NeRF-Art: Text-Driven Neural Radiance Fields Stylization

Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao*

Abstract—As a powerful representation of 3D scenes, the neural radiance field (NeRF) enables high-quality novel view synthesis from multi-view images. Stylizing NeRF, however, remains challenging, especially in simulating a text-guided style with both the appearance and the geometry altered simultaneously. In this paper, we present NeRF-Art, a text-guided NeRF stylization approach that manipulates the style of a pre-trained NeRF model with a simple text prompt. Unlike previous approaches that either lack sufficient geometry deformations and texture details or require meshes to guide the stylization, our method can shift a 3D scene to the target style characterized by desired geometry and appearance variations without any mesh guidance. This is achieved by introducing a novel global-local contrastive learning strategy, combined with the directional constraint to simultaneously control both the trajectory and the strength of the target style. Moreover, we adopt a weight regularization method to effectively suppress cloudy artifacts and geometry noises which arise easily when the density field is transformed during geometry stylization. Through extensive experiments on various styles, we demonstrate that our method is effective and robust regarding both single-view stylization quality and cross-view consistency. The code and more results can be found on our project page: https://cassiepython.github.io/nerfart/.

Index Terms—Stylization, Neural Radiance Fields, CLIP

1 INTRODUCTION

Artistic works depict the world in various creative and imaginative styles, evolving along with human progress. While primarily driven by professionals, the generation of artistic content is now more accessible to average users than ever before, empowered by the recent research on visual artistic stylization. In the era of deep learning, technical advances are gradually reshaping how people create, consume, and share art, from real-time entertainment to concept design. Ever since neural style transfer [1], [2], [3], [4], [5] shows the potential of encoding and changing visual styles with deep neural networks, a significant amount of effort has been devoted to effectively and efficiently migrating the style of an arbitrary image [1], [6], [7], [8] or a specific domain [9], [10] to the content image. Despite the impressive results, these methods are limited to stylizing a single view input captured by the content image.

Motivated by the increasing demand for 3D asset creation, our goal is to stylize 3D content from multi-view input, in contrast to single-image stylization. In the domain of 3D representation, previous methods typically take explicit models (e.g., meshes [11], [12], [13], [14], [15], voxels [16], [17], and point clouds [18], [19]) followed by differentiable rendering for multi-view stylization. These methods enable intuitive control over the geometry but suffer from the limited capacity for modeling and rendering complex scenes. Recently, the implicit representation of neural radiance field (NeRF) [20], [21], [22], [23], [24] significantly improves the quality of novel view synthesis and thus satisfies our needs for a general representation of various scenes and objects. However, while enjoying the superior scene reconstruction quality of NeRF, the curse of its highly implicit volumetric representation of appearance and geometry, parameterized and entangled by dense MLP networks, makes NeRF more challenging to stylize through jointly transforming the encoded color and shape.

Very recently, pioneering NeRF stylization works [25], [26] have made exhilarating progress on appearance style transfer of 3D scenes. However, their style guidance is limited to image reference, which, although being adopted as one common way to specify the target style, is not always a perfect solution for every scenario—obtaining appropriate style images that both reflect the target style and match the source content might not be easy or even possible in many cases. Therefore, finding another simple, natural, and expressive form of guidance becomes an attractive idea. Thanks to the parallel advances in language-vision models, stylization with natural language is no longer a fantasy. As demonstrated by recent text-guided stylization works [27], [28], [29], [30], compared to image-guided approaches, short text prompts provide 1) an extremely intuitive and user-friendly way to specify styles, 2) a flexible control over various styles from abstract ones like a certain concept to very concrete ones like a famous painting or character, and 3) a view-independent representation that is free from content alignment and naturally benefits cross-view consistency.

Yet, with the existing approaches, it is still challenging to stylize the implicit representation of NeRF via a simple
Our NeRF-Art stylizes a pre-trained NeRF to match the desired style described by a text prompt. It modulates not only the appearance but also the geometry of NeRF.

