

NeRF in Robotics: A Survey

Guangming Wang, Lei Pan, Songyou Peng, Shaohui Liu, Chenfeng Xu, Yanzi Miao, Wei Zhan, Masayoshi Tomizuka, *Life Fellow, IEEE*, Marc Pollefeys, *Fellow, IEEE*, and Hesheng Wang, *Senior Member, IEEE*

Abstract—Meticulous 3D environment representations have been a longstanding goal in computer vision and robotics fields. The recent emergence of neural implicit representations has introduced radical innovation to this field as implicit representations enable numerous capabilities. Among these, the Neural Radiance Field (NeRF) has sparked a trend because of the huge representational advantages, such as simplified mathematical models, compact environment storage, and continuous scene representations. Apart from computer vision, NeRF has also shown tremendous potential in the field of robotics. Thus, we create this survey to provide a comprehensive understanding of NeRF in the field of robotics. By exploring the advantages and limitations of NeRF, as well as its current applications and future potential, we hope to shed light on this promising area of research. Our survey is divided into two main sections: *The Application of NeRF in Robotics* and *The Advance of NeRF in Robotics*, from the perspective of how NeRF enters the field of robotics. In the first section, we introduce and analyze some works that have been or could be used in the field of robotics from the perception and interaction perspectives. In the second section, we show some works related to improving NeRF’s own properties, which are essential for deploying NeRF in the field of robotics. In the discussion section of the review, we summarize the existing challenges and provide some valuable future research directions for reference.

Index Terms—Robotic, NeRF, scene perception and interaction, NeRF’s properties, challenges and future directions

I. INTRODUCTION

HUMANS are utilizing Deep Learning (DL) as a tool to design and develop state-of-the-art robots across various fields. These robots are surpassing even the foremost human experts in their respective fields [2], [3]. Neural networks are demonstrating potential by enabling robots to perform tasks more naturally and intelligently, consequently changing the traditional paradigms of perception and motion of robots [4].

Neural Radiance Field (NeRF) [1] trains a neural network whose weights save a specific implicit representation for scenes. Volume rendering [5], as the core of the NeRF framework, enables NeRF to learn 3D scene representation from a set of 2D images with known camera poses and

*This work was supported in part by the Natural Science Foundation of China under Grant U1613218 and 61722309. The first two authors contributed equally. Corresponding Author: Hesheng Wang.

G. Wang is with the University of Cambridge. This paper was partially completed when he was visiting ETH Zurich.

H. Wang is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and the Key Laboratory of System Control and Information Processing, Ministry of Education of China.

L. Pan and Y. Miao are with Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, School of Information and Control Engineering, Advanced Robotics Research Center, China University of Mining and Technology, Xuzhou 221116, China.

S. Peng, S. Liu, and M. Pollefeys are with ETH Zurich, Zurich, Switzerland.

C. Xu, W. Zhan, and M. Tomizuka are with Mechanical Systems Control Laboratory, University of California, Berkeley, USA.

enables continuous photorealistic view rendering from any novel viewpoints. NeRF’s striking ability of the novel view rendering has attracted the interest of various researchers and inspired a series of studies [6]–[13]. These works offer novel possibilities for representing or processing perception and motion in robotics, introducing a generalized NeRF concept with high potential in robotics.

Since the debut of NeRF in 2020, several survey papers [14]–[17] have been published to showcase advances in this growing field. Because of the popularity of NeRF, Dellaert et al. [14] produced the first survey of NeRF in the same year. This concise survey describes the background of NeRF, analyses the strengths and weaknesses of NeRF, and describes related work available at the time that involved extensions to various aspects of NeRF. Xie et al. [15] conducted a survey that provides an extensive review of Neural Fields from techniques and applications. Gao et al. [16] presented a comprehensive NeRF survey containing several classical NeRF works as well as several typical datasets. Rabby et al. [17] focused on detailed summaries and comparisons of related works in terms of attribute enhancement of NeRF. Among them, [15] encompasses a broad range of background and theory knowledge. These surveys [14], [16], [17], on the other hand, focus on NeRF at various developmental stages, summarizing the evolution of this field. We recommend you go through the aforementioned works if you want to gain a comprehensive and multidimensional understanding of the Neural Fields field.

We have observed a profound integration from NeRF to robotics, with a lot of creative ideas. Different from the focus on view synthesis in the surveys mentioned above [14]–[17], our survey is situated within the context of robotics, providing a fresh perspective on NeRF. We comprehensively introduce the application of NeRF in robotics and related works with promising applications in robotics. Additionally, we analyze relevant research endeavours aimed at enhancing NeRF’s performance for more effective deployment in robotic applications. Finally, we delve into the existing challenges within this emerging field and offer insights into future directions. The overall framework of the survey is illustrated in Figure 1.

Section II provides a brief overview of the background knowledge of NeRF, with a primary focus on the core concepts and mathematical principles of NeRF. Section III, as the main body of this survey, categorizes various application directions of NeRF in the field of robotics. Related works are presented and meticulously analyzed. Section IV, from the perspective of enhancing NeRF’s capabilities, introduces relevant enhancement efforts. These enhancements aim to facilitate the effective deployment of NeRF in the field of

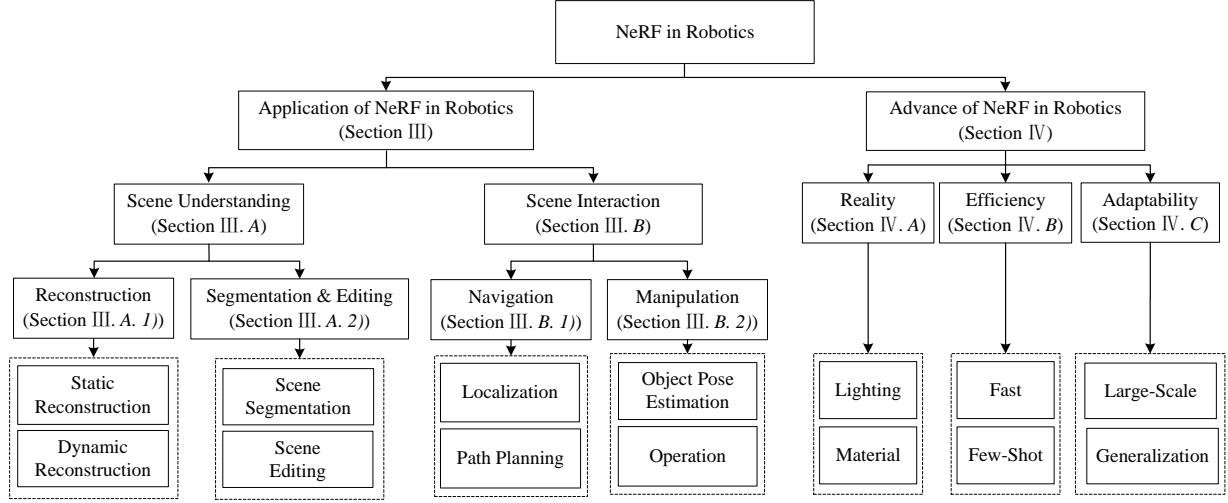


Fig. 1. A taxonomy of NeRF in robotics.

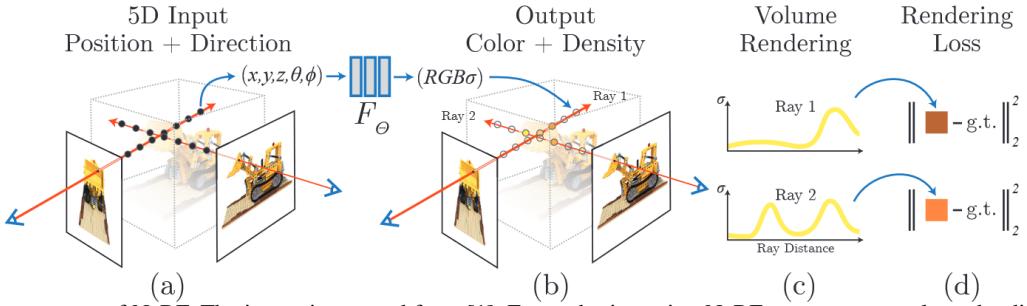


Fig. 2. The training process of NeRF. The image is sourced from [1]. For each viewpoint, NeRF assumes a ray along the direction connecting the camera origin and a pixel of the target image. Multiple points are sampled along this ray in the reconstructed scene. The 5D coordinates of these points (3D position + 2D orientation) are input into an MLP, which outputs their corresponding colour and density values. Next, the volume rendering is performed by integrating the colour and density of sampled points along a ray, producing the estimated colour of the target pixel. Finally, the difference between the estimated colour and the ground truth is used to update the entire network through the rendering loss. The NeRF network is trained through this iterative process.

robotics. Section V summarizes some of the challenges and future directions for NeRF in the field of robotics as the references for researchers. Finally, Section VI concludes this survey.

II. BACKGROUND

A. NeRF Theory

NeRF [1] represents a scene as a 5D vector-valued function which is approximated by a Multi-Layer Perceptron (MLP) $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. The input to the network is a 5D vector (x, y, z, θ, ϕ) which includes a 3D coordinate vector $\mathbf{x} = (x, y, z)$ and a 2D viewing direction vector $\mathbf{d} = (\theta, \phi)$. The output of the network is an RGB colour value $\mathbf{c} = (r, g, b)$ and a volume density σ . NeRF generates target images via volume rendering. The whole network is optimized by updating the weights Θ of the network via comparing rendering images and ground-truth images during training.

The training process of NeRF is shown in Fig. 2, which is divided into four parts:

- (a) NeRF assumes that there exist rays from the camera origin towards each pixel of the picture and through the scene. A set of points are sampled along each ray. The 5D coordinates (3D position + 2D orientation) of such sample points are input into the network.

- (b) The network outputs the volume density σ and color \mathbf{c} of the sampled points. The volume density σ is only related to the position, while the colour \mathbf{c} is related to both the position and the viewing direction.
- (c) Volume rendering generates the colour of the target pixel by integrating the colour and density with weights for sample points on the whole ray.
- (d) The rendering loss is the cross-entropy between the estimated colour and the ground truth colour of the target pixel. This loss is minimized to optimize the network.

Specifically, volume rendering is an integration process. This process is performed by integrating the colour and density of all sampled points on a ray to a pixel on the target image along the viewing direction \mathbf{d} :

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \quad (1)$$

where t_n and t_f are near and far bounds of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$. $T(t)$ is calculated as the transmittance that the ray can travel from t_n to t :

$$T(t) = \exp \left(- \int_{t_n}^t \sigma(\mathbf{r}(s)) ds \right). \quad (2)$$

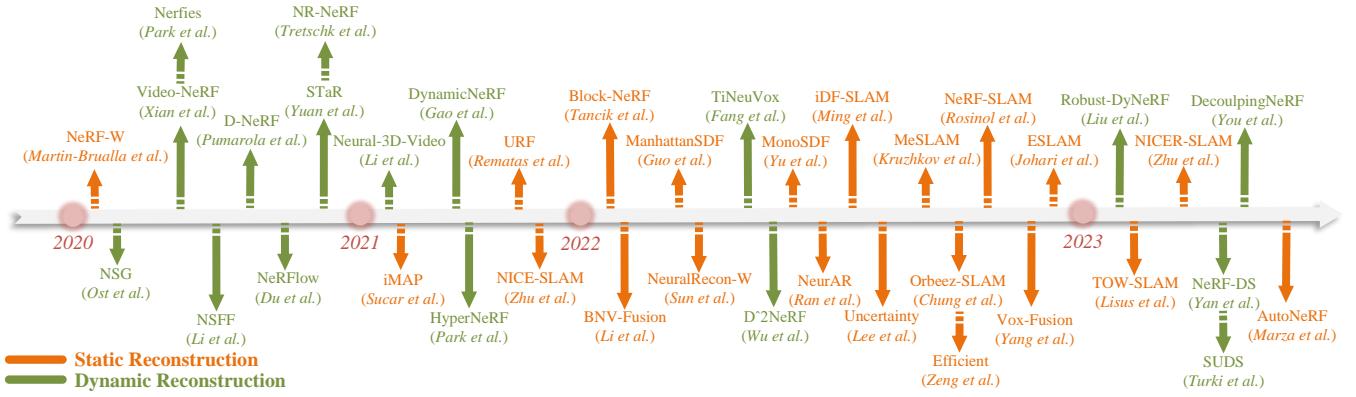


Fig. 3. Chronological: NeRF for Scene ReconstructionIII-A1.

Because of sampling points discretely, NeRF discretizes the above ideal continuous integration process as follows:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i \alpha_i \mathbf{c}_i, \text{ where } T_i = \exp \left(-\sum_{j=1}^{i-1} \sigma_j \delta_j \right), \quad (3)$$

where alpha values $\alpha_i = (1 - \exp(-\sigma_i \delta_i))$. $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples.

Based on the above design, NeRF applies two more techniques: Positional encoding is used to improve quality, and hierarchical volume sampling is used to improve efficiency.

Positional encoding rebuilds $F_\Theta = F'_\Theta \circ \gamma$. $\gamma(p)$ maps the input vector into a high-dimensional space to better represent the high-frequency changes in colour and geometry of the scene:

$$\begin{aligned} \gamma(p) = & (\sin(2^0 \pi p), \cos(2^0 \pi p), \\ & \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p)), \end{aligned} \quad (4)$$

where L is a hyperparameter. $L = 10$ for $\gamma(\mathbf{x})$ and $L = 4$ for $\gamma(\mathbf{d})$ in NeRF [1]. Note that $\gamma(\mathbf{x})$ is injected into the network at the beginning of MLP, and $\gamma(\mathbf{d})$ is injected close to the end, which is proved to avoid degenerate solutions [18].

Hierarchical volume rendering is a coarse-to-fine strategy where N_c points are first coarsely sampled to render. The rendered result then guides fine sampling to sample N_f points. The goal is to sample those points that contribute more to the target pixel.

Finally, the loss function based on hierarchical volume rendering is determined as:

$$L = \sum_{\mathbf{r} \in R} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right], \quad (5)$$

where R is the set of rays, and $C(\mathbf{r})$ is the ground truth, and $\hat{C}_c(\mathbf{r})$ and $\hat{C}_f(\mathbf{r})$ are predicted coarse volume and fine volume.

III. APPLICATION OF NERF IN ROBOTICS

NeRF's advantages [1], such as its capability to enable simplified mathematical models, compact environment storage, and continuous scene representations, make it an appealing tool for robotics applications. These properties play a crucial role in achieving scene understanding in robotics and in completing specific tasks through interaction with the environment.

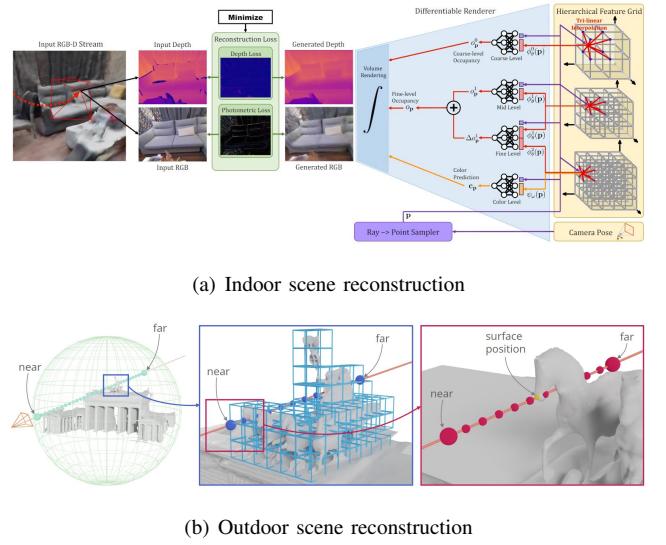


Fig. 4. An illustration of NeRF for static reconstruction. Fig. 4(a) and Fig. 4(b) are originally shown in [7] and [19], respectively.

