GANcraft-Art: A Pipeline for 3D Scene Creation and Text-Driven Artistic Stylization

Xiaochuan Qian

qianxiaochuan23@shphschool.com

Shanghai Pinghe School

December, 2024

Abstract

Creating and stylizing 3D scenes traditionally requires expertise, complex tools, and paired data, limiting accessibility. To address these challenges, we propose GANcraft-Art, a novel two-stage pipeline that transforms semantically labeled voxel-based 3D worlds into photorealistic scenes with customizable styles driven solely by text prompts. Using neural radiance fields enhanced with CLIP (Contrastive Language-Image Pretraining)-guided learning, our method enables flexible, view-consistent stylization without requiring reference images or paired datasets. Validated on Minecraft voxel maps, GANcraft-Art generates high-quality, coherent 3D scenes in diverse styles. While occasional artifacts and reliance on clear text prompts remain limitations, GANcraft-Art significantly lowers the barriers to 3D content creation, opening up new possibilities in game design, film production, and artistic expression.

Keywords: Novel view synthesis, Neural radiance field, CLIP, Text guided stylization.

1. Introduction

Artistic representation has been a key for human development, it allows for the expression of thoughts beyond words. In the contemporary world, our means of expression jumps out of the boxes of traditional art. We took 2D scenes into 3D, enabling us to explore a scene in different perspectives. Traditionally, creating a 3D scene require extensive learning of modeling software such as C4D or Maya, which despite various resources on the open internet and documentations can still be a painstaking task. Not to mention, creating a realistic scene may take years of training and months to complete. With an increasing demand and fierce competition in 3D related industries such as film and game design (Figure 1). There is demand for a faster, reliable way to create 3D scenes.

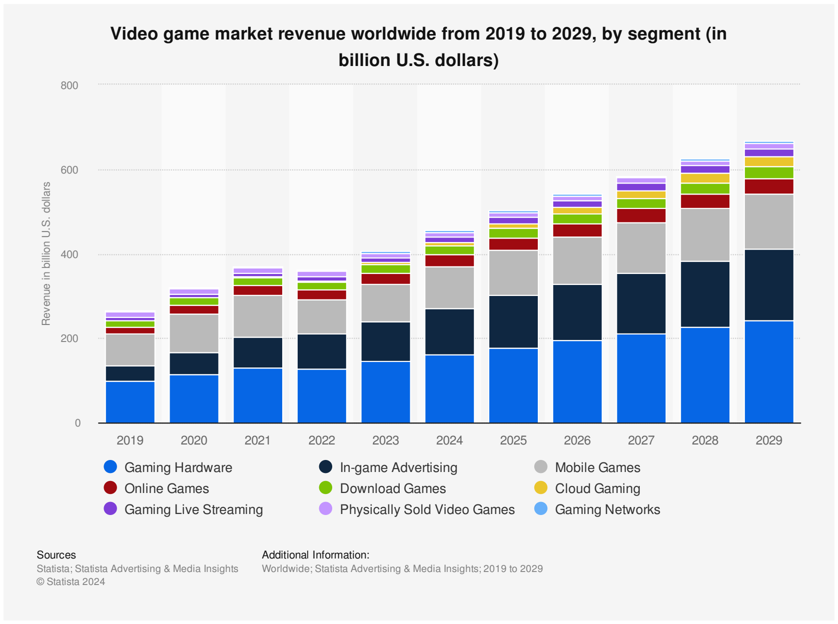


Figure 1 Video game market revenue worldwide from 2019 to 2029, by segment (in billion U.S. dollars)

The most intuitive way for 3D scene construction is to "build it". Like building houses with LEGOs. Desirably, we want to first build the scene with blocks representing the real-life objects. We then feed the built scene into an algorithm to turn it in to a realistic 3D scene with trees water, and grass. Obviously, the real-life application of 3D scenes in beyond representing the real world. We would want to alter it to differing styles to fit our needs. What if we can describe the desired style with words and run another algorithm to transform it to that style? In this way would significantly reduce the workload for prototyping and allow for faster production.

In this paper, we set the goals to create a pipeline that allow for easy style transfer based on a realistically rendered scene build upon 3D block worlds. We use the video game Minecraft as our main tool for 3D block world synthesis. It is a popular game developed by Mojang studio that sold 300 million copies worldwide. (Mojiang Studio, n.d.) Players interact with the world by using

Table 1 Comparison of methods in novel-view synthesis

Representative Methods	View Consistent	Paired Data	Stylization	Text-driven	Voxel Based
NeRF (Mildenhall et al., 2020), NSVF (M.-Y. Liu et al., 2020)	✓	×	×	×	×
Text2Mesh (Michel et al., 2021), AvatarCLIP (Hong et al., 2022)	✓	×	✓	✓	×
StyleGAN-NADA (Gal et al., 2021)	×	×	✓	✓	×
StylizedNeRF (Y.-H. Huang et al., 2022), Chiang (2022), ARF (Zhang et al., 2022)	✓	×	✓	×	×
GANcraft (Hao et al., 2021)	✓	✓	✓	×	✓
NerF-Art (C. Wang et al., 2022)	✓	×	✓	✓	×
Ours	✓	✓	✓	✓	✓

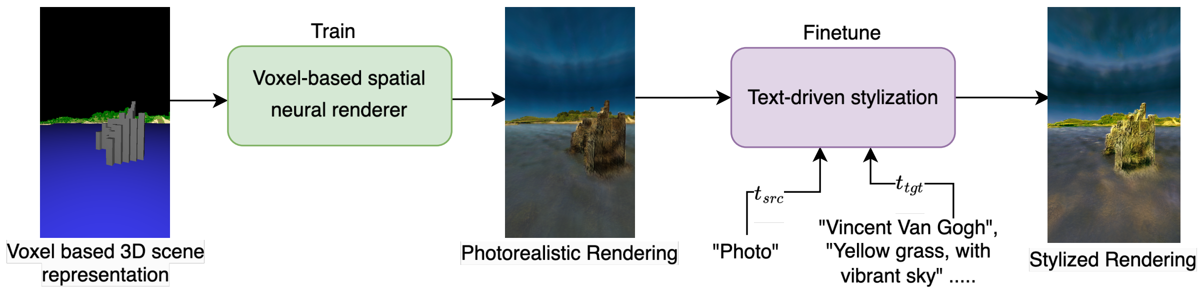

Voxel based 3D scene representation
Train
Voxel-based spatial neural renderer
Photorealistic Rendering
Finetune
Text-driven stylization
t_{src}
t_{tgt}
"Photo"
"Vincent Van Gogh", "Yellow grass, with vibrant sky" .....
Stylized Rendering

Figure 2 GANcraft-Art Pipeline. *This is a two staged process. In the training stage, we first pre-train the Voxel-based spatial neural renderer, which can generate photorealistic rendering of the voxel world. In the finetune stage, our method stylize the model with $\boldsymbol{t}_{tgt}$ and $\boldsymbol{t}_{src}$ as additional input.*

semantically labeled blocks. Different blocks represent different object in the corresponding real world. It is not only a beloved video game, but also a popular creation tool. Players have created numerous buildings, namely replications of the Forbidden City, The Great Wall, and even the city of Shanghai! This owes to the extremely easy, intuitive playing method, allowing for scene creation in a relatively short amount of time.

Building upon this foundation of 3D world creation, previous methods in the domain of 3D view synthesis have explored combining traditional graphics pipelines with neural networks to achieve view-consistent outputs (T. H. Nguyen-Phuoc et al., 2020; Sitzmann et al., 2019). By incorporating differentiable 3D projection and trainable layers operating in both 3D and 2D feature spaces, these approaches successfully model scene geometry and appearance from 2D images. Although some works have integrated adversarial training to eliminate the need for posed training images (T. H. Nguyen-Phuoc et al., 2020; Niemeyer & Geiger, 2021), they remain limited to single objects or simple scenes due to the under-constrained nature of the problem.

To address these limitations, a breakthrough came with NeRF (Mildenhall et al., 2020), which achieves remarkable novel view synthesis by encoding scenes in neural networks that produce volume density and view-dependent radiance. This sparked numerous improvements, including enhanced computational efficiency (M.-Y. Liu et al., 2020), adaptation to unstructured photo collections (Martin-Brualla et al., 2021), and the addition of generative capabilities (Niemeyer & Geiger, 2021). However, despite these advances, challenges remain in handling complex scenes with diverse training data and varying camera pose distributions. Furthermore, the implicit representation of neural radiance field makes it more difficult to stylize the colors and shape embeded in feature radiance fields.

