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

- * *Corresponding author.*
- *C. Wang and J. Liao are with Department of Computer Science, City University of Hong Kong. E-mail: cwang355-c@my.cityu.edu.hk, jingliao@cityu.edu.hk.*
- *R. Jiang is with Department of Computing, The Hong Kong Polytechnic University. E-mail: rui-x.jiang@connect.polyu.hk.*
- *M. Chai is with Snap Inc.. E-mail: cmlatsim@gmail.com.*
- *M. He is with Netflix. E-mail: hmm.lillian@gmail.com.*
- *D. Chen is with the Microsoft Cloud AI. E-mail: cddlyf@gmail.com.*

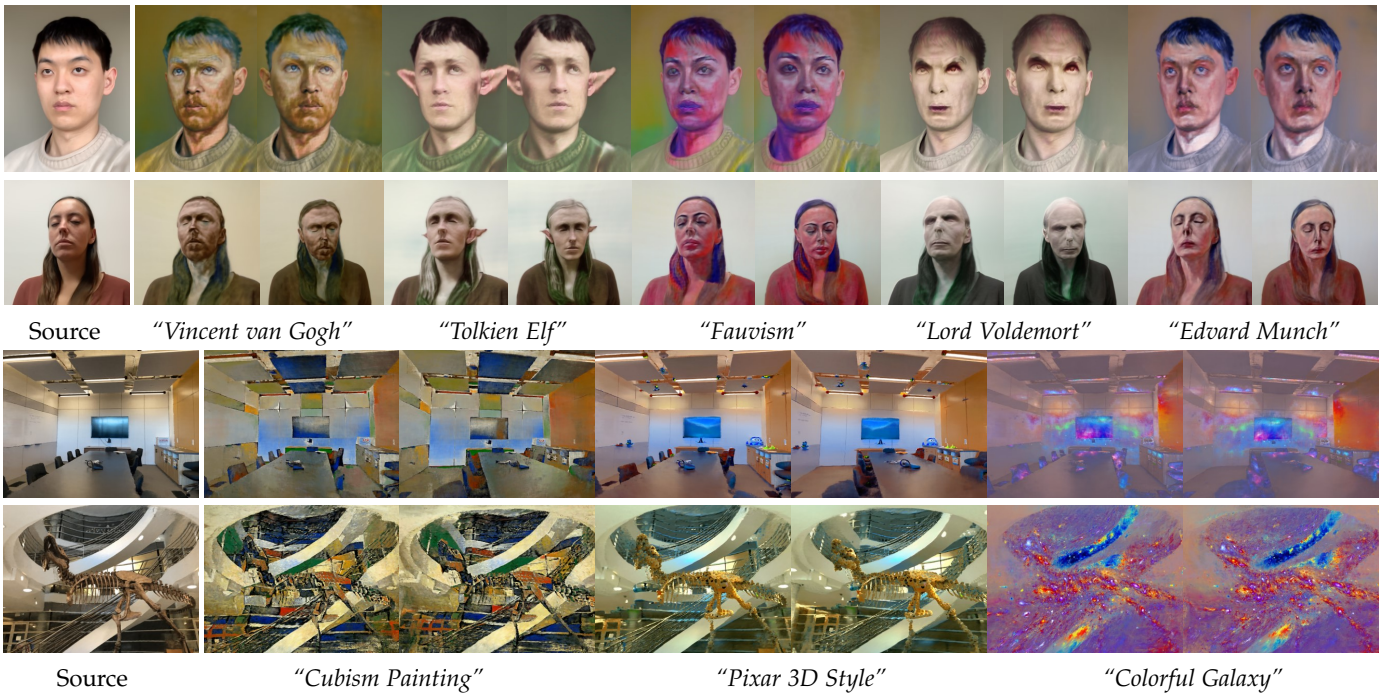


Fig. 1: ***NeRF-Art* Results.** Our *NeRF-Art* stylizes a pre-trained *NeRF* to match the desired style described by a text prompt. It modulates not only the appearance but also the geometry of *NeRF*.

text prompt. Learning a latent space helps constrain the geometry and texture modulations [31], but it is often data-dependent and laborious. Some efforts directly enforce style directions (Fig. 3) between the rendered views of *NeRF* and the text in the *CLIP* [32] embedding space. In addition, background augmentation [33] and mesh guidance [30] have been proposed to improve the geometry and texture modulations. However, they still suffer from insufficient geometry deformations and texture details.

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. In addition, 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 *CNN*s, *SKPN* [16] encodes volume using convolutional blocks and stylizes it by deep features extracted from

