

# NeRF: Neural Radiance Field in 3D Vision: A Comprehensive Review

Kyle (Yilin) Gao, *Member, IEEE*, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, *Member, IEEE*, Jonathan Li, *Fellow, IEEE*

**Abstract**—In March 2020, Neural Radiance Field (NeRF) revolutionized Computer Vision, allowing for implicit, neural network-based scene representation and novel view synthesis. NeRF models have found diverse applications in robotics, urban mapping, autonomous navigation, virtual reality/augmented reality, and more. In August 2023, Gaussian Splatting, a direct competitor to the NeRF-based framework, was proposed, gaining tremendous momentum and overtaking NeRF-based research in terms of interest as the dominant framework for novel view synthesis. We present a comprehensive survey of NeRF papers from the past five years (2020-2025). These include papers from the pre-Gaussian Splatting era, where NeRF dominated the field for novel view synthesis and 3D implicit and hybrid representation neural field learning. We also include works from the post-Gaussian Splatting era where NeRF and implicit/hybrid neural fields found more niche applications.

Our survey is organized into architecture and application-based taxonomies in the pre-Gaussian Splatting era, as well as a categorization of active research areas for NeRF, neural field, and implicit/hybrid neural representation methods. We provide an introduction to the theory of NeRF and its training via differentiable volume rendering. We also present a benchmark comparison of the performance and speed of classical NeRF, implicit and hybrid neural representation, and neural field models, and an overview of key datasets.

**Index Terms**—Neural Radiance Field, NeRF, Computer Vision Survey, Novel View Synthesis, Neural Rendering, Volume Rendering, 3D Reconstruction, Implicit Neural Representation, Neural Fields

## I. INTRODUCTION

NEURAL rendering leverages developments in deep learning for video generation and image synthesis. Implicit neural rendering methods use “hidden” 3D representations (e.g., stored in a neural network as a neural field). Explicit neural rendering methods store 3D representations in explicit data structures. There are also hybrid methods that make use of both implicit and explicit 3D representations. Neural Radiance Fields (NeRF) use differentiable volume rendering to learn a (typically) implicit neural scene representation, using Multi-Layer Perceptrons (MLPs) to store the geometry and lighting of a 3D scene as neural fields. This learned representation can then be used to generate 2D images of the scene under novel, user-specified viewpoints (novel view synthesis). Mildenhall et al. first introduced NeRF at ECCV 2020 [1], and since then, it has achieved state-of-the-art visual quality, produced impressive demonstrations, and inspired many subsequent works. Since 2020, NeRF models and subsequent neural field-based volume rendering models have found applications in photo editing, 3D surface extraction, human avatar modeling, large/city-scale 3D representation and view synthesis, and 3D

object generation.

In 2023, Gaussian Splatting [2], an alternative novel view synthesis framework, overtook NeRF and adjacent methods on many novel view synthesis benchmarks, as well as in applications in 3D vision. As such, much of the research interest has shifted toward Gaussian Splatting. Nonetheless, research in NeRF and NeRF-adjacent neural rendering has persisted since 2023.

NeRF models have important advantages over other classical methods of novel view synthesis and scene representation.

- NeRF models are self-supervised. They can be trained using only multi-view images of a scene. Unlike many other neural representations of 3D scenes, NeRF models require only images and poses to learn a scene and do not require 3D or depth supervision. The poses can be estimated using Structure from Motion (SfM) packages such as COLMAP [3].
- NeRF models are photorealistic. Compared to classical techniques such as [4] [5], as well as earlier novel view synthesis methods such as [6] [7] [8], and neural 3D representation methods [9] [10] [11], the original NeRF model converged to better results in terms of visual quality, with more recent models performing even better.

Compared to Gaussian Splatting-based methods, which have largely overtaken NeRF-based methods in novel view synthesis and adjacent research, NeRF methods have the following disadvantages.

- Gaussian Splatting methods are more photorealistic than NeRF methods, typically converging to representations that generate higher-quality images.
- Gaussian Splatting methods are faster to train. On equal hardware, using the same training images, fully implicit NeRF methods take two to three orders of magnitude longer to converge. Once fully trained, Gaussian Splatting methods also render images faster than implicit NeRF-based methods by orders of magnitude.
- Gaussian Splatting methods use 3D point-based representations and can be easily converted into 3D point clouds, a common data structure for representing 3D scenes. On the other hand, extracting an explicit 3D representation from a typical NeRF method is more difficult.

However, NeRF methods have the following advantages over Gaussian Splatting methods.

- An implicit or hybrid NeRF method has lower storage requirements after training and usually lower memory requirements during training.

- NeRF methods, with neural network-based 3D representations, are better suited for 3D vision pipelines that require or prefer implicit representations.

We organized a survey paper focusing on NeRF methods and NeRF-like neural rendering methods, extending our earlier highly cited preprint on works from 2020 to 2023 with post-Gaussian Splatting (post-GS) methods from 2023 to 2025.

In the post-Gaussian Splatting era, many implicit and hybrid neural representation methods distance themselves from the keyword “NeRF” and instead emphasize the terms neural field and neural representation. We include these methods in the post-Gaussian Splatting chapter if they use NeRF-like differentiable volume rendering in conjunction with a neural field-based representation. We do not make an explicit distinction pre-Gaussian Splatting, since many of these adjacent methods’ neural field-based volume rendering techniques were referred to as NeRF or NeRF-like pre-Gaussian Splatting, even if they used other types of neural fields such as neural signed distance functions or occupancy fields. Post-Gaussian Splatting, we use the same naming conventions as the authors of the respective papers.

The rest of this manuscript is organized as follows.

- Section II introduces existing NeRF survey preprints (II-A), explains the theory behind NeRF volume rendering (II-B), and introduces commonly used datasets (II-C) and quality assessment metrics (II-D).
- Sections III and IV form the first part of the main body of the survey. Section III introduces influential NeRF publications from before the Gaussian Splatting era and presents the taxonomy we created to organize these works. Its subsections detail the different families of NeRF and adjacent methods’ innovations proposed from 2020 to 2023.
- Section IV focuses on the applications of NeRF and adjacent models to various computer vision tasks.
- Section V forms the second part of the main body of the survey. This section introduces progress in neural rendering research post-Gaussian Splatting (2023 onward). We specifically focus on post-Gaussian Splatting applications of implicit and hybrid neural field methods that use NeRF-like volume rendering.
- Sections VI and VII present discussions on the status of this research field post-Gaussian Splatting and provide concluding remarks.

## II. BACKGROUND

### A. Existing Surveys on NeRF with Comparable Scope

In December 2020, Dellaert and Yen-Chen published a concise preprint NeRF survey [12] including approximately 50 NeRF publications/preprints, many of which were eventually published in top-tier computer vision conferences. We took inspiration from this preprint survey and used it as a starting point for our own survey. However, the work is only eight pages long and does not include detailed descriptions. Moreover, it only includes NeRF papers from 2020 and early 2021 preprints and omits several influential NeRF papers published in the latter half of 2021 and beyond.

In November 2021, Xie et al. [13] published a preprint survey titled Neural Fields in Visual Computing and Beyond, which was later revised and published as a state-of-the-art report in Eurographics 2022, and also presented as a tutorial at CVPR 2022. The survey is broad and includes a detailed technical introduction to the use of neural fields in computer vision, as well as applications in visual computing, with a strong focus on Neural Radiance Fields. Compared to this work, we restrict our review to NeRF papers only and include more recent work. We also present more detail on a paper-by-paper basis.

In December 2021, Zhang et al. [14] published a preprint survey on multimodal image synthesis and editing, in which they dedicated a paragraph to NeRF. They mostly focused on multimodal NeRFs such as [15], [16], only citing these two and the original NeRF paper [1] in the main text, with four additional papers [17] [18] [19] [20] included in their supplementary materials.

In May 2022, Tewari et al. [21] published a state-of-the-art report on advances in neural rendering with a focus on NeRF models. It remains the most comprehensive neural rendering survey-style report to date, covering many influential NeRF papers, as well as numerous other neural rendering papers. Our survey differs from this report in that our scope is fully focused on NeRF papers, providing detailed paper-by-paper summaries of selected works. We also present a NeRF innovation technique taxonomy tree and a NeRF application classification tree. We include more recent works and introduce, in detail, the common datasets and evaluation metrics used by NeRF practitioners.

These survey papers were concurrent with our initial manuscript, completed in 2022, and are most similar in scope. Since the Gaussian Splatting era, numerous surveys focusing on specific applications of neural fields have been completed. However, these surveys differ from ours in scope.

### B. Neural Radiance Field (NeRF) Theory

Neural Radiance Fields were first proposed by Mildenhall et al. [1] in 2020 for novel view synthesis. NeRFs achieved highly photo-realistic view synthesis of complex scenes and attracted much attention in the field. In its basic form, a NeRF model represents three-dimensional scenes as a radiance field approximated by a neural network. The radiance field describes color and volume density for every point and for every viewing direction in the scene. This is written as:

$$F(\mathbf{x}, \theta, \phi) \rightarrow (\mathbf{c}, \sigma), \quad (1)$$

where  $\mathbf{x} = (x, y, z)$  is the in-scene coordinate,  $(\theta, \phi)$  represent the azimuthal and polar viewing angles,  $\mathbf{c} = (r, g, b)$  represents color, and  $\sigma$  represents the volume density. This 5D function is approximated by one or more Multi-Layer Perceptron (MLP), sometimes denoted as  $F_\Theta$ . The two viewing angles  $(\theta, \phi)$  are often represented by  $\mathbf{d} = (d_x, d_y, d_z)$ , a 3D Cartesian unit vector. This neural network representation is constrained to be multi-view consistent by restricting the prediction of  $\sigma$ , the volume density (i.e., the content of the scene) to be independent of viewing direction, whereas the

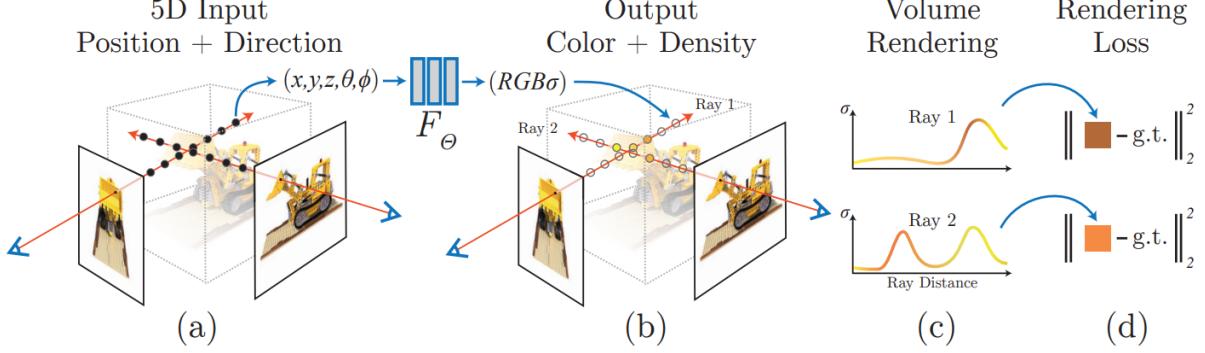


Fig. 1. The NeRF volume rendering and training process (image sourced from [1]). (a) illustrates the selection of sampling points for individual pixels in a to-be-synthesized image. (b) illustrates the generation of densities and colors at the sampling points using NeRF MLP(s). (c) and (d) illustrate the generation of individual pixel color(s) using in-scene colors and densities along the associated camera ray(s) via volume rendering, and the comparison to ground truth pixel color(s), respectively.

color  $\mathbf{c}$  is allowed to depend on both viewing direction and in-scene coordinate. In the baseline NeRF model, this is implemented by designing the MLP to be in two stages. The first stage takes as input  $\mathbf{x}$  and outputs  $\sigma$  and a high-dimensional feature vector (256 in the original paper). In the second stage, the feature vector is then concatenated with the viewing direction  $\mathbf{d}$ , and passed to an additional MLP, which outputs  $\mathbf{c}$ . We note that Mildenhall et al. [1] consider the  $\sigma$  MLP and the  $\mathbf{c}$  MLP to be two branches of the same neural network, but many subsequent authors consider them to be two separate MLP networks.

Broadly speaking, novel view synthesis using a trained NeRF model is as follows.

- For each pixel in the image being synthesized, send camera rays through the scene and generate a set of sampling points (see (a) in Fig. 1).
- For each sampling point, use the viewing direction and sampling location to compute local color and density using the NeRF MLP(s) (as shown in (b) in Fig. 1).
- Use volume rendering to produce the image from these colors and densities (see (c) in Fig. 1).

Given the volume density and color functions of the scene being rendered, volume rendering [22] is used to obtain the color  $C(\mathbf{r})$  of any camera ray  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ , with camera position  $\mathbf{o}$  and viewing direction  $\mathbf{d}$  using

$$C(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{d}) \cdot dt, \quad (2)$$

where  $\sigma(\mathbf{r}(t))$  and  $\mathbf{c}(\mathbf{r}(t), \mathbf{d})$  represent the volume density and color at point  $\mathbf{r}(t)$  along the camera ray with viewing direction  $\mathbf{d}$ , and  $dt$  represents the differential distance traveled by the ray at each integration step.

$T(t)$  is the accumulated transmittance, representing the probability that the ray travels from  $t_1$  to  $t$  without being intercepted, given by

$$T(t) = \exp\left(-\int_{t_1}^t \sigma(\mathbf{r}(u)) \cdot du\right). \quad (3)$$

Novel views are rendered by tracing the camera rays  $C(\mathbf{r})$  through each pixel of the to-be-synthesized image. This integral can be computed numerically. The original implementation [1] and most subsequent methods used a non-deterministic stratified sampling approach, where the ray was divided into  $N$  equally spaced bins, and a sample was uniformly drawn from each bin. Then, equation (2) can be approximated as

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N \alpha_i T_i \mathbf{c}_i, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right). \quad (4)$$

$\delta_i$  is the distance from sample  $i$  to sample  $i+1$ .  $(\sigma_i, \mathbf{c}_i)$  are the density and color evaluated along the sample point  $i$  given the ray, as computed by the NeRF MLP(s).  $\alpha_i$  the transparency-opacity from alpha compositing at sample point  $i$ , is given by

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i). \quad (5)$$

An expected depth can be calculated for the ray using the accumulated transmittance as

$$d(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot t \cdot dt. \quad (6)$$

This can be approximated analogously to equation (4) approximating equation (2) and (3)

$$\hat{d}(\mathbf{r}) = \sum_{i=1}^N \alpha_i t_i T_i. \quad (7)$$

Certain depth regularization [23] [24] [25] [26] methods use the expected depth to restrict densities to delta-like functions at scene surfaces, or to enforce depth smoothness.

For each pixel, a square error photometric loss is used to optimize the MLP parameters. Over the entire image, this is given by

$$L = \sum_{r \in R} \|\hat{C}(\mathbf{r}) - C_{gt}(\mathbf{r})\|_2^2, \quad (8)$$

where  $C_{gt}(\mathbf{r})$  is the ground truth color of the training image's pixel associated to  $\mathbf{r}$ , and  $R$  is the batch of rays associated to the to-be-synthesized image.

NeRF models often employ positional encoding, which was shown by Mildenhall et al. [1] to greatly improve fine detail reconstruction in the rendered views. This was also shown in more detail, with corroborating theory using Neural Tangent Kernels in [27]. (We note a recent work [28] instead tests novel activation functions to address this issue). In the original implementation, the following positional encoding  $\gamma$  was applied to each component of the scene coordinate  $\mathbf{x}$  (normalized to  $[-1,1]$ ) and viewing direction unit vector  $\mathbf{d}$

$$\gamma(v) = (\sin(2^0\pi v), \cos(2^0\pi v), \sin(2^1\pi v), \cos(2^1\pi v), \\ \dots, \sin(2^{N-1}\pi v), \cos(2^{N-1}\pi v)), \quad (9)$$

where  $N$  is a user-determined encoding dimensionality parameter, set to  $N = 10$  for  $\mathbf{x}$  and  $N = 4$  for  $\mathbf{d}$  in the original paper. However, modern researchers have experimented and achieved great results with alternate forms of positional encoding including trainable parametric, integral, and hierarchical variants (see section III).

**Naming conventions:** There are three types of 3D representation: implicit, hybrid, and explicit. In the baseline NeRF, the density and color fields are fully represented by MLPs; this is considered an implicit scene representation. The neural color and density fields together are called the neural radiance field. Methods with hybrid and explicit scene representations are introduced in Sections III-B and III-G1, respectively. Other types of neural fields, such as neural occupancy fields, neural signed distance function fields, neural deformation fields, and neural semantic fields, are introduced throughout this survey.

### C. Datasets

NeRF models are typically trained per-scene and require relatively dense images with relatively varied poses. While some NeRF models have been designed to be trained from sparse input views or unposed images, camera poses can often be extracted using existing Structure-from-Motion libraries such as COLMAP [3].

The original NeRF paper [1] presented a synthetic dataset created using Blender (referred to as the **Realistic Synthetic 360 Degrees dataset** in [1]), and often referred to as the **NeRF Synthetic dataset or the NeRF dataset** in subsequent works. The virtual cameras have the same focal length and are placed at the same distance from the object. The dataset is composed of eight scenes with eight different objects. For six of these, viewpoints are sampled from the upper hemisphere; for the other two, viewpoints are sampled from the entire sphere. The objects are "hotdog", "materials", "ficus", "lego", "mic", "drums", "chair", and "ship". A comparative visualization of the results from the NeRF paper is shown in Figure 2. The images are rendered at  $800 \times 800$  pixels, with 100 views for training and 200 views for testing. This is often the first dataset considered by NeRF researchers, as the scenes are bounded, focused on a single object, and benchmarks of commonly used models on this scene are easily found.

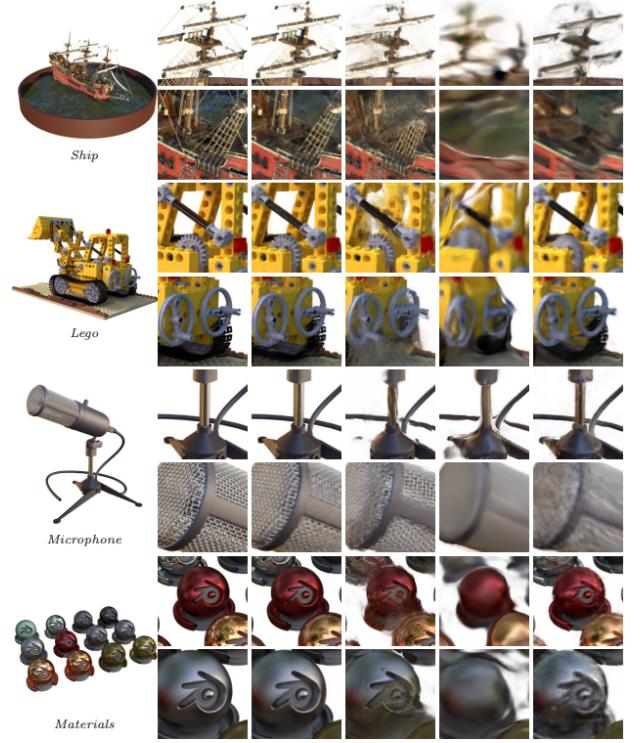


Fig. 2. Original NeRF [1] results on four scenes of the NeRF Synthetic Dataset. Left to right: Ground truth, NeRF [1], LLFF [6], SRN [7], NV [8].

The **LLFF dataset** [6] consists of 24 real-life scenes captured from handheld cellphone cameras. The views are forward-facing towards the central object. Each scene consists of 20-30 images. The COLMAP [3] package was used to compute the poses of the images. The usage of this dataset is comparable to that of the Realistic Synthetic dataset from [1]; the scenes are not too challenging for any particular NeRF model, and the dataset is well benchmarked, offering readily available comparisons to known methods.

The **DTU dataset** [29] is a multi-view stereo dataset captured using a 6-axis industrial robot mounted with both a camera and a structured light scanner. The robot provides precise camera positioning. Both the camera intrinsics and poses are carefully calibrated using the MATLAB calibration toolbox [30]. The light scanner provides reference dense point clouds, which serve as ground truth 3D geometry. Nonetheless, due to self-occlusion, the scans of certain areas in some scenes are incomplete. The original paper's dataset consists of 80 scenes, each containing 49 views sampled on a sphere of radius 50 cm around the central object. For 21 of these scenes, an additional 15 camera positions are sampled at a radius of 65 cm, for a total of 64 views. The entire dataset includes 44 additional scenes that have been rotated and scanned four times at 90-degree intervals. The illumination of scenes is varied using 16 LEDs, with seven different lighting conditions. The image resolution is  $1600 \times 1200$ . This dataset differs from the previous ones by its higher resolution and carefully calibrated camera motion and poses.

The **ScanNet dataset** [31] is a large-scale real-life RGB-D multi-modal dataset containing more than 2.5 million views

of indoor scenes, with annotated camera poses, reference 3D surfaces, semantic labels, and CAD models. The depth frames are captured at  $640 \times 480$  pixels, and the RGB images are captured at  $1296 \times 968$  pixels. The scans were performed using RGB-D sensors attached to handheld devices such as iPhones and iPads. The poses were estimated using BundleFusion [32] and geometric alignment of the resulting mesh. This dataset's rich semantic labels are useful for models that utilize semantic information, such as for scene editing, scene segmentation, and semantic view synthesis.

The **ShapeNet dataset** [33] is a simple large-scale synthetic 3D dataset consisting of 3D CAD models classified into 3,135 classes. The most commonly used subset is the 12 common object categories. This dataset is sometimes used when object-based semantic labels are an important part of a particular NeRF model. From ShapeNet CAD models, software such as Blender is often used to render training views with known poses.

*1) Building-scale Dataset:* The **Tanks and Temples dataset** [34] is a from-video 3D reconstruction dataset. It consists of 14 scenes, including individual objects such as "Tank" and "Train," and large-scale indoor scenes such as "Auditorium" and "Museum." Ground truth 3D data was captured using a high-quality industrial laser scanner. The ground truth point cloud was used to estimate camera poses using least squares optimization of correspondence points. This dataset contains large-scale scenes, some of which are outdoors, and poses a challenge for certain NeRF models. The outdoor scenes are suited for models wishing to handle unbounded backgrounds. Its ground truth point cloud can also be used for certain data fusion methods or to test depth reconstruction.

**Matterport-3D dataset** [35] is a real-life dataset consisting of 10,800 panoramic views from 194,400 RGB-D globally registered images of 90 building-scale scenes. Depth, semantic, and instance annotations are available. Color and depth images are provided at  $1280 \times 1024$  resolution for 18 viewpoints per panoramic picture. Each of the 90 buildings consists of an average of  $2,437 \text{ m}^2$  of surface area. A total of 50,811 object instance labels were provided, which were mapped into 40 object categories.

