SINE: Semantic-driven Image-based NeRF Editing with Prior-guided Editing Field

Chong Bao[1*] Yinda Zhang[2*] Bangbang Yang[1*] Tianxing Fan[1]
Zesong Yang[1] Hujun Bao[1] Guofeng Zhang[1†] Zhaopeng Cui[1†]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Google

https://zju3dv.github.io/sine/

Figure 1. We propose a novel semantic-driven image-based editing approach, which allows users to edit a photo-realistic NeRF with a single-view image or with text prompts, and renders edited novel views with vivid details and multi-view consistency.

Abstract

Despite the great success in 2D editing using user-friendly tools, such as Photoshop, semantic strokes, or even text prompts, similar capabilities in 3D areas are still limited, either relying on 3D modeling skills or allowing editing within only a few categories. In this paper, we present a novel semantic-driven NeRF editing approach, which enables users to edit a neural radiance field with a single image, and faithfully delivers edited novel views with high fidelity and multi-view consistency. To achieve this goal, we propose a prior-guided editing field to encode fine-grained geometric and texture editing in 3D space, and develop a series of techniques to aid the editing process, including cyclic constraints with a proxy mesh to facilitate geometric supervision, a color compositing mechanism to stabilize semantic-driven texture editing, and a feature-cluster-based regularization to preserve the irrelevant content unchanged. Extensive experiments and editing examples on both real-world and synthetic data demonstrate that our method achieves photo-realistic 3D editing using only a single edited image, pushing the bound of semantic-driven editing in 3D real-world scenes.

1. Introduction

Semantic-driven editing approaches, such as stroke-based scene editing [34, 39, 66], text-driven image synthesis and editing [1, 50, 53], and attribute-based face editing [27, 60], have greatly improved the ease of artistic creation. However, despite the great success of 2D image editing and neural rendering techniques [14, 42], similar editing abilities in the 3D area are still limited: **(1)** they require laborious annotation such as image masks [27, 71] and mesh vertices [69, 74] to achieve the desired manipulation; **(2)** they conduct global style transfer [12, 13, 16, 20, 75] while ignoring the semantic meaning of each object part (*e.g.*, windows and tires of a vehicle should be textured differently); **(3)** they can edit on categories by learning a textured 3D latent representation (*e.g.*, 3D-aware GANs with faces and cars *etc.*) [6, 8, 9, 17, 45, 56, 59, 60], or at a coarse level [35, 64] with basic color assignment or object-level disentanglement [30], but struggle to conduct texture editing on objects with photo-realistic textures or out-of-distribution characteristics.

Based on this observation, we believe that, on the way toward semantic-driven 3D editing, the following properties should be ensured. First, the operation of editing should be effortless, *i.e.*, users can edit 3D scenes on a single 2D image in convenient ways, *e.g.*, using off-the-shelf tools such as GAN-based editing [28, 34], text-driven editing [1, 53], Photoshop, or even a downloaded Internet image without pixel-wise alignment, rather than steering 3D modeling software with specific knowledge [69], or repeatedly editing from multi-view images. Second, the editing method should be applicable to real-world scenes or objects and preserve vivid appearances, which is beyond existing 3D-aware generative models [8, 9] due to the limited categories and insufficient data diversity on real-world objects.

To fulfill this goal, we propose a novel **S**emantic-driven **I**mage-based **E**diting approach for **N**eural radiance field in real-world scenes, named SINE. Specifically, our method allows users to edit a neural radiance field with a sin-

[*]Authors contributed equally.
[†]Corresponding authors.

gle image, *i.e.*, either by changing a rendered image using off-the-shelf image editing tools or providing an image for texture transferring (see Sec. 4.4), and then delivers edited novel views with consistent semantic meaning. Unlike previous works that directly fine-tune the existing NeRF model [30, 35, 64], SINE learns a prior-guided editing field to encode geometric and texture changes over the original 3D scene (see Fig. 2), thus enabling fine-grained editing ability. By leveraging guidance from existing neural priors (shape prior models [15] and Vision Transformer models [7], *etc.*), SINE can directly perform semantic-driven editing on photo-realistic scenes without pre-training a category-level latent space. For example, in Fig. 1, users can stretch a car's back or change all four tires to cookies by only editing a single image, and can even cooperate with text-prompts editing [1] to modify a specific object of a scene with vivid appearances.

However, even when guided with neural priors, editing NeRF from a single image with multi-view consistency and accuracy is still challenging. **(1)** The generic NeRF does not necessarily provide an explicit surface or signed distance field, such that it cannot directly work with shape priors [15]. Therefore, we propose to use cyclic constraints with a proxy mesh to represent the edited NeRF's geometry, which facilitates guided editing using coarse shape prior. **(2)** Learning a coordinate-based 3D editing field using a single edited view is not sufficient to capture fine-grained details, and applying semantic supervision [7, 52] directly to the editing field leads to sub-optimal convergence (see Sec. 4.5). To tackle these challenges, we propose a color compositing mechanism by first rendering the template NeRF color and modification color individually, and then deferred blending them to yield the edited view, which significantly improves semantic-driven texture editing. **(3)** Ideally, a user's editing should only affect the desired regions while maintaining other parts untouched. However, in semantic-driven editing, the prior losses require taking the full shape or image as input, which leads to appearance or shape drifting at the undesired area. To precisely control the editing while excluding irrelevant parts from being affected, we generate feature clusters of the editing area using the ViT-based feature field [7, 30], and use these clusters to distinguish whether a location is allowed to be edited or should remain unchanged.

In summary, the contributions of our paper are as follows. **(1)** We propose a novel semantic-driven image-based NeRF editing approach, called SINE, which allows users to edit a neural radiance field simply on just a single view of the rendering. SINE leverages a prior-guided editing field to encode fine-grained geometry and texture changes over the given pre-trained NeRF, thus delivering multi-view consistent edited views with high fidelity. **(2)** To achieve semantic editing functionality, we develop a series of techniques, including cyclic constraints with a proxy mesh for geometric editing, the color compositing mechanism to enhance texture editing, and the feature-cluster-based regularization to control the affected editing area and maintain irrelevant parts unchanged. **(3)** Experiments and editing examples on both real-world/synthetic and object-centric/unbounded 360° scenes data demonstrate superior editing capabilities and quality with effortless operations.