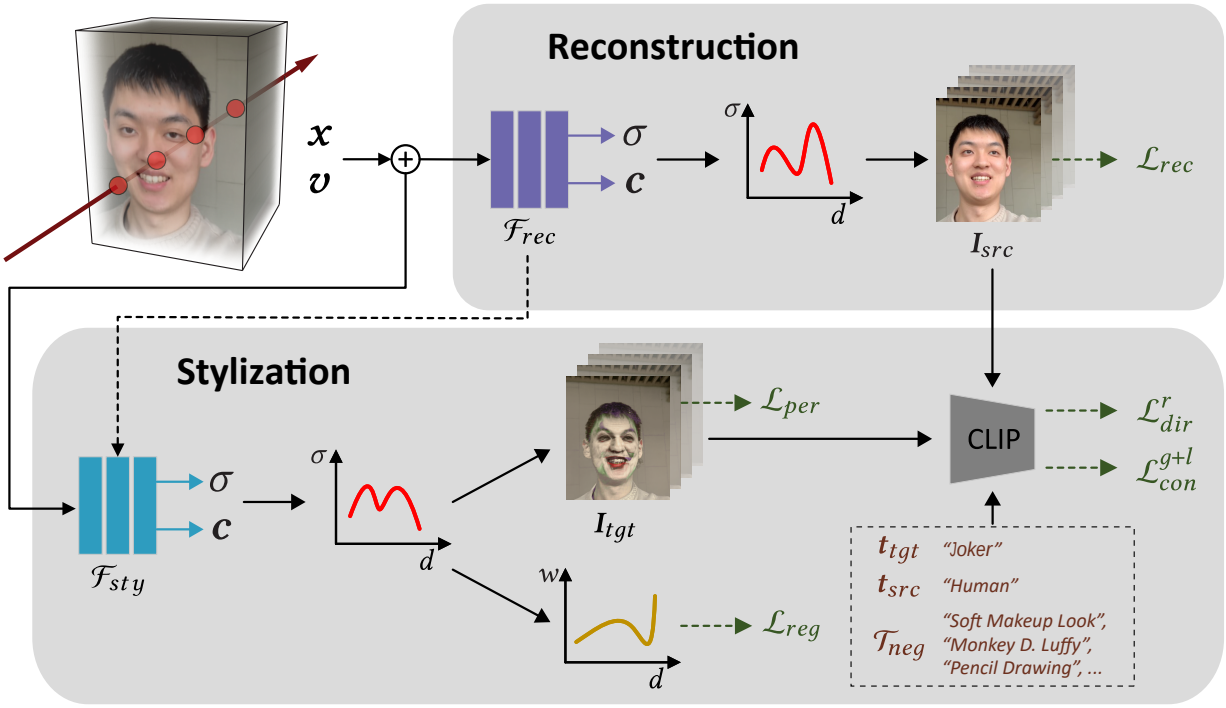


Fig. 2: ***NeRF-Art* Pipeline.** In the *reconstruction* stage, our method first pre-trains the *NeRF* model $\mathcal{F}_{rec}$ of the target scene from multi-view input with reconstruction loss $\mathcal{L}_{rec}$. In the *stylization* stage, our method stylized *NeRF* model $\mathcal{F}_{rec}$ to $\mathcal{F}_{sty}$, guided by a text prompt $\boldsymbol{t}_{tgt}$, using a combination of relative directional loss $\mathcal{L}^{r}_{dir}$ and global-local contrastive loss $\mathcal{L}^{g+l}_{con}$ in the *CLIP* embedding space, plus weight regularization loss $\mathcal{L}_{reg}$ and perceptual loss $\mathcal{L}_{per}$.

a reference image. As for mesh stylization, differential rendering allows for backpropagating style transfer objectives from rendered images to 3D meshes. According to whether the geometry or texture is allowed to be optimized, existing mesh style transfer methods achieve three different effects: texture stylization [12], [44], geometric stylization [45], and joint stylization [11], [46], [47]. Another line of work uses point clouds as the 3D proxy to guarantee 3D consistency in stylizing novel views from either a single image [48] or multiple frames [49]. In these works, point-wise features extracted from pre-trained *PointNet* [50] or *GCN* [51] 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.

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 [52], [53], [54], [55], [56], [57], quality [58], [59], [60], [61], controllablity [31], [62], [63], [64], and generalization [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76]. Inspired by the power of *NeRF*, three very recent works [25], [77], [78] adopt it for 3D stylization. They design the stylization network to predict color-related parameters in the *NeRF* model based on a reference style. And the stylization network is trained either by imposing the image style transfer losses [1], [78] on rendered views [25] or being supervised by a mutually learned image stylization network [77]. These works have achieved consistent results in novel-view stylization. However, their stylization is still restricted to appearance only because they do not adjust density parameters in the *NeRF* model. In contrast, our method supports both appearance and geometric stylization to better mimic the reference style. Moreover, they rely on reference images for stylization, while we seek to stylize the scenes via simple text prompts.

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* [32], which bridges texts and images by jointly learning a shared latent space. The pioneering work *StyleGAN-NADA* [27] proposes a directional *CLIP* loss for transferring the pre-trained *StyleGAN2* model [79] to the target domain with the desired style described by a textual prompt. Hila *et al.* [80] 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* [29] 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* [30] 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 photography 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). Then, we introduce our global-local contrastive learning framework to cope with the stylization strength issue of the directional *CLIP* loss (§ 4.2). Next, we introduce a weight regularization term to alleviate the cloudy artifacts caused and geometry noises by the stylization process (§ 4.3). Finally, we conclude this section with the overall training strategy of the entire pipeline (§ 4.4).

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 $\boldsymbol{x} = (x, y, z)$ and its corresponding view direction $\boldsymbol{d} = (\phi, \theta)$, the network predicts the density σ and view-dependent radiance $\boldsymbol{c} = (r, g, b)$, leading to the final color $C(\boldsymbol{r})$ of the camera ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ by accumulating K sample points along it, given the target view:

$$C(\boldsymbol{r}) = \sum\nolimits_{k=1}^{K} T_k(1 - \omega_k)\boldsymbol{c}_k, \tag{1}$$

where $\omega_k = \exp(-\sigma_k(d_{k+1} - d_k))$ represents the transmittance of the ray segment $(k, k+1)$ and $T_k = \prod_i^{k-1} \omega_i$ 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 $\hat{C}(\boldsymbol{r})$ from the training view and the *NeRF* prediction $C(\boldsymbol{r})$:

$$\mathcal{L}_{rec} = \sum\nolimits_{\boldsymbol{r}} \left\| C(\boldsymbol{r}) - \hat{C}(\boldsymbol{r}) \right\|_2^2. \tag{2}$$

4 Text-Guided *NeRF* Stylization

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

The *CLIP* model aligns the semantics of image and text in a joint embedding space, by utilizing the image encoder $\hat{\mathcal{E}}_i(\cdot)$ and the text encoder $\hat{\mathcal{E}}_t(\cdot)$. 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 $\boldsymbol{I}_{tgt}$ and the target text prompt $\boldsymbol{t}_{tgt}$ (Fig. 3(a)):

$$\mathcal{L}_{dir}^{a} = \sum\nolimits_{\boldsymbol{I}_{tgt}} \Big[1 - \langle \hat{\mathcal{E}}_i(\boldsymbol{I}_{tgt}), \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}) \rangle \Big], \tag{3}$$

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* [81] 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 $\boldsymbol{I}_{src}$ to the target domain guided by the *CLIP*-space trajectory embedded by a pair of text prompts $(\boldsymbol{t}_{src}, \boldsymbol{t}_{tgt})$ instead of a single one (Fig. 3(b)). Here $\boldsymbol{t}_{src}$ 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:

$$\mathcal{L}_{dir}^{r} = \sum\nolimits_{\boldsymbol{I}_{tgt}} \Big[1 - \langle \hat{\mathcal{E}}_i(\boldsymbol{I}_{tgt}) - \hat{\mathcal{E}}_i(\boldsymbol{I}_{src}), \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}) - \hat{\mathcal{E}}_t(\boldsymbol{t}_{src}) \rangle \Big]. \tag{4}$$