The **Replica dataset** [36] is a real indoor dataset consisting of 18 scenes and 35 indoor rooms captured using a custom-built RGB-D rig with an IR projector. Certain 3D features were manually fixed, such as fine-scale mesh details like small holes, and reflective surfaces were manually assigned. Semantic annotations (88 classes) were performed in two steps: once in 2D and once in 3D. Both class-based and instance-based semantic labels are available.

*2) Large-Scale Urban Datasets:* Popular autonomous driving benchmark datasets have multiple data modalities such as images, depth maps, LiDAR point clouds, poses, and semantic maps, which are potentially suitable for certain NeRF models targeting urban scenes. Recently, NeRF models such as those proposed in [51] and [52] have made effective use of these datasets.

**KITTI** [53] [54] [55] [56] is a well-known city-scale 2D-3D computer vision dataset suite created for training and

benchmarking vision algorithms for autonomous driving. The suite contains labeled datasets for stereo 3D semantic and 2D semantic segmentation, flow, odometry, 2D-3D object detection, tracking, lane detection, and depth prediction/completion. These were created from raw LiDAR and video data captured in Karlsruhe, Germany, using a car-based setup with GPS and inertial measurement unit data, recorded with a Velodyne LiDAR scanner and multiple cameras. The depth prediction/completion dataset is by far the largest, containing over 93 thousand depth maps with corresponding RGB images and raw LiDAR scans. This dataset, however, poses a challenge to NeRF training due to its relatively sparse camera coverage compared to NeRF-specific datasets, requiring sparse-view considerations when designing the model. The recent KITTI-360 [57] extension to the suite includes a novel view synthesis benchmark which tabulates many NeRF models.

The **Waymo Open Dataset** [58] is a recently published alternative to KITTI. Covering  $72\text{km}^2$ , the dataset is created from point cloud and video data captured from five LiDAR sensors and five high-resolution pinhole cameras in a car-based setup, captured in the San Francisco Bay, Mountain View, and Phoenix in the United States. In addition to matched point cloud and video data, the dataset also contains annotated labels for 2D and 3D object detection and tracking. The dataset contains 1150 separate scenes (as opposed to KITTI's 22 scenes) and has higher LiDAR and camera resolution. Its object annotations are also more extensive by two orders of magnitude (80K vs 12M).

*3) Human Avatar/Face Dataset:* The **Nerfies** [59] and **HyperNerf** [60] datasets are single-camera datasets focused on human faces, with motion generated by moving two cameras attached to a pole relative to the subject. The former contains five human subjects staying still, as well as four more scenes with moving human subjects, a dog, and two moving objects. The latter focuses on topological changes and includes scenes such as a human subject opening and closing their eyes and mouth, peeling a banana, 3D printing a chicken toy, and a broom deforming.

The **ZJU-MOCap LightStage dataset** [61] is a multi-view (20+ cameras) motion capture dataset consisting of 9 dynamic human sequences consisting of exercise-like motions. The videos were captured using 21 synchronized cameras and have a sequence length between 60 to 300 frames.

The **NeuMan dataset** [62] consists of 6 videos, each 10 to 20 seconds long, captured by a mobile phone camera following a walking human subject performing additional simple actions such as twirling or waving.

The **CMU Panoptic dataset** [63] is a large multi-view, multi-subject dataset consisting of groups of people engaged in social interaction. The dataset contains 65 sequences with 1.5 million labeled skeletons. The sensor system consists of 480 VGA views ( $640 \times 480$ ), over 30 HD ( $1920 \times 1080$ ) views, and 10 RGB-D sensors. Scenes are labeled with individual subject and social group semantics, 3D body poses, 3D facial landmarks, and transcripts with speaker IDs.

#### D. Quality Assessment Metrics

Novel view synthesis via NeRF in the standard setting use visual quality assessment metrics for benchmarks. These metrics attempt to assess the quality of individual images either with (full-reference) or without (no-reference) ground truth images. Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [64], Learned Perceptual Image Patch Similarity (LPIPS) [65] are by far the most commonly used in NeRF literature.

$PSNR\uparrow$  is a no-reference quality assessment metric,

$$PSNR(I) = 10 \cdot \log_{10} \left( \frac{MAX(I)^2}{MSE(I)} \right), \quad (10)$$

where  $MAX(I)$  is the maximum possible pixel value in the image (255 for 8bit integer), and  $MSE(I)$  is the pixel-wise mean squared error calculated over all color channels. PSNR is also commonly used in signal processing and is well understood.

$SSIM\uparrow$  [64] is a full-reference quality assessment metric. For a single patch, this is given by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (11)$$

where  $C_i = (K_i L)^2$ ,  $L$  is the dynamic range of the pixels (255 for 8bit integer), and  $K_1 = 0.01$ ,  $K_2 = 0.03$  are constants chosen by the original authors. We note that there is a more general form of SSIM given by (12) in the original paper. The local statistics  $\mu$ 's,  $\sigma$ 's are calculated within a  $11 \times 11$  circular symmetric Gaussian weighted window, with weights  $w_i$  having a standard deviation of 1.5 and normalized to 1.

$LPIPS\downarrow$  [65] is a full reference quality assessment metric which uses learned convolutional features. The score is given

by a weighted pixel-wise MSE of feature maps over multiple layers.

$$LPIPS(x, y) = \sum_l^L \frac{1}{H_l W_l} \sum_{h,w}^{H_l, W_l} ||w_l \odot (x_{hw}^l - y_{hw}^l)||_2^2, \quad (12)$$

where  $x_{hw}^l, y_{hw}^l$  are the reference and assessed images' feature at pixel width  $w$ , pixel height  $h$ , and layer  $l$ .  $H_l$  and  $W_l$  are the feature maps height and width at the corresponding layer. The original LPIPS paper used SqueezeNet [66], VGG [67] and AlexNet [68] as feature extraction backbone.

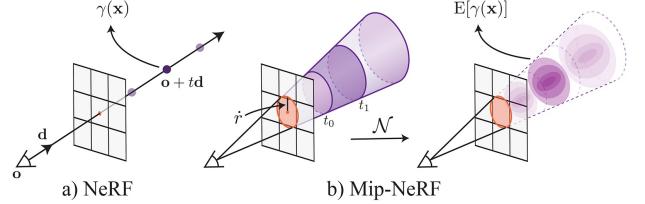


Fig. 3. Diagram of the Integrated Positional Encoding (IPE) of mip-NeRF (Figure 1 in [47]). a) Standard ray-based point sampled of NeRF; b) Cone-sampling of mip-NeRF using IPE, approximating conic frustums with multivariate Gaussian distributions.

#### III. NEURAL RADIANCE FIELD (NERF) PRE GAUSSIAN SPLATTING

In this section, we present select pre-Gaussian Splatting NeRF and adjacent papers organized in a method-based taxonomy, with a separate section reserved for application-based classification. The (arXiv) preprint first draft dates are used to sequence the publications, while conference/journal publication dates can be found in the bibliography. We included a NeRF synthetic dataset [1] benchmark in Table I comparing the most influential (pure) novel-view-synthesis-focused NeRF and adjacent works of that era.

TABLE I  
COMPARISON OF SELECT NERF MODELS ON THE SYNTHETIC NERF DATASET [1]

Method	Representation	PSNR $\uparrow$ (dB)	SSIM $\uparrow$	LPIPS $\downarrow$	Training Iteration	Training Time <sup>1</sup>	Inference Speed <sup>2</sup>
Baseline NeRF (2020) [1]	Implicit	31.01	0.947	0.081	100-300k	>12h	1
Speed Improvement							
JaxNeRF (2020) [37]	Implicit	31.65	0.952	0.051	250k	>12h	~1.3
NSVF (2020) [38]	Hybrid (Learned)	31.74	0.953	0.047	100-150k	-	~10
SNeRG (2021) [39]	Explicit (Baked)	30.38	0.950	0.050	250k	>12h	~9000
PlenOctree (2021) [40]	Explicit (Baked)	31.71	0.958	0.053	2000k	>12h	~3000
FastNeRF (2021) [41]	Explicit (Baked)	29.97	0.941	0.053	300k	>12h	~4000
KiloNeRF (2021) [42]	Explicit (Baked)	31.00	0.95	0.03	600k+150k+1000k	>12h	~2000
Instant-NGP (2022) [43]	Hybrid (Learned)	33.18	-	-	256k	~5m	"orders of magnitude"
Plenoxels (2021) [44]	Explicit (Learned)	31.71	0.958	0.049	128k	~12m	-
DVGO (2021) [45]	Explicit (Learned)	31.95-32.80	0.957-0.961	0.053-0.27	128k	~20m	45x
TensoRF (2022) [46]	Explicit/Hybrid (Learned)	31.56-33.14	0.949-0.963	-	15K-30K	8-25m	~100
Quality Improvement							
mip-NeRF (2021) [47]	Implicit	33.09	0.961	0.043	1000k	~3h	~1
ref-NeRF (2021) [48]	Implicit	35.96	0.967	0.058	250k	-	~1
Sparse View/Few Shots							
MVSNeRF (2021) [49]	Hybrid + Pretrained	27.07	0.931	0.163	10k (*3 views)	~15m*	~1
DietNeRF (2021) [50]	Implicit + Pretrained	23.15	0.866	0.109	200k (*8 views)	-	~1
DS-NeRF (2021) [24]	Implicit	24.9	0.72	0.34	150-200k (*10 views)	-	~1

Key speed, sparse-view and quality improvement models focused on small scale scenes trained on the Synthetic NeRF dataset were selected for comparison.

<sup>1</sup>Training speed are given as in the respective original papers. These are to be taken with "a grain of salt", as hardware differences and hyper-parameters such as image/voxel resolution greatly affect training time.

<sup>2</sup>Inference speeds are given as speedup factor over the baseline NeRF.

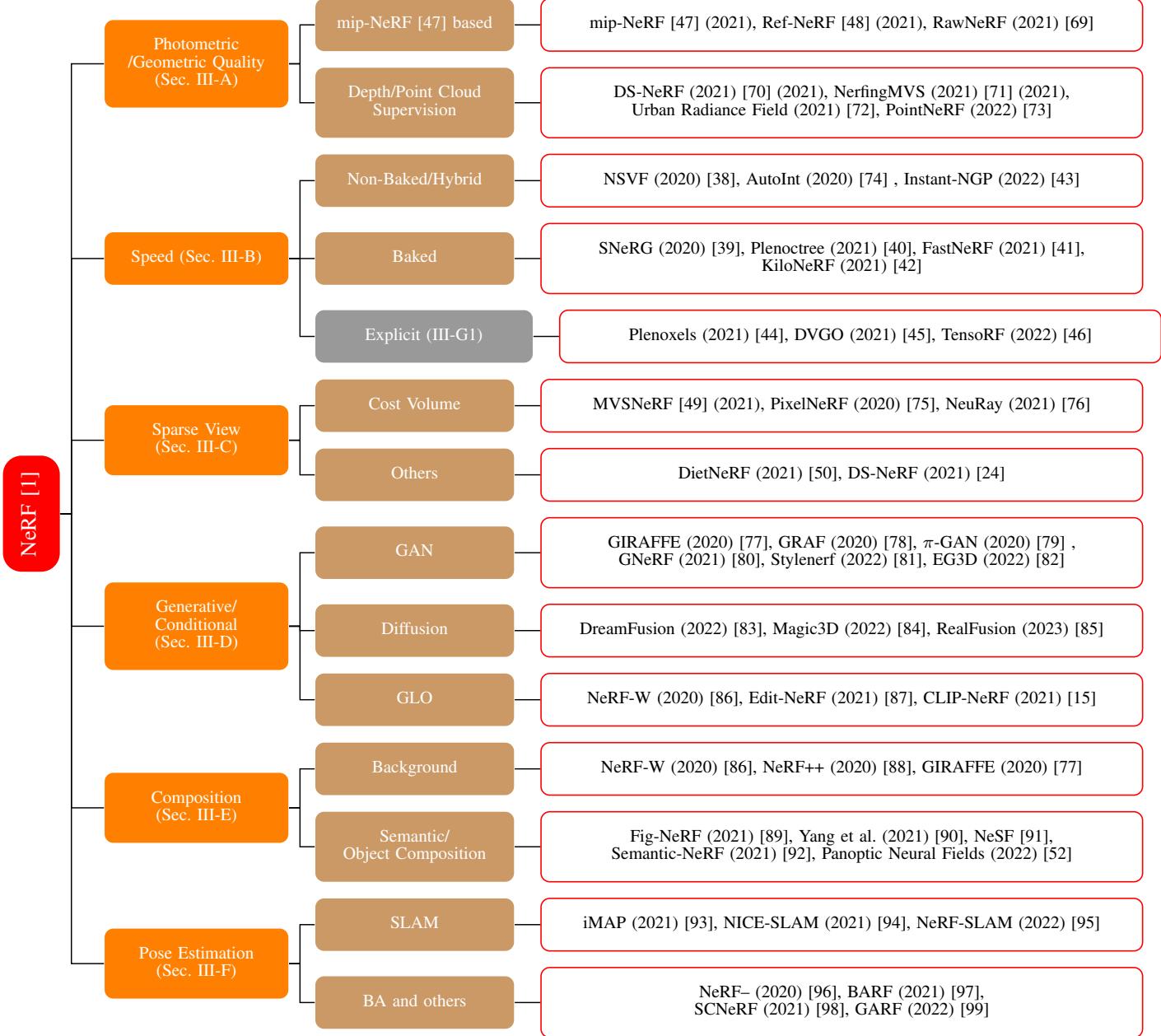


Fig. 4. Taxonomy of selected key NeRF innovation papers. The papers are selected using a combination of citations and GitHub star rating. We note that the MLP-less speed-based models are not strictly speaking NeRF models. Nonetheless, we decided to include them in this taxonomy tree due to their recent popularity and their similarity to speed based NeRF models.

#### A. Improvements in the Quality of Synthesized Views and Learned Geometry

Image quality is the predominant benchmark for view synthesis, and many subsequent pure NeRF research models have focused on improving view synthesis quality. In this section, we highlight important models that aim to enhance the photometric and geometric aspects of NeRF view synthesis and 3D scene representation.

1) *Better View Synthesis: Mip-NeRF* (March 2021) [47] approximated cone tracing instead of using the ray tracing of standard NeRF [1] (March 2020) volume rendering. This was achieved by introducing the Integrated Positional Encoding (IPE) (Fig. 3). Schematically, to generate an individual pixel,

a cone was cast from the camera center along the viewing direction through the center of the pixel. The cone was approximated by a multivariate Gaussian whose mean vector and covariance matrix were derived to match the appropriate geometry (see Appendix A in [47]), resulting in the Integrated Positional Encoding. This is given by

$$\begin{aligned} \gamma(\mu, \Sigma) &= \mathbb{E}_{x \sim \mathcal{N}}(\mu, \Sigma)[\gamma(x)] \\ &= \left[ \sin(\mu_\gamma) \odot \exp(-(1/2)\text{diag}(\Sigma_\gamma)) \right] \\ &\quad \left[ \cos(\mu_\gamma) \odot \exp(-(1/2)\text{diag}(\Sigma_\gamma)) \right], \end{aligned} \quad (13)$$

where  $\mu_\gamma, \Sigma_\gamma$  are the means and variances of the multivariate Gaussian lifted onto the positional encoding basis with  $N$  levels. The resulting Mip-NeRF model was inherently multi-scale

and performed anti-aliasing automatically. It outperformed the baseline NeRF [1], especially at lower resolutions. **Mip-NeRF 360** [100] is a highly influential extension of Mip-NeRF to unbounded scenes. The key technical improvements include a proposal MLP supervised by the NeRF MLP rather than directly by images. This proposal MLP predicts only volumetric density, which guides the sampling intervals. Additionally, a novel scene parameterization was specifically designed for the Gaussians in Mip-NeRF. Finally, a new regularization method was introduced to prevent floating geometric artifacts and background collapse.

**Ref-NeRF** (December 2021) [48] was built on mip-NeRF and was designed to better model reflective surfaces. Ref-NeRF parameterized NeRF radiance based on the reflection of the viewing direction about the local normal vector. They modified the density MLP into a directionless MLP, which not only outputs density and the input feature vector of the directional MLP but also diffuse color, specular color, roughness, and surface normal. Ref-NeRF performed particularly well on reflective surfaces and is capable of accurately modeling specular reflections and highlights (Fig. 5).

**Ray Prior NeRF (RapNeRF)** (May 2022) [101] proposed a NeRF model tailored for view extrapolation, contrasting with standard NeRFs that excel at interpolation. RapNeRF introduced Random Ray Casting (RRC), where for a training ray hitting a surface point  $\mathbf{v} = \mathbf{o} + t_z \mathbf{d}$ , a backward ray is cast from  $\mathbf{v}$  toward a new origin  $\mathbf{o}'$  with uniformly sampled angular perturbations. Additionally, RapNeRF employed a Ray Atlas (RA) by extracting a rough 3D mesh from a pretrained NeRF and mapping training ray directions onto its vertices. Training began with a baseline NeRF to recover the rough mesh. Subsequently, RRC and RA augmented training rays with a predetermined probability. Evaluations on the Synthetic NeRF dataset [1] and the MobileObject dataset demonstrated that these augmentations improve view synthesis quality and are adaptable to other NeRF frameworks.

2) *Depth Supervision and Point Cloud*: By using the supervision of expected depth (6) with point clouds acquired from LiDAR or SfM, these models converge faster, converge to higher final quality, and require fewer training views than the baseline NeRF model. Many of these models were also built for or as few-shot/parsnip view NeRF.

Deng et al. [24] (July 2021) used depth supervision from point clouds with a method named **Depth-Supervised NeRF (DS-NeRF)**. In addition to color supervision via volume rendering and photometric loss, DS-NeRF also performs depth supervision using sparse point clouds extracted from the training images using COLMAP [3]. Depth is modeled as a normal distribution around the depth recorded by the sparse point cloud. A KL divergence term is added to minimize the divergence between the ray's distribution and this noisy depth distribution (see [24] for details).

Concurrent to DS-NeRF is a work by **Roessle et al.** [70] (April 2021). In this work, the authors used COLMAP to extract a sparse point cloud, which was processed by a Depth Completion Network [102] to produce depth and uncertainty maps. In addition to the standard volumetric loss, the authors introduced a depth loss based on predicted depth and uncer-

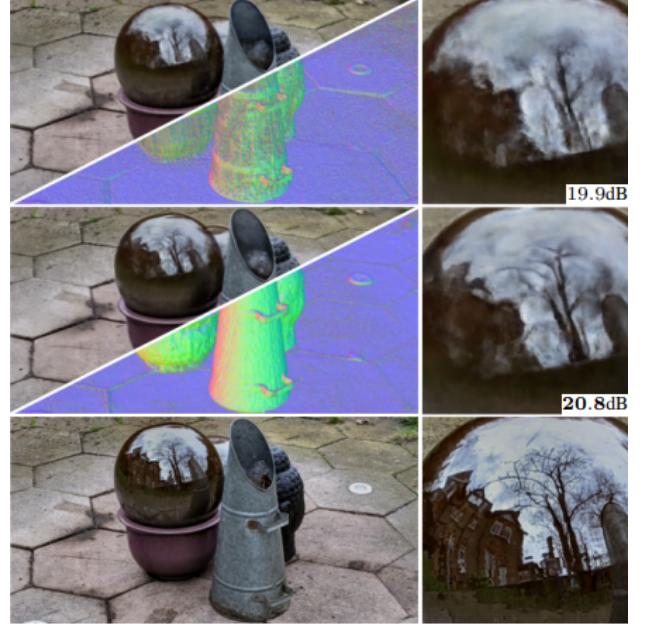


Fig. 5. Ref-NeRF results in the "garden spheres" scene of [39], showing its high performance on reflective scenes and its capability to recover accurate normal vectors of reflective surfaces. (Figure 8 in [48]). Top: mip-NeRF [47]; middle: ref-NeRF [48]; bottom: ground truth.

tainty. The model was trained on RGB-D data from ScanNet [31] and Matterport3D [35] with Gaussian noise added to the depth.

**NerfingMVS** (September 2021) [71] used multi-view images focusing on depth reconstruction. In NerfingMVS, COLMAP was used to extract sparse depth priors as a point cloud. This was then fed into a pretrained monocular depth network [103] fine-tuned on the scene to extract a depth map prior. The depth map prior supervised volume sampling by allowing sampling points only at appropriate depths. During volume rendering, the ray was divided into  $N$  equal bins, with ray bounds clamped using the depth priors.

**PointNeRF** (January 2022) [73] used feature point clouds as an intermediate step before volume rendering. A pretrained 3D CNN [104] generated depth and surface probability  $\gamma$  from a cost volume created from training views, producing a dense point cloud. A pretrained 2D CNN [67] extracted image features from training views. These features populated the point cloud, assigning each point  $p_i$  a surface probability  $\gamma_i$ . Given the input position and view direction, a PointNet [105]-like network regressed local density and color, which were then used for volume rendering. Using point cloud features also allowed the model to skip empty spaces, resulting in a speed-up by a factor of 3 over baseline NeRF.

3) *Other Geometry Improvement*: **SNeS** (June 2022) [106] improved geometry by learning probable symmetries for partly symmetric and partly hidden in-scene objects through soft symmetry constraints on geometry and material properties.

**S<sup>3</sup>-NeRF** (October 2022) [107] used shadow and shading cues to infer scene geometry and enabled single-image NeRF training, with a focus on geometry recovery. S<sup>3</sup>-NeRF employed a UNISURF-based occupancy field 3D representa-

tion instead of density, a modified physics-based rendering equation, and occupancy-based shadow calculation as key implementation differences. The method achieved excellent depth map and surface normal reconstruction from single images on both synthetic and real-world datasets.

### B. Improvements to Training and Inference Speed

In the original implementation by Mildenhall et al. [1], hierarchical rendering was employed to improve computational efficiency. Two networks represented the scene: a coarse network and a fine network. The coarse network’s output guided sampling point selection for the fine network, preventing dense sampling at fine scales. In subsequent works over the next two years, most efforts to accelerate NeRF training and inference fall broadly into two categories.

- 1) Models in the first category train, precompute, and store NeRF MLP evaluations into more accessible data structures. This improves inference speed significantly but does not affect training time. We refer to these as baked models.
- 2) The second category comprises non-baked models, which include various innovations. A common approach is to learn separate scene features from MLP parameters via a hybrid representation. This enables smaller MLPs, improving both training and inference speed at the cost of increased memory. Pushing this further, some methods (see Sec. III-G1) omit neural networks entirely and use purely explicit scene representations. While not strictly NeRF models, we include them here due to their relevance and similarity to NeRF.

Other techniques include ray termination, which stops sampling when accumulated transmittance nears zero, space skipping, and hierarchical sampling using coarse and fine MLPs as in the original NeRF paper. These methods are often combined with per-paper innovations to further improve training and inference speed.

Hybrid and explicit scene representation methods are closely related to baked methods, since scene features are directly optimized within accessible data structures. However, the baked versus non-baked distinction was popular between 2020 and 2022; as such, we organize this section following that convention.