A. Scene Understanding

1) *Reconstruction*: We categorize the related work into static and dynamic reconstruction and present them using a timeline, as illustrated in Fig. 3.

(a) *Static Reconstruction*: Scene reconstruction in robotics refers to the process of modelling a 3D representation of the surroundings by analyzing perceived sensor data. The differences in the attributes of indoor and outdoor scenes present distinct challenges in reconstruction tasks. Therefore, we divide our discussion into *indoor scene reconstruction* and *outdoor scene reconstruction* as in Fig. 4.

Indoor scenes exhibit bounded scope, rich textures, and clear structures. iMAP [20] attempts a combination of MLP structure and volumetric density representation, like NeRF [1], in the context of Simultaneous Localization and Mapping (SLAM) tasks. Through well-designed strategies based on loss-guided sampling and the construction of a replay buffer, iMAP achieves satisfactory SLAM results solely from 2D images. However, the limited capability of the MLP structure results in catastrophic forgetting and time-consuming inference issues, constraining the scale and efficiency of reconstruction. Volume density is a probabilistic representation, having

the appearance-geometry ambiguity [18], resulting in low-precision reconstruction results.

To expand the scale of reconstruction, MeSLAM [21] employs a multi-MLP structure to represent different parts of the scene. NICE-SLAM [7], on the other hand, designs a coarse-to-fine feature grid to extend iMAP’s single-room reconstruction to multi-room. Vox-Fusion [22] uses a tree-like structure to store grid embeddings, allowing the dynamic allocation of new spatial voxels as the scene expands. Lisus et al. [23] demonstrate that depth uncertainty and motion data can be utilized to enhance the accuracy of SLAM and a spherical background model can extend the scene scale. To enhance reconstruction efficiency, ESLAM [24] replaces feature grids with multi-scale axis-aligned perpendicular feature planes, reducing the growth of scene scale from cubic to quadratic. Orbeez-SLAM [25] and NeRF-SLAM [26] utilize existing SLAM odometry modules for localization, improving efficiency.

Different from volume density, the Truncated Signed Distance Function (TSDF) is the distance between sample points and the nearest surface, contributing to a clear geometric surface. Reconstruction techniques based on TSDF find the zero-level set of TSDF and have the ability to naturally depict scene surfaces with sharpness and accuracy [27], [28]. However, the classical volume rendering formula is not directly applicable to TSDF. Fortunately, some recent rendering techniques are available that can adapt to TSDF representations [29]–[33]. In conjunction with these advancements in rendering techniques, MonoSDF [34] integrates a general pre-trained monocular geometric prediction network, which predicts depth and normals as geometric priors, into neural implicit SDF surface reconstruction. Guo et al. [35] improve the SDF reconstruction quality of low-texture areas in indoor scenes by leveraging the Manhattan assumption and semantic guidance. BNV-Fusion [36] introduces a bilateral neural volumetric fusion algorithm that combines depth image features extracted at both local and global levels. The global geometry is supervised using the SDF loss. IDF-SLAM [37] utilizes a pre-trained feature-based neural tracker [38] and a neural implicit mapper, where the mapper specifically learns the TSDF representation. Vox-Fusion [22] employs voxel feature embedding as input, generating RGB and SDF values as output. NICER-SLAM [39] replaces occupancy with TSDF in NICE-SLAM [7] to achieve improved performance.

Unlike indoor scenes, outdoor scenes have additional challenges, like varying illumination and large scale. Sun et al. [19] employ appearance embeddings, like NeRF-W [40], to model appearance variation and propose a combination of voxel-guided sampling and surface-guided sampling to improve efficiency in large-scale scenes. Block-NeRF [6] employs multiple concatenated bounded blocks to cover long-distance streets with complex intersections, and the contribution of these blocks to rendering the target novel view is determined by learned visibility. Rematas et al. [41] fuse LiDAR data to image data and introduce a series of LiDAR-based losses to enhance reconstruction quality.

Recently, active scene reconstruction techniques based on the NeRF architecture have also made some progress. Active

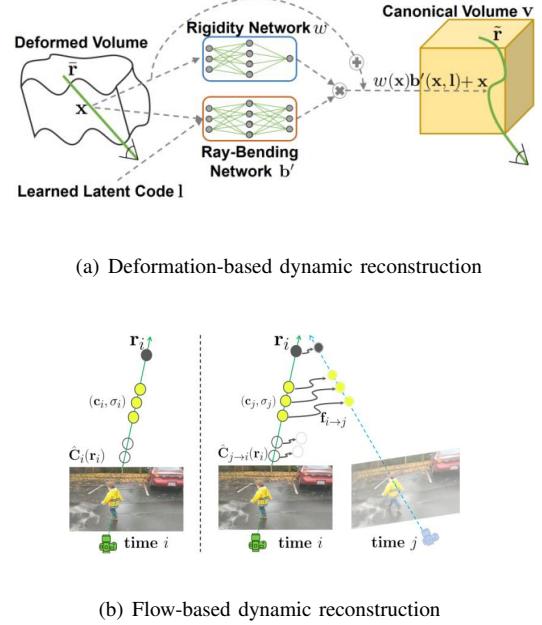


Fig. 5. An illustration of NeRF for dynamic reconstruction. Fig. 5(a) and Fig. 5(b) are originally shown in [46] and [47], respectively.

scene reconstruction technologies aim to explore methods for empowering robots to actively select data that maximizes benefits, thereby achieving a more intelligent reconstruction process. Lee et al. [42] select the next observation view that can most effectively reduce uncertainty by estimating the volume uncertainty. In NeurAR [43], pixel colours are modelled as random variables following a Gaussian distribution, which describes the uncertainty. The uncertainty is directly associated with the Peak Signal-to-Noise Ratio (PSNR) metric and can be used as a proxy to measure the quality of candidate viewpoints. Zeng et al. [44] plan a reconstruction path based on information gain, which is evaluated by comparing the current viewpoint with the 3D reconstruction obtained so far. AutoNeRF [45] utilizes a modular policy exploration approach to learn robotic autonomous data collection strategies, with scene semantics as the evaluation criterion.

(b) Dynamic Reconstruction: Long-term running robots usually face dynamic changes in complex environments. For the vanilla NeRF model based on static scene assumptions, dynamics undoubtedly disrupt the learning process, causing artefacts. Moreover, each moment only contains one observation in dynamic scenes, which makes the presentation severely lack spatial consistency constraints from different views. Therefore, how to learn NeRF-based models in dynamic environments is crucial. Related works are as illustrated in Fig. 5.

In the initial exploration phase, dynamics are expected to be represented end-to-end by adding additional conditions, like time or tracking pose transforms. STaR [48] models a rigidly dynamic NeRF to represent a single moving object within a scene and optimizes time-dependent rigid poses to track the motion. To build the dynamic field, Xian et al. [49] expend

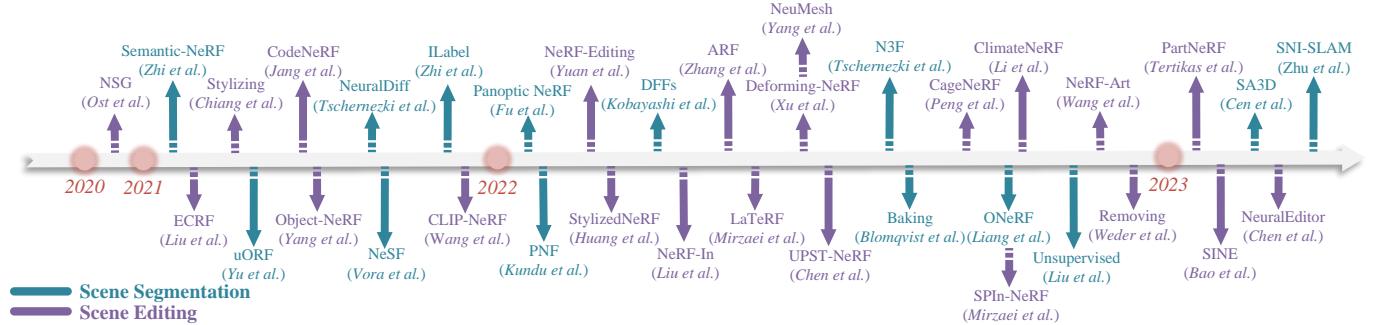


Fig. 6. Chronological: NeRF for Scene Segmentation and EditingIII-A2.

the original 3D spatial coordinates to 4D spatiotemporal coordinates. DyNeRF [50] uses time-dependent coding instead of time as a condition for the dynamic field, enhancing the representation capability for topological variations and transient effects. Ost et al. [51] build a dynamic scene representation based on a graph structure. Each leaf node corresponds to a different local radiance field. Moreover, objects of the same category share weights of the local radiance field.

As the exploration progresses, certain carriers of dynamic representations, such as deformation and flow, are employed to interpret dynamics, enhancing the overall performance.

These deformation-based works [46], [52]–[58] represent motion as deformations of the observed space relative to a multi-frame consistent canonical space represented by a static field. The calculated deformations by deformation fields finely reflect local changes in the scene, including non-rigid deformations. D-NeRF [52] defines the canonical space based on the first frame. The deformation network, conditioned on time, learns the displacements of ray sampling points in the observed space relative to the canonical space. The canonical space in NR-NeRF [46] is not pre-defined but rather learned based on all observed frames. In addition, NR-NeRF employs time-based implicit encoding instead of directly inputting time for better rendering quality. NeRFies [53] utilizes a dense $SE(3)$ field to describe deformations instead of a displacement field and introduces elastic energy constraints to mitigate optimization ambiguity caused by motions. HyperNeRF [54] represents the scene in a hyper-space for topological variations, where each frame observation corresponds to a 3D NeRF as a slice of the hyper-space. Based on HyperNeRF, NeRF-DS [55] alleviates the under-parameterization problem of reflecting colours for dynamic specular objects by adding the surface positions and rotated surface normals of objects as conditions in the colour output branch. To further enhance the quality of deformation-based dynamic scene representation, D²NeRF [56] introduces a shadow field to learn a shadow ratio for the static NeRF for rendering shadow variations. RoDynRF [58] learns deformation NeRFs while jointly estimating camera poses and focal lengths, achieving tracking in dynamic scenes that are difficult to attain with the classical method COLMAP [59]. TiNeuVox [57] employs an explicit structure of time-aware neural voxels to improve efficiency, replacing the time-consuming feature inference process with a querying process.

Different from deformations, flow is more commonly used to reflect the overall motion of objects in the scene, where

some works [47], [60]–[62] use scene flow, while one work [63] uses velocity flow. NSFF [47] predicts scene flow and occlusion weights between the current frame with both forward and backward frames. Gao et al. [60] separately model static NeRF and dynamic NeRF based on foreground masks. The dynamic NeRF predicts forward and backward scene flows while predicting a blending weight for mixing the results of dynamic and static NeRFs. SUDS [61] models static NeRF, dynamic NeRF, and far-field NeRF to adapt to dynamic large-scale urban scenes. The dynamic NeRF estimates 3D scene flow, which is projected onto the image plane and supervised by 2D optical flows predicted by DINO [64]. To break free from reliance on pre-processed 2D optical flow, You et al. [62] propose the surface consistency constraint and the patch-based multi-view constraint as unsupervised regularization terms to learn decoupled object motion and camera motion. Unlike scene flows, Du et al. [63] predict velocity flows of sampled points, which are then integrated to predict future spatial positions of points in upcoming frames.

2) *Segmentation & Editing:* The timeline statistics for the Scene Segmentation and Editing are illustrated in Fig. 6.

(a) *Scene Segmentation:* Scene segmentation refers to the process of partitioning perceived scenes into distinct components on purpose-specific rules. In comparison to 2D segmentation, 3D segmentation better fulfils the operational requirements of robots in real-world settings. NeRF presents an innovative approach to supervise 3D segmentation from 2D posed images. Based on segmentation goals, the related works are categorized into three groups: *object segmentation*, *semantic segmentation*, and *panoptic segmentation*, as illustrated in Fig. 7.

Object segmentation aims to precisely distinguish between the foreground and background within a scene, with the outcomes often utilized for object modelling or integrating a new background for novel view rendering. In this context, uORF [67] leverages object-centric latents extracted from a single image as conditions to train a shared conditional NeRF without supervision, allowing control over rendering results, such as segmentation and composition. ONeRF [68] achieves unsupervised object segmentation by iteratively utilizing feature clustering and 3D consistency of NeRF to generate accurate masks. Kobayashi et al. [69] and N3F [70] utilize teacher-student distillation techniques, where semantic attributes extracted by a 2D teacher network (such as CLIP [71], LSeg [72], or DINO [64]). SA3D [65] combines the segmentation capability of

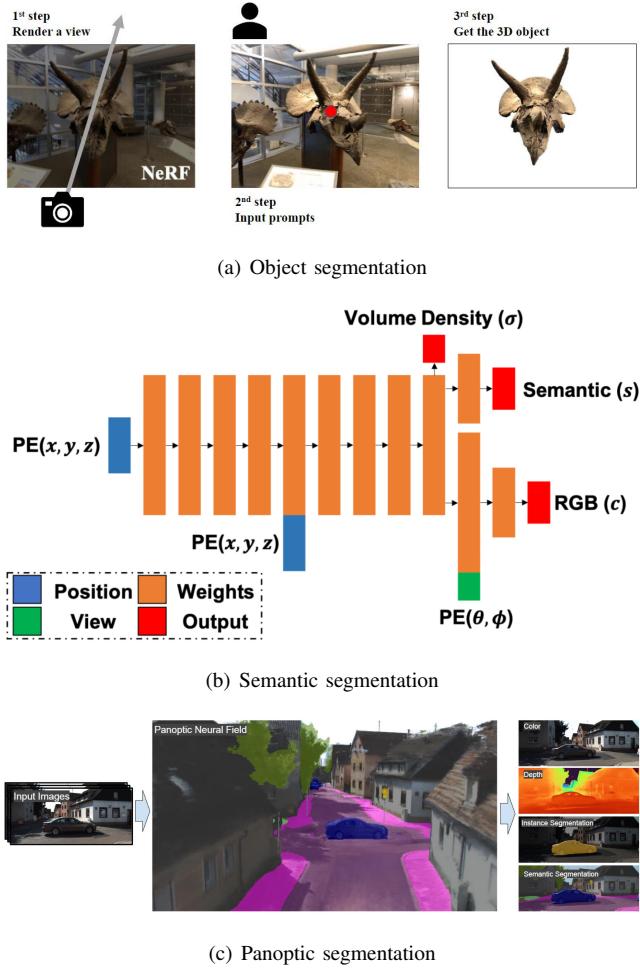


Fig. 7. An illustration of NeRF for scene segmentation. Fig. 7(a), Fig. 7(b), and Fig. 7(c) are originally shown in [65], [66], and [13], respectively.

SAM [73] with the 3D mask propagation capability of NeRF to segment desired 3D models. The mask inverse rendering and cross-view self-promoting progress are performed in different novel views until a detailed 3D object mask is generated. To complete object segmentation of egocentric videos, NeuralDiff [74] employs appropriate inductive biases and utilizes a triple-stream neural rendering network to segment the background, foreground, and actor.

