https://doi.org/10.1007/s11704-025-41176-9

REVIEW ARTICLE

A survey on 3D editing based on NeRF and 3DGS

Chen-Yang ZHU*, Xin-Yao LIU*, Kai XU, Ren-Jiao YI✉

College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China

Received November 2, 2024; accepted March 14, 2025

E-mail: yirenjiao@nudt.edu.cn. * These authors contributed equally to this work.

Abstract

In recent years, 3D editing has become a significant research topic, primarily due to its ability to manipulate 3D assets in ways that fulfill the growing demand for personalized customization. The advent of radiance field-based methods, exemplified by pioneering frameworks such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS), represents a pivotal innovation in scene representation and novel view synthesis, greatly enhancing the effectiveness and efficiency of 3D editing. This survey provides a comprehensive overview of the current advancements in 3D editing based on NeRF and 3DGS, systematically categorizing existing methods according to specific editing tasks while analyzing the current challenges and potential research directions. Our goal through this survey is to offer a comprehensive and valuable resource for researchers in the field, encouraging innovative ideas that may drive further progress in 3D editing.

Keywords

3D editing; NeRF; 3DGS

1 Introduction

3D editing has become an advanced research topic in the field of computer graphics, striving to achieve efficient and precise manipulation of 3D content. The ability to seamlessly modify 3D assets, whether by altering appearances, reshaping geometry, or transforming objects, has revolutionized creative workflows, providing greater flexibility and enhanced realism. It has a wide range of applications, including film production, virtual reality (VR), augmented reality (AR), and digital content creation.

Traditional 3D modeling and editing methods [1–3] offered precise control over the shape and surface of 3D content through explicit representation, laying the foundation of 3D editing. However, these methods faced notable challenges, including computational efficiency, detail preservation, and high storage demands. In addition, 3D editing software [4,5] requires users to make labor-intensive manual adjustments to 3D content. Furthermore, the editing operations are predefined, which may not fully meet the specific needs of users. In the pursuit of improving editing efficiency and quality, researchers are exploring innovative techniques to minimize manual effort and streamline workflows, making the editing process faster and more intuitive.

Recently, the emergence of radiance field-based methods, highlighted by Neural Radiance Fields (NeRF) [6] and 3D Gaussian Splatting (3DGS) [7] has brought about groundbreaking advancements in 3D reconstruction and editing. NeRF [6] utilizes the powerful capabilities of deep neural networks to model complex scenes. It enables high-quality 3D reconstruction and novel view synthesis from a collection of posed images. Conversely, 3DGS [7] diverges from the dependence on neural networks, which utilizes a collection of Gaussians to model the geometry and appearance of 3D scenes. It also employs an efficient rendering method by rasterizing Gaussians into images, making it particularly advantageous for real-time rendering applications. By combining efficient 3D representations with high-quality rendering, NeRF [6] and 3DGS [7] bridge the gap between computational efficiency and visual fidelity. They not only advance 3D reconstruction but also provide a robust foundation for the development of more scalable, flexible, and precise 3D editing techniques.

This survey aims to provide a comprehensive and scholarly examination of the current landscape of 3D scene editing, with a focus on methods based on NeRF [6] and 3DGS [7]. It encompasses a wide range of 3D editing tasks, including appearance editing, object transformation, shape deformation, scene inpainting, and creative editing, as shown in Fig. 1. We also provide a structured map of the key methods involved in these tasks, as shown in Fig. 2.

The remainder of this paper is organized as follows. In Section 2, we summarize the relevant background technologies and popular datasets for 3D editing. In Section 3, we introduce various editing methods categorized by their specific tasks, highlighting their unique features, advantages, and potential limitations. We assess the persistent challenges in Section 4, thereby informing future research directions in 3D editing. A conclusion is drawn in Section 5.

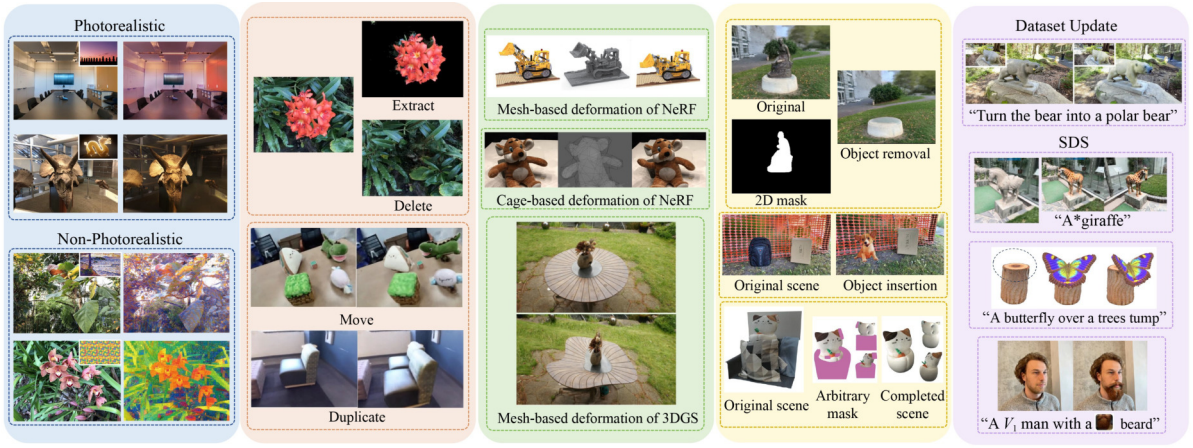


Fig. 1 Overview of the survey, including the key components: (a) arance editing, (b) object transformation, (c) shape deformation, (d) scene inpainting, and (e) creative editing. Each dashed box contains result images from a distinct work. Within each category arranged from left to right, the respective methods, organized from top to bottom, are: FPRF [8], StylizedGS [9], DFF [10], ObjectNeRF [11], NeRFEditing [12], Deforming-NeRF [13], GaMeS [14], SPIn-NeRF [15], InFusion [16], Nerfiller [17], IN2N [18], DreamEditor [19], Focaldreamer [20], and TIP-Editor [21]

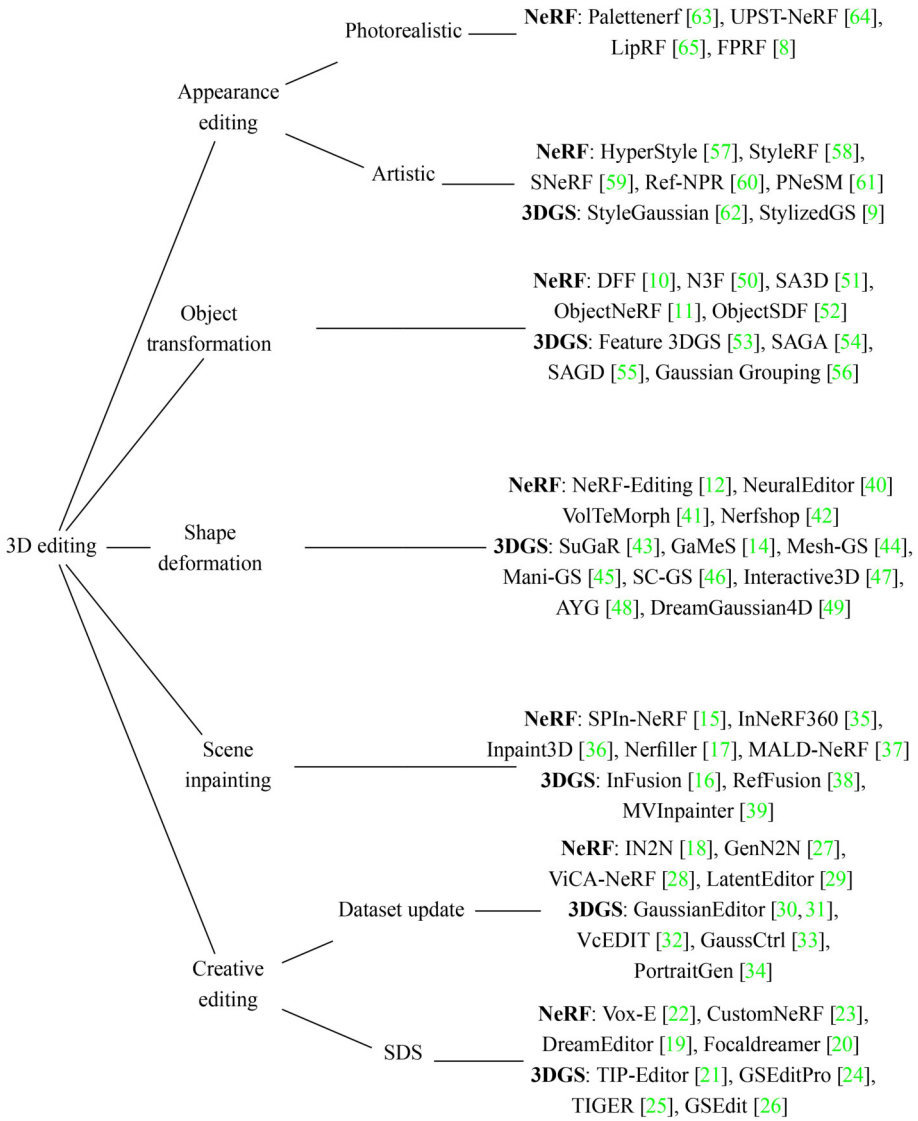


Fig. 2 A structured map of the key methods discussed in the survey, organized by their respective tasks

2 Background

2.1 Radiance fields

Radiance fields have recently emerged as groundbreaking methods for 3D reconstruction and novel view synthesis, with NeRF [6] and 3DGS [7] standing out as key methods in this domain. Below are detailed descriptions of these two methods.

2.1.1 Neural radiance fields

Neural Radiance Fields (NeRF) [6] leverages the expressive power of neural networks to model the geometry and appearance of 3D scenes with remarkable precision.

Through supervised training on a collection of 2D images with their corresponding camera poses, NeRF effectively maps 3D spatial coordinates $\mathbf{x}$ and viewing directions $\mathbf{d}$ to density σ and color $\mathbf{c}$ with a continuous scene function:

$$F_\theta(\mathbf{x},\mathbf{d}) = (\mathbf{c},\sigma).$$

Subsequently, NeRF integrates these predicted values along rays cast from the camera to generate high-quality images through volumetric rendering, defined as:

$$\mathbf{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})\,\mathrm{d}t,$$

where $T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))\,\mathrm{d}s\right)$ is the accumulated transmittance along the ray $\mathbf{r}(t)$ from t_n to t. An overview of NeRF is shown in Fig. 3.

By leveraging the strengths of neural networks to model complex 3D scenes, NeRF has revitalized the integration of the deep learning algorithm with the traditional computer graphics technique. Its subsequent research [66–69] has also made significant progress in various aspects.

2.1.2 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [7] has garnered considerable attention due to high-quality reconstruction and efficient rendering capabilities. Unlike fully implicit methods, 3DGS combines the strengths of neural network-based optimization with the explicit and structured representation of 3D data. It employs a collection of 3D Gaussians to model the geometry and appearance of 3D scenes, effectively capturing the fine details of scene structure and enabling efficient storage and fast rendering.

Each Gaussian has a series of properties, including 3D position x, opacity α, 3D covariance matrix Σ, and color c. The 3D Gaussian is defined as:

$$G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}.$$

3D Gaussians can be rasterized efficiently by projecting them to 2D planes and applying view rendering, the formulation for projecting 3D Gaussians is as follows:

$$\Sigma' = JW\Sigma W^{\mathrm{T}}J^{\mathrm{T}},$$

where $\Sigma = RSS^{\mathrm{T}}R^{\mathrm{T}}$, R is a rotation matrix, S is a scaling matrix, W is the viewing transformation and J is the Jacobian matrix for the projective transformation. The differentiable splatting rendering is as follows:

$$C = \sum_{i\in\mathcal{N}} c_i\alpha_i\prod_{j=1}^{i-1}(1-\alpha_j),$$

where c_i represents the computed color from spherical harmonics (SH) color coefficients and α_i is given by evaluating a 2D Gaussian with covariance Σ' multiplied by the opacity. An overview of 3DGS is illustrated in Fig. 4.

2.2 2D editing

2.2.1 Image style transfer

Image style transfer involves transforming the stylistic attributes of one image to match those of a reference image, which encompasses

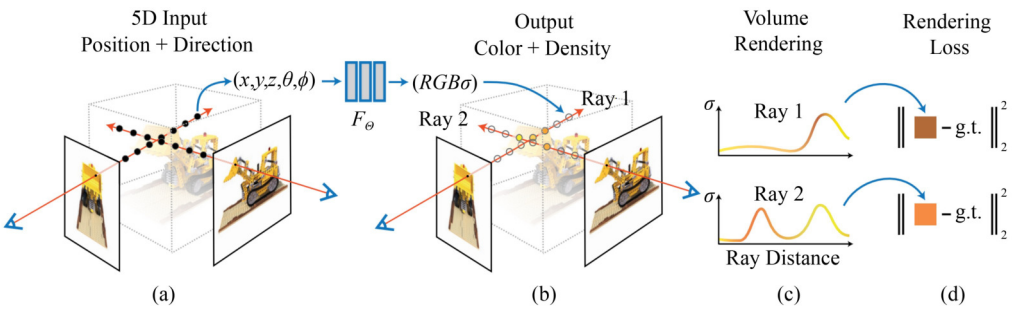


Fig. 3 An overview of neural radiance field scene representation and differentiable rendering procedure. (a) 5D input; (b) Output; (c) Volume rendering; (d) Rendering loss. Image sourced from [6]

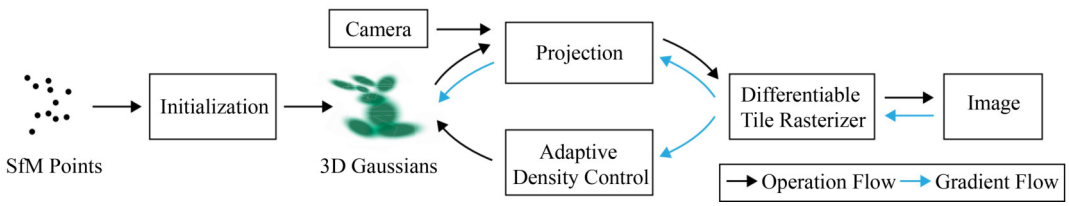


Fig. 4 An overview of 3D Gaussian Splatting. Image sourced from [7]

artistic style transfer and photorealistic style transfer.

Artistic style transfer focuses on producing results that preserve the content structure of the original image while adopting the visual characteristics of an artistic style. This field was pioneered by Gatys et al. [70] with the introduction of Neural Style Transfer (NST), which leverages CNN to apply different styles to a content image. Specifically, it employs VGG-19 [71] as a feature extractor. While this method achieves impressive results, it is computationally expensive and slow. The following methods [72–74] utilize feed-forward networks to speed up the stylization process.