1) *Baked*: A model by Hedman et al. [39] (July 2021) stored a precomputed NeRF on a sparse voxel grid. The method, called **Sparse Neural Voxel Grid (SNeRG)**, stored precomputed diffuse color, density, and feature vectors on the sparse voxel grid in a process sometimes referred to as “baking.” During evaluation, an MLP produced specular color, which, combined with alpha compositing of the specular color along the ray, yielded the final pixel color. The method was approximately 3000 times faster than the original NeRF implementation, with speed comparable to PlenOctree.

Concurrently, the **PlenOctree** [40] approach by Yu et al. (March 2021) achieved inference times approximately 3000 times faster than the original NeRF implementation. The authors trained a spherical harmonic NeRF (NeRF-SH), which predicted spherical harmonic coefficients of the color function

instead of directly predicting color values. They constructed an octree of precomputed spherical harmonic (SH) coefficients derived from the MLP’s colors. During octree construction, the scene was voxelized, and voxels with low transmissivity were eliminated. This procedure can also be applied to standard NeRF models by performing Monte Carlo estimations of the spherical harmonic components. PlenOctrees could be further optimized using initial training images via a fast fine-tuning procedure relative to NeRF training. Notably, the Gaussian Splatting implementation of spherical harmonics color is directly adapted from PlenOctree.

In **FastNeRF** (March 2021) [41], Garbin et al. factorized the color function  $c$  as the inner product of outputs from two MLPs: a position-dependent MLP that also predicts density  $\sigma$ , and a direction-dependent MLP. This decomposition enabled FastNeRF to efficiently cache color and density evaluations on a dense scene grid, resulting in inference speedups exceeding 3000 times. The method also leveraged hardware-accelerated ray tracing [108], which skipped empty space and terminated rays once transmittance saturation was reached.

Reiser et al. (May 2021) [42] improved the baseline NeRF by introducing **KiloNeRF**, which divided the scene into thousands of cells and trained independent MLPs to predict color and density for each cell. These small MLPs were trained using knowledge distillation from a large pretrained teacher MLP, a process closely related to baking. The method also employed early ray termination and empty space skipping. These two techniques alone sped up the baseline NeRF’s rendering by a factor of 71. Further splitting the baseline MLP into thousands of smaller MLPs improved rendering speed by a factor of 36, resulting in an overall 2000-fold acceleration.

A paper by Sun et al. [45] (November 2021) also explored this topic. The authors directly optimized a voxel grid of scalars for density. However, instead of using spherical harmonic coefficients, they used 12- and 24-dimensional features and a small, shallow decoding MLP in a hybrid representation approach. The authors used a sampling strategy analogous to the coarse-fine sampling of the original NeRF paper by training a coarse voxel grid first and then a fine voxel grid based on the geometry of the coarse grid. The model was named **Direct Voxel Grid Optimization (DVGO)**, which outperformed the baseline NeRF (1–2 days of training) with only 15 minutes of training on the Synthetic NeRF dataset.

The **Fourier PlenOctree** [109] approach was proposed by Wang et al. in February 2022. It was designed for human silhouette rendering, utilizing the domain-specific technique of Shape-From-Silhouette. The method also draws inspiration from generalizable image-conditioned NeRFs such as [49] and [75]. Initially, a coarse visual hull was constructed using sparse views predicted from a generalized NeRF and Shape-From-Silhouette. Colors and densities were then densely sampled inside this hull and stored on a coarse PlenOctree. Dense views were sampled from the PlenOctree, with transmissivity thresholding applied to eliminate most empty points. For the remaining points, new leaf densities and spherical harmonic (SH) color coefficients were generated, and the PlenOctree was updated. A Fourier Transform MLP was subsequently used to extract Fourier coefficients of the density and SH color

coefficients, which were fed into an inverse discrete Fourier transform to restore the SH coefficients and density.

The **MobileNeRF** (June 2022) [110] framework trained a NeRF-like model based on a polygonal mesh with color, feature, and opacity MLPs attached to each mesh vertex. Alpha values were discretized, and features were super-sampled for anti-aliasing. During rendering, the mesh with associated features and opacities was rasterized according to the viewing position, and a small MLP was used to shade each pixel. The method demonstrated a speed approximately 10 times faster than SNeRG [39].

The **EfficientNeRF** (July 2022) [111] was based on PlenOctree [40], choosing to use spherical harmonics and cache the trained scene in a tree. However, it introduced several improvements. Most importantly, EfficientNeRF improved training speed by using a momentum density voxel grid to store predicted density via exponential weighted average updates. During the coarse sampling stage, the grid was used to discard sampling points with zero density. During the fine sampling stage, a pivot system was also employed to speed up volume rendering. Pivot points were defined as points  $\mathbf{x}_i$  for which  $T_i\alpha_i > \epsilon$  where  $\epsilon$  was a predefined threshold, and  $T_i$  and  $\alpha_i$  were the transmittance and alpha values as defined in (4) and (5). During fine sampling, only points near the pivot points were considered. These two improvements sped up training time by a factor of 8 over the baseline NeRF [1]. The authors then cached the trained scene into a NeRF tree, resulting in rendering speeds comparable to FastNeRF [41] and exceeding that of baseline NeRF by thousands-fold.

**R2L** (March 2022) [112] distilled neural radiance fields into neural light fields via a deep residual MLP. This architecture improved rendering efficiency without requiring data beyond 2D images. Trained by distilling from a pre-trained NeRF, R2L achieved a 26–35x reduction in FLOPs and a 28–31x speedup in wall-clock time while surpassing NeRF and other efficient synthesis methods in visual quality across synthetic and real-world scenes.

2) *Non-Baked*: A popular early re-implementation of the original NeRF in JAX [113], called **JaxNeRF** [37] (December 2020), was often used as a benchmark comparison for early works seeking to improve on training and rendering speed. This model was slightly faster and more suited for distributed computing than the original TensorFlow implementation of NeRF.

In **Neural Sparse Voxel Fields (NSVF)** (July 2020), Liu et al. [38] developed a voxel-based NeRF model that models the scene as a set of radiance fields bounded by voxels. Feature representations were obtained by interpolating learnable features stored at voxel vertices, which were then processed by a shared MLP that computed  $\sigma$  and  $\mathbf{c}$ . NSVF used a sparse voxel intersection-based point sampling for rays, which was much more efficient than dense sampling or the hierarchical two-step approach of Mildenhall et al. [1]. However, this approach was more memory-intensive due to storing feature vectors on a potentially dense voxel grid.

**AutoInt** (December 2020) [74] approximated the volume rendering step. By separating the discrete volume rendering equation (4) piecewise, they developed AutoInt, which trained

the MLP  $\Phi_\theta$  via its gradient networks  $\Psi_\theta^i$ . These gradient networks shared internal parameters with and were used to reassemble the integral network  $\Phi_\theta$ . This approach allowed the rendering step to use far fewer samples, resulting in a tenfold speed-up over the baseline NeRF with only a slight decrease in quality.

**Light Field Networks (LFNs)** (June 2021) [114] presented a novel neural representation that mapped camera rays directly to radiance in 4D light space, bypassing traditional volumetric queries. This enabled real-time rendering with significantly reduced memory usage and rendering speed improvements by several orders of magnitude. By parameterizing rays using 6D Plücker coordinates, LFNs supported continuous 360° scene representation and encoded both appearance and geometry, from which sparse depth maps could be analytically derived. Although lacking inherent multi-view consistency, LFNs addressed this via a meta-learning framework that enabled light field reconstruction from sparse 2D inputs.

**Deterministic Integration for Volume Rendering (DIVEr)** (November 2021) [115] took inspiration from NSVF [38] by jointly optimizing a feature voxel grid and a decoder MLP while applying sparsity regularization and voxel culling. However, they innovated the volume rendering process by performing deterministic ray sampling on the voxel grid, which produced an integrated feature for each ray interval—defined by the intersection of the ray with a specific voxel. This feature was then decoded by an MLP to produce the density and color of the ray interval, effectively reversing the usual order of volume sampling and MLP evaluation found in NeRF methods. The DIVEr outperformed methods such as PlenOctrees [40], FastNeRF [41], and KiloNeRF [42] in terms of quality, at a comparable rendering speed.

**Instant-Neural Graphics Primitives (INGP)** (January 2022) [43] greatly improved NeRF model training and inference speed. The authors proposed a learned parametric multi-resolution hash encoding that was trained simultaneously with the NeRF model MLPs (Figure 6). They also employed advanced ray marching techniques including exponential stepping, empty space skipping, and sample compaction. This new positional encoding combined with a highly optimized MLP implementation significantly accelerated training and inference, while also enhancing the scene reconstruction accuracy of the resulting NeRF model. Within seconds of training, the method achieved results comparable to hours of training in previous NeRF models.

### C. Few Shot/Sparse Training View NeRF

The baseline NeRF requires dense multi-view images with known camera poses for each scene. A common failure case for the baseline NeRF is if the training views are not varied enough or the samples are not varied enough in pose. This leads to overfitting to individual views and nonsensical scene geometry. However, a family of NeRF methods leverages pre-trained image feature extraction networks, usually a pretrained Convolutional Neural Network (CNN) such as [68] [116], to greatly reduce the number of training samples required for successful NeRF training. Some authors call this process “deep

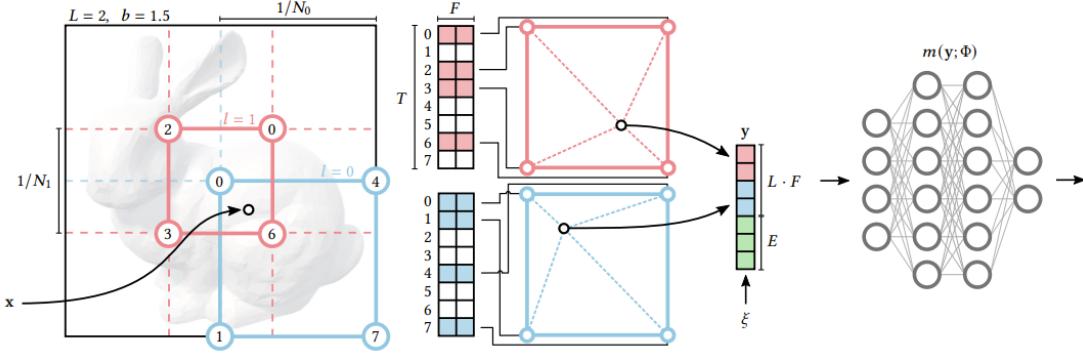


Fig. 6. Example of a hybrid representation scene representation method: Instant Neural Graphics Primitives (INGP) [43]. During training, the scene information is stored in both the hashed hierarchical voxel grid (of the number of levels  $L$ , feature dimension  $F$ , hash table size  $T$ , and resolution  $N_l$  at level  $l$ ) and the decoding MLP  $m(y; \Phi)$ . (See figure 3 in [43] for full details.)

image feature conditioning”. Certain methods [73] also used depth/3D geometry supervision to this effect (Sec. III-A2). These models often have lower training time compared to baseline NeRF models. Certain previously mentioned methods that regularize or improve geometry also reduce the training view requirement. Methods such as [24], [70], and [107] from section III-A2 and section III-A3 approach the sparse view problem by using point clouds for depth supervision.

In **pixelNeRF** (December 2020) [75], Yu et al. used the pretrained layers of a Convolutional Neural Network (and bilinear interpolation) to extract image features. Camera rays used in NeRF were then projected onto the image plane, and the image features were extracted for each query point. The features, view direction, and query points were then passed onto the NeRF MLP, which produced density and color. **General Radiance Field (GRF)** [117] (Oct 2020) by Trevithick et al. took a similar approach, with the key difference being that GRF operated in canonical space as opposed to view-space for pixelNeRF.

**MVSNeRF** (March 2021) [49] took a slightly different approach. It extracted 2D image features using a pretrained CNN. These 2D features were mapped to a 3D voxelized cost volume through plane sweeping and a variance-based cost. A pretrained 3D CNN then produced a 3D neural encoding volume, which generated per-point latent codes by interpolation. During volume rendering, the NeRF MLP used these latent features along with point coordinates and viewing direction to predict density and color. The training jointly optimized the 3D feature volume and the NeRF MLP. On the DTU dataset, MVSNeRF reached results comparable to hours of baseline NeRF training within 15 minutes.

**DietNeRF** (June 2021) [50] introduced the semantic consistency loss  $L_{sc}$  based on image features extracted from Clip-ViT [118], in addition to the standard photometric loss.

$$L_{sc} = \frac{\lambda}{2} \|\phi(I) - \phi(\hat{I})\|_2^2 \quad (14)$$

where  $\phi$  performs the Clip-ViT feature extraction on training image  $I$  and rendered image  $\hat{I}$ . This reduced to a cosine similarity loss for normalized feature vectors (eq. 5 in [50]). DietNeRF was benchmarked on a subsampled NeRF synthetic dataset [1], and DTU dataset [29]. The best-performing

method for single-view novel synthesis was a pixelNeRF [75] model fine-tuned using the semantic consistency loss of DietNeRF.

The **Neural Rays (NeuRay)** approach, by Liu et al. [76] (July 2021) also used a cost volume. From all input views, the authors estimated cost volumes (or depth maps) using multi-view stereo algorithms. From these, a CNN is used to create feature maps  $G$ . During volume rendering, from these features, both visibility and local features are extracted and processed using MLPs to extract color and alpha. The visibility is computed as a cumulative density function written as a weighted sum of sigmoid functions.

NeuRay generalizes well to new scenes and can be further fine-tuned to exceed the performance of the baseline NeRF model.

**GeoNeRF** (November 2021) [119] extracted 2D image features from every view using a pretrained Feature Pyramid Network. This method then constructed a cascading 3D cost volume using plane sweeping. From these two feature representations, for each of the  $N$  query points along a ray, one view-independent and multiple view-dependent feature tokens were extracted. These were refined using a Transformer [120]. Then, the  $N$  view-independent tokens were refined through an AutoEncoder, which returned the  $N$  densities along the ray. The  $N$  sets of view-dependent tokens were each fed into an MLP that extracted color. These networks could all be pretrained and generalized well to new scenes, as shown by the authors. Moreover, they could be fine-tuned per scene, achieving great results on the DTU [29], NeRF synthetic [1], and LLF Forward Facing [6] datasets, outperforming methods such as pixelNeRF [75] and MVSNeRF [49].

Concurrent to GeoNeRF is **LOLNeRF** (November 2021) [121], which was capable of single-shot view synthesis of human faces. It was built similarly to  $\pi$ -GAN [79] but used Generative Latent Optimization [122] instead of adversarial training [123].

**RegNeRF** (December 2021) [23] aimed to solve the problem of NeRF training with sparse input views. Unlike most other methods that approached this task by using image features from pretrained networks as a prior for conditioning NeRF volume rendering, RegNeRF employed additional depth

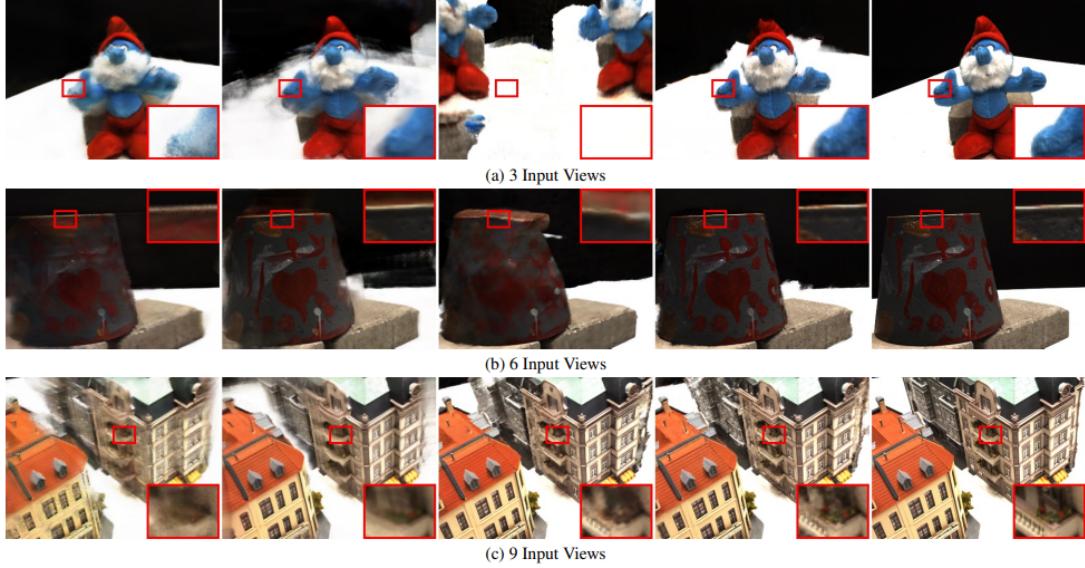


Fig. 7. Visualization of RegNeRF results on the DTU dataset under sparse view conditions with a) 3, b) 6, and c) 9 input views. Left-to-right: PixelNeRF, MVSNeRF, RegNeRF, and Ground truth (Figure 6. in 7).

and color regularization. The depth smoothness encourages world geometry to be piecewise smooth, and is defined as

$$L_{ds} = \sum_r \sum_{ij}^{S_{patch}-1} (d(\mathbf{r}_{ij}) - d(\mathbf{r}_{i+1j}))^2 + (d(\mathbf{r}_{ij}) - d(\mathbf{r}_{ij+1}))^2 \quad (15)$$

where  $d(\mathbf{r}_{ij})$  refers to the expected depth of a ray through pixel  $ij$  of a patch of size  $S_{patch}$ , from a randomly sampled unobserved viewpoint. Color regularization was also used, by estimating and maximizing the likelihood of rendered patches. This was done by training a Normalizing Flow model such as RealNVP [124] on a varied unposed dataset such as JFT-300M [125], then estimating and maximizing the log-likelihood of rendered patches. Let  $\phi$  be a learned bijection from patches to  $R^d$  where  $d = S_{patch} \cdot S_{patch} \cdot 3$ . The color regularization loss is defined as

$$L_{NLL} = \sum_r -\log p_z(\phi(P_r)) \quad (16)$$

where  $P_r$  is the predicted RGB color patch with center pixel at  $r$ , and  $-\log p_z$  is the negative log-likelihood with Gaussian  $p_z$ . In addition, RegNeRF used sampled space annealing, which attempted to fix divergent NeRF modes with high density at the ray origin at the start of training. This was done by limiting the range of sample points to within a small volume defined for all input images before extending to the full scene. Mip-NeRF [47] was used as the backbone NeRF model for these regularization and sampling techniques. The model was tested on DTU [29] and LLFF [6] datasets and outperformed models such as PixelNeRF [75], SRF [126], and MVSNeRF [49]. RegNeRF, which did not require pretraining, achieved comparable performance to these models, which were pre-trained on DTU and fine-tuned per scene, outperforming Mip-NeRF and DietNeRF [50] under sparse view conditions (see Figure 7).

**NeRFusion** (March 2022) [127] also extracted a 3D cost volume from 2D image features extracted from CNN. The volume was then processed by a sparse 3D CNN into a local feature volume. This method performs this step for each frame and then uses a GRU [128] to fuse these local feature volumes into a global feature volume, which was used to condition density and color MLPs. NeRFusion outperformed the baseline NeRF [1], NeRFingMVS [71], MVSNeRF [49] on ScanNet [31], DTU [29], and NeRF Synthetic [1] datasets.

**AutoRF** (April 2022) [129] focused on novel view synthesis of objects without background. Given 2D multi-view images, a 3D object detection algorithm was used with panoptic segmentation to extract 3D bounding boxes and object masks. The bounding boxes were used to define Normalized Object Coordinate Spaces, which were used for per-object volume rendering. An encoder CNN was used to extract appearance and shape codes, which were used in the same way as in GRAF [78]. In addition to the standard photometric loss (8), an additional occupancy loss was defined as

$$L_{occ} = \frac{1}{|W_{occ}|} \sum_{u \in W_{occ}} \log(Y_u(1/2 - \alpha) + 1/2) \quad (17)$$

where  $Y$  is the object mask, and  $W_{occ}$  is either the set of foreground or background pixels. During test time, the shape codes, appearance codes, and bounding boxes were further refined using the same loss function.

**SinNeRF** (April 2022) [26] attempted NeRF scene reconstruction from single images by integrating multiple techniques. They used image warping and known camera intrinsics and poses to create reference depth for depth supervision of unseen views. They employed adversarial training with a CNN discriminator to provide patch-wise texture supervision. Additionally, they used a pretrained ViT to extract global image features from the reference patch and unseen patch, comparing them with an L2 loss term and a global structure

prior. SinNeRF outperformed DS-NeRF [24], PixelNeRF [75], and DietNeRF [50] on the NeRF Synthetic dataset [1], the DTU dataset [29], and the LLFF Forward Facing dataset [6].

As an alternate approach, **GeoAug** (Oct 2022) [130] used data augmentation by rendering (with warping) new training images with new noisy camera poses using DSNeRF [70] as a baseline and using depth as regularization.

#### D. Generative and Conditional Models

Inspired by advances in generative 2D computer vision, generative NeRF models produce 3D geometry conditioned on text, images, or latent codes. This conditioning enables some degree of scene editing. These models fall broadly into two categories: Generative Adversarial Network-based methods and Diffusion-based methods. Typically, they leverage 2D generative models to create images of the ‘scene,’ which are then used to train the NeRF model. A major challenge before Gaussian Splatting was generating 2D images conditioned on camera pose in a way that maintained 2D consistency. Another persistent issue is the multi-face Janus problem, where generative NeRFs create avatars with multiple faces around the head. This Janus problem remains an active research area even after Gaussian Splatting’s introduction.

Compared to later 2D image generation models based on diffusion and flow-matching, the early GAN-based image generation from the early NeRF era was conditioned on latent code and could not be easily controlled with text and image-based conditioning. Without text and image-based conditioning, latent codes were also used to control aspects of scene editing.

In **NeRF-VAE** (January 2021) [131], Kosiorek et al. proposed a generative NeRF model that generalized well to out-of-distribution scenes and removed the need to train on each scene from scratch. The NeRF renderer in NeRF-VAE was conditioned on latent code, which was trained using Iterative Amortized Inference [132] [133] and a ResNet [116] encoder. The authors also introduced an attention-based scene function (as opposed to the typical MLP). NeRF-VAE consistently outperformed the baseline NeRF with a low number (5-20) of scene views, but due to lower scene expressivity, it was outperformed by baseline NeRF when a large number of views were available (100+).