2. Related Works

Neural rendering with external priors. Neural rendering techniques aim at rendering novel views with high-quality [42] or controllable properties [27, 48] by learning from 2D photo capture. Recently, NeRF [42] achieves photo-realistic rendering with volume rendering and inspires many works, including surface reconstruction [31, 65, 73], scene editing [4, 18, 67, 70, 71] and generation [22, 51], inverse rendering [5, 76], SLAM [72, 77], *etc*. For learning from few-shot images [23] or 3D inpainting [43], NeRF's variants use hand-crafted losses [44] or large language-image models [23, 68] as external priors. However, due to insufficient 3D supervision, such methods cannot reconstruct accurate geometry and only produce visually plausible results. Besides, some works [21, 32, 40] use the symmetric assumption to reconstruct category-level objects (*e.g.*, cars, chairs) but cannot generalize on complex scenes.

Neural 2D & 3D scene editing. With the development of neural networks, semantic-driven 2D photo editing allows user editing in various friendly ways, such as controlling attribute of faces [19, 28], stroke-based editing [34, 39, 66], sketch-to-image generation [11, 55], image-to-image texture transferring [62], or text-driven image generation [53] and editing [29]. Nevertheless, in 3D scene editing, similar capabilities are still limited due to the high demand for multi-view consistency. Existing approaches either rely on laborious annotation [27, 69, 71, 74], only support object deformation or translation [30, 61, 63, 74], or only perform global style transfer [12, 13, 16, 20, 75] without strong semantic meaning. Recently, 3D-aware GANs [8, 9, 17, 24, 45, 56, 59] and semantic NeRF editing [35, 64] learn a latent space of the category and enable editing via latent code control. However, the quality and editing ability of these methods mainly depend on the dataset (*e.g.*, human faces [59, 60] or objects in ShapeNet [10]), and they cannot generalize to objects with rich appearances or out-of-distribution features [1]. In contrast, our method allows for semantic-driven editing directly on the given photo-realistic NeRF, and uses a prior-guided editing field to learn fine-grained editing from only a single image.

3. Method

We first formulate the goal of our semantic NeRF editing task as follows. As illustrated in the left part of Fig. 2,

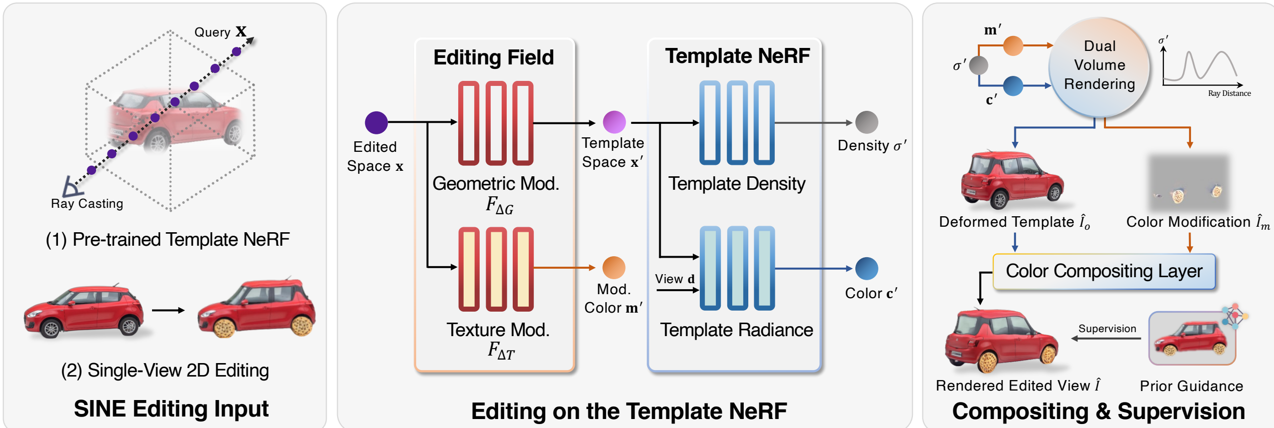


Figure 2. **Overview.** We encode geometric and texture changes over the original template NeRF with a prior-guided editing field, where the geometric modification field $F_{\Delta G}$ transformed the edited space query $\mathbf{x}$ into the template space $\mathbf{x}'$, and the texture modification field $F_{\Delta T}$ encodes modification colors $\mathbf{m}'$. Then, we render deformed template image $\hat{I}_o$ and color modification image $\hat{I}_m$ with all the queries, and use a color compositing layer to blend $\hat{I}_o$ and $\hat{I}_m$ into the edited view $\hat{I}$.

given a pre-trained NeRF of a photo-realistic scene (named template NeRF), we aim at editing the template NeRF using only a single-view 2D image, and then produce novel views with consistent semantic meaning (see Sec. 4.3 and Sec. 4.4). Note that naïvely fine-tuning on the edited single view cannot obtain satisfactory results due to the spatial ambiguity and lack of multi-view supervision (see Sec. 4.2). Therefore, we propose to use a novel prior-guided editing field to encode fine-grained changes (Sec. 3.1) in 3D space, which leverages geometry and texture priors to guide the learning of semantic-driven editing (Sec. 3.2 and Sec. 3.3). Besides, to precisely control the editing area while maintaining other parts unchanged, we design editing regularization with feature cluster-based semantic masking (Sec. 3.4).

3.1. SINE Rendering Pipeline

As illustrated in Fig. 2, we use a dedicated editing field to encode geometry and texture changes over the pre-trained template NeRF. The editing field consists of an implicit geometric modification field $F_{\Delta G}$ and a texture modification field $F_{\Delta T}$, where $F_{\Delta G}$ deforms the query points from the observed edited space to the original template space, as $\mathbf{x}' := F_{\Delta G}(\mathbf{x})$, and $F_{\Delta T}$ encodes the modification color $\mathbf{m}'$, as $\mathbf{m}' := F_{\Delta T}(\mathbf{x})$. Specifically, for each sampled query point $\{\mathbf{x}_i | i = 1, ..., N\}$ along the ray $\boldsymbol{r}$ with view direction $\mathbf{d}$, we first obtain the deformed points $\mathbf{x}'$ (in template space) and modification color $\mathbf{m}'$, and feed $\mathbf{x}'$ and $\mathbf{d}$ to the template NeRF to obtain the density δ' and template colors $\mathbf{c}'$. Then, we perform dual volume rendering both on edited fields and template NeRF following the quadrature rules [38, 42], which is defined as:

$$\hat{C}_o(\boldsymbol{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}'_i, \quad \hat{C}_e(\boldsymbol{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{m}'_i, \qquad T_i = \exp\left(-\sum_{j=1}^{i-1} {\sigma'}_j \delta_j\right), \tag{1}$$

where $\alpha_i = 1 - \exp\left(-{\sigma'}_i \delta_i\right)$, and δ_i is the distance between adjacent samples along the ray. In this way, we obtain the deformed template image $\hat{I}_o$ from the template NeRF's pixel color $\hat{C}_o(\boldsymbol{r})$ and color modification image $\hat{I}_m$ from the modification color $\hat{C}_e(\boldsymbol{r})$. Finally, we apply the color compositing layer (see Sec. 3.3) to blend $\hat{I}_o$ and $\hat{I}_m$ into the resulting edited views $\hat{I}$.