In response to these challenges, NeRF-based 3D stylization has emerged as a promising direction. Several works (Chiang et al., 2022; Y.-H. Huang et al., 2022; Zhang et al., 2022) have adapted NeRF for style transfer by predicting color-related parameters based on reference style images. In parallel, approaches like GANcraft (Hao et al., 2021) successfully mapped a voxel based scene into a realworld-like one using GAN, which consist of a generator and a discriminator engaged in a minimax game, in the absence of paired training data. Nevertheless, while these approaches achieve view-consistent stylization, they are limited to appearance modifications only, without altering geometric properties.

While these reference image-based methods show promise, they still require explicit visual examples for style transfer. To address this limitation, natrual language has emerged as a more intuitive way for humans to interact with machines. Text-guided manipulation has emerged as a powerful approach through CLIP (Radford et al., 2021) which bridges visual and textual information. On one hand, existing methods have achieved text-guided stylization in both 2D (Gal et al., 2021) and 3D domains (Hong et al., 2022; Michel et al., 2021), but they either lack view consistency or require explicit mesh inputs, making them unsuitable for neural radiance field representations. On the other hand, most recently, NeRF-Art (C. Wang et al., 2022) has displayed promising results in style transfer of neural radiance fields, which do not rely on mesh inputs and can product consistant images. Yet, it requires paired ground truth, which in our case is not possible to obtain.

In our work, we propose GANcraft-Art, a two staged pipline that overcomes the challenges view consistancy in the circumstances of missing paired data in novel view synthesis, while also providing text driven stylization that maintains the consistancy. Our approach works as follows: In the first stage, we train upon photorealistic images generated on the go from segmentation maps sampled from the voxel based Minecraft world to create a photorealistic represenation of the 3D scene. Building upon this foundation, we apply stylization with the help of CLIP (Radford et al., 2021) through directional and contrastive learning to result in a styilized scene matching the users desire. In Section 2, we review related works on text-guided synthesis, 3D stylization, and neural radiance fields, highlighting their limitations. In Section 3, we introduce the our data and the preprosses workflow. In Section 4, we describe the GANcraft-Art pipeline, including its photorealistic rendering stage and text-driven stylization using CLIP-based learning. In Section 5, we present experimental results, demonstrating the effectiveness and consistency of our approach compared to existing methods. In Section 6, we discuss the implications, limitations, and future directions of our work.

In summary, our contributions are:

- We propose GANcraft-Art, a novel two-staged pipeline that combines voxel-based 3D scene generation with text-driven stylization, achieving both view consistency and flexible style control without requiring paired training data.
- We develop an efficient approach to generate photorealistic 3D scenes from Minecraft worlds by leveraging on-the-fly segmentation map sampling and adversarial training, making 3D scene creation more accessible to non-experts.
- We provide a user-friendly pipeline that significantly reduces the technical barriers to 3D scene creation and stylization, enabling rapid prototyping and production in various applications such as game design and virtual environment creation.

2. Related Works

In this section we review works related to this work, so that we identify the shortcomings of prior works and the challenges that we face. In Section 2.1, we first introduce works in the field of image-to-image translation, which produces inconsistent views in our setting. Thus, in the following section, we review works in 3D neural rendering that overcome the limitation of incoherent views, and we discuss the implication of 3D neural rendering in the absence of ground truth training images. In Section 2.3, we review existing works in the field of stylization, particularly focusing on neural radiance fields. We first review style translation in Section 2.3.1 with reference images. Then, we explore text-driven approaches in Section 2.3.2 that provide a more intuitive, user-friendly means of style specification.

2.1 Non-view-consistant 2D image-to-image translation

There are many ways to effectively map an image from a domain to another with high fidelity. This task can be approached in both supervised (Isola et al., 2016; Lee et al., 2019; X. Liu et al., 2019; T. Park et al., 2019; T.-C. Wang et al., 2017; Zhu et al., 2017) and unsupervised settings. (M.-Y. Liu et al., 2018, 2019; X. Liu et al., 2019; Sushko et al., 2020; Zhu et al., 2020) In the supervised setting, where pairs of corresponding images are available, methods often utilize stronger losses like the L1 or perceptual loss (Johnson et al., 2016) alongside the adversarial loss. In contrast, the unsupervised setting, which lacks paired data, typically relies on shared-latent space assumptions (M.-Y. Liu et al., 2018) or cycle-consistency losses. (Zhu et al., 2020)

In our scenario, the goal is to translate a scene from one domain to another. The most intuitive way is to generate a view and use image-to-image translation to transform the image to the desired style. However, this may result in differing views across different images. Thus, while these aformentioned 2D translation approaches have shown impressive results, they face fundamental limitations when applied to 3D scenes where view consistency is crucial.

2.2 View-consistant 3D neural rendering

Several works have explored the combination of traditional graphics pipeline strengths, such as 3D-aware projection, with the synthesis capabilities of neural networks to produce view-consistent outputs. By introducing differentiable 3D projection and employing trainable layers that operate in both 3D and 2D feature spaces, recent methods (Aliev et al., 2020; Henzler et al., 2021; T. Nguyen-Phuoc et al., 2019; T. H. Nguyen-Phuoc et al., 2020; Sitzmann et al., 2019; Wiles et al., 2020) have been able to model the geometry and appearance of 3D scenes from 2D images. Some works have successfully integrated neural rendering with adversarial training (Henzler et al., 2021; T. Nguyen-Phuoc et al., 2019; T. H. Nguyen-Phuoc et al., 2020; Niemeyer & Geiger, 2021; Schwarz et al., 2021), eliminating the need for training images to be posed and from the same scene. However, the under-constrained nature of the problem has limited these methods to single objects, synthetic data, or small-scale simple scenes. Hence, the above mentioned is insufficient to handle the larger and more complex nature of our input scenes.

In solving the limitation, most recently, NeRF (Mildenhall et al., 2020) has achieved unseen results in novel view synthesis by encoding the scene in the weights of a neural network that produces volume density and view-dependent radiance at every spatial location. NeRF's outstanding novel view synthesis capabilities has encouraged many following works aimed at improving output quality (L. Liu et al., 2020; Zhang et al., 2020), speeding up training and evaluation (Lindell et al., 2020; L. Liu et al., 2020; Neff et al., 2021; Rebain et al., 2020; Tancik et al., 2021), extending it to deformable objects (Du et al., 2021; Li et al., 2021; K. Park et al., 2021; Pumarola et al., 2020; Xian et al., 2021), accounting for lighting (Bi et al., 2020; Boss et al., 2021; Martin-Brualla et al., 2021; Srinivasan et al., 2020), and enhancing compositionality (Guo et al., 2020; Niemeyer & Geiger, 2021; Ost et al., 2021; Yuan et al., 2020), as well as adding generative capabilities (Chan et al., 2021; Niemeyer & Geiger, 2021; Schwarz et al., 2021). Most relevant to our work are NSVF (L. Liu et al., 2020), NeRF-W (Martin-Brualla et al., 2021), and GIRAFFE (Niemeyer & Geiger, 2021). NSVF (L. Liu et al., 2020) reduces NeRF's computational cost by representing the scene as a set of voxel-bounded implicit fields organized in a sparse voxel octree, obtained by pruning an initially dense cuboid made of voxels. NeRF-W (Martin-Brualla et al., 2021) learns image-dependent appearance embeddings, allowing it to learn from unstructured photo collections and produce style-conditioned outputs.

These novel view synthesis works mentioned above learn the geometry and appearance of scenes from ground truth posed images. In our scenario, the problem is inverted—we are provided with coarse voxel geometry and segmentation labels as input, without any corresponding real images. Similar to NSVF (L. Liu et al., 2020), we assign learnable features to each corner of the voxels to encode geometry and appearance. However, instead of learning the 3D voxel structure of the scene from scratch, we implicitly refine the provided coarse input geometry (e.g., the shape and opacity of trees represented by blocky voxels) during training. Prior work by Riegler et al. (Riegler & Koltun, 2020) also used a mesh obtained by multi-view stereo as coarse input geometry. Like NeRF-W (Martin-Brualla et al., 2021), we use a style-conditioned network, allowing us to learn consistent geometry while accounting for SPADE's (T. Park et al., 2019) view inconsistency. Similar to neural point-based graphics (Aliev et al., 2020) and GIRAFFE (Niemeyer & Geiger, 2021), we use differentiable projection to obtain features for image pixels, followed by a CNN to convert the 2D feature grid into an image.

2.3 Stylization

2.3.1 Neural radiance fields stylization

With the improvement of NeRF's quality, efficiency, generalizablilty, three recent (Chiang et al., 2022; Y.-H. Huang et al., 2022; Zhang et al., 2022) works employ NeRF in the field of stylization. In Zhang et al. (2022) and Chiang et al.'s (2022) work, a style transfer loss (Gatys et al., 2016) is applied during training on the rendered views, while Huang et al.'s solution is to employ a stylization network that is mutually learnt. In addition, in the situation of missing paired training data, GANcraft (Hao et al., 2021) uses the ideas proposed in GANs (Generative Adversarial Network)(Goodfellow et al., 2014), by having a discriminator and a generator playing a minmax game. The above mentioned methods produced view consistant results, but they only alter the surface color of the radiance field. Both apperance stylization and geometry alteration is supported in our method to produce results that better resemble the given style.