Semantic segmentation divides the scene into different components by assigning a semantic label to each 3D point. Semantic-NeRF [66] integrates an additional semantic head alongside colour and density heads, allowing for the estimation of semantics at sampled points. To achieve generic semantic segmentation capability, NeSF [75] trains a multi-scene shared 3D UNet [76] to encode the pre-trained density field of NeRF, along with training a semantic MLP to decode features into semantic information. The generalization is achieved through training on extensive datasets with semantic labels, which places high demands on the quality of those labels. To reduce the reliance on precise pixel-level semantic labels, iLabel [77] and Blomqvist et al. [78] introduce methods for semantic segmentation using only sparse semantic labels from users.

iLabel [77] integrates a semantic prediction branch on top of iMAP [20] to achieve online interactive 3D semantic SLAM. Blomqvist et al. [78] improve the quality of upstream features by baking pre-trained feature extractors on a large amount of data. Liu et al. [79] design a self-supervised semantic segmentation architecture, which includes a semantic segmentation model trained continuously across scenes and semantic-NeRF models [66] for each scene. The segmentation model provides pseudo ground truth for the semantic-NeRF models, and consistencies of the semantic-NeRF models are used to refine semantic labels to further enhance the segmentation model iteratively. SNI-SLAM [80] integrates multi-level features from colour, geometry, and semantics by feature interaction and collaboration, achieving more accurate results, including colour rendering, geometry representation, and semantic segmentation.

Panoptic segmentation can be understood as a combination of instance segmentation and semantic segmentation [81], [82], where all instances are segmented while assigned semantic labels and instance labels. Panoptic NeRF [83] is designed for outdoor driving scenes (e.g., KITTI-360 [84]), assuming available 2D pseudo semantic labels and 3D bounding primitives. Panoptic NeRF [83] builds dual semantic fields, where the fixed semantic field enhances the geometry estimation, and the learnable semantic field improves the semantic estimation. Additionally, 3D bounding primitives provide extra 3D semantic supervision to reduce noise interference in pseudo labels and provide instance labels. PNF [13] uses instance-specific small MLPs to represent individual objects in the foreground, replacing a shared MLP with object encodings as input. This approach enables the semantic prediction and estimation of object pose transformations to track the object motions. Each object is individually represented, with naturally segmented instances within the scene combined with semantic segmentation results to achieve panoptic segmentation.

(b) Scene Editing: Scene editing refers to the process of modifying scene content based on the prompts provided by the user to achieve the desired effects. The edited scenes can serve as a source of training data for robots, and these data are often hard or time-consuming to collect in the real world. NeRF plays a crucial role in enhancing the reality and 3D consistency of the edited results. We categorize related works into *object appearance and geometry editing*, *object insertion and erasure editing*, and *scene stylization editing*, depending on the editing objectives, as illustrated in Fig. 8.

To achieve appearance and geometry editing, a common approach is to construct appearance and geometry encodings as inputs to conditional NeRF. It is worth noting that to avoid mutual interference between appearance and geometry editing, both conditions should be disentangled. To this end, CodeNeRF [88] learns to disentangle object shape and appearance encodings as conditions while learning NeRF weights. CodeNeRF achieves editing by adjusting ideal encodings. In addition to modifying the corresponding encodings, EditNeRF [85] simultaneously updates the weights of specified layers. Without being confined to a single prompt model, CLIP-NeRF [89] leverages the multi-modal capability of the CLIP model [71] to control the generation of appearance and geometry

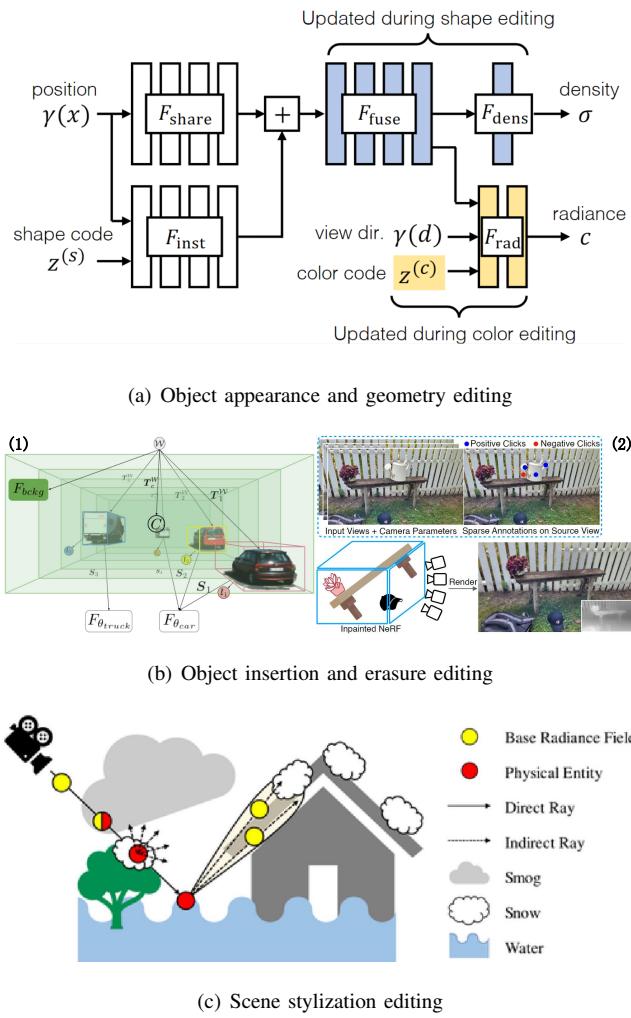


Fig. 8. An illustration of NeRF for scene editing. Fig. 8(a) and Fig. 8(c) are originally shown in [85] and [86], respectively, and Fig. 8(b) in order (1) and (2), sequentially correspond to [51] and [87].

condition biases using text prompts or image exemplars. SINE [90] employs a prior-guided editing field to adjust spatial point coordinates and colours for semantic-driven editing. To achieve local editing of objects, PartNeRF [91] assigns a NeRF representation defined in a local coordinate system to each part. Each NeRF is controlled by a partial encoding mapped from the overall shape and appearance codes.

The works mentioned above have effectively demonstrated the realism of implicit representations in editing tasks. However, it is challenging to achieve precise geometry editing using only implicit representations. Integrating implicit representations into the framework of explicit models is a promising direction that can mitigate this issue. Xu et al. [92] and CageNeRF [93] both assume that a coarse polygonal mesh cage encloses objects. Xu et al. [92] achieve deformation by manipulating the cage vertices to depict the transformation, whereas CageNeRF [93] learns a network that takes the original cage and a novel pose as inputs to generate the deformed cage. NeRF-Editing [94] utilizes the classical mesh deformation technique [95] to allow users to explicitly edit the mesh representation generated based on the density field of

the canonical NeRF. The edited results are used to calculate the deformation of the canonical space for rendering novel views. NeuMesh [96] employs a mesh-based representation, where learnable geometry and appearance encodings, as well as sign indicators that help identify positions, are stored at vertexes of the mesh. Geometry and appearance are edited by editing the mesh vertexes and updating the encodings with corresponding decoders. NeuralEditor [97] introduces a point-cloud-guided NeRF model based on a K-D tree structure and allows for editing by manipulating the point cloud. In this context, geometric editing is explained as the movement of each point in the point cloud to its final position. Simultaneously, Infinitesimal Surface Transformation (IST) is introduced to redirect the viewing direction of each point to ensure the correct direction-appearance correspondence.

Object insertion and erasure editing involve freely adding new objects or removing existing ones from a scene while maintaining scene coherence and harmony. Ost et al. [51] achieve object insertion and erasure by adding and deleting related leaf nodes in the scene graph. LaTeRF [98] extracts interesting objects by introducing an additional output head to regress the probability of each point belonging to interesting objects. For occluded components, LaTeRF utilizes CLIP [71] to fill by combining semantic priors. Yang et al. [99] construct a framework including a scene branch and an object branch while maintaining a library of object activation codes. During the rendering, Yang et al. select and switch the corresponding codes at the target position to control object movement, insertion, and erasure. NeRF-In [100] updates a pre-trained NeRF model to achieve object erasure by using edited RGB-D priors guided by user-drawn erasure masks. SPI-NeRF [87] additionally utilizes a semantic NeRF model to refine the erasure masks for achieving globally consistent object erasure. On the other hand, Weder et al. [101] introduce confidence to the RGB-D views guided by masks, choosing the views that ensure accurate painting and multi-view consistency for the training of object erasure NeRF.

Stylization editing produces various stylistic scene data in response to style prompts. This can reduce overall periods of data collection and enhance the robustness of trained systems. ClimateNeRF [86] achieves realistic rendering in different climate styles, such as fog, snow, and flooding, by combining the instant-NGP framework [102] with physics simulation techniques. Moreover, while these works [103]–[107] primarily emphasize artistic stylization, it is worth exploring relevant adaptations to generate style-specific data for robots.

B. Scene Interaction

Navigation and manipulation are typical scenarios in which robots interact with the environment or humans. The timeline for related work is depicted in Fig. 9.

1) Navigation: The core components of navigation include localization and path planning. Localization addresses the question of where the robot is, while path planning addresses how the robot goes to the destination.

(a) Localization: Localization involves estimating the 6-degree-of-freedom pose (position and orientation) through

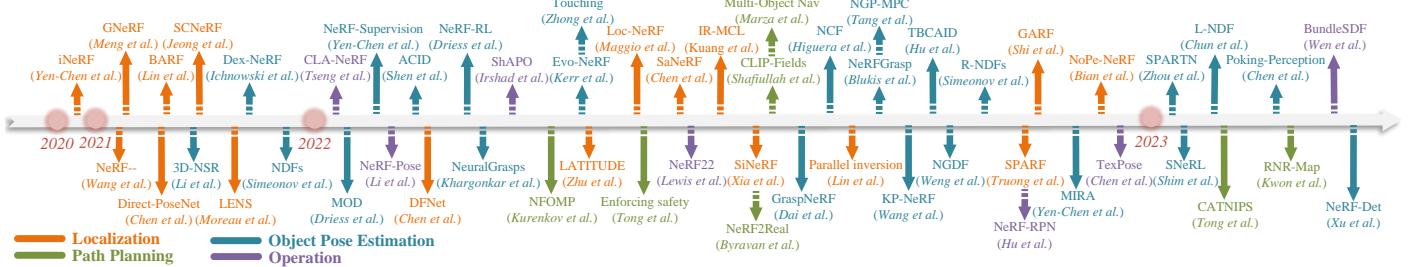


Fig. 9. Chronological: NeRF for Robotic NavigationIII-B1 and ManipulationIII-B2.

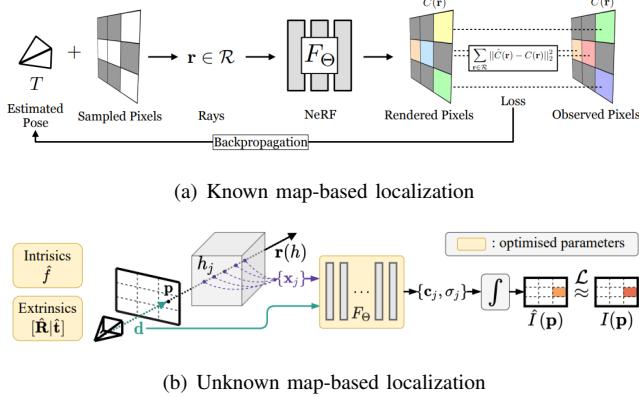


Fig. 10. An illustration of NeRF for Robotic Localization. Fig. 10(a) [108] makes a trained NeRF as the map, and Fig. 10(b) [109] optimizes the camera pose and the model properties jointly.

sensor data analysis. Based on the presence or absence of a prior environment map, these localization approaches can be categorized into two classes: *Known Map-based Localization* and *Unknown Map-based Localization*, as shown in Fig. 10.

In the context of NeRF-based known map-based localization, the maps typically involve pre-trained NeRF or extended NeRF models. iNeRF [108] represents a milestone work as it is the first to regress camera poses using the implicit representation of NeRF. iNeRF introduces an inverse NeRF architecture and uses pixel-level photometric loss to optimize initial rendering poses based on the trained NeRF model. Next, Direct-PoseNet [110] leverages a NeRF model to generate training data for Absolute Pose Regression (APR) networks. LENS [111] positions multiple virtual cameras in high-density areas identified by the NeRF-W model [40] to expand the training data space for APR models. To enhance the localization performance of drones in city-scale scenes, LATITUDE [12] initially estimates coarse poses using an APR network trained by pose-image data generated from the pre-trained Mega-NeRF [112] and then refines the coarse poses based on the inverse NeRF architecture. DFNet [113] optimizes an APR network to enhance robustness to illumination changes by minimizing the matching error between feature maps generated by histogram-assisted NeRF and feature maps extracted by feature extractors.

Another category of methods [9], [114], [115] achieves global robot localization in implicit scene maps by combining the traditional Monte Carlo localization algorithm [116].

These methods define pose estimation as a posterior probability estimation problem, modelling the posterior probability distribution as the distribution of weighted spatial particles. They iteratively update particle weights and perform particle resampling based on the difference between perception and the map until convergence to the correct pose. IR-MCL [114] trains a neural occupancy field as the scene map and updates particle weights by comparing rendered 2D LiDAR scans with real LiDAR scan data. Loc-NeRF [9] directly learns a general NeRF model as the map and calculates particle weights using photometric differences. Lin et al. [115] implement parallel processing of multiple Monte Carlo sampling processes based on the Instant-NGP model [102] to improve localization efficiency.

When robots explore a new environment, the absence of reference maps poses a significant challenge to localization. In addition to some of the works introduced in section III-A1, which can estimate the robot pose, some approaches estimate camera poses using NeRFs without explicit reconstruction.

NeRF— [109] jointly learns environment representation and camera poses from 2D images. BARF [117] draws inspiration from classical 2D image alignment methods and extends the alignment concept to 3D space. SiNeRF [118] leverages the inherent smoothness of SIREN-MLP [119], mitigating the risk of getting trapped in local optima. GARF [120] explores Gaussian activation functions, achieving higher pose estimation accuracy and improving the learnability of the network. GNeRF [121] employs the NeRF model as a generator and trains it using a GAN-based approach. The pose-image pairs generated by the trained NeRF are used to train an inversion network, which regresses coarse poses. These coarse poses are further refined through photometric losses. SCNeRF [122] jointly learns the scene model and the camera parameters through geometric and photometric losses. NoPe-NeRF [123] incorporates additional constraints by learning the undistorted depth maps. SPARF [124] introduces the multi-view correspondence loss and depth consistency loss. The multi-view correspondence loss enforces that corresponding pixels across multi-views must back-project to the same 3D spatial point. The depth consistency loss ensures the consistency between the trained viewpoint depths and the depths in unseen viewpoints warped from the trained viewpoint.