3.2. Prior-Guided Geometric Editing

In this section, we explain how to learn $F_{\Delta G}(\mathbf{x})$ with the geometric prior.

Shape prior constraint on the edited NeRF. We leverage geometric prior models, such as neural implicit shape representation [15, 46] or depth prediction [3], to mitigate the ambiguity of geometric editing based on editing from a single perspective. **(1)** For objects within a certain shape category (*e.g.*, cars, airplanes), we use DIF [15], in which the implicit SDF field and the prior mesh $\hat{M}_P$ can be generated with the condition of an optimizable latent code $\hat{\mathbf{z}}$. We force the edited NeRF's geometry $\hat{M}_E$ to be explainable by a pre-trained DIF model with the geometric prior loss:

$$\begin{aligned}\mathcal{L}_{gp} = \min_{\hat{\mathbf{z}}} \Big(\sum_{\mathbf{p}' \in \hat{M}_E} & f_{\text{SDF}}(\hat{\mathbf{z}}, \mathbf{p}') + \lambda ||\hat{\mathbf{z}}||_2^2 \Big) \\ &+ \sum_{\mathbf{p}'_i \in \hat{M}_E} \min_{\mathbf{p}_t \in \hat{M}_P} ||\mathbf{p}'_i - \mathbf{p}_t||_2^2 \\ &+ \sum_{\mathbf{p}_i \in \hat{M}_P} \min_{\mathbf{p}'_t \in \hat{M}_E} ||\mathbf{p}_i - \mathbf{p}'_t||_2^2.\end{aligned} \tag{2}$$

The first term encourages the sampled surface points on the edited NeRF's geometry $\hat{M}_E$ to lie on the manifold of DIF's latent space with an SDF loss f_{SDF} and the latent code regularization [15]. The last two terms are Chamfer constraints, which enforce the $\hat{M}_E$ close to the DIF's periodically updated prior mesh $\hat{M}_P$ [36] by minimizing the closest surface points. **(2)** For objects without a category-level prior, we can build a finalized shape prior $\hat{M}_P$ beforehand. Practically, we find 3D deforming vertices with 2D correspondence [26] and monocular depth prediction [3], and use ARAP [57] to deform the proxy triangle mesh M_Θ to $\hat{M}_P$. Then, we can inherit the Chamfer loss term in Eq. (2) for prior-guided supervision.

Representing edited NeRF's geometry as a deformed proxy mesh. The edited NeRF has no explicit surface definition or SDF field to directly apply the geometric prior loss (Eq. (2)). Therefore, to obtain the edited mesh surface $\hat{M}_E$, as illustrated in Fig. 3 (a), we first fit the template NeRF geometry with a proxy mesh M_Θ [36, 65], and then learn a forward modification field $F'_{\Delta G}$ to warp the template proxy mesh to the edited space. $F'_{\Delta G}$ is an inverse of the editing field $F_{\Delta G}$, which maps from the template space to the query space [33, 41], as $\mathbf{x} := F'_{\Delta G}(\mathbf{x}')$, and can be supervised using a cycle loss $\mathcal{L}_{\text{cyc}}$ (see the supplementary material for details). Note that the deformed mesh proxy might not reflect fine-grained details of the specific shape identity. It facilitates applying shape priors to the edited field and provides essential guidance during geometric editing.

Learning geometric editing with users' 2D editing. The goal of geometric editing is to deform the given NeRF according to the edited target image while satisfying semantic properties. To this end, apart from the geometric prior loss in Eq. (2), we add the following geometric editing loss in two folds. **(1)** We encourage the edited NeRF to satisfy the user's edited image by directly supervising rendering colors and opacity on N_r rays, which is defined as:

$$\mathcal{L}_{\text{gt}} = \frac{1}{|N_r|} \sum_{\boldsymbol{r} \in N_r} ||\hat{C}(\boldsymbol{r}) - C_t(\boldsymbol{r})||_2^2 + \text{BCE}(\hat{O}(\boldsymbol{r}), O_e(\boldsymbol{r})). \tag{3}$$

The first photometric loss term encourages the rendered color $\hat{C}(\boldsymbol{r})$ close to the edited target color $C_t(\boldsymbol{r})$. The second silhouette loss term enforces the rendered opacity $\hat{O}(\boldsymbol{r})$ close to the edited object's silhouette $O_e(\boldsymbol{r})$ (derived from users' editing tools) by minimizing the binary cross-entropy loss, where $\hat{O}(\boldsymbol{r}) = \sum_{i=1}^{N} T_i \alpha_i$. **(2)** To obtain a spatially smooth deformation and mitigate overfitting to the mesh's surface points, inspired by previous works [15, 48, 49], we also add deformation regularization as:

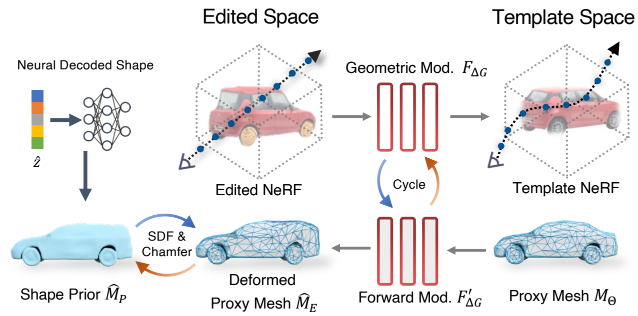
$$\mathcal{L}_{\text{gr}} = \frac{1}{M} \sum_{i=1}^{N} ||\nabla F_{\Delta G}(\mathbf{p}_i)||_2 + ||F_{\Delta G}(\mathbf{p}_i) - F_{\Delta G}(\mathbf{p}_i + \boldsymbol{\epsilon})||_1, \tag{4}$$

where the first term penalizes the spatial gradient of the geometric editing, and the second term encourages the editing to be smooth under a mild 3D positional jitter $\boldsymbol{\epsilon}$.