2.3.2 Text-driven stylization

Using natural languages to interact with machines has always been a much more user-friendly mean than providing a reference picture. Our goal is to use text descriptions to manipulate the style of a scene. By using CLIP (Radford et al., 2021) to bridge the input image with the target text in a shared latent space. Gal et. al. (2021) successfully transformed a pretrained StyleGAN2 into the desired style using text guidance by proposing a directional CLIP loss. However, this method will lead to view inconsistencies due to its image-based nature. Text2Mesh (Michel et al., 2021), AvatarCLIP (Hong et al., 2022) demonstrated the implication of CLIP in the 3D world utilizing text to guide the stylization of a given mesh. Despite this, they require the input of a mesh grid which is not possible in our situation as we are using neural radiance field to synthesize our scene. NeRF-Art (C. Wang et al., 2022) do not rely on mesh inputs and can synthesis consistant results in novel view stylization, but it still do not fit the requirement of our use case: to train a neural radiance field with not avaliable ground truth. Our method improves upon NeRF-Art, by using L2 loss to further preserve geometric consistency and scene structure, even in the absence of ground truth data.

3. Data and data preprocess

This section we introduce the two stages of data preprocessing before the training of neural radiance field. First, we detail our approach to extracting and preprocessing Minecraft world data (Section 3.1), converting it from the game's native format into a representation suitable for neural rendering. Then, we address the challenge of missing paired training data by introducing a pseudo-ground truth generation pipeline (Section 3.2) that bridges the domain gap between Minecraft's voxel-based representation and photorealistic imagery.

3.1 Minecraft world loading and preprocessing

The Minecraft world is generated by the game automatically upon a creation of a new world. However, the game itself does not provide the tools to allow third parties to access the game data. We employ a software Mineways (Eric Haines, n.d.) to export the data from the game. A specific area of the world is chosen and exported. The voxel data is read as a contiguous array of 32-bit integers, which is then reshaped and transposed to match the desired 3D world structure. The resulting tensor represents the Minecraft world, where each voxel is identified by a unique integer ID corresponding to a specific block type. To optimize the world representation for our GAN model, we implemented a series of filtering and simplification steps:

a) Block type filtering: Certain block types that do not significantly contribute to the overall structure (e.g., lily pads, vines) were removed by setting their voxel values to 0 (air). Some block types were remapped to similar, more common types to reduce the overall variety of blocks.

b) Voxel simplification: We applied morphological operations to simplify the voxel structure. Corresponding to the configuration, we implemented a "hollow bottom" technique using binary erosion or first perform binary dilation followed by erosion to close cells before hollowing the bottom. These operations help reduce noise and simplify the internal structure of the world.

To further decrease that computation load, we compute the maximum height of non-air blocks for each (x, z) coordinate. This 2D representation is then used as reference to truncate the voxel data to remove unnecessary empty space above the highest point and below the lowest point of the terrain.

The data points in the exported data are represented as the center of a voxel. We convert this representation to corner representation. This is beneficial for subsequent calculation of ray passing through the neural radiance field, as there is less calculation need to determine the borders of each voxel.

3.2 Pseudo-ground truth data generation

Given that our training setting is exclusively categorized as unsupervised due to the unpaired nature of the training data, the conventional approach involves the application of techniques such as CycleGAN (Zhu et al., 2020) or MUNIT (X. Huang et al., 2018), which incorporate adversarial loss and regularization terms to connect Minecraft segmentations with real

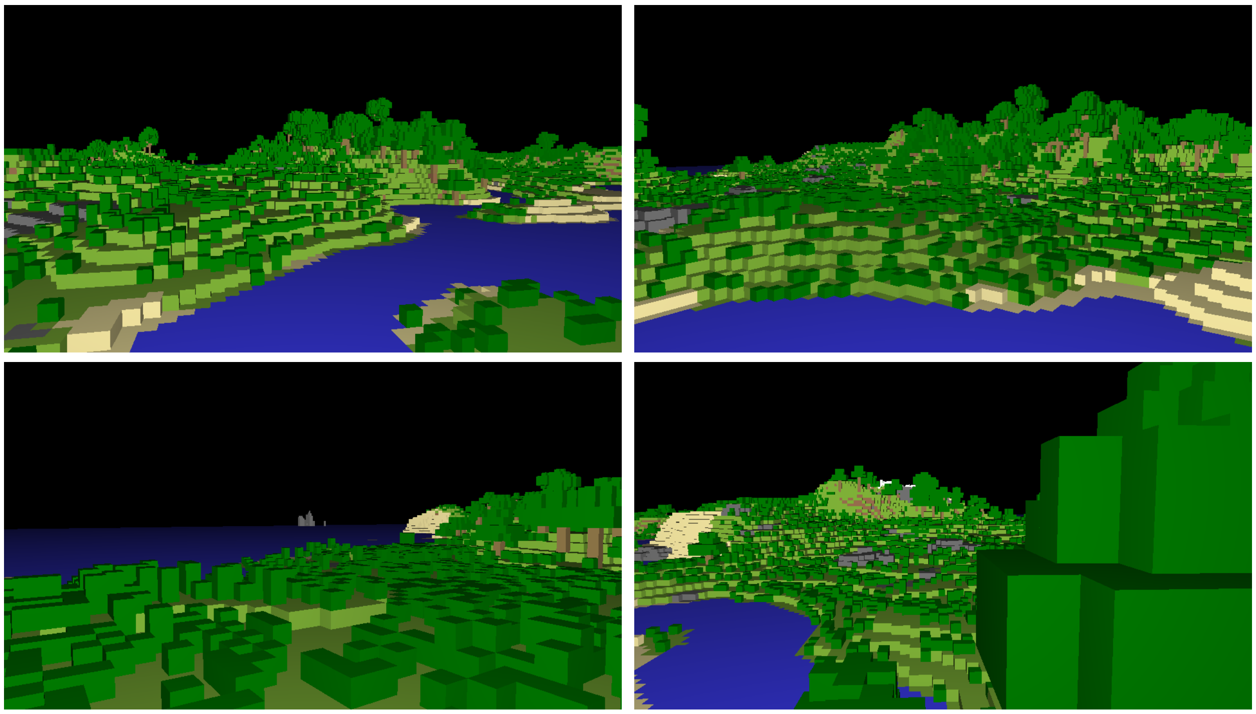


Figure 3 Sample of input voxel world. *The above four photos are from the same Minecraft world in different angle.*

images. Nonetheless, the results achieved are suboptimal, primarily owing to the significant domain gap between the sparsely represented Minecraft environment and the richly detailed real world. A more effective strategy would entail utilizing a rendering model that is trained with ground truth paired data. Subsequently, the incorporation of L2 reconstruction loss into the model would suffice to yield superior outcomes. However, the circumstances we encounter do not permit the application of this method. Rather than seeking an alternative solution, we choose to implement additions and modifications to our data to satisfy the necessary prerequisites.

In order to address the domain disparity between the voxel environment and our real-world context, we proxy the training dataset by generating pseudo-ground truth during training. During each training iteration, we randomly select camera positions from the upper hemisphere and choose a focal length at random. Subsequently, we project the semantic labels of the voxels onto the camera perspective to create a 2D semantic segmentation mask. This segmentation mask, along with a randomly selected style code, is then input into a pretrained image-to-image translation network, specifically SPADE (T. Park et al., 2019) in this instance, to produce a photorealistic pseudo-ground truth image that maintains the same semantic structure as the camera perspective. This methodology allows for the application of reconstruction losses, such as L2 loss, as well as perceptual loss (Johnson et al., 2016), between the pseudo-ground truth and the rendered output from the corresponding camera perspective, in addition to the adversarial loss. This approach notably enhances the overall results.

