Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

TSNeRF: Text-driven Stylized Neural Radiance Fields via Semantic Contrastive Learning

Yi Wang[a], Jing-Song Cheng[a], Qiao Feng[a], Wen-Yuan Tao[a], Yu-Kun Lai[b], Kun Li[a,*]

[a]Tianjin University, Tianjin 300350, China
[b]Cardiff University, Cardiff CF24 4AG, U.K.

ARTICLE INFO

Article history:
Received July 3, 2023

Keywords:
3D scene stylization
Text-driven stylization
CLIP
Semantic contrastive learning

ABSTRACT

3D scene stylization aims to generate impressive stylized images from arbitrary novel views based on the stylistic reference. Existing image-driven 3D scene stylization methods require a specific style reference to be given, and lack the ability to produce diverse stylization results by combining style information from different aspects. In this paper, we propose a text-driven 3D scene stylization method based on semantic contrast learning, which takes Neural Radiance Fields (NeRF) as the 3D scene representation and generates diverse 3D stylized scenes by leveraging the semantic capabilities of the Contrastive Language-Image Pre-Training (CLIP) model. For comprehensively exploiting the semantic knowledge to generate finely stylized results, we design a CLIP-based semantic contrast estimation loss, which can avoid the global stylistic inconsistency caused by the NeRF ray sampling method and avoid the tendency to stylize neutral descriptions due to the semantic averaging of the CLIP space. In addition, to reduce the memory burden arising from NeRF ray sampling, we propose a novel ray sampling method with gradient accumulation to optimize the NeRF rendering process. The experimental results indicate that our method generates high-quality and plausible results with cross-view consistency. Moreover, our method enables the creation of new styles that match the target text by combining multiple domains.

1. Introduction

This paper focuses on the problem of stylizing complex 3D scenes. In recent years, considerable efforts have been made to stylize 3D scenes. For example, some approaches use point clouds [3, 4] or meshes [5, 6] to represent and stylize 3D scenes. However, due to the explicit representation, they require accurate 3D geometry and may have limited capability to represent appearance. In contrast to explicit 3D modeling, the neural radiance field (NeRF) [7, 8, 9] can represent spatially continuous scenes with high fidelity by implicit modeling. Chiang *et al.* [2] use a two-stage NeRF training approach for 3D scene stylization, and Zhang *et al.* [1] apply a style loss based on Nearest Neighbor Feature Matching (NNFM) to create 3D artistic radiance fields. However, the above NeRF methods must be supervised based on images, which limits their capability for diversify stylization in multiple domains, as such style examples may not exist, *e.g.*, "Tropical Fish Painting in the Style of Disney".

In this paper, our goal is to stylize 3D scene according to a given target style text. This allows generating novel view images of a specified style or a completely unseen style according to the textual multiple-domain semantic information, while maintaining consistency across multiple views. To ensure accurate style transfer results of 3D scenes according to given tex-

*Corresponding author.
e-mail: lik@tju.edu.cn (Kun Li)

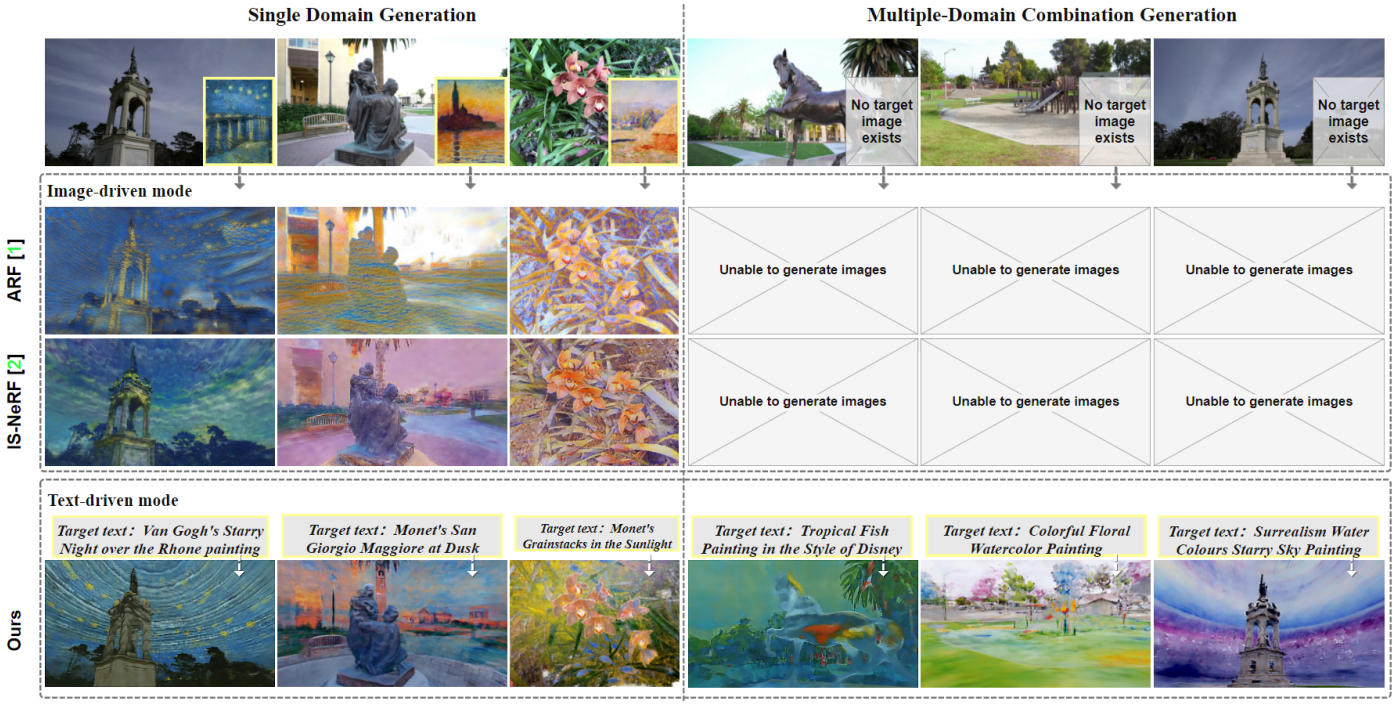

Fig. 1: Previous NeRF stylization methods often require a target image as supervision. ARF method[1] matches the features of the rendered image to the target style image with nearest neighbor (NN) feature vector. IS-NeRF method[2] guides NeRF stylization with stylized latent code. However, such methods cannot work without a target style image. Our proposed Text-driven Stylized NeRF (TS-NeRF) based on semantic contrastive learning can solve this problem. TS-NeRF can generate 3D stylized scenes that match the text description with multi-view consistency. Moreover, by using text descriptions describing combinations of different aspects of the expected style, our method achieves multiple domain combination generation, where no single image style reference exists.

tual description, a semantic contrastive learning strategy is used to supervise NeRF [10] stylization in the CLIP (Contrastive Language-Image Pre-Training) [11] embedding space, which combines a common embedding space for both text and images. Some examples of our NeRF stylization method are presented in Fig. 1.

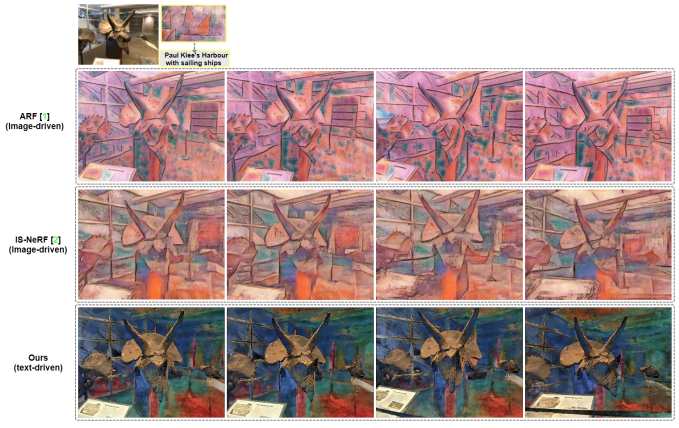

Fig. 2: Comparison of our proposed global-local correlation with gradient accumulation (GL-GA) method and existing NeRF ray sampling method. Although the ARF method [1] obtains the style loss of the full-resolution image with a deferred back-propagation method similar to the purpose of ours. However, the ARF method [1] mainly applies the style loss of the full-resolution image for optimizing the local style rather than the global style. The IS-NeRF method [2] by Chiang *et al.* only utilizes small patches for ray sampling during training, which will result in unsmooth images and inconsistent global stylization. Our ray sampling method generates a smooth global style and consistent results by combining a text-driven framework.

However, there are two issues with text-driven NeRF scene stylization. Firstly, text-guided training via the CLIP model is challenging, and a naive approach is to minimize the difference in the CLIP space between the NeRF-rendered image and the target text. In practice, it is adversely affected by mode-collapsed examples. If the target generator only creates a single image, the CLIP space directions from all sources to this target image will be different. They cannot all align with the textual direction. StyleGAN-NADA [12] encourages generative diversity and avoids adversarial results by aligning the CLIP space orientation between source and target text image pairs. However, StyleGAN-NADA only uses the same CLIP space orientation to guide latent code optimization, which does not explicitly regulate editing strength. Therefore, the model can easily over-edit the latent code, leading to inaccuracy or over-editing. Secondly, it is required for NeRF to query dense 3D points along the ray to render individual pixels. Because of memory limitations, it is hard to render the whole image or sufficiently large patches. The ARF method [1] utilizes a delayed back-propagation method to evaluate full-resolution stylization loss that enhances stylized details and strokes, but has limited effect on the global style. Chiang *et al.* [2] address this limitation by using the patch ray sampling method for NeRF rendering instead of ray sampling the entire image. However, approaches based on smaller size patches tend to suffer from global style inconsistency and image blurring [13]. The issues of the above two methods are shown in Fig. 2.

