3D Text2LIVE

Animikh Aich, Himanshu Patil, Vedika Srivastava

Boston University
animikh@bu.edu, hipatil@bu.edu, vedikas@bu.edu

Abstract

A wide range of editing effects are now available to content creators thanks to extensive research into changing the appearance and style of objects in photographs. However, majority of the research in this field focuses on global editing rather than localized editing. To address this (Bar-Tal et al. 2022) developed an algorithm with localized editing of images using only text prompt. Given the substantial work being done on 3D objects and the widespread usage of 3D models in CAD-modeling and video games, the same flexibility and range of editing effects ought to be available in 3D. Due to this, we propose 3D Text2LIVE, which gives the same degree of creative control over the appearance and style of 3D models as can be done with 2D photographs. Code is available. [1]

Introduction

The development of classic 2D image-recognition tasks including classifications, detections, and instance segmentations marked the beginning of the deep learning era. Research in deep-learning-based computer vision has focused on fundamental 3D computer vision issues as the techniques have improved, one of the most famous of which is synthesizing new perspectives of an object and reconstructing its 3D structure from photos. In many ways, the goal is to build a system to "inflate" 3D geometry out of images after a certain number of training rounds, which is how the problem is typically approached in machine learning. Neural Radiance Fields (NeRF), however, an entirely new direction. The main distinction between a NeRF model and conventional neural networks for 3D reconstruction is that NeRF is an instance-specific implicit representation of an object. In other words, the network learns to represent the 3D object so that new views can be synthesized in a consistent manner with the training set, given a set of photos showing the same item from various angles and their related poses.

A technique for inverse rendering known as Neural Radiance Fields, or NeRF (Mildenhall et al. 2021), combines a volumetric raytracer with a neural mapping from spatial coordinates to color and volumetric density. For neural inverse rendering, NeRF has grown to be a crucial tool (Tewari et al. 2022). NeRF was initially discovered to be effective for "traditional" 3D reconstruction tasks, however NeRF has been optimized to recover the geometry of that particular scene, allowing for the synthesis of unique views of that scene from unsuspected angles. Incorporating NeRF-like models as a building component inside a broader generating system has proven successful for several 3D generative techniques (Schwarz et al. 2020; Chan et al. 2021, 2022; Gu et al. 2021; Liu et al. 2022). One such method, DreamFusion (Poole et al. 2022), which first uses a pre-trained 2D diffusion-based model called Imagen to generate various views of an image conditioned on the input text, then follows it up with a loss based on probability density distillation that makes it possible to use a 2D diffusion model as a prior for optimizing a parametric image generator. A randomly initialized 3D model (NeRF) is optimized using this loss using gradient descent so that its 2D renderings from arbitrary angles have a low loss.

In Text2Live (Bar-Tal et al. 2022), they provide straightforward text cues to communicate the goal edit, drawing on the revolutionary capability of contemporary Vision-Language models. This enables the user to specify the object/region to be modified as well as the intended appearance simply and intuitively. Our approach specifically permits local semantic editing that meets a predetermined target text prompt. We use a Contrastive Language Image Pretraining (CLIP) (Sanghi et al. 2022; Jetchev 2021; Wang et al. 2022) model that has been pre-trained on 400 million samples of text and images. The majority of known techniques combine CLIP with a pre-trained generator (such as a GAN or a Diffusion model).

Dream Fields (Jain et al. 2022) is one such method that employs frozen image-text joint embedding models from CLIP and an optimization-based method to train NeRFs. This work demonstrated that pretrained 2D image-text models may be utilized for 3D synthesis, although the 3D objects created by this method tended to be inaccurate and unreal. Other voxel grids and mesh-based techniques have been guided by CLIP (Sanghi et al. 2022; Jetchev 2021; Wang et al. 2022). The approach we suggest goes a step further by providing 3D mesh manipulation with a 2D image and a text prompt for the necessary adjustments.

Related Work

(Mildenhall et al. 2020) Encoding objects and scenes in the weights of an MLP that directly maps from a 3D spatial lo-

[1] *https://github.com/animikhaich/3D-Text2LIVE*

cation to an implicit representation of the shape, such as the signed distance at that location, is a promising recent direction in computer vision. However, these methods have yet to reproduce realistic scenes with complex geometry with the same fidelity as techniques that use discrete representations such as triangle meshes or voxel grids. MLPs have been used to map low-dimensional coordinates to colors, as well as images, textured materials, and indirect illumination values.

