{app}}|)$, where $|w_l^{\text{app}}|$ is the size of w_l^{app}.

B. Consistency Metric

As described in Section 4.2 of the main paper, to evaluate the consistency of two stylized images $\tilde{\mathcal{I}}_u$ and $\tilde{\mathcal{I}}_v$ at novel views u and v, we use their corresponding ground-truth *non-stylized* images $\mathcal{I}_u$ and $\mathcal{I}_v$ to compute the optical flow, as well as the occlusion mask $\mathcal{O}$. Specifically, we use the pre-trained FlowNet 2.0 [4] to obtain the optical flow. According to the computed optical flow, we warp the stylized image $\tilde{\mathcal{I}}_v$ at the view v to get the corresponding image $\breve{\mathcal{I}}_u$ at the view u. Finally, the consistency metric is computed as

$$\mathcal{E}_{\text{consistency}}(\tilde{\mathcal{I}}_u, \tilde{\mathcal{I}}_v) = \frac{1}{|\mathcal{O}|} \left\| \tilde{\mathcal{I}}_u - \breve{\mathcal{I}}_u \right\|_2^2 \qquad (1)$$

where $|\mathcal{O}|$ denotes the number of non-occluded pixels in $\mathcal{O}$.

C. Ablation Study

Numbers of Training Style Images. For all experiments in the main paper, we use 81330 style images to train the hypernetwork for universal stylization. In this experiment, we demonstrate the generalization ability of the proposed framework by lowering the number of training style images. As shown in Figure 3, the proposed hypernetwork trained with 200 or 2000 style images is still able to produce appealing stylization results.

Joint Training v.s. Proposed Two-Stage Training Strategy. As described in Section 3.3 of the main paper, we design a two-stage training strategy to train the proposed model for stylizing a particular 3D scene. In this experiment, we validate the importance of the two-stage training strategy by comparing with the geometry branch of the neural radiance fields model (i.e., F^{base} and F^{geo}) and hypernetwork Ψ jointly trained with $\mathcal{L}_{\text{second}}$ (see Eq. (4) of the main paper). We present the results in Figure 4. The joint training strategy is a more complicated learning task since it involves the construction of the target 3D scene, and the learning of the stylization. As a result, the proposed model trained with the joint training strategy fails to render the desired style as well as the correct geometry. In contrast, we develop the two-stage training strategy to simplify the training task that the geometry branch is first optimized to model the target 3D scene, the hypernetwork is trained for the universal stylization.

D. More Stylization Results

As shown in Figure 5, we provide more stylization results of our proposed method. In addition, we provide qualitative comparisons against different baselines (i.e., AdaIN [3], WCT [7], LST [6], TPFR [9], ReReVST [10] and MCCNet [1]) in Figure 6 and our project page[1] .

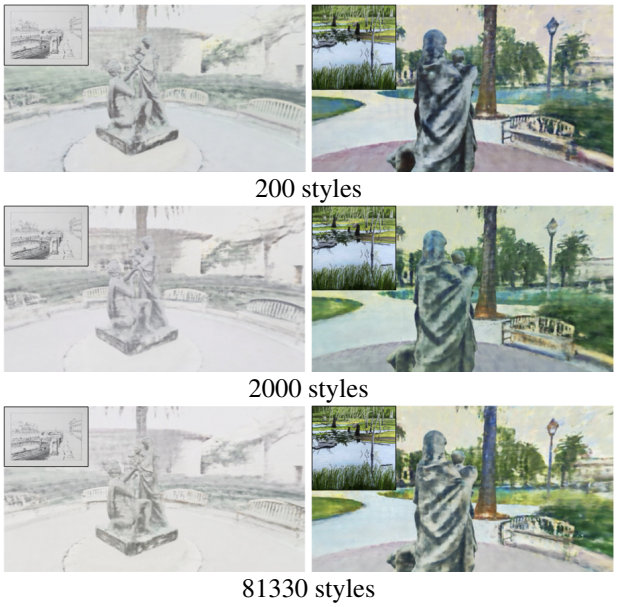

Figure 3: **Study on number of style images used for the stylization training stage.** We present the example stylization results based on the unseen style (shown on the top-left corner of each example) by using the hypernetwork trained on 81330, 2000, and 200 style images. Note that we use the same training hyper-parameters (e.g., learning rate) except the number of training images in this experiment.