To solve the above problems, we propose a framework of multimodal text-guided NeRF for style transfer based on se-

mantic contrastive learning. We also optimize the ray sampling by our global-local correlation with gradient accumulation (GL-GA) method. The framework consists of two parts: training a NeRF network to represent the geometric branches of the scene, and then creating a stylized branch consisting of a CLIP model, a style module, and a semantic optimization module. In the stylization stage, we use the target text latent code encoded by the CLIP text encoder to predict the color of the stylized branch. To handle style transfer directions more precisely and avoid falling into local minima or undesired edits, we design a CLIP-based Hard negative sample Contrastive Learning (CLIP-HCL) loss for semantics. It not only emphasizes the same direction as in the directional CLIP loss [12] but also maximizes (resp. minimizes) positive (resp. negative) sample mutual information in the CLIP space, and thus generates auxiliary information to optimize network learning. Further, the patch-based ray sampling method in NeRF affects global stylization consistency. To address this, we propose an effective method of gradient accumulation. This method splits the input image into patches for forward rendering separately, and then stitches the patches into a whole image for gradient back-propagation, which improves the smoothness and global style consistency of the rendered images.

Our main contributions are summarized as follows:

- We propose a novel text-driven NeRF stylization method for accurate and high-quality 3D scene stylization with a given target text description.
- We propose an approach to enhancing the mutual information of target text image pairs during NeRF stylization by a semantic contrastive loss. It not only improves the consistency of the global style, but also avoids the stylization of neutral descriptions due to the CLIP space semantic averaging.
- We design a global-local correlation method for NeRF ray sampling with gradient accumulation, which optimizes the NeRF sampling and the rendering process from the whole image perspective. Moreover, the method provides full image semantics for text-driven stylization, thus improving the accuracy of CLIP loss.

2. Related Work

2.1. Novel View Synthesis

Novel view synthesis aims to synthesize a target image of an arbitrary camera position from a set of source images. In the last few years, neural networks have advanced tremendously in the implicit representation of 3D models [14, 7, 8, 15, 16]. One representative work is NeRF [10], which uses a Multi-Layer Perceptron (MLP) [17] to implicitly represent the volume density and the color of a scene. It learns view-consistent continuous representations of the 3D scene with 3D points and camera information. Compared to traditional explicit representations, *e.g.*, point clouds [18, 19], voxels [20, 21] and meshes [22, 23], 3D implicit representations already have the potential to reconstruct complex high-resolution geometries and appearances of 3D scenes, with continuity and high fidelity. The success of NeRF has also led to many subsequent works to extend NeRF towards conditional radiance fields [24, 25, 26], editable scenes [27, 28, 29], unbounded scenes [30, 8] *etc.* In addition, CLIP-NeRF [31] presents the first text and image-driven manipulation approach for NeRF, in which a unified framework is employed to provide users with flexible control over 3D content with textual cues or example images and present impressive results. Moreover, GRAF [32] is a primary approach to conditionally synthesizing novel views of NeRF by means of shape and appearance latent codes.

2.2. 3D Style Transfer

3D style transfer is a persistent research topic in the field of computer vision [33]. 3D style transfer aims to change the style of 3D objects while maintaining consistency across multiple views. Most approaches either use point clouds [34, 35] or meshes [5, 6, 36] for style transfer. Some other works perform style transfer on geometry and textures [34, 37, 38], *etc*. However, all the approaches above are limited to the object level rather than the whole scene. For style transfer at the 3D scene level, recent approaches use point clouds [3] or meshes [39] *etc.* However, they do not support the synthesis of stylized novel views with continuous representations, restricted by the ability of discrete geometric representations. The work [2] utilizes NeRF for 3D scene stylization. However, owing to memory restrictions, they conduct ray sampling by the patch method, which tends to result in a lack of global stylistic consistency and thus a blurred and grainy stylized image. In contrast, we propose GL-GA optimization method for full-resolution rendering. By this method, global features are obtained for style loss estimation to maintain global style consistency. In addition, the StylizedNeRF method [40] employs a mutual learning approach to gradually fuse 2D stylized features into 3D space instead of directly stylizing in 3D space, which would make the stylized results highly dependent on the validity of the mutual information between 2D stylized features and 3D spatial features. The ARF method [1] utilizes a delayed back-propagation method to evaluate the full-resolution stylization loss for improving the stylization details, but it has limited effect on the global style and cannot achieve stylization with multiple domains.

2.3. CLIP-driven Image Generation and Manipulation

We use CLIP [11] in our style transfer model, which connects text and images by bringing them closer together in a shared latent space through a comparison learning approach. Several approaches for text-driven image generation and stylization have been proposed with the support of CLIP models. Crowson *et al.* [41] combine CLIP and VQGAN (Vector Quantized Generative Adversarial Network) [42] to synthesize images by optimizing the latent code of a pre-trained VQGAN based on textual conditions defined in the CLIP space. In addition to applying CLIP to GAN models, DiffusionCLIP [43] combines diffusion models [44] and CLIP for text-driven image processing. It acheives a performance comparable to GAN-based image processing approaches, with the advantages of pattern coverage and training stability. However, these methods are restricted to image

processing and are incapable of maintaining multi-view consistency due to the lack of 3D information. In contrast, our approach is guided by CLIP-encoded stylistic latent code for NeRF to conduct appearance reconstruction. The 3D spatial points are directly stylized, so it allows for novel view synthesis in a view-consistent manner. In addition to the above image generation methods, the NeRF-Art method [45] can use CLIP [11] for avatar generation driven by text, which is a meaningful work.

3. Preliminaries

NeRF [10] represents a 3D scene with an implicit neural function that maps 3D query points $\mathbf{q} = (x, y, z)$ and ray directions $\mathbf{d} = (\theta, \phi)$ into radiant colors $\mathbf{c} = (r, g, b)$ and densities σ, *i.e.* $(\mathbf{q}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. Specifically, it assumes that the camera ray r is emitted in a direction $\mathbf{d}$ from the camera center $\mathbf{o}$, *i.e.* $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where t denotes the sampling position along the ray. The predicted color $\hat{C}(\mathbf{r})$ of $\mathbf{r}(t)$ is defined as:

$$\begin{aligned} \hat{C}(r) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t), \mathbf{d})dt \\ T(t) &= \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right), \end{aligned} \tag{1}$$

where t_n and t_f denote the near and far boundaries of the rays and $T(t)$ indicates cumulative transmittance. To improve the model's ability to fit high-frequency data, Fourier coding is applied in NeRF to map the input position $\mathbf{p}$, camera direction $\mathbf{d}$, to high-frequency space with the following expressions:

$$\gamma(v) = \left[\sin(v), \cos(v), \ldots, \sin\left(2^{L-1}v\right), \cos\left(2^{L-1}v\right)\right]^T, \tag{2}$$

where v is an arbitrary value and L is the level of Fourier coding.

4. Method

Our goal is to generate stylized images from arbitrary novel views, with textual prompts only, given a collection of scene images with camera parameters, while maintaining view consistency. For this purpose, we propose a text-driven NeRF stylization solution based on semantic contrastive learning, Fig. 3 shows the framework of our solution, in which we adopt CLIP [11], a powerful model for joint visual language representation. However, if the global cosine loss is directly calculated for NeRF-rendered images with target text in the CLIP space, it can get into local minima or undesired style transfer [12]. Although the directed CLIP loss [12] partly relieves this issue, it tends to allow the model to over-edit the latent code, which leads to degradation of image quality and object semantic meaning. In contrast, we design a novel contrastive loss to guide CLIP toward the optimal direction from different angles by fully leveraging the semantics of CLIP. In this way, the style transfer direction can be edited precisely. Meanwhile, in order to solve the issue of global style inconsistency and image blurring caused by NeRF with patch ray sampling, we propose to use GL-GA method for improvement. The CLIP loss is evaluated with full-resolution rendered images stitched together from multiple patches, thus globally constraining the style of local images.

Next, we first introduce the text-driven 3D stylized implicit representation in Sec. 4.1, and then discuss the ray sampling method of GL-GA in Sec. 4.2. Finally, we discuss how to combine the text-driven stylization method and the optimized NeRF ray sampling method for NeRF network training in Sec. 4.3.

4.1. Multimodal Text-driven Stylized 3D Implicit Representation

We focus more on how to guide NeRF to perform style transfer along the precise textual semantic direction in the embedding space of CLIP, in contrast to the image-driven NeRF stylization approaches [47, 2] which concern how to stylize the scene with NeRF supervised by stylistic features of reference images.

It is widely known that global CLIP loss used in text-guided style transfer may lead to local minimum problems during optimization. Taking 2D StyleCLIP [48] as an example, the global CLIP loss may induce blind stylization of entangled regions other than the target region. Directional CLIP loss [12] alleviates this issue by aligning the CLIP space orientation between the text-image pairs of the source and target. However, constraining the stylistic representation of NeRF with a fixed CLIP space orientation from the source text to the target text would ignore other useful information embedded in the CLIP space, and could lead to over-editing results in stylization.

Consequently, we propose a CLIP-based contrastive learning loss with hard negative samples (CLIP-HCL) to maximize (resp. minimize) the mutual information with positive (resp. negative) samples, which allows to more fully leverage the semantic information of CLIP to generate style latent code with a uniform style. To exploit the CLIP-HCL loss, we first define query samples, positive and negative samples, which are similar to common contrastive losses [49, 50, 51]. Then, CLIP-HCL makes the query closer to the positive samples while moving it away from the negative samples, as shown in Fig. 4. We illustrate the losses as follows.

