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

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

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

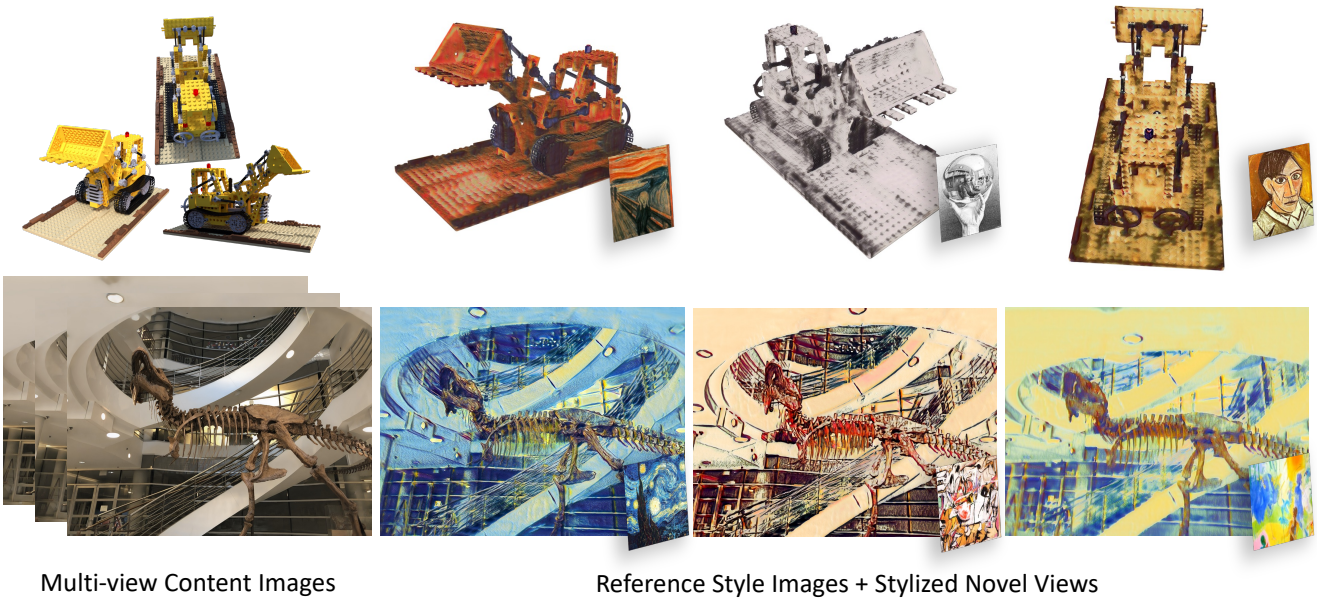


Figure 1. **Zero-shot 3D Style Transfer.** Given a set of *Multi-view Content Images* of a 3D scene, StyleRF can transfer arbitrary *Reference Styles* to the 3D scene in a zero-shot manner, rendering high-quality *Stylized Novel Views* with superb multi-view consistency.

Abstract

3D style transfer aims to render stylized novel views of a 3D scene with multi-view consistency. However, most existing work suffers from a three-way dilemma over accurate geometry reconstruction, high-quality stylization, and being generalizable to arbitrary new styles. We propose StyleRF (Style Radiance Fields), an innovative 3D style transfer technique that resolves the three-way dilemma by performing style transformation within the feature space of a radiance field. StyleRF employs an explicit grid of high-level features to represent 3D scenes, with which high-fidelity geometry can be reliably restored via volume rendering. In addition, it transforms the grid features according to the reference style which directly leads to high-quality zero-shot style transfer. StyleRF consists of two innovative designs. The first is sampling-invariant content transformation that makes the transformation invariant to the holistic statistics of the sampled 3D points and accordingly ensures multi-view consistency. The second is deferred style transformation of 2D feature maps which is equivalent to the transformation of 3D points but greatly reduces memory footprint without degrading multi-view consistency. Extensive experiments show that StyleRF achieves superior 3D stylization quality with precise geometry reconstruction and it can generalize to various new styles in a zero-shot manner. Project website: https://kunhao-liu.github.io/StyleRF/

1. Introduction

Given a set of multi-view images of a 3D scene and an image capturing a target style, 3D style transfer aims to generate novel views of the 3D scene that have the target style consistently across the generated views (Fig. 1). Neural style transfer has been investigated extensively, and state-of-the-art methods allow transferring arbitrary styles in a zero-shot manner. However, most existing work focuses

on style transfer across 2D images [15, 21, 24] but cannot extend to a 3D scene that has arbitrary new views. Prior studies [19, 22, 37, 39] have shown that naively combining 3D novel view synthesis and 2D style transfer often leads to multi-view inconsistency or poor stylization quality, and 3D style transfer should optimize novel view synthesis and style transfer jointly.

However, the current 3D style transfer is facing a three-way dilemma over accurate geometry reconstruction, high-quality stylization, and being generalizable to new styles. Different approaches have been investigated to resolve the three-way dilemma. For example, multiple style transfer [11, 22] requires a set of pre-defined styles but cannot generalize to unseen new styles. Point-cloud-based style transfer [19, 37] requires a pre-trained depth estimation module that is prone to inaccurate geometry reconstruction. Zero-shot style transfer with neural radiance fields (NeRF) [8] cannot capture detailed style patterns and textures as it implicitly injects the style information into neural network parameters. Optimization-based style transfer [17, 39, 63] suffers from slow optimization and cannot scale with new styles.