While SPADE (T. Park et al., 2019) solves the issue of missing paired data, and significantly reduces the domain gap, SPADE can still render blocky images due to the features

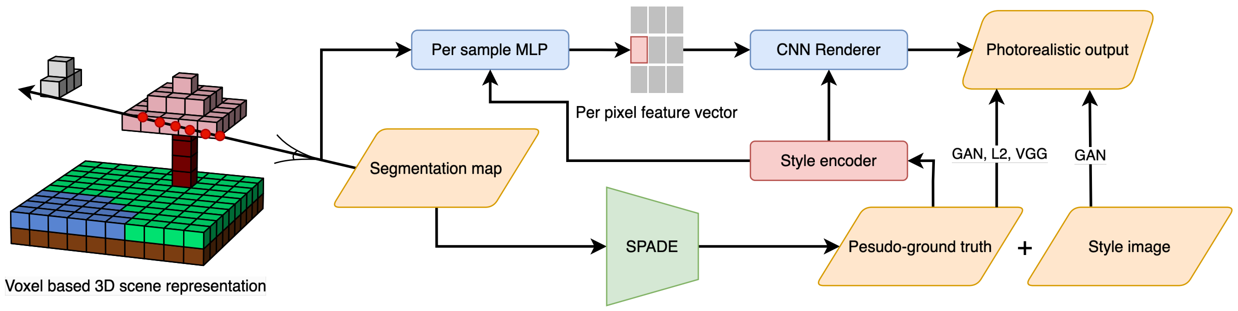


Figure 4 Pipeline of voxel-based spatial neural renderer. *We sample random views from the voxel world and take the segmentation map of the views to generate pseudo-ground truth with SPADE. For the same views, we allow rays to travers through the voxel world and process it in a MLP to obtain the per pixel feature vector, which is then converted to RGB values using the CNN renderer. Note that the style encoder will condition both the MLP and CNN renderer based on the style of the pseudo-ground truth. We then calculate GAN, L2, VGG loss with the pseudo-ground truth and GAN loss with the additional style image.*

parsed into the model from Minecraft. Artifacts can be produced for certain camera angles and style code. Thus, the weights of the reconstruction L2 loss and adversarial loss must be cautiously balanced to achieve successful training and photorealistic results.

4. Method

Essentially, the objective is to transform a scene represented by semantically labeled blocks, which in this case is the world generated by the game Minecraft, into a realistic scene that can be outputted with significant consistency when rendered from different viewpoints. Landscape scenes are our main point of focus in this paper, which differs from the typical single objects and small scene used in training. To be more specific, in all the experiments, Minecraft world with the dimension of 512×512×256 blocks. (a rectangular space with a 512×512 as base, and 256 blocks tall) Moreover, the rendered scene must fill in the details, such as trees, grass, and leaves, of single blocks in an unsupervised manner, since blocks in Minecraft only entail enough detail to let users identify the type of the block, which is far from realistic. Thus, the relative semantic information and geometric arrangement must be constituted into the model. After this objective is accomplished, we set our goals to stylize the reconstructed photorealistic scene using text descriptions as the sole guidance. This involves adapting the appearance of the scene to match the semantic and aesthetic elements dictated by the target text, while preserving the content and geometry of the original reconstruction. The stylization process must ensure consistency across multiple viewpoints, avoiding artifacts or distortions, which is crucial for maintaining coherence in 3D scene rendering.

We will illustrate the voxel rendering procedure in Section 4.1. Then, we introduce stylization calculations in Section 4.2.

4.1 Voxel-based spatial neural renderer

We use the model proposed in GANcraft (Hao et al., 2021) for our 3D neural radiance field representation, which considers the ground part and sky part of each generated photo separately, then, it assembles them by cumulating feature upon a ray passes through the radiance field. We first introduce the representation of neural radiance field bounded by each voxel. Then, take sky into consideration by covering the voxels with an infinitely far sky dome. After that, we travers rays through the radiance field to acquire the feature, which is then transformed into RGB values displayed as the final output.

4.1.1 Voxel neural radiance fields

The Minecraft voxel world can be represented by neural radiance field given by

$$F(\mathbf{p},\mathbf{z}) = \begin{cases} F_i(\mathbf{p},\mathbf{z}), & \text{if } \mathbf{p} \in V_i, \quad i \in \{1,\cdots,K\} \\ (\mathbf{0},0), & \text{otherwise} \end{cases} \tag{1}$$

This is the union of the individual radiance field bounded by each voxel and the empty voxels. Assuming the number of voxels in the world is K, the total blocks in the world will be represented as $V = \{V_1, \dots, V_k\}$. To distinguish the texture that each block represents, voxels are given a semantic label of $\{l_i, \dots, l_k\}$. In the equation, F is the radiance field of the whole scene and F_i is the radiance field of each individual voxels. Upon calling the function with a location coordinate returns a feature vector and a density value. If the query is an empty voxel, the function will return the null state of a feature vector 0 and 0 density. A style code z is given to the model to account for the different requirement in the appearance of the same scene.

F_i is given by

$$F_i(\mathbf{p},\mathbf{z}) = G_\theta(g_i(\mathbf{p}), l_i, \mathbf{z}) = (\mathbf{c}(\mathbf{p}, l(\mathbf{p}), \mathbf{z}), \sigma(\mathbf{p}, l(\mathbf{p}))), \tag{2}$$

where $g_i(\mathbf{p})$ is the location code at $\mathbf{p}$. The notation l_i can be extended into $l(p)$, indicating the label that a given voxel belongs to. G_θ is a MLP that is used to predict the feature $\mathbf{c}$ and density σ at the location $\mathbf{p}$. The G_θ MLP is a shared network covering all the voxels. We implement the same idea in NeRF-W (Martin-Brualla et al., 2021): the color output c is style-dependent, while density σ is style-independent. For spatial encoding, we implement a location code system where each voxel V_i is characterized by learnable feature vectors at its eight vertices, with $g_i(p)$ computed through trilinear interpolation. Assuming operation within a unified coordinate system of unit-dimensioned voxels (1×1×1), we enhance feature vector sharing across adjacent voxels' common vertices, ensuring smooth inter-voxel transitions and output continuity. This approach effectively generalizes NSVF (L. Liu et al., 2020) by introducing comprehensive style and semantic label conditioning mechanisms.

4.1.2 Sky representation

To ensure a photorealistic and continuous render of the scene, the appearance of the sky has to be also dependent on the viewing angle. We employ the same method as used in computer graphics i.e. environment mapping to consider the sky as an infinite-distance element. The sky portion of the photo can be seen as an infinitely big sphere covering the voxel world. We implement this through a MLP $H_{\emptyset}$ that maps viewing direction v to sky color (or feature) $c_{sky} \equiv H_{\emptyset}(v, z)$, with style conditioning through code z.

4.1.3 Volumetric rendering

With the terrain part and the sky prepared, we now use rays traversing through the voxel world to "assemble them up" to form a picture. Within a perspective camera framework, each image pixel corresponds to a parametric ray $\boldsymbol{r}(t) = o + tv$, emanating from the projection center o along direction v. As this ray traverses the radiance field, it accumulates both features and transmittance values.

$$C(\boldsymbol{r}, \boldsymbol{z}) = \int_0^{+\infty} T(t)\sigma\left(\boldsymbol{r}(t), l(\boldsymbol{r}(t))\right) \boldsymbol{c}(\boldsymbol{r}(t), l(\boldsymbol{r}(t)), \boldsymbol{z}) dt \; + \; T(+\infty) c_{sky}(\boldsymbol{v}, \boldsymbol{z}) \, ,$$

$$where\; T(t) = \exp(-\int_0^t \sigma(r(s)) ds). \qquad (3)$$

The accumulated feature along ray r is denoted by $C(\boldsymbol{r}, \boldsymbol{z})$, while $T(t)$ represents the accumulated transmittance at distance t. Given the voxel-bounded nature of our radiance field, rays inevitably exit the voxel structure and intersect with the sky dome, which we model as a fully opaque terminal point along each ray—represented by the final term in the intergral. This continuous integral is discretized through sampling and quadrature rules, following methodologies established in NeRF (Mildenhall et al., 2020).

Our implementation adopts the stratified sampling approach from NSVF (L. Liu et al., 2020) for point sampling within voxel-bounded regions. To enhance computational efficiency, we implement ray truncation upon reaching a predetermined accumulated distance through valid regions, with regularization applied to encourage opacity saturation prior to maximum distance threshold. Point sampling is executed through a modified Bresenham algorithm (Amanatides & Woo, n.d.) , achieving computational complexity of $O(N)$, where N represents the voxel grid's maximum dimension.

4.1.4 Hybrid neural rendering

Our methodology diverges from previous approaches (L. Liu et al., 2020; Martin-Brualla et al., 2021; Mildenhall et al., 2020) by decomposing the rendering pipeline into two distinct stages, rather than directly accumulating colors through volumetric rendering.

a) We first implement volumetric rendering via an MLP to produce per-pixel feature vectors.
b) We employ a CNN to transform these features into final RGB images of corresponding dimensions.

Both networks incorporate activation modulation (X. Huang & Belongie, 2017; T. Park et al., 2019) conditioned on style codes. This method allow for higher image quality and reduced memory usage.

4.1.5 Losses

The training framework of our model incorporates both reconstruction and adversarial components. Our reconstruction objective evaluates the disparity between network predictions and their corresponding pseudo-ground truth counterparts, utilizing a composite loss that combines perceptual (Gatys et al., 2016), and L2 loss. In our adversarial network, we designate generator outputs as 'fake' samples, while categorizing both real images and pseudo-ground truth samples as 'real'. The discriminator architecture, which takes semantic segmentation maps as conditioning input employs hinge loss (Lim & Ye, 2017) as its training criterion.