(b) *Path Planning*: The geometry learned by the NeRF model indicates space occupancy, enabling the direct integration of classical path-planning algorithms for navigation tasks in some works [8], [125], [126]. In pursuit of improved geo-

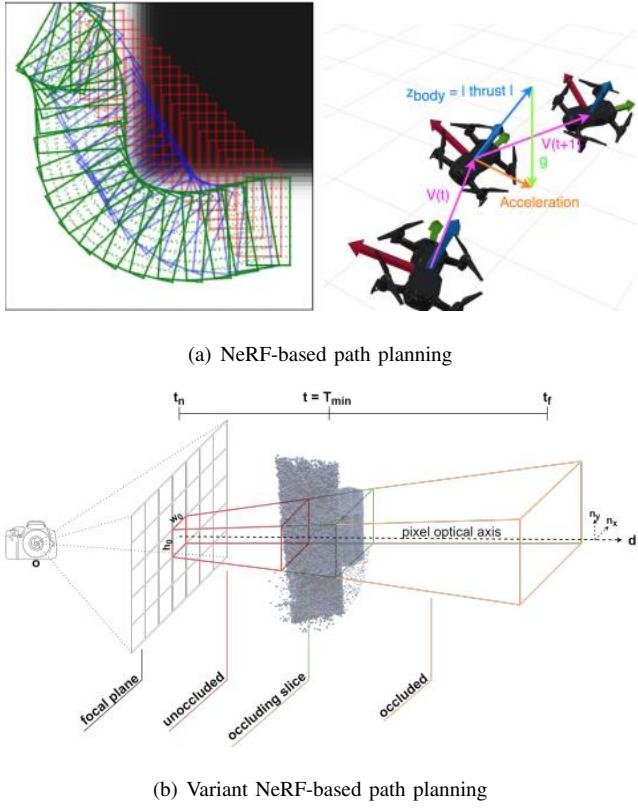


Fig. 11. An illustration of NeRF for Robotic Path Planning. Fig. 11(a) [8] shows planning a path avoiding the high-density area directly, and Fig. 11(b) shows a variant [128] that interprets density as the point density of a Poisson distribution.

metric interpretation compared to vanilla NeRF, some variants [10], [127]–[130] for navigation tasks are being explored. The basic idea of vanilla NeRF-based path planning and variants are as shown in Fig. 11.

NeRF-Navigation [8] achieves safe navigation within a NeRF map by penalizing collision behaviour between the point cloud model of the robot body and the density field. NFOMP [127] learns an obstacle neural field for obstacle avoidance while performing online trajectory optimization. Furthermore, Lagrange multipliers are added to deal with non-holonomic constraints. Tong et al. [125] utilize future visual predictions provided by the learned NICE-SLAM model [7] to implement safety robot control based on visual feedback Control Barrier Functions (CBF). To realize the deployment of navigation strategies in real-world scenarios, a robot simulation system, NeRF2Real [126], is introduced for training visual navigation and obstacle avoidance strategies by leveraging NeRF as a bridge between simulation and real-world settings.

Different from vanilla NeRF, some works extend the neural field to specially designed variant fields for path planning. CATNIPS [128] reinterprets the density field as a collection of points in continuous space that follow the Poisson distribution (i.e. Poisson Point Process), enabling the rigorous quantity of the collision probability. Kwon et al. [129] introduce a visual navigation framework that encompasses mapping, localization, and target searching. In this work, RNR-Map is

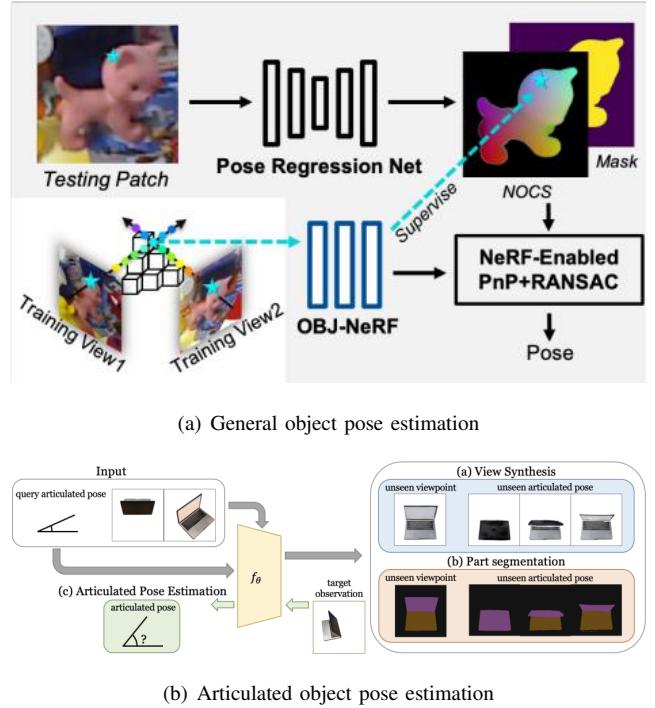


Fig. 12. An illustration of NeRF for Object Pose Estimation. Fig. 12(a) [131] estimates the general object poses. In Fig. 12(b) [132], the pose of the articulated object is estimated based on the specific connectivity properties.

proposed to encode visual information. The features stored in the RNR-Map can be transformed into local NeRFs, and the corresponding encoder-decoder network can be trained using an analysis-by-synthesis pipeline.

To fully leverage the semantic information, Shafullah et al. [10] develop CLIP-Fields to capture both visual and semantic information. CLIP-Fields establish a mapping from spatial positions to semantic embedding vectors. Using learned CLIP-Fields, robots can achieve semantic navigation guided by language instructions. Marza et al. [130] complete multi-object navigation using reinforcement learning (RL) by learning the semantic and structural neural implicit representations online. Semantic information is used to identify object locations, while structural information is utilized to avoid obstacles.

2) Manipulation: Manipulation typically involves using robotic arms or grippers to perform tasks, replacing human hands. In the context of manipulation, accurately estimating the pose of the object is crucial for determining the final state of the robot, such as grasp poses. Between the initial and final states, a series of intermediate states can be generated by operation methods.

(a) Object Pose Estimation: Unlike robot localization, which estimates a robot's 6D pose in the world, object 6D pose estimation refers to the robot inferring the 6D pose of objects in the environment based on visual data. Moreover, we separate the pose estimation of the articulated object from the general object pose estimation due to the specific physical structure, as illustrated in Fig. 12.

ShAPO [133] learns implicit SDF geometry and texture fields from a CAD model dataset to serve as a prior database

for supervising the learning of a single-shot detection and 3D prediction network. TexPose [134] generates a self-supervised dataset for training a 6D pose estimation network through synthetic data with perfect geometric labels and real data with realistic textures. NeRF is used to embed realistic texture information into the model. NeRF-Pose [131] follows the first-reconstruct-then-regress architecture and begins by constructing an OBJ-NeRF model, and then, the object 6D poses are regressed iteratively through a NeRF-Enabled PnP+RANSAC algorithm. Hu et al. [135] introduce NeRF-RPN, a universal framework for object detection extracting features from implicit NeRF models. The entire NeRF-RPN process eliminates the need for time-consuming 3D-to-2D rendering and applies to various feature extraction networks and RPN models. NeRF-Det [136] propose sharing geometric features of the NeRF branch and the 3D detection branch, leveraging NeRF’s multi-view consistency to achieve more accurate detection results. BundleSDF [137] constructs the neural object field simultaneously with the online optimization of the pose graph, estimating object 6D poses in real-time and ensuring the overall consistency of the entire 3D representation.

Due to the specific physical properties of articulated objects, pose estimation can take advantage of these properties. CLA-NeRF [132] additionally estimates the segmentation of different articulated components. Based on NeRF and articulated segmentation, CLA-NeRF can forward render images with novel articulated poses from the articulated deformation matrix and estimate the articulated pose from a given target image in inverse rendering. NARF22 [138] learns various articulating parts and combines parts based on a given configuration (i.e., articulating joint parameters). NARF22 similarly supports rendering images for novel articulating poses and estimating articulating configurations based on a given target image.

(b) *Operation:* The 3D structural bias of NeRF contains richer scene information compared to 2D perception methods and can be directly used for specific operational tasks when combined with some operation planning methods [11], [139]–[152]. With continuous exploration, some neural variant concepts and methods have extended the representation of vanilla NeRF, forming a more targeted expression for operation tasks [153]–[160], and achieving satisfactory performance. As shown in Fig. 13.

The most direct approach is to use NeRF to provide strong 3D scene priors for subsequent operation training. Hu et al. [11] learn a NeRF model of the target object without a known category to generate a large number of template images. These template images are then used to train a detecting network for manipulation. Chen et al. [139] propose to continuously poke the detected object with a robotic arm to obtain the complete visual perception for modelling an unknown target object. The constructed NeRF model is then used to train other pose estimation networks for manipulation. Tang et al. [140] utilize the mesh representation built from a fast NeRF model to compute SDF. Based on the mesh model, a sampling-based Model Predictive Control (MPC) algorithm is used to predict motion. Li et al. [141] train an encoder-decoder network to learn viewpoint-equivalent image states by employing time-contrastive loss and reconstruction loss.

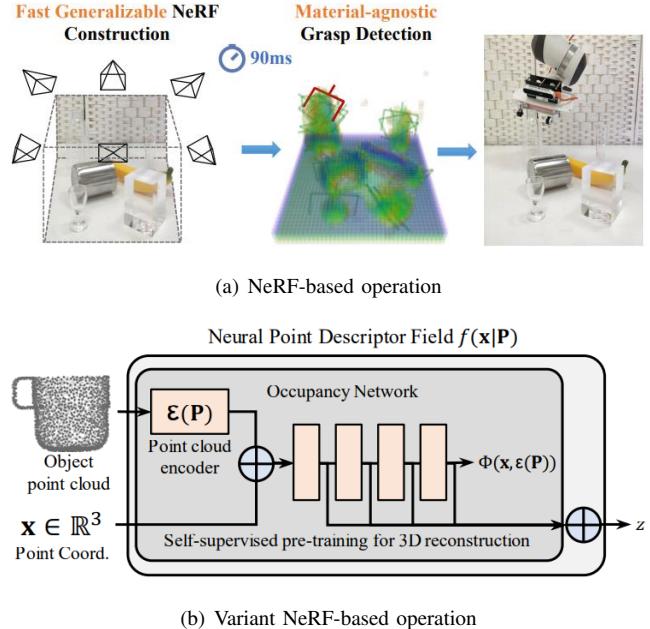


Fig. 13. An illustration of NeRF for Robotic Operation. Subplot(a) [147] illustrates a method that utilizes NeRF as a perceptual tool, Subplot(b) [153] extends neural fields’ boundaries to better serve operational tasks beyond radiance representation.

The viewpoint-equivalent image states can be used to train a motion prediction model to predict future states relevant to actions. Finally, the predicted future states are combined with MPC methods to learn visuomotor control strategies. Driess et al. [142] encode the implicit representation of each object in the dynamic scene. Next, a Graph Neural Network (GNN) is trained to predict the future states of the dynamic NeRF based on current encodings. KP-NeRF [143] incorporates the invariant relative positions between key points and query points as an additional condition to train a dynamic prediction model. MIRA [144] employs orthographic ray casting instead of perspective ray casting to render novel views with invariant object size and appearance for predicting operations by a learned action-value function. ACID [145] models the geometric occupancy of non-rigid objects implicitly based on images and predicts flow to represent dynamic deformations. Simultaneously, correspondence between various deformation states is learned by contrastive learning. Finally, a model-based planning approach is trained to acquire a set of actions by minimizing the cost function. Blukis et al. [157] add a prediction head to estimate the score of sampled grasping poses in the grasping pose space. This involves predicting reasonable grasping poses while rendering novel views of the object.

Moreover, NeRF demonstrates excellent performance in operating scenarios in which it is difficult for common sensors to perceive 3D structures. Dex-NeRF [146] leverages the volume density field of NeRF to capture globally consistent scene geometry, facilitating grasp planning for transparent objects. GraspNeRF [147] aggregates features and predicts TSDF values. Then, a grasp detection network is used to predict the grasping poses of objects, including transparent and

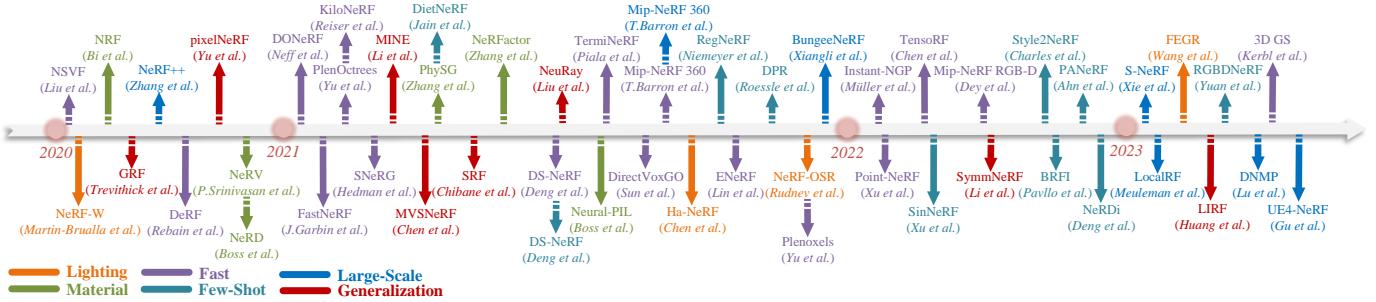


Fig. 14. Chronological: Advances of NeRF related to robotic applications.

specular objects, based on the predicted TSDF values. Evo-NeRF [148] modifies Instant-NGP [102] to support collecting data while training the NeRF model to adapt to continuous grasping operations. A radiance-adjusted grasp network is trained to calculate the grasping pose based on the rendered depth map of transparent objects. NeRF-Supervision [156] learns descriptors of thin and reflective objects from NeRF. The learned descriptors, which are helpful for operation, represent correspondences between object surface points across frames.

Surprisingly, NeRF not only serves as a visual perception tool but also presents application in tactile perception. Zhong et al. [149] train a Generative Adversarial Network (GAN)-based generative network to generate tactile images that represent touchings based on images rendered by NeRF. Higuera et al. [150] propose the Neural Contact Field (NCF) to predict the contact probability of the target object based on historical tactile perception data and the robot end-effector position during operations.

At the same time, the strong 3D structure bias of NeRF has been proven to significantly enhance the performance of reinforcement learning (RL) [151]. NeRF-RL [151] treats novel view rendering as a proxy task, offline training an encoder and a NeRF decoder. During online RL policy learning, the latent space generated by the encoder is treated as the state for action learning. Furthermore, SNeRL [152] enhances supervising the encoder not only with RGB information but also semantics. In addition, the encoder is jointly supervised by a self-supervised teacher network.

With the development of research, some extensions and techniques on neural fields have been proposed to enhance operational task performance. NDFs [153] learn a $SE(3)$ -equivariant and class-equivariant neural descriptor from object point cloud models. With few-shot imitation learning, robots are allowed to operate previously unseen objects of the same category. As a follow-up, the same team subsequently introduces R-NDFs [154] and L-NDFs [155]. The former extends NDFs to object rearrangement tasks, while the latter designs a more general neural descriptor of locally operable components, capturing similar operational priors across different object categories, and breaking category boundaries. Wang et al. [158] learn a neural grasp distance field to estimate the distance from a queried pose to the nearest valid grasp pose. The estimated distance is considered as part of the grasp cost. NeuralGrasps [159] introduces a novel implicit representation

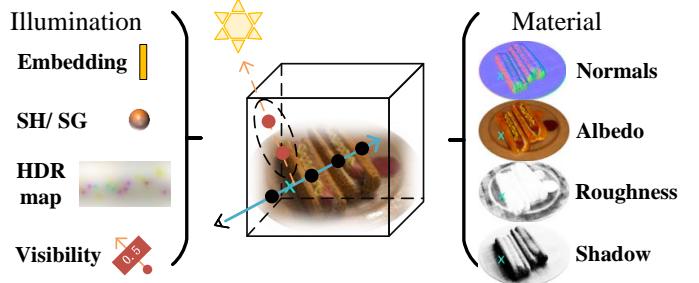


Fig. 15. Reality: Quality Improvement on NeRF Representation. SH: Spherical Harmonics, SG: Spherical Gaussians, HDR: High-Dynamic Range. The images utilized in the “HDR map” are sourced from [161], “Visibility” and “Materials” from [162]. The hotdog image is sourced from the NeRF synthetic dataset, and the hotdog images below are similar.

that establishes correlations between various robot grippers and even between robot grippers and human hands by learning similarity matrixes. SPARTN [160] adds noise perturbations to demonstration trajectories and generates perturbed trajectory-image pairs for offline data augmentation to improve the success rate and robustness.