Neural 3D shape representations: Recent research has looked into the implicit representation of continuous 3D shapes as level sets using deep networks that map xyz coordinates to signed distance functions or occupancy fields. However, these models are constrained by their need for access to ground truth 3D geometry, which is typically obtained from synthetic 3D shape data-sets like ShapeNet. (Niemeyer et al. 2019) represent surfaces as 3D occupancy fields and use a numerical method to find the surface intersection for each ray before calculating an exact derivative using implicit differentiation. Each ray intersection point is fed into a neural 3D texture field, which predicts the point's diffuse color. (Sitzmann, Zollhoefer, and Wetzstein 2019) propose a differentiable rendering function comprised of a recurrent neural network that marches along each ray to determine the location of the surface, employing a less direct neural 3D representation that simply outputs a feature vector and RGB color at each continuous 3D coordinate. Though these techniques can potentially represent complicated and high resolution geometry, they have so far been limited to simple shapes with low geometric complexity, resulting in over-smoothed renderings.

View synthesis and image-based rendering: Photo-realistic novel views can be reconstructed using simple light field sample interpolation techniques given a dense sampling of views. The computer vision and graphics communities have made significant progress in predicting traditional geometry and appearance representations from observed images for novel view synthesis with sparser view sampling. One popular approach employs mesh-based representations of scenes that have either a diffuse or a view-dependent appearance. Differentiable rasterizers or path tracers can use gradient descent to directly optimize mesh representations in order to reproduce a set of input images. However, gradient-based mesh optimization based on image re-projection is frequently difficult due to local minima or poor loss landscape conditioning.

In a different family of techniques, volumetric representations are used to synthesize high-quality photo-realistic views from a group of RGB input photos. Volumetric methods typically yield fewer visually distracting artifacts than mesh-based ones, are better suited for gradient-based optimization, and are able to realistically represent complicated forms and materials. Early volumetric methods directly colored voxel grids using observed images. Deep networks that anticipate a sampled volumetric representation from a series of input photos have been trained more recently using vast data-sets of different scenes. At test time, these networks may render unique perspectives using either alpha-compositing or learnt compositing along rays.For each unique scene, convolution networks (CNNs) and sampled voxel grids have been adjusted in other studies such that the CNN can account for discretization artifacts from low resolution voxel grids or permit the predicted voxel grids to change depending on input time or animation settings.

Methods

In this work, we provide a methodology for turning images from different views with a text-based change into a 3D model. We also outline our prior approach and its associated challenges as was originally proposed.

Current Methodology

Our concept consists of a generator G which takes input as I_s and outputs an edit layer $I_{edit} = \{C, \alpha\}$ consisting of color image C and opacity channel α. The final output image from generator G is shown in Equation 1.

$$I_o = \alpha.C + (1 - \alpha).I_s \tag{1}$$

Generating the RGBA layer for edits allows us to guide generated content and localize it with more precision. Along with this we also split the text in two parts: T_{screen}, which helps with the generation of content in the edit layer, and T_{ROI}, which helps in localizing the generated content based on Region of Interest. For example, the text prompt is T = *"person smoking cigar from his mouth"* can be split as T_{screen} = *smoking cigar over a green screen* and T_{ROI} = *mouth*. Similar to (Bar-Tal et al. 2022), we use three main loss term,

- L_{comp} is a composition loss that enforces I_o to match up to T and is a combination of cosine distance and direction loss.
- L_{screen} is used to have direct supervision of text on generated edit layer I_{edit}
- In image manipulation, we allow significant texture and appearance changes, but along with this we want to preserve the object's spatial structure. For this we use $L_{structure}$

The total loss is a linear combination of L_{comp}, L_{screen}, and $L_{structure}$ as shown in Equation 2.

$$L = L_{comp} + \lambda_1 L_{screen} + \lambda_2 L_{structure} \tag{2}$$

The images obtained from the generator module, is then augmented to get various spatial view and using NeRF (Neural Radiance Fields), we output a 3D model. In NeRF, a static scene is depicted as a continuous 5D function that outputs the radiance emitted in each direction (θ, ϕ) at each location (x, y, z) in space. At each position, a density (ρ) that functions as a differential opacity controls how much radiance is accumulated by a ray passing through (x, y, z). A deep fully-connected neural network (MLP) is optimized to represent this function by regressing from a single 5D coordinate (x, y, z, θ, ϕ) to a single volume density (ρ) and view-dependent RGB color. Further, depending on Density, NeRF uses coarse (low density) and fine (high density) renderings. For both coarse and fine renderings, the loss is the sum of the squared errors between the rendered and true pixel colors as shown in Equation 3.