First, we define the query vector in the loss:

$$Q_I = E_I\left(I_{\text{target}}\right) - E_I\left(I_{\text{source}}\right), \tag{3}$$

where $E_I(\cdot)$ is the CLIP image encoder. Q_I denotes the CLIP embedding space orientation from the source image I_{source} to the NeRF rendered image I_{target}, which is being optimized during network training. We then use text guidance to define positive and negative samples. Denote by $E_T(\cdot)$ the CLIP text encoder. The CLIP space direction D_I from the source image to the target text is defined as

$$D_I = E_T\left(t_{\text{target}}\right) - E_I\left(I_{\text{source}}\right). \tag{4}$$

We then define two types of positive samples as follows:

$$\begin{aligned} D_T^+ &= E_T\left(t_{\text{target}}\right) - E_T\left(t_{\text{source}}\right), \\ \widehat{D}_{\text{I}}^+ &= \frac{1}{2}\left(D_{\text{I}} + D_{\text{I}}^-\right) + \frac{\lambda}{2}\left(D_{\text{I}} - D_{\text{I}}^-\right), \end{aligned} \tag{5}$$

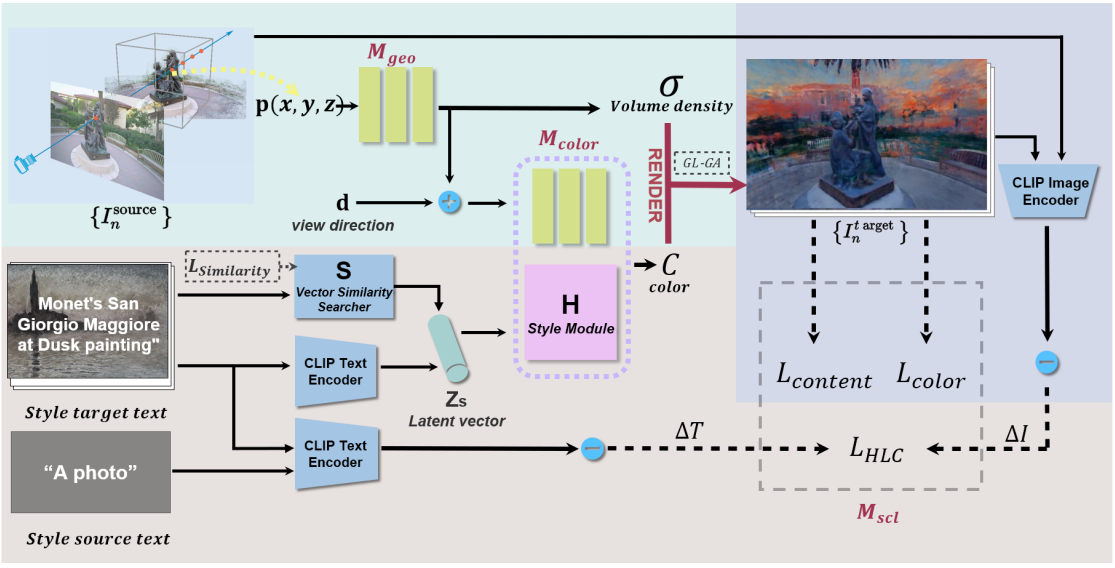

Fig. 3: Our text-driven NeRF stylization model architecture. The framework consists of a geometric prediction branch M_{geo}, an appearance prediction branch M_{color} and a semantic contrastive learning network M_{scl} consisting of a CLIP model. (a) First, the target style text is encoded into the CLIP embedding space, generating a latent code Z_s to optimize the stylization module H to predict the color of the appearance branch M_{color}. In particular, the vector similarity searcher S is responsible for fine-tuning the CLIP space embedding accuracy of latent code Z_s with the ArtBench artwork database [46]. (b) Second, the geometrically pre-trained NeRF network is used to render the multi-view input $\{I_n^{\text{source}}\}$ as stylized images $\left\{I_n^{\text{target}}\right\}$ guided by latent code Z_s with GL-GA sampling. (c) Then the direction ΔI of the CLIP space from $\{I_n^{\text{source}}\}$ to $\left\{I_n^{\text{target}}\right\}$ is conditionally parallel to the direction ΔT of the source text to the target text guided by contrastive learning. The objective functions L_{HCL}, L_{color} , L_{content} and $L_{\text{similarity}}$ are loss terms used to optimize the style module (see Sec. 4.3 for details).

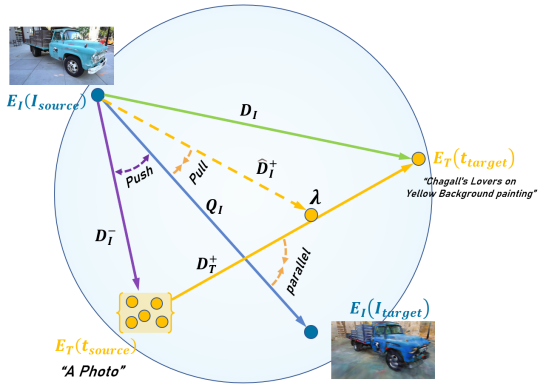

Fig. 4: Illustration of our contrastive learning method CLIP-HCL. Contrastive loss is evaluated in the CLIP embedding space, where the direction of $E_I\left(I_{\text{source}}\right)$ to $E_I\left(I_{\text{target}}\right)$ (blue arrow) is used as the query to be optimized. Two positive samples $\widehat{D}_I^+$ and D_T^+ (shown by orange arrows) are then calculated, where D_T^+ encourages the query to be aligned with the direction from the source text to the target text, and $\widehat{D}_I^+$ regularizes the query direction to pull away from the negative samples. $\widehat{D}_I^+$ is obtained by offsetting the direction D_I from the source image to the target text by the hyperparameter λ, which prevents the query from being overly regularized by the direction D_I. The purple color is the negative sample from the source image to the source text. By pulling positive pairs and pushing negative pairs, CLIP-HCL comprehensively uses the CLIP representation to precisely guide the stylized direction.

where $\lambda \in (0, 1)$ is the hyperparameter controlling the degree of $\widehat{D}_I^+$ offset from D_I and λ is set to 0 by default in this experiment. D_I^- is the CLIP space direction of the source image to the source text, *i.e.*, a negative sample direction.

The first term D_T^+ indicates the CLIP space direction from the source text to the target text. It regularizes the query direction to align with the embedding direction from the source text to the target text.

The second term $\widehat{D}_I^+$ is obtained from an offset of the CLIP space direction D_I from the source image to the target text. In this task, if the query is completely regularized by D_I, it will overfit the positive samples, and thus NeRF rendering color is over-edited, as shown with green boxes in Fig. 5. However, at the same time, keeping $\widehat{D}_I^+$ not far from the original position D_I is beneficial to constrain the text embedding direction D_T^+, because the model overly emphasizes D_T^+, which will ignore other useful information contained in the CLIP space embedding. In order to fully extract the semantic information in the CLIP space, exploiting the comprehensive information in the CLIP space helps to provide thorough guidance for style transfer. Therefore, we define negative samples D_I^- as the CLIP space direction from the source image to various competing textual descriptions, which prevents the lazy operation of the model to generate neutral descriptions from competing texts. Therefore, we add negative samples D_I^- to avoid style transfer towards neutral descriptions:

$$D_I^- = E_T\left(t_{\text{source}}\right) - E_I\left(I_{\text{source}}\right). \tag{6}$$

To increase the diversity of the source text, we use a template of competitive text to fill t_{source} , such as "A photo of a {painting}". According to [52], maximizing mutual information allows diverse and reasonable solutions and helps avoid averaging solutions across instances. Therefore, we achieve our defined semantic contrastive learning terms based on the HCL loss [53]. By maximizing the mutual information between the query and selected positive samples in the CLIP space while minimizing it between the query and negative samples, the CLIP-HCL objective function provides comprehensive guidance in

Fig. 5: Comparison of results using $\widehat{D}_I^+$ and D_I. The first row shows complete regularization of the query by D_I, which causes the stylized results to be over-edited. The second row shows the results after properly offsetting D_I to the positive and negative decision plane by λ, in which the colors return to normal.

the CLIP space and precisely directs style transfer. Specifically, the proposed CLIP-HCL loss can be expressed as follows:

$$L_{HCL} = -\alpha \log \frac{e^{(Q_I \cdot D_T^+/\tau)}}{e^{(Q_I \cdot D_T^+/\tau)} + \sum_{D_I^-} e^{(Q_I \cdot D_I^-/\tau)}} - \beta \log \frac{e^{(Q_I \cdot \widehat{D}_I^+/\tau)}}{e^{(Q_I \cdot \widehat{D}_I^+/\tau)} + \sum_{D_I^-} e^{(Q_I \cdot D_I^-/\tau)}}, \tag{7}$$

where τ is the temperature, which is set to 0.1 in our experiments, the hyperparameter $\alpha \in (0, 1)$ controls the degree of alignment of the style transfer direction with the CLIP direction of the source text to the target text, and $\beta \in (0, 1)$ controls the degree of style transfer direction regularized by the positive text sample.

4.2. Ray Sampling Method With Gradient Accumulation

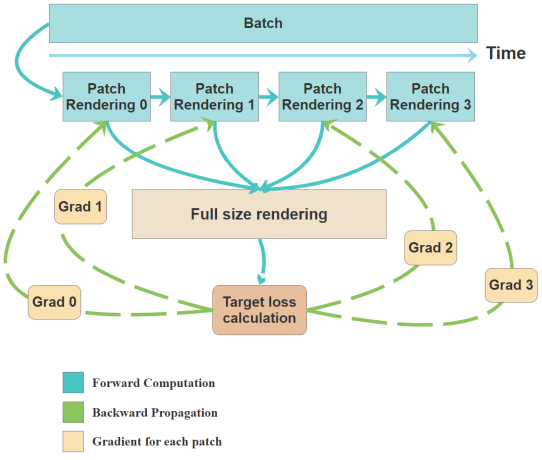

Fig. 6: Illustration of the ray sampling method of GL-GA.