IV. ADVANCE OF NERF IN ROBOTICS

We know that the vanilla NeRF is imperfect, and some novel variant models developed certain properties of NeRF [1] and allow for more effective applications in robotics. The timeline of collected works on enhancing NeRF properties related to robotic applications is presented in Fig. 14.

A. Reality

Reality is an important attribute of NeRF-based models. Vanilla NeRF interprets the imaging process as an integration of the space particle radiance, avoiding calculations of complex ray propagation and reflection. However, some flexibility is sacrificed, especially when dealing with scenes involving varying environmental lighting and different materials, as shown in Fig. 15.

1) Lighting: In the editing section III-A2, these methods [92]–[94], [96] face challenges in handling lighting and shadows, significantly impacting the realism of the edited scenes. This reminds us that accurate representation of lighting effects is crucial for realistic rendering.

To enhance the lighting representation capability, NeRF-W [40] introduces lighting embedding as an additional learnable

condition to represent the illumination. Ha-NeRF [163] additionally trains a CNN encoding network to regress the latent vector of appearance for each image, which serves as an input for the NeRF model. This ensures similarity in lighting while allowing for better generalization to new scenes. NeRF-OSR [164] learns spherical harmonics (SH) coefficients to represent illumination from a set of unstructured outdoor scene images. Moreover, NeRF-OSR employs separate shadow and albedo networks to learn environmental shadows and object albedo, respectively. For urban scenes, FEGR [165] learns the Neural Intrinsic Field (NIF) to model geometry, colour, and material properties. A High-Dynamic Range (HDR) sky dome is learned for lighting. During rendering, FEGR [165] introduces a hybrid rendering, including primary ray rendering based on the neural implicit model and secondary ray rendering based on an explicit mesh model derived from NeRF. The secondary ray rendering captures better lighting effects, such as highlights and shadows.

2) Material: Material properties, which belong to the object itself, typically encompass the reflective characteristics of surfaces within a scene, including diffuse reflection and specular reflection. These properties determine the effect of light after reaching the surface, including the generation of reflections and shadows.

Bi et al. [166] extend NeRF to Neural Reflectance Fields (NRF), where the model not only learns the radiance and volume density for each ray-sampled point but also learns reflective properties, including diffuse albedo and specular roughness that are typically represented by Bidirectional Reflectance Distribution Function (BRDF). NeRV [162] not only models a neural reflectance field to consider the reflective properties but also learns a neural visibility field to regress the visibility of the light source at sampled points. Visibility quantifies the propagation of light rays. Moreover, the direct inference of the visibility field avoids the expensive process of volumetric density inference and integration between light sources and sampled points. Similarly, Boss et al. [167] utilize illumination embedding to represent lighting and propose a pre-integrated light (PIL) network to decode lighting embeddings. This approach directly regresses lighting based on reflection properties at each point, replacing the integration process with a querying process. PhySG [168] uses Signed Distance Functions (SDF) to represent environmental geometry, Spherical Gaussians (SGs) for environmental illumination, and BRDF for object material. All parameters are jointly optimized based on photometric losses. Similarly, NeRD [169] models an explicit decomposition model, synchronously optimizing the shape, reflectance parameters represented by spatially-varying BRDF (SVBRDF), and illumination represented by spherical Gaussians. For unknown lighting conditions, NeRFactor [161] pre-trains additional prediction networks to reduce noise in normals and light visibility, typically calculated from density. NeRFactor [161] models illumination using an HDR light probe image and learns the reflection properties at surface points, including BRDF that absorbs reflection priors from real datasets and albedo for shadows.

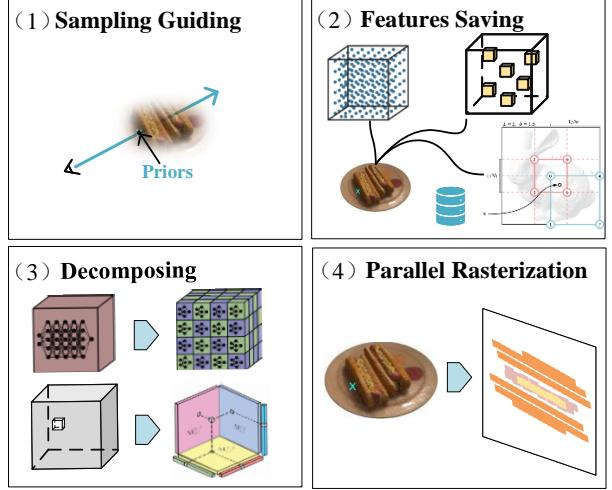


Fig. 16. *Fast*: Speed Improvement on NeRF Representation. Some pictures of subplot(2) are extracted from [170] and [102], while some pictures of subplot(3) are taken from [171] and [172].

B. Efficiency

The efforts made to enhance efficiency are categorized into two aspects in this paper, namely, *fast* and *few-shot*. The former emphasizes enhancements in runtime efficiency, while the latter focuses on ameliorating data utilization efficiency.

1) Fast: The time-consuming multi-point querying process based on the MLP network primarily constrains the speed of Vanilla NeRF. As depicted in Fig. 16, various acceleration strategies from different perspectives are employed to optimize or replace the time-consuming querying process.

NeRF employs a coarse-to-fine sampling strategy, and the sampling process remains a bottleneck for efficiency. To enhance sampling efficiency, these methods [173]–[175] train an additional sampling network to guide the sampling process. These works [174], [176]–[178] use depth as a geometric prior guiding ray sampling on the surface. ENeRF [178] leverages explicit geometry provided by Multi-View Stereo (MVS).

On the other hand, while the implicit representation of NeRF is storage-efficient, in the pursuit of speed advantages, the sacrifice of some storage is usually adopted to enhance efficiency. For instance, attribute parameters are pre-stored in an explicit structure or established tools based on explicit representations are utilized, such as CNNs. Sun et al. [179] combine an explicit voxel grid representation that allows efficient interpolation to model scenes. The strategy of interpolating first and then activating for computing α value in the formula (3) is found to expedite the acquisition of sharp surfaces by some experiments. Moreover, a coarse-to-fine staged approach is designed to skip over invalid regions, optimizing within effective areas. Baking-NeRF [180] compactly stores view-independent diffuse colours in a Sparse Neural Radiance Grid (SNeRG) for querying directly. NSVF [181] learns a set of voxel-bounded implicit radiance fields based on an explicit sparse voxel octree structure. Yu et al. [170] tabulate the density and spherical harmonics (SH) coefficients of the proposed NeRF-SH model and store them in each leaf of a PlenOctree for querying directly. Subsequently, Yu et al. pro-

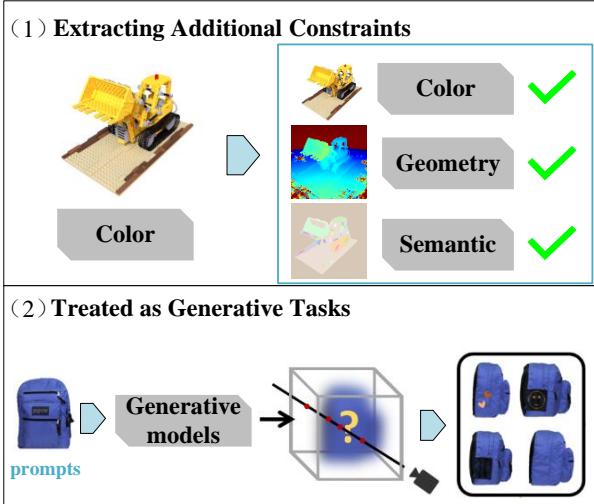


Fig. 17. Few-Shot: Image Utilization Efficiency Improvement on NeRF Representation. One category of approaches extracts additional constraints (such as depth or semantics), as shown in subplot(1), part images of which are taken from [187]. Another category transforms the task into a generative one, utilizing the limited views provided as prompts to guide the generation process, as shown in subplot(2), part images of which are taken from [188].

pose Plenoxels [182], an approach without neural components that explicitly learns occupancy and SH coefficients for each vertex in sparse voxel grids. Instant-NGP [102] constructs a hash table containing different resolution layers, which allows for rapid querying of features. Point-NeRF [183] utilizes pre-trained CNNs to infer and generate a neural point cloud containing scene features. The neural radiance field based on the neural point cloud can achieve desirable results with simple fine-tuning for specific scenes.

Another category of work focuses on efficiency improvement through decomposition. The key lies in decomposing the global, complex, or high-dimensional representation into local, simple, or low-dimensional components. DeRF [184] and KiloNeRF [171] use multiple smaller neural networks to replace a single large neural network, with each representing only a small portion of the scene. FastNeRF [185] calculates the inner product of the decomposed position and direction function results to obtain the final RGB values. TensoRF [172] utilizes tensor decomposition to break down the 4D scene tensor representation into the pointwise multiplication of several compact low-rank tensor components.

Lastly, significant efficiency improvements are achieved through enhancements in acceleration techniques and rendering methods. Kerbl et al. [186] employ a set of 3D Gaussians as the foundational units for scene representation to achieve more realistic rendering results. Sorting techniques and GPU acceleration are employed to ensure realism while enhancing speed. Additionally, a tile-based rasterizer is used to replace time-consuming ray marching rendering.

2) *Few-Shot*: The challenge of rendering a few-shot novel view lies in the scarcity of information available. In scenarios with only a few observations, vanilla NeRF either fails to converge or overfits to a smooth solution [189]. To achieve

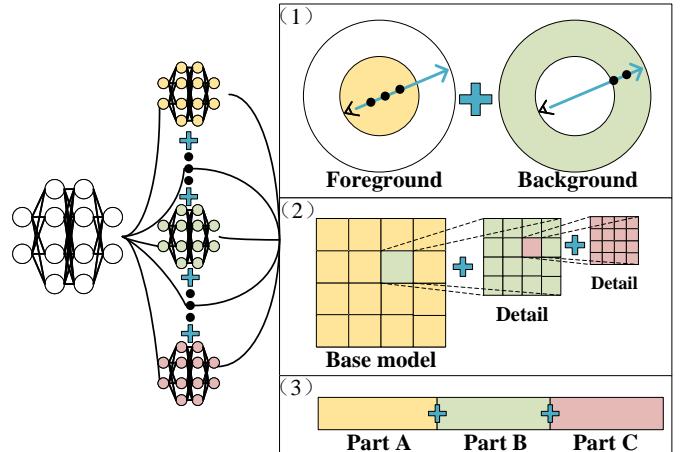


Fig. 18. Large-scale: Adaptability of NeRF to Large-Scale Scenes. Multiple models are employed to model different parts of a large-scale scene according to different rules.

an ideal model in a few-shot setting, additional constraint relationships need to be established, corresponding to the extraction of more valuable prior knowledge, as depicted in Fig.17.

In terms of leveraging geometry, RegNeRF [190] applies appearance and geometric regularization to patches rendered from unseen viewpoints. DS-NeRF [177] and Roessle et al. [191] use depth values generated during the Structure-from-Motion (SfM) process as guidance. Additionally, Roessle et al. further pretrain a depth completion network to densify the depth ground truth. In terms of leveraging semantics, DietNeRF [189] utilizes semantic priors provided by a pre-trained CLIP model to guide the learning process of the NeRF model. The semantic priors encourage high semantic similarity between different viewpoints of the same object. SinNeRF [187] relies on geometry and semantic information to generate a large amount of pseudo-label data from a single reference frame for training. PANeRF [192] warps reference frames to generate pseudo-views and combines the CLIP model to ensure semantic consistency at both local and global scales. Yuan et al. [193] generate pseudo-training data from a coarse mesh modelled from sparse RGB-D observations.

Some work approaches few-shot modelling as a generative task to achieve the desired results. NeRDi [188] leverages the generative capability of a language-guided diffusion model to transform the few-shot NeRF learning task into a generative task. Style2NeRF [194] and Pavllo et al. [195] transform the task of generating novel views from a one-shot image into a 3D perception-based GAN inversion task.

C. Adaptability

The unsatisfactory performance of Vanilla NeRF in large-scale and unseen scenes significantly constrains its adaptability when deployed on robots. Improving performances in both settings would greatly expand its applicability in diverse environmental contexts.

1) *Large-Scale*: In large-scale scenes, only a few viewpoints contain small areas of co-visible observations. Further-

more, distant details of objects are observed less thoroughly in unbounded scenes. Different parts are modelled separately according to different rules, as depicted in Fig. 18. This avoids a single model struggling to compromise across various parts and keeps results smooth.

To parameterize the distant part appropriately, NeRF++ [18] introduces an inverted sphere parameterization and builds a NeRF for distant elements additionally. Similarly, Mip-NeRF 360 [173] adjusts the boundaries of the cone sampling of Mip-NeRF [196], compressing Gaussian samplings outside the set sphere domain into the sphere. Building upon Mip-NeRF 360 and Mip-NeRF, S-NeRF [197] additionally propagates sparse in-vehicle LiDAR signals and constructs a confidence map to supervise the learning process. On the other hand, Mega-NeRF [112] adjusts the unit sphere domain to an ellipsoidal domain, obtaining a more effective bounding region. To address the constraints posed by the finite capacity of an individual neural network, BungeeNeRF [198] introduces a progressive neural network framework. During the modelling, additional residual blocks are continually added as more details are observed. LocalRF [199] designs a time-sliding window strategy for local NeRFs modeling. As the camera moves, the newly appearing contents are modelled by continuously adding local NeRFs. Connections between adjacent NeRFs are established based on their co-visible regions. Similar to Block-NeRF [6], UE4-NeRF [200] divides large scenes into different blocks, builds a NeRF model for each block, and integrates the mesh rasterization rendering pipeline of Unreal Engine 4 (UE4), allowing for real-time rendering. Lu et al. [201] design a novel neural mesh representation element called Deformable Neural Mesh Primitive (DNMP). Modelling the radiance field based on DNMP not only allows extending to large-scale scenes by using efficient rasterization-based rendering but also ensures satisfactory results.

2) *Generalization*: Vanilla NeRF memorizes a scene implicitly in a design that leads to the network overfitting the single scene, making it unable to perform well in unknown scenarios. Therefore, to achieve generalization, the network needs to learn a general ability to handle scene features rather than relying solely on memorization. As shown in Fig. 19.

