Multi-StyleGS: Stylizing Gaussian Splatting with Multiple Styles

Yangkai Lin[1], **Jiabao Lei**[2], **Kui Jia**[2*]

[1]South China University of Technology
[2]School of Data Science, The Chinese University of Hong Kong, Shenzhen
202210182091@mail.scut.edu.cn, jiabaolei@link.cuhk.edu.cn, kuijia@cuhk.edu.cn

Abstract

In recent years, there has been a growing demand to stylize a given 3D scene to align with the artistic style of reference images for creative purposes. While 3D Gaussian Splatting (GS) has emerged as a promising and efficient method for realistic 3D scene modeling, there remains a challenge in adapting it to stylize 3D GS to match with multiple styles through automatic local style transfer or manual designation, while maintaining memory efficiency for stylization training. In this paper, we introduce a novel 3D GS stylization solution termed Multi-StyleGS to tackle these challenges. In particular, we employ a bipartite matching mechanism to automatically identify correspondences between the style images and the local regions of the rendered images. To facilitate local style transfer, we introduce a novel semantic style loss function that employs a segmentation network to apply distinct styles to various objects of the scene and propose a local-global feature matching to enhance the multi-view consistency. Furthermore, this technique can achieve memory-efficient training, more texture details and better color match. To better assign a robust semantic label to each Gaussian, we propose several techniques to regularize the segmentation network. As demonstrated by our comprehensive experiments, our approach outperforms existing ones in producing plausible stylization results and offering flexible editing.

Introduction

Artistic creation has attracted considerable attention, with aesthetic 3D content creation being one of the urgent demands in recent years. Stylizing an already acquired 3D scene is the primary approach to obtain artistic 3D content. In this paper, our focus is on the task of 3D scene stylization, where we aim to transfer reference styles specified by multiple style images to the 3D scene.

Previous work on 3D stylization (Huang et al. 2022; Wang et al. 2023; Pang, Hua, and Yeung 2023; Jung et al. 2024) has predominantly utilized the Neural Radiance Field (NeRF) as a scene representation (Mildenhall et al. 2020). While NeRF is compact and capable of achieving photo-realistic rendering results, it is limited to implicit editing and faces significant performance challenges due to the utilization of a heavy and high-dimensional Multi-Layer Perceptron (MLP) network for scene representation. Balancing computational time and result quality requires a delicate trade-off. Despite some advancements (Müller et al. 2022; Reiser et al. 2021; Sun, Sun, and Chen 2022; Fridovich-Keil et al. 2022; Chen et al. 2022; Hu et al. 2023; Barron et al. 2023) aimed at mitigating the performance issues of NeRF in practical applications, these challenges still persist. Recently, a significant portion of the work on 3D stylization has concentrated on global stylization (Zhang et al. 2022; Nguyen-Phuoc, Liu, and Xiao 2022; Liu et al. 2023; Chiang et al. 2022a), where the same style pattern is applied uniformly to all parts of the 3D content. However, this approach can be suboptimal as not all regions should be treated equally, limiting flexibility and editability. Another portion of the work focus on the local stylization (Miao et al. 2024; Zhang et al. 2023). However, they can only stylize simple scenes (Mildenhall et al. 2019) and struggle to ensure multi-view consistency. Furthermore, techniques that employ Gaussian Splatting (GS) (Kerbl et al. 2023) frequently encounter memory bottleneck issues, impeding the progress for further applications.

To address these challenges, we introduce a novel 3D stylization solution called Multi-StyleGS. This method is designed to deliver flexible and efficient image-based stylization of 3D scenes by empowering explicit local editing.

Specifically, we choose GS (Kerbl et al. 2023) as our base representation, for its real-time rendering performance and explicit characteristic. While promising, we have noted a substantial increase in memory usage and the emergence of multi-view inconsistency in feature matching. To address these challenges, we introduce a novel semantic style loss to mitigate the problem of excessive memory consumption and multi-view inconsistency. Furthermore, to enable local stylization on semantic regions, we introduce an additional semantic feature for each GS, and update them during optimization. This enhancement facilitates automatic local style transfer for region correspondences between multiple style images and the 3D scene. Technically, we perform local style transfer as an additional post-processing step after capturing the original geometry and appearance of the 3D scenes using GS. During the stylization process, we solely optimize the appearance of Gaussians. In addition to reconstruct 3D scene, we introduce an extra segmentation attribute that divides the Gaussians of the scene

*Corresponding author.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: With a set of multi-view images of a 3D scene and several specified style images, our method can transfer artistic styles to the 3D scene, creating high-quality stylized images of novel views with consistency.