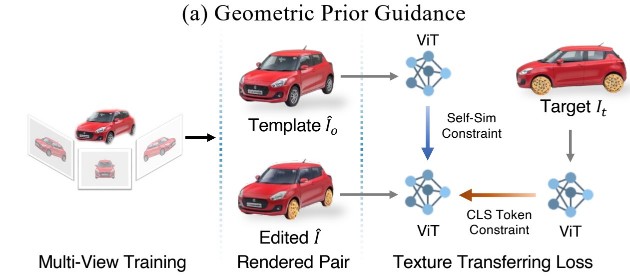
The overall geometric editing loss is defined as:

$$\mathcal{L}_{\text{geo}} = \lambda_{\text{gp}} \mathcal{L}_{\text{gp}} + \mathcal{L}_{\text{gt}} + \lambda_{\text{gr}} \mathcal{L}_{\text{gr}} + \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}}, \tag{5}$$

where we set $\lambda_{\text{gp}} = 0.03$, $\lambda_{\text{gr}} = 0.1$ and $\lambda_{\text{cyc}} = 10$. Intuitively, the geometric editing loss $\mathcal{L}_{\text{geo}}$ jointly optimizes edited NeRF's geometry $\hat{M}_E$ and the latent shape code $\hat{\mathbf{z}}$ (for category-level objects) to best fit the user's 2D editing



(a) Geometric Prior Guidance



(b) Texture Prior Guidance

Figure 3. We leverage geometric [3, 15] and texture [7, 62] priors to guide the learning of semantic-driven NeRF editing.

while maintaining shape prior's semantic properties (*i.e.*, shape symmetry or physical conformity).

3.3. Prior-Guided Texture Editing

Semantic texture prior supervision. In our task, users only conduct editing on a single image, but we hope to naturally propagate editing effects to multi-views with semantic meaning (see Fig. 1). Therefore, we need to utilize semantic texture supervision that supports transferring the editing to the given NeRF across views, rather than using a pixel-aligned photometric loss. Inspired by Tumanyan *et al*. [62], we use a pre-trained ViT model [7] as the semantic texture prior, and apply the texture transferring loss in a multi-view manner as illustrated in Fig. 3 (b), which is defined as:

$$\mathcal{L}_{\text{tex}} = ||t_{\text{CLS}}(I_t) - t_{\text{CLS}}(\hat{I})||_2 + ||S(\hat{I}_o) - S(\hat{I})||_F, \tag{6}$$

where I_t is the user's edited image, $\hat{I}_o$ and $\hat{I}$ are the template image and edited image as introduced in Sec. 3.1. $t_{\text{CLS}}(\cdot)$ and $S(\cdot)$ are the extracted deepest CLS token and the structural self-similarity defined by Tumanyan *et al*. [62]. Essentially, this loss encourages $\hat{I}_o$ and $\hat{I}$ to share a similar spatial structure, and $\hat{I}_t$ and $\hat{I}$ to contain similar image cues.

Decoupled rendering with color compositing layer. To achieve texture modification, a naïve approach is to directly add the modification color $\mathbf{m}'$ from the editing field to the template NeRF's radiance color $\mathbf{c}'$ during volume rendering. However, we find it suffers from sub-optimal convergence when cooperating with texture transferring loss (see Sec. 4.5), since NeRF struggles to learn the global-consistent appearance under the variational supervisory as shown in Fig. 8(a). To tackle this issue, we re-design the rendering pipeline in a decoupled manner. As shown in Fig. 2, we first render the deformed template image $\hat{I}_o$ with template NeRF and the color modification $\hat{I}_m$ with $F_{\Delta T}$, and then use a 2D CNN-based color compositing layer to deferred blend the modification $\hat{I}_m$ into the template image $\hat{I}_o$, which yields final edited view $\hat{I}$. Intuitively, the coordinate-based editing field can encode fine-grained details from photometric constraints but cannot easily learn from coarse semantic supervision, while the proposed color compositing layer can reduce the difficulty by using easy-to-learn CNN layers before applying texture transferring loss. Besides, it also learns view-dependent effects from the semantic prior, making the rendering results more realistic (*e.g.*, the shining diamond effect in Fig. 1).

3.4. Editing Regularization

Feature-cluster-based semantic masking. To precisely edit the desired region while preserving other content unchanged, inspired by previous works [30, 61, 63], we learn a distilled feature field with DINO-ViT [7] to reconstruct scenes/objects with semantic features. However, existing semantic field decomposing approaches [30, 61] are limited to the query-based similarity and require all the editing to be finalized on the 3D field, which is not compatible with our color compositing mechanism. Therefore, we leverage users' editing silhouette M_e to generate several feature clusters from the distilled feature map, and compute semantic masks $\hat{M}_e$ using the closest cosine similarity to cluster centers with a threshold, which will be served for image-based editing regularization.

Regularization on geometric and texture editing. With the semantic masks that indicate the editing area, we can apply editing regularization to the geometric and texture editing, *i.e.*, by enforcing the rendered pixels and the queries at the irrelevant part unchanged, which is defined as:

$$\mathcal{L}_{\text{reg}} = \sum_{\mathbf{x} \in \hat{I} \setminus \hat{M}_e} ||F_{\Delta G}(\mathbf{x})||_1 + \sum_{\boldsymbol{r} \in \hat{I} \setminus \hat{M}_e} ||\hat{C}(\mathbf{r}) - \hat{C}_o(\mathbf{r})||_2^2, \quad (7)$$

where the sampled points $\mathbf{x}$ and rays $\boldsymbol{r}$ are both from the background area of the computed semantic masks $\hat{M}_e$.