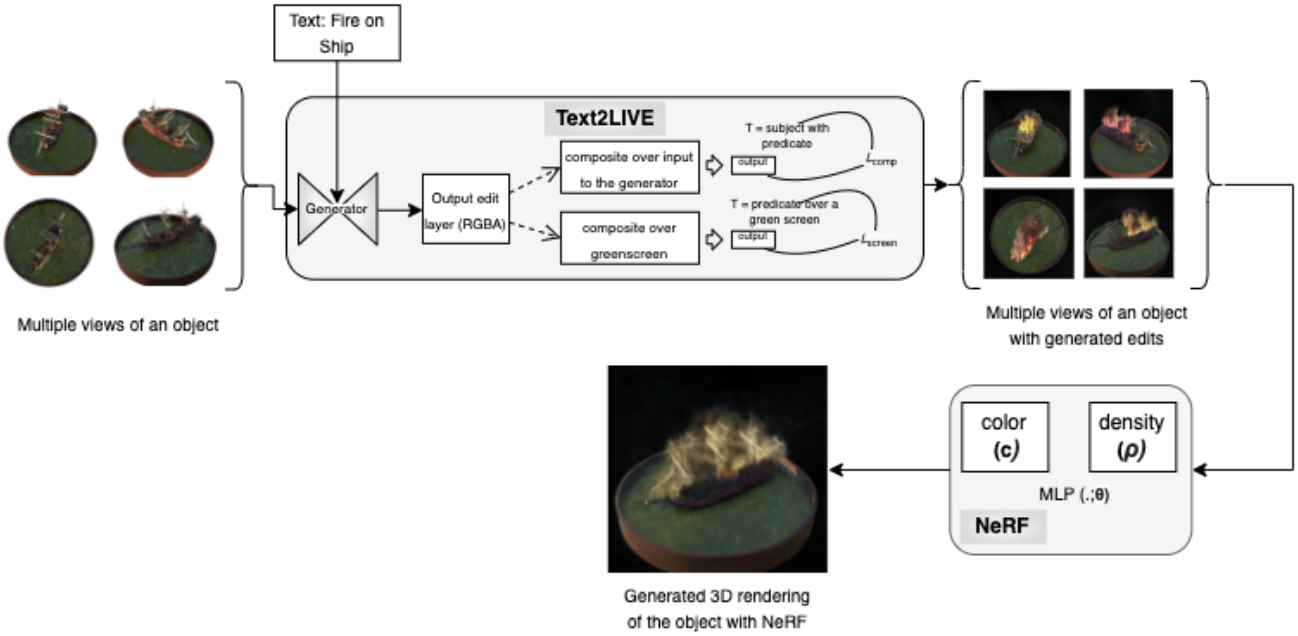

Figure 1: Multiple views of an object is given as input to Text2LIVE (Bar-Tal et al. 2022) along with the text input describing the edit layer to be applied on top. Upon generating the corresponding edits, they are given as inputs along with the corresponding camera angles to NeRF (Mildenhall et al. 2021) to generate the 3D point cloud rendering of the object with the edit layers.

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right] \quad (3)$$

Combining Text2LIVE with NeRF, we have Figure 2. First, multiple views of the same object with known camera angles are taken along with the text describing the edit to be applied, and are given as input to Text2LIVE while preserving the RGBA channels. The model then generates the respective views with the additional edit layer on top. Second, these generated images, along with their corresponding camera view angles are given as input to NeRF which generates the 3D rendered point cloud of the input images.

Evaluating Metrics

For numerical evaluation of the generated 3D samples, we use PSNR which is peak signal-to-noise ratio. It is defined by equation 5 and the mean squared error (MSE) as given by equation 4, where $m \times n$ is a noise-free monochrome image I and K is its noisy approximation.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (4)$$

$$PSNR = 20 \cdot log_{10}(MAX_I) - 10 \cdot log_{10}(MSE) \quad (5)$$

Prior Approach

As outlined in Figure 2, from (Poole et al. 2022) We consider starting point as P, which is the projection of a 3D model from a particular view. At P, we pass an image on which we need edits. Gaussian noise is added to this image I_s to form $I_s + \epsilon$ and is passed to the Text2LIVE (Bar-Tal et al. 2022) generator with a text prompt for the edit. The approximation from the output edit layer of Generator output an edit layer I_{edit} which can then be composed with image I_s which is further mixed with Gaussian noise to generate a noisy image $\epsilon_\phi(z_t|y;t)$. The original noise component ϵ is then subtracted from the resulting noisy image to form $\epsilon_\phi(z_t|y;t) - \epsilon$. Using the composed image, NERF is trained to approximate these images by computing density(ρ) and albedo(τ). From this shape, normals (n) are calculated, and then, using a random light source (l), shading is applied ($n.l$). A Color 3D model is generated and a random 2D projection is taken as P and repeated the process till we get the proper 3D object with edits on it.