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

4.2 *Strength* Control w/ Glocal Contrastive Learning

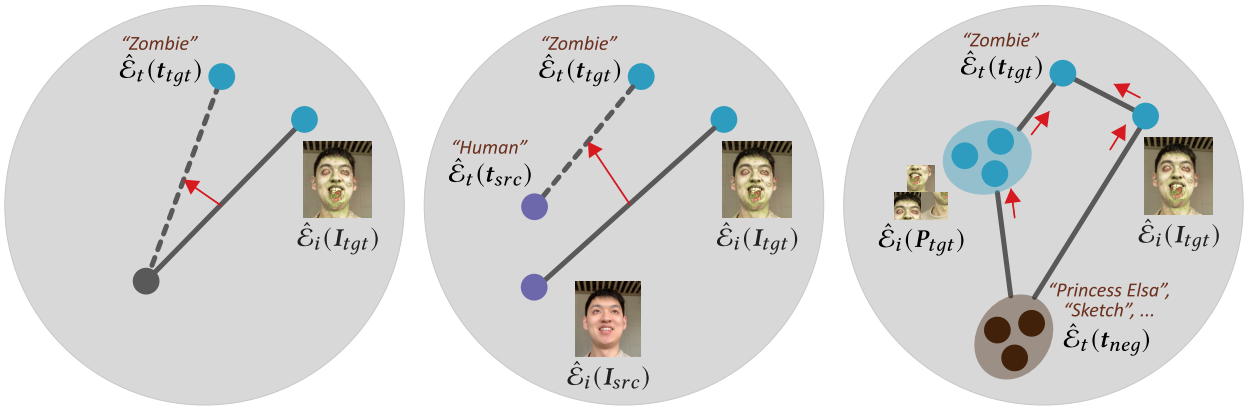
As the directional *CLIP* loss (Equation (4)) works by measuring the similarity between the normalized unit directions of the embedded vectors, it can enforce the relative stylization trajectory. However, it struggles with preserving enough stylization strength in altering the pre-trained *NeRF* model.

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

$$\mathcal{L}_{con} = -\sum\nolimits_{\boldsymbol{I}_{tgt}} \log \left[\frac{\exp(\boldsymbol{v} \cdot \boldsymbol{v}^{+}/\tau)}{\exp(\boldsymbol{v} \cdot \boldsymbol{v}^{+}/\tau) + \sum_{\boldsymbol{v}^{-}} \exp(\boldsymbol{v} \cdot \boldsymbol{v}^{-}/\tau)} \right], \tag{5}$$

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

Ideally, this global contrastive loss cooperates with the directional *CLIP* loss, where the former defines the style



(a) $\mathcal{L}^a_{dir}$ (b) $\mathcal{L}^r_{dir}$ (c) $\mathcal{L}^l_{con}$ & $\mathcal{L}^g_{con}$

Fig. 3: ***CLIP*-Guided Stylization Losses.** (a) The absolute directional loss; (b) The relative directional loss; (c) The global and local contrastive loss.

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 [82], we propose a complementary local contrastive loss $\mathcal{L}^l_{con}$ which sets queries to random local patches $\boldsymbol{P}_{tgt}$ cropped from $\boldsymbol{I}_{tgt}$: $\{\boldsymbol{v} = \hat{\mathcal{E}}_i(\boldsymbol{P}_{tgt}), \boldsymbol{v}^+ = \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}), \boldsymbol{v}^- = \hat{\mathcal{E}}_t(\boldsymbol{t}_{neg})\}$.

Overall, we combine the global and local terms as our final global-local contrastive loss:

$$\mathcal{L}^{g+l}_{con} = \lambda_g \mathcal{L}^g_{con} + \lambda_l \mathcal{L}^l_{con}. \tag{6}$$

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, even if the pre-trained *NeRF* is perfectly clean. To alleviate that, we adopt a weight regularization loss to suppress geometric noises and encourage a more concentrated density distribution that better resembles real-world scenes.

Based on our *NeRF* notations (Equation (1)), the weight of each ray sample is defined as the contribution to the final ray color: $w_k = T_k(1-\omega_k)$, where $\sum_k w_k \leq 1$. Similar to the distortion loss in *mip-NeRF 360* [83], the weight regularization loss is defined as:

$$\mathcal{L}_{reg} = \sum\nolimits_{\boldsymbol{I}_{tgt}} \sum\nolimits_{\boldsymbol{r}} \sum\nolimits_{(i,j)\in K} w_i w_j \left\| d_i - d_j \right\|, \tag{7}$$

where for each ray $\boldsymbol{r}$ of a randomly sampled view $\boldsymbol{I}_{tgt}$, pairs of samples (i, j) with distances $\|d_i - d_j\|$ are sampled. But different from *mip-NeRF 360* that optimizes the distances, we penalize those pairs with scattered large weights to suppress noise peeks and aggregate weights to the correct object surface.

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:

$$\mathcal{L} = (\mathcal{L}^r_{dir} + \mathcal{L}^{g+l}_{con}) + \lambda_p \mathcal{L}_{per} + \lambda_r \mathcal{L}_{reg}. \tag{8}$$

Here we define the perceptual loss $\mathcal{L}_{per}$ between the original and stylized *NeRF* renderings on certain pre-defined *VGG* layers $\psi \in \Psi$:

$$\mathcal{L}_{per} = \sum\nolimits_{\boldsymbol{I}_{tgt}} \sum\nolimits_{\psi\in\Psi} \left\| \psi(\boldsymbol{I}_{tgt}) - \psi(\boldsymbol{I}_{src}) \right\|_2^2. \tag{9}$$

It's practically infeasible to train stylization on all rays 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 [25], [30], [65], [84] or render all rays to low resolution and then upsample with *CNN* networks [85]. 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* [78].

5 Experiments

5.1 Implementation Details

We implement our framework using *PyTorch* with *Adam* optimizer. In the reconstruction training stage, we sample 192