1) *Generative Adversarial Network-based methods:* Adversarial training is often used for generative and/or latent-conditioned NeRF models. First developed in 2014, Generative Adversarial Networks (GANs) [123] are generative models that employ a generator  $G$  which synthesizes images from “latent code/noise,” and a discriminator  $D$  which classifies images as “synthesized” or “real.” The generator seeks to “trick” the discriminator and make its images indistinguishable from “real” training images. The discriminator seeks to maximize its classification accuracy. These two networks are trained adversarially, which is the optimization of the following minimax loss/value function,

$$\min_G \max_D \mathbb{E} x \sim \text{data} [\log D(x)] + \mathbb{E} z \sim p(z) [\log(1 - D(G(z)))] \quad (18)$$

where the generator generates images based on latent code  $z$  sampled from some distribution  $p(z)$ , which the discriminator compares to training image  $x$ . In GAN-based generative NeRF models, the generator  $G$  encompasses all novel-view synthesis steps and is thought of as the NeRF model. The generator in this case also requires an input pose in addition to a latent code. The discriminator  $D$  is usually an image classification CNN. **GRAF** (July 2020) [78] was the first NeRF model trained adversarially. It paved the way for many later works. A NeRF-based generator was conditioned on latent appearance code  $\mathbf{z}_a$  and shape code  $\mathbf{z}_s$ , and is given by

$$G(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z}_s, \mathbf{z}_a) \rightarrow (\sigma, \mathbf{c}). \quad (19)$$

In practice, the shape code, conditioning scene density, was concatenated with the embedded position, which was input to the direction-independent MLP. The appearance code, conditioning scene radiance, was concatenated with the embedded viewing direction, which was input to the direction-dependent MLP. As per baseline NeRF, images were generated via volume sampling. These were then compared using a discriminator CNN for adversarial training.

Within three months of GRAF, Chan et al. developed  $\pi$ -GAN (December 2020) [79], which also used a GAN approach to train a conditional NeRF model. The generator was a SIREN-based [134] NeRF volumetric renderer, with sinusoidal activation replacing the standard ReLU activation in the density and color MLPs.  $\pi$ -GAN outperformed GRAF [78] on standard GAN datasets such as Celeb-A [135], CARLA [136], and Cats [137].

**EG3D** (December 2021) [82] uses a novel hybrid tri-plane representation, with features stored on three axis-aligned planes and a small decoder MLP for neural rendering in a GAN framework. The GAN framework is composed of a pose-conditioned StyleGAN2 feature map generator for the tri-plane, a NeRF rendering module converting tri-plane features into low-resolution images, and a super-resolution module. The super-resolved image is then fed into a StyleGAN2 discriminator. The model achieved state-of-the-art results on the FFHQ dataset, producing realistic images and 3D geometry of human faces.

**StyleNeRF** (January 2022) [81] is a highly influential work that focuses on 2D image synthesis by using NeRF to bring 3D awareness to the StyleGAN [138] [139] image synthesis framework. StyleNeRF uses a style-code-conditioned NeRF with an upsampling module as the generator, and a StyleGAN2 [139] discriminator, and introduces a new path regularization term to the StyleGAN optimization objective.

**Pix2NeRF** (February 2022) [140] was proposed as an adversarially trained (based on  $\pi$ -GAN [79]) NeRF model which could generate NeRF-rendered images given randomly sampled latent codes and poses. In addition to the  $\pi$ -GAN loss, from which the adversarial architecture is based, the Pix2NeRF loss function also includes the following: 1) a reconstruction loss comparing  $z_{predicted}$  and  $z_{sampled}$  to ensure consistency of latent space, 2) a reconstruction loss ensuring image reconstruction quality, between  $I_{real}$  and  $I_{reconstructed}$ , where  $I_{reconstructed}$  is created by the generator from a  $z_{pred}, d_{pred}$  pair produced by the encoder, 3) a conditional adversarial

objective which prevents mode collapse towards trivial poses (see original paper for the exact expressions).

2) *Jointly Optimized Latent Models*: These models use latent codes as a key aspect of view-synthesis but jointly optimize the latent code with the scene model. The models listed in this section are not generative but instead use latent codes to account for various changeable aspects of the scene. In Generative Latent Optimization (GLO) [122], a set of randomly sampled latent codes  $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ , usually normally distributed, is paired to a set of images  $\{I_1, \dots, I_n\}$ . These latent codes are input to a generator  $G$  whose parameters are jointly optimized with the latent code using some reconstruction loss  $L$  such as  $L_2$ . I.e., the optimization is formulated as

$$\min_{G, \mathbf{z}_1, \dots, \mathbf{z}_n} \sum_{i=1}^n L(G(\mathbf{z}_i, \mathbf{u}_i), I_i) \quad (20)$$

where  $\mathbf{u}_i$  represents the other inputs not optimized over (needed in NeRF but not necessarily for other models). According to the GLO authors, this method can be thought of as a Discriminator-less GAN.

One should note that in the 2020–2023 era, many NeRF models use latent codes to condition certain aspects of the scene, such as the appearance and transient embeddings in NeRF-W. These models are typically optimized using GLO. We do not list them in this section unless the latent code is used explicitly for scene editing as a key idea in the paper.

**Edit-NeRF** (June 2021) [87] allowed for scene editing using image conditioning from user input. Edit-NeRF’s shape representation was composed of a category-specific shared shape network  $F_{share}$  and an instance-specific shape network  $F_{inst}$ .  $F_{inst}$  was conditioned on  $\mathbf{z}_s$ , whereas  $F_{shared}$  was not. In theory,  $F_{shared}$  behaved as a deformation field, not unlike [59]. The NeRF editing was formulated as a joint optimization problem of both the NeRF network parameters and the latent codes  $\mathbf{z}_s, \mathbf{z}_a$ , using GLO. They then conducted NeRF photometric loss optimization on latent codes, then on the MLP weights, and finally optimized both latent codes and weights jointly.

Innovating on Edit-NeRF, **CLIP-NeRF**’s (December 2021) [15] neural radiance field was based on the standard latent-conditioned NeRF, i.e., NeRF models conditioned on shape and appearance latent codes. However, by using Contrastive Language–Image Pre-training (CLIP), CLIP-NeRF could extract from user input text or images the induced latent space displacements by using shape and appearance mapper networks. These displacements could then be used to modify the scene’s NeRF representation based on these input text or images. This step allowed for skipping the per-edit latent code optimization used in Edit-NeRF, resulting in a speed-up of a factor of  $\sim 8$ –60, depending on the task. They also used a deformation network, similar to deformable NeRFs [59] (called the *instance-specific network* in Edit-NeRF [87]), to help with modifying the scene based on latent space displacement.

3) *Diffusion NeRF Models*: Diffusion models are a family of image generation and editing methods that have attracted widespread attention in 2022 and have largely overtaken GAN



Fig. 8. Examples of text-to-3D results from Dreamfusion (Figure 1 in [83]). A pretrained text-to-image diffusion model is used to generate a training set in a NeRF-based pipeline.

methods for 2D image generation post-Gaussian Splatting. Diffusion models are trained using a forward and a reverse diffusion process. The forward diffusion process adds noise to some input image/feature map in some T steps [141]. The reverse process is generative and can be used to create images from Gaussian noise [142] [143]. Diffusion models offer a high degree of control over image generation by allowing for text and image-based prompting/conditioning using domain-specific encoders.

**DreamFusion** (September 2022) [83] was proposed as a text-to-3D diffusion NeRF model. The NeRF model in DreamFusion was trained using images from 2D diffusion models. For each object or scene to be generated, text prompts are input into the diffusion model Imagen [142], and a mip-NeRF 360 [100] based NeRF model was trained from scratch. The text prompt allowed for control over the view of the subject at the diffusion image generation stage, with some prompting using keywords such as “overhead view,” “front view,” and “back view.” A key modification to the NeRF training was that surface color was parameterized by an MLP instead of radiance. Despite impressive results, Imagen images were generated at a 64x64 resolution. As such, the resulting NeRF model lacked the capability to produce finer details. Some results are shown in Figure 8.

In **Latent-NeRF** (November 2022) [144], the NeRF model was trained to output 64x64x4 latent features that Stable Diffusion [143] operated over, which then resulted in 512x512x3 RGB images after a decoder step. The method allowed for text guidance and shape guidance, both for further shape refinement and as a strict shape constraint.

Building upon DreamFusion, **Magic3D** (November 2022) [84] targeted the issues caused by low-resolution diffusion images. The authors used a two-stage coarse-fine approach.

In the coarse stage, Magic3D used Instant-NGP [43] as the NeRF model trained from images generated from text prompts using the image diffusion model eDiff-I [145]. The coarse geometry extracted from Instant-NGP was then placed on a mesh, which was optimized over in the fine stage using images generated with a latent diffusion model [143]. The authors noted that their method allowed for prompt-based scene editing, personalized text-to-3D generation via conditioning on an image of a subject, and style-guided text-to-3D generation. Their experiments over 397 prompts generated objects, each rated by three users, also showed a preference toward Magic3D over DreamFusion.

**RealFusion** (February 2023) [85] also used some of the same ideas but focused on single-shot scene learning. The underlying diffusion model is Stable Diffusion [143], and the underlying NeRF model is Instant-NGP [43]. The authors used single-image textual inversion as a substitute for alternate views by augmenting the input 2D image and associating it with a new vocabulary token to optimize the diffusion loss, which ensures the radiance field represents the object in the single-view photography. The 3D scene was then trained in a coarse-to-fine manner using the NeRF photometric loss, which reconstructs the scene.

**SSDNeRF** (April 2023) [146] learns generalizable 3D priors through a single-stage 3D latent diffusion model. Unlike many two-stage approaches that separately train autoencoders and diffusion models, often leading to noisy latent representations, SSDNeRF jointly optimizes NeRF and diffusion components end-to-end from multi-view images. This strategy enables robust learning even with sparse-view input. Additionally, the model supports flexible test-time sampling, allowing 3D reconstruction from arbitrary view counts. Experiments on single-object datasets show strong performance across generative and reconstruction tasks, advancing toward a general-purpose 3D learning framework.

In addition to these generative diffusion models, diffusion models are also used for single-view NeRF scene learning via image conditioning (**NeuralLift-360** (Nov 2022) [147], **NeRFDi** (December 2022) [148], **NerfDiff** (February 2023) [149], **PoseDiff** (Jan 2024) [150]), as well as for geometry regularization, (**DiffusioNeRF** (February 2023) [151]).



Fig. 9. Magic3D: Example two-stage text-to-3D generation (Figure 1 in [84]). Left: low-resolution text-to-3D using NeRF; middle and right: higher resolution text-to-3D with text prompt editing of low-resolution mesh.

### E. Unbounded Scene and Scene Composition

With attempts using NeRF models in outdoor scenes came a need to separate foreground and background, which may contain views of the sky or the horizon. These outdoor scenes also posed additional challenges in image-by-image variation in lighting and appearance. The models introduced in this

section approached this problem using various methods, with many models adapting the latent conditioning via image-by-image appearance codes. Certain models in this research area also perform semantic or instance segmentation to find applications in 3D semantic labeling.

In **NeRF in the Wild (NeRF-W)** [86] (August 2020), Martin-Brualla et al. addressed two key issues of early NeRF models. Real-life photographs of the same scene can contain per-image appearance variations due to lighting conditions, as well as transient objects that differ in each image. The density MLP was kept fixed for all images in a scene. However, NeRF-W conditioned their color MLP on a per-image appearance embedding. Moreover, another MLP conditioned on a per-image transient embedding predicted the color and density functions of transient objects.

Zhang et al. developed the **NeRF++** (October 2020) [88] model, which was adapted to generate novel views for unbounded scenes by separating the scene using a sphere. The inside of the sphere contained all foreground objects and all fictitious camera views, whereas the background was outside the sphere. The outside of the sphere was then reparameterized with radial inversion. Two separate NeRF models were trained, one for the inside of the sphere and one for the outside. The camera ray integral was also evaluated in two parts.

**GIRAFFE** (November 2020) [77] was built with a similar approach to NeRF-W, using generative latent codes and separating background and foreground MLPs for scene composition. GIRAFFE was based on GRAF, a previous model used for generative scene modeling. The framework assigned to each object in the scene its own neural feature field MLP, which produced a scalar density and a deep feature vector replacing color. These MLPs, with shared architecture and weights, took as input shape and appearance latent vectors as well as an input pose. The scene was then composed using a density-weighted sum of features. A small 2D feature map was then created from this 3D volume feature field using volume rendering, which was fed into an upsampling CNN to produce an image. GIRAFFE performed adversarial training using this synthesized image and a 2D CNN discriminator. The resulting model had a disentangled latent space, allowing for fine control over scene generation.

**Fig-NeRF** [89] (April 2021) also took on scene composition but focused on object interpolation and amodal segmentation. They used two separate NeRF models, one for the foreground and one for the background. Their foreground model was the deformable Nerfies model [59]. Their background model was an appearance NeRF conditioned on latent codes. They used two photometric losses, one for the foreground and one for the background. Fig-NeRF achieved good results for amodal segmentation and object interpolation, on datasets such as ShapeNet [33], Gelato [152], and Objectron [153].

**Yang et al.** [90] (September 2021) created a composition model that can edit objects within the scene. They used a voxel-based approach [38], creating a voxel grid of features that is jointly optimized with MLP parameters. They used two different NeRFs, one for objects and one for the scene, both of which were conditioned on interpolated voxel features. The object NeRF was further conditioned on a set of object

activation latent codes. Their method was trained and evaluated on ScanNet [31] as well as an in-house ToyDesk dataset with instance segmentation labels. They incorporated segmentation labels with a mask loss term, identifying each in-scene object.

**NeRFReN** (November 2021) [25] addressed the problem of reflective surfaces in NeRF view synthesis. The authors separated the radiance field into two components, transmitted ( $\sigma^t, \mathbf{c}^t$ ) and reflected ( $\sigma^r, \mathbf{c}^r$ ), with the final pixel value given by

$$I = I_t + \beta I_r \quad (21)$$

where  $\beta$  is the reflection fraction given by the geometry of the transmitted radiance field as

$$\beta = \sum_i T_{\sigma_i^t} (1 - \exp(-\sigma_i^t \delta_i)) \alpha_i. \quad (22)$$

$T_{\sigma_i^t}$  is given by (3), and  $\alpha_i$  by (5). In addition to the standard photometric loss, the authors used a depth smoothness  $L_d$  (eq. 8 in [25]) loss to encourage the transmitted radiance field to produce the correct geometry. Likewise, a bidirectional depth consistency loss  $L_{bdc}$  (eq. 10 in [25]) was used for the reflected radiance field. NeRFReN was able to render reflective surfaces on the authors' RFFR dataset, outperforming benchmark methods such as baseline NeRF [1], and NerfingMVS [71], as well as ablation models. The method was shown to support scene editing via reflection removal and reflection substitution.

#### F. Pose Estimation

NeRF models require both input images and camera poses to train. In the original 2020 paper, unknown poses were acquired by the COLMAP library [3], which was also often used in many subsequent NeRF models when camera poses were not provided. Typically, building models that perform both pose estimation and implicit scene representation with NeRF is formulated as an offline structure from motion (SfM) problem. In these cases, Bundle Adjustment (BA) is often used to jointly optimize the poses and the model. However, some methods

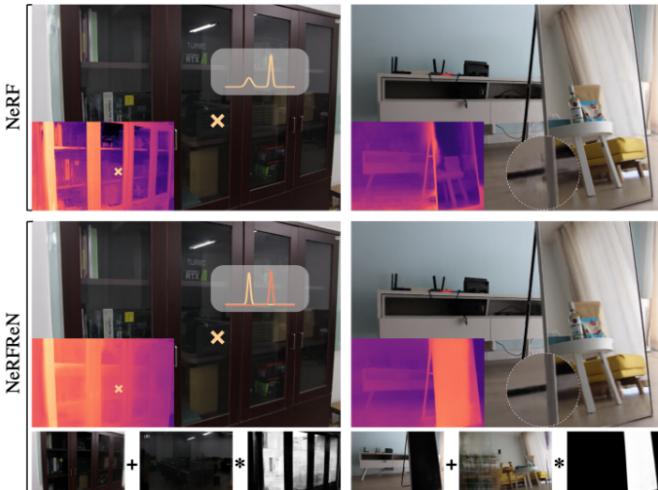


Fig. 10. NeRFReN [25] is capable of accurately reconstructing the depth (and geometry) of reflective surfaces such as glasses, a common failure case for standard approaches (Figure 1 in [25]).

also formulate this as an online simultaneous localization and mapping (SLAM) problem.

**iNeRF** (December 2020) [154] formulated pose reconstruction as an inverse problem. Given a pre-trained NeRF, using the photo-metric loss 8, Yen-Chen et al. optimized the pose instead of the network parameters. The authors used an interest-point detector and performed interest region-based sampling. The authors also performed semi-supervised experiments, where they used iNeRF pose estimation on unposed training images to augment the NeRF training set, and further train the forward NeRF. This semi-supervision was shown by the author to reduce the requirement of posed photos from the forward NeRF by 25 %.

**NeRF-** (February 2021) [96] jointly estimated NeRF model parameters and camera parameters. This allowed for the model to construct radiance fields and synthesize novel view images in an end-to-end manner. NeRF- overall achieved comparable results to using COLMAP with the 2020 NeRF model in terms of view synthesis. However, due to limitations with pose initialization, NeRF- was most suited for front-facing scenes, and struggled with rotational motion and object tracking movements.

Concurrent to NeRF- was the **Bundle-Adjusted Neural Radiance Field (BARF)** (April 2021) [97], which also jointly estimated poses alongside the training of the neural radiance field. BARF also used a coarse-to-fine registration by adaptively masking the positional encoding, similar to the technique used in Nerfies [59]. Overall, BARF results exceeded those of NeRF- on the LLFF forward-facing scenes dataset with unknown camera poses by 1.49 PSNR averaged over the eight scenes, and outperformed COLMAP registered baseline NeRF by 0.45 PSNR. Both BARF and NeRF- used naive dense ray sampling for simplicity.

Jeong et al. introduced a self-calibrating joint optimization model for NeRF (**SCNeRF**) (August 2021) [98]. Their camera calibration model can not only optimize unknown poses, but also camera intrinsic parameters for non-linear camera models such as fish-eye lens models. By using curriculum learning, they gradually introduce the nonlinear camera/noise parameters to the joint optimization. This camera optimization model was also modular and could be easily used with different NeRF models. The method outperformed BARF [97] on LLFF scenes [6].

**GNeRF** (March 2021) [80], a different type of approach by Meng et al., used pose as a generative latent code. GNeRF first obtains coarse camera poses and a radiance field with adversarial training. This is done by using a generator that takes a randomly sampled pose and synthesizes a view using NeRF-style rendering. Then, a discriminator compares the rendered view with the training image. An inversion network then takes the generated image and outputs a pose, which is compared to the sampled pose. This results in a coarse image-pose pairing. The images and poses are then jointly refined via a photometric loss in a hybrid optimization scheme. GNeRF was slightly outperformed by COLMAP-based NeRF on the Synthetic-NeRF dataset and outperformed COLMAP-based NeRF on the DTU dataset.

**GARF** (April 2022) [99] used Gaussian activations as

an effective alternative to positional encoding in NeRF, in conjunction with bundle adjustment for pose estimation. The authors showed that GARF can successfully recover scene representations from unknown camera poses, even in challenging scenes with low-textured regions, making it suitable for real-world applications.

1) *NeRF and SLAM*: Sucar et al. introduced the first NeRF-based dense online SLAM model named **iMAP** (March 2021) [93]. The model jointly optimizes the camera pose and the implicit scene representation in the form of a NeRF model, making use of continual online learning. They used an iterative two-step approach of tracking, which is pose optimization with respect to NeRF, and mapping, which is bundle adjustment for joint optimization of pose and NeRF model parameters. iMAP achieved a pose tracking speed close to the camera framerate by running the much faster tracking step in parallel with the mapping process. iMAP also used keyframe selection by optimizing the scene on a sparse and incrementally selected set of images.

Building on iMAP, **NICE-SLAM** (December 2021) [94] improved various aspects such as keyframe selection and NeRF architecture. Specifically, they used a hierarchical grid-based representation of the scene geometry, which was able to fill in gaps in iMAP reconstruction of large-scale unobserved scene features like walls and floors in certain scenes. NICE-SLAM achieved lower pose estimation errors and better scene reconstruction results than iMAP. NICE-SLAM also used approximately one-quarter of the FLOPs of iMAP, one-third of the tracking time, and half of the mapping time.

**NeRF-SLAM** (October 2022) [95] improved on the existing NeRF-based SLAM approach by using Instant-NGP [43] as its mapping module's NeRF model, in conjunction with a state-of-the-art SLAM pipeline, greatly exceeding previous benchmarks on the Replica dataset [36].

**NICERSLAM-SLAM** (February 2023) [155] is an end-to-end dense SLAM system that performs simultaneous tracking and mapping using only RGB inputs, improving on NICE-SLAM [94]. It introduces a hierarchical neural implicit representation based on signed distance functions (SDFs), enabling detailed 3D geometry and photorealistic novel view synthesis. The system leverages monocular geometric cues, optical flow, and a warping loss to guide optimization without depth supervision. Additionally, it proposes a locally adaptive SDF-to-density transformation tailored for indoor scene dynamics.

#### G. Adjacent Methods for Neural Rendering

1) *Explicit Representation and Fast MLP-less Volume Rendering*: **Plenoxel** (December 2021) [44] followed in Plenoxels's footsteps by voxelizing the scene and storing a scalar for density and spherical harmonics coefficients for direction-dependent color. However, surprisingly, Plenoxel skipped the MLP training entirely and instead fit these features directly on the voxel grid. They also obtained comparable results to NeRF++ and JaxNeRF, with training times faster by a factor of a few hundred. These results showed that the primary contribution of NeRF models is the volumetric rendering of new views given point-wise densities and color, not the density

and color MLPs themselves. **HDR-Plenoxels** [156] (August 2022) adapted this idea to HDR images by learning 3D High Dynamic Range radiance fields, scene geometry, and various camera settings from Low Dynamic Range images.

**TensoRF** (March 2022) [46] stored a scalar density and a vector feature (can work with SH coefficients, or features to be decoded via MLP) as factorized tensors. These were initially represented as a rank 3 tensor  $T_\sigma \in R^{H \times W \times D}$  and a rank 4 tensor  $T_c \in R^{H \times W \times D \times C}$ , where  $H, W, D$  are the height, width, and depth resolution of the voxel grid, and  $C$  is channel dimension. The authors then used two factorization schemes: CANDECOMP-PARAFAC (CP), which factorized the tensors as pure vector outer products, and Vector Matrix (VM), which factorized the tensors as vector/matrix outer products. These factorizations decreased the memory requirement from Plenoxels by a factor of 200 when using CP. Their VM factorization performed better in terms of visual quality, albeit at a memory tradeoff. The training speed was comparable to Pleoxels and much faster than the implicit NeRF models.

**Streaming Radiance Fields** [157] (October 2022) is an explicit representation method that specifically targeted NeRF training from video and improved on standard explicit methods. The authors employed model difference-based compression to reduce the memory requirement of the explicit representation. The method also uses a narrow-band tuning method and various training acceleration strategies. This method achieved a training time approximately 90 times faster than Plenoxels [44], with 100 to 300 times less memory requirement.