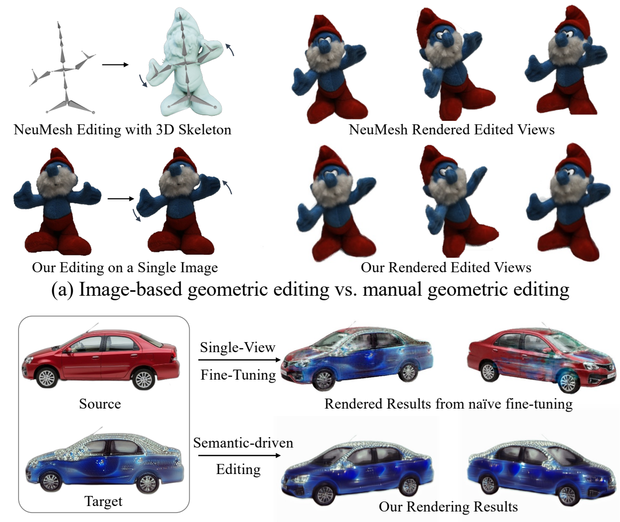
4. Experiments

4.1. Datasets

We evaluate SINE on both real-world/synthetic and object/scene datasets, including real-world car dataset [54], PhotoShape datasets (synthetic chairs) [47], "pinecone", "vasedeck", and "garden" from NeRF real-world 360° scenes [2, 42], "chairs" and "hotdog" from NeRF photo-realistic synthetic data [42], bird status from DTU [25] dataset, and "airplane" from BlenderSwap [37]. Please refer to the supplementary material for more details.

4.2. Semantic-driven vs. Manual Editing

We first clarify the difference between our semantic-driven NeRF editing and manual NeRF editing (*e.g.*, NeuMesh [69], NeRF-Editing [74]). As illustrated in Fig. 4



(a) Image-based geometric editing vs. manual geometric editing

(b) Semantic-driven texture editing vs. naïve single-view fine-tuning

Figure 4. We show the difference between our semantic-driven image-based NeRF editing and manual NeRF editing [69].

(a), our method provides much more effortless ways than manual approaches. For example, they require 3D modeling skills to bind the skeleton of the mesh using Blender [69], or drive models [74] with Mixamo poses [58], while our method can easily achieve similar geometric editing with only a single-view image. Besides, naïvely fine-tuning NeRF on a single-view with a pixel-aligned photometric loss like NeuMesh [69] would only modify visible regions, which leads to inconsistent novel view rendering (*e.g.*, in Fig. 4 (b), the car edited by single-view fine-tuning would expose unpainted red part). On the contrary, our method leverages semantic priors [7] to naturally edit objects with multi-view consistency, which does not require pixel-wise alignment and enables texture transferring between objects with different shapes (see cars and chairs in Fig. 6 (a)).

4.3. Semantic-driven Geometric Editing

We first show our geometric editing results in Fig. 5 (a), where the objects can be faithfully deformed according to users' 2D editing (*e.g.*, the airplane with warped wings [37], green chair with bent legs [42] and deformed bird status [25]). For the usage of geometry prior, we use a pre-trained DIF [15] model for cars [54], chairs [47] and planes [37], and use ARAP-based shape priors for general objects without a category-level prior (*i.e.*, toy in Fig. 4 (a), green chair with unusual shape and status in Fig. 5 (a), *etc*.). Then, we compare our method with EG3D [9], a 3D-aware generative model that learns a latent representation of category-level objects, and EditNeRF [35], a NeRF-variant that supports object editing with single-view user interaction. In addition to editing comparisons, we also used PSNR, SSIM, and LPIPS [42] to measure the edited rendering quality on synthetic cars and chairs. For the geometric editing on EG3D, we first conduct 3D GAN inversion to obtain the style code with multi-view images (same input

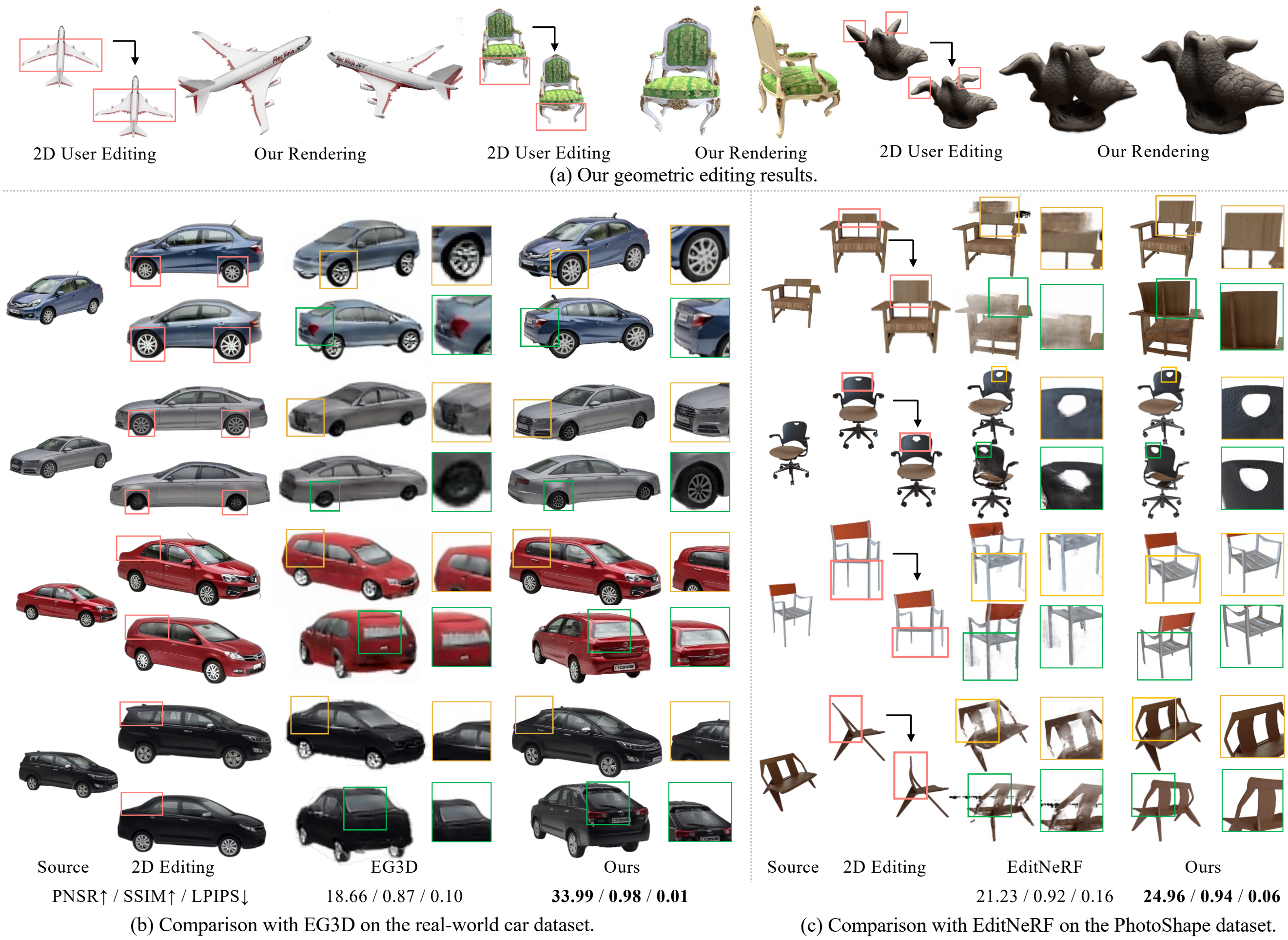


Figure 5. We compare the geometric editing with EG3D [9] and EditNeRF [35] on the real-world cars [54] and PhotoShape [47].