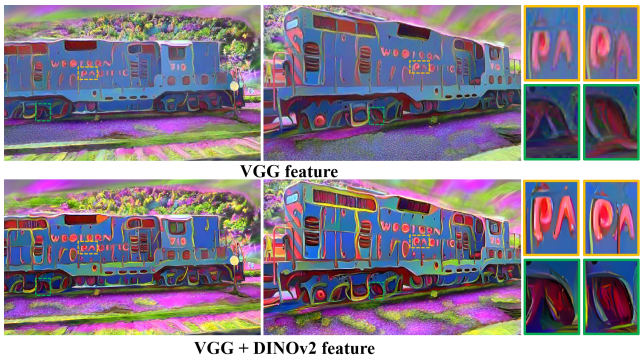

Figure 2: Comparing the stylized results, from various perspectives, the same object may correspond to different VGG features, averaging out the details (as seen in subfigures with orange borders) or displaying varying colors (as seen in subfigures with green borders). DINOv2 enhances the global consistency that VGG features lack, ensuring consistent guidance across different viewpoints.

into multiple parts and paired with multiple style images through an effective bipartite matching mechanism (Luan et al. 2017) to automatically establish local region correspondences based on their feature similarity or manual designation. Subsequently, a novel multi-style loss is applied to guarantee local editability. Additionaly, we observed a multi-view inconsistency issue. Inspired by (El Banani et al. 2024), we utilize DINOv2(Oquab et al. 2023) to extract global features and introduce local-global matching for enhanced multi-view consistency with our novel multi-style loss, improving consistency, texture details, and color accuracy. To address the potential issue of segmentation error due to the high degree of freedom of Gaussians, we introduce Gaussian smoothing regularization to alleviate this problem. Additionally, in order to mitigate the semantic ambiguity problem, we develop a technique called semantic importance filtering, which leverages semantic labels to effectively eliminate those Gaussians exhibiting semantic uncertainty; a negative entropy regularization term is also applied to each Gaussian to enforce semantic clarity. Our segmentation approach leverages SAM's (Kirillov et al. 2023) capability, which we apply to the 3D scene to enhance multi-view consistency.

Our solution is able to handle styles from one single image or multiple images. Extensive experiments conducted on various datasets (Knapitsch et al. 2017; Mildenhall et al. 2019) substantiate the efficacy of our method in generating high-quality, locally matched stylized images in real-time. To summarize, our main contributions are:

- A novel GS-based approach for local stylization of 3D scenes, facilitating the transfer of multiple artistic styles from one or several 2D images to 3D scenes.
- A new style loss that adopts bipartite matching assignment between multiple style image regions and GS points to enable (automatic) local style transfer.
- A local-global feature matching solution to improve multi-view consistency.
- Several regularization terms for removing noisy Gaussians and accuracy segmentation.

Related Works

Overview of 3D Style Transfer

Conventional approaches to stylize 3D scenes use explicit representations like point clouds (Huang et al. 2021; Mu et al. 2022) or meshes (Mu et al. 2022; Michel et al. 2022; Kato, Ushiku, and Harada 2018; Höllein, Johnson, and Nießner 2022). These approaches, however, is error-prone and may fail to capture geometry and texture details. NeRF (Mildenhall et al. 2020) encodes a 3D scene using a neural network, making it a more suitable representation for downstream stylization tasks compared to explicit ones. A common approach to stylizing a NeRF is to optimize and constrain its rendered images to a specific style using content loss and style loss. snerf, arf, ins and CoArf (Nguyen-Phuoc, Liu, and Xiao 2022; Zhang et al. 2022; Fan et al. 2022; Zhang, Fernandez-Labrador, and Schroers 2024) follow this line and optimizes neural networks using style loss. snerf renders blurry results due to refine geometry without supervision, and arf fixs the geometry branch and proposes nearest-neighbor feature matching loss to capture details. ins

decouples NeRF to allow for separately encoding of representations.

Their results do not support diverse stylization results and typically stylize only the foreground of the scenes. HyperNet (Chiang et al. 2022b) uses a hypernetwork to predict the weights of MLP to speed up stylization. LsNeRF (Pang, Hua, and Yeung 2023) introduces a region-matching style loss designed to enhance local stylization of the 3D scenes. Yet, this method faces limitations as it cannot concurrently assimilate styles from multiple images into a single 3D scene. Moreover, it is unable to maintain multi-view stylistic consistency. Our work introduces a matching mechanism (Pang, Hua, and Yeung 2023) to establish region correspondences and a novel style loss to support local style transfer. Besides, thanks to the use of explicit representation of GS (Kerbl et al. 2023), we can nicely stylize the background as well.

Memory-efficient 3D Style Transfer

However, such methods are very memory-inefficient in practice. ARF (Zhang et al. 2022) propose a deferred back-propagation method to enable optimization of memory-intensive NeRF. StyleRF (Liu et al. 2023) proposes a deferred style transformation of 2D feature maps to greatly reduces memory footprint. These approaches are all developed based on NeRF (Mildenhall et al. 2020) and are not applicable to GS (Kerbl et al. 2023). Our novel semantic style loss can achieve memory-efficient training, which enables efficient training on a single RTX 3090.