In this work, we propose NeRF-Art, a new text-driven NeRF stylization method. Given a pre-trained NeRF model and a single text prompt, our method enables consistent novel view synthesis with both appearance and geometry transformed, adhering to the specified style. This is achieved by combining the recent large-scale Language-Vision model (i.e., CLIP) with NeRF, which is non-trivial due to several challenges. Directly applying the supervision from CLIP to NeRF by constraining the similarity between the rendered views and the text in the embedding space as [27] is insufficient to ensure the desired style strength. To tackle this problem, we design a CLIP-based contrastive loss to properly strengthen the stylization, by bringing the results closer to the target style and farther away from other styles pre-defined as negative samples. To further ensure the uniformity of the style over the whole scene, we extend our contrastive constraint to a hybrid global-local framework to cover both global structures and local details. To support geometry stylization jointly with appearance, we relax the constraints on the density of the pre-trained NeRF and adopt a weight regularization to effectively reduce cloudy artifacts and geometry noises when altering the density field. In experiments, we first evaluate text description selection for stylization and then test our method on various styles and demonstrate text guidance’s effectiveness and flexibility for NeRF stylization. Furthermore, we conduct a user study to show that our method achieves the best visual-pleasing results compared to related methods. We also extract the mesh from the stylized NeRF to show the geometry modulation ability of our method and integrate it with different baselines to demonstrate the generalization ability of our method to various NeRF-like models.

2 Related Work

Neural Style Transfer on Images and Videos. Artistic image stylization is a long-standing research area. Traditional methods use handcrafted features to simulate styles [34], [35]. With the fast development of deep learning, neural networks have been applied to style transfer from either an arbitrary image [1], [6], [7], [8], [36], [37] or a specific domain [9], [10], [38], [39] and achieved impressive results. By enforcing temporal smoothness constraints defined on optical flows, neural style transfer has been successfully extended to videos [40], [41], [42]. However, both image and video stylization methods are restricted to the given views. Simply combining the neural style transfer and novel view synthesis methods without considering 3D geometry will lead to blurriness or view inconsistencies.

Neural Stylization on Explicit 3D Representations. With the increasing demand for 3D content, neural style transfer has been extended to explicit 3D representations. The work [43] first considers the cross-view disparity consistency and applies style transfer on stereoscopic images or videos. Later, considering the voxel is the most compatible representation for CNNs, SKPN [16] encodes volume using convolutional blocks and stylizes it by deep features extracted from...
In the reconstruction stage, our method first pre-trains the NeRF model $F_{rec}$ of the target scene from multi-view input with reconstruction loss $L_{rec}$. In the stylization stage, our method stylized NeRF model $F_{rec}$ to $F_{sty}$, guided by a text prompt $t_{tgt}$, using a combination of relative directional loss $L_{dir}$ and global-local contrastive loss $L_{con}$. Additionally, we include weight regularization loss $L_{reg}$ and perceptual loss $L_{per}$ to ensure the stylization is consistent with the reference style.

Neural Stylization on NeRF. To address the inherent limitations of explicit representations, implicit methods have recently received much attention. NeRF is a seminal one that is able to represent complex scenes by parameterizing the implicit function as MLP networks. A large number of follow-up works are presented to improve its efficiency, geometric stylization, and joint stylization. Another line of work uses point clouds as the 3D proxy to guarantee 3D consistency in stylizing novel views from either a single image or multiple frames. In these works, point-wise features extracted from pre-trained PointNet or GCN are stylized by feature transform algorithms, e.g., adaptive normalization, and then rendered to novel views. Despite the successes, these 3D stylization methods are difficult to generalize to complicated objects or scenes with dedicated structures, limited by the expressiveness of explicit 3D representations.

Text-Driven Stylization. Compared to image references, a natural language prompt is a more intuitive and user-friendly way to specify the style. Therefore, a current line of works shifted away from image reference towards text guidance, with the help of the pre-trained CLIP, which bridges texts and images by jointly learning a shared latent space. The pioneering work StyleGAN-NADA proposes a directional CLIP loss for transferring the pre-trained StyleGAN2 model to the target domain with the desired style described by a textual prompt. Hila et al. proposed a method for high-level feature transfer using a blending operator that combines StyleGAN generator and CLIP semantic encoder. However, these image-based methods may lead to inconsistencies when applied to stylizing multiple views. In the 3D world, Text2Mesh uses CLIP to guide the stylization of a given 3D mesh by learning a displacement map for geometry deformation and vertex colors for texture stylization. The contemporary work AvatarCLIP further supports driving a stylized human mesh using natural languages. Despite their success, these methods are limited
to mesh input. In contrast, our method is able to stylize 3D scenes with better visual quality and view consistency without any mesh input.