However, this approach had two primary challenges. First, since Text2LIVE uses CLIP (Wang et al. 2022) and (Sanghi et al. 2022) to condition and generate images, it was not feasible to directly extrapolate the approach used in DreamFusion (Poole et al. 2022) using Diffusion from to approximate a noisy image ($\epsilon_\phi(z_t|y;t)$). Second, the GPU Memory (VRAM) required to train the combined models exceeded what was available.

Hence, we modified our approach to train the model in stages (as detailed in the previous subsection), by first generating multiple modified views of an image with edit lay-

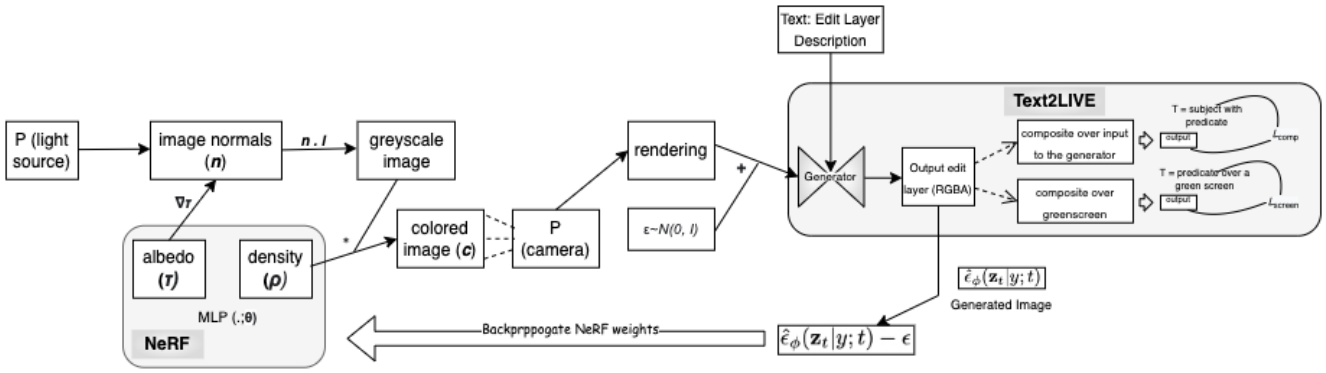

Figure 2: 3D Text2LIVE Architecture inspired from DreamFusion (Poole et al. 2022) and Text2LIVE (Bar-Tal et al. 2022). Consider starting point as P, which is the projection of a 3D model from a particular view. At P, we first pass an image on which we need edits. Gaussian noise is added to this image I_s and is passed to the generator with a text prompt for the edit. Generator output an edit layer I_{edit} which can then be composed with image I_s. Using the composed image, NERF is trained to approximate these images by computing density(ρ) and albedo(τ). From this shape, normals (n) are calculated, and then, using a random light source (l), shading is applied ($n.l$). A Color 3D model is generated and a random 2D projection is taken as P and repeated the process till we get the proper 3D object with edits on it.

Algorithm 1: 3D Text2LIVE

Image edits in Text2LIVE

Input: Multi-view Data $\mathcal{D}^{\theta,\phi}$; ϕ is angle of image. Maximum Iteration $Iter^{max}$; G is network.
Initialization: Creating internal dataset with augmentation.

for $d \in D^{\theta,\phi}$ **do**
  $I_{edit} = G(d)$ ▷ *get edit layer from generator*
  $I_o = \alpha.C + (1-\alpha).I_s$ ▷ *create a composite image using edit layer and main image*
end for

3D model construction

Input: Images obtained from above, along with their angles. Let this data be I_ϕ. MLP is a fully connected network.

for $(I_o, \theta, \phi) \in I_\phi$ **do** ▷ *Sample a image and angles from the dataset above*
  for pixel $\in I_o$ **do**
    $(x, y, z) \sim pixel$
    $MLP(x, y, z, \theta, \phi)$ ▷ *Train the model with 5D continuous input*
  end for
end for

Output: The volume density and view-dependent emitted radiance at that spatial location

ers, and then followed by applying NeRF to generate the 3D view from the generated outputs, as shown in Figure 1.

Results

Both Text2LIVE and DreamFusion require partial training in order to generate images since both of these models overfit on the given image space prior to generating the transformed 2D image or 3D rendering. As such, we utilize Google Cloud Platform (GCP) for the model training due to the high GPU VRAM requirement of either of the models.