In this work, we introduce **StyleRF** to resolve the three-way dilemma by performing style transformation in the feature space of a radiance field. A radiance field is a continuous volume that can restore more precise geometry than point clouds or meshes. In addition, transforming a radiance field in the feature space is more expressive with better stylization quality than implicit methods [8], and it can also generalize to arbitrary styles. We construct a 3D scene representation with a grid of deep features to enable feature transformation. In addition, multi-view consistent style transformation in the feature space could be achieved by either transforming the whole feature grid or transforming the sampled 3D points. We adopt the latter as the former incurs much more computational cost during training to stylize the whole feature grid in every iteration, whereas the latter can reduce computational cost through decreasing the size of training patch and the number of sampled points. However, applying off-the-shelf style transformations to a batch of sampled 3D points impairs the multi-view consistency as they are conditioned on the holistic statistics of the batch. Beyond that, transforming every sampled 3D point is memory-intensive since NeRF needs to query hundreds of sampled points along each ray for rendering a single pixel.

We decompose the style transformation into sampling-invariant content transformation (SICT) and deferred style transformation (DST), the former eliminating the dependency on holistic statistics of sampled point batch and the latter deferring style transformation to 2D feature maps for better efficiency. In SICT, we introduce volume-adaptive normalization that learns the mean and variance of the whole volume instead of computing them from a sampled batch. In addition, we apply channel-wise self-attention to transform each 3D point independently to make it conditioned on the feature of that point regardless of the holistic statistics of the sampled batch. In DST, we defer the style transformation to the volume-rendered 2D feature maps based on the observation that the style transformation of each point is the same. By formulating the style transformation by pure matrix multiplication and adaptive bias addition, transforming 2D feature maps is mathematically equivalent to transforming 3D point features but it saves computation and memory greatly. Thanks to the memory-efficient representation of 3D scenes and deferred style transformation, our network can train with 256×256 patches directly without requiring sub-sampling like previous NeRF-based 3D style transfer methods [8, 11, 22].

The contributions of this work can be summarized in three aspects. *First*, we introduce StyleRF, an innovative zero-shot 3D style transfer framework that can generate zero-shot high-quality 3D stylization via style transformation within the feature space of a radiance field. *Second*, we design sampling-invariant content transformation and deferred style transformation, the former achieving multi-view consistent transformation by eliminating dependency on holistic statistics of sampled point batch while the latter greatly improves stylization efficiency by deferring style transformation to 2D feature maps. *Third*, extensive experiments show that StyleRF achieves superior 3D style transfer with accurate geometry reconstruction, high-quality stylization, and great generalization to new styles.

2. Related Work

Neural scene representations. 3D scene representation has been extensively studied in recent years with different ways of representations such as volumes [23, 27, 46, 49, 59], point clouds [1, 45], meshes [25, 55], depth maps [20, 30], and implicit functions [7, 34, 41, 61]. These methods adopt differentiable rendering which enables model optimization by using 2D multi-view images. Among them, Neural Radiance Field (NeRF) [36] can render a complex 3D scene with high fidelity and accurate geometry. It represents scenes with an implicit coordinate function that maps each 3D coordinate to a density value and a color value, and employs volume rendering to generate images of novel views. However, the implicit coordinate function is represented by a large multilayer perceptron (MLP) that is often hard to optimize and slow to infer. Serval studies adopt a hybrid representation [3, 4, 10, 14, 31, 33, 38, 44, 52, 62] to speed up the reconstruction and rendering. They employ explicit data structures such as discrete voxel grids [14, 52], decomposed tensors [3, 4, 13], hash maps [38], etc. to store features or spherical harmonics, enabling fast convergence and inference.

Although most existing work extracts features as middle-

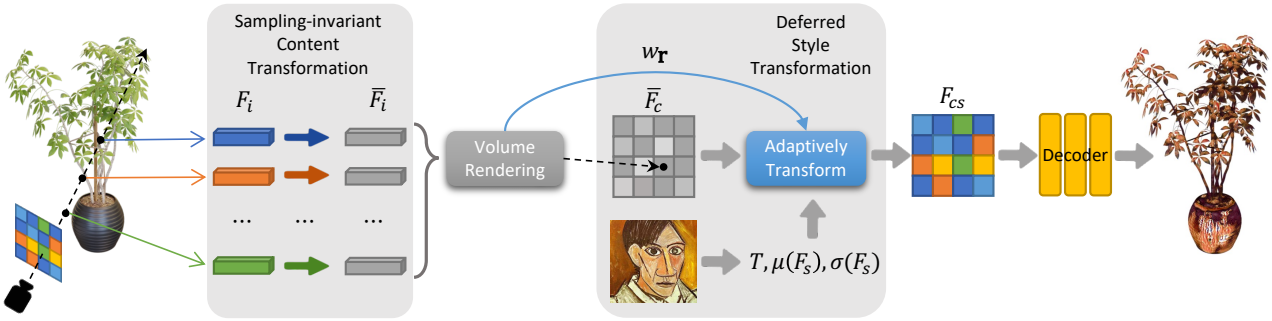


Figure 2. **The framework of StyleRF.** For a batch of sampled points along a ray $\mathbf{r}$, the corresponding features $F_i, i \in [1, 2, ..., N]$ are first extracted, each of which is transformed to $\bar{F}_i$ independently via *Sampling-Invariant Content Transformation*, regardless of the holistic statistics of the point batch. $\bar{F}_i$ is then transformed to a feature map $\bar{F}_c$ via *Volume Rendering*. After that, the *Deferred Style Transformation* transforms $\bar{F}_c$ to the feature map F_{cs} adaptively using the sum weight of the sampled points $w_\mathbf{r}$ along the ray $\mathbf{r}$ and the style information $T, \mu(F_s)$, and $\sigma(F_s)$. Finally, a stylized novel view is generated via a CNN decoder.