Since NeRF performs intensive 3D point queries on each ray, which is a memory-intensive computation, smaller patches are often utilized for sampling the rays to reduce memory overhead. In particular, in the NeRF style generation with CLIP-HCL constraints, we experimentally find that it is hard to align the target text in the CLIP space [12] and this causes the network to converge slowly by only patch sub-sampling for the rendered images. Therefore, we propose a balanced NeRF ray sampling method, which not only enables each patch to evaluate the target loss with global features but also ensures that no memory overflow problem occurs during NeRF ray sampling. The principle of sample rendering is shown in Fig. 6.

Firstly, we divide the full-resolution input image into multiple patches with $\hat{N}$ rows and $\hat{M}$ columns and then ray-sample and forward-render all the patches sequentially in one iteration of training. Subsequently, all the rendered patches are combined into one full-resolution target image for CLIP-HCL loss calculation. It ensures that the CLIP-HCL loss is evaluated based on the whole target image, which can improve the accuracy of the loss and avoid stylization inconsistency between rendered patches. Secondly, to reduce the GPU memory footprint due to gradient accumulation, we randomly cancel the gradient of some patches at a rate of 10%−30% in this experiment during backpropagation. Finally, the network is back-propagated and the gradients of multiple patches are updated simultaneously. The results shown in Fig. 12 demonstrate that our ray sampling method is beneficial for providing complete image semantic information for semantic contrastive learning.

4.3. Training

Our training process for multimodal text-driven 3D stylization consists of a scene geometry training stage and a scene stylization training stage. We will describe the training process and the corresponding objective functions for both stages as follows.

4.3.1. Geometry Training Stage

In the geometry training stage, our goal is to quickly build the implicit geometry scene, which does not involve a global stylized appearance. So for this stage, at each optimization iteration, we randomly sample M rays from a batch of sampled rays $\{r_m\}_{m=1,2,\ldots,M}$ of the input image set and perform a dense 3D point query along each ray. These 3D points and view directions are mapped to Fourier features [10] including multi-scale frequencies, and then these features are used as input to predict the volumetric density and color set of the scene. The

cumulative color values $\hat{c}(r_m)$ for each ray r_m are rendered with a volume renderer, and then the mean squared error (MSE) is calculated between $\hat{c}(r_m)$ and the ground-truth color $c(r_m)$ of the corresponding image pixel of the ray r_m:

$$L_{\text{color}} = \frac{1}{M}\sum_{m=1}^{M} \|\hat{c}(r_m) - c(r_m)\|_2 . \tag{8}$$

4.3.2. Text-driven Stylization Training Stage

In the NeRF-based geometry training stage, we have obtained a complete geometric representation of the scene. In the stylization stage, we first encode the target style text into a style latent vector Z_s in a joint multimodal embedding space by leveraging the text encoder of the CLIP model. To improve the accuracy of the CLIP's text encoding, we exploit the vector similarity searcher S to search the CLIP embedding Z_I of the nearest neighbor painting similar to Z_s in the ArtBench art database [46] under single-domain conditions. We calculate the cosine similarity between Z_s and Z_I to fine-tune Z_s with the following loss function:

$$L_{\text{similarity}} = 1 - \frac{Z_s \cdot Z_I}{\|Z_s\| \, \|Z_I\|}. \tag{9}$$

Then, Z_s is applied to guide the style module H to predict the color c of the neural radiation field. The style module H is a hypernetwork composed of MLPs that are responsible for learning the stylized distribution of Z_s and decoding it into the corresponding stylized appearance. The style module H is defined as follows:

$$c = H_{\text{color}}(Z_s, \mathcal{X}, \mathcal{D}), \tag{10}$$

where $\mathcal{X}$ is the output of branch M_{geo}, $\mathcal{D}$ is the camera direction, and c is the predicted color. The predicted volume density δ and color c are then rendered by the NeRF renderer to obtain stylized images consistent with the target style text. In addition, the geometric representation of the 3D implicit scene should be retained during the stylization training stage, so we fix the weights for the geometry prediction branch M_{geo}. In this stage, we use GL-GA method (in Sec. 4.2) for NeRF ray sampling. According to the resolution of the input image and the memory of the GPU, we set the number of image patches $\hat{M} \times \hat{N}$ and the number of patches $\hat{K}$ that randomly participate in the gradient accumulation. By setting appropriate $\hat{M}$, $\hat{N}$ and $\hat{K}$, as many patches as possible participate in gradient accumulation without overflowing the memory, thus improving the efficiency of NeRF inference. In our implementation, we empirically set $\hat{M} \times \hat{N}$ to 4×4 or 5×5 and set $\hat{K}$ to $8 \sim 12$. Also, the usage of the optimized ray sampling method can allow each patch to be gradient updated using the CLIP-HCL loss of the global image, thus avoiding the problem of non-uniform style of patches and unsmooth rendered images. The specific definition of CLIP-HCL loss in the stylization stage is as follows:

$$L_{HCL} = \frac{\hat{K}}{\hat{M} \times \hat{N}} L_{HCL}^{\hat{M}\times\hat{N}}\left(t_{\text{target}}, t_{\text{source}}, I, \hat{I}\right), \tag{11}$$

where $L_{HCL}^{\hat{M}\times\hat{N}}$ is the CLIP-HCL loss of the stitched image with $\hat{M} \times \hat{N}$ patches (in Sec. 4.1), $\hat{K}$ is the number of those patches that need to be involved in gradient accumulation, t_{target} is the target style text, t_{source} is the source style text, I is the rendered stitched full-resolution map, and $\hat{I}$ is the corresponding NeRF input image. The content loss enforces I and $\hat{I}$ to have similar content features:

$$L_{\text{content}} = \|\rho(I) - \rho(\hat{I})\|_2, \tag{12}$$

where $\rho(\cdot)$ denotes the feature representation obtained from the pre-trained VGG-19 network. The overall objective function of the final text-driven style training stage, in which the gradients are back-propagated to learn the style modules H, is defined as:

$$L_{\text{full}} = L_{\text{content}} + \lambda_h L_{HCL} + \lambda_c L_{\text{color}} + \lambda_s L_{\text{similarity}}, \tag{13}$$

where λ_h, λ_c, λ_s are hyper parameters controlling the impact of loss terms, L_{HCL} is defined in Equation 7, L_{color} is defined in Equation 8, and $L_{\text{similarity}}$ is defined in Equation 9. Specifically, λ_h controls the degree to which the scene is stylized, and we set $\lambda_h \in (10, 30)$ depending on the level of stylization. L_{c} is the color loss in Sec. 4.3.1, and we set λ_{c} to 1 to avoid over-stylization in the style training stage.

4.3.3. Implementation Details

In the geometric training stage, our NeRF network is trained for 200,000 iterations, using a random ray sampling pattern for the input image. For the text-driven stylization stage, the style modules and the part of NeRF responsible for color prediction are trained for 150,000 iterations with GL-GA ray sampling method. The hyper-parameters $[\lambda_h, \lambda_c, \lambda_s]$ were set to $[20, 1, 1]$ for training. We adopt the Adam optimizer [54] for both stages with learning rates set to 0.0005 and 0.001 respectively using the default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$).

5. Experimental Results

In this section, we first give the details of the datasets and compared methods in Sec. 5.1. The comparison results with several state-of-the-art methods are presented in Sec. 5.2. Then, we conduct an ablation study to validate the effectiveness of our model in Sec. 5.3. Finally, we show some limitations of our model in Sec. 6.

5.1. Datasets and Compared Methods

Datasets. We experiment with real-world 3D scenes collected in the Tanks and Temples [55] dataset and llff dataset [56].

Compared Methods. We compare our method with image-driven NeRF stylization methods and text-driven NeRF stylization methods, constituting three baseline methods:

Image-driven NeRF stylization (ARF, IS-NeRF, Stylized-NeRF method): We use recent NeRF stylization method [1], method [2] and method [40] as image-driven NeRF stylization baseline.

Text-driven NeRF stylization (NeRF-Art): We utilize the NeRF-Art method [45] for text-driven NeRF stylization baseline.