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

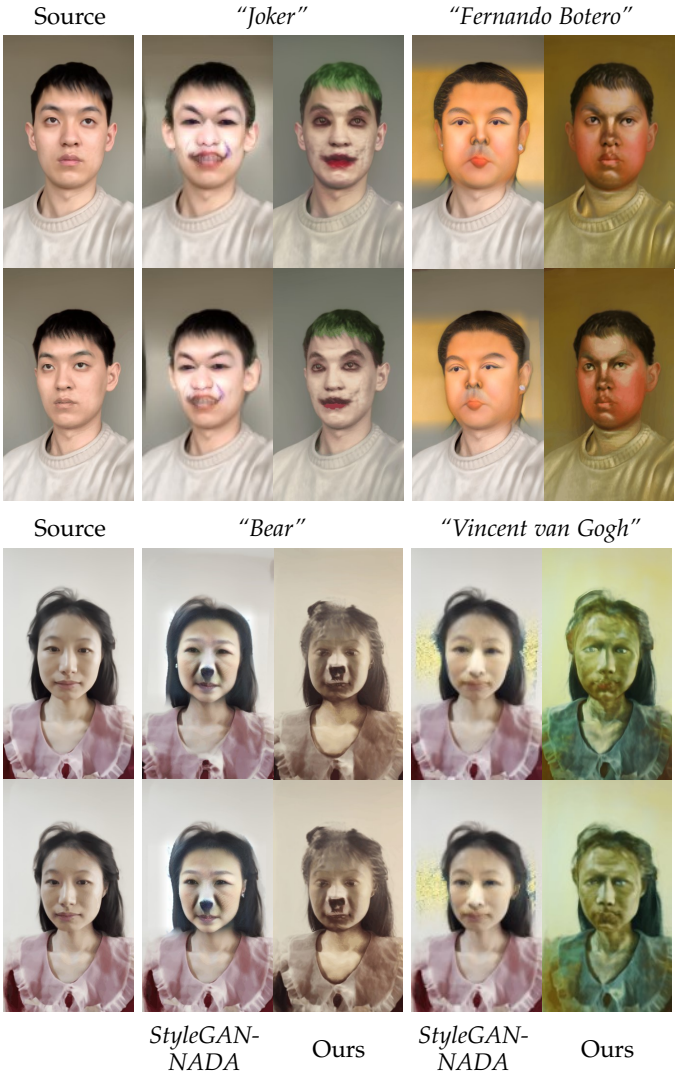


Fig. 5: **Comparisons.** Comparisons with the text-guided image stylization method *StyleGAN-NADA* [27].

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 λ_g, λ_l, λ_p, and λ_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), $\boldsymbol{t}_{src}$ 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 $\boldsymbol{t}_{src}$. 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

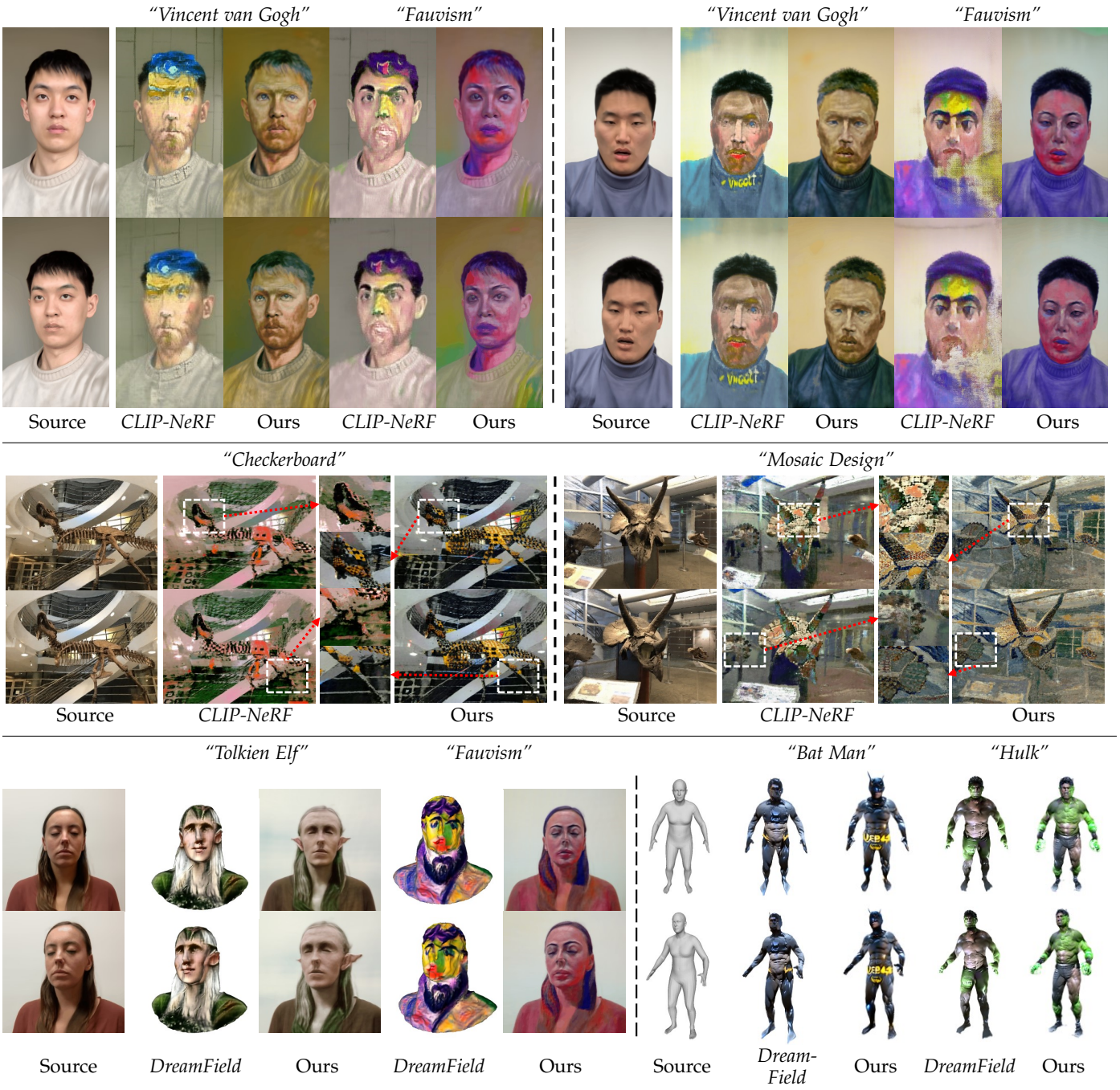


Fig. 6: **Comparisons.** Comparisons to text-guided *NeRF* stylization method *CLIP-NeRF* [31] and *DreamField* [33].