3D Local Stylization

Another line of work (Pang, Hua, and Yeung 2023; Zhang et al. 2023; Miao et al. 2024) investigates local stylization methods, which allow for diverse styles on local regions. However, most of the work can only stylize relatively simple scenes and cannot ensure multi-view consistency. Our method proposes a local-global matching to tackle this issue and conducts extensive experiments on various datasets(Mildenhall et al. 2019; Knapitsch et al. 2017).

Preliminary of Gaussian Splatting

Gaussian Splatting (GS) (Kerbl et al. 2023) represents a 3D scene with a set of 3D Gaussians. Each Gaussian consists of a center location $\boldsymbol{\mu} \in \mathbb{R}^3$, a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$. The covariance matrix $\mathbf{\Sigma}$ can be decomposed into a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a diagonal scaling matrix $\mathbf{S} \in \mathbb{R}^{3\times3}$ as shown by

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T. \tag{1}$$

To render novel views, splatting is utilized to project 3D Gaussians onto 2D canvas. This technique involves a viewing transformation denoted by $\mathbf{W} \in \mathbb{R}^{3\times3}$ and Jacobian $\mathbf{J} \in \mathbb{R}^{2\times3}$ of the affine approximation of the projective transformation. The 2D covariance matrix $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{2\times2}$ can then be given as

$$\hat{\mathbf{\Sigma}} = \mathbf{J}\mathbf{W}\mathbf{\Sigma}\mathbf{W}^T\mathbf{J}^T. \tag{2}$$

We finally leverage α-blending of N overlapped Gaussians at a pixel to accumulate color by

$$\mathbf{c} = \sum_{i=1}^{N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{3}$$

where $\mathbf{c} \in \mathbb{R}^3$ is the rendered pixel color, $\mathbf{c}_i \in \mathbb{R}^3$ and $\alpha_i \in \mathbb{R}$ are color and density of the i-th Gaussian point, respectively.

Our Method

Multi-StyleGS consists of two stages: the reconstruction stage, where a base GS model is trained to recover the original scene and additionally learn semantic correspondences, and the stylization stage, where the GS model is further refined to adjust its appearance to multiple styles specified by the correspondences.

Gaussian Splatting with Semantic Features

To establish local region correspondences for local style transfer, we leverage segmentation maps and match local regions in 3D scenes with those in style images. In particular, we enhance GS by incorporating an extra segmentation branch, as illustrated in Figure 3. In addition to the existing attributes of the Gaussians (*e.g.*, color and opacity), we introduce a new trainable feature $\mathbf{e}_i$ for each Gaussian (Ye et al. 2023; Zhou et al. 2024). This feature $\mathbf{e}_i$ is subsequently decoded by a tiny MLP to predict a semantic category.

To optimize the feature $\mathbf{e}_i$ and the tiny MLP, we render these semantic features into 2D images in a differentiable manner. Specifically, we have the following formula for feature integration:

$$\mathbf{e} = \sum_{i=1}^{N} \mathbf{e}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{4}$$

where $\mathbf{e} \in E$ is the rendered feature and E denotes the rendered feature map. The feature $\mathbf{e}$ subsequently passes through a softmax function to calculate the cross-entropy loss $\mathcal{L}^{\text{seg}}$. However, to compute $\mathcal{L}^{\text{seg}}$, we need ground-truth semantic labels. We use SAM (Kirillov et al. 2023) to automatically generate semantic labels for each 2D image and employ a well-trained zero-shot tracker (Cheng et al. 2023) to propagate and associate semantic labels (Ye et al. 2023). Back-propagation is employed to optimize the feature $\mathbf{e}_i$ and parameters of the tiny MLP.

However, we observe that updating each Gaussian point individually can lead to noisy and unstable outcomes due to the stochastic optimization nature and the restricted granularity of points. To address the issues, we leverage a locality assumption: neighboring points should exhibit similar characteristics. We introduce a regularization loss to enhance the smoothness of segmentation results based on k-nearest neighbor (KNN) by

$$\mathcal{L}^{\text{KNN}} = \sum_{i} \sum_{j\in\mathcal{N}_i} \|\mathbf{e}_i - \mathbf{e}_j\|_2^2 \, e^{-\frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}{\sigma}}, \tag{5}$$