To achieve arbitrary style transfer, Adaptive Instance Normalization (AdaIN) [75] alignes the mean and variance of feature maps extracted from content images with those of style images:

$$\mathrm{AdaIN}(x,y)=\sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right)+\mu(y),$$

where x is a content input and y is a style input. An overview of AdaIN is shown in Fig. 5. Linear Style Transfer (LST) [76] further improves the efficiency by learning linear transformations directly to the feature maps, realizing fast and scalable style transfer across multiple styles.

In contrast, photorealistic style transfer aims to apply stylistic changes while preserving the realism and structure of the content image. Luan et al. [77] proposed a neural style transfer method through semantic segmentation guidance and local affine transforms, effectively maintaining scene structure while transferring artistic styles. Mechrez et al. [78] introduced a screened poisson equation-based framework, effectively preserving structural fidelity while achieving stylization. PhotoWCT [79] employs whitening and coloring transforms (WCT) to align the feature statistics of the content and style images, aiming to generate consistent and visually coherent results. WCT^2 [80] integrates wavelet transforms into neural networks, enabling the preservation of image structure with greater fidelity.

2.2.2 Generative editing

In the academic domain of 2D generative editing, Generative Adversarial Networks (GANs) [81] and diffusion models [82] have emerged as two pivotal techniques due to their remarkable performance in generating and modifying high-quality images.

GANs [81] consists of two neural networks, a generator that produces increasingly realistic images and a discriminator that distinguishes real images from generated ones. These two networks are trained in a competitive framework. GANs have demonstrated effectiveness for a variety of tasks, including super-resolution, style transfer, and image-to-image translation.

Diffusion models [82] operate on the principle of a forward and reverse diffusion process. In the forward process, noise was gradually added to the data over a series of steps, transforming it into a noise distribution. In the reverse process, the model learns to denoise the data step by step, progressively reconstructing the original data distribution from the noise. The directed graphical model of the whole process is shown in Fig. 6. The following diffusion models have garnered significant attention.

Latent Diffusion Models (LDMs) [83] perform diffusion in a latent space, significantly improving computational efficiency and output quality. Stable Diffusion [83] further leverages pre-trained conditional models for stable and efficient image synthesis based on text prompts or other conditional inputs. Additionally, ControlNet [84] introduces additional conditions to guide the generative process, such as canny edge, human pose, and depth, allowing for fine-grained control over image modifications. Low-Rank Adaptation (LoRA) [85] fine-tunes the diffusion model with minimal computational overhead, making it adaptable to new tasks by modifying a few parameters within the model. Furthermore, Instruct-Pix2Pix [86] realizes flexible and intuitive image editing, which is trained based on an image-prompt-image dataset. In this dataset, groundtruth prompts are generated by a large language model (GPT-3 [87]), while the corresponding paired images are produced using Stable Diffusion [83] with Prompt-to-Prompt [88] techniques.

2.3 3D generation

Recent advancements in text-to-image diffusion models have

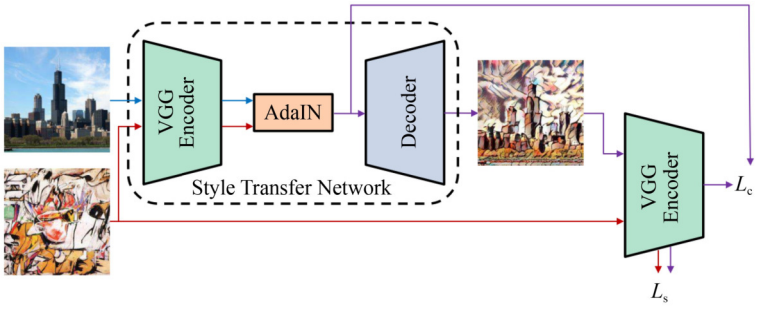


Fig. 5 An overview of AdaIN. Image sourced from [75]

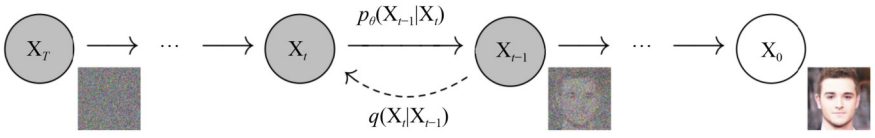


Fig. 6 An overview of denoising diffusion probabilistic models. Image sourced from [82]

introduced novel avenues for 3D content generation. However, applying these successes to 3D synthesis faces unique challenges, primarily due to the lack of extensive labeled 3D datasets and efficient architectures for denoising 3D data.

DreamFusion [89] represents a groundbreaking approach that addressed these limitations by introducing Score Distillation Sampling (SDS) to facilitate text-to-3D synthesis. SDS effectively distills priors from 2D diffusion models for 3D generation, as shown in Fig. 7. The SDS loss calculates the per-pixel gradient as follows:

$$\nabla_\theta \mathcal{L}_{SDS}(\phi, \mathbf{x}) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)\frac{\partial \mathbf{x}}{\partial \theta}\right],$$

where θ is the learnable parameters of the 3D scene, $\epsilon \sim \mathcal{N}(0, I)$ is the Gaussian noise, $\hat{\epsilon}_\phi(\mathbf{z}_t; y, t)$ is the predicted noise by the pretrained 2D diffusion model, $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ is the rendered image with added noise at timestep t, y is the text instruction, and $w(t)$ is a weighting function. A variety of innovative methods, including Variational Score Distillation (VSD) [90], Delta Denoising Score (DDS) [91], and Posterior Distillation Sampling (PDS) [92], has been developed to tackle significant challenges faced by SDS, such as oversaturation, oversmoothing, and insufficient detail.

2.4 Datasets

We have systematically compiled a comprehensive collection of publicly available datasets, which are widely utilized in the field of 3D editing. Table 1 provides an insightful overview of these datasets, including the year of publication, the number of scenes they contain, and a brief description of their unique characteristics.

These datasets serve as essential resources for benchmarking and advancing 3D editing techniques, providing a diverse range of scenarios that span from synthetic to real-world. They enable researchers to evaluate the performance of various 3D editing methods in different tasks.

3 Method

We conduct a systematic overview of key methods in 3D editing, organizing them into five editing tasks, i.e., appearance editing, object transformation, shape deformation, scene inpainting, and creative editing. Each method addresses unique challenges and provides innovative solutions for manipulating the appearance, geometry, and structure of 3D content, collectively driving advancements in the field.

3.1 Appearance editing

The appearance editing of radiance fields involves the manipulation and refinement of the visual attributes within 3D scenes, allowing for precise adjustments to the appearance properties, including color, texture, and lighting. This task can be broadly divided into two

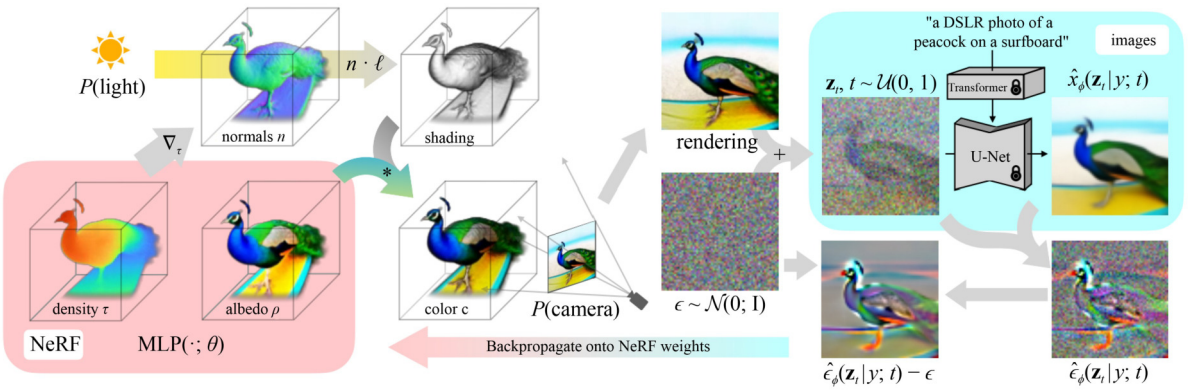


Fig. 7 An overview of DreamFusion. Image sourced from [89]

Table 1 Popular datasets for 3D editing

Dataset	Year	Number of scenes	Highlights
NeRF-LLFF [6]	ECCV 2020	8	A real-world forward-facing scene dataset.
NeRF-Synthetic [6]	ECCV 2020	8	A synthetic dataset generated by Blender.
NeRF-Real-360 [6]	ECCV 2020	2	A real-world 360° scene dataset.
IBRNet [93]	CVPR 2021	about 120	A collection dataset consisting of synthetic and real-world data.
Tanks and Temples [94]	TOG 2017	14	A real-world large-scale scene dataset.
DTU [95]	CVPR 2014	80	A multi-view stereo dataset with precise camera poses.
Mip-NeRF 360 [96]	CVPR 2022	9	A real-world indoor and outdoor scene dataset with complex central objects.
IN2N [18]	ICCV 2023	6	A dataset consisting of 360° scenes, faces, and full-body portraits.
SPIn-NeRF [15]	CVPR 2023	10	A real-world forward-facing scene dataset with annotated object masks.

categories: photorealistic appearance editing and artistic appearance editing. Photorealistic appearance editing focuses on maintaining a high degree of realism and authenticity in the edited scene. On the other hand, artistic appearance editing is not constrained by real-world physical properties and allows for a wide range of artistic and stylized expressions. Table 2 provides a comprehensive overview of a range of appearance editing methods, including their year of publication, the specific tasks they address (photorealistic or artistic), and their base techniques.

3.1.1 Photorealistic appearance editing

Photorealistic appearance editing in the context of 3D radiance fields is an advanced technique aiming at achieving visual alterations consistent with real-world physical phenomena, which involves altering the color or applying the color style from a reference image onto the original 3D scene. The edited NeRF and 3DGS are rendered with a level of detail and realism that is virtually indistinguishable from actual observations.

EditNeRF [109] is a pioneering method for editing the implicit radiance field, introducing a conditional neural field with latent vectors for shape and appearance to facilitate user-directed local edits. It enables color adjustments or selective removal of shape parts within specific object categories. A series of techniques [63,110,111] employ palette-based methods for color editing. They decompose the scene appearance into palette-based colors with weights and achieved view-consistent editing results.

The following methods are dedicated to realizing 3D photorealistic style transfer (PST) with reference style images to produce visually consistent and stylized scenes. UPST-NeRF [64] leverages a hypernetwork to control the features of style images as latent codes for scene stylization. HyperNet takes the features extracted by StyleEncoder with VGG as input to update the weights of HyperLiner, followed by modifying RGBNet information to update the scene style. It also trains a 2D PST network for more realistic scene presentation. LipRF [65] utilizes a Lipschitz network [98] to convert the pre-trained appearance representation into a stylized 3D scene, guided by style emulation through 2D PST methods on each view. Under the Lipschitz condition, adaptive regularization and gradual gradient aggregation are applied to balance reconstruction fidelity, stylization quality, and computational efficiency. Compared to the previous methods, FPRF [8] achieves the photorealistic style transformation with multiple style references in less training time. By utilizing AdaIN [75], FPRF stylizes 3D scenes without optimization, enabling rapid style manipulation. It employs a radiance field to capture scene geometry and content. It also introduces a style dictionary module with style attention to handle diverse content by retrieving matching styles from multiple reference images. It outperforms previous 3D photorealistic appearance editing methods as shown in Fig. 8.

3.1.2 Artistic appearance editing

Artistic appearance editing involves stylization and artistic transformation to achieve expressive and stylized renderings that deviate from photorealism, which combines the original scene structure with the distinctive style patterns of the reference image, ensuring a seamless and contextually appropriate transformation.

HyperStyle [57] pioneers the field of 3D scene stylization based on NeRF. It extends the NeRF model with a geometry branch and an appearance branch, alongside a hypernetwork. It enables arbitrary style transfer by training the hypernetwork to predict the parameters of the appearance branch according to the latent vector extracted from the reference image while fixing the geometry branch.

Table 2 Summary of selected appearance editing methods

Method	Year	Task	Baselines
UPST-NeRF [64]	TVCG 2024	Photorealistic	DVGO [97], AdaIN [75]
LipRF [65]	CVPR 2023	Photorealistic	Plenoxels [67], Lipschitz networks [98]
FPRF [8]	AAAI 2024	Photorealistic	K-Planes [99], AdaIN [75]
HyperStyle [57]	WACV 2022	Artistic	NeRF++ [100], Hypernetworks [101]
StylizedNeRF [102]	CVPR 2022	Artistic	NeRF [6], AdaIN [75]
ARF [103]	ECCV 2022	Artistic	Plenoxels [67]
INS [104]	ECCV 2022	Artistic	SIREN [105]
Ref-NPR [58]	CVPR 2023	Artistic	Plenoxels [67]
SNeRF [59]	SIGGRAPH 2022	Artistic	NeRF++ [100]
StyleRF [60]	CVPR 2023	Artistic	Tensorf [106]
LAENeRF [107]	CVPR 2024	Artistic	Instant-NGP [108], NeRFShop [42]
PNeSM [61]	AAAI 2024	Artistic	DVGO [97]
StyleGaussian [62]	SIGGRAPH Asia 2024	Artistic	3DGS [7]
StylizedGS [9]	Arxiv 2024	Artistic	3DGS [7]

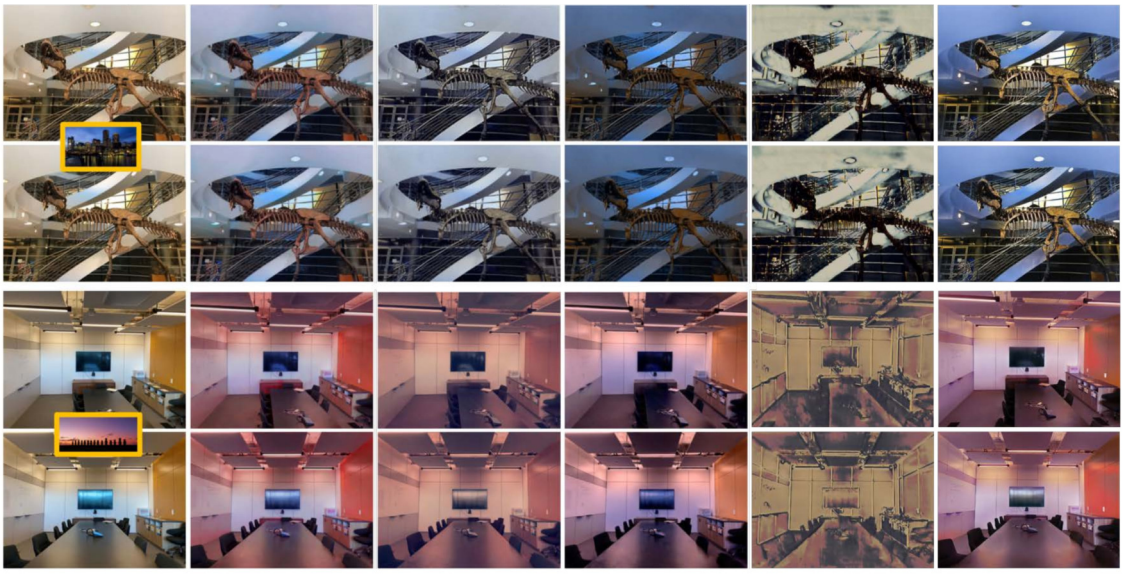


Fig. 8 Compared with other 3D PST methods, FPRF achieves photorealistic stylization in a natural and vivid manner. Results sourced from FPRF [8]. (a) Scene & Style; (b) UPST; (c) CCPL + LipRF; (d) WCT^2+LipRF; (e) StyleRF; (f) FPRF

Additionally, it introduces a patch sub-sampling algorithm to train the hypernetwork with the content and stylization losses. Its overview is illustrated in Fig. 9.