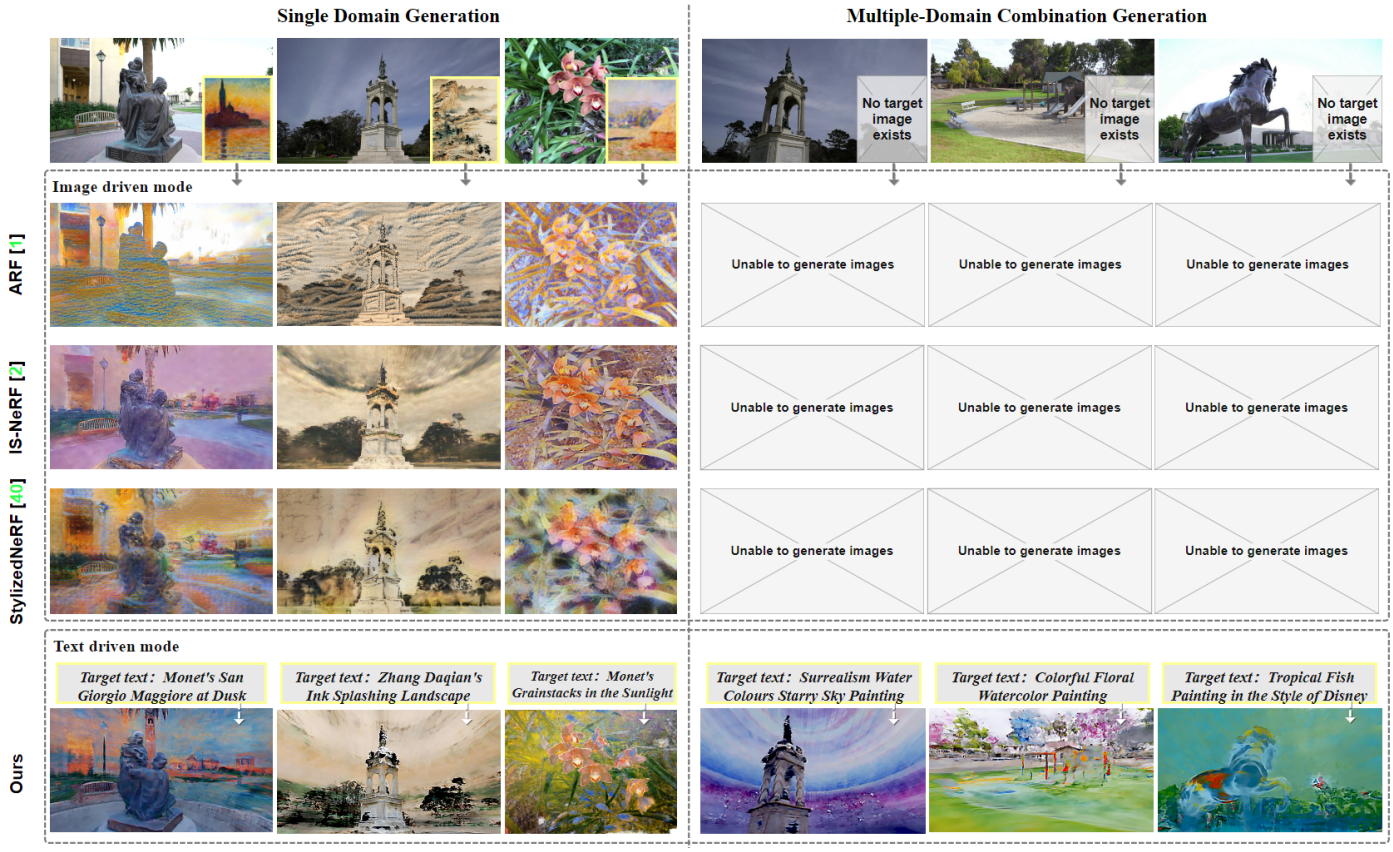

Fig. 7: Qualitative comparison with ARF method [1], IS-NeRF method [2] and StylizedNeRF method [40] on the outdoor dataset. Our method can produce multi-domain combinations of styles (rows 3). Meanwhile, our approach generates more consistent local and global styles than IS-NeRF method [2]. The ARF method [1] focuses on brush strokes and stylized details and lacks global stylized matching in terms of semantics. The stylized quality of the StylizedNeRF method [40] depends on the effectiveness of the mutual information between 2D stylized features and 3D spatial features.

5.2. Comparisons

5.2.1. Qualitative Results

In Fig. 7, We compare qualitatively with the image-based Nerf stylization method [1, 2, 40] on the outdoor dataset.

The IS-NeRF method [2] matches the VGG features of the rendered image with the stylized image by the Gram matrix of the global statistics. However, since IS-NeRF method [2] uses the patch method for NeRF sampling, neither global or local styles are fully evaluated. The ARF method [1] captures local details by finding its nearest neighbor (NN) feature vector in the VGG feature map of the style image and minimizing the distance to the rendered image feature, but the global style of some rendered images is not significant. In addition, the StylizedNeRF method [40] tries to integrate 2D stylized features with 3D spatial features by mutual learning, but the spatial perception is incomplete because stylization is not performed directly in 3D space. In conclusion, the above image-driven NeRF stylization methods [1, 2, 40] only learn the distribution of styles and lack the semantic matching of scene object stylization, which leads to the issue that local and global styles not being associated, as shown in the first three rows of Fig. 7. Our approach maximizes the mutual information between the query and selected positive samples in the CLIP space while minimizing it between the query and negative samples, and enables precise transfer of semantic stylization with comprehensive usage of useful information in CLIP space. Simultaneously, our GL-GA sampling method can perform loss calculation in the full resolution image which ensures the consistency of global and local styles. Finally, our method accurately renders the color features of the target style under single-domain conditions and makes the rendered image globally and locally semantically relevant. Moreover, under multi-domain conditions, diverse stylization can be performed while maintaining multi-view consistency, as shown in the third row of Fig. 7.

In Fig. 8, we compare qualitatively with the image-based NeRF stylization methods [1, 2, 40] on the indoor dataset. The ARF method [1] can stylize the whole scene with nearest neighbor features matching, but the local style lacks variability. The IS-NeRF method [2] utilizes patch sampling for rendering, so the resulting images are blurry, as shown in the second row of Fig. 8. Due to insufficient mutual information between 2D style features and 3D spatial features, the StylizedNeRF method [40] is not fully stylized in the spatial region as shown in the green box in the third row of Fig. 8. In addition as shown in the blue box in the third row of Fig. 8, the object contours in this region are relatively unclear due to the lack of better spatial perception for the StylizedNeRF method [40]. Thanks to the contrast learning method, our method can better stylize the scene based on the semantics of spatial objects and makes the stylized object contours clearer by using the GL-GA method.

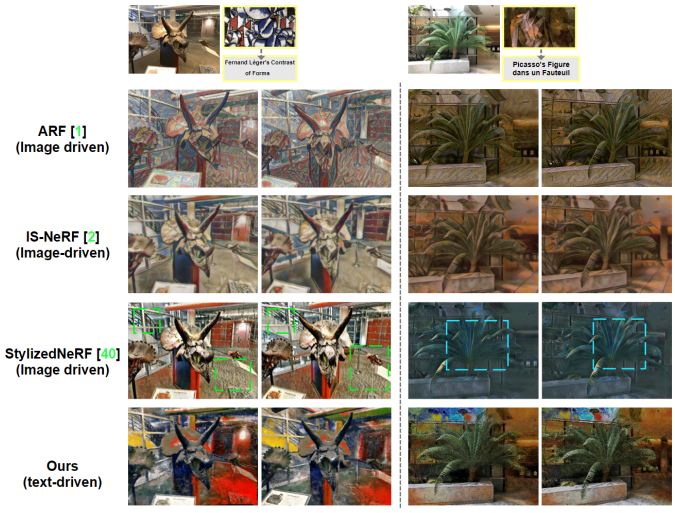

Fig. 8: Qualitative comparison with ARF method [1], IS-NeRF method [2] and StylizedNeRF method [40] on the indoor dataset.

In Fig. 9, we qualitatively compare with the text-based NeRF stylization approach [45]. The NeRF-Art method [45] mainly performs the stylized generation of avatars, and we employ the NeRF-Art method [45] to generate stylized scenes for comparison with our method. As shown in rows 2 and 4 of Fig. 9, our method can generate more abundant and detailed colors than the NeRF-Art method [45] because we can precisely adjust the stylized transfer direction. In particular, as shown in the green box in the second row of Fig. 9, our method generates a picture of "Pixar's Finding Nemo" in the TV.

In Fig. 10, we qualitatively compare the results of the ARF method [1], the IS-NeRF method [2] and the StylizedNeRF method [40] of novel views. Under the condition of similar reconstruction geometric accuracy, the results show that our method has reasonable style variation based on depth of field, rather than simply aligning all of the scene's object styles with the target stylized features. Further results are given in the supplementary document.

5.2.2. *Quantitative Results*

Consistency Measurement. We measure the short-term and long-term consistencies using the warped LPIPS metric [57]. The score between frames I_i and I_j is formulated as:

$$S\left(I_i, I_j\right) = \text{LPIPS}\left(I_i, \mathcal{M}_{i,j}, \mathcal{W}_{i,j}\left(I_j\right)\right), \tag{14}$$

where $\mathcal{W}$ is the warping function and $\mathcal{M}$ is the warping mask. When measuring the long-term and short-term consistencies, only the pixels within the mask $\mathcal{M}_{i,j}$ are calculated.

We evaluate on 5 scenes from the Tanks and Temples dataset [55] and llff dataset [56], where each scene is compared with image pairs selected from stylized frames. For each pair, we evaluate the images stylized with 6 style classes, respectively. We perform short-term and long-term consistency measurements with view pairs at intervals (I_i, I_{i+2}) and (I_i, I_{i+8}). The comparison results of short-term and long-term consistencies are shown in Table 1 and Table 2. Our approach is obviously superior to other methods and maintains more consistent views while obtaining higher style transfer quality.

Table 1: Quantitative comparison of short-term consistency. We evaluate the consistency scores (lower is better) between the stylized images of the novel views at interval (I_i, I_{i+2}).

Method	Horns	Fern	Orchids	Playground	Truck	Average
ARF[1]	0.0634	0.1712	0.0754	0.0814	0.0773	0.09374
IS-NeRF[2]	0.0645	0.0484	0.1685	0.0607	0.1039	0.0892
StylizedNeRF[40]	0.0908	0.1731	0.0536	0.1211	0.2084	0.1294
Ours	0.0410	0.0365	0.0953	0.0513	0.0748	0.05978

Table 2: Quantitative comparison of long-term consistency. We calculate the consistency scores(the lower the better) between stylized images in two distant novel views at interval (I_i, I_{i+8}).

Method	Horns	Fern	Orchids	Playground	Truck	Average
ARF [1]	0.2886	0.4739	0.3263	0.3320	0.3489	0.3539
IS-NeRF [2]	0.1669	0.1776	0.3137	0.2518	0.3383	0.2275
StylizedNeRF[40]	0.3450	0.4126	0.3113	0.3637	0.4602	0.3785
Ours	0.1502	0.1320	0.1993	0.1804	0.2593	0.165475

Stylized Effect Evaluation. We evaluate the stylization quality in comparison with the text-based NeRF-Art stylization method [45], employing the following metrics:

Fréchet Inception Distance(FID) [58] and ***Kernel Inception Distance (KID)*** [59]

Different from image-driven stylization methods that use target stylized images for stylized quality evaluation, text-driven stylization methods can perform stylization quality evaluation using FID and KID metrics. Stylized datasets are generated from the three scenes in Fig. 9, and the corresponding target stylized datasets are collected from the ArtBench art database [46] and the web. As shown in Table 3, our stylized results outperform the NeRF-Art stylization method [45] in both "Cubism Painting" and "Pixar 3D Style" scene.