level representations of scenes, the extracted features are usually an intermediate output of neural networks which have little semantic meanings and are not suitable for the style transfer task. We introduce decomposed tensors [4] to store high-level features extracted by pre-trained CNNs, which enables transformations in feature space as well as efficient training and inference. Though [3, 16, 40] also render feature maps instead of RGB maps, they are computationally intensive and usually work with low-resolution feature maps. StyleRF can instead render full-resolution feature maps (the same as the output RGB images) efficiently and it uses high-level features largely for transformation only.

Neural style transfer. Neural style transfer aims at rendering a new image that contains the content structure of one image and the style patterns of another. The seminal work in [15] shows that multi-level feature statistics extracted from intermediate layers of pre-trained CNNs could be used as a representation of the style of an artistic image, but it treats style transfer as a slow and iterative optimization task. [9, 21, 24, 28, 29, 32, 43, 50, 58] utilize feed-forward networks to approximate the optimization procedure to speed up rendering. Among them, [9, 21, 28, 29, 32, 43, 50, 58] can achieve zero-shot style transfer by applying transformations to the high-level features extracted by pre-trained CNNs, where the feature transformations can be achieved by matching second-order statistics [21, 29], linear transformation [28, 58] , self-attention transformation [9, 32, 43], etc. Video style transfer extends style transfer to videos for injecting target styles consistently across adjacent video frames. Several studies leverage optical flow [5, 18, 47, 56, 57] as temporal constraints to estimate the movement of video contents. They can produce smooth videos, but have little knowledge of the underlying 3D geometry and cannot render consistent frames in arbitrary views [19, 37].

Huang et al. first tackle stylizing complex 3D scenes [19]. They construct a 3D scene by back-projecting image features into the 3D space to form a point cloud and then perform style transformation on the features of 3D points. Their method can achieve zero-shot style transfer, but requires an error-prone pre-trained depth estimator to model scene geometry. [37] also constructs a point cloud for stylization but it mainly focuses on monocular images. Instead, [6, 8, 11, 22, 39, 63] use NeRF [36] as the 3D representation which can reconstruct scene geometry more faithfully. [6] is a photorealistic style transfer method that can only transfer the color tone of style images. [39, 63] achieve 3D style transfer via optimization and can produce visually high-quality stylization, but they require a time-consuming optimization procedure for every reference style. [11, 22] employ latent codes to represent a set of pre-defined styles, but cannot generalize to unseen styles. [8] can achieve arbitrary style transfer by implicitly instilling the style information into MLP parameters. However, it can only transfer the color tone of style images but cannot capture detailed style patterns. StyleRF can transfer arbitrary style in a zero-shot manner, and it can capture style details such as strokes and textures as well.

3. Method

The overview of StyleRF is shown in Fig. 2. For a batch of sampled points along a ray $\mathbf{r}$, the corresponding features $F_i, i \in [1, 2, ..., N]$ are first extracted from the feature grid described in Sec. 3.1, each of which is transformed to $\bar{F}_i$ independently via *Sampling-Invariant Content Transformation (SICT)* described in Sec. 3.2.1, regardless of the holistic statistics of the point batch. $\bar{F}_i$ is then transformed to a feature map $\bar{F}_c$ via *Volume Rendering*. After that, the *Deferred Style Transformation (DST)* described in Sec. 3.2.2 transforms $\bar{F}_c$ to the feature map F_{cs} adaptively using the sum weight of the sampled points $w_\mathbf{r}$ along the ray $\mathbf{r}$ and the style information $T, \mu(F_s)$, and $\sigma(F_s)$. Finally, a stylized novel view is generated via a CNN decoder.

3.1. Feature Grid 3D Representation

To model a 3D scene with deep features, we use a continuous volumetric field of density and radiance. Different from the original NeRF [36], for every queried 3D position $x \in \mathbb{R}^3$, we get a volume density $\sigma(x)$ and a multi-channel feature $F(x) \in \mathbb{R}^C$ instead of an RGB color, where C is the number of the feature channels. Then we can get the feature of any rays $\mathbf{r}$ passing through the volume by integrating sampled points along the ray via approximated volume rendering [36]:

$$F(\mathbf{r}) = \sum_{i=1}^{N} w_i F_i, \tag{1}$$

$$\text{where} \quad w_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)\left(1 - \exp\left(-\sigma_i \delta_i\right)\right), \tag{2}$$

where σ_i, F_i denotes the volume density and feature of sampled point i, w_i denotes the weight of F_i in the ray $\mathbf{r}$,and δ_i is the distance between adjacent samples. We disable view-dependency effect for better multi-view consistency.