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* [90] and

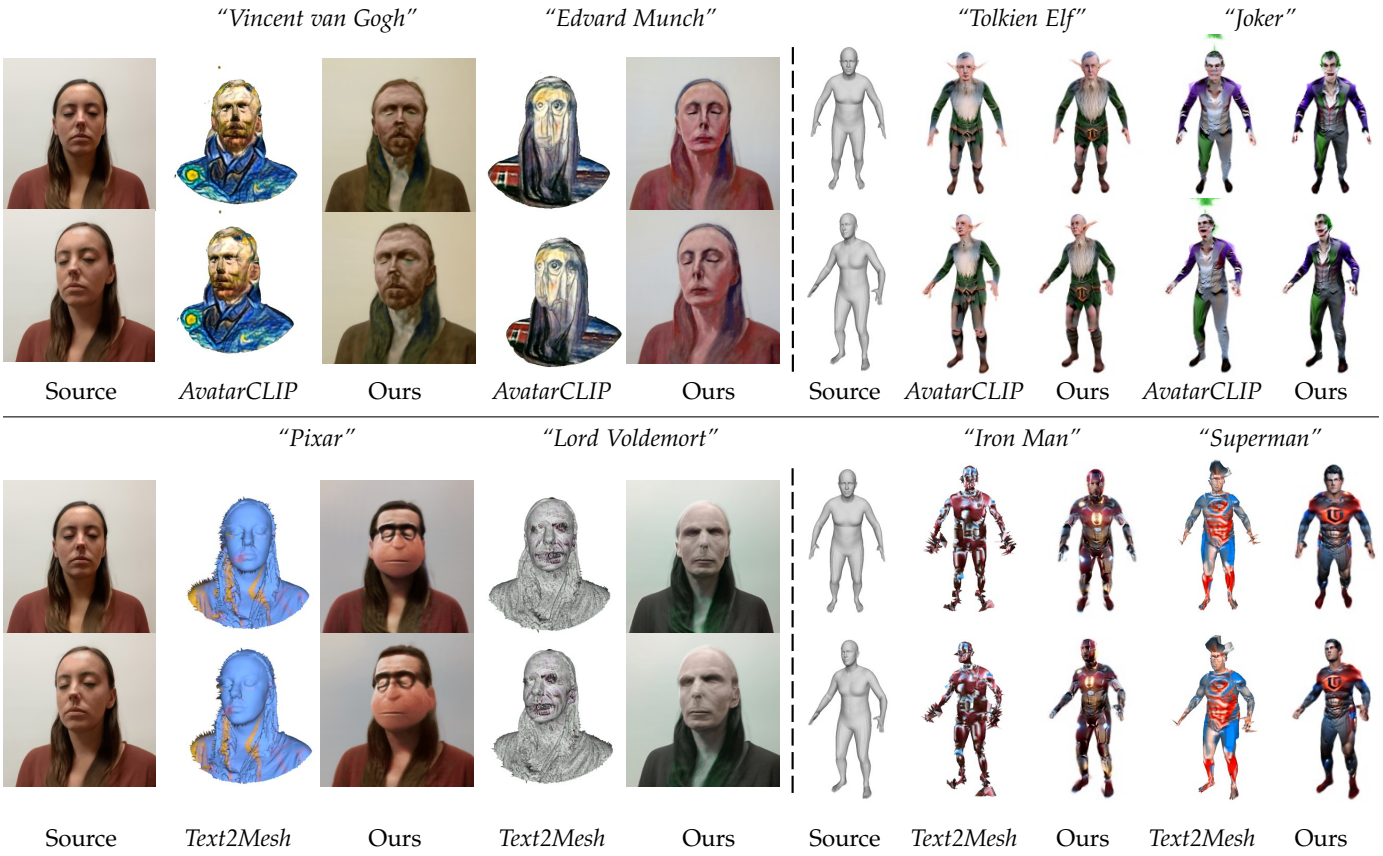


Fig. 7: **Comparisons.** Comparisons to text-guided mesh-based stylization method *Text2Mesh* [29] and *AvatarCLIP* [30].

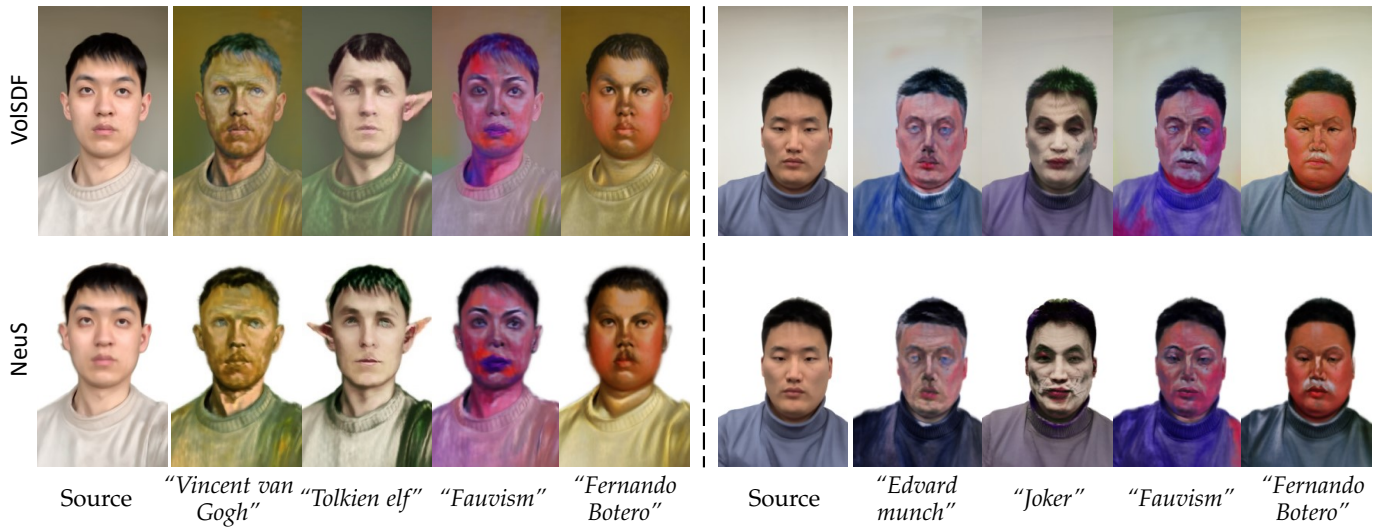


Fig. 8: **Generalization Evaluation.** Generalization evaluation on *VolSDF* and *NeuS*.