Table 3: Quantitative comparison of KID and FID scores. We calculate the KID and FID scores (the smaller the better) between the stylized dataset and the target stylized dataset.

Scene	FID (NeRF-Art [45])	FID (Ours)	KID (NeRF-Art [45])	KID (Ours)
Cubism Painting	480.8813	444.9358	0.2072	0.1956
Pixar 3D Style	527.6206	527.0403	0.1798	0.13029
Colorful Galaxy	358.1642	417.5943	0.1693	0.227

User Study. A user study is performed to compare the stylization and content consistency of our approach with other baselines. We focus on the text-based NeRF stylization method [45] and image-driven NeRF stylization method (ARF [1], IS-NeRF [2], and StylizedNeRF method [40]).

We stylize a series of views of 3D scenes from the Tanks and Temples dataset [55], and llff dataset [56] using different methods [1], [2], [40], [45] and invite 60 participants (including 32 males and 28 females, aged between 18 and 50 years). First, we present participants with stylized images and videos synthesized by our method and other baselines, and then ask them to vote from the style quality and temporal consistency perspectives. Our method gets the best score in stylization, and it is comparable to methods [1], [45] in terms of multi-view

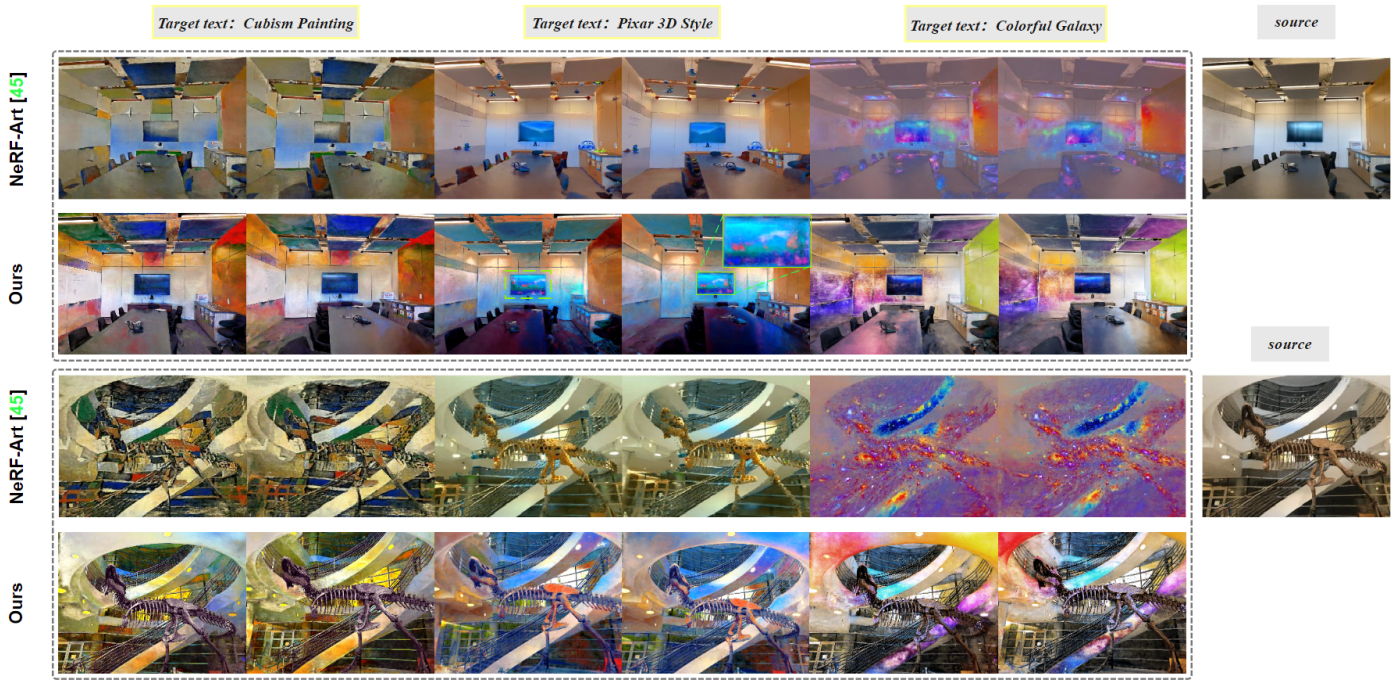

Fig. 9: Qualitative comparison with the text-based NeRF-Art method [45].

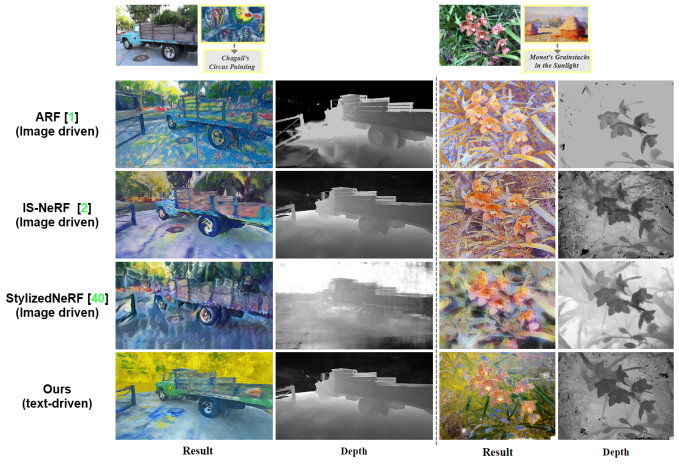

Fig. 10: Qualitative comparison with ARF method [1], IS-NeRF method [2] and StylizedNeRF method [40]. We compare the results of the novel views on the Tanks and Temples dataset [55] and llff dataset [56].

consistency. Although the ARF method [1] is more concerned with generating stylized details and strokes, it is unable to constrain and combine local styles into a complete semantic style, especially where the scene has a significant depth of field. The IS-NeRF method [2] matches the VGG features of the rendered image with those of the stylized image by the Gram matrix of global statistics. However, The IS-NeRF method [2] is limited by the fact that the stylized loss cannot be calculated with full-size rendered images, so the detail and global style quality of the rendered images are impacted. The StylizedNeRF method [40] employs a mutual learning approach to gradually fuse 2D style features into 3D space without directly stylizing in 3D space, which can make the stylization results dependent on the effectiveness of mutual information between 2D style features and 3D space features. Therefore, some scenes of the StylizedNeRF method [40] are not fully stylized. The NeRF-Art method [45] mainly performs avatar stylization, which can be guided by the text to stylize the scene. The NeRF-Art method [45] does not have enough subtle control over the semantics of object stylization in the scene, but our method can flexibly control the stylized transfer direction so that the scene produces abundant color and semantic content. In addition, the NeRF-Art method [45] does not specifically optimize the ray sampling capability of the NeRF, which can lead to insufficient sharpness of the generated images. The results are presented in the form of box line plots in in Fig. 11. Further results are given in the supplementary document.

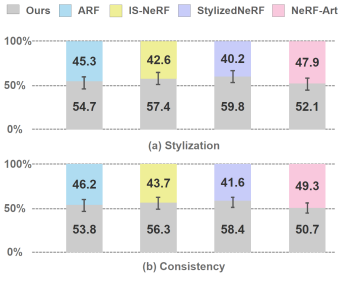

Fig. 11: User study results. We record user preferences in the form of box-line plots. Our results gain more favor in terms of both stylization and consistency quality.

5.3. Ablation Study

5.3.1. Effect of CLIP-HCL Loss on NeRF Style Transfer

To verify the effectiveness of our model using CLIP-HCL loss, we compare it with CLIP-space global loss presented in

[48] and CLIP-space direction loss presented in [12]. CLIP-space global loss directly calculates the cosine similarity between NeRF rendered image and target text in CLIP embedding space, independent of the initial image and domain. CLIP-space direction loss makes the direction from the source image to the rendered image parallel to the direction from the source text to the target text in the CLIP-space.

As shown in Fig. 12, CLIP-space global loss leads to different CLIP space orientations from all sources to target images [12], which can fall into local minima and eventually causes the network to fail to accurately transfer to the target domain. Although the directed CLIP loss relieves this problem by aligning the CLIP space orientations between the source and target text image pairs, a fixed CLIP space orientation is used to constrain the stylization of NeRF, which does not fully exploit the semantic information in the CLIP space and can lead to the degradation of image quality and object semantics. In contrast, CLIP-HCL allows NeRF rendering results to be more accurately aligned with the target text while preserving object geometric details of the 3D scene. Also from the FID and KID metrics in Fig. 12 combined with the target stylized image reference, it can be seen that the NeRF with CLIP-HCL loss has better stylization quality than CLIP-space global loss and CLIP-space direction loss.