as ours), and then fine-tune the code on the target images. As shown in Fig. 5 (b), we conduct different editing operations on four cars from CalWare 360° datasets with DIF shape prior [15], *i.e.*, enlarging/shrinking tires/back. Due to the difficulty of learning a latent textured 3D representation and the limitation of data diversity, 3D-aware generative models like EG3D cannot produce rendering results with fine-grained details, which also results in lower evaluation metrics. For the Photoshape [47], EditNeRF [35] does not provide edited GT images, so we regenerate all testing cases using Blender, which is more challenging than the original ones. Then, we evaluate EditNeRF [35] by fine-tuning the pre-trained models on specific chairs from the PhotoShape dataset. As shown in Fig. 5 (c), EditNeRF produces more blurry rendering results than ours, and cannot achieve satisfactory results with single-view editing (*e.g.*, multi-view inconsistent chair back in the first row, and unmodified or blurry shapes in the third and fourth rows). By contrast, our method consistently delivers high-fidelity rendering results and achieves reliable editing capability by leveraging geometric priors [3, 15]. This demonstrates that, for semantic geometric NeRF editing, learning a prior-guided editing field like ours can maintain better visual quality and achieve greater generalization ability than pre-training a textured 3D generative model or latent model.

4.4. Semantic-driven Texture Editing

We evaluate our semantic texture editing ability on both objects (cars from CalWare 360°, chairs from PhotoShape [47]) and unbounded 360° scenes [2, 42]. Since our method only requires a single image as editing input, we exhibit several editing functionalities as shown in Fig. 6. Users can edit by assigning new textures on the car using Photoshop (adding sea wave windows in Fig. 6 (a)), using a downloaded Internet image with different shapes as a reference (transferring textures of cars and chairs in Fig. 6 (a)). Moreover, we cooperate SINE with off-the-shelf text-prompts editing methods [1] by using a single text-edited image as the target, which enables to change the object's appearance in the 360° scene with vivid effects (*e.g.*, shiny plastic round table or burning pinecone in Fig.6 (b)) while preserving background unchanged. It is noteworthy that our method does not pre-train a latent model within a spe-

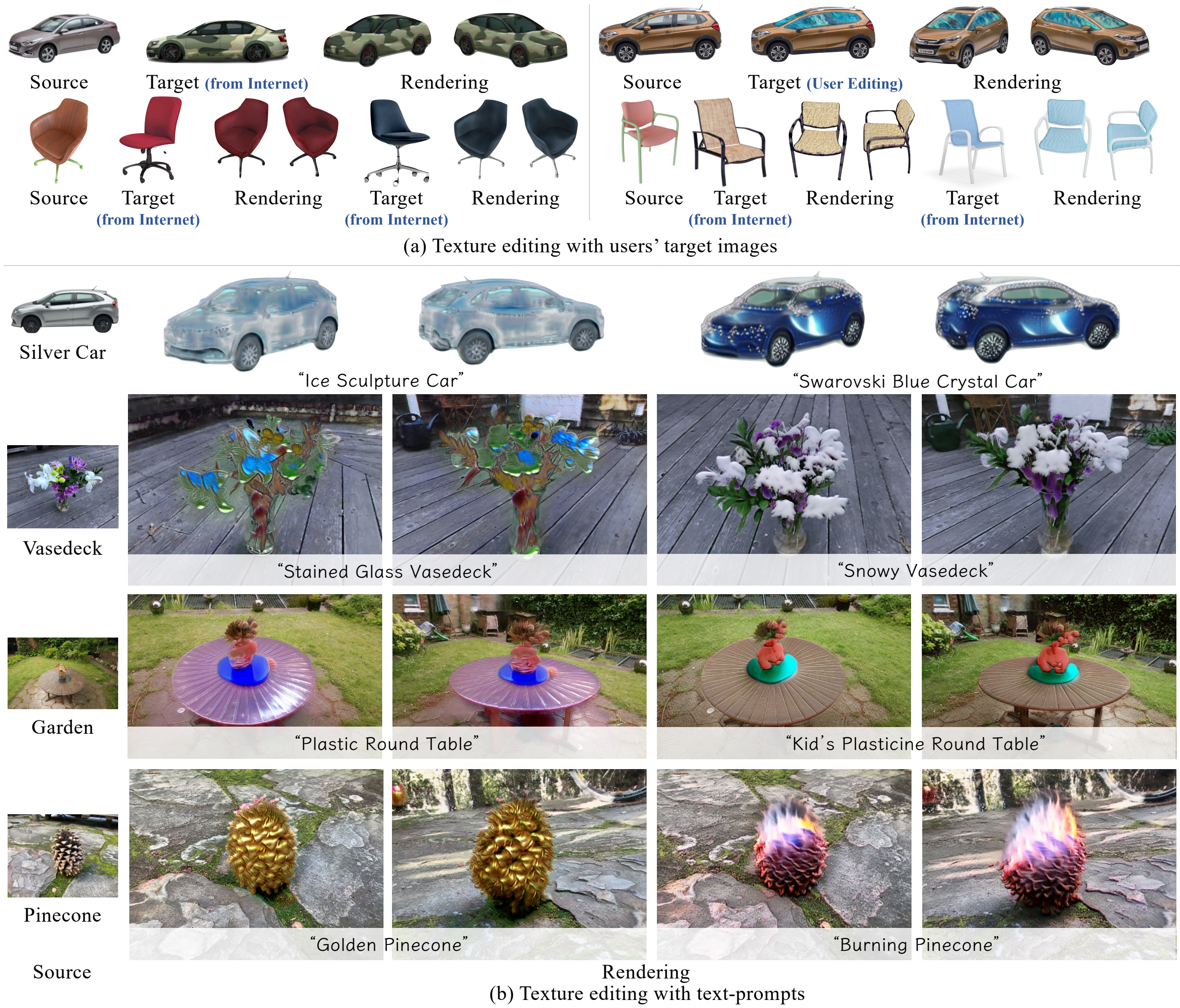


Figure 6. We show our texture editing results when given users' target images and cooperating with text-prompt-based editing methods [1].