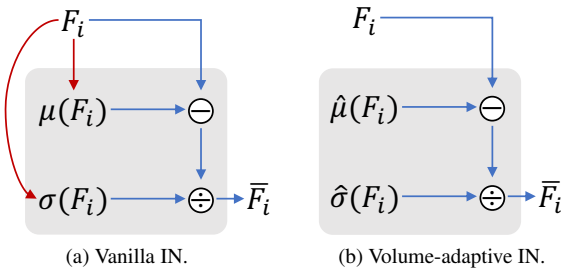
Then we can generate feature maps capturing high-level features and map them to RGB space using a 2D CNN decoder. However, unlike [3, 16, 40], we render full-resolution feature maps which have the same resolution as the final RGB images rather than down-sampled feature maps. Rendering full-resolution feature maps has two unique features: **1)** it discards up-sampling operations which cause multi-view inconsistency in general [16], **2)** it removes aliasing when rendering low-resolution feature maps [2] which causes severe flickering effects in stylized RGB videos.

Directly using 3D voxel grid to store features is memory-intensive. We thus adopt vector-matrix tensor decomposition [4] that relaxes the low-rank constraints for two modes of a 3D tensor and factorizes tensors into compact vector and matrix factors, which lowers the space complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, massively reducing the memory footprint. We employ a density grid to store volume density and a feature grid to store multi-channel features respectively.

3.2. Feature Transformation for Style Transfer

Once we have the feature grid representation of a scene, we can tackle the task of stylizing 3D scenes. Given a reference style image, our goal is to render stylized novel views of the 3D scene with multi-view consistency. To achieve this, we apply transformations to the features of the grid.

One plausible solution to this task is to apply style transfer to the feature grid directly. This solution is efficient in evaluations as it can render any stylized views with a single style transfer process only. However, it is impractical to train such transformation as it needs to stylize the whole feature grid in every iteration. Another solution is to apply an off-the-shelf zero-shot style transfer method to the features of the sampled 3D points. While this solution can reduce computational cost through decreasing the size of training patch and the number of sampled points, it has two problems: **1)** vanilla zero-shot style transformation is conditioned on holistic statistics of the sampled point batch [21, 28, 32], which violates multi-view consistency in volume rendering as the feature transformation of a specific 3D point will vary across different sampled points; **2)** volume rendering requires sampling hundreds of points along a single ray, which makes transformation on the point batch memory-intensive.



(a) Vanilla IN. (b) Volume-adaptive IN.

Figure 3. Comparison between vanilla instance normalization (IN) in (a) and volume-adaptive IN in (b). During evaluation, volume-adaptive IN uses learned mean and standard-deviation, discarding dependency over the sampled point batch's holistic statistics (indicated by the red arrows in the left graph).

Motivated by the observation that style transformation is conditioned on both content information and style information, we decompose the style transformation into sampling-invariant content transformation (SICT) and deferred style transformation (DST). After the decomposition, SICT will be conditioned solely on the content information while DST conditioned solely on the style information, more details to be elaborated in the ensuing subsections.

3.2.1 Sampling-invariant Content Transformation

Given a batch of sampled points, we can get their corresponding features $F_i \in \mathbb{R}^C, i \in [1, 2, ..., N]$ from the feature grid, where N is the number of the sampled points along a ray and C is the number of the feature channels. The goal of SICT is to transform the extracted features F_i so that they can be better stylized. We formulate SICT as a channel-wise self-attention operation to the features after instance normalization (IN) [54]. Specifically, we formulate Q(query), K(key), and V(value) as:

$$Q = q(Norm(F_i)), \tag{3}$$

$$K = k(Norm(F_i)), \tag{4}$$

$$V = v(Norm(F_i)), \tag{5}$$

where q, k, v are 1×1 convolution layers which reduce the channel number from C to C' for computational efficiency, and $Norm$ denotes the IN. However, as shown in Fig. 3,

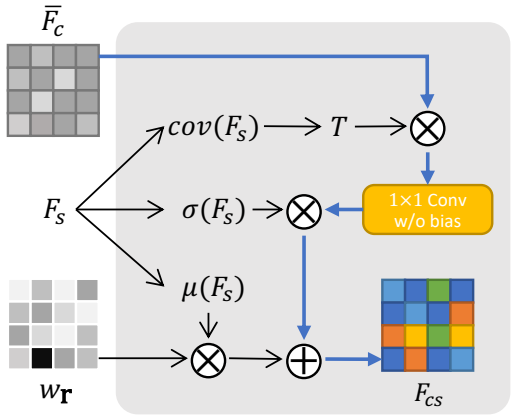


Figure 4. **Deferred style transformation.** We apply the style transformation to the volume-rendered feature maps $\bar{F}_c$ according to the style feature maps F_s. To ensure multi-view consistency, we modulate the bias (e.g. the mean value of the style feature maps $\mu(F_s)$) with the sum weight of sampled points along each ray $w_\mathbf{r}$.

vanilla IN calculates per-dimension mean and standard-deviation of the batch of sampled points, which varies with different sampled points and incurs multi-view inconsistency accordingly. Thus we design volume-adaptive IN which, during training, keeps running estimates of the computed mean and standard-deviation, and uses them for normalization during evaluations (instead of computing from the sampled point batch). Through volume-adaptive IN, we can ensure that the content transformation is consistent regardless of the sampled point batch's holistic statistics.