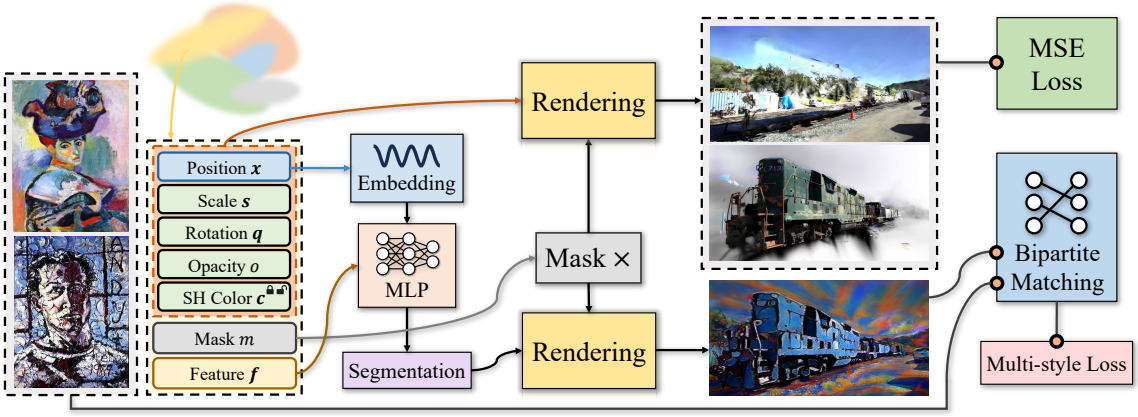

Figure 3: Overview of our pipeline. It firstly reconstructs a GS model from multiple training images, and then stylize the scene using bipartite matching with multiple styles. Upon completion, it can produce consistent free-viewpoint stylized renderings.

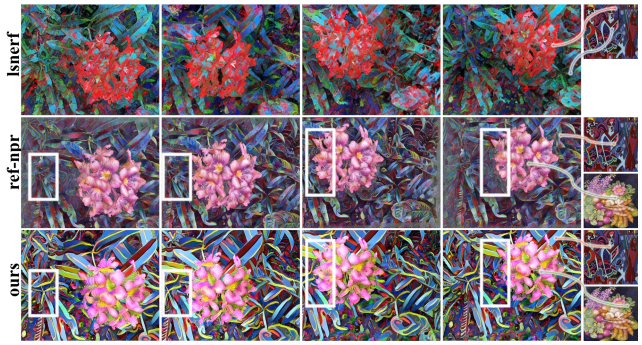

Figure 4: Qualitative comparisions with ref-npr (Zhang et al. 2023), lsnerf (Pang, Hua, and Yeung 2023). The results enclosed in the white boxes are blurred due to the multi-view inconsistency in the ref-npr. In contrast, our model has effectively ensured consistency.

where $\mathcal{N}_i$ gathers k nearest neighbors for the i-th Gaussian, and $\sigma \in \mathbb{R}_{>0}$ determines the influence radius. The essence of $\mathcal{L}^{\text{KNN}}$ lies in weighting the similarity differences according to the influence of their distances. Our assumption encourages local smoothness, avoids excessive randomness, and increases the granularity of influence.

Moreover, we notice that one Gaussian point may be responsible for multiple objects, leading to semantic ambiguity which is unwanted. We incorporate a negative entropy regularization term:

$$\mathcal{L}_{\text{NE}} = -\sum_{i=1}^{N} \text{softmax}(\mathbf{e}_i) \log(\text{softmax}(\mathbf{e}_i)), \tag{6}$$

to enforce each point to choose only one category, eliminating such an ambiguity.

However, some points may be of less semantic importance. We utilize a semantic importance filter to eliminate those with less semantic significance. Specifically, we additionally assign a learnable mask attribute $m \in \mathbb{R}$ to individual Gaussian (Lee et al. 2023) to assess its importance, and utilize semantic labels $\mathbf{e}_i$ to select Gaussians without semantic ambiguity. We also employ the straight-through estimator (Bengio, Léonard, and Courville 2013) for gradient propagation. We apply a mask $m^{\text{b}} \in \{0, 1\}$ to the scale vector $\mathbf{s} \in \mathbb{R}^3$ (diagonal elements of $\mathbf{S}$) and the opacity $o \in \mathbb{R}$ by $\hat{\mathbf{s}} = m^{\text{b}}\mathbf{s}$ and $\hat{o} = m^{\text{b}}o$, respectively, where the binary mask m^{b} can be obtained by

$$\begin{aligned} m^{\text{b}} &= \text{sg}\left[v^{\text{b}} - \sigma\left(m\right)\right] + \sigma(m), \\ \text{with} \quad v^{\text{b}} &= \mathbb{1}_{\sigma(m)>\epsilon_0} \vee \mathbb{1}_{\max(\text{softmax}(\mathbf{e}_i))>\epsilon_1}, \end{aligned} \tag{7}$$

where $\epsilon_0 \in \mathbb{R}$ and $\epsilon_1 \in \mathbb{R}$ are thresholds, "sg" is to stop gradients, σ is the sigmoid function, $\mathbb{1}_A$ is an indicator of event A, and $\vee$ is logical OR operator. During reconstruction, the GS model optimizes the scale, opacity, and mask attributes simultaneously. This approach enables a more holistic consideration of both scale and opacity when assessing the importance of Gaussian components. To promote the decimation of redundant Gaussians, we introduce a mask regularization term given by

$$\mathcal{L}^{\text{mask}} = \sum_{m} \sigma(m). \tag{8}$$

We note that by incorporating $\mathcal{L}^{\text{mask}}$, our model facilitates the automatic elimination of Gaussians through gradient control. By adjusting the weighting coefficient of $\mathcal{L}^{\text{mask}}$, we can achieve a more optimal balance between rendering quality and memory footprint. At specific iterations, we remove certain unnecessary Gaussians based on m^{b}.