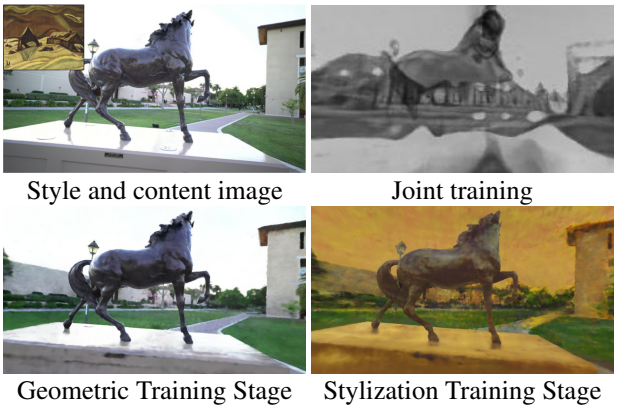

Figure 4: **Importance of the two-stage optimization strategy.** To understand the importance of our two-stage training strategy (as described in Section 3.3 of the main paper), we present the results of optimizing the proposed model in a single-stage, i.e., joint training the neural radiance field model and hypernetwork with using $\mathcal{L}_{\text{second}}$ (see Eq. (4) of the main paper) to learn the geometry and stylization at the same time. The joint training approach fails to capture the geometry of the target scene and render the images with desired style while not adopting the proposed two-stage optimization strategy.

References

[1] Yingying Deng, Fan Tang, Weiming Dong, haibin Huang, Ma chongyang, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *AAAI*, 2021.

[2] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[4] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[6] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[9] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. Consistent video style transfer via relaxation and regularization. *IEEE Transactions on Image Processing (TIP)*, 2020.