3 Overview

As illustrated in Fig. 2, our approach is simply decomposed into reconstruction and stylization stages. In what follows, after briefly reviewing our 3D photographic representation with NeRF (§ 3.1), we put the focus on introducing our text-guided stylization method. Specifically, we first formulate the directional CLIP loss for stylization, which leverages the power of the pre-trained Language-Vision model (§ 4.1). After optimizing the reconstructed NeRF avoids artifacts in the final NeRF stylization that 1) preserves the original content from being scenes with text controls.

3.1 Preliminary on NeRF Scene Representation

We take NeRF as our 3D scene representation, which defines a continuous volumetric field as implicit functions, parameterized by MLP networks \( \mathcal{F} \). Given a single spatial coordinate \( x = (x, y, z) \) and its corresponding view direction \( d = (\phi, \theta) \), the network predicts the density \( \sigma \) and viewpoint-dependent radiance \( c = (r, g, b) \), leading to the final color \( C(r) \) of the camera ray \( r(t) = o + td \) by accumulating \( K \) sample points along it, given the target view:

\[
C(r) = \sum_{k=1}^{K} T_k(1 - \omega_k)c_k,
\]

where \( \omega_k = \exp(-\sigma_k(d_{k+1} - d_k)) \) represents the transmittance of the ray segment \( (k, k + 1) \) and \( T_k = \prod_{i=1}^{k-1} \omega_i \) is the accumulated transmittance from the origin to the sample \( k \).

To train NeRF from a set of multi-view photos, a simple supervised reconstruction loss is adopted between the ground-truth pixel colors \( C(r) \) from the training view and the NeRF prediction \( C(r) \):

\[
\mathcal{L}_{\text{rec}} = \sum_i \| C(r) - \hat{C}(r) \|^2_2.
\]

4 Text-Guided NeRF Stylization

After optimizing the reconstructed NeRF model \( \mathcal{F}_{\text{rc}} \), from the multi-view input (§ 3.1), our goal is to train a stylized NeRF model \( \mathcal{F}_{\text{styl}} \) which satisfies the style control of the target text prompt \( t_{tgt} \) while preserving the content from \( \mathcal{F}_{\text{rc}} \) (Fig. 2).

The CLIP model aligns the semantics of image and text in a joint embedding space, by utilizing the image encoder \( \hat{E}_t(\cdot) \) and the text encoder \( \hat{E}_t(\cdot) \). The semantic power of CLIP bridges the gap between natural language prompts and synthesized image pixels, making it possible to stylize NeRF scenes with text controls.

However, even with the powerful embedding space of CLIP, it remains challenging to achieve text-guided NeRF stylization that 1) preserves the original content from being washed away by the new style, 2) reaches the target style with proper strength that satisfies the semantics of the input text prompt, and 3) maintains cross-view consistency and avoids artifacts in the final NeRF model.

4.1 Trajectory Control w/ Directional CLIP Loss

An intuitive strategy for text-guided NeRF stylization would be to enforce the trajectory of the stylization in the CLIP space with an absolute directional CLIP loss that measures the cosine similarity \( \langle \cdot, \cdot \rangle \) between the stylized NeRF rendering \( I_{tgt} \) and the target text prompt \( t_{tgt} \) (Fig. 3a):

\[
\mathcal{L}_{\text{dir}} = \sum_{I_{tgt}} [1 - \langle \hat{E}_t(I_{tgt}), \hat{E}_t(t_{tgt}) \rangle],
\]

which guides NeRF rendering with a global direction of the target text, not depending on any reference starting point. This loss is first designed in StyleCLIP [31] to guide face image editing and further extended to generative NeRF editing in CLIP-NeRF [31].

However, as observed in StyleGAN-NADA [27], this global loss could easily mode-collapse the generator and hurt the generation diversity of stylization. Therefore, a relative directional loss is proposed, which transfers the source image \( I_{src} \) to the target domain guided by the CLIP-space trajectory embedded by a pair of text prompts \( (t_{src}, t_{tgt}) \) instead of a single one (Fig. 3b). Here \( t_{src} \) indicates the text prompt selected from a pre-defined source text database that refers to natural face portraits (more details in § 5.1).