Channel-wise self-attention can thus be implemented by:

$$\bar{F}_i = V \otimes \text{Softmax}\left(\widetilde{cov}(Q, K)\right), \tag{6}$$

where $\otimes$ denotes matrix multiplication and $\widetilde{cov}(Q, K) \in \mathbb{R}^{N \times C' \times C'}$ denotes the covariance matrix in the channel dimension.

3.2.2 Deferred Style Transformation

After applying SICT to the features of each 3D point, we apply DST to the volume-rendered 2D feature maps $\bar{F}_c$ rather than 3D point features $\bar{F}_i$. To ensure multi-view consistency, we formulate the transformation as matrix multiplication and adaptive bias addition as illustrated in Fig. 4.

Specifically, we first extract feature maps F_s of the reference style S using a pre-trained VGG [51], and then generate the style transformation matrix $T \in \mathbb{R}^{C' \times C'}$ using feature covariance $cov(F_s)$ following [28]. Next, we apply matrix multiplication with T to the feature maps $\bar{F}_c$ and use a 1×1 convolution layer $conv$ without bias to restore the channel number from C' to C. Though these operations can partially instill style information, they are not expressive enough without bias addition containing style information [58]. Thus following [21], we multiply the feature maps with the standard-deviation value $\sigma(F_s)$ and add the mean value $\mu(F_s)$. To ensure it is equivalent when applying the transformation to either 3D point features or 2D feature maps, we adaptively modulate the mean value $\mu(F_s)$ with the sum weight of sampled points along each ray $w_\mathbf{r}$. DST can be mathematically formulated by:

$$F_{cs} = conv\left(T \otimes \bar{F}_c\right) \times \sigma(F_s) + w_\mathbf{r} \times \mu(F_s), \tag{7}$$

$$\text{where} \quad \bar{F}_c = \sum_{i=1}^{N} w_i \bar{F}_i, w_\mathbf{r} = \sum_{i=1}^{N} w_i, \mathbf{r} \in \mathcal{R} \tag{8}$$

where w_i denotes the weight of sampled point i (Eq. (2)), $\bar{F}_i$ denotes the feature of sample i after SICT, and $\mathcal{R}$ is the set of rays in each training batch.

Note $conv$ is a 1×1 convolution layer without bias, so it is basically a matrix multiplication operation. And $\sigma(S), \mu(S)$ are scalars. Together with the adaptive bias modulation $w_\mathbf{r}$, Eq. (7) can be reformulated by:

$$F_{cs} = \sum_{i=1}^{N} w_i \left(\underbrace{conv\left(T \otimes \bar{F}_i\right) \times \sigma(F_s) + \mu(F_s)}_{\text{(i)}} \right), \tag{9}$$

where part (i) can be seen as applying style transformation on every 3D point feature independently before volume rendering. This proves that applying DST on 2D feature maps is equivalent to applying the transformation on 3D points' features, maintaining multi-view consistency. The full derivation of Eq. (9) is provided in the appendix.

Finally, we adopt a 2D CNN decoder to project the stylized feature maps F_{cs} to RGB space to generate the final stylized novel view images.

3.3. Two-stage Model Training

The training of our model is divided into the *feature grid training stage* and the *stylization training stage*, the former is trained with the target of novel view synthesis, and the latter is trained with the target of style transfer.

Feature grid training stage (First stage). We first learn the feature grid 3D representation for the novel view synthesis task, in preparation for performing feature transformation for style transfer. We train the feature grid and the 2D CNN decoder simultaneously, with the supervision of both RGB images and their bilinearly up-sampled feature maps extracted from `ReLU3_1` layer of pre-trained VGG [51]. By aligning the VGG features with the feature grid, the reconstructed features acquire semantic information. We use density grid pre-trained solely on RGB images since the supervising feature maps are not strictly multi-view consistent. The training objective is the mean square error (MSE) between the predicted and ground truth feature maps and RGB images. Following [19, 37], we use perceptual

loss [24] as additional supervision to increase reconstructed image quality. The overall loss function is:

$$\mathcal{L}_{grid} = \sum_{r\in\mathcal{R}} \left\| \hat{F}(\mathbf{r}) - F(\mathbf{r}) \right\|_2^2 + \left\| \hat{I}_{\mathcal{R}} - I_{\mathcal{R}} \right\|_2^2 + \sum_{l\in l_p} \left\| \mathcal{F}^l(\hat{I}_{\mathcal{R}}) - \mathcal{F}^l(I_{\mathcal{R}}) \right\|_2^2, \quad (10)$$

where $\mathcal{R}$ is the set of rays in each training batch, $\hat{F}(\mathbf{r}), F(\mathbf{r})$ are the predicted and ground truth feature of ray $\mathbf{r}$, $\hat{I}_{\mathcal{R}}, I_{\mathcal{R}}$ are the predicted and ground truth RGB image, l_p denotes the set of VGG layers that compute perceptual loss, $\mathcal{F}^l$ denotes the feature maps of the lth layer of pre-trained VGG network.