PixelNeRF [202] and GRF [203] take extracted pixel-level features as an additional input to the neural network, training the network to learn a general feature processing capability rather than memorizing specific scenes. MINE [204] trains a generic encoder-decoder network, decoding the encoded features of source images plane by plane and regressing colour and volume density based on the multiplane images (MPI) structure of the camera frustum. SRF [205], inspired by classical Multi-View Stereo (MVS) methods, trains a radiance field decoder to infer the colour and geometry based on extracted features with high inter-image similarity. Huang et al. [206] propose a Local Implicit Ray Function (LIRF) based on cone sampling, considering the visibility of views. This method interpolates the local region features from the queried image corresponding to the eight vertices of the cone where the sampled point lies. SymmNeRF [207] additionally incorporates a hypernetwork that learns to regress NeRF weight parameters from global image features. Subsequently,

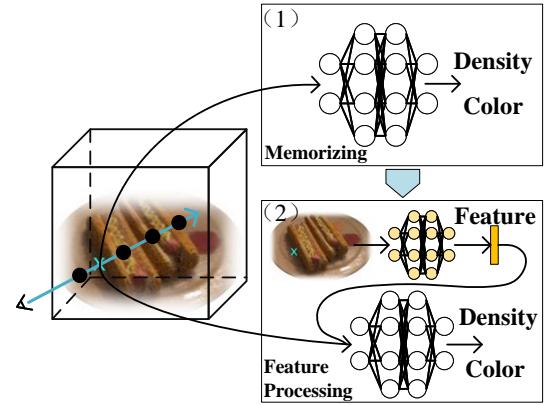


Fig. 19. Generalization: Adaptability of NeRF to Novel Scenes. The core of generalization is training a network to learn a general capability for processing scene features, replacing the learning of memorizing scenes.

the NeRF model utilizes both features of the sampled position and the corresponding symmetrical position to further refine the representation details. MVSNeRF [208] utilizes a generalized MVS-like framework. Firstly, MVSNeRF reconstructs a neural encoding volume using the common MVS method. Subsequently, MVSNeRF learns a rendering network to infer colour and density conditioned on extracted features from the encoding volume. NeuRay [209] predicts the visibility of features extracted using MVS-like methods, quantifying occlusion between different views. This allows for more rational utilization of the extracted features.

V. DISCUSSION

In this section, we present some challenges and analyze some valuable research directions based on these challenges for reference in this research community.

A. Map Fusion

As we know, robots usually move, and the surrounding environment will change as the robot locations and times change. Therefore, the robot needs to update the history map as the robot moves. In addition, for large-scale environments, it is more efficient to use multiple robots to build the 3D mapping jointly. Therefore, map fusion is an important issue for NeRF in the 3D mapping of robots.

Here, we definite two types of fusion: temporal fusion and spatial fusion. Temporal fusion is for changes in the same scene over time, including natural environmental changes and changes resulting from robot interactions, such as illumination variations at different times and changes in object positions from robot interactions. Spatial fusion involves merging NeRF scene maps in large-scale environments, allowing one robot to adapt flexible spatial ranges or multiple NeRF maps from multiple robots.

Temporal fusion focuses on accurately identifying the location of changes, such as these related to dynamic work [47]–[49], [51], [53], [60], [63], [210], [211], updating only the changed parts, and fusing historical maps with current observations. Generally, the content of a scene is unlikely to

change dramatically over a short period, so repetitive global reconstruction is not wise.

Spatial fusion, on the other hand, focuses on precise registration between two or more maps. Accurate and smooth registration involves 2D-2D, 2D-3D, and 3D-3D, and sometimes also includes the time dimension. In addition, there are cases where a robot's exploration is interrupted due to a malfunction. When it returns to the scene, it cannot maintain the same state as before. In such cases, we believe that a multi-scale fusion of historical information is necessary. Regarding map fusion, we have also considered the problem of information sharing among multiple robots in the exploration of an unfamiliar environment. Using multiple robots is one of the most direct methods for efficient exploration of a novel environment. However, the problem of how to fuse the environmental information obtained by different robots remains.

A well-performing spatiotemporal NeRF map fusion method offers accurate and rich map priors, guiding more robust robot actions.

B. Robot Relocation for Scene-level Environments

Once a complete NeRF map is available, the robot needs to use the map and observations to determine its current pose, similar to iNeRF [108]. However, this optimization method may struggle to work at the scene level due to the possibility of gradients being zero. To address this issue, we have two rough ideas.

Firstly, we believe that a multi-scale structure from coarse to fine is effective. Similar to our common sense, a rough initial pose can be obtained by finding a relatively similar area at a larger scale, followed by fine-tuning the pose at a smaller scale. Secondly, we believe that adding additional features as markers to the map and then robot observations could guide the optimization process in the correct direction.

Furthermore, this relocation should not only consider appearance information but should also avoid relocation failure due to changes in the scene through multi-information fusion, such as semantics and multiple sensors.

C. More Generalization Ability

We have introduced some generalization works [202], [204], [208], [209], [212] by learning rendering images conditioned on features that are extracted from an encoded neural network. However, this generalization is only applicable to special scenes similar to training data because of the generation limitation of the encoding network. There is still a distance to the ideal generalization for various scenarios because the real environment is characterized by diverse features, such as different mechanical properties (e.g., rigid bodies, soft bodies, fluids, etc.), geometry structures (e.g., chairs in square and cylindrical shapes), and complex illuminations (e.g., day and night).

We believe there are two directions for enhancing generalization ability based on feature processing. On the one hand, it is a good choice to use or finetune a large pre-trained feature model for multiple scenarios instead of learning a small feature neural network. The evolution of basic network

design allows for training networks with more neurons and deeper layers, utilizing larger-scale datasets. Training a large model enables it to extract high-level effective scene features from diverse scenarios, allowing for generalization to more realistic and complex environments. On the other hand, in contrast to combining large models, precise world physical mechanisms are worth studying for resource-saving small-scale specific networks, like this work [213]. Precise physical mechanisms can serve as the foundation for neural networks to extract desired features of different parts of scenes and integrate different characteristics into NeRF models to improve generalization in real scenarios.

D. Rendering to Real

The ability of NeRF to realistically construct scenes has great potential for generating training data and simulation environments for training robots. NeRF2Real [126] and RialTo [214] have made initial explorations in this direction. The acquisition of robot training data is crucial, especially for generating data from scenes that are difficult to collect in real scenes, such as abnormal vehicle driving data in autonomous driving scenarios or extreme environmental data where it is challenging for humans to operate (such as deserts, deep sea, space, etc.). Not adequately trained robots are prone to failure in corner cases and unfamiliar environments, leading to significant losses. Moreover, training robots in real environments is costly. Traditional environment modelling requires experienced professionals to simulate meticulously to obtain more realistic data, which is inefficient. Therefore, utilizing NeRF-based methods to render data and successfully transferring them to training real robots holds great value. However, this approach faces challenges such as the lack of physical realism and scarcity of learnable data in corner cases and extreme environments.

The lack of physical realism presents as the incorrect rendering of the detailed variations in real lighting and shadows. The scarcity of learnable data in corner cases and extreme environments presents the difficulty of predicting dynamic changes in physical dynamics interactions. To address these challenges, one direction is leveraging rich experience in computer graphics and utilizing virtual engine tools, which have the potential to bring about a qualitative leap. In addition, NeRF-based works about few-shot [187], [189], [190], [193], [215] using constraint mining methods have shown promising results in tackling these issues, which remains a promising direction for further exploration.

We also anticipate more exploration in combining generative models, such as GANs and diffusion models, which exhibit remarkable capabilities in generating ideal data under condition guidance. Furthermore, the generation capabilities of large models are tremendous – some language prompts alone can lead to the generation of images or videos [216]–[218]. The idea of combining NeRF with the generation capabilities of large models to directly create a 3D world is truly exciting.

E. Multi-modal Robot Interaction

In a realistic environment, robots are surrounded by multi-modal information, including colour, geometry, semantics,

voice, flavour, etc. The forms of perceiving multi-modal information are diverse, such as vision, smell, hearing, touch, taste, etc. The NeRF and its extensions primarily focus on visual perception to understand scenes in terms of radiance and geometric information, with some exploration into semantics as well [66], [75]–[80]. A few works have conducted preliminary explorations into auditory and tactile modalities, such as AD-NeRF [219] encoding audio signals from videos for talking head video generation, and works by Zhong et al. [149] and Higuera et al. [150] rendering tactile images. The results indicate that multi-modal research based on NeRF is a worthwhile direction for further research. The reasons can be qualitatively understood as visually challenging scenes may not be difficult when combined with other senses. For example, visual perception may cause significant errors or difficulties when pouring water into a container due to opaque materials and potential occlusions by robotic arms. In contrast, different voice tones produced by containers with varying water levels can be utilized to determine if containers are filled. Therefore, integrating multi-modal scene perception and understanding is an important emerging direction, as we expect perceptive information to enhance, complement, and validate each other mutually, ultimately leading to robots adapting to complex real-world scenarios.

VI. CONCLUSION

The representation of Neural Radiance Fields (NeRF) provides new options for the field of robotics, as a way to understand and interact with scenes. Specifically, NeRF offers a reliable choice for many sub-tasks in robotics, such as scene understanding, reconstruction, dynamic perception, scene editing, object modelling, navigation, and manipulation guidance. NeRF’s potential to enhance reality, efficiency, generalizability, etc., has yet to be fully explored, and this could strengthen the bond between NeRF and robotics. However, integrating NeRF with robotics brings various challenges that need to be addressed, and there are unknown territories waiting to be explored in this field. In this survey, we propose some promising research directions and initial ideas to inspire further investigation. By summarizing the outstanding works in this field and highlighting its potential, we hope to encourage more researchers to explore new possibilities and successfully implement them in real robot platforms.

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 *ECCV*, 2020.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science robotics*, p. eabc5986, 2020.
- [3] K. Elia, B. Leonard, L. Antonio, M. Matthias, K. Vladlen, and S. Davide, “Champion-level drone racing using deep reinforcement learning,” *Nature*, p. 982–987, 2023.
- [4] A. I. Károly, P. Galambos, J. Kuti, and I. J. Rudas, “Deep learning in robotics: Survey on model structures and training strategies,” *SMCS*, pp. 266–279, 2020.
- [5] J. T. Kajiya and B. P. Von Herzen, “Ray tracing volume densities,” *ACM SIGGRAPH*, pp. 165–174, 1984.
- [6] 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 *CVPR*, 2022, pp. 8248–8258.
- [7] 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 *CVPR*, 2022, pp. 12 786–12 796.
- [8] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, “Vision-only robot navigation in a neural radiance world,” *RA-L*, pp. 4606–4613, 2022.
- [9] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, “Loc-nerf: Monte carlo localization using neural radiance fields,” *ICRA*, 2023.
- [10] N. M. M. Shafullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “Clip-fields: Weakly supervised semantic fields for robotic memory,” *RSS*, 2023.
- [11] Z. Hu, R. Tan, Y. Zhou, J. Woon, and C. Lv, “Template-based category-agnostic instance detection for robotic manipulation,” *RA-L*, pp. 12 451–12 458, 2022.
- [12] Z. Zhu, Y. Chen, Z. Wu, C. Hou, Y. Shi, C. Li, P. Li, H. Zhao, and G. Zhou, “Latitude: Robotic global localization with truncated dynamic low-pass filter in city-scale nerf,” *ICRA*, 2022.
- [13] 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 *CVPR*, 2022, pp. 12 871–12 881.
- [14] F. Dellaert and L. Yen-Chen, “Neural volume rendering: Nerf and beyond,” *arXiv preprint arXiv:2101.05204*, 2020.
- [15] 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 *CGF*, 2022, pp. 641–676.
- [16] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, “Nerf: Neural radiance field in 3d vision, a comprehensive review,” *TPAMI*, 2022.
- [17] A. Rabby and C. Zhang, “Beyondpixels: A comprehensive review of the evolution of neural radiance fields,” *arXiv preprint arXiv:2306.03000*, 2023.
- [18] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “Nerf++: Analyzing and improving neural radiance fields,” *arXiv preprint arXiv:2010.07492*, 2020.
- [19] J. Sun, X. Chen, Q. Wang, Z. Li, H. Averbuch-Elor, X. Zhou, and N. Snavely, “Neural 3d reconstruction in the wild,” in *ACM SIGGRAPH*, 2022, pp. 1–9.
- [20] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, “imap: Implicit mapping and positioning in real-time,” in *ICCV*, 2021, pp. 6229–6238.
- [21] E. Krushkov, A. Savinykh, P. Karpyshov, M. Kurenkov, E. Yudin, A. Potapov, and D. Tsetserukou, “Meslam: Memory efficient slam based on neural fields,” in *SMC*, 2022, pp. 430–435.
- [22] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, “Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation,” in *ISMAR*, 2022, pp. 499–507.
- [23] D. Lisis and C. Holmes, “Towards open world nerf-based slam,” *CRV*, 2023.
- [24] M. M. Johari, C. Carta, and F. Fleuret, “Eslam: Efficient dense slam system based on hybrid representation of signed distance fields,” in *CVPR*, 2023, pp. 17 408–17 419.
- [25] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, “Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping,” in *ICRA*, 2023, pp. 9400–9406.
- [26] 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.
- [27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molnyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *ISMAR*, 2011, pp. 127–136.
- [28] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, “Real-time camera tracking and 3d reconstruction using signed distance functions,” in *RSS*, 2013.
- [29] M. Oechsle, S. Peng, and A. Geiger, “Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction,” in *ICCV*, 2021, pp. 5589–5599.
- [30] 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,” (*NeurIPS*, 2021).
- [31] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, “Neural rgbd surface reconstruction,” in *CVPR*, 2022, pp. 6290–6301.
- [32] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, “Volume rendering of neural implicit surfaces,” *NeurIPS*, pp. 4805–4815, 2021.