StylizedNeRF [102] utilizes a mutual learning strategy to optimize both the stylized NeRF and the 2D stylization network while handling the ambiguities of the 2D stylized results via learnable latent codes, ensuring consistency in geometry. ARF [103] introduces a nearest neighbor feature matching loss to impart detailed style structures onto the original NeRF. It also employs a deferred back-propagation technique to optimize with full-resolution images, overcoming GPU memory constraints. INS [104] divides the representation into two modules: a style implicit module and a content implicit module, which are fused through an amalgamation module. It also introduces a self-distilled geometry consistency loss to regularize scene geometry, ensuring view-consistent stylized scenes.

StyleRF [60] is an innovative zero-shot 3D style transfer framework that performs style transformations in the feature space of a radiance field. It introduces sampling-invariant content transformation and defers style transformation to achieve multi-view consistency and improve stylization efficiency. SNeRF [59] proposes an alternating training approach, which renders images from different views, stylizes them with an image stylization module, and then train the NeRF iteratively to match multi-view stylized images. The alternating training method allows all hardware capacity to be allocated separately to either image stylization or NeRF training to address memory limitations.

Ref-NPR [58] enhances scene stylization by preserving geometric and semantic consistency with a stylized reference view. It achieves this by projecting 2D style references into 3D space through a

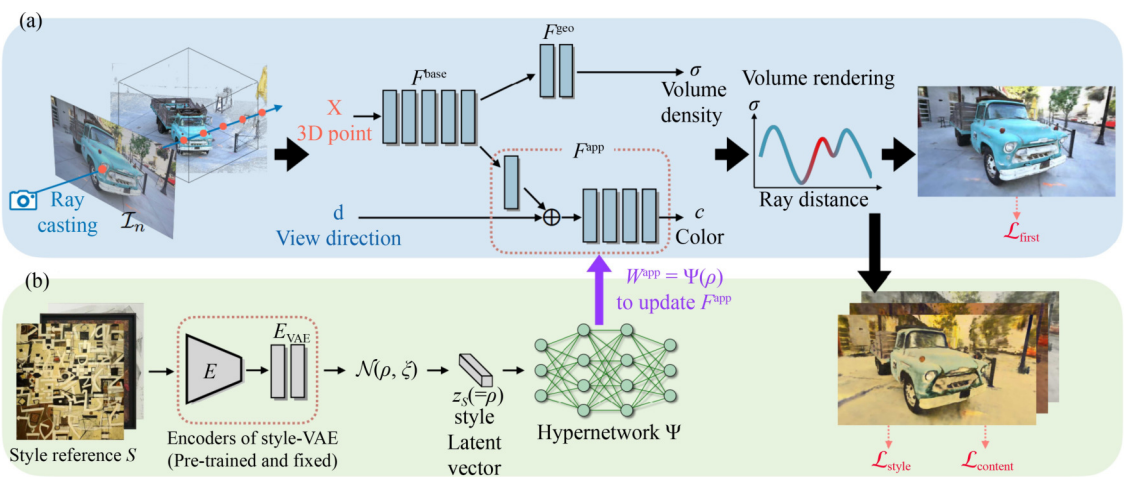


Fig. 9 An overview of HyperStyle, a 3D scene style transfer approach. Image sourced from [57]. (a) Geometric Training Stage; (b) Stylization Training Stage

reference-based ray registration process and applying template-based feature matching to transfer style to occluded regions, ensuring visually consistent and high-quality stylizations. The result of its stylized scenes is shown in Fig. 10.

LAENeRF [107] aims to realize local stylization, which utilizes a 3D grid for content selection and employs region growing for smooth transitions between neighboring regions. It introduces a novel NeRF-like module that learns a palette-based color decomposition within selected regions. By incorporating style loss, the selected regions could be stylized while maintaining low processing time. PNeSM [61] proposes the first framework for arbitrary 3D scene stylization with a single model. It utilizes a voxel-grid UV mapping network for 3D-to-2D projection and an appearance mapping for reconstructing radiance colors. By decoupling style and geometry, the framework achieves unified and adaptable stylization. It also employs a pre-trained 2D stylization network for stylization mapping, which is enhanced with visual prompts, allowing 2D style patterns to be effectively adapted to unique 3D scenes. The stylized results achieved by PNeSM demonstrates superior quality compared with the baselines, as shown in Fig. 11 and Table 3.

Other works [9,62,112,113] focus on stylizing 3DGS models with reference style images. StyleGaussian [62] achieves real-time 3D style transfer with strict multi-view consistency. It embeds VGG features into 3D Gaussians and employs an efficient strategy to render these features by initially rendering lower-dimensional ones and then mapping them to high-dimensional ones. It aligns the stylized features with a style image, akin to AdaIN [75] and a KNN-based 3D CNN decodes the features back to RGB. StylizedGS [9] is a controllable 3D stylization method, which performs color matching with the style image and utilizes a 3DGS filter to remove artifacts. It employs a nearest neighbor feature match loss and a depth preservation loss for optimization.

Existing methods for photorealistic and artistic appearance editing based on NeRF and 3DGS have made significant strides in enhancing the realism and artistic flexibility of 3D content. However, they still face critical challenges, including computational inefficiency, limitations in visual fidelity, and difficulties in handling large-scale scenes. While NeRF excels in generating high-quality renderings, it is computationally expensive and slow. Conversely, 3DGS improves efficiency but struggles to accurately model intricate geometric

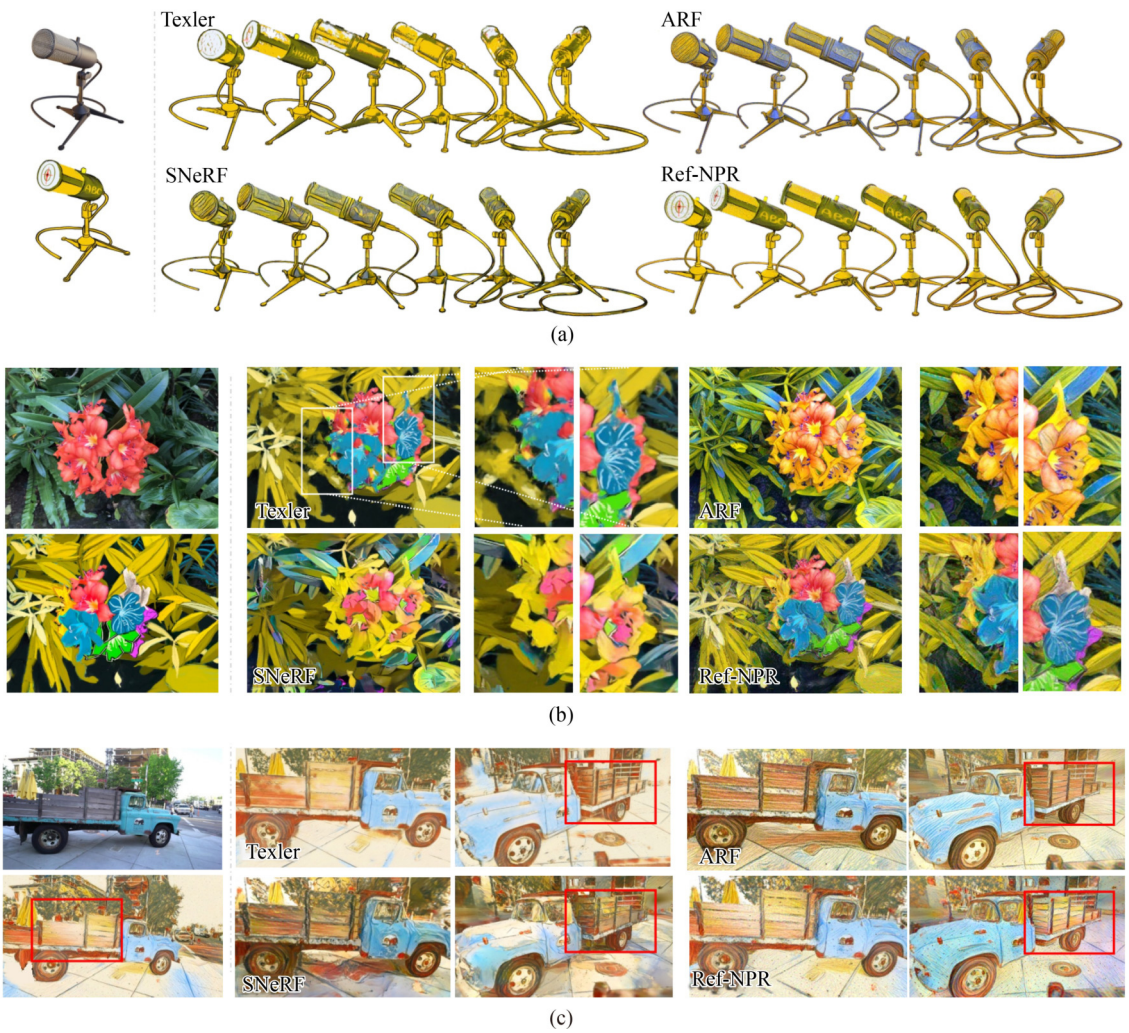


Fig. 10 Compared with other video and scene stylization methods, Ref-NPR excells in maintaining both geometric and semantic-style consistency in stylized scenes. Results sourced from Ref-NPR [58]

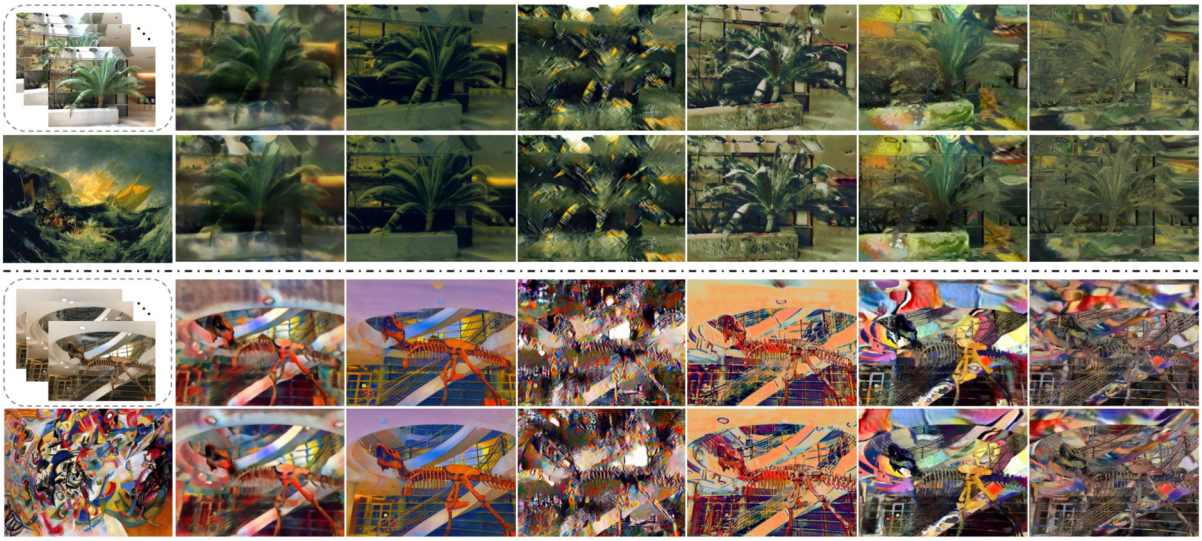


Fig. 11 Compared with previous methods, PNeSM produces stylized scenes with clear geometry and high stylization quality. Results sourced from PNeSM [61]. (a) Original scene and style; (b) StylizedNeRF; (c) ARF; (d) INS (e) StyleRF; (f) PNeSM-SR; (g) PNeSM-SA

Table 3 Short-range and long-range consistency comparison, in terms of LPIPS. Results sourced from [61]

Method	Short-range ↓	Long-range ↓
StylizedNeRF [102]	0.0229	0.0627
ARF [103]	0.0125	0.0353
INS [104]	0.0208	0.0439
StyleRF [60]	0.0235	0.0531
PNeSM [61]	**0.0116**	**0.0351**

details. The quality of the reconstruction impacts the final stylized scenes. Future research could address these limitations by improving 3DGS through the integration of geometric constraints, optimizing methods to achieve high-quality editing results, and leveraging hardware acceleration for real-time processing.

3.2 Object transformation

Object transformation involves manipulating the spatial property of the object, encompassing operations such as object deletion, extraction, translation, rotation, and replication. These operations enable flexible and precise control over the spatial arrangement of objects, facilitating tasks such as scene composition, object reconfiguration, and interactive editing. 3D segmentation is integral to this process, which identifies and delineates objects within a 3D space. It provides accurate boundaries and spatial relationships, ensuring that transformations are applied with precision.

Semantic-NeRF [114] is the first work to integrate semantic information into NeRF. It utilizes RGB images alongside their corresponding semantic labels to train the NeRF network, incorporating an additional semantic output to enhance the semantic understanding of scenes.

To further realize object-based modeling, ObjectNeRF [11] introduces a scene branch and object branch to encode the geometry and appearance of the scene and objects separately, which enables the shifting or addition of objects within the scene. In order to enhance geometry learning and accurately represent individual objects, ObjectSDF [52] utilizes the Signed Distance Function (SDF) to model the scene and objects, which includes an object-SDF for modeling all instances and a scene-SDF for compositing decomposed objects in the scene. It also utilizes semantic labels for supervising individual object SDFs, enabling precise object extraction.

To avoid the use of extensive annotation and instance-specific training, DFF [10] adopts pre-trained feature extraction networks, such as LSeg [115] and DINO [116], as teacher networks for knowledge distillation, resulting in a scene-specific 3D feature field. DFF enables extraction and deletion of target objects based on image patches or text. Similarly, N3F [50], LERF [117], and ISRF [118] also learns additional feature fields to realize self-supervised 3D segmentation.