- **Text2LIVE:** We trained Text2LIVE on a single A100 GPU for 1000 Epochs with an input of a cake and a text prompt "Cake with grass/fish/ice" as shown in Figure 3. The model generated a cake made of grass (3b). However, upon closer inspection, the edges of the cake are softened and are blending with the background. Further, the eggs in the background have also been colored green, which is unexpected. A similar outcome is observed for a cake with "fish" (3c). We observe better results in 3d, where the model is able to generate a cake made of "ice" with nearly flawless execution.
  With the change in our approach, the dataset also changed. We now require dataset with images taken from different camera angles. So we tested Text2LIVE on nerf dataset specifically on ship. When given the prompt "fire on the ship" along with different views of the ship, the model successfully generated a burning ship on fire with realistic flames and smoke (Figure 4). However, the same prompt and similar images have also generated some failure cases with white/blue/red flames that does not resemble a real fire (Figure 5).
- **NeRF:** Preliminary findings are shown in fig. This hazy

(a) Original cake image (b) Cake with "grass"

(c) Cake with "fish" (d) Cake with "ice"

Figure 3: Generated overlays of different materials on a cake using Text2LIVE. Subfigure (a) shows the original image of the cake. Subfigures (b), (c) and (d) shows the generated images for "grass", "fish" and "ice" respectively.

image was a result of applying the same hyperparameters as NeRF. We are conducting trials with hyperparameter adjustment, and the results are better (fig). You can view the 3D model video in the GitHub repository.

- The numerical evaluating metric (PSNR) for each of our experiments is given by Table 1.

Table 1: PSNR for experiments

Experiment	PSNR
Ship on Fire	28.11
Ice Chair	28.72
Golden Hotdog	29.79

Conclusion

In our proposal, we had proposed an approach to generate 2D overlays over an existing 2D image and further generate a 3D render of the image with overlay. To do this, we had proposed replacing the Imagen model of DreamFusion with the Diffusion model used in Text2LIVE (Bar-Tal et al. 2022). However, based on our experimentation, we conclude that the same likely is challenging. This is because DreamFusion (Poole et al. 2022) assumes the presence of a Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, which is a major component in the Imagen diffusion model and back propagating the losses onto NeRF (Mildenhall et al. 2021) weights, which is absent in the model used in Text2LIVE.

Hence, we have modified our approach to generate 3D point cloud of the edited image views with NeRF by directly feeding in the output Text2LIVE to NeRF. Since the models are individually trained, this is also computationally more feasible than the original approach.

The current generated NeRF 3D renderings are currently lacking in consistency and suffer from certain artifacts due to inconsistently in generated edits by Text2LIVE for different views of the object. A potential solution to this is to condition the Text2LIVE to generate consistent edits for each view of the same object. This can be considered for a future extension of our work.

Figure 4: Correctly generated edits for the prompt: Fire on ship

References

Bar-Tal, O.; Ofri-Amar, D.; Fridman, R.; Kasten, Y.; and Dekel, T. 2022. Text2LIVE: Text-Driven Layered Image and Video Editing. *arXiv preprint arXiv:2204.02491*.

Chan, E. R.; Lin, C. Z.; Chan, M. A.; Nagano, K.; Pan, B.; De Mello, S.; Gallo, O.; Guibas, L. J.; Tremblay, J.; Khamis, S.; et al. 2022. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16123–16133.

Chan, E. R.; Monteiro, M.; Kellnhofer, P.; Wu, J.; and Wetzstein, G. 2021. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5799–5809.

Gu, J.; Liu, L.; Wang, P.; and Theobalt, C. 2021. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*.

Figure 5: Incorrectly generated edits for the prompt: Fire on ship

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

Jetchev, N. 2021. Clipmatrix: Text-controlled creation of 3d textured meshes. *arXiv preprint arXiv:2109.12922*.

Liu, Z.; Wang, Y.; Qi, X.; and Fu, C.-W. 2022. Towards Implicit Text-Guided 3D Shape Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17896–17906.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Niemeyer, M.; Mescheder, L.; Oechsle, M.; and Geiger, A. 2019. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*.

Sanghi, A.; Chu, H.; Lambourne, J. G.; Wang, Y.; Cheng, C.-Y.; Fumero, M.; and Malekshan, K. R. 2022. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18603–18613.

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

Sitzmann, V.; Zollhoefer, M.; and Wetzstein, G. 2019. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Tewari, A.; Thies, J.; Mildenhall, B.; Srinivasan, P.; Tretschk, E.; Yifan, W.; Lassner, C.; Sitzmann, V.; Martin-Brualla, R.; Lombardi, S.; et al. 2022. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, 703–735. Wiley Online Library.

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