To enable appearance control through reference imagery, we integrate a style encoder module. This encoder estimates the posterior distribution of style codes from pseudo-ground truth images, working in concert with the reconstruction loss to facilitate style-guided image generation.

As mentioned in Section 3.3.3, we truncate the ray when the it reaches a certain distance. This may lead to potential artifacts, To mitigate this, we implement an opacity regularization term. This regularizer, expressed as $\mathcal{L}_{\text{opacity}} = \sum_{\mathbf{r} \in \mathbf{R}_{\text{trunc}}} T_{\text{out}}(\mathbf{r})$, operates on truncated rays and serves to minimize residual transmittance beyond the truncation threshold.

4.2 Text-driven stylization

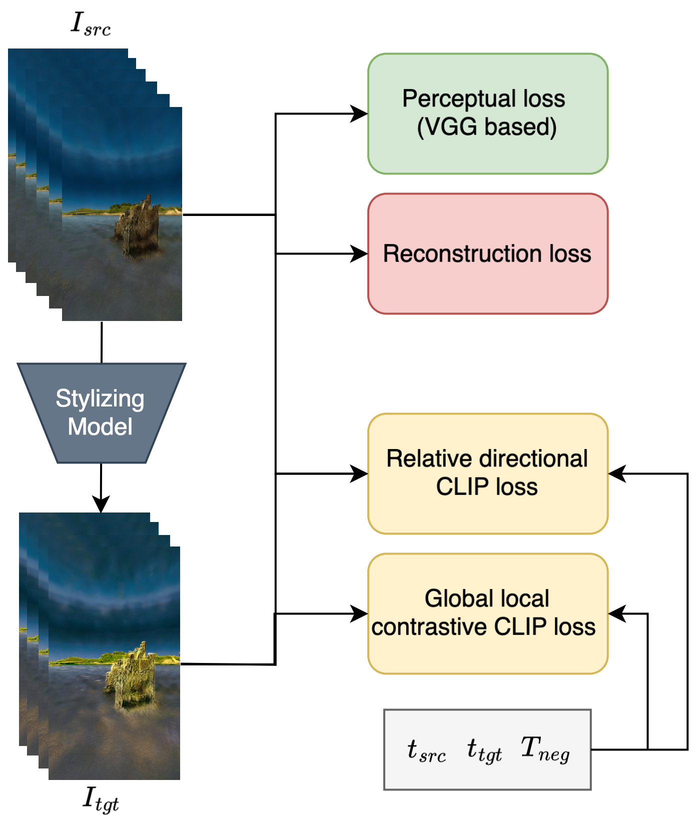
After the reconstruction of voxel world. The objective is to train a stylized model which coincide with the provided target text while preserving the content and structure from the original model.

To bridge this domain gap, we employ CLIP, which facilitates semantic alignment between visual and textual modalities through a shared embedding space, leveraging dedicated image and text encoders, $\hat{\mathcal{E}}_i(\cdot)$ and $\hat{\mathcal{E}}_t(\cdot)$ respectively. This method is inspired by NeRF-Art (C. Wang et al., 2022). While CLIP's semantic representation capabilities enable text-driven stylization of Neural Radiance Fields, several fundamental challenges persist in achieving optimal results. These challenges manifest in three critical aspects:

a) Maintaining content fidelity while incorporating stylistic elements.
b) Calibrating appropriate stylization intensity to accurately reflect the semantic intent of textual prompts.
c) Ensuring view-consistent rendering without introducing artifacts in the resulting radiance field representation.

4.2.1 Directional CLIP loss

The most straightforward approach to achieving text-guided stylization for a neural radiance field is to directly calculate the cosine similarity ($\langle \cdot, \cdot \rangle$) between the target style from the text description provided and the stylized rendering of the voxel-based spatial neural renderer given by:



I_{src}
Perceptual loss (VGG based)
Reconstruction loss
Stylizing Model
Relative directional CLIP loss
Global local contrastive CLIP loss
t_{src} t_{tgt} T_{neg}
I_{tgt}

Figure 5 Pipeline of text-driven stylization. *We stylize the model in the training stage by finetuning it with relative diectional CLIP loss and global local contrastive CLIP loss in the latent CLIP space. In addition, be also calculate perceptual loss and reconstruction loss.*

$$\mathcal{L}_{dir}^{a} = \sum_{I_{tgt}} \left[1 - \langle \hat{\mathcal{E}}_i(\boldsymbol{I}_{tgt}), \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}) \rangle\right], \tag{4}$$

where $\boldsymbol{I}_{tgt}$ is the rendering by the spatial neural renderer, and $\boldsymbol{t}_{tgt}$ is the target text. This absolute directional CLIP loss guides the stylization in the CLIP space with no reference to the starting point of the style. The loss was first proposed in StyleCLIP (Patashnik et al., 2021) and later used in CLIP-NeRF (C. Wang et al., 2021). However, this approach frequently leads to mode collapse, leading to limited diversity in generation. (Gal et al., 2021)

Thus, the solve this problem, we use a relative directional CLIP loss proposed by NeRF-Art, which is given by:

$$\mathcal{L}_{dir}^{r} = \sum_{I_{tgt}} \left[1 - \langle \hat{\mathcal{E}}_i(I_{tgt}) - \hat{\mathcal{E}}_i(I_{src}), \hat{\mathcal{E}}_t(t_{tgt}) - \hat{\mathcal{E}}_t(t_{src}) \rangle\right], \tag{5}$$

where rather than utilizing a single text prompt, we provide CLIP with a pair of text $t_{tgt}$and t_{src}, representing the target and the source text description of the style. I_{tgt} and I_{src} represent the same view, but I_{tgt} is the view rendered from the pretrained neural radiance field, and I_{src} is rendered by the stylized network.

4.2.2 Global contrastive learning

While the directional CLIP loss effectively captures the relative direction of style transformation through normalized vector embeddings, it falls short in delivering sufficient stylization intensity when modifying the pre-trained NeRF model. To enhance stylization control, we improve upon the contrastive learning framework proposed in NeRF-Art (C. Wang et al., 2022). This approach leverages the rendered view I_{tgt} as a query anchor, establishing positive samples from the target style text prompt $\boldsymbol{t}_{tgt}$, while constructing a set of negative samples $t_{neg} \in \mathcal{T}_{neg}$using text descriptions semantically distant from I_{tgt}.

The foundation of our CLIP-space contrastive loss can be expressed as:

$$\mathcal{L}_{con} = -\sum_{I_{tgt}} \log \left[\frac{\exp\left(\boldsymbol{v} \cdot \frac{\boldsymbol{v}^+}{\tau}\right)}{\exp\left(\boldsymbol{v} \cdot \frac{\boldsymbol{v}^+}{\tau}\right) + \sum_{\boldsymbol{v}^-} \exp\left(\boldsymbol{v} \cdot \frac{\boldsymbol{v}^-}{\tau}\right)} \right], \tag{6}$$

In this formulation, $\{\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-\}$ denote the query, positive sample, and negative sample vectors respectively, with τ representing the temperature parameter (maintained at 0.07 throughout our experimentation). For the global contrastive loss $\mathcal{L}_{con}^{g}$, we establish the relationship

$$\{\boldsymbol{v} = \hat{\mathcal{E}}_i(\boldsymbol{I}_{tgt}), \boldsymbol{v}^+ = \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}), \boldsymbol{v}^- = \hat{\mathcal{E}}_t(\boldsymbol{t}_{neg})\} \tag{7}$$

by considering the complete view I_{tgt} as the query anchor.

In principle, the synergy between global contrastive loss and directional CLIP loss should be complementary — the directional component establishes the trajectory of style transformation, while the contrastive element determines the magnitude of stylization along this path. However, we observed that the global contrastive loss alone proves insufficient for achieving uniform stylization across the entire NeRF scene. This limitation manifests as inconsistent stylization intensity, with some areas experiencing oversaturation while others remain understylized. The root cause likely stems from CLIP's inherent tendency to prioritize distinctive local features over holistic scene comprehension, allowing the global contrastive loss to reach minimal values despite uneven or inadequate overall stylization.