With the emergence of the Segment Anything Model (SAM) [119], it becomes possible to rapidly and efficiently generate object labels from text or point prompts, further advancing the field of 3D segmentation. Building upon this, SA3D [51] begins by generating an initial 2D mask with prompts in a single view using the SAM [119]. Then, it alternates between mask inverse rendering and cross-view self-prompting to iteratively refine the 3D mask until a complete object segmentation is achieved.

A recent series of works is based on 3DGS. Feature 3DGS [53] is the first method for feature field distillation based on 3DGS. It learns semantic features for each 3D Gaussian and leverages guidance from 2D foundation models [119,120]. It also employs a low-dimensional feature field that is upsampled using a convolutional decoder to improve rendering efficiency. In addition, SAGA [54] and SAGD [55] also leverage SAM [119] for efficient segmentation within 3DGS. Gaussian Grouping [56] enhances 3DGS by incorporating identity encodings for 3D objects, enabling the system to group

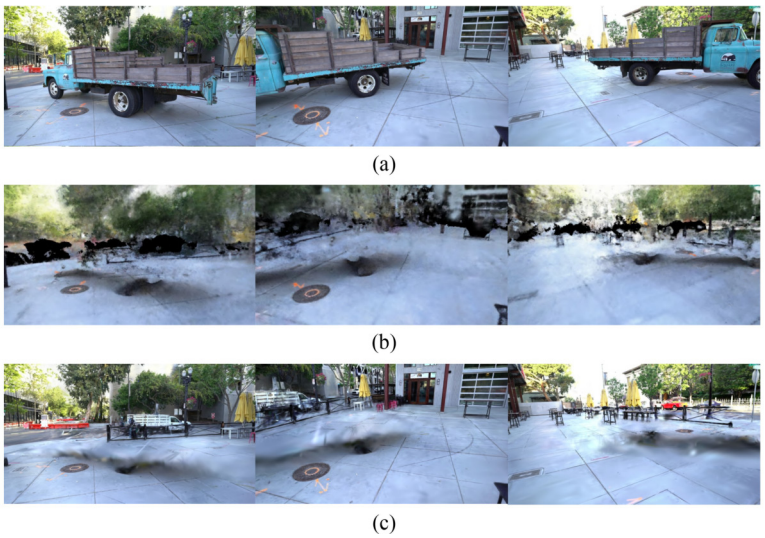


Fig. 12 Compared with DFF [10], Gaussian Grouping can remove the object with greatly reduced artifacts without leaving a blurry background. (a) Rendered image; (b) DFFs; (c) Gaussian Grouping. Results sourced from Gaussian Grouping [56]

objects based on their identity. It enables object removal, inpainting, and recomposition. Its object removal result is shown in Fig. 12.

The methods discussed above have made significant progress in advancing 3D scene manipulation but still struggle with some issues. One major issue is the appearance of holes and artifacts in the background after object movement. This could be resolved by utilizing physics-based constraints or causal relationship guidance. Another challenge was handling dynamic scenes, which may be resolved by optimizing models to enable real-time updates and improve their suitability for dynamic, interactive applications.

3.3 Shape deformation

Shape deformation refers to the modification of the geometry of 3D content while preserving its internal density, texture, and color information. This process enables adjustments to the overall shape and structure of the objects or scenes while ensuring consistency in their visual appearance and physical properties.

NeRF-Editing [12] is the first mesh-based shape deformation method, which leverages explicit representations for scene geometry edits and implicit representations for realistic rendering effects. It begins by extracting an explicit triangular mesh from the NeRF, enabling intuitive user-driven deformations through the classic ARAP deformation method. A tetrahedral mesh is then generated around this triangular mesh to propagate deformations spatially. By employing tetrahedral vertex interpolation, discrete deformations are transformed into a continuous field, ensuring that rays passing through the mesh bend according to user adjustments. The pipeline of NeRF-Editing is shown in Fig. 13.

The subsequent methods also realize the geometric deformation of NeRF with the guidance of mesh, specifically employing tetrahedral mesh [41,42,121,122], utilizing cage-building [13,42,123], associating each mesh vertex with optimizable features [124,125].

Compared with the mesh-based methods, NeuralEditor [40] employs a point cloud-guided NeRF model for shape manipulation. Leveraging K-D trees and a deterministic integration strategy, it achieves accurate point cloud-based rendering, while Phong reflection modeling further enhances color fidelity and geometric detail. This work excels in both shape deformation and scene morphing tasks.

The following methods [14,43–45,126] also utilize triangular meshes for the manipulation of 3DGS. SuGaR [43] efficiently extracts meshes from 3DGS by utilizing a regularization term to align the Gaussians with the scene surface and employing Poisson reconstruction [127] for efficient mesh extraction. Additionally, it bounds the Gaussians to the mesh and jointly optimizes the Gaussians and the mesh, resulting in more accurate meshes, and enabling scene editing using traditional mesh-editing tools. In contrast to SuGaR [43], which relies on a complex dual-stage mesh

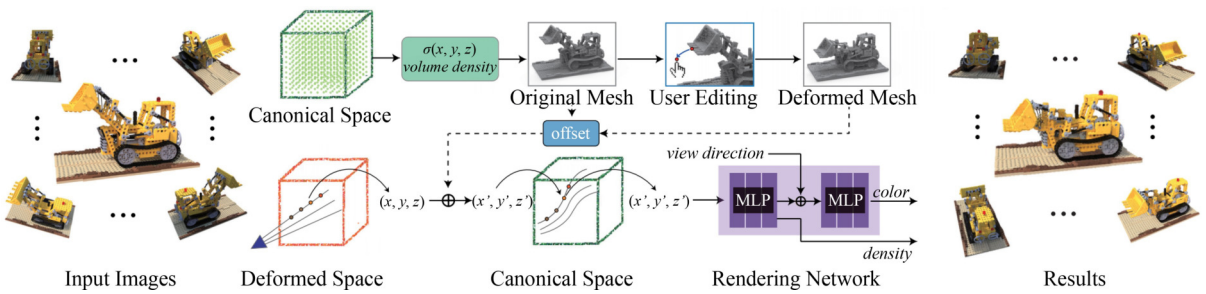


Fig. 13 The pipeline of NeRF-Editing. Image sourced from [12]

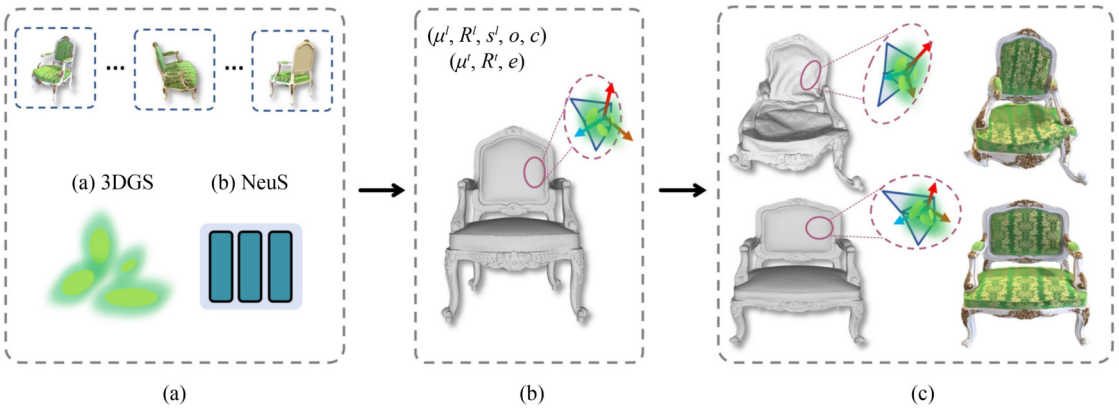


Fig. 14 An overview of Mani-GS, a mesh-based 3DGS deformation method. Image sourced from [45]. (a) Mesh Extraction; (b) GS Binding; (c) GS Manipulation

generation, GaMeS [14] simplifies the process by utilizing the initial mesh or estimating it in a single-stage training process. It introduces a pseudo-mesh, consisting of unconnected triangle faces, with Gaussian components position on these faces to ensure proper alignment. It realizes deformation by modifying pseudo-mesh positions, with these alterations automatically propagating to the Gaussian mesh, enabling real-time animation and scene editing. However, it exhibites artifacts when dealing with significant deformations in meshes with large faces, leading to inaccuracies.

Mesh-GS [44] further realizes large-scale 3DGS deformation, which utilizes a reconstructed explicit mesh derived from calibrated images to initialize Gaussians. It employs face split and normal guidance strategies to guide Gaussian learning, and introduces a regularization loss to preserve Gaussian shape. The user-controlled mesh deformations incorporates mesh-based Gaussian splatting representation to achieve real-time deformation. Gaussian Frosting [126] combines the editing ability of a mesh with the rendering quality of 3DGS, which extracts a base mesh based on SuGaR [43] and introduces an adaptive Gaussian layer with variable thickness to enhance detail and volumetric effects. It introduces a parameterization of the Gaussians to constrain them within the layer and automatically adjust them during base mesh manipulation.

Unlike previous methods that rely heavily on mesh accuracy, Mani-GS [45] defines a local triangle space for each triangle, ensuring robust and high-fidelity results even with inaccurate meshes. Gaussians are bound to the mesh and their attributes are optimized within this space, as shown in Fig. 14. The 3DGS is manipulated by transferring the mesh manipulation directly. It enables large deformation, soft body simulation, and local manipulation. Its shape editing result is shown in Fig. 15.

In addition to mesh-based methods, non-mesh-based methods are also viable for deforming 3DGS. SC-GS [46] utilizes sparse control points to drive dense Gaussians. The control points are linked to time-varying 6 DoF transformations, which are locally interpolated to generate the motion of dense Gaussians. These parameters are predicted by an MLP based on time and location. The method jointly learns the 3D Gaussian parameters and sparse control points in canonical space, along with the MLP for dynamic novel view

Fig. 15 Compared with SuGaR [43], Mani-GS produces fewer artifacts and reduces the blurring effect. (a) Manipulation; (b) SuGaR; (c) Mani-GS. Results sourced from Mani-GS [45].

synthesis. It employs an adaptive strategy to handle motion complexity and an ARAP loss to maintain local rigidity. Its overview is shown in Fig. 16.

Interactive3D [47] operates in two stages to enable interactive 3D object generation and refinement. In the first stage, it optimizes 3D objects through 3DGS and SDS, allowing flexible interactions like adding and removing parts, geometry transformation, deformable or rigid dragging, and semantic editing. In the second stage, the model

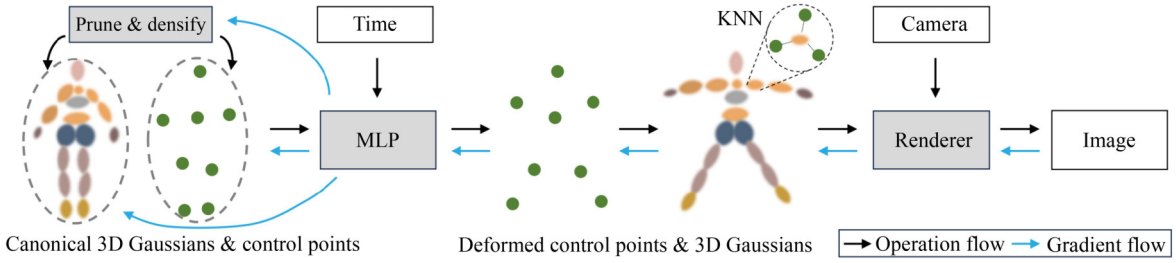


Fig. 16 An overview of SC-GS, a control points-based 3DGS deformation method. Image sourced from [46]

is transformed into an InstantNGP structure, with an interactive hash refinement module applied to improve specific areas with detailed geometry and textures.

Subsequent works have extended 3D or 4D deformation based on a broader range of instruction types, including text, video, and sketch. These advancements enable more versatile and intuitive interactions for editing 3D or 4D content. Align Your Gaussians (AYG) [48] realizes text-to-4D synthesis and deformation, which employs dynamic 3D Gaussians with deformation fields. It starts with a 3D-aware diffusion model and text-to-image model to create an initial 3D shape, followed by employing a text-to-video diffusion model to capture scene motion and utilizing a text-to-image model to maintain high visual quality for all frames. DreamGaussian4D [49] generates dynamic 4D scenes from a single image or video. It proposes DreamGaussianHD based on DreamGaussian [128] for image-to-3D initialization and utilizes HexPlane to model Gaussian deformations for realistic motion. Instead of using video diffusion prior like AYG [48], it learns the motion from a driving video generated by an image-to-video model. The generated 4D GS could be exported as an animated mesh, with optional refinement for better temporal coherence using a video-to-video optimization pipeline.

Another work [129] introduces a sketch-guided 3DGS deformation system. It introduces a cage-based deformation framework regulated by Neural Jacobian Fields, ensuring smooth and geometrically consistent transformations. It leverages ControlNet [84] to process sketches and employs 3D-aware SDS for cross-view consistency.

Existing methods demonstrate strong performance in shape deformation, yet they remaine susceptible to failure under extreme or unreasonable manipulations, often relying on user discretion for optimal results. Furthermore, real-time deformation is computationally expensive, limiting the interactivity and practicality of these methods. To improve their reliability and scalability, it is crucial to enhance the robustness of the models, supporting more intuitive user guidance, and leveraging hardware acceleration.

3.4 Scene inpainting

Scene inpainting is committed to the meticulous restoration of incomplete or occluded 3D scenes by estimating and synthesizing missing geometric and textural information. This process ensures that the newly generated content seamlessly integrates with the existing scene and remains consistent across different viewpoints. Most existing approaches rely on transferring 2D inpainting results from their respective models [83,130,131] to 3D scenes.

NeRF-In [132] realizes a simple 3D inpainting method, which trains NeRF with pixel reconstruction loss guided by 2D inpainted images generated by MST inpainting network [130]. SPIn-NeRF [15] leverages LaMa [131] for color and depth inpainting and improves on NeRF-In [132] by replacing pixel loss with perceptual loss, allowing for more visually coherent results. Additionally, it utilizes depth inpainting results to supervise the geometry of the 3D scene. Its overview is shown in Fig. 17. However, it struggles in tasks where 2D inpainting results are perceptually inconsistent.

Removing-NeRF [133] utilizes posed RGB-D images with corresponding 2D masks as input and optimizes a NeRF with 2D inpainted frames from LaMa [131]. It employs confidence-based view selection to automatically exclude inconsistent views during optimization.