2) *Ray Transformers*: **IBRNet** [158] (February 2021) was published in 2021 as a NeRF adjacent method for view synthesis that is widely used in benchmarks. For a target view, IBRNet selected N views from the training set whose viewing directions are most similar. A CNN was used to extract features from these images. For a single query point, for each of the  $i$  input views, the known camera matrix was used to project onto the corresponding image to extract color  $c_i$  and feature  $f_i$ . An MLP was then used to refine these features  $f'_i$  to be multi-view aware and produce pooling weights  $w_i$ . For density prediction, these features were summed using the weights. This is done for each query point, and the results (of all query points along the ray) were concatenated together and fed into a ray Transformer [120], which predicted the density.

Compared to NeRF models, **Scene Rendering Transformer (SRT)** (November 2021) [159] took a different approach to volume rendering. They used a CNN to extract feature patches from scene images which were then fed into an Encoder-Decoder Transformer [120] along with camera ray and viewpoint coordinates  $\{\mathbf{o}, \mathbf{d}\}$ , which then produced the output color. The entire ray was queried at once, unlike with NeRF models. The SRT is geometry-free, and did not produce the scene's density function, nor did it rely on geometric inductive biases.

The **NeRFormer** (September 2021) [160] is a comparable concurrent model that also uses Transformers as part of the volume rendering process. NerFormer processes ray-depth-ordered feature sequences from multiple source views using alternating pooling and ray-wise attention layers, enabling

effective joint feature aggregation and ray marching. The paper also introduced the Common Objects in 3D dataset.

#### IV. APPLICATIONS OF NERF AND ADJACENT METHODS PRE-GAUSSIAN SPLATTING

This section details select works whose innovations focused on specific applications of NeRF, culminating in an organizational classification tree (Fig. 12). The classification tree also includes certain models previously introduced in Sec. III with a strong focus on applications.

Many of the methods in section III have obvious applications. The SLAM-based methods in section III-F have applications in navigation. The generative methods in III-D have potential applications in 3D graphics and digital entertainment. Additionally, NeRF models with a strong focus on high-quality geometry reconstruction can be used for photogrammetry. This section introduces methods that focus on specific types of scenes (urban, human faces, avatars, and articulated bodies), 3D reconstruction, or specific image processing tasks.

A work by **Adamkiewicz et al.** (October 2021) [161] focused on the localization and navigation aspect, and demonstrated a real-life application of a pretrained NeRF, in assisting the navigation of a robot through a church. The authors represented the environment with a pretrained NeRF model, with the robot itself approximated by a finite collection of points for collision checking. Since the NeRF model is pretrained, this method cannot be classified as a pose-estimation model, but instead demonstrated an interesting real-life use of NeRF.

**Dex-NeRF** [162] (October 2021) used NeRF’s learned density to help robots grasp objects, specifically focusing on transparent objects, which were often failure cases for depth maps produced by certain RGB-D cameras such as RealSense. The paper also presented three novel datasets focused on transparent objects: one synthetic and two real-world. Dex-NeRF improved upon baseline NeRF with respect to computed depths of transparent objects by using a fixed empirical threshold for density along rays. Their NeRF model was then used to produce a depth map used by Dex-Net [163] for grasp planning. **Evo-NeRF** [164] (November 2022) improved upon Dex-NeRF by reusing weights in sequential grasping, early termination, and an improved Radiance-Adjusted-Grasp Network capable of grasp planning with unreliable geometry.

In the following subsections, we classify applications of NeRF methods into urban reconstruction, human face and articulated body reconstruction, surface reconstruction, and low-level image processing, with the understanding that navigation and generative methods are covered in the previous subsection.



Fig. 11. Visualization of the Novel View Synthesis and 3D mesh reconstruction results of Urban Radiance Fields (Figure 1 in [72]).

##### A. Urban

The training of an urban NeRF model poses some unique challenges. First, outdoor environments are unbounded; second, the camera poses typically lack variety; third, large-scale scenes are desired. We detail in this section how these models overcome some or all of these challenges.

**Urban Radiance Fields** [72] (November 2021) aimed at applying NeRF-based view synthesis and 3D reconstruction for urban environments using sparse multi-view images supplemented by LiDAR data. In addition to the standard photometric loss, they also use a LiDAR-based depth loss  $L_{depth}$  and sight loss  $L_{sight}$ , as well as a skybox-based segmentation loss  $L_{seg}$ . These are given by

$$L_{depth} = \mathbb{E}[(z - \hat{z})^2], \quad (23)$$

$$L_{sight} = \mathbb{E}\left[\int_{t_1}^{t_2} (w(t) - \delta(z))^2 dt\right]. \quad (24)$$

$$L_{seg} = \mathbb{E}[S_i(\mathbf{r}) \int_{t_1}^{t_2} (w(t) - \delta(z))^2 dt]. \quad (25)$$

$w(t)$  is defined as  $T(t)\sigma(t)$  as defined in eq(3).  $z$  and  $\hat{z}$  are the LiDAR measured depth and estimated depth (6), respectively.  $\delta(z)$  is the Dirac delta function.  $S_i(\mathbf{r}) = 1$  if the ray goes through a sky pixel in the  $i$ th image, where sky pixels are segmented through a pretrained model [187], 0 otherwise. The depth loss forces the estimated depth  $\hat{z}$  to match the LiDAR-acquired depth. The sight loss forces the radiance to be concentrated at the surface of the measured depth. The segmentation loss forces point samples along rays through to the sky pixels to have zero density. 3D reconstruction was performed by extracting point clouds from the NeRF model during volumetric rendering. A ray was cast for each pixel in the virtual camera. Then, the estimated depth was used to place the point cloud in the 3D scene. Poisson Surface Reconstruction was used to reconstruct a 3D mesh from this generated point cloud (see Figure 11).

**Mega-NeRF** [166] (December 2021) performed large-scale urban reconstruction from aerial drone images. Mega-NeRF used a NeRF++ [88] inverse sphere parameterization to separate foreground from background. However, the authors extended the method by using an ellipsoid, which better fits the aerial point of view. They incorporated the per-image appearance embedding code of NeRF-W [86] into their model as well. They partitioned the large urban scenes into cells, each one represented by its own NeRF module, and trained each module on only the images with potentially relevant pixels. For rendering, the method also cached a coarse rendering of densities and colors into an octree.

**Block-NeRFs** [165] (February 2022) performed city-scale NeRF-based reconstruction from 2.8 million street-level images. Such large-scale outdoor datasets posed problems such as transient appearance and objects. Each individual Block-NeRF was built on mip-NeRF [47] by using its IPE, and NeRF-W [86] by using its appearance latent code optimization. Moreover, the authors used semantic segmentation to mask out transient objects such as pedestrians and cars during

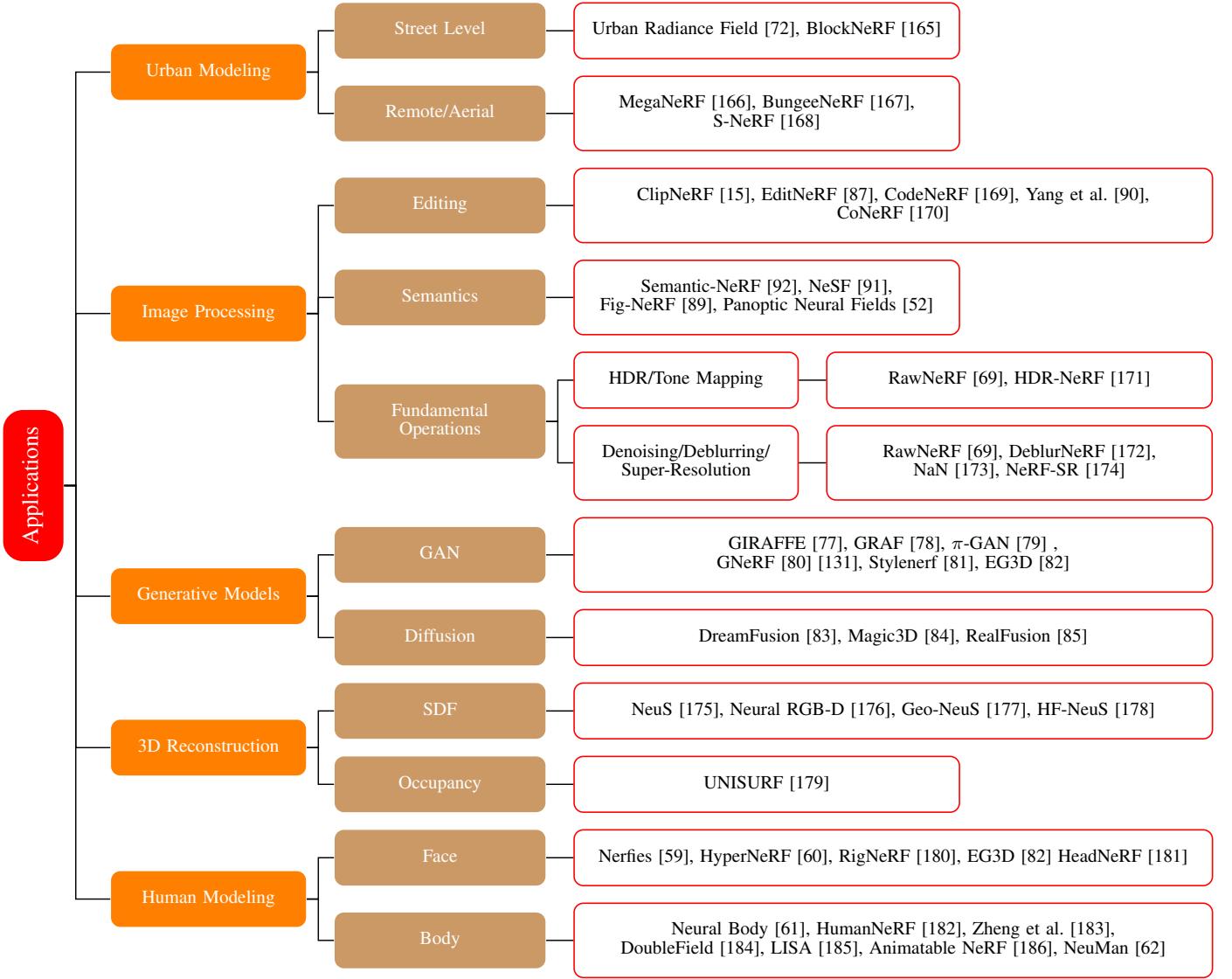


Fig. 12. Application of NeRF models. Papers are classified based on application and selected using citation numbers and GitHub star ratings.

NeRF training. A visibility MLP was trained in parallel, supervised using the transitivity function (3) and the density value generated by the NeRF MLP. These were used to discard low-visibility Block-NeRFs. Neighborhoods were divided into blocks, on which a Block-NeRF was trained. These blocks were assigned with an overlap, and images were sampled from overlapping Block-NeRFs and composited using inverse distance weighting after an appearance code matching optimization.

Other influential methods, such as **S-NeRF** [168] (April 2021), **BungeeNeRF** [167] (December 2021), also perform NeRF-based urban 3D reconstruction and view synthesis, albeit from remote sensing images.

#### B. Human Faces and Body Avatars, and Articulated Objects

A key application of NeRF models was the reconstruction of human avatars, finding applications in virtual/augmented reality, digital entertainment, and communication. Two families

of NeRF models targeted these applications: those that reconstructed human (or animal) faces and those that reconstructed human/articulated bodies. The reconstruction of human faces required the NeRF model to be robust under changes of facial expression, which often manifested as topological changes. Models typically parameterized the deformation field as additional MLP(s), potentially conditioned by latent code(s), allowing for controlled deformation from a baseline human face (see subsection IV-B). It is worth noting that many of the GAN-based NeRF models or NeRF models in GAN frameworks (subsection III-D1) were trained and tuned on datasets of human faces such as CelebA [135] or FFHQ [138], and could arguably be placed in this section. The human body poses a different set of challenges. The NeRF model had to be robust under pose changes for articulated bodies, which were often modeled as a deformation field with a template human body model.

Park et al. introduced **Nerfies** (November 2020) [59], a NeRF model built using a deformation field which strongly

improved the performance of their model in the presence of non-rigid transformations in the scene (e.g., a dynamic scene). By introducing an additional MLP that mapped input observation frame coordinates to deformed canonical coordinates and by adding elastic regularization, background regularization, and coarse-to-fine deformation regularization by adaptive masking the positional encoding, they were able to accurately reconstruct certain non-static scenes which the baseline NeRF completely failed to do. An interesting application the authors found was the creation of multi-view "selfies"<sup>1</sup>. Concurrent to Nerfies was NerFace [188] (December 2020), which also used per-frame learned latent codes, and added facial expression as a 76-dimensional coefficient of a morphable model constructed from Face2Face [189]. Subsequently, Park et al. also introduced **HyperNeRF** (June 2021) [60], which built on Nerfies by extending the canonical space to a higher dimension and using a slicing MLP that described how to return to the 3D representation using ambient space coordinates. The canonical coordinate and ambient space coordinate were then used to condition the usual density and color MLPs of baseline NeRF models. HyperNeRF achieved great results in synthesizing views in scenes with topological changes with examples such as a human opening their mouth, or a banana being peeled.

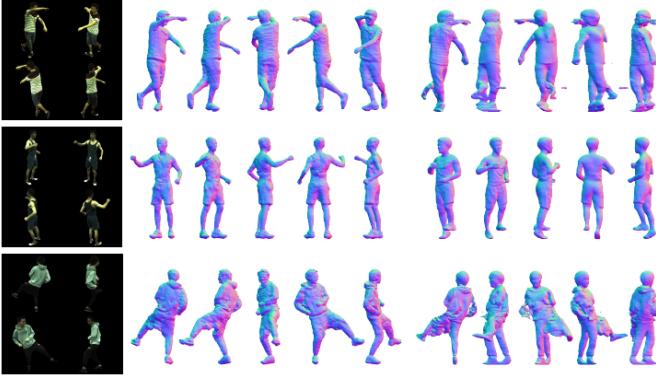


Fig. 13. Visualization of the 3D reconstruction results of Neural Bodies on the ZJU-Mocap dataset. Left-to-right: input views, Neural Bodies results, PIFuHD results. (Figure 5 in [61]).

**Neural Body** [61] (Dec 2020) applied NeRF volume rendering to human avatars with moving poses from videos. The authors first used the input video to anchor a vertex-based deformable human body model (SMPL [190]). Onto each vertex, the authors attached a 16-dimensional latent code  $Z$ . Human pose parameters  $S$  (initially estimated from video during training, can be input during inference) were then used to deform the human body model. The use of a baseline SMPL skeleton model with a neural deformation field became a foundational method in the neural field rendering of human avatars. The results are visualized in Figure 13.

**NELF** (July 2021) [191] presented a neural volumetric rendering framework that modeled scene appearance using light transport vectors, enabling realistic relighting and view synthesis of human portraits from only five input images. A UNet-style CNN extracted per-view features, and an MLP

regressed volume density and transport vectors, while environment maps were estimated to disentangle lighting. Trained on synthetic data and adapted to real images via a domain adaptation module, the approach achieved photo-realistic, lighting-consistent renderings, and outperformed existing methods in both quality and efficiency.

**CoNeRF** [170] (December 2021) was built on HyperNeRF, but allowed for easily controllable photo editing via sliders, whose values were provided to a per-attribute Hypermap deformation field, parameterized by an MLP. This was done via sparse supervised annotation of slider values and image patch masks, with an  $L_2$  loss term for slider attribute value, and a cross-entropy loss for mask supervision. CoNeRF achieved good results, using sliders to adjust facial expressions in their example dataset, which could have broad commercial applications for virtual human avatars. **RigNeRF** [180] (June 2022) also innovated on this topic by using deformation fields MLP guided by a morphable 3D face model, creating a fully 3D face portrait with controllable pose and expression.

Standard NeRF approaches struggled with moving bodies, whereas the mesh deformation approach of Neural Body was able to interpolate between frames and between poses. A popular paradigm for animating an articulated body was established, using a baseline skeleton and equipped on top of it either an MLP-based deformation field or some other implementation of a neural field. In the following two years, this inspired a large number of works such as **A-NeRF** [192] (Feb 2021), **Animatable NeRF** [186] (May 2021) and its follow-up paper **Animatable Implicit Neural Representation** (15 March 2022) [193], **DoubleField** [184] (Jun 2021), **HumanNeRF** [182] (Jan 2022), **Zheng et al. [183]** (March 2022), **NeuMan** [62] (March 2022), **PINA** (March 2022) [194] **TAVA** [195] (June 2022), **Fast-SNARF** (November 2022) [196], **ELICIT** (December 2022) [197], **X-Avatar** (March 2023) [198] which all innovated on this topic.

**PREF** [199] (September 2022) in particular focused on dynamics and motion in image sequences by regularizing estimated motion conditioned on latent embedding. Although PREF was trained and tested on image sequences of human avatars, it should apply to other domains. Many of the aforementioned papers, such as **NeuMan** and **TAVA**, also focus on animating the subject under novel (human subject) poses and motions.

**LISA** (April 2022) specifically targeted the modeling of hands by approximating human hands with a collection of rigid parts. The query points were input into MLPs, which were used to predict geometry (via SDFs) and color.

The other popular sub-area of research focused on face avatars, with constraints/requirements based on animating expressions or face topology. This research area is continuing from and improving upon the pioneering research in HyperNeRF [60] and NeRFies [60]. Some impactful works were **Neural Head Avatar** (December 2021) [200], **IMAvatar** (December 2021) [201], **INSTA** (November 2022) [202].

In 2022, an emerging research area was diffusion-based 3D avatar model generation with text guidance powered by neural fields and NeRF. **DreamAvatar** (April 2023) [203], **DreamHuman** (June 2023) [204], **AvatarVerse** (August

<sup>1</sup>Popular self-portraits in social media

2023) [205] were conceptually similar, using an SMPL model as a shape prior, and used text-guided 2D image generation via diffusion to create training data in a DreamFusion-like [83] 3D generation pipeline combining NeRF and diffusion.

### C. Image Processing

Mildenhall et al. created **RawNeRF** (November 2021) [69], adapting Mip-NeRF [47], to High Dynamic Range (HDR) image view synthesis and denoising. RawNeRF is rendered in a linear color space using raw linear images as training data. This allowed for varying exposure and tone-mapping curves, essentially applying the post-processing after NeRF rendering instead of directly using post-processed images as training data. RawNeRF is supervised with variable exposure images, with the NeRF models’ “exposure” scaled by the training image’s shutter speed, as well as a per-channel learned scaling factor. It achieved impressive results in night-time and low-light scene rendering and denoising. RawNeRF is particularly suited for scenes with low lighting.

Concurrent to RawNeRF, **HDR-NeRF** (November 2021) [171] from Xin et al. also worked on HDR view synthesis. However, HDR-NeRF approached HDR view synthesis by using Low Dynamic Range training images with variable exposure times as opposed to the raw linear images in RawNeRF. RawNeRF modelled a HDR radiance  $e(\mathbf{r}) \in [0, \infty)$  which replaced the standard  $c(\mathbf{r})$  in (1). HDR-NeRF was built on the baseline NeRF [1], using the same positional encoding and sampling strategy. The model was trained on a synthetic HDR dataset collected by the authors. HDR-NeRF strongly outperformed baseline NeRF and NeRF-W [86] on Low Dynamic Range (LDR) reconstruction, and achieved high visual assessment scores on HDR reconstruction.

**DeblurNeRF** (November 2021) [172] modeled the blurring process to restore sharp NeRFs from blurry inputs. It used a deformable sparse kernel (DSK) module to approximate dense blur kernels with sparse rays, jointly optimizing ray origins to capture the mixing of rays from different sources. The DSK adapted spatially varying blur via an MLP parameterization, enabling generalization across blur types. Training relies solely on blurry inputs, while inference removes the DSK to render clear novel views.

**NeRF-SR** (December 2021) [174] introduced a supersampling strategy that enforced multi-view consistency at a sub-pixel level, improving both image and depth super-resolution. The supersampling sampled ray directions from a grid of sub-pixels for any pixel instead of a single ray direction; secondly, it averaged the color of the sub-pixels for supervision. To further enhance details, a patch-wise warp-and-refine method propagated high-resolution reference patches across the scene using estimated 3D geometry, with minimal additional computational cost. Unlike prior methods relying on paired LR-HR data, NeRF-SR only required posed multi-view images and leveraged internal scene statistics. This marked the first framework to produce quality multi-view super-resolution under mostly low-resolution inputs.

**NaN** (April 2022) [173] incorporated inter-view and spatial awareness, enhancing noise robustness, achieving state-of-the-art results in burst denoising under challenging conditions

like large motion and high noise. Building on IBRNet, which generalizes to unseen scenes with minimal input, the approach avoided per-scene training.

*1) Semantic NeRF Models:* The training of the NeRF model with semantic understanding or capability of semantic view synthesis is a key area of development for NeRF research pre-Gaussian Splatting. Many of the subsequent Gaussian-Splatting-based semantic view synthesis and scene understanding models were built upon previous NeRF-based approaches.

**Semantic-NeRF** (March 2021) [92] was a NeRF model capable of synthesizing semantic labels for novel views. This was accomplished using an additional direction-independent MLP (branch) that took position and density MLP features as input and produced a point-wise semantic label  $s$ . The semantic labels were also generated via volume rendering by

$$S(\mathbf{r}) = \sum_i^N T_i \alpha_i \mathbf{s}_i. \quad (26)$$

The semantic labels were supervised using categorical cross-entropy loss. The method was able to train with sparse semantic label data (10% labeled), and recover semantic labels from pixel-wise noise and region/instance-wise noise. The method also achieved good label super-resolution results and label propagation from sparse point-wise annotation. It can also be used for multi-view semantic fusion, outperforming non-deep learning methods. The previously introduced Fig-NeRF [89] also employed a similar approach.

**Panoptic NeRF** [206] (March 2022) specialized in urban environments, focusing on 3D-to-2D label propagation, a key task in extending urban autonomous driving datasets. The method used two semantic fields, one learned by a semantic head and another, rigid, determined by 3D bounding boxes. According to the authors, the rigid bounding box-based semantics forced the model to learn the correct geometry, whereas the learned semantic head improved semantic understanding. Their method was evaluated on the KITTI-360 [57], outperforming previous methods of semantic label transfer.

**Panoptic Neural Fields** [52] (May 2022) first separated the “stuff” (as named by the authors), considered to be the background static objects, from the “things”, considered to be the moving objects in the scene. The “stuff” was represented by a single (two in large scenes, one for foreground, one for background) radiance field MLP, which output color, density, and semantic logits, whereas each dynamic “thing” was represented by its own radiance field inside a dynamic bounding box. The total loss function was the sum of the photometric loss function and the per-pixel cross-entropy function. The model was trained and tested on KITTI [53] and KITTI 360 [57]. In addition to novel-view synthesis and depth prediction synthesis, the model was also capable of semantic segmentation synthesis, instance segmentation synthesis, and scene editing via manipulating object-specific MLPs.