This relative directional CLIP loss for our NeRF stylization is defined as:

\[
\mathcal{L}_{\text{dir}} = \sum_{I_{tgt}} [1 - \langle \hat{E}_t(I_{tgt}) - \hat{E}_t(I_{src}), \hat{E}_t(t_{tgt}) - \hat{E}_t(t_{src}) \rangle].
\]

Different from the single-image setting of StyleGAN-NADA, here, the training target \( I_{tgt} \) stands for an arbitrarily sampled view rendered by the stylized NeRF of the same scene, and the source image \( I_{src} \) is produced by the pre-trained NeRF model and shares the identical view as \( I_{tgt} \). We will follow this convention hereinafter.

4.2 Strength Control w/ Global Contrastive Learning

As the directional CLIP loss (Equation (4)) works by measuring the similarity between the normalized unit directions of the embedded vectors, it can enforce the relative stylization trajectory. However, it struggles with preserving enough stylization strength in altering the pre-trained NeRF model.

To address this issue, we propose a contrastive learning strategy to control the stylization strength (Fig. 3c). Specifically, in the framework of contrastive learning, with the rendered view \( I_{tgt} \) as the query target, we set positive samples to the target text prompt \( t_{tgt} \) with the desired style and construct negative samples \( I_{neg} \in T_{neg} \) by sampling a set of text prompts semantically irrelevant to \( I_{tgt} \). In general, our contrastive loss in the CLIP space is defined as:

\[
\mathcal{L}_{\text{con}} = -\sum_{I_{tgt}} \log \frac{\exp(v \cdot v^+/\tau)}{\exp(v \cdot v^+/\tau) + \sum_{v^-} \exp(v \cdot v^-/\tau)}
\]

where \( \{ v, v^+, v^- \} \) are query, positive sample, and negative sample, respectively, and temperature \( \tau \) is set to 0.07 in all our experiments. When defining the loss globally by treating the entire view \( I_{tgt} \) as the query anchor, we have the global contrastive loss \( \mathcal{L}_{\text{con}} \) with \( v = \hat{E}_t(I_{tgt}), v^+ = \hat{E}_t(t_{tgt}), v^- = \hat{E}_t(t_{neg}) \).

Ideally, this global contrastive loss cooperates with the directional CLIP loss, where the former defines the style
trajectory that aligns with the target text, and the latter, at the same time, ensures the proper stylization magnitude by pushing along the style trajectory. However, the global contrastive loss still has trouble achieving sufficient and uniform stylization on the entire NeRF scene, leading to excessive stylization on certain parts and insufficient stylization in other regions. This may be attributed to the fact that CLIP focuses more attention on local regions with distinguishable features than the entire scene. Thus, this global contrastive loss can deliver a small value even when the overall stylization is insufficient or non-uniform. To achieve a more sufficient and balanced stylization, enforced by a more locally-attended contrastive learning approach, inspired by PatchNCE loss \cite{yang2021patchnce}, we propose a complementary local contrastive loss $L_{\text{con}}$ which sets queries to random local patches $P_{\text{tgt}}$ cropped from $I_{\text{tgt}}$: \begin{align*}
\{ v = \hat{E}_i(P_{\text{tgt}}), \ v^+ = \hat{E}_i(t_{\text{tgt}}), \ v^- = \hat{E}_i(t_{\text{src}}) \}. 
\end{align*}

Overall, we combine the global and local terms as our final global-local contrastive loss:
\begin{equation}
L_{\text{con}}^g + L_{\text{con}}^l + \lambda_r L_{\text{reg}}, \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (8)
\end{equation}

4.3 Artifact Suppression w/ Weight Regularization

Our pipeline aims to change not only the color but also the density of the pre-trained NeRF to achieve a joint stylization of appearance and geometry. However, allowing the training process to alter the density may lead to cloud-like semi-transparent artifacts near the camera and geometry noises, due to backward gradient propagation’s prohibitively huge memory consumption. To address this issue, previous works either sample sparse rays to obtain coarse images or patches \cite{zhou2021nerf++, liu2022depth, zhao2021dense, zhao2021synnerf} or render all rays to low resolution and then upsample with CNN networks \cite{zeni2020approximate}. However, coarse renderings or patches lose style details and semantic structures, while upsampling harms the cross-view consistency. Instead, we adopt a much easier solution, which first renders all rays to obtain the whole image of an arbitrary view, calculates the stylization loss gradients in the forward process, and then back-propagates the gradients through NeRF at the patch level. This significantly reduces memory consumption and allows for high-resolution rendering for better stylization training. Similar techniques are also used in some related works, such as ARF \cite{zhou2021nerf++}.