Reference-guided inpainting [134] only performs 2D inpainting (SD [83] or LaMa [131]) on one reference image and utilizes monocular depth estimation to calculate its disparity map for supervising reference view geometry. It also employs a bilateral solver for rendering view-dependent effects across multiple views, ensuring consistency even in non-reference disoccluded regions. InNeRF360 [35] realizes the deletion of text-specified objects

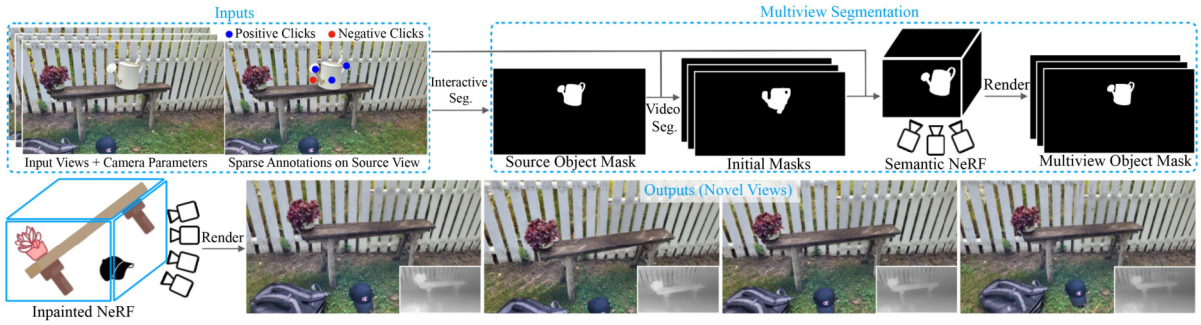


Fig. 17 An overview of SPIn-NeRF, a method for 3D scene inpainting. Image sourced from [15]

through two primary stages, multiview consistent segmentation and inpainting. It initializes segmentations through the SAM [119] and applies depth-warping refinement to obtain view-consistent 2D segmentations. It trains a NeRF from multiview inpainting results generated by a 2D image inpainter [83] and finetunes the scene with 3D diffusion priors to eliminate artifacts.

Inpaint3D [36] utilizes a self-trained 2D inpainting diffusion model, which adopts reconstruction loss for supervising unmasked regions and SDS loss [89] for optimizing masked regions. MVIP-NeRF [135] also utilizes diffusion priors for multiview-consistent inpainting. Given an RGB sequence and per-frame masks, the model is trained with a reconstruction loss in observed regions and an inpainting loss in masked regions, where the inpainting loss leverages SDS to align rendered views with diffusion priors based on a text description. It incorporates a multi-view scoring mechanism to improve consistency and sharpness, especially under large view variations.

Nerfiller [17] focuses on scene completion with arbitrary masks rather than object removal with tight masks. It utilizes a grid-based prior where images are shuffled into 2×2 grids and grids are fed into 2D inpainting diffusion models to generate 3D consistent outputs. It further introduces an iterative framework to distill 2D inpainting results into a single 3D scene. MALD-NeRF [37] employs a per-scene customization latent diffusion model to generate inpainted images, iteratively updates the training set, and utilizes a pixel-level regression loss for reconstruction. It also incorporates a masked patch-based adversarial training strategy with adversarial loss and discriminator feature matching losses to refine results.

Recent advancements [16,38,39,136,137] have increasingly focused on exploring inpainting techniques on 3DGS, benefiting from its outstanding rendering efficiency and high-quality scene reconstruction. InFusion [16] realizes 3D Gaussian inpainting that integrates depth completion based on diffusion models. It selects a reference image for inpainting, employs its self-trained depth completion model to predict the depth map of the inpainting region, and then projects it back into point clouds, thereby achieving accurate initial completion. It also designs a progressive strategy to resolve complex cases that could not be completed from a single view. RefFusion [38] also inpaints one reference image and applies the inpainting LDM to both global and locally cropped versions of the reference view. It utilizes SDS loss [89] to distill the learned priors from the adapted LDM into the scene, incorporating a discriminator loss and depth loss to improve the quality of inpainting results. Its object removal result is shown in Fig. 18 and the quantitative evaluation is presented in Table 4.

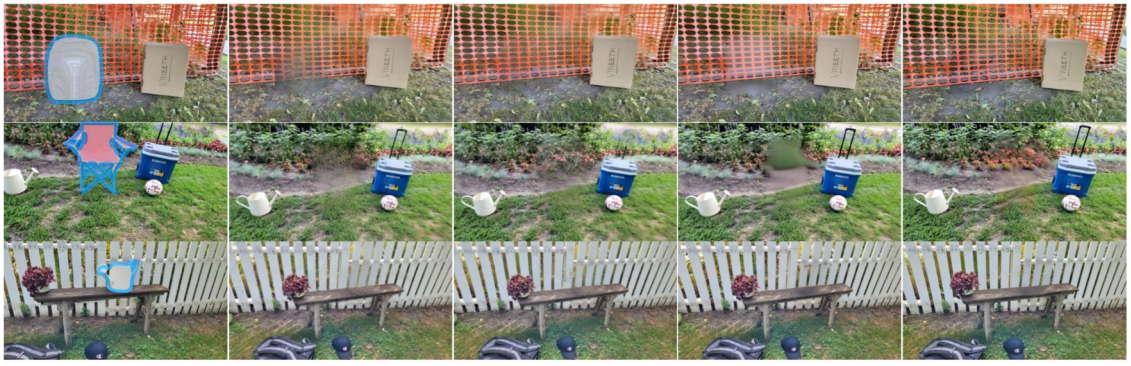
MVInpainter [39] is a multi-view consistent inpainting model designed to bridge 2D and 3D scene editing. It employs domain adapters and motion modules as video priors to enhance cross-view consistency. It also introduces a Reference Key&Value concatenation technique for appearance consistency while incorporating slot attention to aggregate optical flow features for controlling camera movement without explicit pose conditions.

Despite notable progress, existing scene inpainting methods face several limitations and challenges. A primary issue is accurately removing large target objects or inpainting areas with complex geometry and textures. Improving the personalization of 2D diffusion models for target objects is crucial to enhancing the performance and adaptability of 3D inpainting techniques.

Another critical challenge is ensuring multi-view and temporal consistency, as artifacts and holes frequently emerge when the camera viewpoint changes significantly or in dynamic scenes. Recent developments in multi-view and video diffusion models show promise in enhancing spatial and temporal consistency. By

Table 4 Quantitative results of completions after object removal, RefFusion surpassed all baselines in terms of LPIPS_{dir}. Results sourced from [38]

Method	LPIPS ↓
NeRF-In [132]	0.4884
SPIn-NeRF (SD) [15]	0.5701
SPIn-NeRF (LaMa) [15]	0.4654
Inpaint3D [36]	0.5150
Reference-guided inpainting (SDV2) [134]	0.4532
Reference-guided inpainting (SDXL) [134]	0.4453
RefFusion [38]	**0.4283**



(a) (b) (c) (d) (e)

Fig. 18 Compared with baseline methods, RefFusion obtaines sharper reconstructions and more realistic inpainting results. Results sourced from RefFusion [38]. (a) Masked Image; (b) SPIn-NeRF; (c) Reference-guided NeRF; (d) Inpaint3D; (e) RefFusion

addressing these limitations, scene inpainting methods can become more versatile and practical for complex real-world applications.

3.5 Creative editing

Creative Editing refers to the process of modifying 3D scenes based on user-provided prompts, allowing for changes in the geometry and texture of 3D content. Creative editing empowers users to intuitively control and customize 3D content, providing flexibility for artistic expression while maintaining coherence within the scene.

Early work on 3D creative editing primarily rely on the Contrastive Language-Image Pretraining (CLIP) model [138], an advanced multimodal learning framework that train both an image encoder and a text encoder using contrastive learning, as illustrated in Fig. 19.

CLIP similarity loss and CLIP directional similarity loss are pivotal techniques derived from the CLIP model. The CLIP similarity loss aims to quantify the similarity between the image I and the text prompt T in the embedding space, defined by:

$$\mathcal{L}_{\text{CLIP}}(I,T) = 1 - sim(E_{\text{img}}(I), E_{\text{txt}}(T)),$$

where E_{img} and E_{txt} are the pre-trained CLIP image encoder and text encoder, respectively. $sim(A,B)$ is the cosine similarity between two feature vectors. The CLIP directional similarity loss extends the concept of similarity by considering the alignment between the changes in images and text prompts before and after editing:

$$\mathcal{L}_{\text{dir}}(I_{\text{target}}, I_{\text{source}}, T_{\text{target}}, T_{\text{source}}) = 1 - sim(\Delta I, \Delta T),$$

where $\Delta I = E_{\text{img}}(I_{\text{target}}) - E_{\text{img}}(I_{\text{source}})$ is the change in images and $\Delta T = E_{\text{txt}}(T_{\text{target}}) - E_{\text{txt}}(T_{\text{source}})$ is the change in texts.

CLIP-NeRF [139] is the pioneering CLIP-based NeRF editing method, introducing a disentangled conditional NeRF with separate shape and appearance codes. It employs shape and appearance code mappers, optimized using CLIP similarity loss, to project CLIP features into the latent space, allowing independent manipulation of shape and appearance codes. NeRF-Art [140] stylizes the original NeRF by leveraging relative directional loss and global-local contrastive loss within the CLIP embedding space. It also introduces a weight regularization to enhance results and reduce the presence of artifacts.

Blended-NeRF [141] and Blending-NeRF [142] are committed to enabling local editing. Blended-NeRF [141] utilizes a CLIP similarity loss to generate a new object and blended the edited scene seamlessly into the original scene, allowing for targeted editing within the specified regions of interest (ROI) box. Blending-NeRF [142], on the other hand, employs CLIPSeg [120], a pre-trained 2D segmentation model, to localize the targeted region. It leverages CLIP similarity loss and CLIP directional loss to optimize the edited NeRF, blending the pre-trained and editable NeRFs for localized adjustments.

The CLIP-based 3D editing and generation methods [139–143] have achieved promising results. However, due to the limited capabilities of the CLIP model, its effectiveness is restricted to a limited range of tasks. Consequently, subsequent research has predominantly shifted towards more capable diffusion models, which offer significantly enhanced flexibility and performance.

Diffusion models [83,86] have become paragons of excellence in the domain of high-fidelity image generation and editing. By leveraging the remarkable ability of diffusion models, recent studies have unlocked new possibilities for manipulating and customizing 3D scenes. The integration of natural text and image instructions allows for the precise and intuitive modification of scenes, enabling users to obtain desired results with remarkable ease and specificity. The optimization approaches for diffusion-based editing methods can be categorized into two main types: Dataset Update (DU) and Score Distillation Sampling (SDS) algorithm.

In Table 5, we provide a comprehensive overview of various diffusion-based editing methods. This table details the year of publication, the types of instructions they utilize, optimization approaches, and base techniques.

3.5.1 Dataset update

This series of work leverage diffusion models to directly manipulate the scene images and update the training dataset, which enables precise and intuitive alterations of NeRF and 3DGS.

Instruct-NeRF2NeRF (IN2N) [18] is the first work to edit NeRF by iteratively updating the training dataset with the edited image. It performs image editing through the text-based image editing model InstructPix2Pix (IP2P) [86]. An overview of IN2N is shown in Fig. 20. However, since IN2N [18] applieds edits to entire 2D

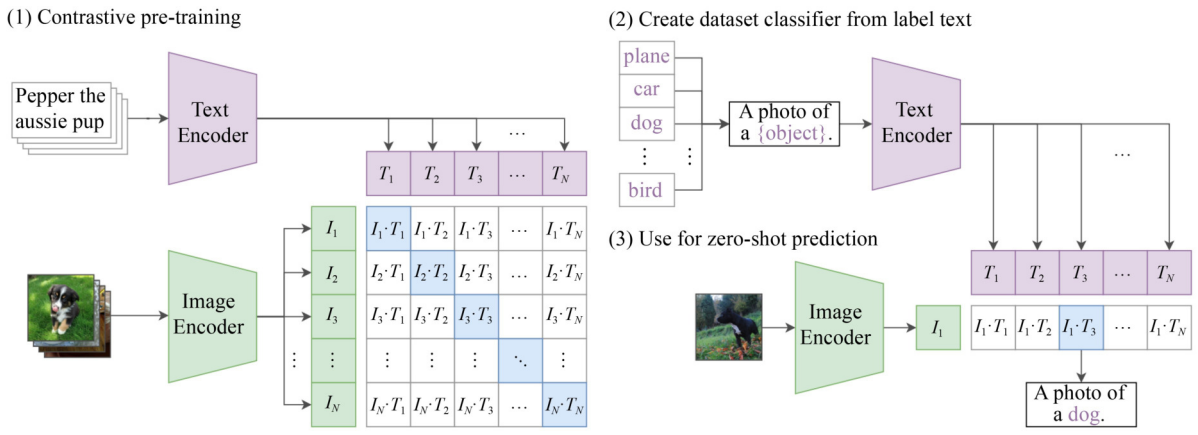


Fig. 19 An overview of CLIP. Image sourced from [138]

Table 5 Summary of selected diffusion-based creative editing methods

Method	Year	Prompt	Optimization	Base techniques
Instruct-NeRF2NeRF [18]	ICCV 2023	text	DU	NeRFStudio [144], IP2P [86]
ViCA-NeRF [28]	NeurIPS 2023	text	DU	NeRFStudio [144], IP2P [86]
Watch your steps [145]	ECCV 2024	text	DU	NeRFStudio [144], IP2P [86]
LatentEditor [29]	ECCV 2024	text	DU	NeRF [6], IP2P [86]
GenN2N [27]	CVPR 2024	text	DU	NeRFStudio [144], IP2P [86]
GaussianEditor [30]	CVPR 2024	text	DU	3DGS [7], IP2P [86], SAM [119]
GaussianEditor [31]	CVPR 2024	text	DU or DDS	3DGS [7], IP2P [86], SAM [119]
View-consistent Editing [32]	ECCV 2024	text	DU	3DGS [7], LDM [83]
GaussCtrl [33]	ECCV 2024	text	DU	3DGS [7], ControlNet [84], SAM [119]
Instruct 3D-to-3D [146]	arxiv 2023	text	SDS	DVGO [97], IP2P [86]
SKED [147]	ICCV 2023	text	SDS	Instant-NGP [108], SD v1.4 [83]
Vox-E [22]	ICCV 2023	text	SDS	ReLU Fields [148], SD v2.1 [83]
MaskEditor [149]	PRCV 2024	text	VSD	DVGO [97], SD v2.1 [83]
DreamEditor [19]	SIGGRAPH Asia 2023	text	SDS	NeuS [150], SD v2.0 [83], DreamBooth [151]
Focaldreamer [20]	AAAI 2024	text	SDS	DMTet [152], SD v2.1 [83]
CustomNeRF [23]	CVPR 2024	text or image	SDS	Instant-NGP [108], SD v1.5 [83]
TIP-Editor [21]	SIGGRAPH 2024	text and image	SDS	3DGS [7], SD v2.0 [83], DreamBooth [151]
GSEditPro [24]	PG 2024	text	SDS	3DGS [7], SD v2.1 [83], DreamBooth [151]
TIGER [25]	arxiv 2024	text	CSD	3DGS [7], MaskCLIP [153], FeatUp [154]

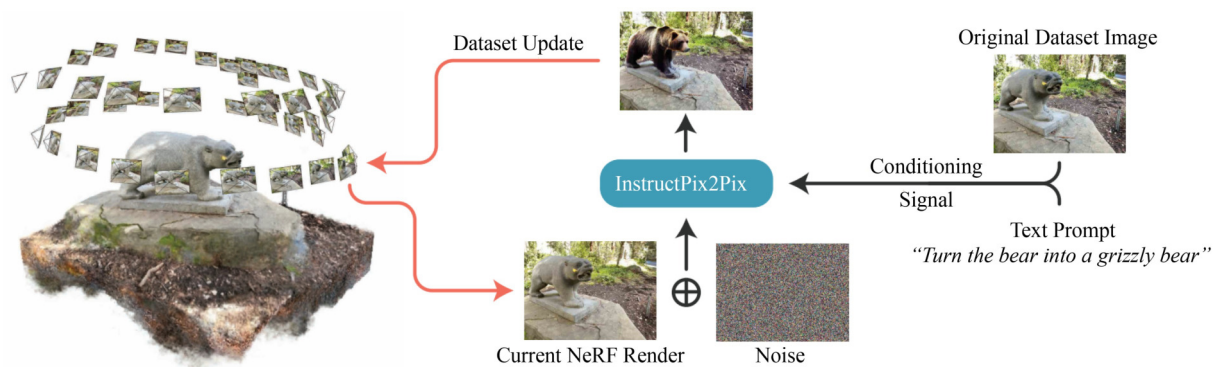


Fig. 20 An overview of Instruct-NeRF2NeRF, a method of 3D editing of NeRF via dataset update. Image sourced from [18]

images, it lacks the ability to perform precise, localized modifications.