**Kobayashi et al.** (May 2022) [207] distilled the knowledge of an off-the-shelf 2D feature extractor into 3D feature fields, which they optimized in conjunction with in-scene radiance fields to produce a NeRF model with semantic understanding that allowed for scene editing. The distillation from the CLIP-

based feature extractor allowed for zero-shot segmentation from an open set of text labels or queries.

**SS-NeRF** (June 2022) [208] employed an encoding function and two position decoding functions, one direction-dependent and one direction-independent, all represented by multi-layer perceptrons. The network was trained to produce a variety of scene properties, tested on the Replica dataset [36]: color, semantic labels, surface normal, shading, keypoints, and edges using a combination of losses including MSE for color and surface normals; MAE for shading, keypoints and edges; and cross-entropy for semantic labels. This work showed that scene property synthesis was easily achievable via volume rendering and simple NeRF training without making use of advanced neural architectures.

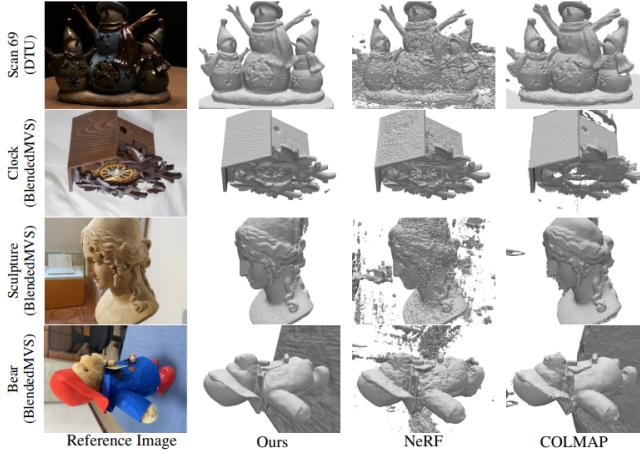


Fig. 14. NeuS [175] surface reconstruction results on the BlendedMVS [209] and DTU dataset [29]. (Figure 5. in [175])

#### D. Surface Reconstruction

The scene geometry of the NeRF model is implicit and hidden inside the neural networks. However, for certain applications, more explicit representations, such as 3D mesh, are desired. For the baseline NeRF, it is possible to extract a rough geometry by evaluating and thresholding the density MLP. The methods introduced in this subsection used innovative scene representation strategies that change the fundamental behavior of the density MLP. Strictly speaking, these methods are not NeRF and are instead categorized as general neural fields. Post-Gaussian Splatting, authors tend to emphasize this difference.

**UNISURF** (April 2021) [179] reconstructed scene surfaces by replacing the alpha value  $a_i$  at the  $i$ -th sample point used in the discretized volume rendering equation, given by (5), with a discrete occupancy function  $o(\mathbf{x}) = 1$  in occupied space, and  $o(\mathbf{x}) = 0$  in free space. This occupancy function was also computed by an MLP and essentially replaced the volume density. Surfaces were then retrieved via root-finding along rays. UNISURF outperformed benchmark methods, including using a density threshold in baseline NeRF models, as well as IDR [210]. The occupancy MLP was used to define an explicit surface geometry for the scene. A recent workshop by Tesla [211] showed that the autonomous driving module’s 3D

understanding was driven by one such NeRF-like occupancy network.

The **Neural Surface (NeuS)** [175] (June 2021) model performed volume rendering like the baseline NeRF model. However, it used signed distance functions to define scene geometries. It replaced the density-outputting MLP an MLP that outputs the signed distance function value. The density  $\rho(t)$ , which replaced  $\sigma(t)$  in the volume rendering equation (2), was then constructed as

$$\rho(t) = \max \left( \frac{-\frac{d\Phi}{dt}(f(\mathbf{r}(t)))}{\Phi(f(\mathbf{r}(t)))}, 0 \right) \quad (27)$$

where  $\Phi(\cdot)$  was the sigmoid function, and its derivative  $\frac{d\Phi}{dt}$  was the logistic density distribution. The authors showed that their model outperformed the baseline NeRF model and provided both theoretical and experimental justifications for their method and its implementation of SDF-based scene density. **HF-NeuS** (June 2022) [178] improved on NeuS by separating low-frequency details into a base SDF and high-frequency details into a displacement function, greatly increasing reconstruction quality. Concurrently, **Geo-NeuS** (May 2022) [177] introduced new multi-view constraints in the form of a multi-view geometry constraint for the SDF supervised by a sparse point cloud, and a multi-view photometric consistency constraint. **SparseNeus** (June 2022) [212], also concurrent, improved on NeuS by focusing on sparse-view SDF reconstruction using a geometry encoding volume with learnable image features as a hybrid representation method.

A concurrent work by **Azinovic et al.** [176] (April 2021) also replaced the density MLP with a truncated SDF MLP. They instead computed their pixel color as the weighted sum of sampled colors.

$$C(\mathbf{r}) = \frac{\sum_{i=1}^N w_i \mathbf{c}_i}{\sum_{i=1}^N w_i}. \quad (28)$$

$w_i$  was given by a product of sigmoid functions:

$$w_i = \Phi \left( \frac{D_i}{tr} \right) \cdot \Phi \left( -\frac{D_i}{tr} \right) \quad (29)$$

where  $tr$  was the truncation distance, which cut off any SDF value too far from individual surfaces. To account for possible multiple ray-surface intersections, subsequent truncation regions were weighted to zero and did not contribute to the pixel color. The authors also used a per-frame appearance latent code from NeRF-W [86] to account for white-balance and exposure changes. They reconstructed the triangular mesh of the scene by using Marching Cubes [213] on their truncated SDF MLP and achieved clean reconstruction results on ScanNet [31] and a private synthetic dataset.

#### V. POST-GAUSSIAN SPLATTING NEURAL RENDERING AND NERF

3D Gaussian splatting [2] is a method for 3D scene representation and novel view synthesis that represents a scene using a set of anisotropic 3D Gaussians. Each Gaussian encodes position, scale, orientation, opacity, and color, allowing the scene to be rendered through a fast differentiable splatting

process that projects and blends these primitives in screen space. Gaussian Splatting methods are typically much faster, and produce slightly better quality images, but require more memory and storage space.

Gaussian splatting-based methods, since the original paper [2] have overtaken NeRF and NeRF-adjacent Neural Rendering methods for novel-view synthesis and adjacent tasks. The shift in research momentum was so severe that implicit and hybrid neural field methods distanced themselves from the "NeRF" keyword. Nonetheless, these methods remained popular in certain applications where implicit neural field-based representations are desirable. In this section, we detail the relevant implicit and hybrid neural field methods and NeRF methods. These works are organized into the following subsections.

- Improvements in Differentiable Volume Rendering with Implicit/Hybrid Neural Field Representation (Section V-A)
- Developments in 3D Scene Representation (Section V-B)
- Diffusion and Neural Fields (Section V-C)
- SLAM with Implicit and Hybrid Neural Fields (Section V-D)
- Human avatars with Implicit and Hybrid Neural Fields (Section V-E)

#### *A. Improvements in Differentiable Volume Rendering with Implicit/Hybrid Neural Field Representation*

**NeuRBF** (September 2023) [214] is a hybrid neural field model that enhances representation accuracy and compactness by combining adaptive radial basis functions (RBFs) with grid-based RBF interpolation. It generalizes earlier feature grid-based neural field methods, introducing multi-frequency sinusoidal composition to extend the frequencies encoded by each basis function. These features are then decoded by an MLP for volume rendering with SDF reconstruction capabilities. NeuRBF achieves state-of-the-art performance across 2D image fitting, 3D signed distance field reconstruction, and neural radiance field synthesis.

**FastSR-NeRF** (December 2023) [215] introduces a simple super-resolution/upsampling CNN into the NeRF pipeline. The method trains a small, fast, and efficient NeRF model to produce low-resolution 3D consistent features and uses a fast SR model to upscale these features, significantly reducing the computational cost of volume rendering. Unlike prior NeRF+SR methods that rely on complex training procedures, distillation, or high-resolution reference images, FastSR-NeRF requires no architectural changes or heavy computing. It introduces a novel augmentation technique called random patch sampling, which improves SR performance by increasing patch diversity. The method is especially suitable for consumer-grade hardware, making neural rendering more accessible.

**Viewing Direction Gaussian Splatting (VDGS)** (December 2023) [216] is a hybrid approach that combines the fast, efficient rendering of Gaussian Splatting with the view-dependent modeling capabilities of NeRF. VDGS uses the 3D Gaussian Splatting representation of geometry and a NeRF-based encoding of color and opacity. VDGS inherits Gaussian

Splatting's real-time inference performance while significantly reducing view-dependent artifacts.

**MulFAGrid** (May 2024) [217] is a general-purpose grid-based neural field model that integrates multiplicative filters with Fourier features. Guided by a new Grid Tangent Kernel (GTK) theory, the method emphasizes spectral efficiency in high-frequency regimes, offering improved generalization and learning capacity over prior models such as InstantNGP [43] and NeuRBF [214]. MulFAGrid supports both regular and irregular grids and is trained via a joint optimization of grid and kernel features. Results across 2D image fitting, 3D signed distance field reconstruction, and novel view synthesis show excellent performance, with strong results in NeRF-based benchmarks. While slower than real-time renderers like 3DGS, MulFAGrid offers a robust, flexible alternative for neural field representation.

#### *B. Developments in 3D Scene Representation*

1) *3D Scene Understanding and Semantics:* **GP-NeRF** (November 2023) [218] is a unified framework that integrates NeRF with 2D semantic segmentation modules to enable context-aware 3D scene understanding. Unlike prior methods that treat semantic labels and radiance fields independently, GP-NeRF jointly learns radiance and semantic embedding fields using a Field Aggregation Transformer and Ray Aggregation Transformer. This architecture allows joint rendering and optimization of both fields in novel views.

**DP-RECON** (March 2025) [219] presents a decompositional 3D reconstruction method that integrates a generative diffusion prior with neural implicit representations. Given posed multiview images, the method reconstructs individual objects and backgrounds while optimizing geometry and appearance using Score Distillation Sampling (SDS) from a pre-trained Stable Diffusion model. To address conflicts between the generative prior and observed data, a novel visibility-guided optimization is introduced. This visibility map, learned via a differentiable grid based on volume transmittance, modulates SDS and reconstruction loss per pixel. The method achieves high-fidelity reconstructions, especially in occluded regions, outperforming baselines even with significantly fewer input views. Additionally, it supports detailed scene editing, stylization, and outputs decomposed meshes with UV maps.

2) *Language and Grounding for NeRF and Adjacent Neural Rendering Methods:* **Language Embedded Radiance Fields (LERF)** (May 2023) [220] is a method for integrating natural language understanding directly into NeRFs by embedding CLIP features into the 3D radiance field. Although not strictly speaking a post-GS method, this influential paper is included in this section due to its impact on the novel and mostly post-GS research area of grounding NeRF representations. LERF constructs the language field using a multi-scale feature pyramid from training views, associating each 3D location with scale-aware language semantics. To enhance semantic stability and structure, the framework also incorporates self-supervised DINO features through a shared bottleneck. The result is a model that produces 3D-consistent relevancy maps in response to natural language queries, outperforming 2D-based

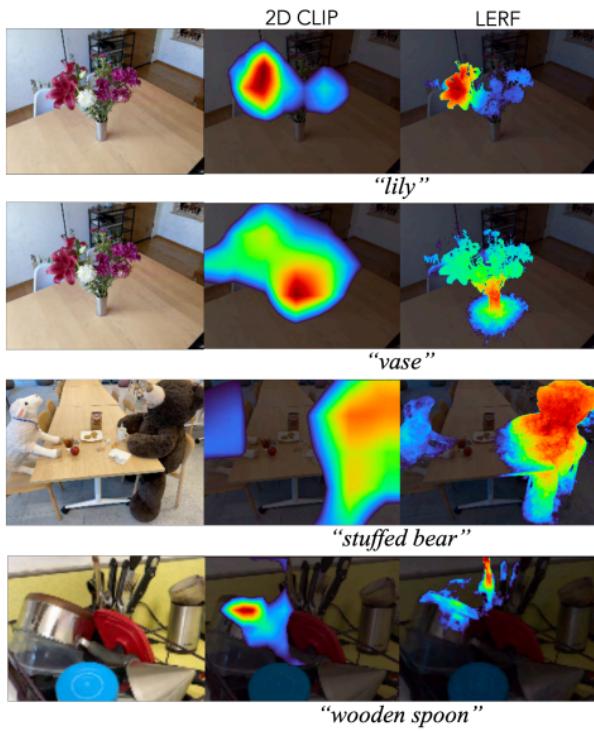


Fig. 15. Visualization of LERF embedding vision-language similarity score heatmap different text prompts, image sourced from [220]. Left: image, middle: 2D CLIP visualization (interpolated over patchwise CLIP embeddings), right: LERF visualization.

open-vocabulary detectors projected into 3D. LERF enables real-time, semantically aware 3D interaction, supporting use cases in robotics, scene understanding, and vision-language grounding.

**OV-NeRF** (February 2024) [221] is a NeRF model which performs open-vocabulary 3D semantic segmentation. The method enhances single-view semantic precision using Region Semantic Ranking (RSR), which leverages region-level cues from SAM [222] to improve boundary quality in semantic maps. To address semantic inconsistency across views, OV-NeRF introduces Cross-view Self-enhancement (CSE), which exploits NeRF’s 3D consistency to refine relevancy maps and generate novel semantic views for additional supervision. These combined strategies reduce CLIP-induced ambiguities and improve multi-view coherence. Experiments on Replica and ScanNet show substantial improvements in mIoU over prior methods, demonstrating the effectiveness and generalizability of OV-NeRF in open-vocabulary 3D scene segmentation.

**Hierarchical Neural Radiance (HNR)** (April 2024) [223] enhances vision-and-language navigation (VLN) by predicting robust, multi-level semantic features of future candidate environments. Leveraging CLIP-based vision-language embeddings, the model encodes 3D-aware language-aligned visual features into a hierarchical feature cloud and uses volume rendering to infer semantic context for unseen or occluded regions. This hierarchical encoding improves prediction quality and spatial understanding compared to prior 2D generation approaches. Integrated into a lookahead VLN framework

comprised of a cross-modal graph encoding transformer for path planning through a future path tree. The entire framework allows for language-based path planning using a Neural Field-based 3D vision system.

**Large Language and NeRF Assistant (LLaNA)** (June 2024) [224] is a multimodal language model with NeRF integration. The NeRF MLP weights are embedded into the latent space of a pre-trained language model using an encoder. This approach bypasses the need to render images or extract geometry, preserving the NeRF representation. The authors also present a new NeRF–language dataset derived from ShapeNet for NeRF-based question-answering (QA) tasks. The authors also introduce large-scale training in their subsequent work **Scaling-LLaNA** (April 2025) [225] introducing a new large-scale NeRF-Language dataset, and additional analysis on LLM size.

### C. Diffusion and Neural Fields

1) *Diffusion for 3D Generation and Editing*: Shum et al. (September 2023) [226] present a language-driven 3D scene editing using text-to-image diffusion models integrated with NeRFs. The method enables object insertion and removal by synthesizing multi-view images that incorporate both the target object and background guided by a text prompt. These images are used to iteratively refine the NeRF through a pose-conditioned dataset update strategy, which gradually integrates new views to maintain consistency and stabilize training. Unlike prior approaches that rely on explicit geometry, depth, or masks, this method requires only rough user input via 3D bounding boxes. The authors demonstrate the system’s ability to perform high-quality, view-consistent edits with minimal manual input and validate its effectiveness through extensive experiments, showcasing state-of-the-art results in NeRF-based scene manipulation.

ReconFusion (December 2023) [227] uses 2D diffusion-based priors to enhance NeRF quality, especially under sparse view conditions. A multiview-conditioned diffusion model, finetuned from a pretrained latent diffusion backbone, is trained on both real and synthetic datasets to synthesize novel views. This model serves as a regularizer within the NeRF training loop via a score distillation-like approach. The method improves reconstruction fidelity across diverse settings—mitigating artifacts such as floaters and fog in dense captures, and enabling plausible geometry in limited-view scenarios. The approach offers a general, effective prior for robust NeRF optimization.

**Comps4D** (March 2024) [228] introduces a framework for generating compositional 4D scenes (i.e. animated 3D scenes). The method moves beyond previous object-centric approaches. It decouples the process into two main stages: (1) scene decomposition to create static 3D assets and (2) motion generation guided by large language models (LLMs). The static objects are generated using a NeRF representation. The LLM plans global trajectories based on textual input, while local deformations are learned through a deformable 3D Gaussian representation. This setup allows flexible rendering and robust motion learning, even with occlusions. A compositional score

distillation mechanism optimizes object dynamics. Results show superior visual fidelity, realistic motion, and coherent object interactions compared to existing methods.

**LN3Diff** (March 2024) [229] introduces a latent-space 3D diffusion framework for conditional 3D generation. The pipeline employs a variational autoencoder to map input images into a compact, 3D-aware latent space, which is decoded into triplane representation via a transformer-based architecture. Training leverages differentiable rendering with multi-view or adversarial supervision, requiring as few as two views per scene. A convolutional tokenizer and transformer layers enable structured attention across 3D tokens, promoting coherent geometry. The latent representation supports fast amortized inference and scalable diffusion learning. LN3Diff achieves state-of-the-art performance on ShapeNet, FFHQ, and Objaverse for both 3D reconstruction and generation, outperforming existing GAN and diffusion-based baselines while offering up to 3x faster inference.

2) *Diffusion Aiding Image Processing: Inpaint4DNeRF* (December 2023) [230] introduces a text-guided generative NeRF inpainting method using diffusion models, with natural extension to 4D dynamic scenes. Given a user-specified foreground mask and text prompt, the method inpaints select seed views using Stable Diffusion, then infers coarse geometry from these views. The remaining views are refined with diffusion-based inpainting guided by the seed images and their geometry, ensuring multiview consistency.

**DiSR-NeRF** (April 2024) [231] addresses the challenge of generating high-resolution, view-consistent NeRFs from only low-resolution (LR) multi-view images, since using LR images is a common practice in NeRF training due to computational cost. Naive 2D super-resolution leads to inconsistent details across views. To fix this, DiSR-NeRF introduces two novel components. First, Iterative 3D Synchronization (I3DS) alternates between 2D diffusion-based super-resolution and NeRF training, progressively aligning details in 3D space. Second, the Renoised Score Distillation (RSD) is introduced in this paper. It refines the diffusion process by optimizing over intermediate denoised latents to produce sharper and more consistent results. Without needing high-resolution training data, DiSR-NeRF outperforms existing methods in generating high-fidelity, super-resolved NeRFs.

**MVIP-NeRF** (May 2024) [232] introduces a diffusion-based method for multiview-consistent inpainting of Neural Radiance Fields. Unlike prior approaches that rely on independent 2D inpainting per view—often resulting in inconsistencies and poor geometry, MVIP-NeRF jointly optimizes across views to ensure consistency. It employs Score Distillation Sampling (SDS) with a text-conditioned diffusion model to guide inpainting in masked regions, alongside RGB reconstruction in visible areas. To enforce geometric consistency, the method also distills normal maps. A new multi-view SDS formulation further enhances consistency under large view changes. MVIP-NeRF achieves state-of-the-art results for NeRF inpainting.

**Neural Gaffer** (June 2024) [233] presents a category-agnostic, single-view relighting framework based on a 2D diffusion model. Unlike previous models limited to specific

object classes, it generalizes across arbitrary categories and lighting environments using HDR environment maps. Trained on a synthetic dataset with physically-based materials and HDR lighting, the model captures rich lighting priors, enabling accurate and high-quality relighting from a single image. Neural Gaffer outperforms existing approaches on both synthetic and real data, integrates with image editing tasks, and extends to 3D relighting via NeRFs. It establishes a versatile diffusion-based prior for relighting in both 2D and 3D domains.

#### D. SLAM with Implicit and Hybrid Neural Fields

**CP-SLAM** (November 2023) [234] is a neural point-based (hybrid neural field) SLAM system that enables multi-agent cooperative localization and mapping, while supporting loop closure for individual agents. It introduces a new keyframe-associated neural point representation inspired by Point-NeRF, allowing per-point features to be easily adjusted during pose graph optimization. To ensure cross-agent consistency, CP-SLAM employs a two-stage distributed-to-centralized training scheme: initial decoders are trained separately for each agent, then fused and fine-tuned jointly. The system integrates odometry, loop detection, sub-map fusion, and global refinement into a unified framework. A visualization of the 3D reconstruction results is shown in Figure 16.

**SNI-SLAM** (November 2023) [235] is a dense NeRF-based RGB-D SLAM system designed for accurate real-time 3D semantic mapping. It addresses two core challenges in semantic SLAM: (1) the interdependence of appearance, geometry, and semantics, and (2) the mutual inconsistency of multi-view appearance and semantic optimization. To tackle these, SNI-SLAM introduces a hierarchical semantic encoding and cross-attention mechanism that enables mutual reinforcement between modalities. It further proposes a novel one-way decoder design to enhance inter-modal information flow without reverse interference.

**DNS-SLAM** (November 2023) [236] is a dense semantic-aware SLAM framework that builds on class-wise scene decomposition with a hybrid point-based Neural Field mapping module. Unlike prior work such as vMAP, which focuses only on reconstruction, DNS-SLAM introduces a multi-class neural scene representation that explicitly links object classes to camera poses. It leverages 2D semantic priors and multi-view image features to strengthen pose estimation through back-projected geometric constraints. A lightweight coarse model, trained via self-supervision, accelerates tracking. To further refine geometry, DNS-SLAM supervises occupancy with Gaussian-distributed priors.

**Neural Graph Mapping** (December 2023) [237] introduces a dynamic multi-field scene representation composed of small, lightweight neural fields anchored to keyframes in a pose graph. These fields deform with updated poses during loop closure, enabling consistent volumetric mapping without costly reintegration or fixed scene boundaries. The proposed RGB-D SLAM framework merges accurate sparse visual tracking with dense neural mapping, achieving robust performance across diverse scenes.