- [33] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman, “Stylesdf: High-resolution 3d-consistent image and geometry generation,” in *CVPR*, 2022, pp. 13 503–13 513.
- [34] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, “Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction,” *NeurIPS*, pp. 25 018–25 032, 2022.
- [35] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou, “Neural 3d scene reconstruction with the manhattan-world assumption,” in *CVPR*, 2022, pp. 5511–5520.
- [36] K. Li, Y. Tang, V. A. Prisacariu, and P. H. Torr, “Bnv-fusion: Dense 3d reconstruction using bi-level neural volume fusion,” in *CVPR*, 2022, pp. 6166–6175.
- [37] Y. Ming, W. Ye, and A. Calway, “Idf-slam: End-to-end rgb-d slam with neural implicit mapping and deep feature tracking,” *arXiv preprint arXiv:2209.07919*, 2022.
- [38] M. El Banani, L. Gao, and J. Johnson, “Unsuperviseddr&r: Unsupervised point cloud registration via differentiable rendering,” in *CVPR*, 2021, pp. 7129–7139.
- [39] 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,” *3DV*, 2023.
- [40] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *CVPR*, 2021, pp. 7210–7219.
- [41] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, “Urban radiance fields,” in *CVPR*, 2022, pp. 12 932–12 942.
- [42] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, “Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields,” *RA-L*, 2022.
- [43] Y. Ran, J. Zeng, S. He, J. Chen, L. Li, Y. Chen, G. Lee, and Q. Ye, “Neuar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations,” *RA-L*, pp. 1125–1132, 2023.
- [44] J. Zeng, Y. Li, Y. Ran, S. Li, F. Gao, L. Li, S. He, J. Chen, and Q. Ye, “Efficient view path planning for autonomous implicit reconstruction,” in *ICRA*, 2023, pp. 4063–4069.
- [45] P. Marza, L. Matignon, O. Simonin, D. Batra, C. Wolf, and D. S. Chaplot, “Autonerf: Training implicit scene representations with autonomous agents,” *ICLR*, 2024.
- [46] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, “Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video,” in *ICCV*, 2021, pp. 12 959–12 970.
- [47] Z. Li, S. Niklaus, N. Snavely, and O. Wang, “Neural scene flow fields for space-time view synthesis of dynamic scenes,” in *CVPR*, 2021, pp. 6498–6508.
- [48] W. Yuan, Z. Lv, T. Schmidt, and S. Lovegrove, “Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering,” in *CVPR*, 2021, pp. 13 144–13 152.
- [49] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, “Space-time neural irradiance fields for free-viewpoint video,” in *CVPR*, 2021, pp. 9421–9431.
- [50] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe *et al.*, “Neural 3d video synthesis from multi-view video,” in *CVPR*, 2022, pp. 5521–5531.
- [51] J. Ost, F. Mannan, N. Thuerey, J. Knott, and F. Heide, “Neural scene graphs for dynamic scenes,” in *CVPR*, 2021, pp. 2856–2865.
- [52] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, “D-nerf: Neural radiance fields for dynamic scenes,” in *CVPR*, 2021, pp. 10 318–10 327.
- [53] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, “Nerfies: Deformable neural radiance fields,” in *ICCV*, 2021, pp. 5865–5874.
- [54] 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 TOG*, 2021.
- [55] Z. Yan, C. Li, and G. H. Lee, “Nerf-ds: Neural radiance fields for dynamic specular objects,” in *CVPR*, 2023, pp. 8285–8295.
- [56] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, “D~2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video,” *NeurIPS*, pp. 32 653–32 666, 2022.
- [57] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, “Fast dynamic radiance fields with time-aware neural voxels,” in *SIGGRAPH*, 2022, pp. 1–9.
- [58] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, and J.-B. Huang, “Robust dynamic radiance fields,” *CVPR*, 2023.
- [59] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *CVPR*, 2016, pp. 4104–4113.
- [60] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, “Dynamic view synthesis from dynamic monocular video,” in *ICCV*, 2021, pp. 5712–5721.
- [61] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, “Suds: Scalable urban dynamic scenes,” in *CVPR*, 2023, pp. 12 375–12 385.
- [62] M. You and J. Hou, “Decoupling dynamic monocular videos for dynamic view synthesis,” *arXiv preprint arXiv:2304.01716*, 2023.
- [63] Y. Du, Y. Zhang, H.-X. Yu, J. B. Tenenbaum, and J. Wu, “Neural radiance flow for 4d view synthesis and video processing,” in *ICCV*, 2021, pp. 14 304–14 314.
- [64] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *ICCV*, 2021, pp. 9650–9660.
- [65] J. Cen, Z. Zhou, J. Fang, W. Shen, L. Xie, D. Jiang, X. Zhang, Q. Tian *et al.*, “Segment anything in 3d with nerfs,” *NeurIPS*, pp. 25 971–25 990, 2023.
- [66] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, “In-place scene labelling and understanding with implicit scene representation,” in *ICCV*, 2021, pp. 15 838–15 847.
- [67] H.-X. Yu, L. J. Guibas, and J. Wu, “Unsupervised discovery of object radiance fields,” *ICLR*, 2022.
- [68] S. Liang, Y. Liu, S. Wu, Y.-W. Tai, and C.-K. Tang, “Onerf: Unsupervised 3d object segmentation from multiple views,” *arXiv preprint arXiv:2211.12038*, 2022.
- [69] S. Kobayashi, E. Matsumoto, and V. Sitzmann, “Decomposing nerf for editing via feature field distillation,” *NeurIPS*, pp. 23 311–23 330, 2022.
- [70] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, “Neural feature fusion fields: 3d distillation of self-supervised 2d image representations,” *3DV*, 2022.
- [71] 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 *ICML*, 2021, pp. 8748–8763.
- [72] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, “Language-driven semantic segmentation,” in *ICLR*, 2022.
- [73] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” *arXiv:2304.02643*, 2023.
- [74] V. Tschernezki, D. Larlus, and A. Vedaldi, “Neuraldiff: Segmenting 3d objects that move in egocentric videos,” in *3DV*, 2021, pp. 910–919.
- [75] S. Vora, N. Radwan, K. Greff, H. Meyer, K. Genova, M. S. Sajjadi, E. Pot, A. Tagliasacchi, and D. Duckworth, “Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes,” *TMLR*, 2022.
- [76] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *MICCAI*, 2016, pp. 424–432.
- [77] S. Zhi, E. Sucar, A. Mouton, I. Haughton, T. Laidlow, and A. J. Davison, “Ilabel: Interactive neural scene labelling,” *arXiv preprint arXiv:2111.14637*, 2021.
- [78] K. Blomqvist, L. Ott, J. J. Chung, and R. Siegwart, “Baking in the feature: Accelerating volumetric segmentation by rendering feature maps,” *arXiv preprint arXiv:2209.12744*, 2022.
- [79] Z. Liu, F. Milano, J. Frey, M. Hutter, R. Siegwart, H. Blum, and C. Cadena, “Unsupervised continual semantic adaptation through neural rendering,” *CVPR*, 2023.
- [80] S. Zhu, G. Wang, H. Blum, J. Liu, L. Song, M. Pollefeys, and H. Wang, “Sni-slam: Semantic neural implicit slam,” *CVPR*, 2024.
- [81] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, “Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation,” in *CVPR*, 2020, pp. 12 475–12 485.
- [82] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, “Panoptic segmentation,” in *CVPR*, 2019, pp. 9404–9413.
- [83] 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,” *3DV*, 2022.
- [84] Y. Liao, J. Xie, and A. Geiger, “KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *TPAMI*, pp. 3292–3310, 2022.
- [85] S. Liu, X. Zhang, Z. Zhang, R. Zhang, J.-Y. Zhu, and B. Russell, “Editing conditional radiance fields,” in *ICCV*, 2021, pp. 5773–5783.

- [86] Y. Li, Z.-H. Lin, D. Forsyth, J.-B. Huang, and S. Wang, “Climatenerf: Extreme weather synthesis in neural radiance field,” in *ICCV*, 2023, pp. 3227–3238.
- [87] A. Mirzaei, T. Aumentado-Armstrong, K. G. Derpanis, J. Kelly, M. A. Brubaker, I. Gilitschenski, and A. Levinstein, “Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields,” in *CVPR*, 2023, pp. 20 669–20 679.
- [88] W. Jang and L. Agapito, “Codenerf: Disentangled neural radiance fields for object categories,” in *ICCV*, 2021, pp. 12 949–12 958.
- [89] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, “Clip-nerf: Text-and-image driven manipulation of neural radiance fields,” in *CVPR*, 2022, pp. 3835–3844.
- [90] C. Bao, Y. Zhang, B. Yang, T. Fan, Z. Yang, H. Bao, G. Zhang, and Z. Cui, “Sine: Semantic-driven image-based nerf editing with prior-guided editing field,” in *CVPR*, 2023, pp. 20 919–20 929.
- [91] K. Tertikas, P. Despoina, B. Pan, J. J. Park, M. A. Uy, I. Emiris, Y. Avrithis, and L. Guibas, “Partnerf: Generating part-aware editable 3d shapes without 3d supervision,” *CVPR*, 2023.
- [92] T. Xu and T. Harada, “Deforming radiance fields with cages,” in *ECCV*, 2022, pp. 159–175.
- [93] Y. Peng, Y. Yan, S. Liu, Y. Cheng, S. Guan, B. Pan, G. Zhai, and X. Yang, “Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation,” *NeurIPS*, pp. 31 402–31 415, 2022.
- [94] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao, “Nerf-editing: geometry editing of neural radiance fields,” in *CVPR*, 2022, pp. 18 353–18 364.
- [95] O. Sorkine and M. Alexa, “As-rigid-as-possible surface modeling,” in *SGP*, 2007, pp. 109–116.
- [96] B. Yang, C. Bao, J. Zeng, H. Bao, Y. Zhang, Z. Cui, and G. Zhang, “Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing,” in *ECCV*, 2022, pp. 597–614.
- [97] J.-K. Chen, J. Lyu, and Y.-X. Wang, “Neuraleditor: Editing neural radiance fields via manipulating point clouds,” in *CVPR*, 2023, pp. 12 439–12 448.
- [98] A. Mirzaei, Y. Kant, J. Kelly, and I. Gilitschenski, “Laterf: Label and text driven object radiance fields,” in *ECCV*, 2022, pp. 20–36.
- [99] 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 *ICCV*, 2021, pp. 13 779–13 788.
- [100] H.-K. Liu, I. Shen, B.-Y. Chen *et al.*, “Nerf-in: Free-form nerf inpainting with rgb-d priors,” *CG&A*, pp. 100–109, 2022.
- [101] S. Weder, G. Garcia-Hernando, A. Monszpart, M. Pollefeys, G. Brostow, M. Firman, and S. Vicente, “Removing objects from NeRFs,” in *CVPR*, 2023.
- [102] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM TOG*, pp. 1–15, 2022.
- [103] Y. Chen, Q. Yuan, Z. Li, Y. L. W. W. C. Xie, X. Wen, and Q. Yu, “Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene,” *arXiv preprint arXiv:2208.07059*, 2022.
- [104] P.-Z. Chiang, M.-S. Tsai, H.-Y. Tseng, W.-S. Lai, and W.-C. Chiu, “Stylizing 3d scene via implicit representation and hypernetwork,” in *WACV*, 2022, pp. 1475–1484.
- [105] Y.-H. Huang, Y. He, Y.-J. Yuan, Y.-K. Lai, and L. Gao, “Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning,” in *CVPR*, 2022, pp. 18 342–18 352.
- [106] K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely, “Arf: Artistic radiance fields,” in *ECCV*, 2022, pp. 717–733.
- [107] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao, “Nerf-art: Text-driven neural radiance fields stylization,” *TVCG*, 2023.
- [108] 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 *IROS*, 2021, pp. 1323–1330.
- [109] 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.
- [110] S. Chen, Z. Wang, and V. Prisacariu, “Direct-posenet: absolute pose regression with photometric consistency,” in *3DV*, 2021, pp. 1175–1185.
- [111] A. Moreau, N. Piasco, D. Tsishkou, B. Stanculescu, and A. de La Fortelle, “Lens: Localization enhanced by nerf synthesis,” in *CoRL*, 2022, pp. 1347–1356.
- [112] H. Turki, D. Ramanan, and M. Satyanarayanan, “Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs,” in *CVPR*, 2022.
- [113] S. Chen, X. Li, Z. Wang, and V. Prisacariu, “DFNet: Enhance absolute pose regression with direct feature matching,” in *ECCV*, 2022.
- [114] H. Kuang, X. Chen, T. Guadagnino, N. Zimmerman, J. Behley, and C. Stachniss, “Ir-mcl: Implicit representation-based online global localization,” *RA-L*, 2022.
- [115] Y. Lin, T. Müller, J. Tremblay, B. Wen, S. Tyree, A. Evans, P. A. Vela, and S. Birchfield, “Parallel inversion of neural radiance fields for robust pose estimation,” in *ICRA*, 2023, pp. 9377–9384.
- [116] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *ICRA*, 1999, pp. 1322–1328.
- [117] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, “Barf: Bundle-adjusting neural radiance fields,” in *ICCV*, 2021, pp. 5741–5751.
- [118] Y. Xia, H. Tang, R. Timofte, and L. V. Gool, “Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction,” in *BMVC*, 2022.
- [119] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” *NeurIPS*, pp. 7462–7473, 2020.
- [120] Y. Shi, D. Rong, B. Ni, C. Chen, and W. Zhang, “Garf: Geometry-aware generalized neural radiance field,” *arXiv preprint arXiv:2212.02280*, 2022.
- [121] 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 *ICCV*, 2021, pp. 6351–6361.
- [122] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, “Self-calibrating neural radiance fields,” in *ICCV*, 2021.
- [123] W. Bian, Z. Wang, K. Li, J. Bian, and V. A. Prisacariu, “Nope-nerf: Optimising neural radiance field with no pose prior,” 2023.
- [124] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, “Sparf: Neural radiance fields from sparse and noisy poses,” in *CVPR*, 2023, pp. 4190–4200.
- [125] M. Tong, C. Dawson, and C. Fan, “Enforcing safety for vision-based controllers via control barrier functions and neural radiance fields,” *ICRA*, 2022.
- [126] A. Byravan, J. Humplík, L. Hasenclever, A. Brussee, F. Nori, T. Haarnoja, B. Moran, S. Bohez, F. Sadeghi, B. Vujatovic *et al.*, “Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields,” in *ICRA*, 2023, pp. 9362–9369.
- [127] M. Kurenkov, A. Potapov, A. Savinykh, E. Yudin, E. Kruzhkov, P. Karpyshov, and D. Tsetserukou, “Nfomp: Neural field for optimal motion planner of differential drive robots with nonholonomic constraints,” *RA-L*, pp. 10 991–10 998, 2022.
- [128] T. Chen, P. Culbertson, and M. Schwager, “Catnips: Collision avoidance through neural implicit probabilistic scenes,” *arXiv preprint arXiv:2302.12931*, 2023.
- [129] O. Kwon, J. Park, and S. Oh, “Renderable neural radiance map for visual navigation,” in *CVPR*, 2023, pp. 9099–9108.
- [130] P. Marza, L. Matignon, O. Simonin, and C. Wolf, “Multi-object navigation with dynamically learned neural implicit representations,” *arXiv preprint arXiv:2210.05129*, 2022.
- [131] F. Li, S. R. Vutukur, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, “Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation,” in *ICCV*, 2023, pp. 2123–2133.
- [132] W.-C. Tseng, H.-J. Liao, L. Yen-Chen, and M. Sun, “Cla-nerf: Category-level articulated neural radiance field,” in *ICRA*, 2022, pp. 8454–8460.
- [133] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, and A. Gaidon, “Shapo: Implicit representations for multi object shape appearance and pose optimization,” 2022.
- [134] H. Chen, F. Manhardt, N. Navab, and B. Busam, “Texpose: Neural texture learning for self-supervised 6d object pose estimation,” in *CVPR*, 2023, pp. 4841–4852.
- [135] B. Hu, J. Huang, Y. Liu, Y.-W. Tai, and C.-K. Tang, “Nerf-rpn: A general framework for object detection in nerfs,” in *CVPR*, 2023, pp. 23 528–23 538.
- [136] C. Xu, B. Wu, J. Hou, S. Tsai, R. Li, J. Wang, W. Zhan, Z. He, P. Vajda, K. Keutzer *et al.*, “Nerf-det: Learning geometry-aware volumetric representation for multi-view 3d object detection,” in *CVPR*, 2023, pp. 23 320–23 330.
- [137] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, “Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects,” in *CVPR*, 2023, pp. 606–617.
- [138] S. Lewis, J. Pavlasek, and O. C. Jenkins, “Narf22: Neural articulated radiance fields for configuration-aware rendering,” in *IROS*, 2022, pp. 770–777.