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-

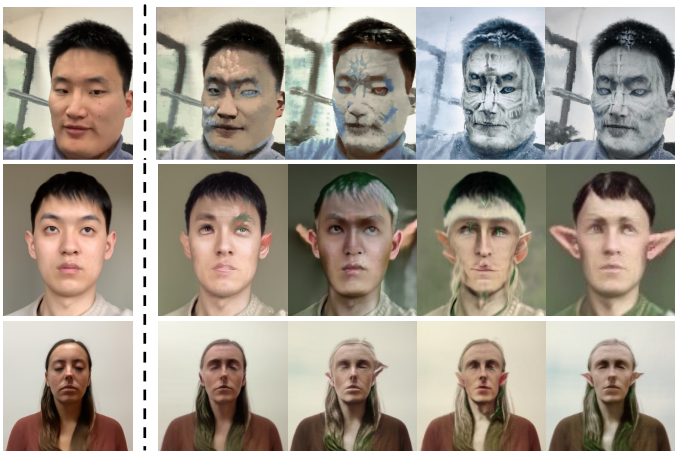


Fig. 9: **Ablations on *CLIP*-Guided Losses.** Without our global-local contrastive losses, the results suffer from insufficient or non-uniform stylization. The target prompts are *"White Walker"* and *"Tolkien Elf"* respectively.

tions when stylizing the *'Human'*. For example, we use text prompts including *"Tolkien Elf"*, *"the back of Tolkien Elf"*, and *"the face of Tolkien Elf"* for the detailed refinement.

The visual comparisons are demonstrated in Fig. 5, Fig. 6, and Fig. 7. For video results, please see the supplementary material.

Comparisons to text-driven image stylization. Compared to *StyleGAN-NADA*, our method can better ensure the desired style strength in all examples by introducing global-local contrastive learning. *StyleGAN-NADA* achieves visual-pleasing results on sampled faces but reflects a degradation for in-the-wild faces partly due to the latent code inversion. Moreover, as a 3D stylization method, ours can preserve view consistencies in the stylized results. In contrast, *StyleGAN-NADA* stylizes each view independently, thus introducing inconsistent shapes or textures to different views. This may lead to flickering artifacts when applied to video applications. Moreover, *StyleGAN-NADA* is less friendly to real faces as the input image has to be inverted back to the *StyleGAN* latent space before stylization, which will inevitably lead to some detail loss and identity change. Unlike it, *NeRF-Art* is not constrained by any latent space of pre-trained networks and does not need the inversion step.

Comparisons to text-driven NeRF stylization. Compared with *CLIP-NeRF*, our advantages are two-fold. First, *CLIP-NeRF* stylizes *NeRF* using the absolute directional loss, which does not put enough stylizations. Moreover, it suffers from uneven stylizations. For example, we only see enough stylizations on the nose and hair for style *"Fauvism"*, but the man's cheek has not been fully stylized. In contrast, we design a global-local contrastive learning strategy to ensure the desired style strength. Second, as no weight regularization is used in *CLIP-NeRF*, its results may appear as severe geometry noises. In contrast, our weight regularization suppresses geometric noises by encouraging a more concentrated density distribution. *DreamField* also adopts the absolute directional loss to stylize *NeRF*, which cannot guarantee sufficient and uniform stylization. *DreamField* adopts a random background augmentation to *CLIP*'s attention on the foreground, which requires view-consistent masks, while ours does not. Moreover, our method consistently outperforms *DreamField* in detailed cloth wrinkles, facial attributes, and fine-grained geometry deformations, like muscle shapes and antennas. In summary, our *NeRF-Art* outperforms these methods by proposing a contrastive learning technique to achieve sufficient and uniform stylization and designing a weight regularization to remove cloudy artifacts and geometry noises.

Comparisons to text-driven mesh-based 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, ours 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

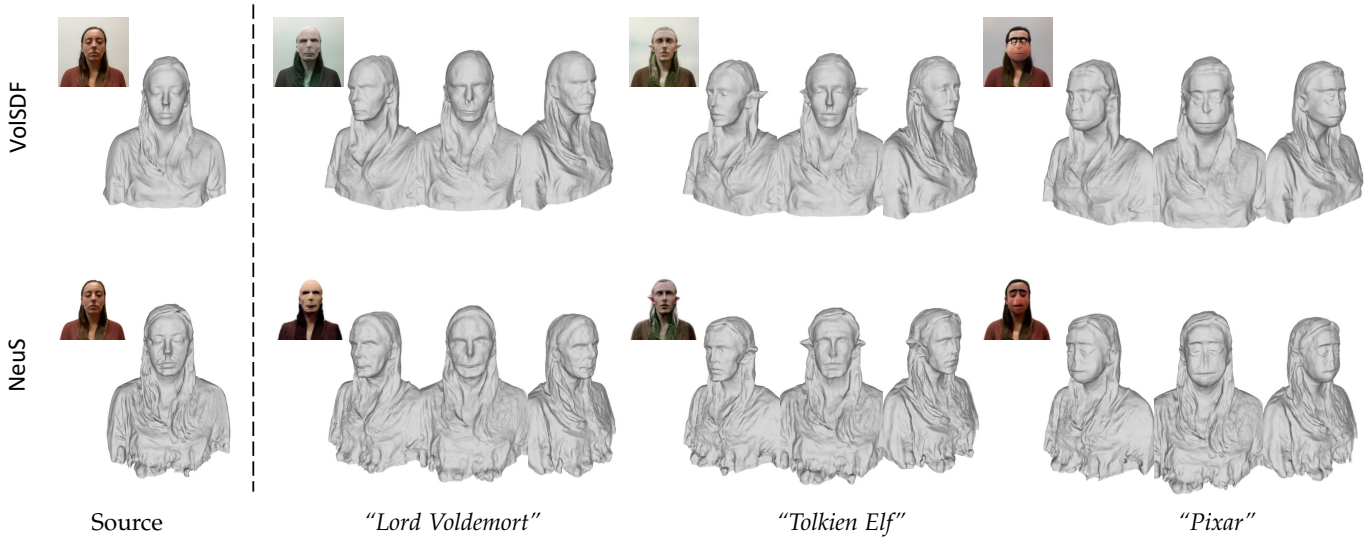


Fig. 10: **Geometry Evaluation.** Our method modulates the geometry and color simultaneously of a pre-trained *NeRF* to match the desired style described by a text prompt.

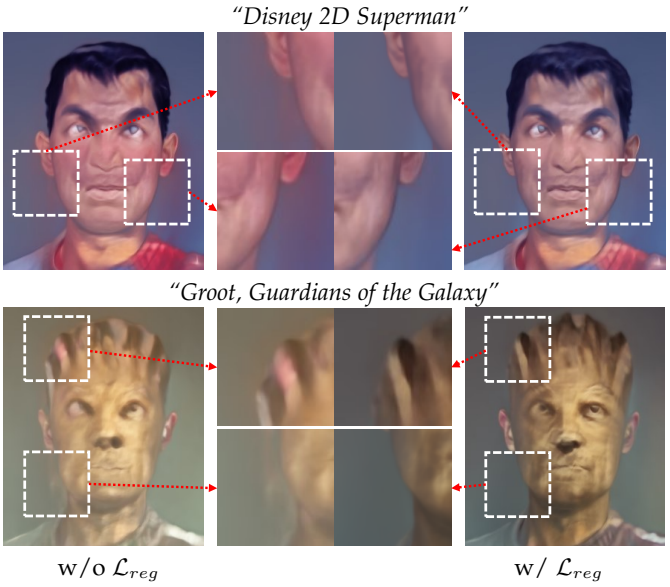


Fig. 11: **Ablations on Weight Regularization.** Cloudy artifacts near the corner or geometric noises are observed without the weight regularization loss.