Preliminary of Style Loss

Given a pair of rendered output image I and style image S, the style loss typically operates on high-level features $\mathbf{f}_I = F(I)$, $\mathbf{f}_S = F(S)$, where F is a pretrained VGG19 (Simonyan and Zisserman 2015a) network. For instance, StyleGaussian (Liu et al. 2024) employs AdaIN for stylistic transformation, while ARF (Zhang et al. 2022) introduces a nearest-neighbor feature matching (NNFM) loss to achieve style transfer. The NNFM loss introduced in (Zhang et al.

Figure 5: Qualitative comparisons with snerf(Nguyen et al. 2022), arf(Zhang et al. 2022) and ref-npr(Zhang et al. 2023) on tnt datasets in single style setting. **Black boxes (A-D): inconsistency may lead to blurry results**; **green boxes (E-H): previous solutions may produce incomplete stylized results**; **blue boxes (I-L): our method can preserve texture details**.

2022) uses the following formulation,

$$\mathcal{L}_{\text{NNFM}}^{\text{naive}} = \sum_{f_i \in \mathbf{f}_I} \min_{f_j \in \mathbf{f}_S} d(f_i, f_j), \tag{9}$$

where every individual feature vector $f_i \in \mathbf{f}_I$ is paired with the closest style feature $f_j \in \mathbf{f}_S$ according to cosine distance d. However, the style loss is to match the global statistics between the rendered output image I and style image S, and can not support diverse stylization results. We incorporate bipartite matching to augment NNFM loss to support local style transfer, which will be detailed in the next coming section.

However, VGG feature is 2D local and has no 3D awareness. Using VGG features only for matching can lead to multi-view inconsistency issues. Moreover, a substantial increase in memory usage is observed when utilizing GS as the base representation for stylization. To address these issues, we propose a novel semantic style loss.

Semantic Multi-style Loss

To facilitate local style transfer, we firstly establish region correspondences for the Gaussian point set $\{g_i\}_{i=1}^N$ of the scene and the set of input style images $\{S_i\}_{i=1}^M$. After reconstruction, the feature $\mathbf{e}_i$ of each Gaussian will indicate the semantic label to which the object it belongs, categorizing the Gaussians into C distinct classes. The initial step in our pipeline involves partitioning $\{g_i\}_{i=1}^N$ into multiple point set $\{G_i\}_{i=1}^C$ as show in Figure 3.

Local-global Feature Matching Since Gaussian points, when observed from various perspectives, may align with distinct style features, resulting in multi-view inconsistency, as illustrated in Figure 7. We found that features from VGG tend to suffer from such a problem stemming from poor global consistency.

One paper (El Banani et al. 2024) assessed the 3D awareness of visual models and posits that DINOv2 (Oquab et al. 2023) demonstrates superior 3D consistency. Therefore, we extract DINOv2 feature and VGG feature and concatenate them along the channel dimension, then perform nearest feature matching on concatenative feature as follows,

$$\mathbf{C}_S = \text{concat}(\mathbf{f}_S, \phi(S)), \mathbf{C}_I = \text{concat}(\mathbf{f}_I, \phi(I)), \tag{10}$$

$$\mathcal{L}_{\text{NNFM}} = \sum_{f_i \in \mathbf{C}_I} \min_{f_j \in \mathbf{C}_S} d(f_i, f_j), \tag{11}$$

where ϕ is the DINOv2 feature extractor, "concat" is to concatenate two feature maps along the channel dimension. VGG feature can provide local details and DINOv2 feature can provide global consistency. $\mathcal{L}_{\text{NNFM}}$ not only enhances multi-view consistency but also better improves matching results, ensuring that the same area, when viewed from another perspective, exhibits consistency and is endowed with richer and clearer details, as shown in Figure. 2.

Local Style Loss To prevent multiple scene regions from being stylized with the same local pattern, we incorporate a bipartite matching mechanism (Pang, Hua, and Yeung 2023) to automatically identify local region correspondences between multiple point set $\{G_i\}_{i=1}^C$ and multiple style images $\{S_i\}_{i=1}^M$. We construct a cost matrix $\mathbf{Q} \in \mathbb{R}^{C \times M}$, where each entry Q_{ij} represents the correlation between regions G_i and S_j. We first render each point set G_i into image I_i,

Figure 6: Qualitative comparisons with snerf (Nguyen et al. 2022), arf (Zhang et al. 2022) and ref-npr (Zhang et al. 2023) on tnt datasets in multiple style setting. **Yellow boxes (O, P, U, V): inconsistency can lead to blurry results; orange boxes (Q, R): incorrect color blending between multiple regions; blue boxes (S, T): our method can produce finer texture details.**