- [139] L. Chen, Y. Song, H. Bao, and X. Zhou, "Perceiving unseen 3d objects by poking the objects," *ICRA*, 2023.
- [140] Z. Tang, B. Sundaralingam, J. Tremblay, B. Wen, Y. Yuan, S. Tyree, C. Loop, A. Schwing, and S. Birchfield, "Rgb-only reconstruction of tabletop scenes for collision-free manipulator control," in *ICRA*, 2023, pp. 1778–1785.
- [141] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *CoRL*, 2022, pp. 112–123.
- [142] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *CoRL*, 2023, pp. 1755–1768.
- [143] W. Wang, A. S. Morgan, A. M. Dollar, and G. D. Hager, "Dynamical scene representation and control with keypoint-conditioned neural radiance field," in *CASE*, 2022, pp. 1138–1143.
- [144] Y.-C. Lin, P. Florence, A. Zeng, J. T. Barron, Y. Du, W.-C. Ma, A. Simeonov, A. R. Garcia, and P. Isola, "Mira: Mental imagery for robotic affordances," in *CoRL*, 2023, pp. 1916–1927.
- [145] B. Shen, Z. Jiang, C. Choy, L. J. Guibas, S. Savarese, A. Anandkumar, and Y. Zhu, "Acid: Action-conditional implicit visual dynamics for deformable object manipulation," *RSS*, 2022.
- [146] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *CoRL*, 2021.
- [147] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "Graspnerf: multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf," in *ICRA*, 2023, pp. 1757–1763.
- [148] 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 *CoRL*, 2022.
- [149] S. Zhong, A. Albini, O. P. Jones, P. Maiolino, and I. Posner, "Touching a nerf: Leveraging neural radiance fields for tactile sensory data generation," in *CoRL*, 2023, pp. 1618–1628.
- [150] C. Higuera, S. Dong, B. Boots, and M. Mukadam, "Neural contact fields: Tracking extrinsic contact with tactile sensing," in *ICRA*, 2023, pp. 12 576–12 582.
- [151] D. Driess, I. Schubert, P. Florence, Y. Li, and M. Toussaint, "Reinforcement learning with neural radiance fields," *NeurIPS*, 2022.
- [152] D. Shim, S. Lee, and H. J. Kim, "Snerl: Semantic-aware neural radiance fields for reinforcement learning," *ICML*, 2023.
- [153] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *ICRA*, 2022, pp. 6394–6400.
- [154] A. Simeonov, Y. Du, Y.-C. Lin, A. R. Garcia, L. P. Kaelbling, T. Lozano-Pérez, and P. Agrawal, "Se (3)-equivariant relational rearrangement with neural descriptor fields," in *CoRL*, 2023, pp. 835–846.
- [155] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," *ICRA*, 2023.
- [156] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "Nerf-supervision: Learning dense object descriptors from neural radiance fields," *ICRA*, 2022.
- [157] V. Blukis, K.-J. Yoon, T. Lee, J. Tremblay, B. Wen, I.-S. Kweon, D. Fox, and S. Birchfield, "One-shot neural fields for 3d object understanding," in *CVPRW*, 2023.
- [158] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," in *ICRA*, 2023, pp. 1814–1821.
- [159] N. Khargonkar, N. Song, Z. Xu, B. Prabhakaran, and Y. Xiang, "Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands," in *CoRL*, 2023, pp. 516–526.
- [160] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn, "Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis," in *PCVPR*, 2023, pp. 17 907–17 917.
- [161] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM TOG*, pp. 1–18, 2021.
- [162] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *CVPR*, 2021, pp. 7495–7504.
- [163] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang, "Hallucinated neural radiance fields in the wild," in *CVPR*, 2022, pp. 12 943–12 952.
- [164] V. Rudnev, M. Elgarhib, W. Smith, L. Liu, V. Golyanik, and C. Theobalt, "Nerf for outdoor scene relighting," in *ECCV*, 2022, pp. 615–631.
- [165] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler, "Neural fields meet explicit geometric representations for inverse rendering of urban scenes," in *CVPR*, 2023, pp. 8370–8380.
- [166] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Haşan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, "Neural reflectance fields for appearance acquisition," *arXiv preprint arXiv:2008.03824*, 2020.
- [167] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch, "Neural-pil: Neural pre-integrated lighting for reflectance decomposition," *NeurIPS*, pp. 10 691–10 704, 2021.
- [168] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *CVPR*, 2021, pp. 5453–5462.
- [169] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *ICCV*, 2021, pp. 12 684–12 694.
- [170] 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, pp. 5752–5761.
- [171] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *ICCV*, 2021, pp. 14 335–14 345.
- [172] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorrf: Tensorial radiance fields," in *ECCV*, 2022, pp. 333–350.
- [173] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *CVPR*, 2022, pp. 5470–5479.
- [174] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger, "Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks," in *CGF*, 2021, pp. 45–59.
- [175] M. Piala and R. Clark, "Terminerf: Ray termination prediction for efficient neural rendering," in *3DV*, 2021, pp. 1106–1114.
- [176] A. Dey, Y. Ahmine, and A. I. Comport, "Mip-nerf rgb-d: Depth assisted fast neural radiance fields," *WSCG*, 2022.
- [177] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *CVPR*, 2022, pp. 12 882–12 891.
- [178] H. Lin, S. Peng, Z. Xu, Y. Yan, Q. Shuai, H. Bao, and X. Zhou, "Efficient neural radiance fields for interactive free-viewpoint video," in *SIGGRAPH Asia*, 2022, pp. 1–9.
- [179] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *CVPR*, 2022, pp. 5459–5469.
- [180] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *ICCV*, 2021, pp. 5875–5884.
- [181] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *NeurIPS*, pp. 15 651–15 663, 2020.
- [182] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *CVPR*, 2022, pp. 5501–5510.
- [183] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *CVPR*, 2022, pp. 5438–5448.
- [184] D. Rebain, W. Jiang, S. Yazdani, K. Li, K. M. Yi, and A. Tagliasacchi, "Derf: Decomposed radiance fields," in *CVPR*, 2021, pp. 14 153–14 161.
- [185] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentim, "Fastnerf: High-fidelity neural rendering at 200fps," in *ICCV*, 2021, pp. 14 346–14 355.
- [186] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM TOG*, pp. 1–14, 2023.
- [187] 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," *ECCV*, 2022.
- [188] 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," in *CVPR*, 2023, pp. 20 637–20 647.
- [189] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," in *ICCV*, 2021, pp. 5885–5894.
- [190] 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 *CVPR*, 2022, pp. 5480–5490.
- [191] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, "Dense depth priors for neural radiance fields from sparse input views," in *CVPR*, 2022, pp. 12 892–12 901.

- [192] Y. C. Ahn, S. Jang, S. Park, J.-Y. Kim, and N. Kang, “Panerf: Pseudo-view augmentation for improved neural radiance fields based on few-shot inputs,” *arXiv preprint arXiv:2211.12758*, 2022.
- [193] Y.-J. Yuan, Y.-K. Lai, Y.-H. Huang, L. Kobbelt, and L. Gao, “Neural radiance fields from sparse rgb-d images for high-quality view synthesis,” *TPAMI*, 2022.
- [194] J. Charles, W. Abbeloos, D. O. Reino, and R. Cipolla, “Style2nerf: An unsupervised one-shot nerf for semantic 3d reconstruction.” in *BMVC*, 2022, p. 104.
- [195] D. Pavllo, D. J. Tan, M.-J. Rakotosaona, and F. Tombari, “Shape, pose, and appearance from a single image via bootstrapped radiance field inversion,” in *CVPR*, 2023, pp. 4391–4401.
- [196] 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 *ICCV*, 2021, pp. 5855–5864.
- [197] Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang, “S-nerf: Neural radiance fields for street views,” *ICLR*, 2023.
- [198] 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 *ECCV*, 2022, pp. 106–122.
- [199] A. Meuleman, Y.-L. Liu, C. Gao, J.-B. Huang, C. Kim, M. H. Kim, and J. Kopf, “Progressively optimized local radiance fields for robust view synthesis,” in *CVPR*, 2023.
- [200] J. Gu, M. Jiang, H. Li, X. Lu, G. Zhu, S. A. A. Shah, L. Zhang, and M. Bennamoun, “Ue4-nerf: Neural radiance field for real-time rendering of large-scale scene,” *NeurIPS*, 2023.
- [201] F. Lu, Y. Xu, G. Chen, H. Li, K.-Y. Lin, and C. Jiang, “Urban radiance field representation with deformable neural mesh primitives,” in *ICCV*, 2023, pp. 465–476.
- [202] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelnerf: Neural radiance fields from one or few images,” in *CVPR*, 2021, pp. 4578–4587.
- [203] A. Trevithick and B. Yang, “Grf: Learning a general radiance field for 3d representation and rendering,” in *ICCV*, 2021, pp. 15 182–15 192.
- [204] 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 *ICCV*, 2021, pp. 12 578–12 588.
- [205] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, “Stereo radiance fields (srif): Learning view synthesis for sparse views of novel scenes,” in *CVPR*, 2021, pp. 7911–7920.
- [206] X. Huang, Q. Zhang, Y. Feng, X. Li, X. Wang, and Q. Wang, “Local implicit ray function for generalizable radiance field representation,” in *CVPR*, 2023, pp. 97–107.
- [207] X. Li, C. Hong, Y. Wang, Z. Cao, K. Xian, and G. Lin, “Symmnerf: Learning to explore symmetry prior for single-view view synthesis,” in *ACCV*, 2022, pp. 1726–1742.
- [208] 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 *ICCV*, 2021, pp. 14 124–14 133.
- [209] 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 *CVPR*, 2022, pp. 7824–7833.
- [210] G. Gafni, J. Thies, M. Zollhofer, and M. Nießner, “Dynamic neural radiance fields for monocular 4d facial avatar reconstruction,” in *CVPR*, 2021, pp. 8649–8658.
- [211] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2face: Real-time face capture and reenactment of rgb videos,” in *CVPR*, 2016, pp. 2387–2395.
- [212] 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 *CVPR*, 2021, pp. 4690–4699.
- [213] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, “Physgaussian: Physics-integrated 3d gaussians for generative dynamics,” *CVPR*, 2024.
- [214] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal, “Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation,” *arXiv preprint arXiv:2403.03949*, 2024.
- [215] S. Hu, L. Yu, H. Lanqing, T. Hu, G. H. Lee, Z. Li *et al.*, “Masknerf: Masked neural radiance fields for sparse view synthesis,” 2022.
- [216] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni *et al.*, “Make-a-video: Text-to-video generation without text-video data,” *ICLR*, 2023.
- [217] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo *et al.*, “Improving image generation with better captions,” *Computer Science*, p. 8, 2023.
- [218] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh, “Video generation models as world simulators,” 2024. [Online]. Available: <https://openai.com/research/video-generation-models-as-world-simulators>
- [219] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, “Adnerf: Audio driven neural radiance fields for talking head synthesis,” in *ICCV*, 2021, pp. 5784–5794.



Guangming Wang (Member, IEEE) received the B.S. degree from Department of Automation from Central South University, Changsha, China, in 2018, and Ph.D. degree in Control Science and Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2023. He visited Department of Computer Science of ETH Zurich from 2022 to 2023. He is currently a Research Associate with the Department of Engineering, University of Cambridge, UK. His research interests include SLAM, 3D computer vision, and autonomous driving.

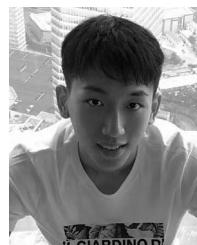


Lei Pan received the B.S. degree from the Xuhai college, China University of Mining and Technology, Xuzhou, China, in 2019. He is currently pursuing the M.S. degree in Information and Control Engineering with China University of Mining and Technology.

His research interests include NeRF and computer vision.



Songyou Peng received the Erasmus Mundus MSc degree in Computer Vision and Robotics, in 2017. Between 2016 and 2017, he spent some time doing research at INRIA Grenoble and Technical University of Munich. He received PhD from ETH Zurich and Max Planck Institute for Intelligent Systems. He is currently a Senior Researcher/Postdoc at ETH Zurich and also an incoming Research Scientist at Google Research. His research interests are computer vision and machine learning.



Shaohui Liu (Student Member, IEEE) received the BS degree from Tsinghua University, Beijing, China. He is currently working toward the direct doctorate degree from the Computer Science Department, ETH Zurich, Switzerland. His current research interests include 3D reconstruction, SLAM and visual localization. During 2017–2021 he has worked with Sensetime, Microsoft Research Asia and ByteDance as a research intern.



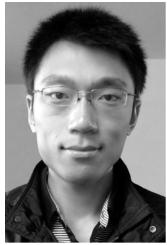
Chenfeng Xu (Student Member, IEEE) received the bachelor's degree from the Huazhong University of Science and Technology, in 2019. He is currently working toward the PhD degree with the Department of Mechanical Engineering, University of California, Berkeley. His research interests include deep learning, efficient 3D vision, and motion prediction.



Marc Pollefeys is a Prof. of Computer Science at ETH Zurich and Director of Science at Microsoft. He is best known for his work in 3D computer vision, but also for works on robotics, graphics, machine learning, and camera-based self-driving cars and drones. He received a M.Sc. and a PhD from the KU Leuven in Belgium in 1994 and 1999, respectively. He became an assistant professor at the University of North Carolina in Chapel Hill in 2002 and joined ETH Zurich as a full professor in 2007.



Yanzi Miao (Member, IEEE) received the Ph.D. degree in control science and engineering from the China University of Mining and Technology, Xuzhou, China, in 2009. She is a joint Ph.D. candidate and a Visiting Scholar with the Department of Informatics, University of Hamburg, Hamburg, Germany, in 2007 and 2017, respectively. She is a Professor with the School of Information and Control Engineering, China University of Mining and Technology. Her current research interests include intelligent perception and fusion, machine vision, and active olfaction.



Wei Zhan (Member, IEEE) received the Ph.D. degree from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 2019. He is currently an Assistant Professional Researcher with UC Berkeley and the Co-Director of the Berkeley DeepDrive Center. His research has been targeting scalable and interactive autonomy at the intersection of computer vision, machine learning, robotics, and control and intelligent transportation. He is the lead author of the INTERACTION dataset, and organized its prediction challenges in NeurIPS 2020 and ICCV

2021. His publications received the Best Student Paper Award in IV 2018 and Best Paper Award-Honorable Mention in IEEE ROBOTICS AND AUTOMATION LETTERS.



Hesheng Wang (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2002, and the M.Phil. and Ph.D. degrees in automation and computer-aided engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2007, respectively.

He is currently a Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual servoing, service robot, computer vision, and au-

tonomous driving.

Dr. Wang is an Associate Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE ROBOTICS AND AUTOMATION LETTERS, Robotic Intelligence and Automation, and the International Journal of Humanoid Robotics, and a Senior Editor for the IEEE/ASME TRANSACTIONS ON MECHATRONICS, an Editor of Conference Editorial Board (CEB) of IEEE Robotics and Automation Society. From 2015 to 2019, he was an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS, and from 2020 to 2023, a Technical Editor for IEEE/ASME TRANSACTIONS ON MECHATRONICS. He was the General Chair of IEEE ROBIO 2022 and IEEE RCAR 2016, and the Program Chair of the IEEE ROBIO 2014 and IEEE/ASME AIM 2019. He will be the General Chair of IEEE/RSJ IROS 2025.



Masayoshi Tomizuka (Life Fellow, IEEE) received the Ph.D. degree in mechanical engineering from MIT in February 1974. In 1974, he joined as the Faculty Member of the Department of Mechanical Engineering at the University of California at Berkeley, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests include optimal and adaptive control, digital control, signal processing, motion control, and control problems related to robotics, precision motion control and vehicles. He was the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF from 2002 to 2004. He was a Technical Editor of the ASME Journal of Dynamic Systems, Measurement and Control (J-DSMC) from 1988 to 1993, and the Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS from 1997 to 1999.

He is a Fellow of the ASME and IFAC. He was a recipient of the Charles Russ Richards Memorial Award (ASME), in 1997, the Rufus Oldenburger Medal (ASME), in 2002, and the John R. Ragazzini Award in 2006.