To address these spatial inconsistencies we employ a method proposed by NeRF-Art which is built upon PatchNCE loss (T. Park et al., 2020). This approach uses a local contrastive loss term $\mathcal{L}_{con}^{l}$ that operates on randomly sampled patches P_{tgt} from I_{tgt}, where

$$\{\boldsymbol{v} = \hat{\mathcal{E}}_i(\boldsymbol{P}_{tgt}), \boldsymbol{v}^+ = \hat{\mathcal{E}}_t(\boldsymbol{t}_{tgt}), \boldsymbol{v}^- = \hat{\mathcal{E}}_t(\boldsymbol{t}_{neg})\}. \tag{8}$$

The final loss function combines both global and local components:

$$\mathcal{L}_{con}^{g+l} = \lambda_g \mathcal{L}_{con}^{g} + \lambda_l \mathcal{L}_{con}^{l}. \tag{9}$$

The final training objectives are: text-driven losses (global and local contrastive loss), content preservation loss (perceptual loss (Gatys et al., 2016)), and reconstruction loss (L2 loss). We calculate the total loss as follow:

$$\mathcal{L} = \left(\mathcal{L}_{dir}^{r} + \mathcal{L}_{con}^{g+l}\right) + \lambda_p \mathcal{L}_{per} + \lambda_{l2} \mathcal{L}_{l2} \tag{10}$$

where the perceptual loss is applied between the original photorealistic rendering and the stylized neural radiance field on a pre-determined VGG layer $\psi \in \Psi$:

$$\mathcal{L}_{per} = \sum_{\mathbf{I}_{tgt}} \sum_{\psi \in \Psi} \parallel \psi(\mathbf{I}_{tgt}) - \psi(\mathbf{I}_{src}) \parallel_2^2. \tag{11}$$

4.2.3 Training

Training consists of two distinct training phases: initial model training followed by style-specific fine-tuning. In the first phase, we train our neural radiance field model to capture the scene's geometric and appearance properties using a combination of adversarial, reconstruction, and regularization losses. The second phase focuses on stylization, where we fine-tune the pre-trained model to adapt its rendering to match a target style while preserving the learned scene structure. Below, we detail the specific training procedures and hyperparameters for each phase.

We use Xavier initialization (Glorot & Bengio, 2010) to initialize the weights with gain of 0.02. During training, we employ distinct learning rates across network components: 4e-4 for the discriminator, 1e-4 for the generator, and an elevated rate of 5e-3 for processing voxel features. Our loss function combines multiple terms: perceptual and L2 losses are weighted most heavily (10.0 each), while GAN and L1 losses receive lower weights (1.0 each). To ensure proper regularization, we incorporate an opacity term weighted at 0.5 and apply a 0.05 weight to the style encoder's Kullback-Leibler divergence. We sample randomly sample 2 positions in the scene that are above the ground. One determines the position of the camera, the other indicate where the camera is directed. Ray sampling is performed with 24 points per ray, with a maximum traversal distance capped at 3 units. In our implementation, the distance metric only considers ray segments that intersect with voxels, excluding empty space traversal. Camera poses are filtered using two criteria: the depth map's mean value must exceed 2 units, and the segmentation mask's label entropy must be greater than 0.75. These thresholds ensure sufficient geometric information is encoded in the segmentation masks. This ensures that enough information is provided to the SPADE generator to produce pseudo-ground truth outputs that accurately reflect the underlying scene structure.

We then findtune the model by first rendering views from the trained network. This will be I_{src} mentioned in Section 4.2. We provide a description of I_{src} as t_{src} and the target style prompt

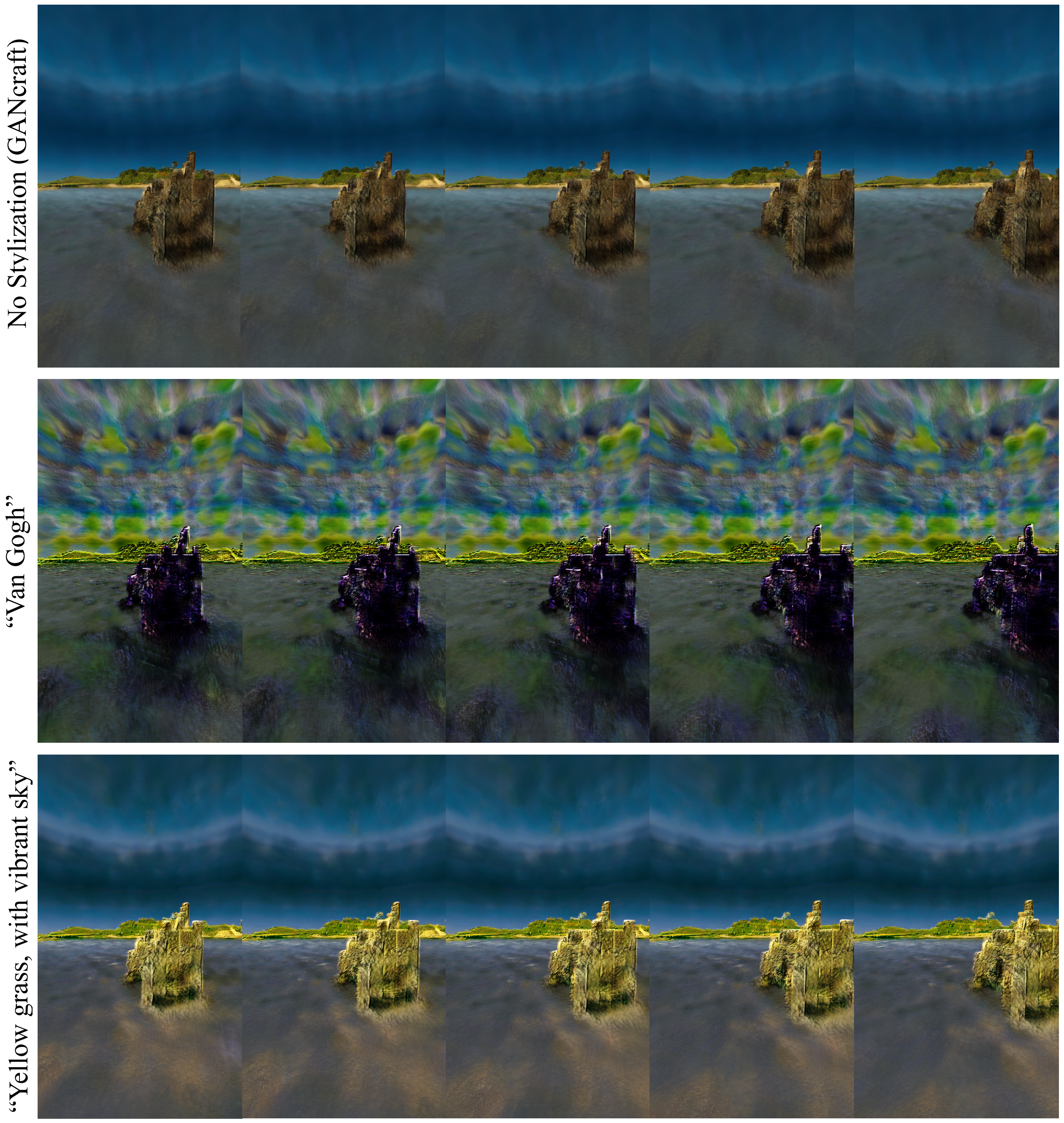


Figure 6 View consistency evaluation. *We compare two differing results, stylized with “Van Gogh” and “Yellow grass, with vibrant sky”, with the unstylized rendering in the same viewing angle. We demonstrate that the model can produce consistent views both before stylization and after stylization.*

t_{tgt}. We adopt 1.5e-4 as the learing rate for the generator and 1e-3 for learning voxel features. The weight λ_g, λ_l, λ_p and, λ_{l2} are 0.2, 0.1, 4 and, 10 respectively. For computing the local contrastive loss, patches are extracted at one-tenth the dimensions of the input image. To manage memory constraints during training, we implement a two-stage process: first, we perform a full-image render across all rays to generate a complete view, computing stylization loss gradients during forward propagation. Subsequently, we conduct patch-wise backpropagation through the neural

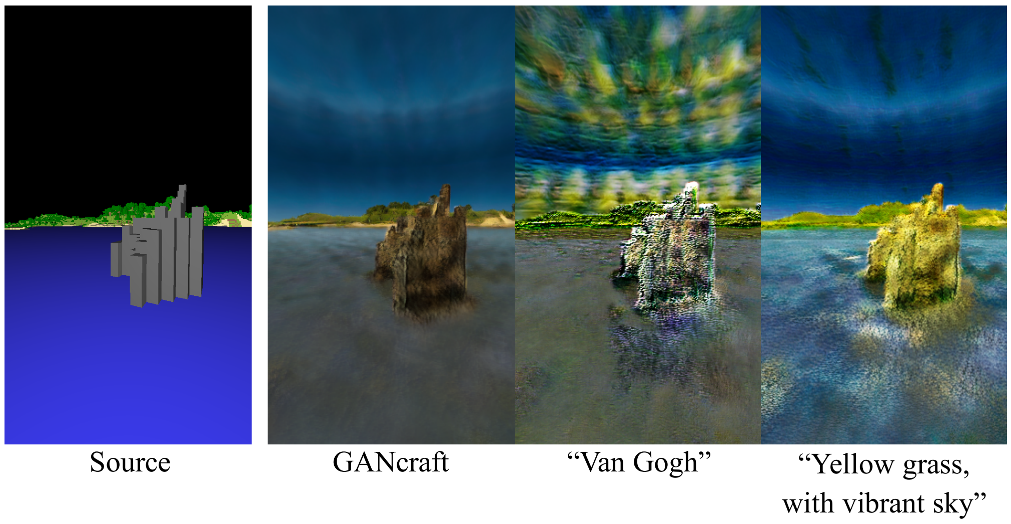


Figure 7 Stylization comparison. *We compare the results of GANcraft with two stlization results, one with the text prompt “Van Gogh” and the other with “Yellow grass, with vibrant sky”, to illustrate that while GANcraft has stylization ability, the target style is fixed. Our stylization method, on the other hand, can be guided with natural text prompts.*

radiance field model, rather than processing all rays simultaneously. This optimization strategy enables training with high-resolution imagery while substantially reducing memory overhead, ultimately leading to improved stylization quality.