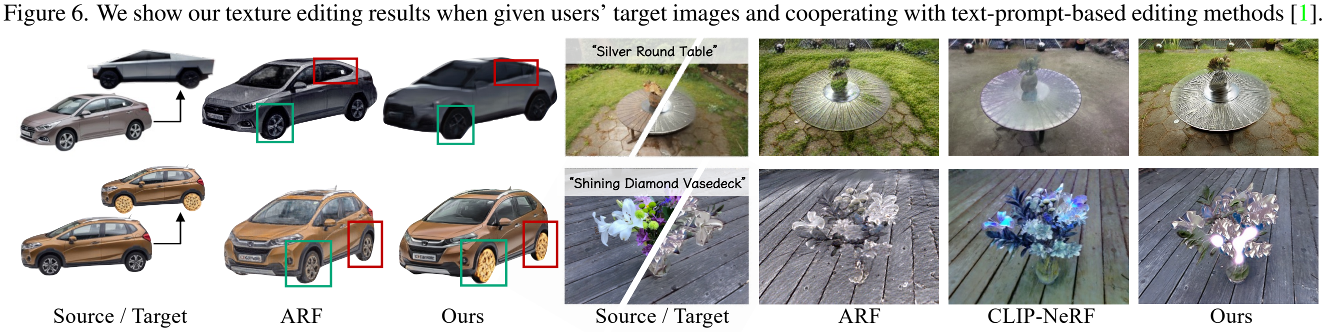
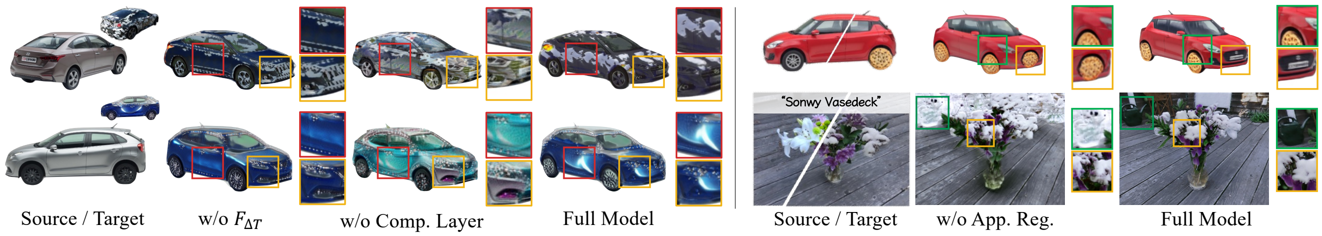


Figure 7. We compare our texture editing with ARF [75] and CLIP-NeRF [64] on the real-world cars [54] and 360° [2, 42] scene dataset.

cific category like cars or chairs, yet still transfers texture between objects with correct semantic meaning, *e.g.*, the texture styles of chair legs and cloths in the edited views are precisely matched to the target images in Fig. 6 (a). Besides, we also compare our methods with ARF [75], a NeRF stylization method that also takes a single reference image as input, and CLIP-NeRF [64], which supports text-driven NeRF editing using the large language model [52]. As demonstrated in Fig. 7, ARF globally changes appearance colors to the given target images but fails to produce



(a) Ablation studies of color compositing mechanism. (b) Ablation studies of editing regularization.

Figure 8. We analyze the effectiveness of the color compositing mechanism and editing regularization in texture editing.

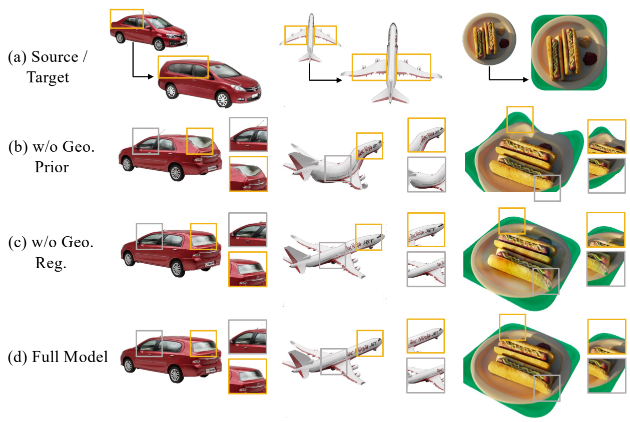


Figure 9. We inspect the efficacy of the geometric prior constraint and editing regularization in geometric editing.

fine-grained details (*e.g.*, cookie tires in Fig. 7 (a)). For CLIP-NeRF, since it directly fine-tunes NeRF's color layers, the results only show color/hue adjustment on the original scene (*e.g.*, in Fig. 7, the round table turns gray instead of a realistic silver texture, the vasedeck turns blue instead of a shining diamond). Thanks to the prior-guided editing field, our method learns more fine-grained editing details than the others, and achieves texture editing with consistent semantic meaning to the given target images (*e.g.*, similar appearance to the Tesla's cybertruck in Fig. 7 (a)), and delivers rich appearance details and vivid effects (*e.g.*, silver texture and shining diamond effects in Fig. 7 (b)).

User study. We also perform user studies to compare our texture editing (including target-image-based editing and text-prompts editing as Fig. 7) with ARF [75] and CLIP-NeRF [64] on 43 cases with 30 users. The results show that users prefer our methods (89.5% / 83.3%) to ARF (10.4% / 7.4%) and CLIP-NeRF (9.3%). Please refer to the supplementary material for more details.

4.5. Ablation Studies

Geometric prior constraints. We first analyze the effectiveness of geometric prior constraints (Sec. 3.2) by ablating geometric prior loss (Eq. (2)) in Fig. 9 (*i.e.*, deforming cars and airplanes with DIF shape prior, and adding plates with general shape prior). As shown in Fig. 9 (b), when learning without geometric prior constraints, the object will be distorted when rendered from other views (*e.g.*, collapsed car back, twisted airplanes, and warped hotdog plates). By applying geometric prior constraints, we successfully mitigate the geometric ambiguity for single-image-based editing and produce plausible rendering results from novel views.