4.4 Training Strategy

During training, we finetune the pre-trained NeRF model for stylization. The overall objective consists of three parts: text-guided stylization losses (including directional CLIP loss and global-local contrastive loss to control style trajectory and strength, respectively), content-preservation loss (we adopt VGG-based perceptual loss), and artifact suppression regularization loss:
\begin{equation}
\mathcal{L} = (L_{\text{dir}} + L_{\text{con}}^g + \lambda_p L_{\text{per}} + \lambda_r L_{\text{reg}}. 
\end{equation}

Here we define the perceptual loss $L_{\text{per}}$ between the original and stylized NeRF renderings on certain pre-defined VGG layers $\psi \in \Psi$:
\begin{equation}
L_{\text{per}} = \sum_{i_{\text{tgt}}} \sum_{\psi \in \Psi} \| \psi(I_{\text{tgt}}) - \psi(I_{\text{src}}) \|^2_2. 
\end{equation}

5 Experiments

5.1 Implementation Details

We implement our framework using PyTorch with Adam optimizer. In the reconstruction training stage, we sample 192
points for each ray and train our model for 6 epochs with the learning rate of 0.0005. While in the stylization training stage, we train our model for 4 epochs with the learning rate of 0.001. We set hyper-parameters $\lambda_g$, $\lambda_l$, $\lambda_p$, and $\lambda_r$ as 0.2, 0.1, 2.0, and 0.1, respectively. To construct the negative samples, we manually collect around 200 text descriptions from Pinterest website, describing various styles, like “Zombie”, “Tolkien elf”, and “Self-Portrait by Van Gogh”. We set the patch size as the 1/10 of the original input in the local contrastive loss. In our relative directional loss (Equation 4), $t_{src}$ is automatically selected using a source text database containing various types of texts, such as “human”, “human face”, and “portrait”, and so on. To choose the appropriate source text, we use the CLIP similarity metric to compare the similarity between the source renderings and each text in the database. The text with the highest similarity score is then chosen as $t_{src}$. Without loss of generality, we adopt VolSDF [86] as the basic NeRF model for stylization.

5.2 Data Collection

Three self-portrait datasets are gathered under an in-the-wild condition by asking three users to capture selfies video for around 10 seconds with the front-facing camera. We finally received six video clips in around 10 seconds. After collecting these video clips under different views and expressions, we extract 100 frames for each video clip using FFmpeg with 15 fps. Then these frames are resized to 270×480. Then we estimate camera poses for these frames using COLMAP [87] with rigid relative camera pose constraints. We suppose frames in a video share the same intrinsics. We also reconstruct a lady from the H3DS dataset [88]. We remove noise frames and obtain 31 sparse views. Moreover, we use the image size with 256×256 for stylization. We also adopt the Local Light Field Fusion (LLFF) dataset [89] to stylize non-face scenes. LLFF dataset is composed of forward-facing scenes, with around 20 to 60 images.

5.3 Text Evaluation

As CLIP [82] is sensitive to text prompts, we conduct a text description evaluation in Fig. 4. When a text description refers to a style in general, not anyone in particular, the stylization can be insufficient. For example, “Fauvism” only induces stylization around the mouth as it describes the

Fig. 4: Text Evaluation. We present descriptions at different detail levels for a specific style.

Fig. 5: Comparisons. Comparisons with the text-guided image stylization method StyleGAN-NADA [27].
general meaning, like artists “Henri Matisse” and “Kees van Dongen” or “Brutalist painting”. And the same observations when comparing “Chinese Painting” and “Chinese Ink Painting”. In contrast, when a text refers to a specific object or style, the language ambiguity will disappear. For example, “Lord Voldemort”, “Head of Lord Voldemort”, and “Head of Lord Voldemort in fantasy style” reveal similar stylization results.