5. Results and analysis

We finetune the model with two prompts on the same base model. We compare our results with GANcraft and NeRF-Art in two aspects: view consistency and stylization. Results from NeRF-Art cannot be compared due to the voxel-based nature of our model, as NeRF-Art can only be trained using ground truth images, which we lack in our condition.

5.1 View Consistency Evaluation

As can be seen in Figure 6, GANcraft can successfully translate a source view from the voxel world to a photorealistic image, while maintaining consistency. Same for our model, we produce consistent image both with no stylization and stylization.

It can be clearly distinguished in the render with “Van Gogh” stylization that the pattern of the sky remains the same regardless of the viewing angle. In addition, the rendered views demonstrate the relative position of each object i.e. the position change of the hill in the foreground of the image is differs from the position change in the sky and the coastline in the background. Both methods show tantamount ability of generation coherent view in the context of missing ground truth data. However, GANcraft can only translate the voxel-based scene to a photorealistic one, lacking the flexibility to customize the style due to the predetermined style setting in SPADE and style images fed to perceptual loss during training. To change the style outputted by GANcraft, SPADE must be trained on the desired style, which requires plethora computational power. On the contrary, our method posses the ablility stylize by using only textual guidance.

5.2 Stylization comparisons

To experiment the stylization capability of our model, we finetune our model in two styles: “Vincent Van Gogh” and “yellow grass with vibrant sky” (Figure 7). This is to test the ability of both using names to represent the target style and using general description to guide the stylization. Comparing to the rendering with no stylization, there is significant change in the appearance of the image.

In the generation using “Van Gogh” as target style, the image by large resemble the style of Vincent Van Gogh, exemplified in the unique bush stroke and swirling patterned sky. The “yellow grass with vibrant sky” rendition changes the style by brightening the whole scene and covering it with more saturated colors.

Despite the notable successes in view consistency and stylization, artifacts are often generated during the process. As shown in Figure 7, the hill in the foreground of the "Van Gogh" images has turned black. A possible explanation for this phenomenon lies in the substantial differences between the sky and ground regions. These two parts of the image exhibit drastically different color and surface characteristics, which are challenging to effectively represent or capture using a single loss function. The pronounced discrepancy between these regions makes it difficult for the loss function to balance or generalize across the distinct features, thereby leading to the observed artifacts.

6. Conclusion

In this work, we introduced GANcraft-Art, a novel, two-staged pipeline for voxel-based 3D scene generation and text-driven stylization. Using voxel worlds such as Minecraft as a starting point, our method transforms semantically labeled block environments into detailed photorealistic 3D scenes and enabling dynamic stylization through natural language input. Unlike prior approaches like GANcraft and NeRF-Art, which lack flexible style customization or require paired ground truth data, GANcraft-Art combines voxel-based neural radiance fields with CLIP-based directional and contrastive learning to achieve both view consistency and stylistic adaptability. Experimental results demonstrate our method’s ability to synthesize high-quality, coherent views across varying styles indicating substantial improvements in user-friendly and intuitive workflows. These results underscore GANcraft-Art's potential to redefine rapid prototyping and production in applications such as game design and virtual environment creation.

Despite its contributions, our method has limitations. While effective, stylization sometimes introduces artifacts, especially in regions with stark differences, such as the transition between sky and ground. This highlights challenges in balancing loss functions across diverse features. Furthermore, the quality of results heavily depends on the clarity of the textual prompt, with ambiguous or complex descriptions occasionally producing inconsistent outputs. Additionally, while patch-based training mitigates memory overhead, it introduces computational complexity and may limit scalability in larger, more detailed environments. Future work could explore improved loss balancing for artifact reduction or calculating the loss value of sky and ground separately, better disentanglement for stylistic control, and enhancements to CLIP's embedding space for handling ambiguous prompts. By addressing these challenges, GANcraft-Art could further advance the capabilities of neural rendering, facilitating seamless, user-friendly pipelines for creative 3D scene generation and stylization.

References

Aliev, K.-A., Sevastopolsky, A., Kolos, M., Ulyanov, D., & Lempitsky, V. (2020). *Neural Point-Based Graphics* (No. arXiv:1906.08240). arXiv. https://doi.org/10.48550/arXiv.1906.08240

Amanatides, J., & Woo, A. (n.d.). *A Fast Voxel Traversal Algorithm for Ray Tracing*.

Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hašan, M., Hold-Geoffroy, Y., Kriegman, D., & Ramamoorthi, R. (2020). *Neural Reflectance Fields for Appearance Acquisition* (No. arXiv:2008.03824). arXiv. https://doi.org/10.48550/arXiv.2008.03824

Boss, M., Braun, R., Jampani, V., Barron, J. T., Liu, C., & Lensch, H. P. A. (2021). *NeRD: Neural Reflectance Decomposition from Image Collections* (No. arXiv:2012.03918). arXiv. https://doi.org/10.48550/arXiv.2012.03918

Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., & Wetzstein, G. (2021). *pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis* (No. arXiv:2012.00926). arXiv. https://doi.org/10.48550/arXiv.2012.00926

Chiang, P.-Z., Tsai, M.-S., Tseng, H.-Y., Lai, W.-S., & Chiu, W.-C. (2022). Stylizing 3D Scene via Implicit Representation and HyperNetwork. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 215–224. https://doi.org/10.1109/WACV51458.2022.00029

Du, Y., Zhang, Y., Yu, H.-X., Tenenbaum, J. B., & Wu, J. (2021). *Neural Radiance Flow for 4D View Synthesis and Video Processing* (No. arXiv:2012.09790). arXiv. https://doi.org/10.48550/arXiv.2012.09790

Eric Haines. (n.d.). *Mineways, a Minecraft mapper and exporter*. Erich666/Mineways: Exports Models from Minecraft for 3D Printing or Rendering. Retrieved September 4, 2024, from https://github.com/erich666/Mineways

Gal, R., Patashnik, O., Maron, H., Chechik, G., & Cohen-Or, D. (2021). *StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators* (No. arXiv:2108.00946). arXiv. http://arxiv.org/abs/2108.00946

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423. https://doi.org/10.1109/CVPR.2016.265

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics*. https://api.semanticscholar.org/CorpusID:5575601

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks* (No. arXiv:1406.2661). arXiv. https://doi.org/10.48550/arXiv.1406.2661

Guo, M., Fathi, A., Wu, J., & Funkhouser, T. (2020). *Object-Centric Neural Scene Rendering* (No. arXiv:2012.08503). arXiv. https://doi.org/10.48550/arXiv.2012.08503

Hao, Z., Mallya, A., Belongie, S., & Liu, M.-Y. (2021). *GANcraft: Unsupervised 3D Neural Rendering of Minecraft Worlds* (No. arXiv:2104.07659). arXiv. https://doi.org/10.48550/arXiv.2104.07659

Henzler, P., Mitra, N., & Ritschel, T. (2021). *Escaping Plato's Cave: 3D Shape From Adversarial Rendering* (No. arXiv:1811.11606). arXiv. https://doi.org/10.48550/arXiv.1811.11606

Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., & Liu, Z. (2022). *AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars* (No. arXiv:2205.08535). arXiv. https://doi.org/10.48550/arXiv.2205.08535

Huang, X., & Belongie, S. (2017). *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization* (No. arXiv:1703.06868). arXiv. https://doi.org/10.48550/arXiv.1703.06868

Huang, X., Liu, M.-Y., Belongie, S., & Kautz, J. (2018, April 12). *Multimodal Unsupervised Image-to-Image Translation*. arXiv.Org. https://arxiv.org/abs/1804.04732v2