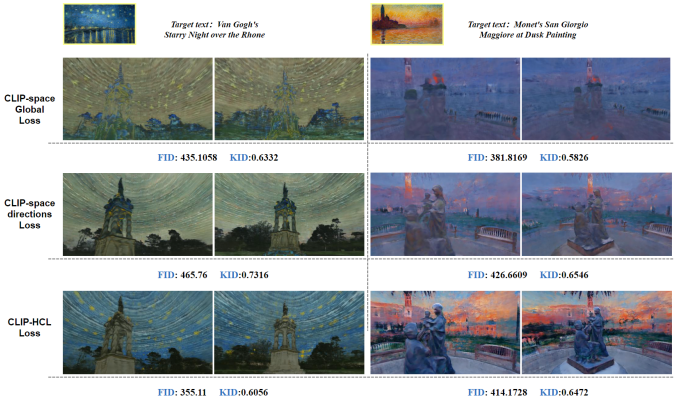

Fig. 12: Ablation study on the effects of CLIP loss. Also from the figure, it can be seen that CLIP-HCL loss enables the style learned by the network to be closer to the target style, while CLIP-space global loss presented in [48] causes the style learned by the network to deviate from the target. Although CLIP-space direction Loss presented in [12] endeavors to align to the target text, the image semantic information still degrades, leading to stylization of neutral descriptions. Below each image are the FID and KID values being calculated between this image and the target stylized image, which are used to evaluate the feature similarity and stylized quality of the stylized images to the target stylized images, the smaller the value the better.

5.3.2. *Effects of Gradient Accumulation*

To verify the effectiveness of GL-GA method, we compare it with the patch ray sample approach [2]. For each iteration, the patch ray sample method uses a single patch for sampling, resulting in a mismatch between the target style and the patch sampling content, and ultimately renders the images with inconsistent global styles [13]. Instead, our GL-GA sampling method conducts CLIP semantic loss calculated with global features stitched together by multiple patches, which can improve the accuracy of CLIP-HCL loss. This ensures the global consistency of the rendering style and the integrity of the geometric features of the scene objects. As shown in Fig. 13, the first row uses patch ray sampling. We use green boxes to highlight problems with incomplete geometric features of the object and blue boxes to mark image style discontinuity, which are due to the inconsistency of the global style. In the second row, our approach produces the results that are more complete in terms of image and semantics, and more closely aligned to the target style. The reason for this is that the global features spliced by patches can provide complete semantics, prompting CLIP-HCL to precisely guide the network for style transfer. Meanwhile, it can be seen from the FID and KID metrics in Fig. 13 that the NeRF with GL-GA sampling has better stylization quality than that with patch sampling.

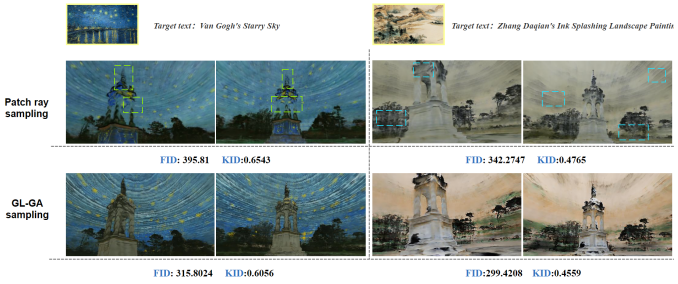

Fig. 13: Ablation study on the effects of GL-GA method. We compare the GL-GA method with the patch ray sample approach [2]. Our GL-GA method ensures the integrity of the object geometric features and the consistency of the global style. Below each image are the FID and KID values being calculated between this image and the target stylized image, which are used to evaluate the feature similarity and stylized quality of the stylized images to the target stylized images, the smaller the value the better.

5.3.3. *The Impact of Hyperparameter λ in the HCL loss*

To verify the impact of the hyperparameter λ in the HCL loss, we set $\lambda \in \{-1, -0.5, 0, 0.5, 1\}$. As shown in Fig. 14, the images present the effect of color degradation when the stylized direction is fully controlled by the negative samples $\lambda = -1$, due to the description of the neutral semantics of the negative samples. In contrast, when the stylized direction is completely controlled by the positive sample ($\lambda = 1$), the images present an over-edited effect, due to overfitting the single direction of the positive sample. When $\lambda = 0$, the images present a normal effect, owing to the stylized direction being between the positive and negative decision plane.

5.3.4. *Impact of semantic similarity searcher*

Firstly, the semantic similarity searcher (SS) helps to improve the accuracy of stylized generation with the ArtBench art database [46] under the condition of multiple domains. For accurate evaluation, we utilize a single-domain stylized model. From Fig. 15, it can be seen that in the single-domain condition, the semantic similarity searcher (SS) can enrich the color of the image because the target text is weighted with the semantic vector of nearest neighbors.

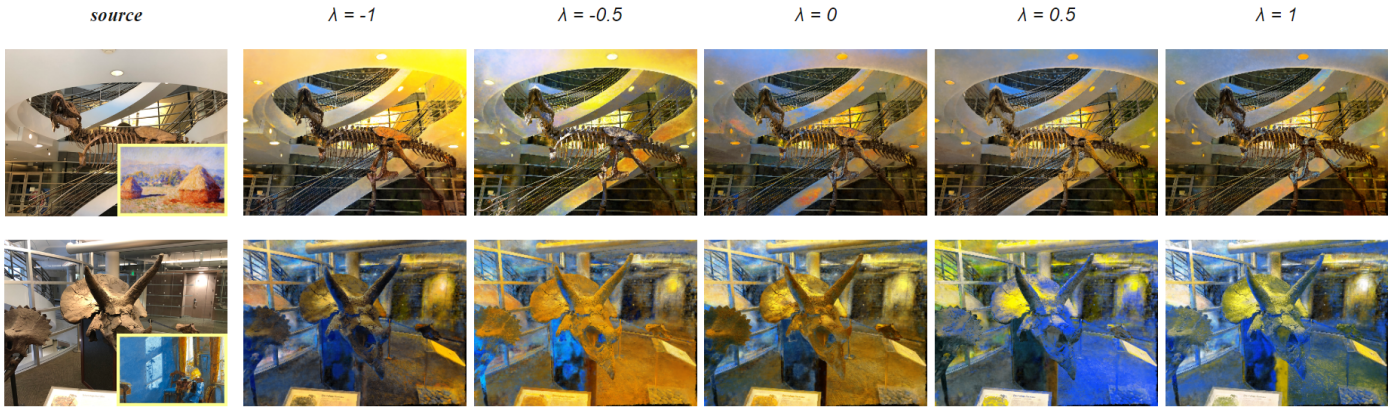

Fig. 14: Ablation study on the impact of Hyperparameter λ in the *HCL* loss. We compare the impact of various λ value on the stylized transfer direction. For each row in the image, the further to the left, the closer the stylized transfer direction is to the negative sample direction, and the further to the right, the closer it is to the positive sample direction.

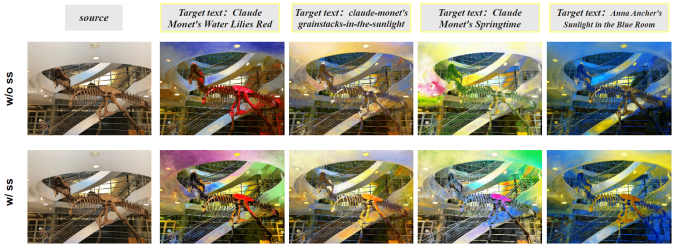

Fig. 15: Ablation study on the impact of semantic similarity searcher.

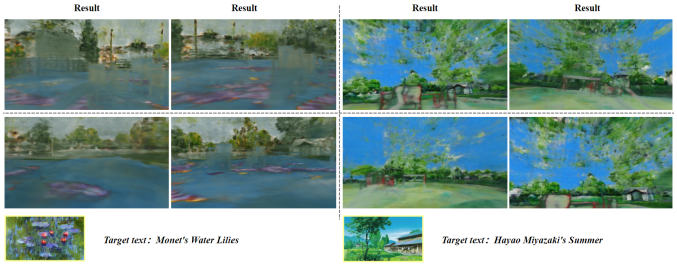

Fig. 16: Failure cases. As shown in columns 1, 2 of the figure, the semantics of the text image pairs are not precisely aligned, resulting in tonal deviations. In columns 3, 4, the stylized content bias is due to the fact that the stylized objects cannot be specified.

6. Discussion and limitations

Although our model generates promising results in most cases, some results do not meet expectations because the stylized object of the scene cannot be specified. Fig. 16 shows some of the undesirable examples generated by our model. Our model can generate style transfer results that match the target text. However, limited by the alignment accuracy of text-image pairs in the CLIP space, some rendered images may not be precisely migrated to the target domain, resulting in style transfer bias. Moreover, under a single-domain condition, the stylized generation will have better quality when the target text is a well-known painting, especially when it is collected in the ArtBench artwork database [46]. In the future, we will investigate joint data augmentation of multimodal vision-language models to enhance the models' capabilities and thus improve the accuracy of style transfer.

7. Conclusion

We propose an novel approach for stylized representation of 3D scenes. A novel 3D stylization framework based on semantic contrastive learning is presented, which comprehensively exploits the spatial semantic information of CLIP to precisely guide the stylized representation of NeRF. Specifically, A CLIP-based semantic contrastive estimation loss is designed for more accurate style transfer, which includes two aspects: First, the global style inconsistency caused by the NeRF patch ray sampling method is avoided by aligning the global style of the scene with the semantics of the target text in the CLIP space. Second, stylization of neutral descriptions due to the semantic averaging of CLIP is corrected by maximizing (resp. minimizing) the mutual information between positive (resp. negative) pairs. In addition, we design a GL-GA approach for NeRF ray sampling, which fully optimizes the NeRF sampling and rendering process and improves the semantic contrastive estimation loss accuracy. Experimental results indicate that our method outperforms state-of-the-art methods in terms of rendering style uniformity. Moreover, when transferring styles in multiple domains, our method generates new views that match the completely new style of the target text. Prospectively, we will exploit a unified NeRF stylization solution applicable to both indoor, outdoor and dynamic scenes and adopt Jittor model [60] to improve the training efficiency of the solution.

CRediT authorship contribution statement

Yi Wang: Methodology, Software. **Jing-Song Cheng**: Methodology, Writing. **Qiao Feng**: Conceptualization. **Wen-Yuan Tao**: Conceptualization. **Yu-Kun Lai**: Writing, Project administration. **Kun Li**: Writing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is freely available and will be automatically downloaded together with the software.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62171317 and 62122058). We are grateful to Associate Editor and anonymous reviewers for their help in improving this paper.