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

[1] Project page: https://ztex08010518.github.io/3dstyletransfer/

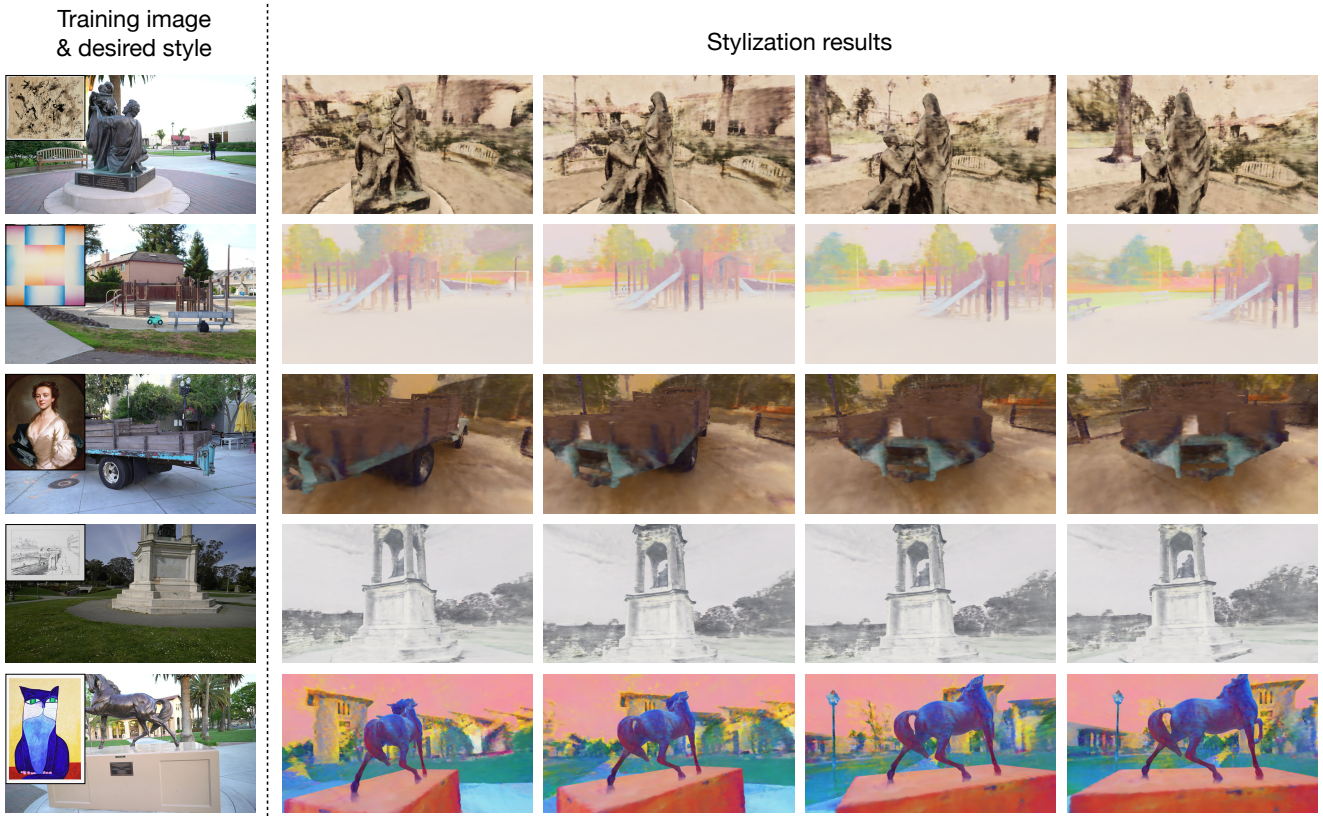

Figure 5: **Qualitative results of our proposed framework of 3D scene stylization.** For each row, the leftmost column presents one of the training images of the target scene together with the input reference (style) image on the top-left corner, while the remaining columns demonstrate the stylization results at various novel views.

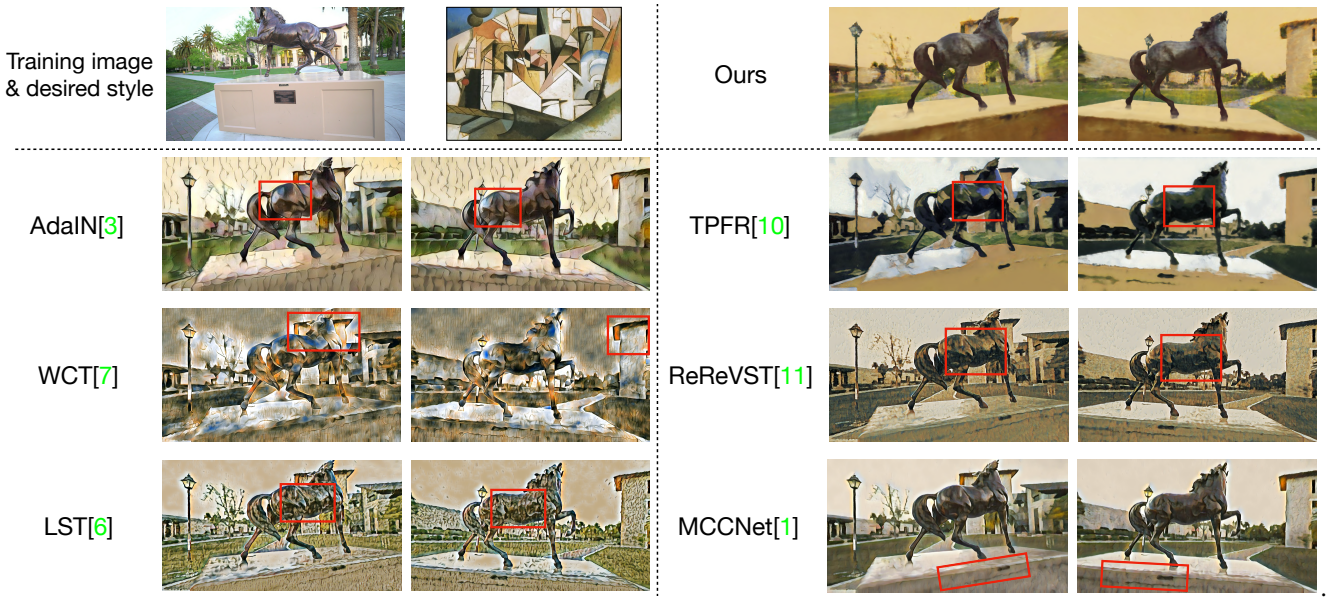

Figure 6: **Qualitative comparisons.** The bottom row presents one of the training images of the target scene with the input reference (style) image and the stylization results of our proposed approach. The red boxes highlight the inconsistent stylization across different views, while our proposed method is consistent across different view angles with desired style.