Huang, Y.-H., He, Y., Yuan, Y.-J., Lai, Y.-K., & Gao, L. (2022). StylizedNeRF: Consistent 3D Scene Stylization as Stylized NeRF via 2D-3D Mutual Learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18321–18331. https://doi.org/10.1109/CVPR52688.2022.01780

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2016, November 21). *Image-to-Image Translation with Conditional Adversarial Networks*. arXiv.Org. https://arxiv.org/abs/1611.07004v3

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). *Perceptual Losses for Real-Time Style Transfer and Super-Resolution* (No. arXiv:1603.08155). arXiv. https://doi.org/10.48550/arXiv.1603.08155

Lee, C.-H., Liu, Z., Wu, L., & Luo, P. (2019, July 27). *MaskGAN: Towards Diverse and Interactive Facial Image Manipulation*. arXiv.Org. https://arxiv.org/abs/1907.11922v2

Li, Z., Niklaus, S., Snavely, N., & Wang, O. (2021). *Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes* (No. arXiv:2011.13084). arXiv. https://doi.org/10.48550/arXiv.2011.13084

Lim, J. H., & Ye, J. C. (2017). Geometric GAN. *ArXiv*, *abs/1705.02894*. https://api.semanticscholar.org/CorpusID:9010805

Lindell, D. B., Martel, J. N. P., & Wetzstein, G. (2020, December 3). *AutoInt: Automatic Integration for Fast Neural Volume Rendering*. arXiv.Org. https://arxiv.org/abs/2012.01714v2

Liu, L., Gu, J., Zaw Lin, K., Chua, T.-S., & Theobalt, C. (2020). Neural Sparse Voxel Fields. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 15651–15663). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/b4b758962f17808746e9bb832a6fa4b8-Paper.pdf

Liu, M.-Y., Breuel, T., & Kautz, J. (2018). *Unsupervised Image-to-Image Translation Networks* (No. arXiv:1703.00848). arXiv. https://doi.org/10.48550/arXiv.1703.00848

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., & Kautz, J. (2019, May 5). *Few-Shot Unsupervised Image-to-Image Translation*. arXiv.Org. https://arxiv.org/abs/1905.01723v2

Liu, M.-Y., Huang, X., Yu, J., Wang, T.-C., & Mallya, A. (2020). *Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications* (No. arXiv:2008.02793). arXiv. https://doi.org/10.48550/arXiv.2008.02793

Liu, X., Yin, G., Shao, J., Wang, X., & Li, H. (2019, October 15). *Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis*. arXiv.Org. https://arxiv.org/abs/1910.06809v3

Martin-Brualla, R., Radwan, N., Sajjadi, M. S. M., Barron, J. T., Dosovitskiy, A., & Duckworth, D. (2021). *NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections* (No. arXiv:2008.02268). arXiv. https://doi.org/10.48550/arXiv.2008.02268

Michel, O., Bar-On, R., Liu, R., Benaim, S., & Hanocka, R. (2021). *Text2Mesh: Text-Driven Neural Stylization for Meshes* (No. arXiv:2112.03221). arXiv. https://doi.org/10.48550/arXiv.2112.03221

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis* (No. arXiv:2003.08934). arXiv. https://doi.org/10.48550/arXiv.2003.08934

Mojiang Studio. (n.d.). *What is Minecraft? Build, Discover Realms & More | Minecraft*. Retrieved November 12, 2024, from https://www.minecraft.net/en-us/about-minecraft

Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J. H., Chaitanya, C. R. A., Kaplanyan, A., & Steinberger, M. (2021, March 4). *DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks*. arXiv.Org. https://doi.org/10.1111/cgf.14340

Nguyen-Phuoc, T. H., Richardt, C., Mai, L., Yang, Y., & Mitra, N. (2020). BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 6767–6778). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/4b29fa4efe4fb7bc667c7b301b74d52d-Paper.pdf

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., & Yang, Y.-L. (2019). *HoloGAN: Unsupervised learning of 3D representations from natural images* (No. arXiv:1904.01326). arXiv. https://doi.org/10.48550/arXiv.1904.01326

Niemeyer, M., & Geiger, A. (2021). *GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields* (No. arXiv:2011.12100). arXiv. https://doi.org/10.48550/arXiv.2011.12100

Ost, J., Mannan, F., Thuerey, N., Knodt, J., & Heide, F. (2021). *Neural Scene Graphs for Dynamic Scenes* (No. arXiv:2011.10379). arXiv. https://doi.org/10.48550/arXiv.2011.10379

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., & Martin-Brualla, R. (2021). *Nerfies: Deformable Neural Radiance Fields* (No. arXiv:2011.12948). arXiv. https://doi.org/10.48550/arXiv.2011.12948

Park, T., Efros, A. A., Zhang, R., & Zhu, J.-Y. (2020). Contrastive Learning for Unpaired Image-to-Image Translation. *European Conference on Computer Vision*. https://api.semanticscholar.org/CorpusID:220871180

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). *Semantic Image Synthesis with Spatially-Adaptive Normalization* (No. arXiv:1903.07291). arXiv. https://doi.org/10.48550/arXiv.1903.07291

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., & Lischinski, D. (2021). StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2065–2074.

Pumarola, A., Corona, E., Pons-Moll, G., & Moreno-Noguer, F. (2020). *D-NeRF: Neural Radiance Fields for Dynamic Scenes* (No. arXiv:2011.13961). arXiv. https://doi.org/10.48550/arXiv.2011.13961

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision* (No. arXiv:2103.00020). arXiv. https://doi.org/10.48550/arXiv.2103.00020

Rebain, D., Jiang, W., Yazdani, S., Li, K., Yi, K. M., & Tagliasacchi, A. (2020). *DeRF: Decomposed Radiance Fields* (No. arXiv:2011.12490). arXiv. https://doi.org/10.48550/arXiv.2011.12490

Riegler, G., & Koltun, V. (2020, August 12). *Free View Synthesis*. arXiv.Org. https://arxiv.org/abs/2008.05511v1

Schwarz, K., Liao, Y., Niemeyer, M., & Geiger, A. (2021). *GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis* (No. arXiv:2007.02442). arXiv. https://doi.org/10.48550/arXiv.2007.02442

Sitzmann, V., Thies, J., Heide, F., Niessner, M., Wetzstein, G., & Zollhofer, M. (2019, June). DeepVoxels: Learning Persistent 3D Feature Embeddings. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., & Barron, J. T. (2020, December 7). *NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis*. arXiv.Org. https://arxiv.org/abs/2012.03927v1

Sushko, V., Schönfeld, E., Zhang, D., Gall, J., Schiele, B., & Khoreva, A. (2020, December 8). *You Only Need Adversarial Supervision for Semantic Image Synthesis*. arXiv.Org. https://arxiv.org/abs/2012.04781v3

Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P. P., Barron, J. T., & Ng, R. (2021). *Learned Initializations for Optimizing Coordinate-Based Neural Representations* (No. arXiv:2012.02189). arXiv. https://doi.org/10.48550/arXiv.2012.02189

Wang, C., Chai, M., He, M., Chen, D., & Liao, J. (2021). CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3825–3834.

Wang, C., Jiang, R., Chai, M., He, M., Chen, D., & Liao, J. (2022). *NeRF-Art: Text-Driven Neural Radiance Fields Stylization* (No. arXiv:2212.08070). arXiv. https://doi.org/10.48550/arXiv.2212.08070

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2017, November 30). *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*. arXiv.Org. https://arxiv.org/abs/1711.11585v2

Wiles, O., Gkioxari, G., Szeliski, R., & Johnson, J. (2020, June). SynSin: End-to-End View Synthesis From a Single Image. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xian, W., Huang, J.-B., Kopf, J., & Kim, C. (2021). *Space-time Neural Irradiance Fields for Free-Viewpoint Video* (No. arXiv:2011.12950). arXiv. https://doi.org/10.48550/arXiv.2011.12950

Yuan, W., Lv, Z., Schmidt, T., & Lovegrove, S. (2020). *STaR: Self-supervised Tracking and Reconstruction of Rigid Objects in Motion with Neural Rendering* (No. arXiv:2101.01602). arXiv. https://doi.org/10.48550/arXiv.2101.01602

Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., & Snavely, N. (2022). *ARF: Artistic Radiance Fields* (No. arXiv:2206.06360). arXiv. http://arxiv.org/abs/2206.06360

Zhang, K., Riegler, G., Snavely, N., & Koltun, V. (2020, October 15). *NeRF++: Analyzing and Improving Neural Radiance Fields*. arXiv.Org. https://arxiv.org/abs/2010.07492v2

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks* (No. arXiv:1703.10593). arXiv. https://doi.org/10.48550/arXiv.1703.10593

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017, November 30). *Toward Multimodal Image-to-Image Translation*. arXiv.Org. https://arxiv.org/abs/1711.11586v4