Color compositing mechanism. We then inspect the efficacy of the color compositing mechanism (Sec. 3.3) by disabling the texture modification field $F_{\Delta T}$ and the color compositing layer in turn. As demonstrated in Fig. 8 (a), when learning texture editing without $F_{\Delta T}$, the rendered edited object can show a similar global appearance to the target, but lose vivid local patterns (*e.g.*, gray and white grains and blue shininess). When ablating the color compositing layers, the editing effect might not be properly applied to every part of the object (*e.g.*, the uncovered gray part of the car's front). When all the compositing mechanism is enabled, we successfully learn NeRF editing with fine-grained local patterns and globally similar appearance.

Editing regularization. We finally evaluate the editing regularization (Sec. 3.4) in geometric and texture editing by ablating regularization loss (Eq. (7)). As shown in Fig. 9 (c) and Fig. 8 (b), when learning editing without regularization, the irrelevant part would be inevitably changed (*e.g.*, bent car's front and airplane's head in Fig. 9 (c), a spurious cookie at the car's front and snowy background in Fig. 8 (b)). By adding editing regularization, we can modify the user-desired objects precisely while preserving other content unchanged.

Please refer to the supplementary material for more experiments (*e.g.*, ablation on more loss terms, visualization of color composition layer, discussion with external supervision, *etc.*).

5. Conclusion

We have proposed a novel semantic-driven NeRF editing approach, which supports editing a photo-realistic template NeRF with a single user-edited image, and delivers edited novel views with high-fidelity and multi-view consistency. As a limitation, our approach does not support editing with topology changes, which can be future work. Besides, our method assumes users' editing to be semantically meaningful, so we cannot use target images with meaningless random paintings.

Acknowledgment. This work was partially supported by NSF of China (No. 62102356).

References

[1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *arXiv preprint arXiv:2204.02491*, 2022. 1, 2, 6, 7

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6, 7

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 3, 4, 6

[4] Boming Zhao and Bangbang Yang, Zhenyang Li, Zuoyue Li, Guofeng Zhang, Jiashu Zhao, Dawei Yin, Zhaopeng Cui, and Hujun Bao. Factorized and controllable neural re-rendering of outdoor scene for photo extrapolation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022. 2

[5] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021. 2

[6] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3981–3990, June 2022. 1

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2, 4, 5

[8] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 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 (CVPR)*, 2021. 1, 2

[9] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5, 6

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[11] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018. 2

[12] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu. UPST-NeRF: Universal photorealistic style transfer of neural radiance fields for 3d scene. In *arXiv preprint arXiv:2208.07059*, 2022. 1, 2

[13] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1475–1484, 2022. 1, 2

[14] Frank Dellaert and Lin Yen-Chen. Neural volume rendering: Nerf and beyond. *arXiv preprint arXiv:2101.05204*, 2020. 1

[15] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021. 2, 3, 4, 5, 6

[16] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. *arXiv preprint arXiv:2204.01943*, 2022. 1, 2

[17] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 1, 2

[18] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[19] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019. 2

[20] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022. 1, 2

[21] Eldar Insafutdinov, Dylan Campbell, João F Henriques, and Andrea Vedaldi. Snes: Learning probably symmetric neural surfaces from incomplete data. *arXiv preprint arXiv:2206.06340*, 2022. 2

[22] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2

[23] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[24] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021. 2

[25] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large Scale Multi-view Stereopsis Evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 5

[26] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021. 4

[27] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022. 1, 2

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1, 2

[29] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2

[30] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 1, 2, 5

[31] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2

[32] Xingyi Li, Chaoyi Hong, Yiran Wang, Zhiguo Cao, Ke Xian, and Guosheng Lin. Symmnerf: Learning to explore symmetry prior for single-view view synthesis. *arXiv preprint arXiv:2209.14819*, 2022. 2

[33] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver 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*, pages 6498–6508, 2021. 4

[34] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34:16331–16345, 2021. 1, 2

[35] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5773–5783, 2021. 1, 2, 5, 6

[36] William E Lorensen and Harvey E Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 3, 4

[37] John Roper Matthew Muldoon. Blenderswap. https://www.blenderswap.com/, 2022. Accessed: 2022-11-10. 5

[38] Nelson L. Max. Optical Models for Direct Volume Rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995. 3

[39] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1, 2

[40] Lu Mi, Abhijit Kundu, David Ross, Frank Dellaert, Noah Snavely, and Alireza Fathi. im2nerf: Image to neural radiance field in the wild. *arXiv preprint arXiv:2209.04061*, 2022. 2

[41] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10461–10471, 2021. 4

[42] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 5, 6, 7

[43] Ashkan Mirzaei, Yash Kant, Jonathan Kelly, and Igor Gilitschenski. Laterf: Label and text driven object radiance fields. *arXiv preprint arXiv:2207.01583*, 2022. 2

[44] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[45] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 2

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 3

[47] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *ACM Trans. Graph.*, 37(6), Nov. 2018. 5, 6

[48] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2, 4

[49] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 4

[50] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 1

[51] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

ing transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 7

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2

[54] Arun Kumar Sahlam. Carwale. https://www.carwale.com/, 2022. Accessed: 2022-11-10. 5, 6, 7

[55] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2017. 2

[56] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1, 2

[57] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 4

[58] Nazim Kareemi Stefano Corazza. Mixamo. https://www.mixamo.com/, 2022. Accessed: 2022-11-10. 5

[59] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *arXiv preprint arXiv:2205.15517*, 2022. 1, 2

[60] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 1, 2

[61] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022. 2, 5

[62] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10748–10757, 2022. 2, 4

[63] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*, 2021. 2, 5

[64] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 1, 2, 7, 8

[65] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2, 4

[66] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1, 2

[67] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. *arXiv preprint arXiv:2207.09686*, 2022. 2

[68] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. *arXiv preprint arXiv:2204.00928*, 2022. 2

[69] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision*, pages 597–614. Springer, 2022. 1, 2, 5

[70] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural rendering in a room: amodal 3d understanding and free-viewpoint rendering for the closed scene composed of pre-captured objects. *ACM Transactions on Graphics (TOG)*, 41(4):1–10, 2022. 2

[71] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 1, 2

[72] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2

[73] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021. 2

[74] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 1, 2, 5

[75] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 1, 2, 7, 8

[76] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2

[77] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 2