Stylization training stage (Second stage). Our model learns to stylize novel views in the second stage. We freeze the feature grid, train the style transfer module, and fine-tune the CNN decoder. Thanks to the memory-efficient representation of 3D scenes and DST, unlike [8, 11, 48], our model can be trained directly on 256×256 patches, making patch sub-sampling algorithm [8, 11, 22, 48] unnecessary. We use the same loss as [21] where the content loss $\mathcal{L}_c$ is the MSE of the feature maps and the style loss $\mathcal{L}_s$ is the MSE of the channel-wise feature mean and standard-deviation:

$$\mathcal{L}_{stylization} = \mathcal{L}_c + \lambda \mathcal{L}_s, \quad (11)$$

where λ balances the content preservation and the stylization effect.

4. Experiments

We evaluate StyleRF extensively with qualitative experiments in Sec. 4.1, quantitative experiments in Sec. 4.2 and ablation studies in Sec. 4.3. We demonstrate two applications of StyleRF in Sec. 4.4. The implementation details are provided in the appendix.

4.1. Qualitative Experiments

We evaluate StyleRF over two public datasets including LLFF [35] that contains real scenes with complex geometry structures and Synthetic NeRF [36] that contains 360° views of objects. In addition, we benchmark StyleRF with two state-of-the-art zero-shot 3D style transfer methods LSNV [19] and Hyper [8] with their released codes. We perform comparisons on LLFF dataset [35].

Fig. 5 shows qualitative comparisons. We can see that StyleRF achieves clearly better stylization with more precise geometry reconstruction. Specifically, StyleRF can generate high-definition stylization with realistic textures and patterns of style images. The superior stylization is largely attributed to our transformation design that allows working in the feature space with full-resolution feature maps. As illustrated in the highlight boxes, StyleRF can successfully restore the intricate geometry of complex scenes thanks to its radiance field representations. In addition, only StyleRF faithfully transfers the squareness texture in the second style image. Furthermore, StyleRF can robustly generalize to new styles in a zero-shot manner and can adapt well to 360° dataset as illustrated in Fig. 1. As a comparison, LSNV [19] fails to capture fine-level geometry like the bones of the T-Rex and the petals of the flower while Hyper [8] produces very blurry stylization.

Method	Short-range Consistency		Long-range Consistency	
	LPIPS	**RMSE**	**LPIPS**	**RMSE**
AdaIN [21]	0.152	0.123	0.220	0.186
CCPL [60]	0.110	0.106	0.191	0.174
ReReVST [57]	0.098	0.080	0.186	0.146
LSNV [19]	0.093	0.092	0.181	0.155
Hyper [8]	0.084	0.068	0.131	0.101
Ours	0.072	0.082	0.149	0.137

Table 1. **Results on consistency.** We compare StyleRF with the state-of-the-art on consistency using LPIPS (↓) and RMSE (↓).

4.2. Quantitative Results

3D style transfer is a very new and under-explored task and there are few metrics for quantitative evaluation of stylization quality. Hence, we manage to evaluate the multi-view consistency only. In our experiments, we warp one view to the other according to the optical flow [53] using softmax splatting [42], and then computed the masked RMSE score and LPIPS score [64] to measure the stylization consistency. Following [8, 11, 19], we compute the short-range and long-range consistency scores which compare adjacent views and far-away views respectively. We compare StyleRF against two state-of-the-art zero-shot 3D style transfer methods Hyper [8] and LSNV [19], one SOTA single-frame-based video style transfer method CCPL [60], one SOTA multi-frames-based video style transfer method ReReVST [57], and one classical image style transfer method AdaIN [21].

It can be seen from Tab. 1 that StyleRF significantly outperforms image style transfer approach [21] and video style transfer approach [57, 60] which capture little information about the underlying 3D geometry. In addition, StyleRF achieves better consistency than point-cloud-based 3D style transfer [19] as well. Note Hyper [8] achieves slightly better LPIPS and RMSE scores than our method, largely because it produces over-smooth results and inadequate stylization as shown in Fig. 5.

4.3. Ablation Studies

We design two innovative techniques to improve the stylization quality and maintain the multi-view consistency.