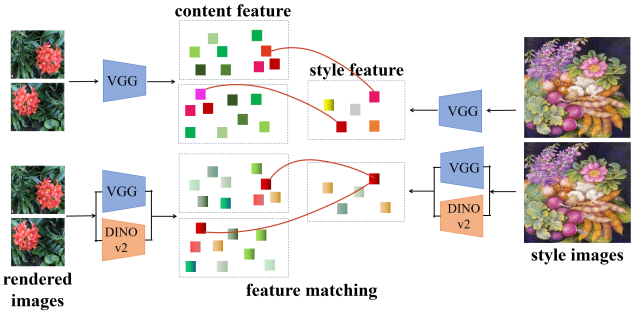

Figure 7: The VGG features do not ensure consistency across different viewpoints, leading to the same object being associated with distinct features when viewed from various angles. However, by incorporating DINOv2, we can maintain local details while also achieving enhanced consistency in feature matching, regardless of the viewing perspective.

then utilize VGG19 (Simonyan and Zisserman 2015b) to extract features from both the rendered image I_i and the stylized image S_j. The correlation is determined by the cosine feature distance between the means of features of I_i and S_j.

Given the cost matrix $\mathbf{Q}$, an optimal mapping $\mathcal{M} : [1, C] \mapsto [1, M]$ can be generated by Hungarian algorithm. Our multi-style loss can be finally formulated as

$$\mathcal{L}^{\text{style}} = \sum_{j=1}^{C} \sum_{\mathbf{f} \in C_{I_j}} \min_{\substack{k=\mathcal{M}(j) \\ \mathbf{g} \in C_{S_k}}} d(\mathbf{f}, \mathbf{g}), \tag{12}$$

where $\mathbf{f}$ and $\mathbf{g}$ are pixel-wise features, d measures the cosine distance. Through the minimization of our multi-style loss, we augment the GS model with the ability to perform stylization with multiple styles. Such a design not only enables local style transfer but also significantly alleviates the burden on GPU memory. By strategically categorizing Gaussians into several distinct categories, our model circumvents the need to apply splatting to all Gaussians in a single pass. Moreover, the stylization process with these categorized Gaussians naturally ensures that the resulting appearance exhibits seamless continuity and unambiguity.

Training Details

Reconstruction stage. Our GS model is trained with

$$\mathcal{L}^{\text{recon}} + \lambda^{\text{seg}}\mathcal{L}^{\text{seg}} + \lambda^{\text{KNN}}\mathcal{L}^{\text{KNN}} + \lambda^{\text{NE}}\mathcal{L}^{\text{NE}} + \lambda^{\text{mask}}\mathcal{L}^{\text{mask}}, \tag{13}$$

where $\mathcal{L}^{\text{recon}}$ is the Mean Squared Error (MSE) reconstruction loss as outlined in (Kerbl et al. 2023). We typically assign values of $\lambda^{\text{seg}} = 0.02$, $\lambda^{\text{KNN}} = 0.005$, $\lambda^{\text{NE}} = 0.005$.

Stylization stage. After reconstruction, we can obtain a region mapping $\mathcal{M}$. During the stylization stage, we utilize the mapping $\mathcal{M}$ and train the model by minimizing

$$\lambda^{\text{cont}}\mathcal{L}^{\text{cont}} + \lambda^{\text{style}}\mathcal{L}^{\text{style}}, \tag{14}$$

where $\mathcal{L}^{\text{cont}}$ is the content loss, which measures the MSE between the encoded feature map and the ground truth.

Experiments

Datasets

We conducted extensive experiments on a diverse set of real-world scenes, including outdoor environments from the Tanks and Temples (shortened as "tnt" in our paper) dataset (Knapitsch et al. 2017) and forward-facing scenes from the llff dataset (Mildenhall et al. 2019).

Evaluation Metrics

We conduct a user study to evaluate the stylization quality. We perform quantitative comparisions on multi-view consistency (Chiang et al. 2022a). Additionally, we provide visual comparisons and results.

Baselines

On llff datasets (Mildenhall et al. 2019), we compare our method to the SOTA methods, *e.g.*, arf (Zhang et al. 2022), lsnerf (Pang, Hua, and Yeung 2023), snerf (Nguyen-Phuoc,