To enhance editing efficiency and consistency, ViCA-NeRF [28] builds upon IN2N [18] with some improvements. It focuses on editing key views with IP2P [86], projecting them onto other views, and subsequently refining them with a blending model. Then, the original NeRF is optimized using the edited dataset.

Watch your steps [145] further realizesd local editing. It leverages IP2P [86] to generate relevance maps, which are derived from the difference between noise predictions conditioned on text instructions and empty text. The binarized relevance maps are then extended into 3D via relevance fields to realize local editing. LatentEditor [29] also focuses on locally controlled editing. It initializes and refines a NeRF model within the latent domain, using a refinement adapter with residual and self-attention mechanisms. It iteratively updates the training set with edited latents and employes a delta module for targeted editing guided by prompts.

GenN2N [27] proposes a versatile editing framework for various editing tasks. For each view, multiple edited images are generated by 2D editing methods, forming a set of translated images. A latent

distill module is introduced to map edited images to edit code vectors, which are then constrained to a Gaussian distribution through a KL loss. The NeRF-to-NeRF translation is performed based on edit codes and the translated NeRF is optimized by using reconstruction loss, adversarial loss, and contrastive loss.

The latest series of work is based on 3DGS, achieving higher editing efficiency and superior rendering quality. GaussianEditor [30] tailored for local editing on 3DGS, which consists of three primary steps: extracting the region of interest (RoI) from text instructions, aligning the text RoI with 3D Gaussians through the image space, and delicately editing within the identified 3D Gaussian RoI. Another GaussianEditor [31] introduces Gaussian semantic tracing for local editing and hierarchical Gaussian splatting (HGS) for adaptive generative guidance. The authors implemented two versions of this method, GaussianEditor-iN2N guided by IN2N [18] and GaussianEditor-DDS guided by DDS [91]. GaussianEditor-iN2N achieves superior results, as shown in Fig. 21.

View-consistent Editing (VcEDIT) [32] enhances the consistency of editing results with two modules, the cross-attention consistency module (CCM) and the editing consistency module (ECM). CCM harmonizes cross-attention maps across all views for coherent edits, while ECM calibrates inconsistent editing outputs by fine-tuning a source-cloned 3DGS and rendering it back to images. Additionally, it introduces an iterative pattern of editing rendered images and updating the 3DGS for refinement, integrating 3DGS for fast rendering and InfEdit [155] for rapid processing. GaussCtrl [33] also focuses on multi-view consistent editing, which employs ControlNet [84] for depth-conditioned editing on rendered images from the original 3DGS, with attention-based latent code alignment ensuring geometry and appearance consistency.

3.5.2 SDS

Similar to the task of 3D content generation, another series of 3D editing works utilize the SDS algorithm [89] to optimize NeRF and 3DGS.

Instruct 3D-to-3D [146] realizes 3D-to-3D conversions following text instructions with InstructPix2Pix. By integrating the source 3D scene as a conditional input, it utilizes SDS loss [89] to optimize the target scene and integrates dynamic scaling to effectively modify scene geometry.

SKED [147] is a 3D editing method guided by sketches and text prompts. It employs SDS loss [89] to edit the original object. It proposes silhouette loss to facilitate the addition of new objects within the sketch region and preservation loss to maintain fidelity to the original scene.

The following methods are dedicated to achieving localized 3D object editing guided by text prompts. Vox-E [22] is a grid-based method, which leverages SDS loss [89] for object editing and introduced a volumetric regularization to preserve the original structure. It also optimizes a 3D cross-attention grid based on 2D cross-attention maps, deriving a binary volumetric mask to merge the initial and edited volumetric fields, enhancing the preservation of unaffected areas. MaskEditor [149] is another grid-based 3D editing method. It introduces a 3D mask grid trained on the 2D masks from the SAM [119] to locate the target object accurately. Instead of utilizing SDS [89], it employs Variational Score Distillation (VSD) [90] along with composited rendering and coarse-to-fine editing strategy to improve the editing quality.

In addition to employing grid-based scene representation, certain methods utilized mesh-based representation for local editing. DreamEditor [19] firstly converts the original neural field into a mesh-based representation and then back-projects the 2D mask onto the mesh to identify editing regions. It utilizes SDS loss [89] to optimize color and geometry features and vertex positions within the selected regions, facilitating precise local adjustments.

Focaldreamer [20] is designed to generate new objects in empty

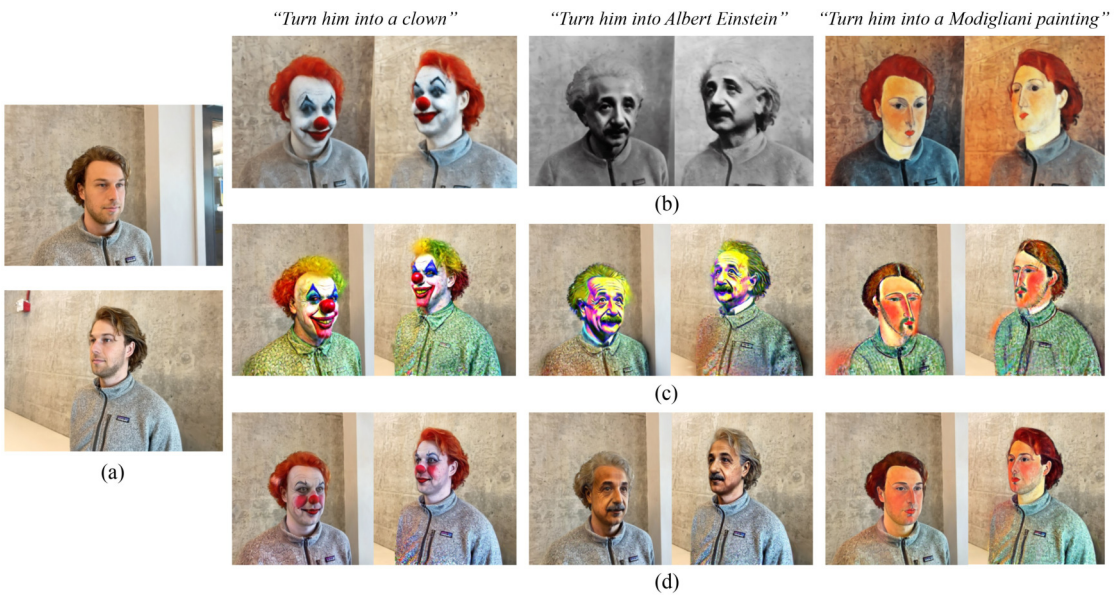


Fig. 21 Compared with Instruct-N2N [18], GaussianEditor maintaines precise control over the editing area. (a) Original view; (b) Instruct-N2N; (c) GSEditor-DDS; (d) GSEditor-iN2N. Results sourced from GaussianEditor [31]

regions according to text prompts. It initiates with a base shape and editable ellipsoids in focal regions, utilizing rendered normal maps as shape encoding of T2I models for geometry optimization. The shape is rendered with both base and editable textures, unified by a pixel-wise discriminative mask. In addition to the SDS loss [89], it incorporates several regularizations to improve editing quality. In addition to using text prompts, CustomNeRF [23] can also utilize images as editing prompts to achieve specific editing. It introduces a local-global iterative editing (LGIE) training scheme to preserve background integrity and proposes a class-guided regularization to improve consistency across different views.

The latest series of methods [21,24-26, 156] employ SDS loss [89] within the 3DGS to facilitate creative editing, enabling precise and flexible modifications.

TIP-Editor [21] aims to edit existing 3DGS via intuitive text and image prompts. It introduces a novel stepwise 2D personalization strategy that includs a localization loss for scene adaptation and a content personalization step based on LoRA for reference image alignment. It optimizes the identified Gaussians with SDS loss [89] and pixel-level image refinement to realize accurate local editing. Its editing result is shown in Fig. 22. Table 6 demonstrates that the editing results achieved by TIP-Editor exhibit superior quality compared with the baselines.

GSEditPro [24] leverages attention-based localization to precisely identify editing areas based on cross-attention maps from T2I models. It also employs a detailed optimization process that combined DreamBooth [151] and pseudo-GT images, ensuring high-quality editing results while minimizing unnecessary modifications. TIGER [25] incorporates fine-grained language features through MaskCLIP [153] and FeatUp [154] to enable open-vocabulary querying. Moreover, it introduces Coherent Score Distillation Sampling (CSD) to ensure edits remain consistent across multiple views.

In addition to the manipulation of static scenes, several studies have focused on the creative editing of complex dynamic scenes based on NeRF and 3DGS. These works extend beyond traditional scene editing by tackling the unique challenges associated with dynamic scenes, particularly the need to seamlessly modify both spatial structures and time-varying elements.

AvatarStudio [157] leverages NeRF to represent the head avatar and utilizes a diffusion model for text-based editing. It samples keyframes from multi-view performance captures, and fine-tunes a pre-trained model with a unique text identifier. The view- and time-aware Score Distillation Sampling (VT-SDS) enable the high-quality personalized editing across the time and view domain. Additionally, an annealing strategy is employed to prevent overfitting while allowing for high-frequency edits. CoDeF [158] is a novel video representation method comprising two components: a 2D canonical

Fig. 22 Compared with Instruct-N2N [18] and DreamEditor [19], TIP-Editor achieves superior editing quality and fidelity to reference images. (a) Reference image; (b) Instruct-N2N; (c) DreamEditor; (d) TIP-Editor. Results sourced from TIP-Editor [21]

Table 6 Quantitative editing results on Instruct-N2N [18], DreamEditor [19], and TIP-Editor [21], in terms of CLIP_{dir}, DINO_{sim}, and User Study. User Study is conducted from two aspects (overall "Quality", and "Alignment" to the reference image). Results sourced from [21]

Method	$\text{CLIP}_{\text{dir}}\uparrow$	$\text{DINO}_{\text{sim}}\uparrow$	$\text{Vote}_{\text{quality}}$	$\text{Vote}_{\text{alignment}}$
Instruct-N2N [18]	8.3	36.4	21.6	8.8
DreamEditior [19]	11.4	36.8	7.6	10.0
TIP-Editor [21]	**15.5**	**39.5**	**70.8**	**81.2**

content field and a 3D temporal deformation field, both using multi-resolution hash tables. It allows the direct application of image processing algorithms [84] to the canonical image, with results propagated across time via the temporal deformation field, enabling efficient video processing. However, it is limited by optical flow accuracy and has difficulty in modeling complex motion. PortraitGen [34] focuses on editing efficiency and rendering quality. It embeds the 3D Gaussian field on the surface of SMPL-X [159] for spatial and temporal consistency and utilizes a Neural Gaussian Texture mechanism to get a 3D Gaussian feature field for high-quality rendering. It employs the iterative dataset update strategy for portrait editing, incorporating expression similarity guidance, and a face-aware module to maintain facial structure and expression accuracy.

To further advance 3D creative editing, enhancing the capabilities of 2D diffusion models is essential. Current editing methods primarily rely on diffusion models for dataset updating or SDS. However, 2D diffusion models often lack the spatial and geometric awareness required for precise 3D editing and struggle to perform effectively in some complex editing tasks. To overcome these limitations, future advancements should focus on extending diffusion models to incorporate inherent 3D awareness and support a broader range of editing tasks. Meanwhile, further optimizing the model and introducing additional mechanisms to enable automated and efficient local editing, ensure multi-view consistency, and achieve real-time interactive 3D editing are promising and impactful directions for future research.

4 Future work

While 3D editing methods based on NeRF and 3DGS have achieved significant advancements, the following aspects are worth improving and optimizing in future work.

- *Efficient editing.* Although current editing methods produce impressive visual effects, they still suffer from low computational efficiency, limiting their practicality for real-time applications and large-scale deployments. Future efforts should focus on implementing innovative approaches and architectures to accelerate the editing process. By utilizing techniques such as pre-trained models, advancements in hardware acceleration, and model parallelization, it is possible to create fast, responsive editing workflows without sacrificing visual fidelity.
- *Multi-view consistency.* Ensuring consistency of editing results across various viewpoints remains a challenge in 3D editing. Future work should prioritize the enhancement of multi-view consistency by developing more robust optimization mechanisms that ensure updates from different perspectives remain synchronized and aligned with the underlying 3D structure. This could involve incorporating multi-view stereo constraints or leveraging depth constraints to preserve the integrity of the 3D model throughout the editing process.
- *Diversity.* Enhancing support for diverse instruction types is crucial for advancing 3D editing methods. Although significant progress has been made in text- and image-driven editing, several instruction types are insufficient to meet the increasingly complex and diverse user demands. To address this, incorporating additional instruction types, such as voice commands, gesture-based inputs, and even physical sensing, could significantly broaden the scope of 3D editing capabilities.
- *Scalability.* Scalability is another critical issue, as existing editing methods often struggle with large-scale scenes due to memory and computational limitations. To address this challenge, future work should explore scalable scene representation methods and investigate hierarchical or multi-scale modeling and editing approaches, which should be capable of handling extensive environments without sacrificing detail or performance, thereby facilitating complex edits in large-scale scenes.
- *Robustness and accessibility.* Enhancing the robustness of 3D editing methods to handle variations in input data is crucial for widespread application. This includes improving the ability of models to process noisy, incomplete, or low-resolution input data. Furthermore, to make these technologies more accessible to non-specialists, it is crucial to develop intuitive user interfaces and tools that simplify the editing process. Such interfaces could involve visual tools that enable users to manipulate 3D content seamlessly, without the need for specialized technical expertise.