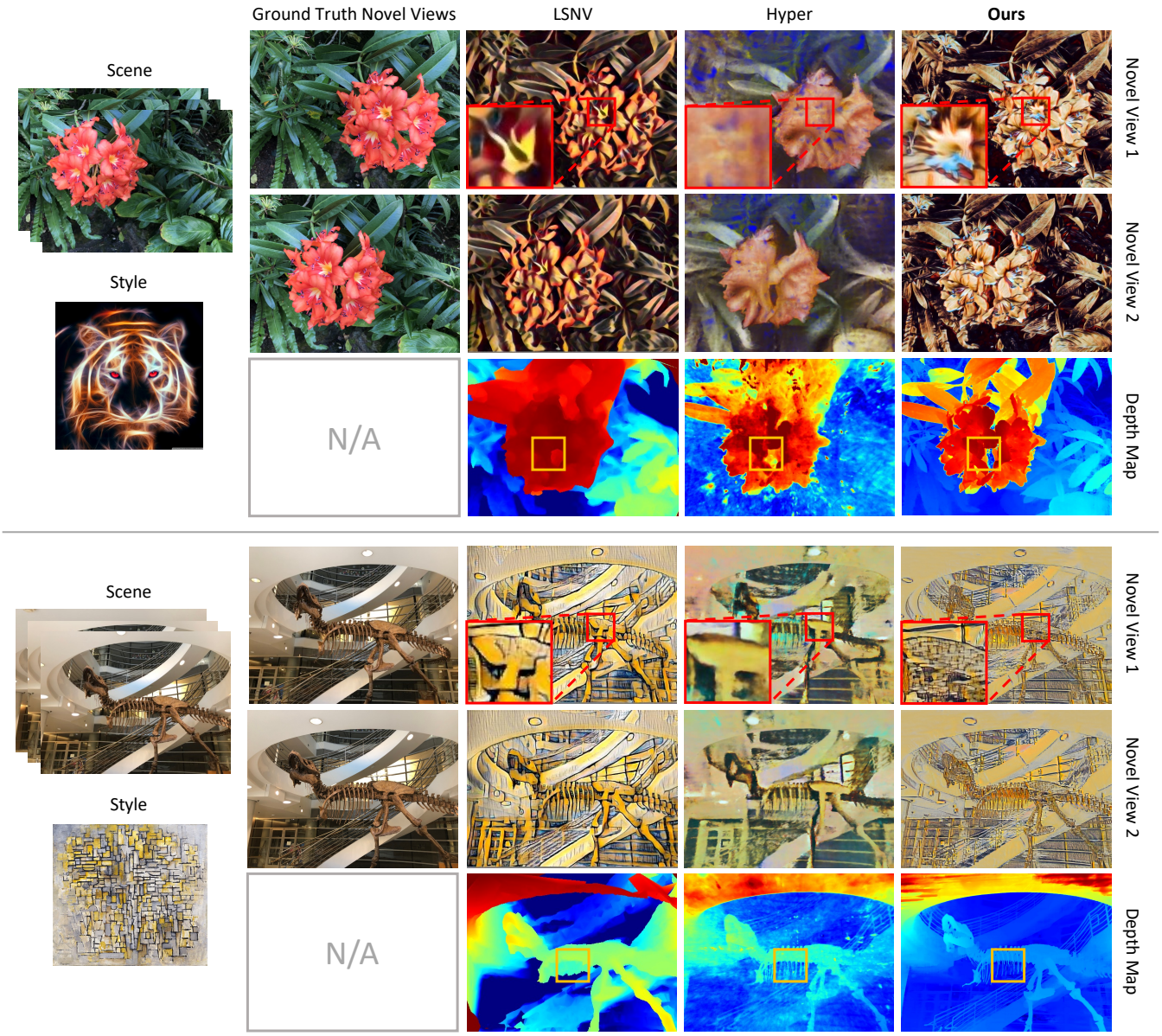


Figure 5. Comparison of StyleRF with two state-of-the-art zero-shot 3D style transfer methods LSNV [19] and Hyper [8]. For each of the two sample *Scenes* and reference *Styles*, StyleRF produces clearly better 3D style transfer and depth estimation. Check zoom-in for details.

The first is volume-adaptive instance normalization which uses the learned mean and variance of the whole volume during inference, eliminating the dependency on holistic statistics of the sampled point batch. The second is the adaptive bias addition in DST, which improves the stylization quality using bias capturing style information. We evaluate the two designs to examine how they contribute to the overall stylization of our method.

Volume-adaptive instance normalization. We compare our volume-adaptive instance normalization (IN) with vanilla IN and StyleRF without IN. As Fig. 6 (c) shows, vanilla IN produces severe block-shape artifacts as the transformation of each batch is conditioned on the holistic statistics of itself, thus each batch (i.e. block in the image) produces inconsistent stylization which leads to the artifacts. However, if we discard IN as shown in Fig. 6 (d), the multi-view consistency can maintain but the stylization quality compromises a lot, failing to capture the correct color tone of the reference style image. This is because IN removes the original style information of the content image which facilitates the transfer of the reference style [21].

Adaptive bias addition. As illustrated in Fig. 6 (b), the stylization quality degrades a lot if we eliminate the adaptive bias addition in DST (Sec. 3.2.2), producing unnatural

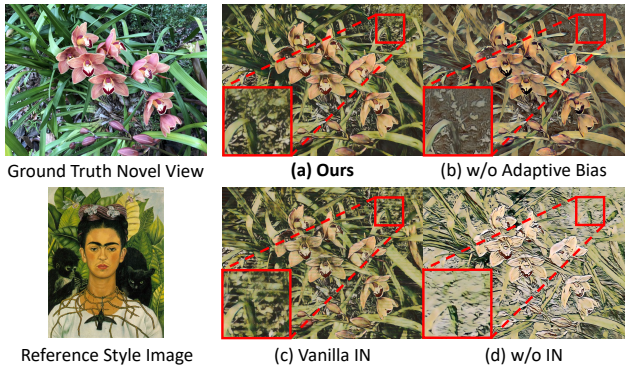


Figure 6. **Ablation studies.** (a) shows the stylization of our full pipeline. (b) shows the stylization without the adaptive bias. (c) shows the stylization when replacing the volume-adaptive instance normalization (IN) with vanilla IN. (d) shows the stylization without any IN.

Figure 7. **Multi-style interpolation.** StyleRF can smoothly interpolate between arbitrary styles by interpolating features of the scene.

stylization compared to the stylization of our full pipeline in Fig. 6 (a). This is because bias usually contains crucial style information such as the overall color tone [58]. StyleRF employs bias addition that is adaptively modulated by the weight of each ray, improving the stylization quality and keeping multi-view consistency concurrently.