We also see similar results concerning the Pixar style. In the interests of brevity, we use “Fauvism” to represent “painting, oil on canvas, Fauvism style” and “Vincent van Gogh” to represent “painting, oil on canvas, Vincent van Gogh self-portrait style” in other experiments. We also use the same prompt augmentation strategy for other painting styles, including “Edvard Munch” and “Fernando Botero”.

5.4 Comparisons

We compare with most related works following three categories: 1) Text-driven image stylization: StyleGAN-NADA [27]; 2) Text-driven mesh-based stylization: Text2Mesh [29] and AvatarCLIP [30]; and 3) Text-driven NeRF stylization: CLIP-NeRF [31] and DreamField [33]. To make fair comparisons with these methods, we adopt author-released codes and accommodate the input to each method as required. For StyleGAN-NADA, we follow its steps to first conduct a face alignment under the setting of FFHQ [30] and
then invert these faces using e4e [91] into latent codes, before inputting them to StyleGAN-NADA. We have also tried pSp [92] to invert latent codes but finally adopt e4e to obtain better stylization results. Per the authors’ advice, we train 600 iterations and sample faces presenting visual-pleasing stylized results. We place the final stylized faces back on the input images by inversing the face alignment process. As for Text2Mesh, the input mesh of one example (‘Lady’) is provided by the H3DS [88], while the input mesh of another example (‘Human’) is fetched from AvatarCLIP. Both meshes are normalized into -1 to 1, before inputting them to Text2Mesh. We follow the training setting of Text2Mesh in stylizing the person object to stylize ‘Lady’ and ‘Human’. We compare to DreamField and AvatarCLIP following the shape sculpting and texture generation process of AvatarCLIP. Similar to AvatarCLIP, we also adopt prompt augmenta-
StyleGAN-NADA local contrastive learning. We have proven that by introducing global-local contrastive losses, our method can better ensure the desired style strength in all examples by introducing global-local contrastive learning strategies to achieve sufficient and uniform stylization and designing a weight regularization to remove cloudy artifacts and geometry noises. Compared to text-driven image stylization.

**Comparisons to text-driven NeRF stylization.**

Text2Mesh also supports geometry deformation and texture stylization of a 3D model like ours. However, it assumes there exists a synergy between the input 3D geometry and the target prompt and is more likely to fail when stylizing a 3D mesh towards a less related prompt, such as "Pixar" for the 'Lady')'s model in Fig. 7. With carefully-designed loss constraints, ours is more robust to different prompts, either related to the 3D scenes or not. Moreover, limited by the expressivity of the mesh representation, Text2Mesh fails most runs and presents unstable stylization results, resulting in irregular deformations and indentations on the edge or surface. Authors of AvatarCLIP also report similar results when comparing to Text2Mesh. Similar to DreamField, AvatarCLIP adopts a random background augmentation to lead CLIP to focus on the foreground and prevent floating artifact generations. Nevertheless, this process requires view-consistent masks while ours does not. Moreover, AvatarCLIP adds an additional color network to constrain the general shape of the avatar as well as introducing random shading and lighting augmentations on the textured renderings to strengthen the stylization. Even with these augmentations, AvatarCLIP still fails to produce satisfying texture and geometry details. In contrast, our method reveals a fine-grained beard, detailed wrinkles of garments, and clearer face attributes. Noteworthy, our NeRF-Art supports stylizing in-the-wild faces, while AvatarCLIP requires a 3D mesh as input to conduct these augmentations. Finally, AvatarCLIP can still generate random bumps in the background and make the extracted surface noisy. This is because AvatarCLIP sampled sparse rays (112 × 112) to construct coarse renderings for CLIP constraints, due to the out-of-memory problem. We found worse results with more noise when reducing sampled ray numbers. In contrast, our method supports training stylization on all rays by imposing a memory-saving technique. In conclusion, NeRF-Art achieves better stylization using the proposed contrastive learning strategies without any mesh guidance.

**5.5 User Study**

To evaluate stylization quality from human perception, we conducted a user study. For each compared category, we used two subjects. For each subject, we selected 5 prompts from our text descriptions dataset and finally obtained 10 test cases for each category and 50 in total. For every test case, we showed one sample of input frames, the textual prompt, and the results of different methods in two views and random order. The participants were given unlimited time to select the best stylization results.
Fig. 10: **Geometry Evaluation.** Our method modulates the geometry and color simultaneously of a pre-trained NeRF to match the desired style described by a text prompt.