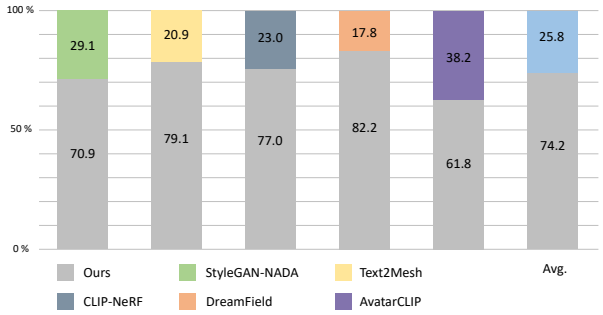


Fig. 12: **User Study.** Our method consistently outperforms state-of-the-art text-guided stylization methods on user preference rates (%).

by jointly considering three aspects: preservation of the content, faithfulness to the style, and view consistency. We finally collected 23 questionnaires completed by 10 male and 13 female participants. Statistics of the user study are shown in Fig. 12. Our method outperforms *StyleGAN-NADA*, *CLIP-NeRF*, *Text2Mesh*, *DreamField*, and *AvatarCLIP* by achieving much higher user preference rates. We conduct further repeated-measures analyses of variance (ANOVAs) on the results of the user study, and we find that our method consistently demonstrates significant superiority over all competitors ($p < 0.005$).

5.6 Ablation Study

Why global-local contrastive learning? A straightforward way to stylize *NeRF*s 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 3×3 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* [86] and *NeuS* [95]. 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

	Source	*StyleGAN-NADA*	*CLIP-NeRF*	*DreamField*	*Text2Mesh*	*AvatarCLIP*	Ours
CLIP	0.3901	0.7061	0.6392	0.6523	0.6753	0.6982	**0.7356**

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.

	Female	Man	Room	Trex	Flower	Fern	Human
Source	0.0130	0.0087	0.0056	0.0047	0.0023	0.0038	0.0010
CLIP-NeRF	0.0182	0.0228	0.0126	0.0093	0.0085	0.0124	0.0071
DreamFields	0.0161	0.0197	-	-	-	-	0.0043
AvatarCLIP	0.0155	0.0178	-	-	-	-	0.0022
Ours	0.0137	0.0092	0.0061	0.0054	0.0028	0.0042	0.0014

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 $\tilde{V}_t$. Between $M \cdot V_t$ and $M \cdot \tilde{V}_t$, 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 strength. 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

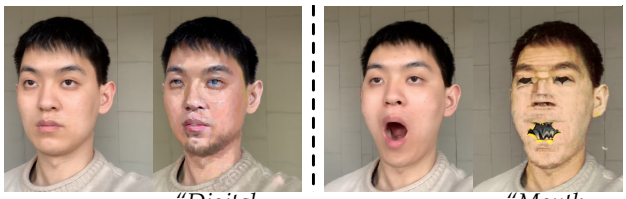


Fig. 13: **Limitations.** Linguistic ambiguity (left) or semantically meaningless words (right) may lead to unexpected results.

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

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[2] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1897–1906.

[3] Y. Shu, R. Yi, M. Xia, Z. Ye, W. Zhao, Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Gan-based multi-style photo cartoonization," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[4] Y. Zhao, X. Jin, Y. Xu, H. Zhao, M. Ai, and K. Zhou, "Parallel style-aware image cloning for artworks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 2, pp. 229–240, 2014.

[5] B. Sheng, P. Li, C. Gao, and K.-L. Ma, "Deep neural representation guided face sketch synthesis," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 12, pp. 3216–3230, 2018.

[6] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[7] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *Advances in neural information processing systems*, vol. 30, 2017.

[8] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *arXiv preprint arXiv:1705.01088*, 2017.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[10] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2402–2417, 2020.

[11] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907–3916.

[12] L. Höllein, J. Johnson, and M. Nießner, "Stylemesh: Style transfer for indoor 3d scene reconstructions," *arXiv preprint arXiv:2112.01530*, 2021.

[13] F. Han, S. Ye, M. He, M. Chai, and J. Liao, "Exemplar-based 3d portrait stylization," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[14] Z. Ye, M. Xia, Y. Sun, R. Yi, M. Yu, J. Zhang, Y.-K. Lai, and Y.-J. Liu, "3d-carigan: an end-to-end solution to 3d caricature generation from normal face photos," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[15] M. Zhang, J. Liao, and J. Yu, "Deep exemplar-based color transfer for 3d model," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[16] J. Guo, M. Li, Z. Zong, Y. Liu, J. He, Y. Guo, and L.-Q. Yan, "Volumetric appearance stylization with stylizing kernel prediction network," *ACM Trans Graph*, vol. 40, pp. 1–15, 2021.

[17] O. Klehm, I. Ihrke, H.-P. Seidel, and E. Eisemann, "Property and lighting manipulations for static volume stylization using a painting metaphor," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 7, pp. 983–995, 2014.

[18] X. Cao, W. Wang, K. Nagao, and R. Nakamura, "Psnet: A style transfer network for point cloud stylization on geometry and color," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3337–3345.

[19] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3d object reconstruction," in *proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[21] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun, "Fov-nerf: Foveated neural radiance fields for virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3854–3864, 2022.

[22] G.-W. Yang, W.-Y. Zhou, H.-Y. Peng, D. Liang, T.-J. Mu, and S.-M. Hu, "Recursive-nerf: An efficient and dynamically growing nerf," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[23] H. Zhang, F. Li, J. Zhao, C. Tan, D. Shen, Y. Liu, and T. Yu, "Controllable free viewpoint video reconstruction based on neural radiance fields and motion graphs," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[24] K. Wang, S. Peng, X. Zhou, J. Yang, and G. Zhang, "Nerfcap: Human performance capture with dynamic neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[25] P.-Z. Chiang, M.-S. Tsai, H.-Y. Tseng, W.-S. Lai, and W.-C. Chiu, "Stylizing 3d scene via implicit representation and hypernetwork," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1475–1484.