4.4. Applications

StyleRF can be easily extended along different directions with different applications. We provide two possible extensions in the ensuing subsections.

Multi-style interpolation. StyleRF can smoothly interpolate different styles thanks to its high-level feature representation of a 3D scene. As illustrated in Fig. 7, we linearly interpolate the feature maps of a specific view by using four different styles at four corners. Unlike previous NeRF-based 3D style transfer that supports style interpolation by interpolating one-hot latent vectors [11], StyleRF can interpolate arbitrary numbers of unseen new styles by interpolating features of the scene, yielding more smooth and harmonious interpolation. Hence, StyleRF can not only transfer arbitrary styles in a zero-shot manner but also generate non-existent stylization via multi-style interpolation.

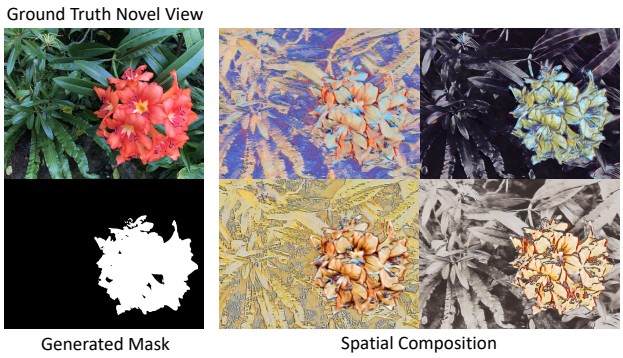


Figure 8. **Compositional 3D style transfer.** Given the 3D-consistent segmentation masks, StyleRF can create infinite combinations of styles by spatial composition.

Compositional 3D style transfer. Thanks to its precise geometry reconstruction, StyleRF can be seamlessly integrated with NeRF-based object segmentation [12, 26, 65] for compositional 3D style transfer. As shown in Fig. 8, we apply 3D-consistent segmentation masks to the feature maps and apply different styles to stylize the contents inside and outside the masks separately. We can see that the edges of the masks can be blended more softly by applying the segmentation masks to the feature maps instead of RGB images. Due to its zero-shot nature, StyleRF can create infinite combinations of styles without additional training, producing numerous artistic creations and inspirations.

5. Conclusion

In this paper, we present StyleRF, a novel zero-shot 3D style transfer method that balances the three-way dilemma over accurate geometry reconstruction, high-quality stylization, and being generalizable to arbitrary new styles. By representing the 3D scene with an explicit grid of high-level features, we can faithfully restore high-fidelity geometry through volume rendering. Then we perform style transfer on the feature space of the scene, leading to high-quality zero-shot stylization results. We innovatively design sampling-invariant content transformation to maintain multiview consistency and deferred style transformation to increase efficiency. We demonstrate that StyleRF achieves superior 3D stylization quality than previous zero-shot 3D style transfer methods, and can be extended to various interesting applications for artistic 3D creations.

Acknowledgement

This project is funded by the Ministry of Education Singapore, under the Tier-1 project scheme with project number RT18/22.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 4

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2, 3, 4

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 2, 3, 4

[5] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017. 3

[6] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu. Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene. In *arxiv*, 2022. 3

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[8] 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, 6, 7

[9] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2719–2727, 2020. 3

[10] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14304–14313, 2021. 2

[11] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. *arXiv preprint arXiv:2204.01943*, 2022. 2, 3, 6, 8

[12] Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Nerf-sos: Any-view self-supervised object segmentation on complex scenes. *arXiv preprint arXiv:2209.08776*, 2022. 8

[13] Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. *arXiv preprint arXiv:2301.10241*, 2023. 2

[14] 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

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2, 3

[16] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 3, 4

[17] Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6198–6208, 2022. 2

[18] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017. 3

[19] 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, 3, 5, 6, 7

[20] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[21] 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, 3, 4, 5, 6, 7

[22] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022. 2, 3, 6

[23] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017. 2

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2, 3, 6

[25] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[26] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 8

[27] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 2

[28] 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. 3, 4, 5

[29] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017. 3

[30] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 2

[31] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[32] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6649–6658, 2021. 3, 4

[33] Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788*, 2021. 2

[34] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[35] 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. 6

[36] 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, 3, 4, 6

[37] 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, 3, 5

[38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 2

[39] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *arXiv preprint arXiv:2207.02363*, 2022. 2, 3

[40] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 3, 4

[41] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 2

[42] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 6

[43] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5880–5888, 2019. 3

[44] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 2

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[46] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2

[47] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, 2018. 3

[48] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 6

[49] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. 2

[50] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8242–8250, 2018. 3

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[52] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2

[53] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 6

[54] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[55] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[56] Wenjing Wang, Jizheng Xu, Li Zhang, Yue Wang, and Jiaying Liu. Consistent video style transfer via compound regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12233–12240, 2020. 3

[57] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. Consistent video style transfer via relaxation and regularization. *IEEE Trans. Image Process.*, 2020. 3, 6

[58] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14618–14627, 2021. 3, 5, 8

[59] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[60] 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. 6

[61] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2

[62] Fangneng Zhan, Lingjie Liu, Adam Kortylewski, and Christian Theobalt. General neural gauge fields. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[63] 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. 2, 3

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[65] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 8