Fig. 11: **Ablations on Weight Regularization.** Cloudy artifacts near the corner or geometric noises are observed without the weight regularization loss.

Fig. 12: **User Study.** Our method consistently outperforms state-of-the-art text-guided stylization methods on user preference rates (%).

5.6 Ablation Study

*Why global-local contrastive learning?* A straightforward way to stylize NeRFs is to apply the directional CLIP loss proposed by StyleGAN-NADA [27] to the rendered views. Unfortunately, the directional CLIP loss can enforce the right stylization trajectory but struggles to reach a sufficient magnitude, as shown in the 2nd column of Fig. 9. This is because the loss only measures the directional similarity between the normalized embedded vectors but ignores their actual distances. In contrast, our global contrastive loss (3rd column of Fig. 9) can ensure the proper stylization magnitude by pushing it as close as possible to the target. However, the global contrastive loss still cannot guarantee a sufficient and uniform stylization of the whole scene. The stylization shows excess on certain parts and insufficiency on others, e.g., insufficient stylized faces and excessively stylized eyes in the “Tolkien Elf” example in the 3rd column of Fig. 9. This may attribute to the fact that CLIP focuses more attention on regions with distinguishable features than on other regions. Our local contrastive loss helps achieve more balanced stylized results by stylizing every local re-
gion of the scene (4th column of Fig. 9). However, this local contrastive loss without global information may produce excessive facial attributes, e.g., generating more eyes in the “White Walker” example and two left ears in the “Tolkien Elf” example. This attributes to insufficient semantics involved in a local patch. This problem can be avoided by adding the global contrastive loss at the same time.

By combining both global and local contrastive loss with the directional CLIP, our method successfully achieves uniform stylization with both correct stylization direction and sufficient magnitude (5th column of Fig. 9).

**Why weight regularization?** Altering the geometry of NeRF may potentially cause cloudy artifacts. In Fig. 11 we demonstrate that the weight regularization loss can suppress cloudy artifacts and geometric noises by encouraging a more concentrated density distribution for stylization.

### 5.7 Generalization Evaluation

We conduct a generalization evaluation on VolSDF and NeuS in Fig. 8 to evaluate NeRF-Art’s ability in adapting to different NeRF-like models. For NeuS, we adopt foreground segmentation using RVM [93] for better reconstructions and dilate the mask with two iterations of $3 \times 3$ kernel to allow for certain geometric variations. In Fig. 8, our method presents similar stylization results on VolSDF and NeuS, which demonstrates that our NeRF-Art has the ability to adapt to different NeRF-like models.

### 5.8 Geometry Evaluation

To evaluate whether the geometry will be correctly modulated in the stylization process, we show the geometry evaluation results in Fig. 10. We extract meshes using *Marching Cubes* [94] before and after the stylization for comparison and report results on two widely-used NeRF-like models VolSDF [56] and NeuS [55]. We clearly see geometry changes by comparison with the source mesh. For example, “Lord Voldemort” flattens the girl’s nose, “Tolkien Elf” sharpens the girl’s ears, and “Pixar” rounds the jaw. Moreover, we find the same observations on both VolSDF and NeuS. In summary, we conclude that our method can correctly modulate the geometry of NeRF to match the desired style.

### 5.9 Quantitative Analysis

<table>
<thead>
<tr>
<th>Source</th>
<th>StyleGAN-NADA</th>
<th>CLIP-NeRF</th>
<th>DreamFields</th>
<th>Text2Mesh</th>
<th>AvatarCLIP</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP</td>
<td>0.7931</td>
<td>0.7364</td>
<td>0.6992</td>
<td>0.6625</td>
<td>0.6675</td>
<td>0.7356</td>
</tr>
</tbody>
</table>

**TABLE 1: Image Quality Evaluation.** We compute the CLIP similarity between the stylized views and the target data, which shows that our method outperforms other methods.

**Image quality evaluation.** It is impractical to generate a large number of stylized images that are sufficient to reliably evaluate the FID scores of optimization-based methods, including AvatarCLIP, Text2Mesh, and our method. Instead, we utilize CLIP similarity as the evaluation metric to evaluate the quality of the stylized images. Specifically, we collect 50 images from the Internet based on the description of the target style and then calculate the CLIP similarity between the stylized renderings (30 views) and these collected images. As the baseline, we also calculate the CLIP similarity between the source renderings (30 views) and these collected images. We experiment on five vastly-different styles and report the average values in Table 1. Our method significantly outperforms the compared methods, indicating its potential to produce stylized renderings that are more faithful to target styles.