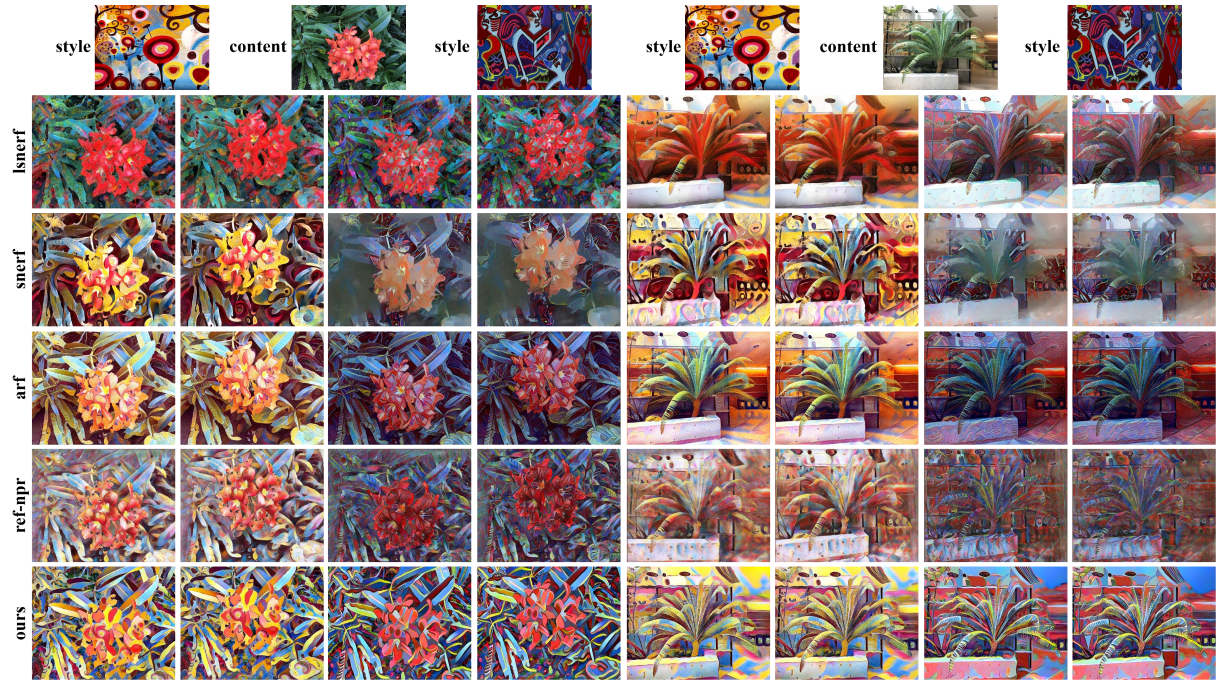

Figure 8: Qualitative comparisons with snerf (Nguyen et al. 2022), arf (Zhang et al. 2022), ref-npr (Zhang et al. 2023) and lsnerf (Pang, Hua, and Yeung 2023) on *flower* and *fern* scenes (Mildenhall et al. 2019).

Liu, and Xiao 2022) and ref-npr (Zhang et al. 2023). Arf and snerf stylize the whole scene with a style image; lsnerf establishs region correspondences between the style image and the content to support local style transfer; ref-npr stylizes scenes with a reference image and VGG matching.

Qualitative and Quantitative Comparisons

In Figure. 8, we qualitatively compare with other methods on **llff dataset in single style setting**. Our method provides clearer colors and more accurately stylized texture. Lsnerf fails to fully transfer styles; snerf generates error geometry and produce blurred images; arf employs nnfm in VGG matching, yet suffers from incorrect color blending due to VGG's inability of 3D awareness; ref-npr transfers styles from reference view using VGG but is limited to simple style and struggles with high-frequency signals (*fern* scene). In Figure. 4, we compare our result with two multi-style methods on **llff datasets in multiply style setting**, lsnerf and ref-npr suffer from multi-view inconsistency, which results in a blurred background. Our method delivers consistent outcomes, detailed textures, and improved color matching.

In Figure. 5, we compare our results with four SOTA methods on **tnt datasets in single style setting**. Snerf generates blurry images due to geometry error; ARF underperforms in the *horse* scene due to its inability to capture fine details and maintain color consistency; ref-npr can only transfer smooth style image and render blurry background in *style2*. Our method adeptly reconstructs scenes, meticulously maintaining the original's geometric and semantic information. Figure. 6 presents results with **multiple styles on tnt datasets**; for the *truck* scene, ref-npr struggles with backgrounds and blends with two styles, whereas our model cleanly distinguishes and separates them from the truck.

Methods	truck,horse,flower	Avg.
snerf (Nguyen et al. 2022)	3.00 , 2.7 , 0.78	2.16
arf (Zhang et al. 2022)	2.23 , 1.50 , 0.16	1.29
ref-npr (Zhang et al. 2023)	3.12 , 1.78 , 0.88	1.92
ours (vgg)	1.71 , 1.70 , **0.14**	1.18
ours	**1.55** , **1.21** , **0.14**	**0.96**

Table 1: Quantitative comparisions of multi-view consistency.

In Table. 1, we use the metrics from (Chiang et al. 2022a) to measure the consistency. We generate rendered videos for each scene and randomly sample 50 frames 5 times to calculate their consistency. Our method achieves the best multi-view consistency scores in all metrics. Ablation Studies: ***Due to limited space, we move additional experimental analysis and ablation studies to the supplementary material.***

Limitations and Future Works

We also notice that our model is incapable of instant style transfer and thus requires retraining for different styles. We will leave this for future improvement. Additionally, incomplete segmentation map will lead to blurry stylization results.

References

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-NeRF: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 333–350. Springer.

Cheng, H. K.; Oh, S. W.; Price, B.; Schwing, A.; and Lee, J.-Y. 2023. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1316–1326.