5 Conclusion

In this survey, we presented a comprehensive overview of 3D editing methods based on NeRF and 3DGS. We systematically classified existing methods into five categories based on their editing tasks, analyzing the advancements, current challenges, and future research directions. This survey seeks to serve as a foundational reference for researchers and developers, aiming to foster the development of 3D editing.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62372457, 62325211, and 62132021), the Major Program of Xiangjiang Laboratory (23XJ01009), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), the Natural Science Foundation of Hunan Province of China (2022RC1104) and the NUDT Research Grants (ZK22-52).

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

References

[1] Nieto J R, Susín A. Cage based deformations: a survey. In: Hidalgo M G, Torres A M, Gómez J V, eds. Deformation Models: Tracking, Animation and Applications. Dordrecht: Springer, 2013, 75−99

[2] Gao L, Lai Y K, Yang J, Zhang L X, Xia S, Kobbelt L. Sparse data driven mesh deformation. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(3): 2085−2100

[3] Wang Y, Aigerman N, Kim V G, Chaudhuri S, Sorkine-Hornung O. Neural cages for detail-preserving 3D deformations. In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, 75−83

[4] Mullen T. Mastering blender. John Wiley & Sons, 2011

[5] Hu J Y. The application of computer software—3d studio max, lightscape and v-ray in the environmental artistic expression. Advanced Materials Research, 2013, 631: 1379–1384

[6] Mildenhall B, Srinivasan P P, Tancik M, Barron J T, Ramamoorthi R, Ng R. NeRF: representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 2022, 65(1): 99−106

[7] Kerbl B, Kopanas G, Leimkuehler T, Drettakis G. 3D Gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (TOG), 2023, 42(4): 139

[8] Kim G, Youwang K, Oh T H. FPRF: feed-forward photorealistic style transfer of large-scale 3D neural radiance fields. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 2750−2758

[9] Zhang D, Yuan Y J, Chen Z, Zhang F L, He Z, Shan S, Gao L. StylizedGS: controllable stylization for 3D Gaussian splatting. 2024, arXiv preprint arXiv: 2404.05220

[10] Kobayashi S, Matsumoto E, Sitzmann V. Decomposing NeRF for editing via feature field distillation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1694

[11] Yang B, Zhang Y, Xu Y, Li Y, Zhou H, Bao H, Zhang G, Cui Z. Learning object-compositional neural radiance field for editable scene rendering. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 13779−13788

[12] Yuan Y J, Sun Y T, Lai Y K, Ma Y, Jia R, Gao L. NeRF-Editing: geometry editing of neural radiance fields. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 18353−18364

[13] Xu T, Harada T. Deforming radiance fields with cages. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 159−175

[14] Waczyńska J, Borycki P, Tadeja S, Tabor J, Spurek P. GaMeS: mesh-based adapting and modification of Gaussian splatting. 2024, arXiv preprint arXiv: 2402.01459

[15] Mirzaei A, Aumentado-Armstrong T, Derpanis K G, Kelly J, Brubaker M A, Gilitschenski I, Levinshtein A. SPIn-NeRF: multiview segmentation and perceptual inpainting with neural radiance fields. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 20669−20679

[16] Liu Z, Ouyang H, Wang Q, Cheng K L, Xiao J, Zhu K, Xue N, Liu Y, Shen Y, Cao Y. InFusion: inpainting 3D Gaussians via learning depth completion from diffusion prior. 2024, arXiv preprint arXiv: 2404.11613

[17] Weber E, Holynski A, Jampani V, Saxena S, Snavely N, Kar A, Kanazawa A. NeRFiller: completing scenes via generative 3D inpainting. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 20731−20741

[18] Haque A, Tancik M, Efros A A, Holynski A, Kanazawa A. Instruct-NeRF2NeRF: editing 3D scenes with instructions. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 19740−19750

[19] Zhuang J, Wang C, Lin L, Liu L, Li G. DreamEditor: text-driven 3D scene editing with neural fields. In: Proceedings of the SIGGRAPH Asia 2023 Conference Papers. 2023, 26

[20] Li Y, Dou Y, Shi Y, Lei Y, Chen X, Zhang Y, Zhou P, Ni B. FocalDreamer: text-driven 3D editing via focal-fusion assembly. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 3279−3287

[21] Zhuang J, Kang D, Cao Y P, Li G, Lin L, Shan Y. TIP-Editor: an accurate 3D editor following both text-prompts and image-prompts. ACM Transactions on Graphics (TOG), 2024, 43(4): 121

[22] Sella E, Fiebelman G, Hedman P, Averbuch-Elor H. Vox-E: text-guided voxel editing of 3D objects. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 430−440

[23] He R, Huang S, Nie X, Hui T, Liu L, Dai J, Han J, Li G, Liu S. Customize your NeRF: adaptive source driven 3D scene editing via local-global iterative training. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 6966−6975

[24] Sun Y, Tian R, Han X, Liu X, Zhang Y, Xu K. GSEditPro: 3D Gaussian splatting editing with attention-based progressive localization. Computer Graphics Forum, 2024, 43(7): e15215

[25] Xu T, Chen J, Chen P, Zhang Y, Yu J, Yang W. TIGER: text-instructed 3D Gaussian retrieval and coherent editing. 2024, arXiv preprint arXiv: 2405.14455

[26] Mendiratta M, Pan X, Elgharib M, Teotia K, Mallikarjun B R, Tewari A, Golyanik V, Kortylewski A, Theobalt C. AvatarStudio: text-driven editing of 3D dynamic human head avatars. ACM Transactions on Graphics (ToG), 2023, 42(6): 1−18

[27] Liu X, Xue H, Luo K, Tan P, Yi L. GenN2N: generative NeRF2NeRF translation. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 5105−5114

[28] Dong J, Wang Y X. ViCA-NeRF: view-consistency-aware 3D editing of neural radiance fields. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2686

[29] Khalid U, Iqbal H, Karim N, Tayyab M, Hua J, Chen C. LatentEditor: text driven local editing of 3D scenes. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 364−380

[30] Wang J, Fang J, Zhang X, Xie L, Tian Q. GaussianEditor: editing

3D Gaussians delicately with text instructions. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 20902−20911

[31] Chen Y, Chen Z, Zhang C, Wang F, Yang X, Wang Y, Cai Z, Yang L, Liu H, Lin G. GaussianEditor: swift and controllable 3D editing with Gaussian splatting. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 21476−21485

[32] Wang Y, Yi X, Wu Z, Zhao N, Chen L, Zhang H. View-consistent 3D editing with Gaussian splatting. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 404−420

[33] Wu J, Bian J W, Li X, Wang G, Reid I, Torr P, Prisacariu V A. GaussCtrl: multi-view consistent text-driven 3D Gaussian splatting editing. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 55−71

[34] Gao X, Xiao H, Zhong C, Hu S, Guo Y, Zhang J. Portrait video editing empowered by multimodal generative priors. In: Proceedings of the SIGGRAPH Asia 2024 Conference Papers. 2024, 104

[35] Wang D, Zhang T, Abboud A, Susstrunk S. InNeRF360: text-guided 3D-consistent object inpainting on 360° neural radiance fields. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 12677−12686

[36] Prabhu K, Wu J, Tsai L, Hedman P, Goldman D B, Poole B, Broxton M. Inpaint3D: 3D scene content generation using 2D inpainting diffusion. 2023, arXiv preprint arXiv: 2312.03869

[37] Lin C H, Kim C, Huang J B, Li Q, Ma C Y, Kopf J, Yang M H, Tseng H Y. Taming latent diffusion model for neural radiance field inpainting. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 149−165

[38] Mirzaei A, De Lutio R, Kim S W, Acuna D, Kelly J, Fidler S, Gilitschenski I, Gojcic Z. RefFusion: reference adapted diffusion models for 3D scene inpainting. 2024, arXiv preprint arXiv: 2404.10765

[39] Cao C, Yu C, Wang F, Xue X, Fu Y. MVInpainter: learning multi-view consistent inpainting to bridge 2D and 3D editing. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024

[40] Chen J K, Lyu J, Wang Y X. NeuralEditor: editing neural radiance fields via manipulating point clouds. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 12439−12448

[41] Garbin S J, Kowalski M, Estellers V, Szymanowicz S, Rezaeifar S, Shen J, Johnson M A, Valentin J. VolTeMorph: real-time, controllable and generalizable animation of volumetric representations. Computer Graphics Forum, 2024, 43(6): e15117

[42] Jambon C, Kerbl B, Kopanas G, Diolatzis S, Leimkühler T, Drettakis G. NeRFshop: interactive editing of neural radiance fields. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2023, 6(1): 1

[43] Guédon A, Lepetit V. SuGaR: surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 5354−5363

[44] Gao L, Yang J, Zhang B T, Sun J M, Yuan Y J, Fu H, Lai Y K. Mesh-based Gaussian splatting for real-time large-scale deformation. 2024, arXiv preprint arXiv: 2402.04796

[45] Gao X, Li X, Zhuang Y, Zhang Q, Hu W, Zhang C, Yao Y, Shan Y, Quan L. Mani-GS: Gaussian splatting manipulation with triangular mesh. 2024, arXiv preprint arXiv: 2405.17811

[46] Huang Y H, Sun Y T, Yang Z, Lyu X, Cao Y P, Qi X. SC-GS: sparse-controlled Gaussian splatting for editable dynamic scenes. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 4220−4230

[47] Dong S, Ding L, Huang Z, Wang Z, Xue T, Xu D. Interactive3D: create what you want by interactive 3D generation. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024

[48] Ling H, Kim S W, Torralba A, Fidler S, Kreis K. Align your Gaussians: text-to-4D with dynamic 3D Gaussians and composed diffusion models. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024

[49] Ren J, Pan L, Tang J, Zhang C, Cao A, Zeng G, Liu Z. DreamGaussian4D: generative 4d Gaussian splatting. 2023, arXiv preprint arXiv: 2312.17142

[50] Tschernezki V, Laina I, Larlus D, Vedaldi A. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In: Proceedings of 2022 International Conference on 3D Vision. 2022, 443−453

[51] Cen J, Zhou Z, Fang J, Yang C, Shen W, Xie L, Jiang D, Zhang X, Tian Q. Segment anything in 3D with NeRFs. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 25971−25990

[52] Wu Q, Liu X, Chen Y, Li K, Zheng C, Cai J, Zheng J. Object-compositional neural implicit surfaces. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 197−213

[53] Zhou S, Chang H, Jiang S, Fan Z, Zhu Z, Xu D, Chari P, You S, Wang Z, Kadambi A. Feature 3DGS: supercharging 3D Gaussian splatting to enable distilled feature fields. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 21676−21685

[54] Cen J, Fang J, Yang C, Xie L, Zhang X, Shen W, Tian Q. Segment any 3D Gaussians. 2023, arXiv preprint arXiv: 2312.00860

[55] Hu X, Wang Y, Fan L, Luo C, Fan J, Lei Z, Li Q, Peng J, Zhang Z. SAGD: boundary-enhanced segment anything in 3D Gaussian via Gaussian decomposition. 2024, arXiv preprint arXiv: 2401.17857

[56] Ye M, Danelljan M, Yu F, Ke L. Gaussian grouping: segment and edit anything in 3D scenes. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 162−179

[57] Chiang P Z, Tsai M S, Tseng H Y, Lai W S, Chiu W C. Stylizing 3D scene via implicit representation and HyperNetwork. In: Proceedings of 2022 IEEE/CVF Winter Conference on Applications of Computer Vision. 2022, 1475−1484

[58] Zhang Y, He Z, Xing J, Yao X, Jia J. Ref-NPR: reference-based non-photorealistic radiance fields for controllable scene stylization. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 4242−4251

[59] Nguyen-Phuoc T, Liu F, Xiao L. SNeRF: stylized neural implicit representations for 3D scenes. ACM Transactions on Graphics (TOG), 2022, 41(4): 142

[60] Liu K, Zhan F, Chen Y, Zhang J, Yu Y, El Saddik A, Lu S, Xing E.