[26] Z. Fan, Y. Jiang, P. Wang, X. Gong, D. Xu, and Z. Wang, "Unified implicit neural stylization," *arXiv preprint arXiv:2204.01943*, 2022.

[27] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," *arXiv preprint arXiv:2108.00946*, 2021.

[28] T. Wei, D. Chen, W. Zhou, J. Liao, Z. Tan, L. Yuan, W. Zhang, and N. Yu, "Hairclip: Design your hair by text and reference image," *arXiv preprint arXiv:2112.05142*, 2021.

[29] O. Michel, R. Bar-On, R. Liu, S. Benaim, and R. Hanocka, "Text2mesh: Text-driven neural stylization for meshes," *arXiv preprint arXiv:2112.03221*, 2021.

[30] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "Avatarclip: Zero-shot text-driven generation and animation of 3d avatars," in *Proceedings of the ACM SIGGRAPH*, 2022.

[31] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," *arXiv preprint arXiv:2112.05139*, 2021.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[33] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 867–876.

[34] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 453–460.

[35] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th annual confer-*

ence on Computer graphics and interactive techniques, 2001, pp. 327–340.

[36] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[37] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 051–10 060.

[38] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 172–189.

[39] J. Huang, J. Liao, and S. Kwong, "Unsupervised image-to-image translation via pre-trained stylegan2 network," *IEEE Transactions on Multimedia*, vol. 24, pp. 1435–1448, 2021.

[40] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German conference on pattern recognition*. Springer, 2016, pp. 26–36.

[41] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1105–1114.

[42] X. Chen, Y. Zhang, Y. Wang, H. Shu, C. Xu, and C. Xu, "Optical flow distillation: Towards efficient and stable video style transfer," in *European Conference on Computer Vision*. Springer, 2020, pp. 614–630.

[43] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6654–6663.

[44] A. Mordvintsev, N. Pezzotti, L. Schubert, and C. Olah, "Differentiable image parameterizations," *Distill*, vol. 3, no. 7, p. e12, 2018.

[45] H.-T. D. Liu, M. Tao, and A. Jacobson, "Paparazzi: surface editing by way of multi-view image processing." *ACM Trans. Graph.*, vol. 37, no. 6, pp. 221–1, 2018.

[46] F. Han, S. Ye, M. He, M. Chai, and J. Liao, "Exemplar-based 3d portrait stylization," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[47] K. Yin, J. Gao, M. Shugrina, S. Khamis, and S. Fidler, "3dstylenet: Creating 3d shapes with geometric and texture style variations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 456–12 465.

[48] F. Mu, J. Wang, Y. Wu, and Y. Li, "3d photo stylization: Learning to generate stylized novel views from a single image," *arXiv preprint arXiv:2112.00169*, 2021.

[49] H.-P. Huang, H.-Y. Tseng, S. Saini, M. Singh, and M.-H. Yang, "Learning to stylize novel views," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 869–13 878.

[50] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[51] G. Li, M. Müller, G. Qian, I. C. D. Perez, A. Abualshour, A. K. Thabet, and B. Ghanem, "Deepgcns: Making gcns go as deep as cnns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[52] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," *arXiv preprint arXiv:2107.02791*, 2021.

[53] D. B. Lindell, J. N. Martel, and G. Wetzstein, "Autoint: Automatic integration for fast neural volume rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 556–14 565.

[54] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 346–14 355.

[55] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.

[56] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[57] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[58] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[59] R. Arandjelović and A. Zisserman, "Nerf in detail: Learning to sample for view synthesis," *arXiv preprint arXiv:2106.05264*, 2021.

[60] L. Ma, X. Li, J. Liao, Q. Zhang, X. Wang, J. Wang, and P. V. Sander, "Deblur-nerf: Neural radiance fields from blurry images," *arXiv preprint arXiv:2111.14292*, 2021.

[61] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[62] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–18, 2021.

[63] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7495–7504.

[64] S. Liu, X. Zhang, Z. Zhang, R. Zhang, J.-Y. Zhu, and B. Russell, "Editing conditional radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5773–5783.

[65] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5885–5894.

[66] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[67] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," *arXiv preprint arXiv:2112.00724*, 2021.

[68] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[69] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[70] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[71] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 959–12 970.

[72] A. Noguchi, X. Sun, S. Lin, and T. Harada, "Neural articulated radiance field," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5762–5772.

[73] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 314–14 323.

[74] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6498–6508.

[75] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, "Space-time neural irradiance fields for free-viewpoint video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9421–9431.

[76] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5712–5721.

[77] Y.-H. Huang, Y. He, Y.-J. Yuan, Y.-K. Lai, and L. Gao, "Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning," in *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[78] K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely, "Arf: Artistic radiance fields," *arXiv preprint arXiv:2206.06360*, 2022.

[79] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[80] H. Chefer, S. Benaim, R. Paiss, and L. Wolf, "Image-based clip-guided essence transfer," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*. Springer, 2022, pp. 695–711.

[81] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2085–2094.

[82] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *European Conference on Computer Vision*. Springer, 2020, pp. 319–345.

[83] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," *CVPR*, 2022.

[84] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.

[85] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 453–11 464.

[86] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4805–4815, 2021.

[87] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[88] E. Ramon, G. Triginer, J. Escur, A. Pumarola, J. Garcia, X. Giro-i Nieto, and F. Moreno-Noguer, "H3d-net: Few-shot high-fidelity 3d head reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5620–5629.

[89] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[90] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[91] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[92] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2287–2296.

[93] S. Lin, L. Yang, I. Saleemi, and S. Sengupta, "Robust high-resolution video matting with temporal guidance," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 238–247.

[94] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

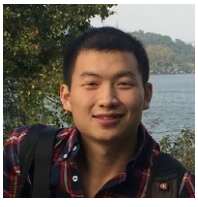
[95] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 171–27 183, 2021.

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, HKSAR, China. His research interests include Computer Graphics and Computer Vision, with a focus on neural rendering.

Menglei Chai is a Lead Research Scientist with the Creative Vision team in Snap Research. He received his PhD degree in Computer Science from Zhejiang University in 2017. His research interests lie in the intersection between Computer Vision and Computer Graphics, especially on human digitization, image manipulation, 3D reconstruction, and physics-based animation.

Mingming 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 generation, 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.