Chiang, P.-Z.; Tsai, M.-S.; Tseng, H.-Y.; Lai, W.-S.; and Chiu, W.-C. 2022a. Stylizing 3D Scene via Implicit Representation and HyperNetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Chiang, P.-Z.; Tsai, M.-S.; Tseng, H.-Y.; Lai, W.-S.; and Chiu, W.-C. 2022b. Stylizing 3D Scene via Implicit Representation and HyperNetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

El Banani, M.; Raj, A.; Maninis, K.-K.; Kar, A.; Li, Y.; Rubinstein, M.; Sun, D.; Guibas, L.; Johnson, J.; and Jampani, V. 2024. Probing the 3D Awareness of Visual Foundation Models. In *CVPR*.

Fan, Z.; Jiang, Y.; Wang, P.; Gong, X.; Xu, D.; and Wang, Z. 2022. Unified implicit neural stylization. In *European Conference on Computer Vision*, 636–654. Springer.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5501–5510.

Höllein, L.; Johnson, J.; and Nießner, M. 2022. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6198–6208.

Hu, W.; Wang, Y.; Ma, L.; Yang, B.; Gao, L.; Liu, X.; and Ma, Y. 2023. Tri-MipRF: Tri-Mip Representation for Efficient Anti-Aliasing Neural Radiance Fields. In *ICCV*.

Huang, H.-P.; Tseng, H.-Y.; Saini, S.; Singh, M.; and Yang, M.-H. 2021. Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13869–13878.

Huang, Y.-H.; He, Y.; Yuan, Y.-J.; Lai, Y.-K.; and Gao, L. 2022. 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*, 18342–18352.

Jung, H.; Nam, S.; Sarafianos, N.; Yoo, S.; Sorkine-Hornung, A.; and Ranjan, R. 2024. Geometry Transfer for Stylizing Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8565–8575.

Kato, H.; Ushiku, Y.; and Harada, T. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3907–3916.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4).

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2023. Compact 3D Gaussian Representation for Radiance Field. *arXiv preprint arXiv:2311.13681*.

Liu, K.; Zhan, F.; Chen, Y.; Zhang, J.; Yu, Y.; Saddik, A. E.; Lu, S.; and Xing, E. 2023. StyleRF: Zero-shot 3D Style Transfer of Neural Radiance Fields.

Liu, K.; Zhan, F.; Xu, M.; Theobalt, C.; Shao, L.; and Lu, S. 2024. StyleGaussian: Instant 3D Style Transfer with Gaussian Splatting. *arXiv preprint arXiv:2403.07807*.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4990–4998.

Miao, X.; Bai, Y.; Duan, H.; Wan, F.; Huang, Y.; Long, Y.; and Zheng, Y. 2024. ConRF: Zero-shot Stylization of 3D Scenes with Conditioned Radiation Fields. arXiv:2402.01950.

Michel, O.; Bar-On, R.; Liu, R.; Benaim, S.; and Hanocka, R. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13492–13502.

Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4): 1–14.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*, 405–421. Springer.

Mu, F.; Wang, J.; Wu, Y.; and Li, Y. 2022. 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*, 16273–16282.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4): 1–15.

Nguyen-Phuoc, T.; Liu, F.; and Xiao, L. 2022. SNeRF: Stylized Neural Implicit Representations for 3D scenes. In *ACM Transactions on Graphics*.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Howes, R.; Huang, P.-Y.; Xu, H.;

Sharma, V.; Li, S.-W.; Galuba, W.; Rabbat, M.; Assran, M.; Ballas, N.; Synnaeve, G.; Misra, I.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINOv2: Learning Robust Visual Features without Supervision.

Pang, H.-W.; Hua, B.-S.; and Yeung, S.-K. 2023. Locally stylized neural radiance fields. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 307–316. IEEE Computer Society.

Reiser, C.; Peng, S.; Liao, Y.; and Geiger, A. 2021. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14335–14345.

Simonyan, K.; and Zisserman, A. 2015a. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society.

Simonyan, K.; and Zisserman, A. 2015b. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society.

Sun, C.; Sun, M.; and Chen, H. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *CVPR*.

Wang, C.; Jiang, R.; Chai, M.; He, M.; Chen, D.; and Liao, J. 2023. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*.

Ye, M.; Danelljan, M.; Yu, F.; and Ke, L. 2023. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*.

Zhang, D.; Fernandez-Labrador, C.; and Schroers, C. 2024. Coarf: Controllable 3d artistic style transfer for radiance fields. In *2024 International Conference on 3D Vision (3DV)*, 612–622. IEEE.

Zhang, K.; Kolkin, N.; Bi, S.; Luan, F.; Xu, Z.; Shechtman, E.; and Snavely, N. 2022. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, 717–733. Springer.

Zhang, Y.; He, Z.; Xing, J.; Yao, X.; and Jia, J. 2023. Ref-NPR: Reference-Based Non-Photorealistic Radiance Fields for Controllable Scene Stylization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4242–4251.

Zhou, S.; Chang, H.; Jiang, S.; Fan, Z.; Zhu, Z.; Xu, D.; Chari, P.; You, S.; Wang, Z.; and Kadambi, A. 2024. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21676–21685.