**TABLE 2: View Consistency Evaluation.** To assess the consistency of views before and after stylization, we use warped LPIPS and evaluate the results. We are unable to provide values for DreamFields and AvatarCLIP on LLFF scenes due to the fact that both methods are designed for objects with masks. We observe that NeRF-Art exhibits the least degradation in view consistency compared to other methods.

**View consistency evaluation.** We evaluate the view consistency using warped LPIPS. Specifically, we randomly render a pair of source and target views $V_i$ and $V_t$, with their respective poses $P_i$ and $P_t$. In addition, we calculate the depth of the source view as $d_i$. Then, we warp the pose of the source view $V_i$ from $P_i$ to $P_t$ using the depth $d_i$ and obtain the warped view $\tilde{V}_i$. Between $M \cdot V_i$ and $M \cdot \tilde{V}_i$, we calculate their perceptual similarity in LPIPS, where $M$ is the warping mask. We evaluate the view consistency of NeRF-Art against previous methods and the original NeRF model on seven cases, each using 30 pairs of views. As shown in Table 2, our NeRF-Art achieves the least degradation in view consistency compared to other methods.

### 6 Conclusion

In this paper, we present NeRF-Art, the text-guided NeRF stylization approach based on CLIP. Unlike existing approaches that require mesh guidance in the stylization process or trap in insufficient geometry deformations and texture details in stylization, ours modulate its geometry and appearance simultaneously to match the desired style and show visual-pleasing results of geometry deformations and texture details with only text guidance. To achieve it, we introduce a carefully-designed combination of the directional constraint to control the style trajectory and a novel global-local contrastive loss to enforce the proper style direction. Moreover, we propose a weight regularization strategy to alleviate the cloudy artifacts and geometry noises in deforming the geometry. Extensive experiments on real faces and general scenes show that our method is effective and robust in both stylization quality and view consistency.

**Limitations.** Despite the success in most cases, our method still has some limitations. First, some text prompts are linguistically ambiguous, like “Digital painting”, which describes a wide range of styles, including oil paintings, pencil sketches, 3D rendering images, cartoon drawings, etc. This ambiguity might confuse the CLIP and make the final result unexpected, as shown in Fig. 13. Semantically meaningless
words cause another kind of unexpected result. For example, if we combine the words “Mouth” and “Batman” as a prompt, the result unexpectedly puts a bat shape on the mouth, which may not be what the user desires. These are interesting problems worth exploring in the future. An additional limitation of our model is that it requires re-optimization whenever the target text is modified, which is a common shortcoming for optimization-based methods such as AvatarCLIP and DreamField. We plan to investigate a fast feedforward approach in our future research to address this limitation.

**References**


Can Wang is a Ph.D. student in the Department of Computer Science, City University of Hong Kong, HK, People’s Republic of China. He got his M.S. in Computer Science and Technology from University of Science and Technology of China in 2020. His current research interests include affective computing and image manipulating.

Ruixiang Jiang is a BSc. student at the Department of Computing, The Hong Kong Polytechnic University, HK, SAR, China. His research interests include Computer Graphics and Computer Vision, with a focus on neural rendering.

Mengming He is a Senior Research Scientist in the team of Creative Algorithms and Technology at Netflix. She obtained her Ph.D. degree from Computer Science & Engineering, HKUST in 2018, and her M.S. degree and B.E. degree from Zhejiang University in 2014 and 2011. Her research interests include Computer Graphics and Computer Vision, mainly focusing on computational photography and image/video processing.

Dongdong Chen is a principal researcher from Microsoft Research. He received his PhD degree under the joint phd program between University of Science and Technology of China and MSRA. His research interests mainly include style transfer, image restoration, low-level image processing, and general representation learning.
Jing Liao received the dual Ph.D. degrees from Zhejiang University and Hong Kong University of Science and Technology in 2014 and 2015 respectively. She was a researcher in MSRA from 2015 to 2018. She is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. Her research interests include image and video processing, computational photography, non-photorealistic rendering.