StyleRF: zero-shot 3D style transfer of neural radiance fields. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pat tern Recognition. 2023, 8338−8348

[61] Chen J, Xing W, Sun J, Chu T, Huang Y, Ji B, Zhao L, Lin H, Chen H, Wang Z. PNeSM: arbitrary 3D scene stylization via prompt-based neural style mapping. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 1091−1099

[62] Liu K, Zhan F, Xu M, Theobalt C, Shao L, Lu S. StyleGaussian: instant 3D style transfer with Gaussian splatting. In: Proceedings of the SIGGRAPH Asia 2024 Technical Communications. 2024, 21

[63] Wu Q, Tan J, Xu K. PaletteNeRF: palette-based color editing for NeRFs. Communications in Information and Systems, 2023, 23(4): 447−475

[64] Chen Y, Yuan Q, Li Z, Liu Y, Wang W, Xie C, Wen X, Yu Q. UPST-NeRF: universal photorealistic style transfer of neural radiance fields for 3D scene. IEEE Transactions on Visualization and Computer Graphics, 2025, 31(4): 2045−2057

[65] Zhang Z, Liu Y, Han C, Pan Y, Guo T, Yao T. Transforming radiance field with lipschitz network for photorealistic 3D scene stylization. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 20712−20721

[66] Barron J T, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R, Srinivasan P P. Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 5855−5864

[67] Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A. Plenoxels: radiance fields without neural networks. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 5501−5510

[68] Tang Y, Zhang J, Yu Z, Wang H, Xu K. MIPS-Fusion: multi-implicit-submaps for scalable and robust online neural RGB-D reconstruction. ACM Transactions on Graphics (TOG), 2023, 42(6): 246

[69] Ye Y, Yi R, Gao Z, Zhu C, Cai Z, Xu K. NEF: neural edge fields for 3D parametric curve reconstruction from multi-view images. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 8486−8495

[70] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. 2016, 2414−2423

[71] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations. 2015

[72] Johnson J, Alahi A, Li F F. Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of the 14th European Conference on Computer Vision. 2016, 694−711

[73] Ulyanov D, Lebedev V, Vedaldi A, Lempitsky V. Texture networks: feed-forward synthesis of textures and stylized images. In: Proceedings of the 33rd International Conference on Machine Learning. 2016, 1349−1357

[74] Li Y, Fang C, Yang J, Wang Z, Lu X, Yang M H. Diversified texture synthesis with feed-forward networks. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. 2017, 3920−3928

[75] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of 2017 IEEE International Conference on Computer Vision. 2017, 1501−1510

[76] Li X, Liu S, Kautz J, Yang M H. Learning linear transformations for fast arbitrary style transfer. 2018, arXiv preprint arXiv: 1808.04537

[77] Luan F, Paris S, Shechtman E, Bala K. Deep photo style transfer. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. 2017, 4990−4998

[78] Mechrez R, Shechtman E, Zelnik-Manor L. Photorealistic style transfer with screened Poisson equation. In: Proceedings of the British Machine Vision Conference 2017. 2017

[79] Li Y, Liu M Y, Li X, Yang M H, Kautz J. A closed-form solution to photorealistic image stylization. In: Proceedings of the 15th European Conference on Computer Vision. 2018, 453−468

[80] Yoo J, Uh Y, Chun S, Kang B, Ha J W. Photorealistic style transfer via wavelet transforms. In: Proceedings of 2019 IEEE/CVF International Conference on Computer Vision. 2019, 9036−9045

[81] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. 2014, 2672−2680

[82] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 574

[83] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 10684−10695

[84] Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 3836−3847

[85] Hu E J, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. LoRA: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations. 2022

[86] Brooks T, Holynski A, Efros A A. InstructPix2Pix: learning to follow image editing instructions. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 18392−18402

[87] Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 159

[88] Hertz A, Mokady R, Tenenbaum J, Aberman K, Pritch Y, Cohen-Or D. Prompt-to-prompt image editing with cross attention control. 2022, arXiv preprint arXiv: 2208.01626

[89] Poole B, Jain A, Barron J T, Mildenhall B. DreamFusion: text-to-3D using 2D diffusion. In: Proceedings of the 11th International Conference on Learning Representations. 2023

[90] Wang Z, Lu C, Wang Y, Bao F, Li C, Su H, Zhu J. ProlificDreamer: high-fidelity and diverse text-to-3D generation with

variational score distillation. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 368

[91] Hertz A, Aberman K, Cohen-Or D. Delta denoising score. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 2328−2337

[92] Koo J, Park C, Sung M. Posterior distillation sampling. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 13352−13361

[93] Wang Q, Wang Z, Genova K, Srinivasan P P, Zhou H, Barron J T, Martin-Brualla R, Snavely N, Funkhouser T. IBRNet: learning multi-view image-based rendering. In: Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 4690−4699

[94] Knapitsch A, Park J, Zhou Q Y, Koltun V. Tanks and temples: benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG), 2017, 36(4): 78

[95] Jensen R, Dahl A, Vogiatzis G, Tola E, Aanæs H. Large scale multi-view stereopsis evaluation. In: Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014, 406−413

[96] Barron J T, Mildenhall B, Verbin D, Srinivasan P P, Hedman P. Mip-NeRF 360: unbounded anti-aliased neural radiance fields. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 5470−5479

[97] Sun C, Sun M, Chen H T. Direct voxel grid optimization: super-fast convergence for radiance fields reconstruction. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 5459−5469

[98] Scaman K, Virmaux A. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018, 3839−3848

[99] Fridovich-Keil S, Meanti G, Warburg F R, Recht B, Kanazawa A. K-planes: explicit radiance fields in space, time, and appearance. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 12479−12488

[100] Zhang K, Riegler G, Snavely N, Koltun V. NeRF++: analyzing and improving neural radiance fields. 2020, arXiv preprint arXiv: 2010.07492

[101] Ha D, Dai A, Le Q V. HyperNetworks. 2016, arXiv preprint arXiv: 1609.09106

[102] Huang Y H, He Y, Yuan Y J, Lai Y K, Gao L. StylizedNeRF: consistent 3D scene stylization as stylized NeRF via 2D-3D mutual learning. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 18342−18352

[103] Zhang K, Kolkin N, Bi S, Luan F, Xu Z, Shechtman E, Snavely N. ARF: artistic radiance fields. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 717−733

[104] Fan Z, Jiang Y, Wang P, Gong X, Xu D, Wang Z. Unified implicit neural stylization. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 636−654

[105] Sitzmann V, Martel J N P, Bergman A W, Lindell D B, Wetzstein G. Implicit neural representations with periodic activation functions. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 626

[106] Chen A, Xu Z, Geiger A, Yu J, Su H. TensoRF: tensorial radiance fields. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 333−350

[107] Radl L, Steiner M, Kurz A, Steinberger M. LAENeRF: local appearance editing for neural radiance fields. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 4969−4978

[108] Müller T, Evans A, Schied C, Keller A. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (TOG), 2022, 41(4): 102

[109] Liu S, Zhang X, Zhang Z, Zhang R, Zhu J Y, Russell B. Editing conditional radiance fields. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 5773−5783

[110] Gong B, Wang Y, Han X, Dou Q. RecolorNeRF: layer decomposed radiance fields for efficient color editing of 3D scenes. In: Proceedings of the 31st ACM International Conference on Multimedia. 2023, 8004−8015

[111] Kuang Z, Luan F, Bi S, Shu Z, Wetzstein G, Sunkavalli K. PaletteNeRF: palette-based appearance editing of neural radiance fields. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 20691−20700

[112] Saroha A, Gladkova M, Curreli C, Muhle D, Yenamandra T, Cremers D. Gaussian splatting in style. 2024, arXiv preprint arXiv: 2403.08498

[113] Yu X Y, Yu J X, Zhou L B, Wei Y, Ou L L. InstantStyleGaussian: efficient art style transfer with 3D Gaussian splatting. 2024, arXiv preprint arXiv: 2408.04249

[114] Zhi S, Laidlow T, Leutenegger S, Davison A J. In-place scene labelling and understanding with implicit scene representation. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 15838−15847

[115] Li B, Weinberger K Q, Belongie S J, Koltun V, Ranftl R. Language-driven semantic segmentation. In: Proceedings of the 10th International Conference on Learning Representations. 2022

[116] Caron M, Touvron H, Misra I, Jegou H, Mairal J, Bojanowski P, Joulin A. Emerging properties in self-supervised vision transformers. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 9650−9660

[117] Kerr J, Kim C M, Goldberg K, Kanazawa A, Tancik M. LERF: language embedded radiance fields. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 19729−19739

[118] Goel R, Sirikonda D, Saini S, Narayanan P J. Interactive segmentation of radiance fields. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 4201−4211

[119] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg A C, Lo W Y, Dollár P, Girshick R. Segment anything. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 4015−4026

[120] Lüddecke T, Ecker A. Image segmentation using text and image prompts. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 7086−7096

[121] Liu R, Xiang J, Zhao B, Zhang R, Yu J, Zheng C. Neural impostor: editing neural radiance fields with explicit shape manipulation.

Computer Graphics Forum, 2023, 42(7): e14981

[122] Wang C, He M, Chai M, Chen D, Liao J. Mesh-guided neural implicit field editing. 2023, arXiv preprint arXiv: 2312.02157

[123] Peng Y, Yan Y, Liu S, Cheng Y, Guan S, Pan B, Zhai G, Yang X. CageNeRF: cage-based neural radiance field for generalized 3D deformation and animation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2277

[124] Yang B, Bao C, Zeng J, Bao H, Zhang Y, Cui Z, Zhang G. NeuMesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 597−614

[125] Zhou K, Hong L, Xie E, Yang Y, Li Z, Zhang W. SERF: fine-grained interactive 3D segmentation and editing with radiance fields. 2023, arXiv preprint arXiv: 2312.15856

[126] Guédon A, Lepetit V. Gaussian frosting: editable complex radiance fields with real-time rendering. 2024, arXiv preprint arXiv: 2403.14554

[127] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the 4th Eurographics Symposium on Geometry Processing. 2006

[128] Tang J, Ren J, Zhou H, Liu Z, Zeng G. DreamGaussian: generative Gaussian splatting for efficient 3D content creation. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[129] Xie T, Aigerman N, Belilovsky E, Popa T. Sketch-guided cage-based 3D Gaussian splatting deformation. 2024, arXiv preprint arXiv: 2411.12168

[130] Cao C, Fu Y. Learning a sketch tensor space for image inpainting of man-made scenes. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. 2021, 14509−14518

[131] Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V. Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of 2022 IEEE/CVF Winter Conference on Applications of Computer Vision. 2022, 2149−2159

[132] Liu H K, Shen I C, Chen B Y. NeRF-in: free-form NeRF inpainting with RGB-D priors. 2022, arXiv preprint arXiv: 2206.04901

[133] Weder S, Garcia-Hernando G, Monszpart Á, Pollefeys M, Brostow G, Firman M, Vicente S. Removing objects from neural radiance fields. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 16528−16538

[134] Mirzaei A, Aumentado-Armstrong T, Brubaker M A, Kelly J, Levinshtein A, Derpanis K G, Gilitschenski I. Reference-guided controllable inpainting of neural radiance fields. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 17815−17825

[135] Chen H, Loy C C, Pan X. MVIP-NeRF: multi-view 3D inpainting on NeRF scenes via diffusion prior. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 5344−5353

[136] Huang J, Yu H. Point'n move: interactive scene object manipulation on Gaussian splatting radiance fields. 2023, arXiv preprint arXiv: 2311.16737

[137] Wang Y, Wu Q, Zhang G, Xu D. GScream: learning 3D geometry and feature consistent Gaussian splatting for object removal. 2024, arXiv preprint arXiv: 2404.13679

[138] 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. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 8748−8763

[139] Wang C, Chai M, He M, Chen D, Liao J. CLIP-NeRF: text-and-image driven manipulation of neural radiance fields. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 3835−3844

[140] Wang C, Jiang R, Chai M, He M, Chen D, Liao J. *NeRF-Art*: text-driven neural radiance fields stylization. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(8): 4983−4996

[141] Gordon O, Avrahami O, Lischinski D. Blended-NeRF: zero-shot object generation and blending in existing neural radiance fields. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 2941−2951

[142] Song H, Choi S, Do H, Lee C, Kim T. Blending-NeRF: text-driven localized editing in neural radiance fields. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 14383−14393

[143] Jain A, Mildenhall B, Barron J T, Abbeel P, PooleB. Zero-shot text-guided object generation with dream fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, 867–876

[144] Tancik M, Weber E, Ng E, Li R, Yi B, Wang T, Kristoffersen A, Austin J, Salahi K, Ahuja A, Mcallister D, Kerr J. Nerfstudio: a modular framework for neural radiance field development. In: Proceedings of the ACM SIGGRAPH 2023 Conference. 2023, 72

[145] Mirzaei A, Aumentado-Armstrong T, Brubaker M A, Kelly J, Levinshtein A, Derpanis K G, Gilitschenski I. Watch your steps: local image and scene editing by text instructions. In: Proceedings of the 18th European Conference on Computer Vision. 2025, 111−129

[146] Kamata H, Sakuma Y, Hayakawa A, Ishii M, Narihira T. Instruct 3D-to-3D: text instruction guided 3D-to-3D conversion. 2023, arXiv preprint arXiv: 2303.15780

[147] Mikaeili A, Perel O, Safaee M, Cohen-Or D, Mahdavi-Amiri A. SKED: sketch-guided text-based 3D editing. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 14607−14619

[148] Karnewar A, Ritschel T, Wang O, Mitra N. ReLU fields: the little non-linearity that could. In: Proceedings of the ACM SIGGRAPH 2022 Conference. 2022, 27

[149] Liu X, Xu K, Huang Y, Yi R, Zhu C. MaskEditor: instruct 3D object editing with learned masks. In: Proceedings of the 7th Chinese Conference on Pattern Recognition and Computer Vision. 2025, 285−298

[150] Wang P, Liu L, Liu Y, Theobalt C, Komura T, Wang W. NeuS: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 27171−27183

[151] Ruiz N, Li Y, Jampani V, Pritch Y, Rubinstein M, Aberman K. DreamBooth: fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of 2023 IEEE/CVF Conference on

Computer Vision and Pattern Recognition. 2023, 22500−22510
[152] Munkberg J, Chen W, Hasselgren J, Evans A, Shen T, Müller T, Gao J, Fidler S. Extracting triangular 3D models, materials, and lighting from images. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 8280−8290
[153] Zhou C, Loy C C, Dai B. Extract free dense labels from CLIP. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 696−712
[154] Fu S, Hamilton M, Brandt L, Feldman A, Zhang Z, Freeman W T. FeatUp: a model-agnostic framework for features at any resolution. 2024, arXiv preprint arXiv: 2403.10516
[155] Xu S, Huang Y, Pan J, Ma Z, Chai J. Inversion-free image editing with natural language. 2023, arXiv preprint arXiv: 2312.04965
[156] Zhang Q, Xu Y, Wang C, Lee H Y, Wetzstein G, Zhou B, Yang C. 3DitScene: editing any scene via language-guided disentangled Gaussian splatting. 2024, arXiv preprint arXiv: 2405.18424
[157] Mendiratta M, Pan X, Elgharib M, Teotia K, Mallikarjun B R, Tewari A, Golyanik V, Kortylewski A, Theobalt C. AvatarStudio: text-driven editing of 3D dynamic human head avatars. ACM Transactions on Graphics (ToG), 2023, 42(6): 226
[158] Ouyang H, Wang Q, Xiao Y, Bai Q, Zhang J, Zheng K, Zhou X, Chen Q, Shen Y. CoDeF: content deformation fields for temporally consistent video processing. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 8089−8099
[159] Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman A A, Tzionas D, Black M J. Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, 10975−10985

Chen-Yang ZHU is an associate professor at the College of Computer Science and Technology, National University of Defense Technology (NUDT), China. He received his Bachelor's and Master's degrees from NUDT, in 2011 and 2013 respectively, and completed his PhD at Simon Fraser University, Canada. He is interested in computer graphics, 3D vision, and robotics.

Xin-Yao LIU is an MEng student at the College of Computer Science and Technology, National University of Defense Technology, China. Her primary research interests include 3D vision, text-driven 3D generation, and editing, etc.

Kai XU (Senior Member, IEEE) received his PhD degree in computer science from the National University of Defense Technology (NUDT), China, in 2011. From 2008 to 2010, he worked as a visiting PhD with the GrUVi Laboratory, Simon Fraser University, Canada. He is currently a professor at the College of Computer Science and Technology, NUDT, China. He is also an adjunct professor at Simon Fraser University, Canada. His current research interests include 3D vision and embodied intelligence

Ren-Jiao YI received her Bachelor's degree from the National University of Defense Technology (NUDT), China in 2013 and her PhD from Simon Fraser University, Canada in 2019. She is currently an associate professor at the NUDT, China. She is interested in 3D vision and computer graphics, including inverse rendering, image relighting, and scene reconstruction.