References

[1] Zhang, K, Kolkin, N, Bi, S, Luan, F, Xu, Z, Shechtman, E, et al. Arf: Artistic radiance fields. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI. 2022, p. 717–733.

[2] Chiang, PZ, Tsai, MS, Tseng, HY, Lai, WS, Chiu, WC. Stylizing 3D scene via implicit representation and hypernetwork. In: IEEE/CVF Winter Conference on Applications of Computer Vision. 2022, p. 1475–1484.

[3] Cao, X, Wang, W, Nagao, K, Nakamura, R. PSNet: A style transfer network for point cloud stylization on geometry and color. In: IEEE/CVF Winter Conference on Applications of Computer Vision. 2020, p. 3337–3345.

[4] Riegler, G, Koltun, V. Stable view synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 12216–12225.

[5] Hauptfleisch, F, Texler, O, Texler, A, Krivánek, J, Sỳkora, D. Styleprop: real-time example-based stylization of 3d models. In: Computer Graphics Forum; vol. 39. Wiley Online Library; 2020, p. 575–586.

[6] Hedman, P, Philip, J, Price, T, Frahm, JM, Drettakis, G, Brostow, G. Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (TOG) 2018;37(6):1–15.

[7] Barron, JT, Mildenhall, B, Tancik, M, Hedman, P, Martin-Brualla, R, Srinivasan, PP. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 5855–5864.

[8] Martin-Brualla, R, Radwan, N, Sajjadi, MS, Barron, JT, Dosovitskiy, A, Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 7210–7219.

[9] Liu, S, Zhang, X, Zhang, Z, Zhang, R, Zhu, JY, Russell, B. Editing conditional radiance fields. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 5773–5783.

[10] Mildenhall, B, Srinivasan, PP, Tancik, M, Barron, JT, Ramamoorthi, R, Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 2021;65(1):99–106.

[11] Radford, A, Kim, JW, Hallacy, C, Ramesh, A, Goh, G, Agarwal, S, et al. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PMLR; 2021, p. 8748–8763.

[12] Gal, R, Patashnik, O, Maron, H, Chechik, G, Cohen-Or, D. Stylegannada: Clip-guided domain adaptation of image generators. arXiv preprint arXiv:210800946 2021;.

[13] Huang, X, Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In: IEEE International Conference on Computer Vision. 2017, p. 1501–1510.

[14] Chen, A, Xu, Z, Zhao, F, Zhang, X, Xiang, F, Yu, J, et al. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 14124–14133.

[15] Greff, K, Belletti, F, Beyer, L, Doersch, C, Du, Y, Duckworth, D, et al. Kubric: A scalable dataset generator. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 3749–3761.

[16] Müller, T, Evans, A, Schied, C, Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) 2022;41(4):102:1–15.

[17] Suter, BW. The multilayer perceptron as an approximation to a bayes optimal discriminant function. IEEE Transactions on Neural Networks 1990;1(4):291.

[18] Qi, CR, Su, H, Mo, K, Guibas, LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 652–660.

[19] Gong, J, Ye, Z, Ma, L. Neighborhood co-occurrence modeling in 3d point cloud segmentation. Computational Visual Media 2022;8(2):303–315.

[20] Yang, B, Rosa, S, Markham, A, Trigoni, N, Wen, H. Dense 3d object reconstruction from a single depth view. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018;41(12):2820–2834.

[21] Wang, D, Cui, X, Chen, X, Zou, Z, Shi, T, Salcudean, S, et al. Multi-view 3d reconstruction with transformers. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 5722–5731.

[22] Gkioxari, G, Malik, J, Johnson, J. Mesh r-cnn. In: IEEE/CVF International Conference on Computer Vision. 2019, p. 9785–9795.

[23] Nash, C, Ganin, Y, Eslami, SA, Battaglia, P. Polygen: An autoregressive generative model of 3d meshes. In: International Conference on Machine Learning. PMLR; 2020, p. 7220–7229.

[24] Niemeyer, M, Geiger, A. Giraffe: Representing scenes as compositional generative neural feature fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 11453–11464.

[25] Gu, J, Liu, L, Wang, P, Theobalt, C. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. arXiv preprint arXiv:211008985 2021;.

[26] Chan, ER, Lin, CZ, Chan, MA, Nagano, K, Pan, B, De Mello, S, et al. Efficient geometry-aware 3D generative adversarial networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 16123–16133.

[27] Srinivasan, PP, Deng, B, Zhang, X, Tancik, M, Mildenhall, B, Barron, JT. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 7495–7504.

[28] Barron, XZPSBDTF Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:201007492 2020;.

[29] Boss, M, Braun, R, Jampani, V, Barron, JT, Liu, C, Lensch, H. NeRD: Neural reflectance decomposition from image collections. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 12684–12694.

[30] Dellaert, F, Yen-Chen, L. Neural volume rendering: Nerf and beyond. arXiv preprint arXiv:210105204 2020;.

[31] Wang, C, Chai, M, He, M, Chen, D, Liao, J. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 3835–3844.

[32] Schwarz, K, Liao, Y, Niemeyer, M, Geiger, A. Graf: Generative radiance fields for 3d-aware image synthesis. Advances in Neural Information Processing Systems 2020;33:20154–20166.

[33] Gatys, L, Ecker, AS, Bethge, M. Texture synthesis using convolutional neural networks. Advances in Neural Information Processing Systems 2015;28.

[34] Huang, H, Wang, H, Luo, W, Ma, L, Jiang, W, Zhu, X, et al. Real-time neural style transfer for videos. In: IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 783–791.

[35] Segu, M, Grinvald, M, Siegwart, R, Tombari, F. 3dsnet: Unsupervised shape-to-shape 3d style transfer. arXiv preprint arXiv:201113388 2020;.

[36] Hedman, P, Srinivasan, PP, Mildenhall, B, Barron, JT, Debevec, P. Baking neural radiance fields for real-time view synthesis. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 5875–5884.

[37] Kanazawa, A, Tulsiani, S, Efros, AA, Malik, J. Learning category-specific mesh reconstruction from image collections. In: European Conference on Computer Vision (ECCV). 2018, p. 371–386.

[38] Xiang, F, Xu, Z, Hasan, M, Hold-Geoffroy, Y, Sunkavalli, K, Su, H. Neutex: Neural texture mapping for volumetric neural rendering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 7119–7128.

[39] Höllein, L, Johnson, J, Nießner, M. Stylemesh: Style transfer for indoor

3d scene reconstructions. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 6198–6208.
[40] Huang, YH, He, Y, Yuan, YJ, Lai, YK, Gao, L. 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. 2022, p. 18342–18352.
[41] Crowson, K, Biderman, S, Kornis, D, Stander, D, Hallahan, E, Castricato, L, et al. Vqgan-clip: Open domain image generation and editing with natural language guidance. arXiv preprint arXiv:220408583 2022;.
[42] Esser, P, Rombach, R, Ommer, B. Taming transformers for high-resolution image synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, p. 12873–12883.
[43] Kim, G, Kwon, T, Ye, JC. Diffusionclip: Text-guided diffusion models for robust image manipulation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 2426–2435.
[44] Song, Y, Ermon, S. Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems 2019;32.
[45] Wang, C, Jiang, R, Chai, M, He, M, Chen, D, Liao, J. Nerf-art: Text-driven neural radiance fields stylization. IEEE Transactions on Visualization and Computer Graphics 2023;.
[46] Liao, P, Li, X, Liu, X, Keutzer, K. The artbench dataset: Benchmarking generative models with artworks. arXiv preprint arXiv:220611404 2022;.
[47] Huang, YH, He, Y, Yuan, YJ, Lai, YK, Gao, L. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 18342–18352.
[48] Patashnik, O, Wu, Z, Shechtman, E, Cohen-Or, D, Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 2085–2094.
[49] Zhan, F, Yu, Y, Wu, R, Zhang, J, Lu, S, Zhang, C. Marginal contrastive correspondence for guided image generation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 10663–10672.
[50] Zhan, F, Zhang, J, Yu, Y, Wu, R, Lu, S. Modulated contrast for versatile image synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 18280–18290.
[51] Zhang, J, Lu, S, Zhan, F, Yu, Y. Blind image super-resolution via contrastive representation learning. arXiv preprint arXiv:210700708 2021;.
[52] Park, T, Efros, AA, Zhang, R, Zhu, JY. Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision. Springer; 2020, p. 319–345.
[53] Robinson, J, Chuang, CY, Sra, S, Jegelka, S. Contrastive learning with hard negative samples. arXiv preprint arXiv:201004592 2020;.
[54] Kingma, DP, Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 2014;.
[55] Knapitsch, A, Park, J, Zhou, QY, Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) 2017;36(4):1–13.
[56] Mildenhall, B, Srinivasan, PP, Ortiz-Cayon, R, Kalantari, NK, Ramamoorthi, R, Ng, R, et al. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) 2019;38(4):1–14.
[57] Andonian, A, Park, T, Russell, B, Isola, P, Zhu, JY, Zhang, R. Contrastive feature loss for image prediction. In: IEEE/CVF International Conference on Computer Vision. 2021, p. 1934–1943.
[58] Heusel, M, Ramsauer, H, Unterthiner, T, Nessler, B, Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems 2017;30.
[59] Bińkowski, M, Sutherland, DJ, Arbel, M, Gretton, A. Demystifying mmd gans. arXiv preprint arXiv:180101401 2018;.
[60] Hu, SM, Liang, D, Yang, GY, Yang, GW, Zhou, WY. Jittor: a novel deep learning framework with meta-operators and unified graph execution. Science China Information Sciences 2020;63(12):1–21.