**DDN-SLAM** (January 2024) [238] integrates semantic priors with NeRF-based representation to distinguish between

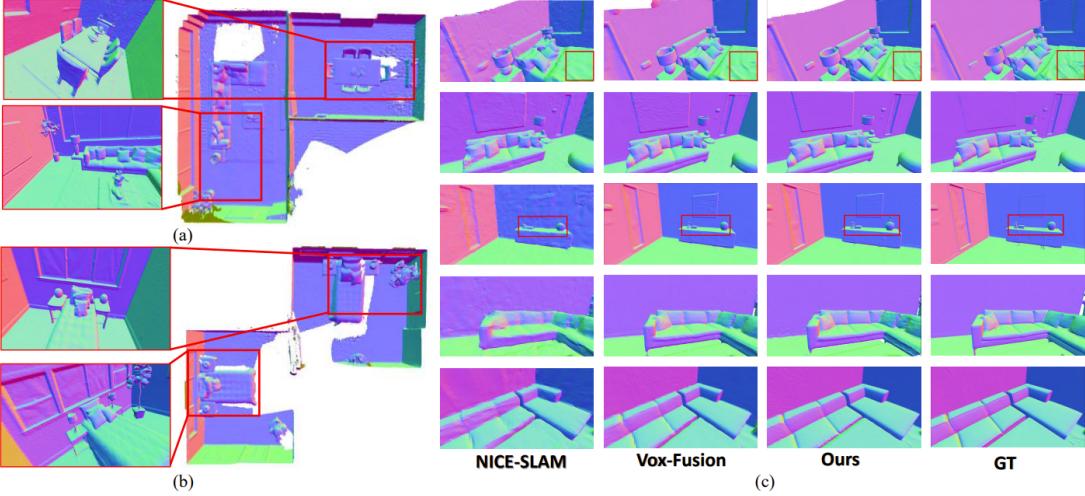


Fig. 16. Visualization of CP-SLAM 3D reconstruction results in two different scenes a) Apartment-1 and b) Apartment-2 of the Replica dataset [36]. A comparison of results against NICE-SLAM and Vox-Fusion is shown in c).

dynamic and static objects. The tracking and NeRF-based mapping are separated into four threads. The segmentation thread identifies and suppresses dynamic feature points and regions. The tracking thread extracts features, filters them via semantic and geometric cues, computes static optical flow, and produces camera poses and keyframes. The mapping thread integrates input sparse point cloud to guide the NeRF-based ray sampling and uses dynamic-aware masks to drive keyframe selection and volume rendering, preserving static surface geometry. The loop detection thread detects revisited areas and performs global bundle adjustment, enhancing long-range consistency.

**PIN-SLAM** (January 2024) [239] globally consistent SLAM system using a point-based implicit neural (PIN) representation. It replaces grid structures with neural feature points, offering spatial flexibility and elastic correction during loop closure. Mapping alternates with odometry. The mapping step is based on a hybrid neural SDF representation with explicit neural points and implicit MLP decoders. Odometry is performed correspondence-free via second-order scan-to-map optimization. A sliding window replay buffer ensures stable incremental updates. Loop closures trigger pose graph optimization and elastic deformation of neural points, enabling consistent large-scale mapping.

**KN-SLAM** (March 2024) [240] integrates local feature correspondences for coarse pose initialization for the NeRF-based mapping module, jointly optimizing the photometric loss and feature reprojection loss. Global image features and local matches are used for explicit loop detection, followed by pose graph optimization and global refinement of the neural map to ensure consistency.

**SLAIM** (April 2024) [241] introduces a coarse-to-fine tracking pipeline and improves photometric bundle adjustment through a Gaussian-filtered image signal, enhancing convergence in image alignment. It maintains NeRF's original volume density formulation while introducing a KL regularization over the ray termination distribution. It addresses the challenge of high-frequency renderings in NeRF hindering

image alignment.

**HERO-SLAM** (July 2024) [242] employs a novel multi-scale patch-based loss that aligns feature points, maps, and RGB-D pixels through warpings. An INGP [43]-like multi-resolution hybrid feature grid+MLP representation is used for neural SDF learning. Extensive evaluations on standard benchmarks show superior performance and robustness over prior implicit field-based SLAM methods, especially under challenging conditions.

**MNE-SLAM** (June 2025) [243] is the first fully distributed multi-agent neural SLAM framework that enables accurate collaborative mapping and robust camera tracking without centralized training or raw data exchange. The system uses a triplane+MLP hybrid neural field representation as a mapping module. It introduces an intra-to-inter loop closure strategy to reduce pose drift and align submaps across agents through peer-to-peer feature sharing and global consistency loss. To support benchmarking, the authors created the INS dataset, a real-world dataset with high-precision, time-continuous trajectories, and 3D mesh ground truth, suitable for evaluating various neural SLAM systems under realistic conditions.

#### E. Human avatars with Implicit and Hybrid Neural Fields

1) **Face:** **HQ3D** (March 2023) [244] introduces a method for generating highly photorealistic facial avatars using a voxelized feature grid with multiresolution hash encoding with decoding MLP in a hybrid neural implicit field. Trained on multi-view video data, the model operates with only monocular RGB input at test time and requires no mesh templates or space pruning. A novel canonical space, conditioned on video-extracted features, is regularized via an optical flow loss for artifact-free, temporally coherent reconstructions. The approach supports novel views and expressions, renders at 2K resolution, trains 4–5× faster than prior work, and runs in real-time. A new 4K multi-view dataset of 16 identities is also introduced.

Qin et al. (October 2023) [245] introduce a 3D head avatar framework that overcomes the limitations of global expres-

sion conditioning in NeRFs by proposing Spatially-Varying Expression (SVE). Unlike prior methods that use uniform global expression codes across 3D space, SVE integrates both spatial and expression features to enable fine-grained control over facial geometry and rendering. A generation network produces SVE by combining 3DMM expression parameters with position-specific features. A coarse-to-fine training strategy further refines geometry and rendering through initialization and adaptive sampling. The resulting method captures intricate details like wrinkles and eye motion with significantly higher fidelity than global-expression-based NeRFs.

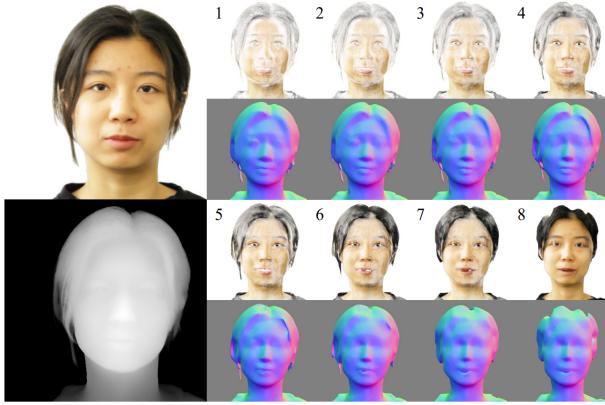


Fig. 17. Visualization of the rendered RGB and depth images, and per-layer appearance and mesh for BakedAvatar (Figure 5 in [246]).

**BakedAvatar** (November 2023) [246] proposes a novel representation for real-time 4D head avatar rendering that targets photorealism and efficiency on commodity devices. Unlike traditional mesh or NeRF-based methods, which either struggle with fine details like hair or demand heavy sampling, BakedAvatar introduces a learned manifold closely aligned with the head surface. From this, layered mesh proxies are extracted to approximate volumetric rendering while enabling fast rasterization (see Figure 17).

Bai et al. (April 2024) [247] propose a 3D neural avatar system that achieves real-time rendering with high fidelity and fine-grained control through mesh-anchored hash table blendshapes. Each vertex of a 3DMM mesh is linked to a local hash table, allowing for expression-dependent embeddings and localized facial deformations. These local blendshapes are combined using per-vertex weights predicted in UV space from driving signals. A hash encoding with decoding MLP in a hybrid neural implicit field is used to predict color and density from 3D queries using volume rendering.

**LightAvatar** (September 2024) [248] a neural light field (NeLF)-based [191] head avatar model that eliminates reliance on explicit meshes or volume rendering for a streamlined, efficient pipeline. A pretrained avatar model supervises LightAvatar via distillation. To avoid performance limitations from teacher supervision, training combines both pseudo and real data. However, since 3DMM fitting is imperfect on real data, a warping field network is introduced to correct fitting errors and enhance quality. The rendering is done in low resolution and a super-resolution module is used to generate high-resolution

images.

2) *Body*: Xu et al. (August 2023) [249] presents a method for creating relightable and animatable human avatars from sparse or monocular video. The avatar is modeled as MLPs that predict material properties (light visibility, albedo, roughness) and geometric properties (SDF and surface normal) in canonical space, transformed into world space via a neural deformation field. A hierarchical distance query algorithm blends world-space KNN and canonical SDF distances, enabling accurate pixel-surface intersection via sphere tracing and improving rendering under arbitrary poses. The method also extends distance field soft shadow (DFSS) computation to deformed SDFs, allowing efficient soft shadow rendering.

**NECA** (March 2024) [250] is a customizable neural avatar framework enabling photorealistic rendering under arbitrary pose, view, and lighting, while supporting fine-grained editing of shape, texture, and shadow. NECA learns human representation jointly in a canonical space and a surface-based UV-tangent space to capture both shared structure and high-frequency pose-dependent detail. Geometry, albedo, and shadow are predicted via separate MLPs, with optimized environmental lighting. Trained via self-supervision using photometric and normal constraints, the framework is built upon an SMPL model and uses attribute-based neural fields (MLPs) for SDF, albedo, and shadow.

**MeshAvatar** (July 2024) [251] introduces a hybrid representation for triangular human avatars that enables end-to-end learning from multi-view videos by combining explicit mesh geometry with neural signed distance and material fields. The system leverages differentiable marching tetrahedra (DMTet) to bridge mesh and implicit components, allowing compatibility with traditional rendering pipelines and hardware-accelerated ray tracing. To enhance surface reconstruction and relighting, the method integrates shadow-aware physics-based rendering (PBR), pose-driven 2D neural encoders for high-frequency detail, and stereo-estimated normal maps for weak supervision. This design achieves high-quality dynamic geometry and appearance without requiring surface tracking or pre-defined templates.

**HumanAvatar** (October 2024) [252] introduces a fast and smooth Dynamic Human NeRF model that reconstructs animatable human avatars from monocular videos. It combines HuMoR [253] for temporally coherent pose estimation, Instant-NGP [43] for accelerated canonical shape learning, and Fast-SNARF [196] for efficient deformation into pose space. To overcome the inefficiency of traditional volume rendering in dynamic settings, the method proposes a posture-sensitive space reduction and dynamic occupancy grid for skipping empty regions during rendering. This hybrid design significantly improves reconstruction quality and speed.

## VI. DISCUSSION

### A. NeRF vs. Gaussian Splatting

NeRF and Gaussian Splatting are both novel view synthesis methods. They differ in **representation**: NeRF and adjacent neural field rendering methods use implicit or hybrid neural fields to represent the 3D scene, whereas Gaussian Splatting

methods use an explicit 3D point cloud-like representation of the scene. They differ in **rendering paradigm**: NeRF and the adjacent neural field rendering methods presented in this survey use a ray-tracing-like differentiable volume rendering, sampling the neural density and color field along virtual camera rays whereas Gaussian Splatting methods use a differentiable rasterization based on 2D projection of elliptical 3D Gaussian primitives (and do not explicitly sample color values along camera rays). As such, NeRF-like methods are typically more memory and storage-efficient. However, NeRF-like methods are typically much slower than Gaussian Splatting methods and often have slightly lower view synthesis quality.

Many Gaussian Splatting methods (2023-2025) were directly adapted or drew heavy inspiration from NeRF research from the 2020-2022 era. Despite the momentum of novel view synthesis research shifting towards Gaussian Splatting in recent years, NeRF and neural field-based approaches still have certain advantages. In terms of the technical aspects, as previously mentioned, implicit and hybrid representations, such as neural fields, trade speed for memory and storage efficiency compared to explicit representations such as Gaussian Splatting. The implementation of the "splatting"-based rasterization in Gaussian Splatting is also faster than the volume rendering approach of NeRF and neural field methods, without sacrificing view synthesis quality. However, the volumetric rendering approach is more suited for volumetric scene elements such as dust or fog: these scene elements result in floaters in a standard Gaussian Splatting approach. And finally, neural field approaches are more suited for certain computations. Neural fields can be queried on a continuum of 3D coordinates, and are well-suited for representing spatially distributed properties. This contrasts with the discrete 3D point cloud-like representation, which must be further engineered to represent spatially distributed properties.

Because of the faster training and inference time and higher view synthesis quality, Gaussian Splatting methods have largely overtaken NeRF-adjacent methods for novel view synthesis and adjacent research areas, including 3D model generation, view synthesis with scene semantics, and 3D scene representation-reconstruction-editing. This is evidenced by the much lower number and impact of implicit and hybrid neural field publications in these research areas in the post-Gaussian Splatting era.

#### B. Applications of NeRF and Neural Field Rendering post-Gaussian Splatting

With the rapid development of Large Language Models, Vision Language Models [118] and pretrained 2D foundational models [222], [254], 3D scene understanding and 3D grounding emerged as a new area of research. However, despite some recent advances in this area, the field of grounded 3D scene representation and Vision Language Model-based 3D representation research, including 3D question answering and semantic understanding, is largely dominated by Gaussian Splatting-based methods [255].

SLAM and 3D human avatars remained a popular area of research. One possible reason is that implicit and hybrid

neural field representations can be advantageous for these two applications. These representations require less memory and storage and are easier to query (by simply calling the neural field at a particular point in space), as opposed to a point-cloud-like 3D Gaussian Splatting representation. 3D fields can also arise naturally from the framework's formulation, as is the case with articulated human avatar modeling.

In SLAM, the implicit and hybrid representations have lower memory and storage requirements than explicit representations such as Gaussian Splatting. This can be relevant for methods designed to perform SLAM onboard the platform itself. Additionally, as presented in Section V-D, an emerging area of research is combining SLAM with autonomous agent-based navigation (as opposed to user-controlled navigation). Certain autonomous navigation algorithms may prefer easy-to-query implicit 3D representations to a 3D point-cloud-like Gaussian Splatting representation [256].

For 3D human avatars, the dominant paradigm is to build neural fields on top of a baseline articulated SMPL [190] skeleton model. NeRF and adjacent methods fit this framework more naturally. Many Gaussian Splatting methods built upon SMPL use a combination of neural fields and Gaussian primitive representations [257]–[259]. Therefore, it is not surprising that implicit and hybrid neural field methods remain popular in this field of research.

## VII. CONCLUSION

Since the original paper by Mildenhall et al., NeRF and implicit/hybrid neural field rendering methods have made tremendous progress in terms of speed, quality, and training view requirements, addressing the weaknesses of the original model. NeRF models have found numerous applications in areas such as urban mapping, photogrammetry, image editing, labeling, processing, and 3D reconstruction and view synthesis of human avatars and urban environments. Although the research interest of the computer vision community has shifted towards Gaussian Splatting in many key research areas, there remains much interest in NeRF and implicit/hybrid neural field rendering in applications where implicit/hybrid representation, or a volumetric rendering approach, is advantageous. Moreover, many Gaussian Splatting methods drew inspiration from earlier NeRF methods. By studying earlier NeRF and neural field rendering papers, future authors may find further inspiration for other novel view synthesis-based research.

NeRF is an exciting and interesting paradigm for novel view synthesis, 3D reconstruction, 3D scene representation, and applications thereof. By providing this survey, we aim to introduce more computer vision practitioners to this field, provide a helpful reference of existing NeRF models and datasets, and motivate future research with our discussions.

## REFERENCES

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.
- [2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

- [3] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [4] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 31–42.
- [5] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 43–54.
- [6] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.
- [7] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [8] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: learning dynamic renderable volumes from images," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.
- [9] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3504–3515.
- [10] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3d shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4857–4866.
- [11] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.
- [12] F. Dellaert and L. Yen-Chen, "Neural volume rendering: NeRF and beyond," *arXiv preprint arXiv:2101.05204*, 2020.
- [13] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 641–676.
- [14] F. Zhan, Y. Yu, R. Wu, J. Zhang, and S. Lu, "Multimodal image synthesis and editing: A survey," *arXiv preprint arXiv:2112.13592*, 2021.
- [15] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.
- [16] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, "Ad-nerf: Audio driven neural radiance fields for talking head synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5784–5794.
- [17] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 867–876.
- [18] K. Jo, G. Shim, S. Jung, S. Yang, and J. Choo, "Cg-nerf: Conditional generative neural radiance fields," *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.
- [19] J. Sun, X. Wang, Y. Shi, L. Wang, J. Wang, and Y. Liu, "Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis," *arXiv preprint arXiv:2205.15517*, 2022.
- [20] Y. Chen, Q. Wu, C. Zheng, T.-J. Cham, and J. Cai, "Sem2nerf: Converting single-view semantic masks to neural radiance fields," *European conference on computer vision*, 2022.
- [21] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.
- [22] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.
- [23] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5480–5490.
- [24] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.
- [25] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, "Nerfren: Neural radiance fields with reflections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 409–18 418.
- [26] D. Xu, Y. Jiang, P. Wang, Z. Fan, H. Shi, and Z. Wang, "Sinnerf: Training neural radiance fields on complex scenes from a single image," 2022.
- [27] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.
- [28] S. Ramasinghe and S. Lucey, "Beyond periodicity: towards a unifying framework for activations in coordinate-mlps," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 142–158.
- [29] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 406–413.
- [30] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab*. CaltechDATA, May 2022. [Online]. Available: <https://data.caltech.edu/records/20164>
- [31] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.
- [32] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration," *ACM Transactions on Graphics 2017 (TOG)*, 2017.
- [33] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.
- [34] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.
- [35] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.
- [36] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.
- [37] B. Deng, J. T. Barron, and P. P. Srinivasan, "JaxNeRF: an efficient JAX implementation of NeRF," 2020. [Online]. Available: <https://github.com/google-research/google-research/tree/master/jaxnerf>
- [38] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.
- [39] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5875–5884.
- [40] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *ICCV*, 2021.
- [41] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentín, "Fastnerf: High-fidelity neural rendering at 200fps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 346–14 355.
- [42] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.
- [43] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: <https://doi.org/10.1145/3528223.3530127>
- [44] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.
- [45] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

- [46] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 333–350.
- [47] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [48] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, “Ref-nerf: Structured view-dependent appearance for neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5491–5500.
- [49] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, “Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.
- [50] A. Jain, M. Tancik, and P. Abbeel, “Putting nerf on a diet: Semantically consistent few-shot view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5885–5894.
- [51] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee, “Mine: Towards continuous depth mpi with nerf for novel view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 578–12 588.
- [52] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. J. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, “Panoptic neural fields: A semantic object-aware neural scene representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 871–12 881.
- [53] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [54] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [55] J. Fritsch, T. Kuehnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [56] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [57] Y. Liao, J. Xie, and A. Geiger, “Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [58] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.
- [59] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, “Nerfies: Deformable neural radiance fields,” *ICCV*, 2021.
- [60] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, “Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields,” *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021.
- [61] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, “Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9054–9063.
- [62] W. Jiang, K. M. Yi, G. Samei, O. Tuzel, and A. Ranjan, “Neuman: Neural human radiance field from a single video,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 402–418.
- [63] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, “Panoptic studio: A massively multiview system for social motion capture,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3334–3342.
- [64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [66] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.
- [67] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [68] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [69] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, “Nerf in the dark: High dynamic range view synthesis from noisy raw images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 190–16 199.
- [70] B. Roessler, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, “Dense depth priors for neural radiance fields from sparse input views,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 892–12 901.
- [71] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, “Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5610–5619.
- [72] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, “Urban radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.
- [73] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, “Point-nerf: Point-based neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.
- [74] D. B. Lindell, J. N. Martel, and G. Wetzstein, “Autoint: Automatic integration for fast neural volume rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 556–14 565.
- [75] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelNeRF: Neural radiance fields from one or few images,” in *CVPR*, 2021.
- [76] Y. Liu, S. Peng, L. Liu, Q. Wang, P. Wang, C. Theobalt, X. Zhou, and W. Wang, “Neural rays for occlusion-aware image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7824–7833.
- [77] M. Niemeyer and A. Geiger, “Giraffe: Representing scenes as compositional generative neural feature fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 453–11 464.
- [78] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, “Graf: Generative radiance fields for 3d-aware image synthesis,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.
- [79] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, “pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5799–5809.
- [80] Q. Meng, A. Chen, H. Luo, M. Wu, H. Su, L. Xu, X. He, and J. Yu, “Gnerf: Gan-based neural radiance field without posed camera,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6351–6361.
- [81] J. Gu, L. Liu, P. Wang, and C. Theobalt, “Stylenet: A style-based 3d aware generator for high-resolution image synthesis,” in *Tenth International Conference on Learning Representations*, 2022, pp. 1–25.
- [82] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, “Efficient geometry-aware 3d generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 123–16 133.
- [83] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “Dreamfusion: Text-to-3d using 2d diffusion,” *arXiv preprint arXiv:2209.14988*, 2022.
- [84] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, “Magic3d: High-resolution text-to-3d content creation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 300–309.
- [85] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi, “Realfusion: 360° reconstruction of any object from a single image,” *arXiv e-prints*, pp. arXiv–2302, 2023.
- [86] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections,” in *CVPR*, 2021.
- [87] S. Liu, X. Zhang, Z. Zhang, R. Zhang, J.-Y. Zhu, and B. Russell, “Editing conditional radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5773–5783.

- [88] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “Nerf++: Analyzing and improving neural radiance fields,” *arXiv:2010.07492*, 2020.
- [89] C. Xie, K. Park, R. Martin-Brualla, and M. Brown, “Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling,” in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 962–971.
- [90] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, “Learning object-compositional neural radiance field for editable scene rendering,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 779–13 788.
- [91] S. Vora\*, N. Radwan\*, K. Greff, H. Meyer, K. Genova, M. S. M. Sajjadi, E. Pot, A. Tagliasacchi, and D. Duckworth, “Neural semantic fields for generalizable semantic segmentation of 3d scenes,” *Transactions on Machine Learning Research*, 2022, <https://openreview.net/forum?id=ggPhsYCsm9>.
- [92] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, “In-place scene labelling and understanding with implicit scene representation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.
- [93] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, “imap: Implicit mapping and positioning in real-time,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.
- [94] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, “Nice-slam: Neural implicit scalable encoding for slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.
- [95] A. Rosinol, J. J. Leonard, and L. Carlone, “Nerf-slam: Real-time dense monocular slam with neural radiance fields,” *arXiv preprint arXiv:2210.13641*, 2022.
- [96] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, “NeRF—: Neural radiance fields without known camera parameters,” *arXiv preprint arXiv:2102.07064*, 2021.
- [97] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, “Barf: Bundle-adjusting neural radiance fields,” in *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [98] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, “Self-calibrating neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5846–5854.
- [99] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey, “Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 264–280.
- [100] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.
- [101] J. Zhang, Y. Zhang, H. Fu, X. Zhou, B. Cai, J. Huang, R. Jia, B. Zhao, and X. Tang, “Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 376–18 386.
- [102] X. Cheng, P. Wang, and R. Yang, “Learning depth with convolutional spatial propagation network,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2361–2379, 2019.
- [103] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman, “Learning the depths of moving people by watching frozen people,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4521–4530.
- [104] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, “Mvsnet: Depth inference for unstructured multi-view stereo,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 767–783.
- [105] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [106] E. Insafutdinov, D. Campbell, J. F. Henriques, and A. Vedaldi, “Snes: Learning probably symmetric neural surfaces from incomplete data,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 367–383.
- [107] W. Yang, G. Chen, C. Chen, Z. Chen, and K.-Y. K. Wong, “S<sup>3</sup>-nerf: Neural reflectance field from shading and shadow under a single viewpoint,” in *Advances in Neural Information Processing Systems*, 2022.
- [108] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison *et al.*, “Optix: a general purpose ray tracing engine,” *AcM transactions on graphics (tog)*, vol. 29, no. 4, pp. 1–13, 2010.
- [109] L. Wang, J. Zhang, X. Liu, F. Zhao, Y. Zhang, Y. Zhang, M. Wu, J. Yu, and L. Xu, “Fourier plenoctrees for dynamic radiance field rendering in real-time,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 524–13 534.
- [110] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, “Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- [111] T. Hu, S. Liu, Y. Chen, T. Shen, and J. Jia, “Efficientnerf: efficient neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 902–12 911.
- [112] H. Wang, J. Ren, Z. Huang, K. Olszewski, M. Chai, Y. Fu, and S. Tulyakov, “R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*. Springer, 2022, pp. 612–629.
- [113] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018. [Online]. Available: <http://github.com/google/jax>
- [114] V. Sitzmann, S. Rezhikov, B. Freeman, J. Tenenbaum, and F. Durand, “Light field networks: Neural scene representations with single-evaluation rendering,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 313–19 325, 2021.
- [115] L. Wu, J. Y. Lee, A. Bhattacharjee, Y.-X. Wang, and D. Forsyth, “Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 200–16 209.
- [116] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [117] A. Trevithick and B. Yang, “Grf: Learning a general radiance field for 3d representation and rendering,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 182–15 192.
- [118] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.
- [119] M. M. Johari, Y. Lepoittevin, and F. Fleuret, “Geonerf: Generalizing nerf with geometry priors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 365–18 375.
- [120] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [121] D. Rebain, M. Matthews, K. M. Yi, D. Lagun, and A. Tagliasacchi, “Lolnerf: Learn from one look,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1558–1567.
- [122] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, “Optimizing the latent space of generative networks,” *arXiv preprint arXiv:1707.05776*, 2017.
- [123] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [124] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” *arXiv preprint arXiv:1605.08803*, 2016.
- [125] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.
- [126] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, “Stereo radiance fields (srif): Learning view synthesis for sparse views of novel scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7911–7920.
- [127] X. Zhang, S. Bi, K. Sunkavalli, H. Su, and Z. Xu, “Nerfusion: Fusing radiance fields for large-scale scene reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5449–5458.
- [128] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

- [129] N. Müller, A. Simonelli, L. Porzi, S. R. Bulò, M. Nießner, and P. Koutschedier, “Autorf: Learning 3d object radiance fields from single view observations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3971–3980.
- [130] D. Chen, Y. Liu, L. Huang, B. Wang, and P. Pan, “Geoaug: Data augmentation for few-shot nerf with geometry constraints,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*. Springer, 2022, pp. 322–337.
- [131] A. R. Kosiorek, H. Strathmann, D. Zoran, P. Moreno, R. Schneider, S. Mokrá, and D. J. Rezende, “Nerf-vae: A geometry aware 3d scene generative model,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 5742–5752.
- [132] Y. Kim, S. Wiseman, A. Miller, D. Sontag, and A. Rush, “Semi-amortized variational autoencoders,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2678–2687.
- [133] J. Marino, Y. Yue, and S. Mandt, “Iterative amortized inference,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 3403–3412.
- [134] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.
- [135] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in *European conference on computer vision*. Springer, 2016, pp. 87–102.
- [136] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [137] W. Zhang, J. Sun, and X. Tang, “Cat head detection-how to effectively exploit shape and texture features,” in *European conference on computer vision*. Springer, 2008, pp. 802–816.
- [138] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.
- [139] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.
- [140] S. Cai, A. Obukhov, D. Dai, and L. Van Gool, “Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3981–3990.
- [141] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.
- [142] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *arXiv preprint arXiv:2205.11487*, 2022.
- [143] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10684–10695.
- [144] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or, “Latent-nerf for shape-guided generation of 3d shapes and textures,” *arXiv preprint arXiv:2211.07600*, 2022.
- [145] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro *et al.*, “ediffi: Text-to-image diffusion models with an ensemble of expert denoisers,” *arXiv preprint arXiv:2211.01324*, 2022.
- [146] H. Chen, J. Gu, A. Chen, W. Tian, Z. Tu, L. Liu, and H. Su, “Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 2416–2425.
- [147] D. Xu, Y. Jiang, P. Wang, Z. Fan, Y. Wang, and Z. Wang, “Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360° views,” *arXiv e-prints*, pp. arXiv–2211, 2022.
- [148] C. Deng, C. Jiang, C. R. Qi, X. Yan, Y. Zhou, L. Guibas, D. Anguelov *et al.*, “Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors,” *arXiv preprint arXiv:2212.03267*, 2022.
- [149] J. Gu, A. Trevithick, K.-E. Lin, J. Susskind, C. Theobalt, L. Liu, and R. Ramamoorthi, “Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion,” *arXiv preprint arXiv:2302.10109*, 2023.
- [150] S. Lee and J. Lee, “Posediff: Pose-conditioned multimodal diffusion model for unbounded scene synthesis from sparse inputs,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5007–5017.
- [151] J. Wynn and D. Turmukhambetov, “Diffusionerf: Regularizing neural radiance fields with denoising diffusion models,” *arXiv preprint arXiv:2302.12231*, 2023.
- [152] R. Martin-Brualla, R. Pandey, S. Bouaziz, M. Brown, and D. B. Goldman, “Gelato: Generative latent textured objects,” in *European Conference on Computer Vision*. Springer, 2020, pp. 242–258.
- [153] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, “Objectron: A large scale dataset of object-centric videos in the wild with pose annotations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7822–7831.
- [154] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, “Inerf: Inverting neural radiance fields for pose estimation,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.
- [155] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, “NICER-SLAM: Neural implicit scene encoding for RGB slam,” in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 42–52.
- [156] K. Jun-Seong, K. Yu-Ji, M. Ye-Bin, and T.-H. Oh, “Hdr-plenoxels: Self-calibrating high dynamic range radiance fields,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 384–401.
- [157] L. Li, Z. Shen, L. Shen, P. Tan *et al.*, “Streaming radiance fields for 3d video synthesis,” in *Advances in Neural Information Processing Systems*, 2022.
- [158] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, “Ibrnet: Learning multi-view image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.
- [159] M. S. Sajjadi, H. Meyer, E. Pot, U. Bergmann, K. Greff, N. Radwan, S. Vora, M. Lučić, D. Duckworth, A. Dosovitskiy *et al.*, “Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6229–6238.
- [160] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny, “Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10901–10911.
- [161] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, “Vision-only robot navigation in a neural radiance world,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.
- [162] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, “Dex-nerf: Using a neural radiance field to grasp transparent objects,” in *Conference on Robot Learning*. PMLR, 2022, pp. 526–536.
- [163] J. Maher, J. Liang, S. Niway, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv preprint arXiv:1703.09312*, 2017.
- [164] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, “Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects,” in *Conference on Robot Learning*. PMLR, 2022.
- [165] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.
- [166] H. Turki, D. Ramanan, and M. Satyanarayanan, “Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12922–12931.
- [167] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, “Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering,” in *The European Conference on Computer Vision (ECCV)*, 2022.
- [168] D. Derksen and D. Izzo, “Shadow neural radiance fields for multi-view satellite photogrammetry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1152–1161.

- [169] W. Jang and L. Agapito, "Codenerf: Disentangled neural radiance fields for object categories," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 949–12 958.
- [170] K. Kania, K. M. Yi, M. Kowalski, T. Trzcinski, and A. Tagliasacchi, "Conerf: Controllable neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 623–18 632.
- [171] X. Huang, Q. Zhang, Y. Feng, H. Li, X. Wang, and Q. Wang, "Hdnerf: High dynamic range neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 398–18 408.
- [172] L. Ma, X. Li, J. Liao, Q. Zhang, X. Wang, J. Wang, and P. V. Sander, "Deblur-nerf: Neural radiance fields from blurry images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 861–12 870.
- [173] N. Pearl, T. Treibitz, and S. Korman, "NAN: Noise-Aware NeRFs for Burst-Denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 672–12 681.
- [174] C. Wang, X. Wu, Y.-C. Guo, S.-H. Zhang, Y.-W. Tai, and S.-M. Hu, "Nerf-sr: High quality neural radiance fields using supersampling," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6445–6454.
- [175] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 171–27 183, 2021.
- [176] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgbd surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6290–6301.
- [177] Q. Fu, Q. Xu, Y.-S. Ong, and W. Tao, "Geo-neus: geometry-consistent neural implicit surfaces learning for multi-view reconstruction," in *Advances in Neural Information Processing Systems*, 2022.
- [178] Y. Wang, I. Skorokhodov, and P. Wonka, "Hf-neus: Improved surface reconstruction using high-frequency details," in *Advances in Neural Information Processing Systems*, 2022.
- [179] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.
- [180] S. Athar, Z. Xu, K. Sunkavalli, E. Shechtman, and Z. Shu, "Rignerf: Fully controllable neural 3d portraits," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 364–20 373.
- [181] Y. Hong, B. Peng, H. Xiao, L. Liu, and J. Zhang, "Headnerf: A real-time nerf-based parametric head model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 374–20 384.
- [182] F. Zhao, W. Yang, J. Zhang, P. Lin, Y. Zhang, J. Yu, and L. Xu, "Humannerf: Efficiently generated human radiance field from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7743–7753.
- [183] Z. Zheng, H. Huang, T. Yu, H. Zhang, Y. Guo, and Y. Liu, "Structured local radiance fields for human avatar modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 893–15 903.
- [184] R. Shao, H. Zhang, H. Zhang, M. Chen, Y.-P. Cao, T. Yu, and Y. Liu, "Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 15 872–15 882.
- [185] E. Corona, T. Hodan, M. Vo, F. Moreno-Noguer, C. Sweeney, R. Newcombe, and L. Ma, "Lisa: Learning implicit shape and appearance of hands," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 533–20 543.
- [186] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 14 314–14 323.
- [187] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [188] G. Gafni, J. Thies, M. Zollhofer, and M. Nießner, "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8649–8658.
- [189] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.
- [190] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.
- [191] T. Sun, K.-E. Lin, S. Bi, Z. Xu, and R. Ramamoorthi, "NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting," 2021.
- [192] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin, "A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 278–12 291, 2021.
- [193] X. Zhou, S. Peng, Z. Xu, J. Dong, Q. Wang, S. Zhang, Q. Shuai, and H. Bao, "Animatable implicit neural representations for creating realistic avatars from videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 6, pp. 4147–4159, 2024.
- [194] Z. Dong, C. Guo, J. Song, X. Chen, A. Geiger, and O. Hilliges, "PINN: Learning a personalized implicit neural avatar from a single RGB-D video sequence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 470–20 480.
- [195] R. Li, J. Tanke, M. Vo, M. Zollhöfer, J. Gall, A. Kanazawa, and C. Lassner, "Tava: Template-free animatable volumetric actors," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 419–436.
- [196] X. Chen, T. Jiang, J. Song, M. Rietmann, A. Geiger, M. J. Black, and O. Hilliges, "Fast-SNARF: A fast deformer for articulated neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11 796–11 809, 2023.
- [197] Y. Huang, H. Yi, W. Liu, H. Wang, B. Wu, W. Wang, B. Lin, D. Zhang, and D. Cai, "One-shot implicit animatable avatars with model-based priors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8974–8985.
- [198] K. Shen, C. Guo, M. Kaufmann, J. J. Zarate, J. Valentin, J. Song, and O. Hilliges, "X-avatar: Expressive human avatars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 911–16 921.
- [199] L. Song, X. Gong, B. Planche, M. Zheng, D. Doermann, J. Yuan, T. Chen, and Z. Wu, "Pref: Predictability regularized neural motion fields," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*. Springer, 2022, pp. 664–681.
- [200] P.-W. Grassal, M. Prinzler, T. Leistner, C. Rother, M. Nießner, and J. Thies, "Neural head avatars from monocular RGB videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 653–18 664.
- [201] Y. Zheng, V. F. Abrevaya, M. C. Bühlert, X. Chen, M. J. Black, and O. Hilliges, "IMavatar: Implicit morphable head avatars from videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 545–13 555.
- [202] W. Zielonka, T. Bolkart, and J. Thies, "Instant volumetric head avatars," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 4574–4584.
- [203] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong, "Dreamavatar: Text-and-shape guided 3D human avatar generation via diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 958–968.
- [204] N. Kolotouros, T. Alldieck, A. Zanfir, E. Bazavan, M. Fieraru, and C. Sminchisescu, "Dreamhuman: Animatable 3d avatars from text," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10 516–10 529, 2023.
- [205] H. Zhang, B. Chen, H. Yang, L. Qu, X. Wang, L. Chen, C. Long, F. Zhu, D. Du, and M. Zheng, "Avatarverse: High-quality & stable 3d avatar creation from text and pose," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7124–7132.
- [206] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, and Y. Liao, "Panoptic NeRF: 3d-to-2d label transfer for panoptic urban scene segmentation," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 1–11.
- [207] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing NeRF for editing via feature field distillation," *Advances in neural information processing systems*, vol. 35, pp. 23 311–23 330, 2022.
- [208] M. Zhang, S. Zheng, Z. Bao, M. Hebert, and Y.-X. Wang, "Beyond rgb: Scene-property synthesis with neural radiance fields," in *Proceedings of*

- the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 795–805.
- [209] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan, “Blendedmvs: A large-scale dataset for generalized multi-view stereo networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1790–1799.
- [210] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, “Multiview neural surface reconstruction by disentangling geometry and appearance,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.
- [211] A. Elluswamy, Tesla, workshop on autonomous driving, CVPR 2022. [Online]. Available: <https://www.youtube.com/watch?v=jPCV4GKX9Dw>
- [212] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang, “Sparseneus: Fast generalizable neural surface reconstruction from sparse views,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 210–227.
- [213] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [214] Z. Chen, Z. Li, L. Song, L. Chen, J. Yu, J. Yuan, and Y. Xu, “Neurbf: A neural fields representation with adaptive radial basis functions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4182–4194.
- [215] C.-Y. Lin, Q. Fu, T. Merth, K. Yang, and A. Ranjan, “FastSR-NeRF: improving NeRF efficiency on consumer devices with a simple super-resolution pipeline,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6036–6045.
- [216] D. Malarz, W. Smolak-Dyżewska, J. Tabor, S. Tadeja, and P. Spurek, “Gaussian splatting with NeRF-based color and opacity,” *Computer Vision and Image Understanding*, vol. 251, p. 104273, 2025.
- [217] Z. Zhao, F. Fan, W. Liao, and J. Yan, “Grounding and enhancing grid-based models for neural fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19425–19435.
- [218] H. Li, D. Zhang, Y. Dai, N. Liu, L. Cheng, J. Li, J. Wang, and J. Han, “Gp-nerf: Generalized perception nerf for context-aware 3d scene understanding,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 21708–21718.
- [219] J. Ni, Y. Liu, R. Lu, Z. Zhou, S.-C. Zhu, Y. Chen, and S. Huang, “Decompositional neural scene reconstruction with generative diffusion prior,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6022–6033.
- [220] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, “Lerf: Language embedded radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19729–19739.
- [221] G. Liao, K. Zhou, Z. Bao, K. Liu, and Q. Li, “OV-NeRF: Open-vocabulary neural radiance fields with vision and language foundation models for 3D semantic understanding,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
- [222] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo et al., “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.
- [223] Z. Wang, X. Li, J. Yang, Y. Liu, J. Hu, M. Jiang, and S. Jiang, “Lookahead exploration with neural radiance representation for continuous vision-language navigation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 13753–13762.
- [224] A. Amaduzzi, P. Zama Ramirez, G. Lisanti, S. Salti, and L. Di Stefano, “Llanal: Large language and nerf assistant,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 1162–1195, 2024.
- [225] A. Amaduzzi, P. Z. Ramirez, G. Lisanti, S. Salti, and L. Di Stefano, “Scaling LLaNA: Advancing NeRF-Language Understanding Through Large-Scale Training,” *arXiv preprint arXiv:2504.13995*, 2025.
- [226] K. C. Shum, J. Kim, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, “Language-driven object fusion into neural radiance fields with pose-conditioned dataset updates,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 5176–5187.
- [227] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole et al., “Reconfusion: 3d reconstruction with diffusion priors,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 21551–21561.
- [228] D. Xu, H. Liang, N. P. Bhatt, H. Hu, H. Liang, K. N. Plataniotis, and Z. Wang, “Comp4d: Llm-guided compositional 4d scene generation,” *arXiv preprint arXiv:2403.16993*, 2024.
- [229] Y. Lan, F. Hong, S. Yang, S. Zhou, X. Meng, B. Dai, X. Pan, and C. C. Loy, “Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation,” in *European Conference on Computer Vision*. Springer, 2024, pp. 112–130.
- [230] H. Jiang, H. Sun, R. Li, C.-K. Tang, and Y.-W. Tai, “Inpaint4DNeRF: Promptable Spatio-Temporal NeRF Inpainting with Generative Diffusion Models,” *arXiv preprint arXiv:2401.00208*, 2023.
- [231] J. L. Lee, C. Li, and G. H. Lee, “DiSR-NeRF: Diffusion-Guided View-Consistent Super-Resolution NeRF,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20561–20570.
- [232] H. Chen, C. C. Loy, and X. Pan, “Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5344–5353.
- [233] H. Jin, Y. Li, F. Luan, Y. Xiangli, S. Bi, K. Zhang, Z. Xu, J. Sun, and N. Snavely, “Neural gaffer: Relighting any object via diffusion,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [234] J. Hu, M. Mao, H. Bao, G. Zhang, and Z. Cui, “CP-SLAM: Collaborative neural point-based SLAM system,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 39429–39442, 2023.
- [235] S. Zhu, G. Wang, H. Blum, J. Liu, L. Song, M. Pollefeys, and H. Wang, “SNI-SLAM: Semantic neural implicit slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21167–21177.
- [236] K. Li, M. Niemeyer, N. Navab, and F. Tombari, “DNS-SLAM: Dense Neural Semantic-Informed SLAM,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 7839–7846.
- [237] L. Bruns, J. Zhang, and P. Jensfelt, “Neural graph map: Dense mapping with efficient loop closure integration,” in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 2900–2909.
- [238] M. Li, Y. Zhou, G. Jiang, T. Deng, Y. Wang, and H. Wang, “DDN-SLAM, Real-time dense dynamic neural implicit SLAM,” *arXiv preprint arXiv:2401.01545*, 2024.
- [239] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss, “PIN-SLAM: LiDAR SLAM using a point-based implicit neural representation for achieving global map consistency,” *IEEE Transactions on Robotics*, 2024.
- [240] X. Wu, Z. Liu, Y. Tian, Z. Liu, and W. Chen, “KN-SLAM: Keypoints and neural implicit encoding SLAM,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.
- [241] V. Cartillier, G. Schindler, and I. Essa, “Slaim: Robust dense neural slam for online tracking and mapping,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2862–2871.
- [242] Z. Xin, Y. Yue, L. Zhang, and C. Wu, “Hero-SLAM: Hybrid enhanced robust optimization of Neural SLAM,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8610–8616.
- [243] T. Deng, G. Shen, C. Xun, S. Yuan, T. Jin, H. Shen, Y. Wang, J. Wang, H. Wang, D. Wang et al., “MNE-SLAM: Multi-Agent Neural SLAM for Mobile Robots,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1485–1494.
- [244] K. Teota, X. Pan, H. Kim, P. Garrido, M. Elgharib, and C. Theobalt, “Hq3davatar: High-quality implicit 3d head avatar,” *ACM Transactions on Graphics*, vol. 43, no. 3, pp. 1–24, 2024.
- [245] M. Qin, Y. Liu, Y. Xu, X. Zhao, Y. Liu, and H. Wang, “High-fidelity 3d head avatars reconstruction through spatially-varying expression conditioned neural radiance field,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 5, 2024, pp. 4569–4577.
- [246] H.-B. Duan, M. Wang, J.-C. Shi, X.-C. Chen, and Y.-P. Cao, “Bakedavat: Baking neural fields for real-time head avatar synthesis,” *ACM Transactions on Graphics (ToG)*, vol. 42, no. 6, pp. 1–17, 2023.
- [247] Z. Bai, F. Tan, S. Fanello, R. Pandey, M. Dou, S. Liu, P. Tan, and Y. Zhang, “Efficient 3D implicit head avatar with mesh-anchored hash table blendshapes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1975–1984.
- [248] H. Wang, F. Tan, Z. Bai, Y. Zhang, S. Liu, Q. Xu, M. Chai, A. Prabhu, R. Pandey, S. Fanello et al., “Lightavatar: Efficient head avatar as dynamic neural light field,” in *European Conference on Computer Vision*. Springer, 2025, pp. 183–201.

- [249] Z. Xu, S. Peng, C. Geng, L. Mou, Z. Yan, J. Sun, H. Bao, and X. Zhou, “Relightable and animatable neural avatar from sparse-view video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 990–1000.
- [250] J. Xiao, Q. Zhang, Z. Xu, and W.-S. Zheng, “Neca: Neural customizable human avatar,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20091–20101.
- [251] Y. Chen, Z. Zheng, Z. Li, C. Xu, and Y. Liu, “Meshavtar: Learning high-quality triangular human avatars from multi-view videos,” in *European Conference on Computer Vision*. Springer, 2024, pp. 250–269.
- [252] Z. Huang, S. M. Erfani, S. Lu, and M. Gong, “Efficient neural implicit representation foYr 3D human reconstruction,” *Pattern Recognition*, vol. 156, p. 110758, 2024.
- [253] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, “HuMoR: 3D human motion model for robust pose estimation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11488–11499.
- [254] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, “Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection,” in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.
- [255] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, “3D Gaussian splatting as new era: A survey,” *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [256] K. Yang, Y. Cheng, Z. Chen, and J. Wang, “SLAM meets NeRF: A survey of implicit SLAM methods,” *World Electric Vehicle Journal*, vol. 15, no. 3, p. 85, 2024.
- [257] M. Kocabas, J.-H. R. Chang, J. Gabriel, O. Tuzel, and A. Ranjan, “Hugs: Human Gaussian splats,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 505–515.
- [258] A. Moreau, J. Song, H. Dhamo, R. Shaw, Y. Zhou, and E. Pérez-Pellitero, “Human Gaussian Splatting: Real-time Rendering of Animatable Avatars,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 788–798.
- [259] S. Hu, T. Hu, and Z. Liu, “Gauhuman: Articulated Gaussian splatting from monocular human videos,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 20418–20431.